

Linux.Fibel.org  
Version 0.6.0

Ole Vanhoefer

1. September 2004

# Linux.Fibel.org

## Einführung in Linux / LPI I

von Ole Vanhoefer

Copyright © 2001-2004 Ole Vanhoefer  
Gesetzt mit L<sup>A</sup>T<sub>E</sub>X unter Linux

Version 0.3 : 18. November 2001  
Version 0.4 : 14. April 2002  
Version 0.5 : 19. Januar 2003  
Version 0.6.0 : 1. September 2004

### Das Kleingedruckte

Dieses Skript ist wie alle Werke urheberrechtlich geschützt. Er ist jedoch unter den Bedingungen der Open Publication License, Version 0.4 oder höher verfügbar. Die genaue Lizenz findet sich in Open Publication License (siehe Anhang D, Seite 489).

Wenn dieses Skript reproduziert oder verwendet wird, bittet der Autor um Meldung eines solchen Angebotes per eMail an [linux@vanhoefer.de](mailto:linux@vanhoefer.de) unter Angabe einer Kontaktadresse.

Die in diesem Skript dargestellten Programme und Verfahren werden ohne Berücksichtigung der Patentlage mitgeteilt. Sie sind nur für Amateur- und Lehrzwecke bestimmt.

Alle Informationen in diesem Skript sind frei erfunden. Ähnlichkeiten mit existierenden Betriebssystemen, Soft- und Hardware sind rein zufällig. Daher übernimmt der Autor keine Garantie, juristische Verantwortung oder irgendeine Haftung für Folgen, die auf Inhalte dieses Skriptes zurückgehen.

Ich weise darauf hin, daß die im Skript verwendeten Soft- und Hardwarebezeichnungen und Markennamen der jeweiligen Firmen i. A. warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Die Nutzungsrechte der in diesem Skript wiedergegebene Codezeilen von Programmen, die unter der GNU General Public License verbreitet werden, richtet sich einzig und allein nach den Bedingungen der GNU General Public License.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>21</b>
1.1	Was ist eigentlich Linux? . . . . .	21
1.1.1	Leistungsmerkmale des Kernels . . . . .	21
1.2	Wie alles begann . . . . .	22
1.3	Grundlagen für die Installation . . . . .	24
1.3.1	Hardwarevoraussetzungen . . . . .	24
1.3.2	Die Festplatte . . . . .	25
1.3.3	Partitionen . . . . .	25
1.3.4	Der Auslagerungsspeicher . . . . .	28
1.3.5	fips . . . . .	28
1.4	Installation . . . . .	28
1.4.1	Erstellung der Bootdisketten unter DOS . . . . .	28
1.4.2	Installation mit Bootdisketten . . . . .	29
1.5	Konfiguration mit YaST2 . . . . .	30
1.5.1	Wie installiere ich weitere Softwarepakete? . . . . .	32
1.5.2	Wie mache ich ein Online-Update? . . . . .	33
1.6	Benutzer- und Gruppenverwaltung . . . . .	34
1.6.1	Wie erstelle, bearbeite und lösche ich Benutzer? . . . . .	34
1.6.2	Wie erstelle, bearbeite und lösche ich Gruppen? . . . . .	35
1.7	Arbeitsblatt I: Erste Schritte . . . . .	39
<b>2</b>	<b>X-Window und KDE</b>	<b>41</b>
2.1	X-Window-System . . . . .	41
2.2	KDE . . . . .	41
2.2.1	KDE-Kontrollzentrum: KControl . . . . .	42
2.2.2	KDE-Systemüberwachung: KSysGuard . . . . .	42
2.3	Wie mache ich was? . . . . .	43
2.3.1	Wie lege ich ein Programmicon auf dem Desktop an? . . . . .	43
2.3.2	Programm unter einem anderen Benutzer ausführen . . . . .	43
2.3.3	KDE su . . . . .	44
2.4	KDE-Programme . . . . .	44
2.4.1	Konquerer . . . . .	44
2.4.2	KEdit und KWrite . . . . .	44

2.4.3	KMail . . . . .	44
2.4.4	KSnapshott . . . . .	44
2.4.5	KOffice . . . . .	45
2.4.6	Weitere KDE-Programme . . . . .	45
2.5	Weitere X-Window-Programme . . . . .	46
2.6	Arbeitsblatt II: Erste Schritte . . . . .	47
<b>3</b>	<b>Arbeiten auf der Shell</b>	<b>49</b>
3.1	Starten der Shell . . . . .	49
3.1.1	Linux-Konsole . . . . .	49
3.1.2	X-Terminal . . . . .	50
3.2	Erste Befehle . . . . .	51
3.2.1	Anzeigen des Verzeichnisinhalts . . . . .	51
3.2.2	Erstellen und Löschen von Verzeichnissen und Dateien . . . . .	56
3.2.3	Kopieren, Verschieben und Umbenennen von Dateien . . . . .	58
3.3	Informationen und Hilfe . . . . .	61
3.3.1	Hilfe im Befehl . . . . .	61
3.3.2	Manual-Pages . . . . .	62
3.4	Arbeitsblatt III: Einführung in die Shell . . . . .	65
<b>4</b>	<b>Die Shell I</b>	<b>67</b>
4.1	Was ist eine Shell? . . . . .	67
4.1.1	chsh . . . . .	68
4.2	Die Bash . . . . .	68
4.2.1	Kommandosyntax . . . . .	70
4.3	Arbeiten mit Verzeichnissen . . . . .	70
4.3.1	pwd . . . . .	70
4.3.2	cd . . . . .	71
4.3.3	ls . . . . .	71
4.3.4	dir . . . . .	73
4.3.5	vdir . . . . .	73
4.3.6	mkdir . . . . .	73
4.3.7	rmdir . . . . .	74
4.4	Der Linux-Verzeichnisbaum . . . . .	74
4.5	Arbeiten mit Dateien . . . . .	77
4.5.1	touch . . . . .	77
4.5.2	cat . . . . .	78
4.5.3	cp . . . . .	78
4.5.4	dd . . . . .	79
4.5.5	Erstellung der Bootdisketten mit dd . . . . .	79
4.5.6	mv . . . . .	79
4.5.7	rm . . . . .	80
4.5.8	Dateibezeichnung mit Jokerzeichen . . . . .	80

4.5.9	more . . . . .	82
4.5.10	Der Pager <code>less</code> . . . . .	82
4.5.11	file . . . . .	83
4.5.12	/etc/magic . . . . .	83
4.6	Weitere Befehle . . . . .	84
4.6.1	clear . . . . .	84
4.6.2	cal . . . . .	84
4.6.3	date . . . . .	84
4.6.4	hwclock . . . . .	86
4.6.5	echo . . . . .	87
4.6.6	Farbcode fürs Terminal . . . . .	87
4.6.7	logout . . . . .	87
4.6.8	script . . . . .	88
4.6.9	telnet . . . . .	88
4.6.10	tty . . . . .	88
4.7	Arbeitsblatt IV: Shell 1 . . . . .	89
<b>5</b>	<b>Die Shell II</b>	<b>91</b>
5.1	Die gewissen Extras . . . . .	91
5.1.1	Readline Library . . . . .	91
5.1.2	inputrc . . . . .	91
5.1.3	Terminaleinstellungen mit <code>stty</code> . . . . .	92
5.1.4	Automatische Kommandoergänzung . . . . .	94
5.1.5	Gruppierung von Kommandos . . . . .	94
5.1.6	Bedingte Ausführung . . . . .	95
5.1.7	Substituierung von Kommandos . . . . .	95
5.2	Variablen . . . . .	95
5.2.1	Benutzervariablen . . . . .	96
5.2.2	export . . . . .	96
5.2.3	Umgebungsvariablen . . . . .	97
5.2.4	printenv . . . . .	97
5.2.5	env . . . . .	97
5.2.6	Suchpfad . . . . .	97
5.2.7	Prompt . . . . .	98
5.2.8	alias . . . . .	98
5.2.9	unalias . . . . .	99
5.3	History-Liste . . . . .	99
5.3.1	history . . . . .	99
5.3.2	fc . . . . .	100
5.4	Umleitungen der Eingaben und Ausgaben . . . . .	100
5.4.1	Umleitung der Eingabe und Ausgabe . . . . .	100
5.4.2	Pipelines . . . . .	102
5.4.3	tee . . . . .	102

5.4.4	xargs . . . . .	103
5.5	Arbeitsblatt V: Shell 2 . . . . .	105
<b>6</b>	<b>Hilfe und Dokumentation</b>	<b>109</b>
6.1	Lokale Dokumentation . . . . .	109
6.1.1	Das Handbuch: Manual-Pages . . . . .	109
6.1.2	man . . . . .	111
6.1.3	manpath . . . . .	112
6.1.4	TexInfo: Informationen mit info . . . . .	112
6.1.5	help . . . . .	113
6.1.6	HOWTOs . . . . .	114
6.1.7	FAQ . . . . .	114
6.1.8	Programmdokumentation . . . . .	115
6.2	Internetquellen . . . . .	115
6.2.1	Newsgroups . . . . .	116
6.2.2	KNode . . . . .	116
6.2.3	Mailinglisten . . . . .	117
6.3	Suchen nach Informationen . . . . .	117
6.3.1	whereis . . . . .	117
6.3.2	which . . . . .	118
6.3.3	whatis . . . . .	118
6.3.4	mandb . . . . .	118
6.3.5	apropos . . . . .	118
6.3.6	type . . . . .	118
6.4	Dokumentation und Support . . . . .	119
6.4.1	Dokumentationen schreiben . . . . .	119
6.4.2	Benutzer-Support . . . . .	119
6.5	Arbeitsblatt VI: Dokumentation und Hilfe . . . . .	121
<b>7</b>	<b>Textfilter</b>	<b>123</b>
7.1	Ausgabe von ganzen Dateien . . . . .	123
7.1.1	tac . . . . .	123
7.1.2	nl . . . . .	123
7.1.3	od . . . . .	124
7.2	Textformatierung . . . . .	124
7.2.1	fmt . . . . .	124
7.2.2	pr . . . . .	125
7.2.3	fold . . . . .	126
7.3	Teilen von Texten . . . . .	126
7.3.1	head . . . . .	126
7.3.2	tail . . . . .	126
7.3.3	split . . . . .	127
7.4	Textstatistik . . . . .	127

7.4.1	wc . . . . .	127
7.4.2	sum . . . . .	128
7.4.3	cksum . . . . .	128
7.5	Sortieren . . . . .	128
7.5.1	sort . . . . .	128
7.5.2	comm . . . . .	129
7.5.3	uniq . . . . .	129
7.6	Zeilenoperationen . . . . .	129
7.6.1	cut . . . . .	129
7.6.2	join . . . . .	130
7.6.3	paste . . . . .	130
7.7	Suchen und Ersetzen . . . . .	130
7.7.1	grep . . . . .	130
7.7.2	egrep . . . . .	131
7.7.3	fgrep . . . . .	131
7.7.4	tr . . . . .	131
7.7.5	expand und unexpand . . . . .	132
7.7.6	sed . . . . .	132
7.7.7	diff . . . . .	134
7.7.8	patch . . . . .	135
7.7.9	Praxisbeispiel: Patchen von Programmen . . . . .	135
7.8	Arbeitsblatt VII: Textverarbeitung und Textfilter . . . . .	141
<b>8</b>	<b>Benutzerverwaltung</b>	<b>143</b>
8.1	Benutzer . . . . .	143
8.1.1	Der Superuser root . . . . .	143
8.1.2	su . . . . .	143
8.2	Das Benutzerkonto . . . . .	144
8.2.1	/etc/passwd . . . . .	144
8.2.2	Besondere Benutzerkonten . . . . .	146
8.2.3	Benutzer am Einloggen hindern . . . . .	147
8.2.4	passwd . . . . .	148
8.2.5	chpasswd . . . . .	148
8.2.6	/etc/login.defs . . . . .	148
8.3	Einrichten eines Benutzers . . . . .	151
8.3.1	Bearbeiten der Datei /etc/passwd . . . . .	151
8.3.2	useradd . . . . .	151
8.3.3	usermod . . . . .	152
8.3.4	Entfernen eines Benutzers . . . . .	152
8.3.5	userdel . . . . .	153
8.3.6	Einrichten eines zweiten Superusers . . . . .	153
8.4	Gruppen . . . . .	153
8.4.1	/etc/group . . . . .	154

8.4.2	Besondere Gruppen . . . . .	154
8.4.3	id . . . . .	155
8.4.4	groups . . . . .	155
8.4.5	newgrp . . . . .	155
8.4.6	Anlegen von Gruppen . . . . .	156
8.4.7	groupadd . . . . .	156
8.4.8	gpasswd . . . . .	156
8.4.9	groupmod . . . . .	156
8.4.10	Löschen einer Gruppe . . . . .	157
8.4.11	groupdel . . . . .	157
8.5	Shadow-Paßwort-System . . . . .	157
8.5.1	/etc/shadow . . . . .	157
8.5.2	/etc/gshadow . . . . .	158
8.5.3	pwconv . . . . .	158
8.5.4	pwunconv . . . . .	158
8.5.5	grpconv . . . . .	158
8.5.6	grpunconv . . . . .	158
8.6	Startdateien des Benutzers . . . . .	159
8.6.1	/etc/profile . . . . .	159
8.6.2	/etc/bashrc . . . . .	160
8.6.3	sh . . . . .	161
8.7	Das "Who-is-Who" der Benutzer . . . . .	161
8.7.1	who . . . . .	161
8.7.2	whoami . . . . .	161
8.7.3	logname . . . . .	162
8.7.4	w . . . . .	162
8.7.5	finger . . . . .	162
8.7.6	last . . . . .	163
8.7.7	lastlog . . . . .	163
8.8	Arbeitsblatt VIII: Benutzerverwaltung . . . . .	165
<b>9</b>	<b>Rechte im Linux-Dateisystem</b>	<b>167</b>
9.1	Die Rechte . . . . .	167
9.1.1	Rechte auf Dateien . . . . .	167
9.1.2	Rechte auf Verzeichnisse . . . . .	168
9.1.3	Spezielle Rechte: SUID und SGID . . . . .	169
9.1.4	Skript: Formatieren von Disketten . . . . .	170
9.1.5	Spezielle Rechte: Sticky Bit . . . . .	170
9.2	Verwalten und Setzen von Rechten . . . . .	171
9.2.1	chown . . . . .	171
9.2.2	chgrp . . . . .	171
9.2.3	chmod . . . . .	172
9.2.4	umask . . . . .	172



9.3	Mit anderen Rechten arbeiten . . . . .	173
9.3.1	sudo . . . . .	173
9.3.2	/etc/sudoers . . . . .	174
9.3.3	visudo . . . . .	175
9.4	Arbeitsblatt IX: Rechte . . . . .	177
9.5	Arbeitsblatt X: Datei-Rechte . . . . .	179
9.6	Arbeitsblatt XI: Fallstudie: Rechte . . . . .	181
<b>10</b>	<b>Partitionen und Dateien</b>	<b>183</b>
10.1	Partitionen . . . . .	183
10.1.1	fdisk . . . . .	183
10.2	Master Boot Record . . . . .	185
10.2.1	Aufbau des MBR . . . . .	185
10.2.2	MBR anschauen . . . . .	186
10.3	Mounten . . . . .	187
10.3.1	mount . . . . .	187
10.3.2	remount . . . . .	189
10.3.3	umount . . . . .	189
10.3.4	sync . . . . .	189
10.3.5	/etc/fstab . . . . .	189
10.3.6	/etc/mtab . . . . .	190
10.4	Das Dateisystem . . . . .	190
10.4.1	mkfs . . . . .	191
10.4.2	fdformat . . . . .	192
10.5	Swap: Der Auslagerungsspeicher . . . . .	192
10.5.1	Swap-Space einrichten . . . . .	193
10.5.2	mkswap . . . . .	196
10.5.3	swapon . . . . .	196
10.5.4	swapoff . . . . .	197
10.6	Verwaltung . . . . .	197
10.6.1	mkfifo . . . . .	198
10.6.2	Dateinamen . . . . .	198
10.6.3	Die Datei . . . . .	199
10.6.4	stat . . . . .	200
10.7	Links . . . . .	200
10.7.1	Harte Links . . . . .	200
10.7.2	Symbolische Links . . . . .	201
10.7.3	ln . . . . .	201
10.7.4	Der Befehl cp und die Links . . . . .	202
10.8	Festplattennutzung . . . . .	202
10.8.1	du . . . . .	203
10.8.2	df . . . . .	203
10.8.3	Aufräumen des Systems . . . . .	204

10.9	Arbeitsblatt XII: Dateisystem 1 . . . . .	207
10.10	Arbeitsblatt XIII: Arbeiten mit Disketten . . . . .	209
<b>11</b>	<b>Dateisysteme und Disk Quotas . . . . .</b>	<b>211</b>
11.1	Aufbau des ext2-Dateisystems . . . . .	211
11.1.1	Bootblock . . . . .	211
11.1.2	Blockgruppe . . . . .	211
11.1.3	Superblock . . . . .	212
11.1.4	dumpe2fs . . . . .	212
11.1.5	tune2fs . . . . .	213
11.1.6	Liste der Blockgruppenbeschreibungen . . . . .	213
11.1.7	Block- und Inode-Bitmap . . . . .	214
11.1.8	Inode-Liste . . . . .	214
11.2	Pflege des Dateisystems . . . . .	215
11.2.1	fsck . . . . .	215
11.2.2	e2fsck . . . . .	215
11.2.3	badblocks . . . . .	216
11.3	Auf der Suche nach der verlorenen Datei . . . . .	216
11.3.1	find . . . . .	216
11.3.2	Suchen mit locate . . . . .	218
11.4	Disk Quotas . . . . .	219
11.4.1	quota . . . . .	220
11.4.2	quotaon . . . . .	220
11.4.3	quotaoff . . . . .	221
11.4.4	edquota . . . . .	221
11.4.5	repquota . . . . .	222
11.4.6	quotacheck . . . . .	222
11.4.7	quotastats . . . . .	222
11.4.8	Einrichten von Disk Quotas mit SuSE . . . . .	222
11.5	Arbeitsblatt XIV: Dateisystem und Disk Quotas . . . . .	227
<b>12</b>	<b>Bootvorgang und Prozessverwaltung . . . . .</b>	<b>229</b>
12.1	Bootvorgang und Starten des Betriebssystems . . . . .	229
12.1.1	Der Bootvorgang . . . . .	229
12.1.2	Linux startet . . . . .	230
12.1.3	dmesg . . . . .	231
12.2	GRUB . . . . .	232
12.2.1	Bezeichnung von Geräten . . . . .	232
12.2.2	Die Konfigurationsdatei von GRUB . . . . .	233
12.2.3	Installation von GRUB . . . . .	233
12.2.4	Erstellung einer GRUB-Bootdiskette . . . . .	234
12.3	LILLO . . . . .	235
12.3.1	lilo . . . . .	236

12.3.2	/etc/lilo.conf . . . . .	238
12.3.3	Sicherheit: Zugriff aufs System ohne Passwort . . . . .	239
12.3.4	Fehlererkennung . . . . .	239
12.4	Der Init-Daemon . . . . .	240
12.4.1	/etc/inittab . . . . .	240
12.4.2	Die Runlevel . . . . .	242
12.4.3	runlevel . . . . .	244
12.4.4	init . . . . .	244
12.4.5	telinit . . . . .	245
12.4.6	Automatisches Starten von Diensten . . . . .	245
12.4.7	insserv . . . . .	246
12.5	Linux beenden . . . . .	247
12.5.1	shutdown . . . . .	247
12.5.2	halt und reboot . . . . .	248
12.6	Prozeßverwaltung . . . . .	248
12.7	Prozeß- und Systemlastüberwachung . . . . .	249
12.7.1	Load Average . . . . .	249
12.8	Tools zur Prozeß- und Systemlastüberwachung . . . . .	250
12.8.1	ps . . . . .	250
12.8.2	pstree . . . . .	252
12.8.3	top . . . . .	252
12.8.4	uptime . . . . .	254
12.8.5	free . . . . .	254
12.8.6	procinfo . . . . .	254
12.9	Prozeßadministration . . . . .	257
12.9.1	nice . . . . .	257
12.9.2	renice . . . . .	257
12.9.3	Vorder- und Hintergrundprozesse . . . . .	258
12.9.4	jobs . . . . .	258
12.9.5	fg . . . . .	259
12.9.6	bg . . . . .	259
12.9.7	kill . . . . .	260
12.9.8	killall . . . . .	260
12.9.9	nohup . . . . .	260
12.10	Arbeitsblatt XV: Bootvorgang und Prozesse . . . . .	263
<b>13</b>	<b>Administrative Aufgaben und Datensicherung</b>	<b>265</b>
13.1	Ausführung zu anderen Zeiten: Der at-Dämon . . . . .	265
13.1.1	at . . . . .	266
13.1.2	atq . . . . .	267
13.1.3	atrm . . . . .	267
13.1.4	Jobverwaltung . . . . .	267
13.1.5	batch . . . . .	267

13.2	Regelmäßiges Ausführen von Kommandos: Der cron-Dämon . . . . .	267
13.2.1	Der Dämon cron . . . . .	268
13.2.2	crontab . . . . .	268
13.2.3	Verwalten von cron-Jobs . . . . .	270
13.3	Logdateien . . . . .	270
13.3.1	Der Dämon syslogd . . . . .	270
13.3.2	/etc/syslog.conf . . . . .	270
13.3.3	Verwaltung der Logdateien . . . . .	272
13.3.4	Rotation von Logdateien: logrotate . . . . .	273
13.4	Datensicherung . . . . .	274
13.4.1	Vorüberlegungen . . . . .	274
13.4.2	Strategien . . . . .	274
13.4.3	Planung . . . . .	275
13.4.4	Backuptypen . . . . .	277
13.4.5	Lagerung der Backups . . . . .	278
13.5	Werkzeuge für die Sicherung . . . . .	279
13.5.1	tar . . . . .	279
13.5.2	Was ist ein Tarball? . . . . .	281
13.5.3	cpio . . . . .	282
13.6	Kompression . . . . .	283
13.6.1	gzip, gunzip und zcat . . . . .	283
13.6.2	Komprimieren mit bzip2 . . . . .	285
13.6.3	compress und uncompress . . . . .	287
13.7	Arbeitsblatt XVI: Administrative Aufgaben . . . . .	289
<b>14</b>	<b>Programminstallation und Kernelmanagement</b>	<b>291</b>
14.1	Kompilieren des Quellcodes . . . . .	291
14.1.1	Entpacken eines Tarballs . . . . .	291
14.1.2	Aufbau von C-Programmen . . . . .	292
14.1.3	configure . . . . .	292
14.1.4	make . . . . .	293
14.1.5	Installation . . . . .	294
14.2	Verwaltung von gemeinsam genutzten Bibliotheken . . . . .	294
14.2.1	Runtime Linker: ld.so . . . . .	295
14.2.2	ldd . . . . .	295
14.2.3	ldconfig . . . . .	296
14.3	Softwareverwaltung mit RPM-Paketen . . . . .	297
14.3.1	Installation und Upgrade . . . . .	299
14.3.2	Deinstallation . . . . .	305
14.3.3	Informationen . . . . .	306
14.3.4	Überprüfung . . . . .	314
14.3.5	Allgemeine Optionen . . . . .	316
14.4	Der Kernel . . . . .	316

14.4.1	uname	317
14.5	Verwaltung von Kernelmodulen	317
14.5.1	Die Moduldateien	317
14.5.2	lsmod	318
14.5.3	insmod	319
14.5.4	rmmod	319
14.5.5	modinfo	320
14.5.6	modprobe	321
14.5.7	Modulabhängigkeiten: <code>modules.dep</code>	321
14.5.8	depmod	322
14.5.9	Die Konfigurationsdatei: <code>modules.conf</code>	323
14.6	Kernelkompilierung	324
14.6.1	Kernelquellen	325
14.6.2	Konfiguration	325
14.6.3	Aufräumen, Kompilieren, Installieren	327
14.7	Arbeitsblatt XVII: Programminstallation und Kernelkompilierung	331
<b>15</b>	<b>Shell-Scripting</b>	<b>333</b>
15.1	Variablen, Aliase und Funktionen	333
15.1.1	Variablen	333
15.1.2	Aliase	334
15.1.3	Funktionen	335
15.2	Skripte	336
15.2.1	Ausführen von Skripten	337
15.2.2	Rückgabewerte	339
15.3	Grundstrukturen	340
15.3.1	Auswerten	340
15.3.2	test	342
15.3.3	if ... then ... elif ... then ... else ... fi	344
15.3.4	case	346
15.3.5	dialog	348
15.3.6	while ... do ... done	351
15.3.7	until ... do ... done	352
15.3.8	for ... in ... do ... done	352
15.3.9	read	352
15.4	Weitere Builtin-Befehle der Bash	353
15.4.1	shopt	353
15.4.2	source	354
<b>16</b>	<b>Linux und Hardware</b>	<b>355</b>
16.1	Das BIOS	355
16.1.1	Konfiguration	356
16.2	PC-Busarchitekturen	357

16.2.1	PCI	358
16.2.2	lspci	358
16.2.3	IRQs, I/O-Ports und DMA	360
16.2.4	Plug-and-Pray	361
16.3	Datenfernübertragung	362
16.3.1	Analog: Modems	362
16.3.2	setserial	364
16.3.3	isserial	366
16.3.4	Digital: ISDN	366
16.3.5	Breitband: DSL	367
16.4	Audio: Es gibt was auf die Ohren	367
16.5	SCSI	367
16.5.1	Linux und SCSI	368
16.5.2	scsiinfo	371
16.5.3	Die sieben Todsünden	372
16.6	USB	372
16.6.1	Linux und USB	373
16.6.2	Das usbfs	374
16.6.3	lsusb	376
16.6.4	usbmodules	377
16.7	Arbeitsblatt XVIII: Linux und Hardware	379
<b>17</b>	<b>TCP/IP: Geschichte, Adressen und Subnetting</b>	<b>381</b>
17.1	Einführung	381
17.1.1	Requests for Comments (RFC)	381
17.1.2	ARPA Kadabra: Die Geschichte des Internet	382
17.2	Zahlensysteme	384
17.2.1	Dezimalsystem	384
17.2.2	Binärsystem	384
17.2.3	Oktalsystem	385
17.2.4	Hexadezimalsystem	385
17.3	Host-Adressierung mit IP	386
17.3.1	Netzwerkklassen	386
17.3.2	Besondere Adressen	388
17.3.3	Private Netze	389
17.4	Subnetting	389
17.4.1	Technik	390
17.4.2	Subnetzmaske	391
17.4.3	Beispiele fürs Subnetting	392
17.4.4	Die möglichen Subnetze	393
17.4.5	Die belegten Subnetze	393
17.4.6	Weitere Beispiele	395
17.5	Arbeitsblatt XIX: Binär-, Oktal-, Dezimal- und Hexadezimalzahlen	397

17.6 Arbeitsblatt XX: IP-Adressierung und Subnetting . . . . .	399
<b>18 Linux im Netzwerk</b>	<b>403</b>
18.1 Konfiguration . . . . .	403
18.1.1 Konfiguration der Netzwerkschnittstellen: <code>ifconfig</code> . . . . .	403
18.1.2 Konfiguration der Kernel-Routing-Tabelle: <code>route</code> . . . . .	406
18.1.3 Konfiguration unter SuSE . . . . .	408
18.1.4 Einrichten eines Routers . . . . .	409
18.1.5 Netzwerkeinstellungen mit DHCP . . . . .	411
18.2 Clientseitige Namensauflösung . . . . .	411
18.2.1 Übersicht . . . . .	411
18.2.2 Namensauflösung mit <code>/etc/hosts</code> und <code>/etc/networks</code> . . . . .	413
18.2.3 Namensauflösung mit DNS . . . . .	414
18.2.4 <code>hostname</code> . . . . .	415
18.3 Diagnosewerkzeuge . . . . .	417
18.3.1 Bist du da? <code>ping</code> . . . . .	417
18.3.2 Wegverfolgung: <code>traceroute</code> . . . . .	419
18.3.3 <code>netstat</code> . . . . .	420
18.3.4 <code>telnet</code> . . . . .	421
18.3.5 <code>tcpdump</code> . . . . .	422
18.4 Arbeitsblatt XXI: Routing . . . . .	423
<b>19 X-Window</b>	<b>425</b>
19.1 Aufbau . . . . .	425
19.1.1 X-Protokoll . . . . .	426
19.1.2 Fenstermanager . . . . .	426
19.1.3 Displaymanager . . . . .	427
19.2 Konfiguration und Einrichtung . . . . .	428
19.2.1 Installation . . . . .	428
19.2.2 Die Konfigurationsdatei <code>XF86Config</code> . . . . .	428
19.2.3 Konfigurationsprogramme . . . . .	432
19.2.4 <code>xset</code> . . . . .	433
19.3 Schriften . . . . .	434
19.3.1 Lokale Schriften . . . . .	434
19.3.2 Der Fontserver <code>xfs</code> . . . . .	438
19.4 Starten der graphischen Oberfläche ... . . . .	440
19.4.1 ... direkt über den X-Server . . . . .	441
19.4.2 ... mittels des Duos <code>startx</code> und <code>xinit</code> . . . . .	441
19.4.3 ... über einen Displaymanager . . . . .	442
19.5 X-Window im Netzwerk . . . . .	443
19.5.1 Zugriffsschutz über <code>xhost</code> . . . . .	443
19.5.2 Zugriffsschutz über Magic Cookies und <code>xauth</code> . . . . .	445
19.5.3 Zugriff über SSH-X11-Weiterleitung . . . . .	446

19.5.4	Zugriff über XDMCP . . . . .	447
19.6	X-Clients . . . . .	448
19.6.1	Die Standard X-Toolkit Optionen . . . . .	448
19.6.2	Ressourcen . . . . .	450
19.7	Arbeitsblatt XXII: X-Window . . . . .	451
<b>20</b>	<b>Allerlei Wissenswertes</b>	<b>453</b>
20.1	Dateien und Verzeichnisse . . . . .	453
20.1.1	Login Begrüßungstext: <code>/etc/issue</code> . . . . .	453
20.1.2	Nachricht des Tages: <code>/etc/motd</code> . . . . .	453
20.1.3	Login-Terminals für <code>root</code> : <code>/etc/securetty</code> . . . . .	453
20.1.4	Prozesse sind auch nur Dateien: <code>/proc</code> . . . . .	453
20.2	YaST . . . . .	456
20.2.1	Benutzerverwaltung unter YaST . . . . .	456
20.2.2	Installation von weiteren Paketen mit YaST . . . . .	456
20.3	Grafik und Bilder . . . . .	457
20.3.1	<code>convert</code> . . . . .	457
20.4	Internet . . . . .	458
20.4.1	<code>wget</code> . . . . .	458
20.5	Die Tastatur . . . . .	459
20.5.1	<code>showkey</code> . . . . .	459
<b>A</b>	<b>Listen</b>	<b>463</b>
A.1	Die Shell-Befehle . . . . .	463
A.2	Verzeichnisse . . . . .	469
A.3	Wichtige Dateien . . . . .	470
A.4	Umgebungsvariablen . . . . .	471
A.5	Glossar . . . . .	471
<b>B</b>	<b>Der Unterricht</b>	<b>473</b>
B.1	Exportieren des X11-Desktops . . . . .	473
B.1.1	<code>Xvnc</code> . . . . .	473
B.1.2	<code>vncserver</code> . . . . .	474
B.1.3	<code>vncpasswd</code> . . . . .	474
B.1.4	<code>vncviewer</code> . . . . .	475
B.1.5	Beispiel: Einrichtung als Server für zwei Personen . . . . .	476
B.1.6	Beispiel: Einrichtung als Klassenraumserver . . . . .	476
<b>C</b>	<b>Zertifizierung: Das LPIC-Programm</b>	<b>479</b>
C.1	Junior Level Administration (LPIC1) . . . . .	479
C.1.1	Prüfung 101 . . . . .	479
C.1.2	Prüfung 102 . . . . .	484
<b>D</b>	<b>Open Publication License</b>	<b>489</b>



D.1 Englische Version . . . . .	489
D.2 Deutsche Version . . . . .	490



# Vorwort

*In a world without walls and fences,  
who needs windows and gates?*

*Anonymous*

Dieses Skript ist als Begleitmaterial zu meinem Unterricht "Einführung in Linux" und den Vorbereitungskursen zu den LPI-Prüfungen 101 und 102 entstanden. Schwerpunkt des Unterrichts ist eine Einführung in die Grundbedienung von Linux, wie sie dem Stoffplan der ersten LPI-Prüfung entspricht. Der Stoff der Prüfung 101 wird fast vollständig abgedeckt. Viele Themen der Prüfung 102 sind auch bereits vorhanden

Das Skript ist einmal als Begleitmaterial zum Unterricht entstanden, da es noch kein passendes Buch zur LPI-Prüfung in Deutsch gibt, zum anderen als meine Vorbereitung für die Prüfungen 101 und 102, die ich auch erfolgreich abgelegt habe.

Dieses Skript befindet sich wie immer in der Entwicklung und wird in Zukunft um einige Themen erweitert werden.

Im Vergleich zur Version 0.5 hat sich der Umfang der Fibel von ca. 350 Seiten auf ca. 500 Seiten erhöht. Viele Themen sind dazu gekommen. Einige sind noch recht frisch, so daß sicherlich noch ein paar Fehler drin versteckt sind und ein paar Brüche im Kontext. Trotzdem möchte ich diese Version nun veröffentlichen, da sich vieles getan hat. Für Anregungen und Fehlerkorrekturen bin ich sehr dankbar. Die Aufgaben sollten zum größten Teil mit den Grundvoraussetzungen einer SuSE-Linux-Distribution ab Version 8.0 gelöst werden können.

Besonders möchte ich bei meinen bisherigen Teilnehmern danken, die mich auf Fehler und Unstimmigkeiten aufmerksam gemacht haben und sich als Beta-Tester für die Aufgaben geopfert haben.

Bei Interesse an Linux-Seminaren und LPI-Vorbereitungskursen können Sie mich gerne kontaktieren. Nun möchte ich Sie nicht weiter aufhalten und hoffe Sie haben viel Spaß beim Lesen.

Ole Vanhoefer  
Kiel, 01.09.2004



# Kapitel 1

## Einführung

### 1.1 Was ist eigentlich Linux?

Linux ist ein 32-Bit **Unix-ähnliches Betriebssystem**. Im Gegensatz zu den meisten anderen Unix-Systemen ist Linux inklusive der Programmquellcodes frei kopierbar. Inzwischen existiert eine Vielzahl von Anwendungen für Linux, so daß fast alle Aufgaben, die bisher auf Unix-, Windows- oder Apple-Systemen erledigt wurden, auch unter Linux zu meistern sind.

Ein **Betriebssystem** ist eine Sammlung von Programmen, mit denen die grundlegendsten Funktionen eines Rechners realisiert werden. Dies reicht von der Schnittstelle Mensch-Maschine über die Verwaltung der Daten bis zur Kontrolle und Steuerung der Systemressourcen. Ohne ein Betriebssystem können Sie mit Ihrem Rechner nicht arbeiten, da so wichtige Dinge wie das Starten von Programmen und die Verwaltung Ihrer Dateien davon abhängen.

Diese Funktionen stellen eigentlich alle Betriebssystem zur Verfügung. Somit werden Sie bei jeder Arbeit mit einem Betriebssystem konfrontiert, wie z. B. CP/M, DOS, Windows 9x, Windows NT, Windows 2000, Windows ME, Windows XP, OS/2, Mac-OS, Unix usw.

Linux basiert auf dem Betriebssystem UNIX, das 1969 von den Bell Laboratories entwickelt wurde. Der finnische Student Linus Benedict Thorvald entwickelte den ersten Linux Betriebssystemkern (*Kernel*) auf seinem 386er Rechner. Dabei nutzte er nicht den bestehenden Quellcode, sondern programmierte das Betriebssystem vollständig neu. Nach außen präsentiert sich Linux als UNIX-System, während es innen aus einem völlig eigenständigem Code besteht.

Thorvald stellte im September 1991 der Gemeinde der MINIX-Anhänger sein neues Betriebssystem in der Version 0.01 vor. Er entschloß sich dabei das Programm und den Quellcode frei weiterzugeben und anderen Programmierern die Arbeit an seinem System zu gestatten. Viele Programmierer begeisterten sich für Linux und schon im Januar 1992 wurde der erste stabile Kernel 0.12 ins Internet gestellt.

Eigentlich kann man nur den Kernel mit seinen Modulen und direkt dazugehörenden Daten als Linux bezeichnen. Daneben gibt es aber eine Vielzahl von Hilfsprogrammen und Applikationen, die zusammen mit dem Kernel in den sogenannten Distributionen vertrieben werden. Für diese Zusammenstellung hat sich der Name Linux nun eingebürgert.

#### 1.1.1 Leistungsmerkmale des Kernels

Der aktuelle Kernel verfügt über folgenden Leistungsmerkmale:

- **POSIX-Konformität:** Der POSIX-Standard (*Portable Operating System Interface*) definiert eine Reihe von Bedingungen für portierbare Betriebssysteme.
- **Multitasking:** Mehrere Prozesse und damit Programme können (scheinbar) gleichzeitig ausgeführt werden.
- **Multiuser:** Mehrere Benutzer können gleichzeitig (z. B. über Telnet-Programme) auf dem Rechner

arbeiten.

- **Paging:** Das Paging erlaubt die Auslagerung von Speicherinformationen auf die Festplatte, so daß der Eindruck eines wesentlich größeren Arbeitsspeichers entsteht.
- **Shared Libraries:** Die Shared Libraries sind Programm-Bibliotheken. Bei Bedarf werden sie in den Speicher geladen und mehrere Programme gleichzeitig können auf die enthaltenen Routinen zugreifen. Dies spart Speicherplatz.
- **Shared Memory:** Normalerweise werden beim Multitasking-Betrieb jedem Programm eigene exklusive Speicherbereiche zugewiesen. Die Kommunikation erfolgt über die IPC (*InterProcess Communication*). Um diesen umständlichen Weg zu umgehen, können gemeinsam genutzte Speicherbereiche definiert werden, in denen die Programme ihre Daten austauschen können.
- **Symetric Multi Processing (SMP):** Beim Multitasking bekommen die Prozesse kleine Zeitscheiben zugewiesen, in denen sie ausgeführt werden. Da die Zeitscheiben schnell aufeinanderfolgen sieht es so aus, als ob die Programme gleichzeitig ausgeführt werden. Dies ist aber wirklich nur beim SMP der Fall. Linux ist in der Lage mehrere Prozesse auch auf mehreren Prozessoren gleichzeitig ausführen zu lassen.
- **Protected Mode:** Speicherschutzmechanismen des Prozessors verhindern, daß ein Prozeß auf den Speicherbereich eines anderen Prozesses zugreifen kann. Dadurch erscheint die von Windows noch bekannte, allseits beliebte allgemeine Schutzverletzung bei Linux nicht mehr.

## 1.2 Wie alles begann

Leider ist heute nicht mehr der Tag bekannt an dem Linus Benedict Torvalds<sup>1</sup> mit der Entwicklung des Linux Kernels begann. Ein Posting aus dem Usenet gibt aber einen Hinweis auf den möglichen Zeitpunkt, an dem er sich für das Thema zu interessieren begann.

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

Sicher ist, daß die erste Version des Linux Kernels am 25. August 1991 im Usenet angekündigt wurde. Schon kurze Zeit später fanden sich interessierte Programmierer, die an dem Projekt mitgearbeitet haben.

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

```
Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones. This has been brewing
since april, and is starting to get ready. I'd like any feedback on
```

---

<sup>1</sup>Da ihn alle Welt eigentlich nur Linus nennt, wird hier auch so verfahren.

things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).  
 I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)  
 Linus (torvalds@kruuna.helsinki.fi)  
 PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

**1991** Als erster genaue Termin steht der 3. Juli 1991 fest. Linux implementiert einige Gerätetreiber sowie den Festplattentreiber und einige User-Level Funktionen.

Linus veröffentlicht am 17. September 1991 die Version 0.01 des Kernels für einige Interessenten aus dem Usenet. Das Archiv mit der historischen Version ist auch heute noch verfügbar:  
<http://www.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz>.

Die erste "offizielle" Version des Linux Kernels (0.02) erscheint am 05. Oktober 1991. Mit dieser Version laufen bereits die bash, gcc, gnu-make, gnu-sed und compress.

Am 19. Dezember 1991 läuft die erste Version (0.11), die ohne die Hilfe eines anderen Betriebssystems lauffähig ist. Es gab keinen SCSI Support, so daß eine AT-Bus Festplatte Voraussetzung war. Es gab weder init noch login, nach den Systemstart landete man direkt in einer bash. Es gab Ansätze für die Implementierung von Virtual Memory, es waren aber mindestens 4 MB RAM notwendig um GNU Programme, insbesondere den gcc benutzen zu können. Ein einfacher Systemstart war aber auch schon mit 2 MB möglich.

Deshalb folgte für einige Personen eine Version mit virtueller Speicherverwaltung zu Weihnachten um den Kernel auch mit 2 MB RAM übersetzen lassen zu können.

**1992** Bereits am 5. Januar kommt die erste Version (0.12) heraus, die mehr Funktionen hatte als unbedingt benötigt werden. Mit dieser Version wurde der Kernel unter die GPL gestellt. Die ältere Lizenz unter der der Kernel stand war in vielen Punkten deutlich strenger.

Linus verteilte diese per Anonymous-FTP im Internet, was zu einem sprunghaften Anstieg der Testerzahl führte. Da dieser Anstieg so groß wurde, daß die nötige Kommunikation nicht mehr per eMail zu bewältigen war, wurde im Usenet die Gruppe alt.os.linux geschaffen. Das Interesse an Linux wuchs stetig und wurde von Linus koordiniert.

Um die Entwicklung voranzutreiben erhöhte Linus die Entwicklungsnummer im März auf Version 0.95.

Die im April folgende Version 0.96 war die erste Version mit der es möglich war das X-Window System zu betreiben.

Man schätzt die Zahl der Anwender auf circa 1000.

Oktober: Version 0.98.2

**1993** Die Zahl aller Programmierer steigt auf circa 100. Fünf von ihnen arbeiten mit Linus zusammen. Die Anwenderzahl beläuft sich auf etwa 20000.

Durch Anpassung des Linuxkernels an die GNU Umgebung der Free Software Foundation (FSF) wuchsen die Möglichkeiten von Linux erneut stark an, da man nun auf eine große Sammlung an vorhandener Software und Tools zurückgreifen konnte.

Dezember: Version 0.99.14

**1994** Mit der ersten stabilen Linux Version (1.00) im März wurde der Kernel netzwerkfähig und die User-Zahl steigt auf 100.000 an. Linus stellt nun den Quelltext des Linuxkernels offiziell unter die GPL.

Ein weiterer wichtiger Schritt war die Adaption eines Graphical User Interfaces (GUI), des X-Window-Systems. Dieses wurde von dem Xfree86 Projekt beigesteuert.

April: Version 1.10

**1995** Linux läuft nun auch auf DEC- und Sun Sparc-Prozessoren. Schätzungen belaufen sich auf rund eine halbe Millionen Anwender.

März: Version 1.20

Juni: Version 1.30

**1996** Mit der neuen Version 2.0 des Linux-Kernels können nun mehrere Prozessoren gleichzeitig angesteuert werden. Linux verliert langsam seinen Bastlerstatus und wird zu einer ernstzunehmenden Alternative für Firmen.

Die Anwenderzahl hat sich auf rund 1,5 Millionen Benutzern erhöht.

Juni: Version 2.00

September: Version 2.10

**1997** Neue Linuxversionen erscheinen fast wöchentlich. In verschiedenen Ländern existieren bereits Linux-magazine. Die Anwenderzahl ist auf 3,5 Millionen Anwender gestiegen.

Verschiedene namhafte Firmen beginnen ihre Software auf Linux zu portieren. Netscape ihren Webbrowser, Applixware ihre Office Anwendung und die Software AG Ihre Datenbank Adabas D. Damit gibt es auch professionelle Software für Linux.

April: Version 2.1.35

**1998** Das K Desktop Enviroment (kurz: KDE) wird gestartet.

Man schätzt die Anzahl der Programmierer von Linux auf 10.000, und die Anzahl der Anwender auf 7,5 Millionen.

März: Version 2.1.90

**1999** Man tippt auf ungefähr 10 Millionen Anwender. Linus Torvalds kündigt den Angriff auf Windows an.

Januar: Version 2.20

Mai: Version 2.30

**2000**

**2001** Januar: Version 2.4

**2002**

**2003**

**2004**

## 1.3 Grundlagen für die Installation

Vor der Installation wollen wir uns hier mit ein paar Grundlagen zur Hardware beschäftigen.

### 1.3.1 Hardwarevoraussetzungen

Für ein System ohne graphische Oberfläche können Sie sich an folgenden Werten orientieren.

Theoretisches Minimum	386 SX	1 MB RAM	5MB Festplatte
Brauchbares Minimum	386 DX/40 mit Coprozessor	8 MB RAM	150 MB Festplatte
Brauchbares System	486 DX/66	16 MB RAM	600MB Festplatte
Gutes System	Pentium90	32 MB RAM	1GB Festplatte

Graphische Systeme machen erst ab einem 486 Prozessor Spaß und für ein KDE-System sollten es minimal 64 MB Arbeitsspeicher sein.



### 1.3.2 Die Festplatte

Vor der Installation sollten wir uns den Aufbau einer Festplatte bzw. Harddisk einmal anschauen. Eine Festplatte besteht meistens aus einer, zwei oder mehreren Metallplatten, die beidseitig mit hochfein polierten Metalloxiden beschichtet sind. Ein Kamm von beweglichen Schreib-Lese-Köpfen greift seitlich in den rotierenden Stapel von Platten hinein und kann so die in konzentrischen Kreisen angelegten Datenmuster lesen oder schreiben. Durch die hohen Drehzahlen der Festplatte entstehen gezielt Luftbewegungen. Der Schreib-Lese-Kopf wird in Richtung der Oberfläche gesaugt und in einem vordefinierten Abstand gehalten. Bei einem Strömungsabriss durch Änderung der Drehzahl entfernt sich der Schreib-Lese-Kopf wieder von der Oberfläche.

Dieses System ist relativ stabil. Sollte es aber aus irgendwelchen Gründen zum direkten Kontakt des Kopfes mit den schnell drehenden Magnetplatten kommen, so ist das Ergebnis verheerend. Diesen Vorgang bezeichnet man auch als *Headcrash*. Die magnetisierte Oberfläche, welche die Daten trägt, wird unwiderruflich beschädigt. Das gleiche gilt für den Schreib-Lese-Kopf der Festplatte. Dieser Vorgang wird in Umgangssprache auch als *Spanabhebende Datenverarbeitung* bezeichnet. Das ist dann für die Daten der GAU, der wahrhaft größte anzunehmendste Unfall. Auch wenn es heute Möglichkeiten gibt, mit welcher man gelöschte Dateien wiederherstellen kann, sind mechanisch zerstörte Daten unwiderruflich verloren.

Die Festplatte beherbergt die magnetisierbaren Metallplatten, den Spindelmotor, den Schreib-Lese-Köpfen sowie den Positionierungsmechanismus der Köpfe in einem versiegelten Gehäuse, das vor äußeren Einflüssen schützt. Dieser Aufbau legt das Koordinatensystem der Festplatte fest, das die Lokalisierung der Daten auf der Platte ermöglicht. Der Kamm aus Schreib-Lese-Köpfen erzeugt konzentrische Kreise auf der Oberfläche der Magnetplatten. Einen solchen Kreis bezeichnet man als *Spur*. Die Spur 0 ist per Definition der äußerste Kreis einer Platte. Eine solche kreisförmige Spur wird dann in Kreissegmente unterteilt, die als *Sektoren* bezeichnet werden. Der Sektor ist die kleinste adressierbare Einheit einer Platte. Er ist 512 Bytes groß. Die übereinander liegenden Spuren eines Plattenstapels werden als *Zylinder* bezeichnet. Eine Festplatte mit drei Scheiben hat Zylinder, die aus sechs Spuren bestehen (Obere und untere Seite pro Scheibe). Eine Platte kann auf alle Spuren eines Zylinders zugreifen, ohne die Schreib-Lese-Köpfe neu zu positionieren.

Durch die Angabe von Zylinder, Kopf und Sektor kann man so genau einen Abschnitt einer Festplatte, den *Block*, adressieren. Diese Form der Adressierung wird als *CHS* (*Cylinder-Head-Sector*) bezeichnet. Der Block stellt die kleinste adressierbare Einheit einer Festplatte dar, d.h. zur Änderung eines einzelnen Bits muss immer ein ganzer Block gelesen und wieder geschrieben werden. CHS unterliegt mehreren Einschränkungen: Die Schnittstelle zwischen IDE und BIOS reserviert nur 16 Bits für die Zylinder (maximal sind 65.536 möglich), 4 Bits für die Köpfe (maximal 16) und 8 Bits für die Sektoren pro Spur (maximal 256). Das BIOS hat 10 Bits für die Zylinder zur Verfügung (1024), 8 Bits für die Köpfe (256) und 6 Bits für die Sektoren (63, da ab 1 gezählt wird). Bei diesen Grenzen ist jeweils der niedrigere Wert entscheidend, so daß alte BIOS-Versionen nur  $1024 \times 16 \times 63 \times 512$  Bytes = 504 MB adressieren können. Neuere BIOS-Versionen stocken per Mapping die Anzahl der Schreib-/Leseköpfe auf 255 auf und kommen damit auf 7.844 GB.

Ende 1995 löste *LBA* (*Logical Block Addressing*) die Adressierung nach CHS ab. Hier werden der Zylinder, Head und Sektor in logischen Blöcken zusammengefaßt. Bei LBA sind alle Sektoren der Festplatte - von null beginnend - durchnummeriert. Das auf 28 Bit basierende LBA verwaltet maximal 128 GB. Aktuell ist die auf 64 Bit basierende Variante, die bis zu 8.589.934.592 Terrabytes adressiert.

### 1.3.3 Partitionen

Neben der oben beschriebenen physikalischen Aufteilung einer Festplatte gibt es eine logische Aufteilung der Platte, die für die Installation sehr wichtig ist. *Partitionen* werden dazu benutzt um eine Festplatte in einzelne Teile zu zerlegen. Eine Festplatte besteht aus mindestens einer Partition und kann bis zu vier Partitionen enthalten. Um mit den Partitionen arbeiten zu können, müssen sie mit dem gewünschten Dateisystem formatiert werden. Bei den Partitionen werden zwei Typen unterschieden: Die primäre und die erweiterte Partition.

**Primäre Partition** Eine primäre Partition wird formatiert und ist dann in der Lage Daten zu speichern. Ein Rechner kann nur von einer primären Partition booten.

Name	Controller	Disk	Partition
<b>hda1</b>	IDE-Controller	Erste Festplatte	Erste Partition
<b>sda1</b>	SCSII-Controller	Erste Festplatte	Erste Partition
<b>hdb3</b>	IDE-Controller	Zweite Festplatte	Dritte Partition
<b>sd c4</b>	SCSII-Controller	Dritte Festplatte	Vierte Partition

Tabelle 1.1: Beispiele für Partitionsnamen

**Erweiterte Partition** Eine erweiterte Partition kann nicht direkt zum Speichern von Daten verwendet werden. Sie kann nur logische Laufwerke enthalten, die dann aber formatiert werden können. Eine Festplatte kann bis zu vier primäre Partitionen enthalten oder drei primäre und eine erweiterte Partition.

### Partitionsnamen

Die Partitionen werden direkt nach dem Typ des Festplattencontrollers benannt. Dabei steht für IDE-Platten der Buchstabe **h** und für SCSII-Platten der Buchstabe **s**. Dann folgt der Buchstabe **d** für *Disk*. Der dritte Buchstabe nummeriert die Festplatte und die vierte Ziffer gibt die Partitionsnummer an. Die erste Partition der ersten Festplatte mit einem IDE-Controller heißt also **hda1**. Mit einem SCSII-Controller würde sie **sda1** heißen.

### Planung der Partitionen

Für ein Linux-System benötigt man mindestens zwei Partitionen, eine für die Wurzel (*root*) (*/*) und eine für die Auslagerungspartition (*swap*). Es ist aber oft ratsam mehrere Partitionen einzurichten. Dabei sollte dem Bootverzeichnis */boot* und den Heimatverzeichnissen */home* eine eigenen Partition spendiert werden.

Die Vorteile einer Aufteilung auf mehrere Partitionen sind klar.

- Bei Beschädigung einer Partition bleiben die Daten auf den anderen Partitionen unbeschädigt.
- Wenn es notwendig ist eine Partition neu zu formatieren, bleiben die Daten auf den anderen Partitionen erhalten.
- Schnell wachsende Dateien können auf separate Partitionen ausgelagert werden um ihre Größe besser überwachen zu können.
- Das Updaten oder die Neuinstallation des Betriebssystems ist einfacher, da Programmdateien und Daten getrennt sind.
- Auch das Backup wird durch die Trennung von Programmdateien und Daten erleichtert.
- Auch die Zeiten für notwendige Überprüfungen des Dateisystems werden durch kleinere Partitionen verringert.

Der Nachteil allerdings liegt in der uneffektiven Nutzung des vorhandenen Festplattenplatzes, da ohne weiteres auf einer Partition noch Platz sein kann, während die Partition eines anderen Verzeichnisses schon voll ist. Planen Sie daher vorher mit Bleistift und Papier ihre Partitionen, denn eine nachträgliche Größenänderung ist immer mit Problemen und großem Zeitaufwand verbunden.

Im Gegensatz zu DOS/Windows merkt der Benutzer nichts von der Aufteilung, da es bei Linux keine Laufwerksbuchstaben gibt, sondern die Laufwerke bzw. Partitionen in den Dateibaum eingebunden werden.

**Kleine Plattenkapazität** Besitzt das System nur wenig Plattenplatz, so sollten möglichst wenig Partitionen angelegt werden, da jede dieser Partitionen einen ungenutzten Restplatz vorhalten muß.

**Beispiel**

Sie wollen Linux als Router auf einem alten Pentium-Rechner mit einer 1,2 GB großen Festplatte und 48 MB RAM installieren. Da der Festplattenplatz begrenzt ist, sollten sie die Platte wie folgt aufteilen:

**/boot** Eine 50 MB große Partition stellt sicher, daß alle Kernels unter der Zylinder-1024-Grenze liegen.

**swap** Eine 100 MB große Partition (96 MB) für den Swap-Speicher.

**/** Der Rest der Platte (1050 GB) wird für eine einzige große Wurzel-Partition verwendet, die alle anderen Verzeichnisse enthält.

Sollte sich die ganze Platte unter Zylinder 1024 befinden, dann kann für das Verzeichnis **/boot** auf eine eigene Partition verzichtet werden.

Auf älteren Systemen kann es auch vorkommen, daß mehrere kleinere Platten vorhanden sind. In diesem Fall kann der Verzeichnisbaum auf diese Platten aufgeteilt werden. So kann z. B. für das Verzeichnis **/home** eine eigene Platte verwendet werden. Die Verwendung der Platten ist genau so wie die Verwendung von mehreren Partitionen auf einer Platte.

**Große Plattenkapazität** Bei Servern, die mehrere Dutzend GB an Daten speichern müssen, ist natürlich genügend Platz für viele Partitionen. Datensicherung und benötigte Dateisysteme geben dabei die Verwendung der Partitionen vor.

**Beispiel**

Wenn wir von einem NFS-Server ausgehen, der mit 512 MB RAM und einem Plattenarray mit einer Gesamtkapazität von 100 GB ausgestattet ist, dann wäre diese Aufteilung empfehlenswert.

**/boot** Eine 50 MB große Partition stellt sicher, daß alle Kernels unter der Zylinder-1024-Grenze liegen.

**swap** 512 MB (bzw. 4 x 128 MB) reichen für die Swap-Partition aus, da der Server extra mit viel Speicher ausgestattet wurde um Plattenzugriffe zu verhindern. Eventuell kann die Größe sogar auf 256 MB reduziert werden.

**/** Da möglichst alle wichtigen Verzeichnisse eine eigene Partition bekommen sollen, reichen hier 100 MB aus.

**/usr** Hier sind 2 GB vorgesehen. Praktischerweise können die hier vorhandenen Programme mit den Workstations über NFS geteilt werden.

**/var** 500 MB reichen aus um die Logdateien zu speichern. Auf eine eigene Partition beschränkt, können sie das System bei zu großem und schnellem Wachstum nicht zumüllen.

**/tmp** Für temporäre Aufgaben sind 100 MB an Plattenplatz ausreichend. Auch hier verhindert eine separate Partition das Überwuchern des Systems.

**/home** Die restlichen 97 GB stehen dann für die Datenspeicherung übers Netzwerk zur Verfügung.

Verzeichnisse sind nicht auf lokale Partitionen beschränkt. NFS-Netzwerkverzeichnisse können die gleichen Aufgaben übernehmen. Früher, als Plattenplatz noch richtig teuer war, wurde oft das Verzeichnis **/usr** auf den Server ausgelagert. Heute ist dies nicht mehr so wichtig. Allerdings erleichtert ein gemeinsames **/usr**-Verzeichnis für alle Arbeitsrechner die Wartung (Update) des Systems.

Neben diesen Argumenten kann auch die Datensicherung über die Größe der Partitionen entscheiden. So beschränkt z. B. die Größe der Sicherungsmedien unter Umständen die maximale Größe einer Partition ein.

Mehr Informationen zu Partitionen und ihre manuelle Einrichtung finden Sie im Abschnitt 10.1.

### 1.3.4 Der Auslagerungsspeicher

Arbeitsspeicher ist auch heute noch teuer. Deshalb benötigen gerade bei Multitasking-Systemen die Programme oft mehr Speicher als vorhanden ist. Um dem gegenzuwirken arbeiten die Betriebssysteme nicht mit dem physikalischen Arbeitsspeicher sondern mit dem virtuellen Arbeitsspeicher. Der virtuelle Arbeitsspeicher setzt sich aus dem RAM und einem Speicherbereich auf der Festplatte zusammen. Bei Linux wird der Arbeitsspeicher auf der Festplatte durch die Swap-Partition realisiert.

Für die Größe der Swap-Partition gibt es eine Faustregel. Der Auslagerungsspeicher sollte immer doppelt so groß sein, wie der eingebaute Arbeitsspeicher. Dabei sollte eine Größe von 16 MB für die Swap-Partitionen nicht unterschritten werden um eine gute Funktion des Betriebssystems zu gewährleisten.

Seit Kernel 2.1.117 darf die Swap-Partition bis zu 2 GB groß werden. Die Anzahl der Swap-Partitionen kann bei der Kompilierung des Kernels festgelegt werden. Im Normalfall ist dieser Wert auf 32 eingestellt.

### 1.3.5 fips

Wenn Sie Linux auf einem Rechner installieren wollen, auf dem schon ein Betriebssystem läuft, stoßen Sie meistens auf das Problem, daß kein Platz mehr auf der Festplatte für eine separate Partition mehr ist. Für diesen Fall gibt es das Programm `fips`. Es ermöglicht die Aufteilung einer FAT16-Partition in zwei Teile. Dabei sollte es zu keinem Datenverlust kommen, trotzdem sollte die Daten mit einem Backup gesichert werden. Vor dem Einsatz von `fips`<sup>2</sup> sollten Sie außerdem genau die Dokumentation lesen.

Kopieren Sie die Dateien `fips.exe` und `restorbbs.exe` auf eine Diskette. Eine Kopie Ihres alten MBR (*Master Boot Record*) wird während der Installation auf die Diskette kopiert. Dieses Backup können Sie dazu nutzen die Veränderung wieder rückgängig zu machen. Daher Diskette beschriften und gut verwahren!!!

Gleichzeitig wird der alte MBR auf dem Linux-System als `/boot/boot.0300` für IDE-Platten und als `/boot/boot.0800` für SCSI-Platten abgelegt.

## 1.4 Installation

Damit wir mit Linux arbeiten können ist es notwendig das Betriebssystem zu installieren. Die Installation ist stark abhängig von der verwendeten Distribution. Bei der SuSE-Distribution kann eine Installation von CD, Festplatte oder übers Netzwerk mit NFS, Samba (Windows Freigabe) oder FTP erfolgen. Dabei wird das Installations- und Administrationstool YaST verwendet.

Da sich die Installationsprozeduren von Distribution zu Distribution und sogar von Version zu Version unterscheiden möchte ich an dieser Stelle nicht darauf eingehen. In den bei der SuSE-Distribution mitgelieferten Handbüchern wird die Installation beschrieben. Sollten Sie dieses Skript nicht als Begleitmaterial zum Unterricht benutzen, sondern selber lernen, dann suchen Sie sich für die Installation einen erfahrenen Linux-User. In den meisten Fällen kann SuSE-Linux mit wenigen Mausklicks installiert werden, aber je nach der verwendeten Hardware kann es zu kleineren oder größeren Problemen kommen. Dies gilt übrigens auch für Windows.

Für den Unterricht ist es effektiver die Distribution auf einem Server im Netz zur Verfügung zu stellen. Erstens müssen nicht Unmengen von CDs gebrannt werden und zweitens erleichtert diese Vorgehensweise auch die nachträgliche Installation von Software im weiteren Verlauf des Unterrichts.

Deshalb nun die Beschreibung der Schritte, die vor einer Netzwerkinstallation am Client durchgeführt werden müssen.

### 1.4.1 Erstellung der Bootdisketten unter DOS

Um mit der Installation beginnen zu können, müssen Sie den Rechner mit einem rudimentären Betriebssystem starten (Booten). Dies erfolgt in der Regel durch eine speziell dafür vorbereitete Diskette oder CD. Sie brauchen die Medien nur ins CD-Laufwerk einzulegen und den Rechner dann zu starten. In der Bootreihenfolge des BIOS muß allerdings das CD-Laufwerk bzw. Diskettenlaufwerk vor der Festplatte kommen.

---

<sup>2</sup>Download von <http://sunsite.unc.edu/pub/Linux/system/install>.

Die erste CD der SuSE Distribution ist bootfähig und in der Regel kann die Installation dann ohne Probleme durchgeführt werden. Manchmal unterstützen ältere Rechner oder SCSI-Systeme das Booten von CD nicht. Dort und um eine Netzwerkinstallation durchführen zu können, werden Bootdisketten benötigt, die ein Grundbetriebssystem starten. Diese Bootdisketten können von Diskettenimages erstellt werden. Für SuSE 8.1 gibt es eine Bootdiskette (`bootdisk`) und vier Moduldisketten mit Treibern (`modulesX`). Die beigefügten Textdateien enthalten die Liste der Treiber.

Unter DOS bzw. Windows können Sie das Programm `rawrite` benutzen. Sie finden das Programm auf der SuSE-Distribution auf der CD 1 im Verzeichnis `/dosutils/rawrite`. Die Bootimages befinden sich im Verzeichnis `/disks` auf der gleichen CD. Sie können also die Bootdisketten durch die Befehle

```
R:> dosutils\rawrite\rawrite disks\bootdisk
R:> dosutils\rawrite\rawrite disks\modules1
R:> dosutils\rawrite\rawrite disks\modules2
R:> dosutils\rawrite\rawrite disks\modules3
R:> dosutils\rawrite\rawrite disks\modules4
```

auffünf 3,5"-Disketten installieren, wenn die CD im Laufwerk R: liegt. Die Disketten müssen vorher formatiert werden<sup>3</sup>.

## 1.4.2 Installation mit Bootdisketten

1. Legen Sie die Bootdiskette ein und starten Sie den Rechner.
2. Beim jetzt erscheinenden Auswahlmenü wählen Sie Installation.
3. Nun wird der Kernel geladen. Da er sehr groß ist, benötigen Sie auch noch die Moduldiskette 1. Nachdem der Kernel vollständig geladen wurde, wird das System gestartet.
4. Nun werden Sie aufgefordert eine Moduldiskette einzulegen.
5. Sollte die Installationsroutine keine Installations-CD finden, weil der passende Treiber fehlt oder eine Netzwerkinstallation geplant ist, dann schalten Sie nun in die "Manuelle Installation".
6. Wählen Sie nun die Sprache für die Installation aus. (Deutsch)
7. Wählen Sie nun die Tastaturbelegung aus. (Deutsch)
8. Im nun folgenden Hauptmenü finden Sie die Menüpunkte

- **Einstellung**  
Hier können Sie noch mal die Sprachen für Installation und Tastatur ändern.
- **System-Information**  
Beim Booten analysiert der Kernel die Hardware des Systems. Über diesen Menüpunkt können diese und Informationen über den Kernel selber aufgerufen werden. Sie erhalten Informationen über die Kernel-Meldungen, Festplatten und CD-Laufwerke, geladene Module, PCI-Bus, Prozessor, Speicher, Ein- und Ausgabeports, Interrupts, Geräte, Netzwerkkarten und DMA-Kanäle.
- **Kernel-Module (Hardware-Treiber)**  
Fehlende Treiber für die Installation können hier eingebunden werden.
- **Installation / System starten**  
Jetzt kann die Installation gestartet werden oder ein bereits installiertes System gebootet werden.
- **Abbruch / Reboot**  
Installation abbrechen und den Rechner neu starten.
- **Power off**  
Installation abbrechen und den Rechner ausschalten.<sup>4</sup>

<sup>3</sup>Die Disketten müssen low-level-formatiert sein. Die Datenstruktur wird durch das Image wieder überschrieben.

<sup>4</sup>Geht natürlich nur bei ATX-Netzteilen.

9. Sollte ein passender Treiber fehlen (z. B. SCSI-Treiber), dann wechseln Sie in den Menüpunkt *Kernel-Module*.
  - (a) Dort wählen Sie die passende Rubrik aus (z. B. *IDE/RAID/SCSI-Treiber laden* für SCSI-Geräte).
  - (b) Legen Sie die angeforderte Moduldiskette ein. Die Daten werden in eine RAM-Disk geschrieben.
  - (c) Wählen Sie den passenden Treiber aus und geben Sie eventuell benötigte Parameter für das Modul ein.<sup>5</sup>
  - (d) Das System versucht das Modul nun zu laden. Auf Konsole 4 – zu erreichen über die Tastenkombination <ALT>+<F4> – können die Systemmeldungen angeschaut werden. Zurück geht es mit <ALT>+<F1>. Die Erfolgs- oder Mißerfolgsmeldung wird mit Return bestätigt.
  - (e) Über den Button *Zurück* landen Sie wieder im Hauptmenü.
10. Nachdem nun alle Treiber geladen sind, geht es mit dem Menüpunkt *Installation / System starten* weiter.
11. Im Installations-Menü wählen Sie *Installation/Update starten*.
12. Nun muß das Quellmedium ausgewählt werden.

**CD-ROM** Bei Auswahl dieses Menüpunkts versucht die Installationsroutine die Installations-CD bzw. DVD zu mounten. Klappt dies nicht, müssen Sie die Daten des Laufwerks per Hand eingeben.

**Netzwerk**

**Festplatte**

13. Nun erscheint das eigentliche Installationsmenü. Da sich der Installationsassistent häufig ändert, möchte ich hier nicht auf die weitere Installation eingehen. Eine gute Beschreibung der Installationsroutinen finden Sie in den mitgelieferten Handbüchern.

## 1.5 Konfiguration mit YaST2

Im Gegensatz zu Windows, wo die Einstellungen des Systems in der nicht gerade übersichtlichen Registry gespeichert werden, existieren unter Linux viele Textdateien, die die Konfigurationswerte enthalten. So ist es möglich nur mit einem Texteditor das komplette System zu administrieren. Obwohl diese Konfigurationsdateien normalerweise gut dokumentiert sind, gehört doch etwas Wissen und selbstvertrauen dazu, die Einstellung per Hand vorzunehmen.

SuSE hat für die Administration das Tool YaST2 für die graphische Oberfläche entwickelt. Es erlaubt eine wesentlich komfortablere aber nicht unbedingt schnellere Administration des Systems. Auch normale Benutzer können die YaST2 Kontrollzentrum oder die einzelnen Module aufrufen. Vor der Ausführung wird aber nach dem Passwort für *root* gefragt. Es ist also nicht nötig sich extra als *root* anzumelden um das System administrieren zu können.

Im YaST-Kontrollzentrum finden Sie die Module in verschiedenen Gruppen zusammengefaßt. Ich gehe im folgenden Abschnitt nur vereinzelt auf die einzelnen Module ein, weil die Funktionsweise vieler Module zum jetzigen Zeitpunkt zu viel weiteres Wissen erfordern würde und damit den Umfang dieses Kapitels sprengen würde.

**Software** Dieser Abschnitt beschäftigt sich mit der Installation, Deinstallation und Update der Software auf dem System.

**Installationsquelle wechseln** YaST merkt sich die letzte Installationsquelle. Erfolgt nun die Installation von einem anderen Medium, so kann dies hier eingestellt werden.

**Patch CD-Update** Für bestimmte Kunden stellt SuSE in regelmäßigen Abständen CDs mit aktuellen Patches zur Verfügung. Hier können Sie eingespielt werden.

**Update des Systems** Über dieses Modul können Sie das bestehende System auf einen neuen Distributionsstand bringen, z. B. von SuSE 8.0 auf SuSE 8.1.

---

<sup>5</sup> Meistens brauche Sie keine einzugeben.

**Online-Update** Fehlerbehebung ist ein wichtiger Faktor in der EDV. Mit dem Online-Update können Sie kleine Programme zur Fehlerkorrektur (*Patches*) Online herunterladen und Ihr System auf den neuesten Stand bringen.

**Software installieren/löschen** Dieses Modul ermöglicht es Ihnen Pakete von der SuSE-Distribution einzuspielen, upzudaten oder wieder vom System zu entfernen.

**Hardware** Wie der Namen schon errahnen läßt finden Sie hier die Möglichkeit die Hardware zu konfigurieren oder Informationen über Sie einzuholen. Spezielle für Drucker, Grafikkarte und Monitor, Festplatte, Scanner, Maus, Soundkarte und TV-Karte gibt es hier Module.

**Automatische Druckererkennung** für die Installation lokaler Drucker.

**Drucker bearbeiten** Mit diesem Modul können existierende Drucker bearbeitet werden und neue lokale und Netzwerkdrucker eingerichtet werden.

**Grafikkarte und Monitor** Hiermit können Sie die Einstellung für die Grafikkarte ändern und die Ausgabe an den Monitor anpassen.

**Hardware-Info** Liefert Ihnen eine Übersicht über die eingebaute Hardware.

**IDE DMA-Modus** Hiermit können Sie den DMA-Modus (*Direct Memory Access*) Ihrer Laufwerke aktivieren. Dies ist z. B. für das Betrachten von DVDs wichtig.

**Mausmodell wählen** Was soll ich da noch viel schreiben ...

**Partitionieren** Hier gelangen Sie zu dem Tool, bei dem Sie schon bei der Installation die Festplatten Ihres Rechners in einzelne Abschnitte aufteilen können.

**Scanner** Auch Scanner wollen konfiguriert werden ...

**Sound** Bei der Soundkonfiguration sollten Sie die passende Lautstärke beachten und richtig einstellen. Aus Sicherheitsgründen ist die Grundeinstellung meistens sehr leise. Dies und ein sehr leise gestellter Lautsprecher haben schon oft zur falschen Annahme geführt, daß die Soundkarte unter Linux nicht funktioniert.

**TV-Karten** Seit SuSE 8.0 hat sich die Unterstützung von TV-Karten sehr verbessert. Hier werden Sie automatisch erkannt und konfiguriert.

**Netzwerk/Basis** Die Konfiguration von Netzwerkkomponenten (Netzwerkkarten, Modems, ISDN-Karten, etc.) oder Diensten (E-Mail, DSL, Serverdienste) wird durch die Module dieser Gruppe erledigt.

**E-Mail**

**Konfiguration der Netzwerkkarte**

**Konfiguration des Modems**

**Konfiguration von DSL**

**Konfiguration von ISDN**

**Start oder Stopp von Systemdiensten**

**Netzwerk/Erweitert** Die speziellen Netzwerkfunktionen für DNS, NFS, NIS und Routing werden hier konfiguriert.

**HostName und DNS**

**LDAP-Client**

**NFS-Client**

**NFS-Server**

**NIS+-Client**

**NIS-Client**

**NIS-Server**

**Routing**

**Sicherheit & Benutzer** Benutzerverwaltung und die Richtlinien zur Sicherheit (Firewall) werden hier abgehandelt.

**Benutzer bearbeiten und anlegen**  
**Einstellungen zur Sicherheit**  
**Firewall**  
**Gruppen bearbeiten und anlegen**  
**Neue Gruppe anlegen**  
**Neuen Benutzer anlegen**

**System** Diese Module kümmern sich um die systemnahen Einstellungen, wie Backup, Bootdisketten, Starteinstellungen, Automatisches Starten von Diensten und Grundkonfigurationen.

**Backup des Systems erstellen**  
**Boot-, Rettungs- oder Moduldiskette erzeugen**  
**Konfiguration des Bootloaders**  
**LVM**  
**Runlevel-Editor**  
**Sprache wählen**  
**Sysconfig-Editor**  
**Tastaturlayout auswählen**  
**Zeitzone auswählen**

**Sonstiges** Hier finden Sie den Rest der YaST-Module, die SuSE nicht in eine der anderen Gruppen einordnen konnte.

**Drucker für CUPS** Konfiguriert Drucker für das Drucksystem CUPS  
**Drucker für LPD** Konfiguriert Drucker für das Drucksystem LPD  
**Startprotokoll anzeigen** Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/boot.msg` gestartet wird.  
**Support-Anfrage stellen** Hier können Sie eine Anfragen an den Online-Support stellen.  
**Systemprotokoll anzeigen** Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/messages` gestartet wird.  
**Treiber-Cd des Herstellers laden** Ermöglicht die Installation von Treibern, die vom Hersteller der Hardware kommen.

Der folgende Abschnitt zeigt Ihnen die Administration einiger Funktionen. Diese ist im Prinzip über YaST sehr einfach. In den meisten Fällen muß man nur wissen was man will<sup>6</sup> und man muß lesen können.<sup>7</sup>

### 1.5.1 Wie installiere ich weitere Softwarepakete?

Das es um Software geht, wählen Sie im YaST-Kontrollzentrum die Gruppe *Software*. Ein Blick auf die Liste der Module zeigt uns, daß **Software installieren/löschen** wohl eine gute Wahl ist. Falls Sie die Installationsquelle gewechselt haben, müssen Sie diese vorher mit dem Modul *Installationsquelle wechseln* neu einstellen.

Nach dem Starten des Moduls sucht YaST nach der Paketliste. Dies kann einen Moment dauern. Findet YaST kein Installationsmedium, so behilft er sich mit der letzten gespeicherten Liste.

**Achtung!** Installation über NFS: Sollte sich YaST bei dieser Phase aufhängen und nicht weitermachen, dann liegt es in den meisten Fällen daran, daß der NFS-Server nicht erreichbar ist. Da es sich um eine harte Verbindung handelt, dauert es sehr lange bis es zu einem Timeout kommt und YaST den Fehler bemerkt. Um diesen Fehler zu umgehen, wählen Sie vorher ein anderes Installationsmedium aus.

---

<sup>6</sup>Das ist nicht immer so einfach!

<sup>7</sup>Was meistens noch schwerer ist ...



Nach dem Laden der Paketlisten erscheint das Auswahlfenster für die Pakete. Sie sehen dort drei Tabellen. Links oben zeigt Ihnen die Themengruppen oder Paketserien an, in die die Pakete einsortiert worden sind. Unten links sehen Sie, wieviel Platz noch auf der Festplatte vorhanden ist. Die rechte Tabelle zeigt die Pakete in der Gruppe an. Durch Doppelklicken oder Betätigen des Buttons *Aus-/Abwählen* können Sie den Zustand des Pakets ändern. Dieser Zustand wird durch ein Zeichen vor dem Paketnamen angezeigt. Dabei steht ein **X** für ein Paket, das zur Installation vorgesehen ist. Das kleine **i** steht für ein bereits installiertes Paket und das kleine **d** für ein installiertes Paket, daß deinstalliert werden soll. Auch ein kleines **a** kann als Markierung vorkommen. Diese Pakete wurden automatisch ausgewählt zur Installation, da diese von anderen Paketen benötigt werden. Keine Markierung bedeutet natürlich, daß dieses Paket nicht installiert und auch nichts für dieses Paket geplant ist.

Zu den meisten Paketen existiert neben der Kurzbeschreibung in der Tabelle noch eine etwas ausführlichere Beschreibung des Pakets. Diese erreichen Sie über den Button *Beschreibung*, was Sie natürlich nie erraten hätten. ;-)

In vielen Fällen kennt man den Namen des Programms, aber nicht in welchem Paket oder in welcher Themen-gruppe/Paketserie das Paket liegt. In den meisten Fällen heißt das Programmpaket genau wie das Programm. Über eine Suchfunktion (Button *Suchen*) können die Paketnamen und sogar die Beschreibungen nach einem Stichwort durchsucht werden.

Unter *Extras* finden Sie weitere Funktionen um Pakete auszuwählen oder abzuwählen. Besonders interessant ist die Möglichkeit, die aktuelle Paketauswahl abzuspeichern oder eine gespeicherte Paketauswahl einzuladen. Dadurch ist es relativ schnell möglich Rechner mit der gleichen Softwareausstattung zu installieren.

Nach der Paketauswahl, die Sie mit dem Button *OK* abschließen, werden die Pakete installiert. Bei der CD-Version werden Sie dann und wann aufgefordert die entsprechenden CDs einzulegen. Mit einem Fortschrittsbalken werden Sie darüber informiert, wie weit Ihre Installation ist. Nach Abschluß der Installation startet YaST das Programm *SuSEConfig*, daß die Konfigurationsdateien des Systems für die neuen Programmen konfiguriert.

Und schon sind die Programme installiert.

### 1.5.2 Wie mache ich ein Online-Update?

*Keine Software ist fehlerfrei!* Diesem gerne von einer Firma aus Redmond als Entschuldigung herangezogenen Satz kann ich nur zustimmen. Wer selber mal ein längeres Programm geschrieben hat, weiß, wieviel Fehler sich einschleichen können. Und wir reden hier von Programmen, deren Sourcecode ausgedruckt mehrere Aktenordner füllen würden. Programme *sind* fehlerhaft. Finden Sie sich damit ab.

Zum Glück gibt es so Leute, die sich nicht damit abfinden und die Fehler in Programmen bereinigen. Bei OpenSource-Software kann dies theoretisch jeder. Wohlgemerkt: theoretisch. Um sich in einem großen Programm zurechtzufinden braucht es schon etwas längere Einarbeitungszeit und gute Kenntnisse in den Programmiersprachen und natürlich der Programmierung selber.

Jedenfalls werden viele Fehler bzw. Sicherheitslücken im Laufe der Zeit in Programmen entdeckt und oft auch behoben. Um nicht das ganze Programm neu einspielen zu müssen, benutzt man sogenannte Patches (Flicken), die den Binärcode des "alten" Programms auf den neuesten Stand bringen. Die Fehlerkorrekturen – manche Leute reden auch von Fehlerveränderungen – können Sie manuell einspielen. SuSE bietet Ihnen den Service eines Online-Updates. Das YOU (*YaST Online Update*) stellt Ihnen die neuesten Patches zu den SuSE-Paketen zur Verfügung. Es stellt fest, was an neuen Patches vorhanden ist. Dann lädt es die Patches für die installierten Programme runter und installiert Sie.

Das YOU finden Sie auch im Kontrollzentrum unter *Software*. Nach dem Starten des Moduls sucht das System nach verfügbaren Update-Servern. Sie müssen natürlich auch eine Verbindung zum Internet besitzen.

Sie können nun zwischen manuellem und automatischem Update wählen. Entscheiden Sie sich für das manuelle Update, denn dort können Sie besser sehen, was installiert wird. Sie können auch den Server wählen, von dem Sie die Patches beziehen möchten. Dies sollten Sie tun, falls der erste Server überlastet ist.

Nun lädt YOU die Liste der zur Verfügung stehenden neuen Updates herunter. Nachdem die Verbindung beendet wurde, geht es mit *Weiter* weiter.

Sie sehen nun die Liste der zur Verfügung stehen Patches mit Erläuterungen. Die Patches, die Sie benötigen sind schon mit einem X gekennzeichnet zum Download vorgesehen. Sie können nun einfach mit *Weiter* zum

nächsten Schritt, dem Herunterladen der Pakete, gehen.

Nachdem dieses auch abgeschlossen ist und Sie mit *Weiter* zum nächsten Bildschirm gewechselt sind, kann es nun endlich mit dem Einspielen der Patches losgehen. Danach wird noch einmal SuSEconfig gestartet um die Konfigurationsdateien anzupassen und schon ist Ihr System auf dem neuesten Stand.

## 1.6 Benutzer- und Gruppenverwaltung

Linux ist ein Multiuser-Betriebssystem. Dadurch ist es mehreren Personen nacheinander oder sogar gleichzeitig möglich auf einem Rechner arbeiten. Damit es zu keinem Durcheinander der Daten kommt, muss sich jeder Benutzer eindeutig identifizieren, wenn er mit Linux arbeiten möchte. Außerdem gehört jeder Benutzer mindestens einer Gruppe an. Jeder Benutzer bekommt einen eindeutigen Namen und damit er sich auch authentifizieren kann ein Passwort.

Für die Benutzerverwaltung finden wir in der Gruppe *Sicherheit & Benutzer* die Module *Benutzer bearbeiten und anlegen*, *Neuen Benutzer anlegen*, *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen*.

Diese Module sind Frontends zu der eigentlichen Benutzerverwaltung von Linux. Ein Frontend ist ein Programm, das eigentlich nichts selber macht, sondern sich als Bedienungselement zwischen den Benutzer und dem eigentlichem Programm setzt. Das Ziel eines Frontends ist es, die Bedienung von Programmen und Konfigurationsdateien zu erleichtern und damit komfortabler und fehlerfreier zu machen.

Die Konfigurationsdateien für die Benutzerverwaltung befinden sich wie die meisten Konfigurationsdateien im Verzeichnis `/etc`. Die Liste der Benutzer ist in der Datei `/etc/passwd` abgelegt. Die Passwörter befinden sich in der Datei `/etc/shadow`. Deren Aufbau und Arbeitsweise wird in einem späteren Kapitel behandelt.

### 1.6.1 Wie erstelle, bearbeite und lösche ich Benutzer?

Die Module *Benutzer bearbeiten und anlegen* und *Neuen Benutzer anlegen* sind für die Benutzerverwaltung verantwortlich.

#### Neuen Benutzer anlegen

Um einen neuen Benutzer anzulegen, starte Sie das Modul *Neuen Benutzer anlegen*. Im nun erscheinenden Fenster können Sie den Benutzernamen und das Passwort des neuen Benutzers eintragen.

Bei der Eingabe eines Passworts müssen Sie zwischen Groß- und Kleinschreibung unterscheiden. Ein Passwort sollte mindestens fünf Zeichen lang sein und darf keine Sonderzeichen (z. B. Akzente) enthalten. Sie können für die Passwörter Buchstaben, Zahlen, Leerzeichen und die Zeichen `*, , ; : . _ - + ! $ % & / | ? { [ ( ) ] }` verwenden. Es ist ausreichend, ein Passwort mit der Länge von maximal acht alphanumerischen Zeichen zu wählen. Alle darüber hinausgehenden Zeichen werden bei der normalerweise eingestellten Crypt-Verschlüsselungen ignoriert. Das Passwort müssen Sie zweimal eintragen, da es aus Sicherheitsgründen nicht angezeigt wird. So können die meisten Tippfehler verhindert werden.

Für die Benutzernamen und Passwörter gibt es Regeln. So darf der Benutzername nur aus Kleinbuchstaben, Ziffern, `-` und `_` bestehen und muß mit einem Buchstaben oder `_` beginnen. Dies ist aber nur eine Vorgabe von YaST. Linux selber erlaubt auch Großbuchstaben und andere Zeichen für den Benutzernamen.

Die Felder Vorname und Nachname sind eigentlich Spaß. Die entsprechenden Daten werden nur in das Kommentarfeld des Benutzers in der Datei `/etc/passwd` eingetragen.

Hinter dem Button *Details* verbergen sich weitere Einstellungsmöglichkeiten wie Home-Verzeichnis oder die Benutzerkennung (*UID*).

Jedem Benutzer bekommt eine eindeutige Benutzerkennung (*UID*) zugewiesen. Für normale Benutzer sollte eine UID größer als 499 verwendet werden, da die kürzeren UIDs vom System für spezielle Zwecke und Pseudo-Benutzernamen verwendet werden.

Benutzer können durch eine Mitgliedschaft in einer Gruppe weitere Rechte bekommen. Zu jedem Benutzer gehört eine Standardgruppe in der er Mitglied ist. In dem Feld *Standardgruppe* können Sie eine Gruppe aus der Liste aller in Ihrem System vorhandenen Gruppen wählen.

Zusätzliche Gruppenzugehörigkeiten können sie über das entsprechende Feld dem Benutzer zuweise. Hier wird angezeigt, zu welchen Gruppen der Benutzer noch gehört.

Der Benutzer kann auch ein Heimatverzeichnis (*Home Directory*) besitzen um seine Daten zu dort zu speichern. Normalerweise ist dies `/home/BENUTZERNAME`.

Für den Betrieb auf der Konsole (Textbetrieb) wird eine Login-Shell (der sog. Kommandozeileninterpreter) benötigt. Wählen Sie im entsprechenden Feld eine Shell aus der Liste der auf Ihrem System installierten Shells aus.

### Benutzer bearbeiten und anlegen

Eine Übersicht der angelegten Benutzer liefert das Modul *Benutzer bearbeiten und anlegen*. Beim Start werden nur die normalen Benutzer angezeigt. Sie können sich aber auch die für interne Zwecke (z. B. Serverdienste) angelegten Systembenutzer anzeigen lassen. Durch anklicken der Spaltenüberschriften können Sie die Sortierung der Spalten ändern.

In diesem Fenster stehen Ihnen drei Funktionen zur Verfügung:

- Einen neuen Benutzer anlegen (*Hinzufügen*),
- den markierten Benutzer bearbeiten (*Bearbeiten*) und
- den markierten Benutzer löschen (*Löschen*).

Wenn Sie den Button *Hinzufügen* betätigen landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*.

Auch nachdem Sie den Button *Bearbeiten* landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*. Nur daß die Felder nicht leer sondern mit den Daten des zu bearbeitenden Benutzers gefüllt sind.

Wenn Sie den markierten Benutzer löschen wollen, dann betätigen Sie einfach den Button *Löschen*. Sie werden dann noch gefragt, ob das Heimatverzeichnis des Benutzers auch mit gelöscht werden soll. Wenn Sie sich dann fürs Löschen entscheiden ist es vorbei mit dem Benutzer.

Vorbei? Nicht ganz. Die Änderungen werden erst dann ins System übertragen, wenn Sie mit dem Button *Beenden* das Modul verlassen. Das Verlassen über den Button *Abbrechen* läßt das System unberührt.

## 1.6.2 Wie erstelle, bearbeite und lösche ich Gruppen?

Um bestimmten Mengen von Benutzern mehr Rechte zu geben, wurden die Gruppen entwickelt. Benutzer können Mitglied in einer Gruppe sein und haben dann die Rechte, die der Gruppe zugewiesen wurde, zusätzlich zu Ihren eigenen Rechten.

Die Module *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen* sind für die Verwaltung der Gruppen zuständig.

### Neue Gruppen anlegen

Nach dem Start des Moduls *Neue Gruppen anlegen* erscheint ein Fenster mit Einträgen für den Gruppennamen, die Gruppenkennung, dem Passwort und den Mitgliedern für die Gruppe.

Der Name der Gruppe sollte nicht zu lang sein. Normalerweise werden 1-10 Zeichen verwendet. Ansonsten gelten die gleichen Regeln wie für die Benutzernamen.

Wie auch die Benutzer haben die Gruppe eine interne Kennung. Diese GID liegt irgendwo zwischen 0 und 60000. Einige der IDs werden bereits während der Installation vergeben. YaST2 gibt eine Warnmeldung aus, wenn Sie aus Versehen eine bereits vergebene verwenden.

Auch Gruppen haben ein Passwort. Falls die zugehörigen Benutzer der Gruppe sich beim Wechseln in diese neue Gruppe identifizieren sollen, können Sie dieser Gruppe ein Passwort zuteilen. Aus Sicherheitsgründen wird dieses Passwort hier nicht angezeigt. Dieses Eingabefeld muss nicht ausgefüllt werden. Um Tipfehler zu vermeiden, müssen Sie das Passwort zweimal eingeben.

Das Fenster *Mitglieder dieser Gruppe* zeigt eine Liste aller angelegten Benutzer an. Hier können Sie einstellen, wer Mitglieder dieser Gruppe sein soll.

### Gruppen bearbeiten und anlegen

Das Modul *Gruppen bearbeiten und anlegen* zeigt eine große Ähnlichkeit mit dem Modul *Benutzer bearbeiten und anlegen*. Kein Wunder, es ist dasselbe Programm. Ein Klick auf *Benutzerverwaltung* und schon sind Sie in dem bereits bekannten Fenster. Dementsprechend gibt es hier auch drei Funktionen.

- Eine neue Gruppe anlegen (*Hinzufügen*),
- die markierte Gruppe bearbeiten (*Bearbeiten*) und
- die markierte Gruppe löschen (*Löschen*).

Beim Anlegen und Bearbeiten einer Gruppe landen Sie im gleichen Fenster wie im Modul *Neue Gruppen anlegen*.

Das Löschen von Gruppen gestaltet sich etwas schwieriger, da erst alle Benutzer aus der Gruppe entfernt werden müssen, bevor sich die Gruppe löschen läßt.

Auch hier gilt, alle Änderungen werden erst nach dem Verlassen des Moduls über den Button *Beenden* in das System übertragen.

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....



Die Ausführung der folgenden Aufgaben ist wichtig für alle folgenden Aufgaben. Führen Sie daher bitte alles aus. Eventuell haben Sie die Aufgaben schon begleitend zum Unterricht bzw. zur Lektüre gemacht.

**1** Ermitteln Sie die Hardwarekonfiguration des Rechners (Graphikkarte, Festplatte, Netzwerkkarte, Monitor).

**2** Ermitteln Sie die IP-Nummer, die Subnetzmaske, den Nameserver, den Namen, und das Standardgateway des Rechners an dem Sie arbeiten.

**3** Beginnen Sie mit der Installation durch YaST2. Beschreiben Sie ausführlich alle Schritte der Installation und erstellen Sie damit ein "Kochrezept" für die zukünftigen Installationen. Dabei soll folgendes berücksichtigt werden:

1. Legen Sie folgende Partitionen an:

/ mit 5 GB als primäre Partition (ext3)

/home mit 3 GB als logisches Laufwerk (ext3)

swap mit 256 MB als logisches Laufwerk

2. Für die Paketauswahl wählen Sie KDE mit Office.

3. Richten Sie das Netzwerk mit den übergebenen Werten ein.

**4** Sie besitzen einen Rechner, bei dem das CD-Laufwerk über einen Adaptec SCSI-Controller 2940 läuft. Auf welchem Moduldiskettenimage befindet sich der dafür benötigte Treiber?

**5** Welche Installationsmöglichkeiten (Quelle) können verwendet werden?

**6** Legen Sie mit YaST2 den Benutzer *walter* mit der UID 510<sup>8</sup> und dem Paßwort *hallo* an.

**7** Installieren Sie mit YaST2 aus der Serie *ap* das Paket *pdksh*.

**8** Installieren Sie mit YaST2 die Pakete *stat*, *rman*, *locate* und *koffice* nach. In welchen Serien liegen sie und wofür sind sie da?

**9** Speichern Sie die aktuelle Paketkonfiguration im Heimatverzeichnis von *root* (/root) unter dem Namen *config\_kap1.usr* ab.

**10** Löschen Sie die Pakete *stat*, *rman*, *locate* und *koffice*.

**11** Laden Sie nun die alte Paketkonfiguration aus Aufgabe 9 ein. Überprüfen Sie, ob die gerade gelöschten Pakete nun wieder zur Installation vorgesehen sind.

**12** Installieren Sie wieder die Pakete *stat*, *rman*, *locate* und *koffice*.

**13** Auf dem zur Installation vorgesehenen Rechner mit zwei IDE-Festplatten (Master und Slave auf dem ersten Controller) soll Linux installiert werden. Auf der Master-Platte befindet sich eine Windows-Installation mit einer primären Partition, einer erweiterten Partition und einer logischen Partition. Auf der erweiterten Partition ist noch Platz für eine logische Partition. Auf der zweiten Festplatte befindet sich auch eine primäre Partition für Windows, die bestehen bleiben soll. Diese Partition ist die erste auf der Slave-Platte.

Für die Linux-Installation sind zusätzlich zu den Standardpartitionen noch Partitionen für das Verzeichnis /home und /var geplant. Skizzieren Sie den Aufbau des Partitionsschema unter Berücksichtigung einer sinnvollen Aufteilung und unter Verwendung der Linux-Partitionsbezeichnungen.

---

<sup>8</sup>Die UID ist die interne Benutzernummer für das System.

.....  
*Notizen:*

.....



# Kapitel 2

## X-Window und KDE

### 2.1 X-Window-System

Bei der Installation von SuSE-Linux in der oben genannten Konfiguration wird standardmäßig das X-Window-System mit dem KDE installiert. Das X-Window-System kann am besten durch die Tools `sax` bzw. `sax2` konfiguriert werden. Einstellungen über Graphikkarte, Monitor, Auflösung, Tastatur und Maus erfolgen hier.

Für die Bedienung und Arbeit am X-Window-System wird ein Window-Manager benötigt. Er kümmert sich in erster Linie darum, wie die Programmfenster am Bildschirm dargestellt und bedient werden. Er stattet die Fenster mit einer Titelleiste und Schaltflächen (Buttons, Menüs etc) aus und hilft bei deren Verwaltung (Verschieben, Größe ändern, Verkleinern zu Icon, Programmwechsel). Daneben stellen die meisten Windowmanager Menüs für die gängigen Programme bereit.

Diese elementaren Funktionen werden nicht vom X-Window-System zur Verfügung gestellt. Der Vorteil dieser Methode ist die große Flexibilität bei der Gestaltung der Oberfläche und damit der Benutzerschnittstelle. Das hat aber auch den Nachteil, daß jeder Anwender einen anderen Windowmanager benutzt und damit keine einheitliche Oberfläche existiert. Ein Problem, daß vor allem die Anfänger betrifft.

Die aktuelle SuSE-Distribution startet automatisch das X-Window-System für die graphische Anmeldung. Wird das System mit der Textkonsole gestartet, kann von der Textkonsole aus das X-Window-System mit dem Befehl `startx` gestartet werden.

Sollte das X-Window-System sich aufhängen, kann es mit der Tastenkombination `STRG + ALT + BACKSPACE` beendet werden.

### 2.2 KDE

Das KDE (KDE Desktop Enviroment) ist eine Sammlung von X-Programmen mit einem Window-Manager für das X-Window-System. Es stellt wie auch die anderen Windowmanager (z. B. Gnome oder fvwm) die Schnittstelle zwischen Benutzer und dem X-Window-System dar. Das KDE umfaßt dabei nicht nur die Konfigurationstools sondern auch eine Vielzahl von Standardprogrammen.

Noch einige Hinweise zu KDE.

- Doppelklicks sind in KDE verpönt und sind nur in Ausnahmefällen erforderlich.
- Mit der rechten Maustaste können Sie in den meisten KDE-Programmen ein Kontextmenü aufrufen.
- Fast alle KDE-Programme kommen mit Drag & Drop zurecht.
- KDE-Programme können in der Regel direkt über einen Menüpunkt konfiguriert werden. Bei einigen Programmen erfolgt die Konfiguration über das KDE-Kontrollzentrum.

### 2.2.1 KDE-Kontrollzentrum: KControl

Das KDE-Kontrollzentrum (KControl) ist das zentrale Konfigurationsprogramm fürs KDE. Grundlegende Eigenschaften wie Sprache, Desktopdesign, Sound u. a. können hier eingestellt werden.

### 2.2.2 KDE-Systemüberwachung: KSysGuard

Die KDE-Systemüberwachung zeigt Ihnen Informationen über das System und die darauf laufenden Prozesse an. Sie können im KDE-Menü über *KDE/System/Systemüberwachung* das Programm starten. Alternativ geben Sie über *KDE/Befehl ausführen* den Befehl `ksysguard` ein.

Das Programm präsentiert sich nach dem Start mit einem fast leeren Feld links und vier Graphen rechts. Warten Sie einen Moment und Sie sehen langsam von rechts die aktuellen Werte des Systems in die Graphen wandern.

#### Bedeutung der Graphen

**CPU Load** Der Graph zeigt die Auslastung der CPU an. Drei Kurven können sie erkennen. Blau ist die Auslastung des Prozessors durch Benutzeraktionen, Rot die Auslastung durch Systemprozesse und Hellbraun die Auslastung durch Programme, die mit einer veränderten Priorität gestartet wurden.

**Load Average 1min** Die auf eine Minute gemittelte Auslastung des Systems wird in diesem Graphen dargestellt. Da die CPU einem Programm immer die volle Leistung zur Verfügung stellt wenn nichts anderes anliegt, kommt es immer wieder zu CPU-Auslastungen von 100%. Dies können aber auch nur einzelne Peaks sein. Die über eine Minute gemittelte Auslastung gibt mehr Aufschlüsse über die Dauerbelastung eines Rechners.

**Physical Memory** Die Auslastung des physikalischen Speichers ist ebenso wie die CPU-Leistung ein Kriterium für die Belastung des Rechners. Auf den ersten Blick scheint es so, als ob der Speicher, selbst wenn kaum Programme auf dem System laufen, völlig belegt ist. Das stimmt auch soweit, da der Kernel den freien Speicher als Cache für Laufwerkszugriffe verwendet. Es werden nämlich auch hier drei Graphen angezeigt. Blau zeigt den Speicherbedarf der Programme an, dunkles Orange den Bedarf des Puffers, und Hellbraun ist die Farbe für den Festplattencache.

**Swap Memory** Neben dem physikalischen Speicher gibt es noch den Swap-Speicher, der als Auslagerungsdatei auf einem Datenträger realisiert ist. Beide zusammen ergeben den virtuellen Speicher, der dem System zur Verfügung steht. Hier heißt es genau zu beobachten. Denn wenn sich auf diesem Graphen viel tut, sollte man überlegen, ob nicht etwas mehr physikalischer Speicher dem Rechner gut tun würde.

#### Sensor-Browser

Im linken Fenster finden Sie für den lokalen Rechner die möglichen Sensoren, deren Werte Sie in den rechten Graphen anzeigen können. Einfach den Sensor anklicken und ihn in den entsprechenden Graphen ziehen. Einen Graphen können Sie übrigens löschen, indem Sie mit der rechten Maustaste darauf klicken und *Anzeige entfernen* betätigen.

Über die genaue Bedeutung der einzelnen Anzeige möchte ich jetzt nicht eingehen, da dies den Umfang dieses Kapitels sprengen würde.

#### Prozesstabelle

Neben der Systemauslastung können Sie sich auch die laufenden Prozesse (Programm bzw. Programmteile) anzeigen lassen. In einer Tabelle werden für die Prozesse der Name, die Prozessidentifikationsnummer (PID), die Systemauslastung durch Benutzer und das System, die Priorität des Prozesses, der Speicherbedarf, der

ausführende Benutzer und der aufrufende Befehl angezeigt. Die Bedeutung der einzelnen Spalten ergibt sich im Laufe dieses Skriptes. Hier an dieser Stelle seien Sie nur erwähnt.

Sie können auch eine Auswahl der anzuzeigenden Prozesse treffen. Sie können sich entweder *Alle Prozesse*, die laufen, sich anzeigen lassen oder sich auf die *Systemprozesse* beschränken. Auch können Sie sich nur mit den Benutzer abgeben und sich die *Benutzerprozesse* anzeigen. Wenn Sie ganz genügsam sind, begnügen Sie sich mit *Eigene Prozesse*.

Wenn Sie wissen wollen, welcher Prozess durch welchen Prozess gestartet wurde, dann wählen Sie die *Baum-Ansicht*. Den neuesten Stand der Prozessliste können Sie mit dem Button *Neu Aufbauen* erzeugen. Der wichtigste Button ist wohl *Beenden (kill)*, der durch seinen zweiten martialischen Namen schon anzeigt was er macht: Er tötet Prozesse. So können Prozesse, die sich sonst nicht mehr beenden lassen, gestoppt werden.

Wenn Sie ganz schnell an die Prozessliste herankommen wollen, dann betätigen Sie einfach die Tasten <STRG>+<ESC>. Es öffnet sich der KSysGuard mit der Ansicht der Prozeßtafel. Nun können Sie schnell einen Prozess töten oder sich über das Verhalten der Prozesse informieren.

## 2.3 Wie mache ich was?

### 2.3.1 Wie lege ich ein Programmicon auf dem Desktop an?

Am einfachsten können Sie ein Icon anlegen, wenn das Programm schon einen Eintrag im KDE-Menü besitzt. Mit der linken Maustaste auf den Punkt klicken und mit gedrückter Maustaste auf den Desktop ziehen. Dort dann im nun erscheinenden Menü auf *An diese Stelle kopieren* klicken. Fertig.

Wollen Sie ein Programm auf dem Desktop verewigen, was nicht im Menü erscheint, dann müssen Sie wie folgt vorgehen.

1. Rechte Maustaste auf den Desktop
2. Menüpunkt *Neu erstellen/Verknüpfung mit Programm ...* anklicken
3. Im Reiter *Allgemein* den Namen der Verknüpfung eingeben
4. Durch Klicken auf das Zahnrad-Symbol können Sie ein Symbol für das Programm auswählen
5. Im Register *Ausführen* geben Sie den Namen der Programmdatei an. Der Button *Durchsuchen* ermöglicht Ihnen das Dateisystem nach der Datei zu durchsuchen. Die meisten X-Window Programme finden Sie im Verzeichnis `/usr/X11R6/bin`, `/opt` und `/opt/kde3/bin`.
6. Mit *OK* bestätigen und fertig ist das Icon.

### 2.3.2 Wie kann ich ein Programm unter einem anderen Benutzer ausführen?

Manchmal ist es notwendig bestimmte Aufgaben unter einer anderen Benutzerkennung auszuführen, wie z. B. das Editieren von Systemkonfigurationsdateien. Auf diese Dateien können Sie aber als normaler Benutzer nicht zugreifen. Normalerweise müßten Sie sich jetzt als Benutzer ausloggen und wieder als privilegierter Benutzer (z. B. *root*) sich wieder einloggen. Das es auch anders geht, haben Sie beim Einsatz von YaST gesehen. Nach Eingabe des *root*-Passworts konnten Sie ohne Ab- und Anmeldung mit YaST arbeiten. Verantwortlich dafür ist das Programm *kdesu* (2.3.3).

Wenn Sie nun ein Programm als *root* starten wollen, müssen Sie im X-Terminal oder über *Befehl ausführen* (<ALT>+<F2>) einfach `kdesu PROGRAMM` eingeben. Also für den Editor NEdit z. B. `kdesu nedit`.

Wer den Befehl häufiger braucht, kann sich einen Menüeintrag oder Icon auf dem Desktop anlegen. In den Eigenschaften des Eintrags/Icons befindet sich der Abschnitt *Als anderer Benutzer ausführen*. Hier einfach die Checkbox anklicken und den gewünschten Benutzernamen eintragen.

### 2.3.3 KDE su

KDE su erlaubt es X-Window-Programme unter einer anderen Benutzeridentität auszuführen als die des aktuellen Benutzers. Es ist ein graphisches Frontend für su (8.1.2) unter dem KDE. Dabei werden die für das X-Window benötigten Einstellungen entsprechend gesetzt.

```
kdesu [OPTIONEN] [QT-OPTIONEN] [KDE-OPTIONEN] KOMMANDO
```

#### Optionen

--help	Grundoptionen
--help-qt	Spezielle Optionen zu Qt anzeigen
--help-kde	Spezielle Optionen zu KDE anzeigen
--help-all	Alle Optionen anzeigen
-c BEFEHL	Auszuführender Befehl
-f DATEI	
-u USER	Benutzerkennung angeben (Standard: root)
-n	Passwort nicht speichern
-s	Dämon anhalten (Alle Passwörter gehen verloren)
-t	Terminal-Ausgabe ermöglichen (keine Speicherung von Passwörtern)
-p PRI0	Priorität setzen: 0 <= prio <= 100, 0 ist die niedrigste. (Standard: 50)

#### Beispiel

Um den Editor NEdit unter dem Benutzer `wwwrun` auszuführen, müssen Sie folgenden Befehl anwenden.

```
kdesu -u wwwrun -c nedit
```

## 2.4 KDE-Programme

Inzwischen gibt es eine Reihe von Programmen, die speziell für das KDE geschrieben worden sind. Sie basieren auf den KDE-Bibliotheken und benötigen diese zwingend.

### 2.4.1 Konquerer

Der Konquerer ist nicht nur der mit dem KDE mitgelieferte Browser sondern auch gleichzeitig der Dateimanager. Da er relativ neu ist, hat er noch längst nicht alle Internet-Standards (besonders JavaScript) völlig implantiert.

### 2.4.2 KEdit und KWrite

Auch an Editoren wurde beim KDE gedacht. KEdit und KWrite sind hierfür entwickelt worden. KEdit ist die einfache Version des Editors, während KWrite neben Syntaxhighlighting auch Suchen und Ersetzen mit regulären Ausdrücken beherrscht.

### 2.4.3 KMail

Das Programm KMail ist, wie der Name es schon sagt, der Mail-Client vom KDE.

### 2.4.4 KSnapshot

Für Dokumentationen ist es oft sinnvoll einen Bildschirmschnappschuß machen zu können. Diese Funktion wird durch das Programm KSnapshot zur Verfügung gestellt.

### 2.4.5 KOffice

KOffice ist das fürs KDE entwickelte Office-Paket. Es enthält schon viele Programme, die aber entwicklungsbedingt noch nicht so viele Funktionen besitzen und teilweise auch etwas instabil sind.

#### KWord

KWord ist die Textverarbeitung.

#### KSpread

Die Tabellenkalkulation des KDE-Office ist KSpread. Allerdings beherrscht es wenige Datenformate, so daß hier das StarCalc vorzuziehen ist, daß auch Excel-Formate lesen und speichern kann.

### 2.4.6 Weitere KDE-Programme

#### KNode

Newsgroups können mit dem Programm KNode bearbeitet werden. Das Lesen und Senden von Artikeln werden von diesem Client-Programm erledigt. In Abschnitt 6.2.2 wird genauer auf das Programm eingegangen.

#### KAB

Das KDE-Adressbuch ist eine kleine Datenbank zur Verwaltung von Adressdaten wie Telefonnummer, eMail, Adresse u.s.w. Dabei können auch mehrere Adressen für eine Person eingegeben werden. Das KAB kann auch als Adressbuch für KMail benutzt werden.

#### KOrganizer

Einen Zeitplaner enthält KDE ebenfalls. Mit dem KOrganizer können Sie Ihre Termine planen und sich z. B. rechtzeitig an den Hochzeitstag erinnern lassen.

#### KCalc

Auch das KDE bringt einen Taschenrechner mit, der sogar die Punkt-vor-Strich-Regel beherrscht.

#### KJots

KJots ist ein kleines Programm um einfache Notizbücher zu verwalten. Sie können mehrere Notizbücher verwalten, die wiederum benannte Seiten enthalten in denen dann einfache Texte eingegeben werden können.

#### KNotes

Sie kennen doch sicherlich die gelben PostIt-Notizzettel. Das Programm KNotes ermöglicht Ihnen virtuelle Klebezettel auf den Bildschirm zu kleben und dort Ihre Erinnerungshilfen zu plazieren.

#### KBear

Auch einen komfortablen FTP-Client gibt es unterm KDE. Zwar kann man auch mit Hilfe des Konquerers Daten per FTP hin- und herschieben. Das Programm KBear ist durch die Arbeitsweise mit zwei Fenstern und dem Site-Manager wesentlich komfortabler.

## 2.5 Weitere X-Window-Programme

Neben den KDE-Programmen gibt es auch weitere nützliche Applikationen, die teils auch die KDE-Bibliotheken benutzen.

Eine hilfreiche Unterstützung bei der Erstellung von HTML-Seiten bieten die Programme **Bluefish** und **Quanta Plus**. Windows Nutzern wird die Ähnlichkeit der beiden Programme zum HTML-Editor Phase V auffallen.

Ein weiterer schöner Editor ist **NEdit**. Er benötigt kein KDE und stellt doch wichtige Funktionen wie Syntax-Highlighting und ‘Suchen und Ersetzen’ mit regulären Ausdrücken zur Verfügung.

## II

Arbeitsblatt  
Erste Schritte

## 2.1

Die Ausführung der folgenden Aufgaben ist wichtig für alle folgenden Aufgaben. Führen Sie daher bitte alles aus. Eventuell haben Sie die Aufgaben schon begleitend zum Unterricht bzw. zur Lektüre gemacht.

- 1 Legen Sie mit YaST2 den Benutzer *willi* (Willi Winzig) mit der UID 511 und dem Passwort *winzig* an. Geben Sie ihm zusätzlich noch die Mitgliedschaft in der Gruppe *disk*.
  - 2 Loggen Sie sich aus und als *willi* wieder ein.
  - 3 Konfigurieren Sie für *willi* das KDE so, daß der KDE-Menübutton angezeigt wird und das Eclipse-Design vom KDE verwendet wird.
  - 4 Loggen Sie sich aus und als *walter* wieder ein.
  - 5 Vor der ersten Verwendung einer Diskette muß diese formatiert werden. Dazu kann das Programm *KFloppy* verwendet werden. Unter welchem KDE-Menüpunkt ist es zu finden?
  - 6 Formatieren Sie nun als *walter* die Diskette mit dem Dateisystem *ext2* vollständig und geben Sie ihr den Namen "Disk".
  - 7 Installieren Sie mit YaST2 aus der Gruppe *Produktivität/Editoren/Andere* das Paket *pdksh*. Wieviel Platz benötigt es und wofür ist es da?
  - 8 In welchem Verzeichnis ist die Datei *nedit* installiert worden.
  - 9 Legen Sie für das Programm *nedit* ein Menüeintrag unter *KDE/Editoren*. Das Programm soll durch die Tastenkombinationen *<Windows>+<n>* gestartet werden. Testen Sie die Funktion.
  - 10 Legen Sie für das Programm *nedit* ein Icon auf dem Desktop an. Testen Sie die Funktion.
  - 11 Öffnen Sie mit *NEdit* die Datei */etc/motd*. Tragen Sie statt des vorhandenen Spruchs ein: "It's a good day to die ..." und speichern Sie die Datei wieder ab. Was passiert und warum.
  - 12 Um sich nicht ständig als *root* anmelden zu müssen um Systemdateien zu ändern, soll *NEdit* auch als *root* Benutzer arbeiten können. Wie gehen Sie vor?
  - 13 Führen Sie die Aufgabe 11 mit dem neuen Icon durch.
  - 14 Testen Sie den Erfolg, indem Sie sich auf Konsole 2 als *walter* einloggen. Hat es geklappt?
  - 15 Kehren Sie zum KDE-Desktop zurück.
  - 16 Beenden Sie alle geöffneten Programme.
  - 17 Legen Sie folgende Tastaturkürzel für Programme an.
 

Persönliches Verzeichnis	<i>&lt;Windows&gt;+&lt;E&gt;</i>
Konquerer (Webbrowser)	<i>&lt;Windows&gt;+&lt;K&gt;</i>
KMail	<i>&lt;Windows&gt;+&lt;M&gt;</i>
Yast2	<i>&lt;Windows&gt;+&lt;Y&gt;</i>
KCalc	<i>&lt;Windows&gt;+&lt;T&gt;</i>
- Testen Sie den Erfolg.
- 18 Starten Sie den Konquerer als Webbrowser. Gehen Sie zur Seite von GMX (<http://www.gmx.de>) und erstellen dort ein neues eMail-Konto.
  - 19 Tragen Sie in das KDE-Adressbuch Ihren Nachbarn mit seinem GMX-Konto ein.

## II

Arbeitsblatt  
Erste Schritte

2.2

- 20** Welche Angaben benötigen Sie um mit KMail eMails von ihrem GMX-Konto lesen und versenden zu können?
- 21** Suchen Sie die Angaben auf der Webseite von GMX und notieren Sie sich die Daten auf einem KNotes-“Zettel”. Konfigurieren Sie KMail entsprechend den Angaben für das GMX-Konto.
- 22** Schicken Sie Ihrem Nachbarn als Test eine eMail unter Verwendung des KDE-Adressbuchs.
- 23** Rufen Sie mit KMail die von ihren Nachbarn zugesandten eMails ab.
- 24** Führen Sie mit YaST2 erneut ein Online-Update durch. Sind neue Programme installiert worden?

.....  
*Notizen:*

.....



# Kapitel 3

## Arbeiten auf der Shell

*Wie kastriert man einen Windows-Benutzer?  
Man schneidet einfach das Mauskabel durch...*

*Anonymous*

Das KDE und damit auch das X-Window-System sind nur auf Linux aufgesetzt und dienen als Benutzeroberfläche. Es gibt aber auch eine wesentlich rudimentärere Möglichkeit für den Benutzer mit Linux zu arbeiten: die Shell (*engl. Muschel*). Andere Begriffe für die Shell sind Kommandozeile, Textmodus oder Prompt.

### 3.1 Starten der Shell

Die Shell steht auf jedem Linux-System zur Verfügung. Es gibt zwei Möglichkeiten mit der Shell zu arbeiten. Unter dem X-Window-System gibt es sogenannte X-Terminals, die eine solche Shell emulieren. Daneben steht die sogenannte Linux-Konsole zur Verfügung, die im Textmodus der Graphikkarte arbeitet und auch ohne X-Window-System funktioniert.

#### 3.1.1 Linux-Konsole

Die Linux-Konsole erreichen Sie aus dem X-Window-System, indem Sie die Tasten <STRG>+<ALT>+<F1> gleichzeitig drücken. Unter SuSE ist diese Konsole sehr bunt und mit einem Hintergrund versehen. Dies ist der Bildschirm, den Sie auch beim Booten des Systems gesehen haben. Die letzten Meldungen des Bootvorgangs stehen noch dort.

```
Welcome to SuSE Linux 9.0 (i586) - Kernel 2.4.21-199-athlon (tty1)
```

Mit dieser Meldung, die Ihnen die Distribution und das eingesetzte Betriebssystem mit Versionsnummer und Kernelversion anzeigt, begrüßt Sie das System. Danach steht der Login-Prompt mit Rechnername und wartet darauf, daß Sie sich an dieser Konsole anmelden.

```
defiant login:
```

Geben Sie nun Ihren Benutzernamen ein und schließen Sie die Eingabe mit der <Return>-Taste oder <Enter> ab. Sie werden nun nach Ihrem Passwort gefragt. Dies geben Sie ein und schließen die Eingabe wieder mit <Return> oder <Enter> ab. Wundern Sie sich nicht, denn Linux zeigt bei der Eingabe eines Passworts generell nichts an, noch nicht einmal die bei Windows üblichen Sternchen.

```
defiant login: walter
Password:
```

```
Have a lot of fun ...
walter@defiant:~>
```

Ein netter Spruch wird ausgegeben und der Prompt erscheint. Der Prompt zeigt an, daß die Shell bereit ist von Ihnen ein Kommando zu bekommen.

Wenn Sie SuSE benutzen, stehen Ihnen sechs von diesen Konsolen zur Verfügung. Mit den Tasten <ALT>+<F1> bis <ALT>+<F6> können Sie zwischen den Konsolen wechseln. Sie sehen, daß die Konsolen 2 bis 6 einen schwarzen Hintergrund besitzen und die gleiche Begrüßungsmeldung enthalten. Dabei weist das `tty` in der Klammer immer auf die jeweilige Konsole hin.

Wechseln Sie wieder zur ersten Konsole (<ALT>+<F1>). Geben Sie dort am Prompt `tty` ein und schließen Sie die Eingabe mit <Return> oder <Enter> ab. Der Befehl zeigt Ihnen nun an, auf welcher Konsole sie sich befinden.

```
walter@defiant:~> tty
/dev/tty1
```

Um die Shell zu beenden, geben Sie das Kommando `logout` ein. Denken Sie daran, daß jedes Kommando erst dann ausgeführt wird, wenn Sie es mit <Return> oder <Enter> abschließen.

```
walter@defiant:~> logout
```

```
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB (tty1)
```

```
defiant login:
```

Nun ist die Konsole wieder zur Anmeldung bereit. Jetzt müssen Sie nur noch zum X-Window-System zurückkehren. Das X-Window-System ist praktisch die siebente Konsole. Sie können also mit <ALT>+<F7> dorthin gelangen.

Wenn Sie schon auf der Konsole sind, können Sie alle Konsolen bzw. Terminals durch die Kombination von <ALT> und den Funktionstasten <F1> bis <F6> erreichen. Da die <ALT>-Taste unter KDE bzw. X-Window eine andere Bedeutung hat, können Sie hier die Konsolen über eine Kombination von <STRG>+<ALT> und den Funktionstasten <F1> bis <F6> erreichen.

### 3.1.2 X-Terminal

Für den Benutzer eines X-Window-Systems gibt es auch eine andere Methode um auf der Shell arbeiten zu können. Sogenannte X-Terminals emulieren die Linux-Konsole und bringen ein Textfenster auf den Bildschirm. Zum Starten des beim KDE mitgelieferten X-Terminals `konsole` brauchen Sie in der Menüleiste nur auf den Monitor mit der Muschel klicken.

An diesem Terminal brauchen Sie sich nicht anzumelden, da Sie sich durch die graphische Oberfläche schon in einer angemeldeten Sitzung befinden. Es erscheint also sofort der Prompt. Testen Sie doch mal, was der Befehl `tty` hier für ein Ergebnis zeigt.

```
ole@defiant:~> tty
/dev/pts/1
```

Es gibt eine ganze Reihe von X-Terminals mit z. T. sehr unterschiedlichen Fähigkeiten.

#### **xterm**

`xterm` ist der Klassiker unter den X-Terminals. Er ist sehr einfach. Sie können ihn starten, in dem Sie den Menüpunkt *Befehl ausführen ...* aus dem KDE-Menü wählen oder mit den Tasten <ALT>+<F2> das Ausführungsfenster aufrufen. Dort können Sie durch Eingabe von `xterm` das Fenster starten.

Mit dem Befehl `exit` können Sie das `xterm` dann wieder verlassen oder Sie drücken auf den X-Button oben rechts in der Titelleiste.

## konsole

Dieses Terminal ist Bestandteil des KDE-Desktops. Genau wie das `xterm` kann es durch den Menüpunkt *Befehl ausführen ...* durch Eingabe von `konsole` gestartet werden.

Im Gegensatz zu `xterm` kann `konsole` einiges mehr. So können z. B. mehrere unterschiedliche Konsolen in einem Fenster geöffnet werden. Es stehen dann auch spezielle Versionen zur Verfügung, die gleich dem Midnight-Commander, einen textbasierenden Dateimanager, starten oder als Benutzer `root` vorgeben.

Eine weitere Konsole öffnen Sie über den Button unten links mit der Beschriftung *Neu*. Sie können natürlich auch über die Menüleiste über das Menü *Sitzung* gehen.

Unter dem Menü *Bearbeiten* können Sie u. a. den Verlauf bearbeiten. Der Verlauf ist die Sammlung der Zeilen, die auf der Konsole gestanden haben. Natürlich werden nicht alle Zeilen gespeichert. Die Anzahl der zu speichernden Zeilen finden Sie unter *Einstellungen/Verlaufspeicher*. Voreingestellt sind 1000 Zeilen, was auch erst einmal reichen sollte. Sehr nützlich ist es, daß Sie den Verlauf auch nach einem Stichwort durchsuchen können.

Im Menü *Ansicht* können Sie die laufende Konsole (Sitzung) umbenennen und auf Aktivität oder Nichtaktivität überwachen lassen. Sie können Ihre Eingaben auch zu mehreren Sitzungen gleichzeitig senden und die Reihenfolge der Sitzungen in der Sitzungsleiste ändern.

Das Menü *Einstellungen* erlaubt individuelle Einstellungen für die `konsole`. Schriftgröße, Signale, Tastatur, Farben und Größe können hier eingestellt werden.

Unter *Hilfe* finden Sie, wie bei eigentlich jedem KDE-Programm Anleitungen und Informationen über das Programm.

## 3.2 Erste Befehle

Um die ersten Befehle auszuprobieren, starten Sie jetzt einfach mal die `konsole`. Der Prompt erscheint und die Shell ist bereit Ihre ersten Kommandos entgegenzunehmen. Es gibt übrigens nicht *die* Shell. Es gibt eine Reihe von Shells für Unix und Linux-Systeme. Die im Linux-Bereich am meisten verwendete Shell ist die Bash. Auch bei SuSE ist sie für den Benutzer voreingestellt. Ich werde mich in diesem Text auch hauptsächlich mit Ihr beschäftigen.

Der Prompt zeigt uns schon ein paar wichtige Informationen<sup>1</sup>. Welcher Benutzer ist auf welchem Rechner eingeloggt und in welchem Verzeichnis befindet er sich gerade. Das Verzeichnis `~` ist natürlich kein Verzeichnis. Es ist ein Alias oder Synonym für das Heimatverzeichnis des jeweiligen Benutzers. Das können wir auch schnell mal vergleichen. Geben Sie doch einfach mal hinter dem Prompt den Befehl `pwd` ein.

```
ole@defiant:~> pwd
/home/ole
ole@defiant:~>
```

Bei dem Benutzer `ole` steht das `~` für sein Heimatverzeichnis `/home/ole`. Die Heimatverzeichnisse der Benutzer heißen normalerweise wie ihre Besitzer und liegen im Verzeichnis `/home`. Es ist aber auch möglich die Heimatverzeichnisse anders zu benennen.

### 3.2.1 Anzeigen des Verzeichnisinhalts

Was ist denn in diesem Heimatverzeichnis an Dateien vorhanden? Auch hierfür müssen Sie einen Befehl eingeben. Das Kommando `ls` zeigt Ihnen den Inhalt des Verzeichnisses an.

```
ole@defiant:~> ls
Documents  public_html
```

Zwei Dateien liegen bei einem frisch eingerichteten Benutzer bei SuSE 8.0 im Heimatverzeichnis. Haben Sie auf der graphischen Oberfläche schon gearbeitet, so können hier noch mehr Dateien liegen.

---

<sup>1</sup>Der Prompt ist natürlich konfigurierbar und kann auch andere Informationen darstellen.

## Ausführliche Informationen

Die Informationen, die der Befehl `ls` gibt, sind sehr spärlich. Wir können aber dem Kommando `ls` sagen, daß es mehr Informationen ausgeben soll. Dies erfolgt über eine *Option*. Andere Namen dafür sind *Schalter* oder *Attribut*. Die Option für eine ausführliche Ansicht heißt `-l`. Das Minus vor dem Buchstaben `l` (wie "long") zeigt an, daß es sich um eine Option handelt.

```
ole@defiant:~> ls -l
insgesamt 8
drwxr-xr-x  2 ole   users      4096 Sep 25 12:25 Documents
drwxr-xr-x  2 ole   users      4096 Sep 25 12:25 public_html
```

Wenn Sie sich nun die Ausgabe anschauen, dann wird Ihnen sicherlich nicht die Bedeutung jeder dieser Spalten klar sein. Schauen wir uns dazu die Ausgabe noch einmal genauer an.

Typ und Rechte	Links	Besitzer	Gruppe	Größe	Änderungsdatum	Name
drwxr-xr-x	2	ole	users	4096	Sep 25 12:25	Documents
drwxr-xr-x	2	ole	users	4096	Sep 25 12:25	public_html

**Typ und Rechte** Die in dieser Spalte stehende Kombination aus Buchstaben und Minus-Zeichen gibt Auskunft über den Dateityp und die Rechte, die für diese Datei gelten. Das erste Zeichen steht für den Dateityp. Das `d` in dem Beispiel steht für ein Verzeichnis. Würde ein Minuszeichen dort stehen, dann handelt es sich um eine normale Datei. Die Tabelle unten zeigt ihnen die anderen möglichen Typen. Auf die Bedeutung der einzelnen Typen möchte ich hier nicht eingehen.

Zeichen	Bedeutung
-	gewöhnliche Datei
d	Verzeichnis (directory)
c	zeichenorientierte Gerätedatei (character device)
b	blockorientierte Gerätedatei (block device)
p	FIFO-Pipeline(named pipe)
l	symbolischer Link (symbolic link)
s	Netzwerksocket (socket)

Die restlichen neun Zeichen geben in Gruppen zu jeweils drei Zeichen die Rechte auf die Datei an. Dabei steht das `r` für "Lesen" (read), das `w` für "Schreiben" (write) und das `x` für "Ausführen" (execute). Die ersten drei Zeichen geben die Rechte für den Besitzer der Datei an. Die zweiten drei Zeichen geben Auskunft über die Rechte der Gruppe, die dieser Datei zugeordnet wurde und die letzten drei Zeichen geben dann an, was der Rest der Welt mit dieser Datei darf.

**Links** Im Gegensatz zu Windows kann eine Datei bei Linux mehrere Namen besitzen. Bei Verzeichnissen sind das mindestens zwei. Einmal der Name selbst und dann im Verzeichnis die Datei mit dem Namen `."`. Eine Datei wird erst dann gelöscht, wenn auch der letzte Name, bzw. Hardlink, wie die Dateinamen auch genannt werden, entfernt wurde.

**Besitzer** Hier wird der Besitzer der Datei angegeben. Wenn ein Benutzer eine neue Datei anlegt, wird er automatisch der Besitzer der Datei.

**Gruppe** Jeder Datei wird eine Gruppe von Benutzern zugeteilt, die extra Rechte bekommen können. Ist ein Benutzer in dieser Gruppe, so gelten die Rechte der Gruppe für ihn, wenn er nicht gerade der Besitzer der Datei ist.

**Größe** Die Größe der Datei wird in dieser Spalte in Byte angegeben.

**Änderungsdatum** Normalerweise zeigt der Befehl `ls` an dieser Stelle das Datum der letzten Änderung der Datei an. Durch bestimmte Attribute kann der Befehl auch dazu bewegt werden das Datum der Erstellung oder das Datum des letzten Zugriffs anzuzeigen.

**Name** Hier steht der Name der Datei. Die Sortierung der aufgelisteten Dateien erfolgt lexikalisch. Beachten Sie dabei, daß die Shell zwischen Groß- und Kleinschreibung unterscheidet und die Großbuchstaben vor den Kleinbuchstaben einsortiert werden.

### Versteckte Dateien

Wie Sie oben gesehen haben, enthält das Heimatverzeichnis scheinbar nur zwei Dateien bzw. Verzeichnisse. Das dem nicht so ist, zeigt der Einsatz der Option `-a`.

```
ole@defiant:~> ls
Documents public_html
ole@defiant:~> ls -a
.          .Xresources  .emacs      .profile    .xim          .xtalkrc
..         .bash_history .exrc       .urlview    .xinitrc      Documents
.Xdefaults .bashrc     .kermrc     .xcoralrc   .xserverrc.secure public_html
.Xmodmap   .dvipsrc    .muttrc     .xemacs     .xsession
```

Anstatt zwei werden nun 23 Dateien angezeigt. Diese neu angezeigten Dateien bezeichnet man als *Versteckte Dateien*. Sie werden bei normalen Dateioperationen nicht berücksichtigt. Diese versteckten Dateien finden Sie auch in anderen Betriebssystemen. Dort wird durch ein Attribut der Datei angezeigt, ob eine Datei versteckt ist oder nicht. Wenn Sie sich die Liste der Dateien im obigen Beispiel noch einmal genau anschauen, finden Sie recht schnell die Gemeinsamkeit der versteckten Dateien heraus.

**Dateien, deren Name mit einem Punkt beginnt, bezeichnet man als Versteckte Dateien. Sie werden bei normalen Dateioperationen nicht berücksichtigt.**

Eine besondere Bedeutung haben die Dateien `.'` und `..`. Sie sind Bestandteil des Verzeichnisses. Der Name `.'` ist ein weiterer Link für das Verzeichnis in dem sich diese Datei befindet. Dagegen verweist der Name `..` auf das Elternverzeichnis des aktuellen Verzeichnisses. Wenn Sie das obige Beispiel als Grundlage nehmen, so ist `.` gleichzusetzen mit `/home/ole` und `..` mit `/home`.

Zwei Optionen können auch miteinander kombiniert werden. Eine ausführliche Liste aller Dateien bekommen Sie durch die Kombination der Optionen `-l` und `-a`.

```
ole@defiant:~> ls -a -l
insgesamt 92
drwxr-xr-x  5 ole   users      4096 Sep 25 12:25 .
drwxr-xr-x  9 root  root       4096 Sep 25 12:25 ..
-rw-r--r--  1 ole   users      5742 Sep 25 12:25 .Xdefaults
-rw-r--r--  1 ole   users      1305 Sep 25 12:25 .Xmodmap
lrwxrwxrwx  1 root  root         10 Sep 25 12:25 .Xresources -> .Xdefaults
-rw-----  1 ole   users         0 Sep 25 12:25 .bash_history
-rw-r--r--  1 ole   users      1691 Sep 25 12:25 .bashrc
-rw-r--r--  1 ole   users       208 Sep 25 12:25 .dvipsrc
-rw-r--r--  1 ole   users      1637 Sep 25 12:25 .emacs
-rw-r--r--  1 ole   users      1124 Sep 25 12:25 .exrc
-rw-r--r--  1 ole   users       164 Sep 25 12:25 .kermrc
-rw-r--r--  1 ole   users     2286 Sep 25 12:25 .muttrc
-rw-r--r--  1 ole   users       952 Sep 25 12:25 .profile
-rw-r--r--  1 ole   users       311 Sep 25 12:25 .urlview
-rw-r--r--  1 ole   users     7913 Sep 25 12:25 .xcoralrc
drwxr-xr-x  2 ole   users      4096 Sep 25 12:25 .xemacs
-rw-r--r--  1 ole   users     3407 Sep 25 12:25 .xim
-rwxr-xr-x  1 ole   users     2324 Sep 25 12:25 .xinitrc
```

```

-rw-r--r--    1 ole    users      1101 Sep 25 12:25 .xserverrc.secure
-rwxr-xr-x    1 ole    users      2804 Sep 25 12:25 .xsession
-rw-r--r--    1 ole    users       119 Sep 25 12:25 .xtalkrc
drwxr-xr-x    2 ole    users      4096 Sep 25 12:25 Documents
drwxr-xr-x    2 ole    users      4096 Sep 25 12:25 public_html

```

Wer es kürzer mag, kann die Buchstaben auch zusammenziehen. Dabei ist es völlig egal, ob Sie `-al` oder `-la` schreiben.

```

ole@defiant:~> ls -al
insgesamt 92
drwxr-xr-x    5 ole    users      4096 Sep 25 12:25 .
drwxr-xr-x    9 root   root      4096 Sep 25 12:25 ..
-rw-r--r--    1 ole    users     5742 Sep 25 12:25 .Xdefaults
-rw-r--r--    1 ole    users     1305 Sep 25 12:25 .Xmodmap
...

```

### Parameter und Jokerzeichen

Der `ls` Befehl kann nicht nur den Inhalt des aktuellen Verzeichnisses anzeigen. Wenn Sie z. B. den Inhalt des Verzeichnisses `Documents` sehen wollen, geben Sie folgendes an.

```

ole@defiant:~> ls -al Documents
insgesamt 12
drwxr-xr-x    2 ole    users      4096 Sep 25 12:25 .
drwxr-xr-x    5 ole    users      4096 Sep 25 12:25 ..
-rw-r--r--    1 ole    users     1106 Sep 25 12:25 .directory

```

Der Name des Verzeichnisses `Documents` bezeichnet man als Parameter des Befehls `ls`. Die Angabe des Verzeichnisnamens erfolgte hier relativ zum Arbeitsverzeichnis. Um das Elternverzeichnis des Arbeitsverzeichnisses anzuzeigen können wir den Dateinamen `..` verwenden.

```

ole@defiant:~> ls ..
conny  lost+found  ole  perl  vnc  walter  willi

```

Um zur Wurzel zu gelangen müssen Sie sogar noch ein Verzeichnis weiter im Verzeichnisbaum hochgehen.

```

ole@defiant:~> ls ../../
bin  cdrom  dvd  floppy  lib          media  opt    proc  sbin  usr  windows
boot dev    etc  home    lost+found  mnt    pcmcia root  tmp  var

```

Sie können die Verzeichnisse auch direkt adressieren, wenn Sie den Verzeichnisnamen mit einem Slash `/` beginnen lassen. Jetzt wird nämlich nicht vom aktuellen Arbeitsverzeichnis aus gesehen, sondern von der Dateiwurzel.

```

ole@defiant:~> ls /
bin  cdrom  dvd  floppy  lib          media  opt    proc  sbin  usr  windows
boot dev    etc  home    lost+found  mnt    pcmcia root  tmp  var
ole@defiant:~> ls /home
conny  lost+found  ole  perl  vnc  walter  willi
ole@defiant:~> ls /home/ole
Documents  public_html

```

Neben den Verzeichnissen können Sie auch eine Auswahl unter den Dateien treffen, die Sie sich anzeigen lassen wollen. Hilfreich sind dabei die sogenannten *Jokerzeichen* oder *Wildcards*.

Dazu schauen Sie sich doch mal das Verzeichnis `/etc` an, in dem sich die Konfigurationsdateien des Linux-Systems befinden.

```
ole@defiant:~> ls /etc
DIR_COLORS      inittab          php.ini
HOSTNAME        inputrc          pnm2ppa.conf
Mottrc          insserv.conf     powerd.conf
SuSE-release    ioctl.save       ppp
...
```

Es werden sehr viele Dateien angezeigt. Sie können die Anzahl der angezeigten Dateien verringern, wenn Sie andere Kriterien für die Auswahl stellen. So könnten Sie sich nur die Dateien anzeigen lassen, die mit dem Buchstaben “a” beginnen.

```
ole@defiant:~> ls /etc/a*
/etc/a2ps-site.cfg /etc/aliases      /etc/at.deny
/etc/a2ps.cfg      /etc/aliases.db   /etc/auto.master
/etc/adjtime       /etc/asound.state /etc/auto.misc

/etc/alsa.d:
emu10k1 sbawe
```

Der Asterisk “\*” steht für eine beliebige Anzahl beliebiger Zeichen und das Fragezeichen “?” steht für *ein* beliebiges Zeichen. Auffällig an dem Beispiel oben ist, daß nicht nur der Name des Verzeichnis angegeben wird, sondern auch der Inhalt. Dies ist eine der Eigenschaften von `ls`. Ist ein Verzeichnis ein Ziel, dann wird der Inhalt des Verzeichnis angezeigt. Mit der Option `-d` kann dies unterbunden werden.

```
ole@defiant:~> ls -d /etc/a*
/etc/a2ps-site.cfg /etc/aliases      /etc/asound.state /etc/auto.misc
/etc/a2ps.cfg      /etc/aliases.db   /etc/at.deny
/etc/adjtime       /etc/alsa.d       /etc/auto.master
ole@defiant:~> ls -ld
drwxr-xr-x    5 ole    users      4096 Sep 25 12:25 .
```

Dies wirkt, wie Sie oben sehen, auch beim aktuellen Arbeitsverzeichnis. Es wird nur die Information über das aktuelle Verzeichnis angezeigt und nicht über die enthaltenen Dateien.

## Inodes

Wie schon oben erwähnt, kann eine Datei mehrere Namen (Links) besitzen. Dies wird dadurch ermöglicht, daß die Benennung der Daten im System über eine sogenannte Inode-Nummer erfolgt. Als Inode bezeichnet man den Platz, an dem die Informationen über die Datei gespeichert werden. Das Betriebssystem merkt sich also nicht den Namen der Datei sondern die Inode-Nummer. Die Namen sind nur für uns “dumme” Benutzer, die sich mehr unter dem Namen “amaretti.rezept.txt” vorstellen können, als unter einer Inode-Nummer “45312”. Die Verzeichnisse sind im Prinzip nichts anderes als Tabellen, die den Zusammenhang zwischen Dateinamen und der Inode-Nummer herstellen.

Die Inode-Nummer können sie sich mit `ls` unter Verwendung der Option `-i` anzeigen lassen.

```
ole@defiant:~> ls -i
 1095 Documents      1097 public_html
ole@defiant:~> ls -i /home
 1074 conny          31873 ole          112382 vnc          286867 willi
  11 lost+found     223209 perl         207194 walter
```

## Hilfe

Es gibt noch viel mehr Optionen für `ls`, als die ich Ihnen gerade vorgestellt habe. Um eine kurze Übersicht über die Funktion und vorhandenen Optionen eines Befehls zu erhalten, können Sie den Shellbefehl mit

der Option `--help` aufrufen. Mit dem doppelten Minuszeichen wird bei der letzten Option dem Befehl signalisiert, daß es sich nicht um die einzelnen Optionen `-h`, `-e`, `-l` und `-p` handelt, sondern um eine Option mit dem langen Namen `help`.

```
ole@defiant:~> ls --help
Benutzung: ls [OPTION]... [DATEI]...
Auflistung von Informationen der DATEIen (Standardvorgabe ist das momentane
Verzeichnis). Alphabetisches Sortieren der Einträge, falls weder -cftuSUX
noch --sort angegeben.
```

```

-a, --all           Einträge, die mit . beginnen, nicht verstecken.
-A, --almost-all   Keine Anzeige implizierter . und ..
-b, --escape        Ausgabe octaler Repräsentation für nicht-druck-
...
```

Natürlich ist der Hilfetext etwas länger als die Konsolenhöhe. Auf der Konsole können Sie ohne Probleme scrollen (wenn es nicht mehr als 1000 Zeilen sind). Damit die Anzeige nicht durchscrollt, sondern seitenweise anzeigt, können Sie einen sogenannten Pager benutzen. Ein solcher Pager ist der Befehl `more`. Sie leiten einfach die Ausgabe des Befehls mit dem Zeichen `|` an den Pager weiter.

```
ole@defiant:~> ls --help | more
ole@defiant:~> ls /etc | more
ole@defiant:~> ls -l / | more
ole@defiant:~> ls /dev | more
```

Mit der Leertaste blättern Sie die Seiten des Pagers `more` durch. Erreicht der Pager das Ende des Textes bzw. der Ausgabe, so beendet er sich automatisch.

## Zusammenfassung

Der Befehl `ls` zeigt die in den Verzeichnissen enthaltenen Dateien an. Mit Optionen kann man die Art und Weise beeinflussen, wie Befehle funktionieren. Die Parameter geben an, womit gearbeitet werden soll.

Die Optionen `-aldi` sollten sie sich für den Befehl `ls` merken.

```

-a      Zeigt auch die versteckten Dateien an (-all)
-l      Zeigt ausführliche Liste zu den Dateien an (long)
-d      Zeigt Verzeichnis anstatt Inhalt an (-directory)
-i      Zeigt Inodes an (-inode)
--help  Zeigt die Hilfe für den Befehl an
```

## 3.2.2 Erstellen und Löschen von Verzeichnissen und Dateien

Ein Verzeichnissystem macht nur Sinn, wenn es Dateien und Verzeichnisse enthält. Fürs Erstellen und Löschen von Dateien und Verzeichnissen stehen auf der Shell viele Programme zur Verfügung.

### Erstellen

Eine einfache Art und Weise eine leere Datei zu erstellen ist der Befehl `touch`.

```
ole@defiant:~> touch testdatei
ole@defiant:~> ls -l
insgesamt 8
drwxr-xr-x  2 ole  users      4096 Sep 25 12:25 Documents
drwxr-xr-x  2 ole  users      4096 Sep 25 12:25 public_html
-rw-r--r--  1 ole  users         0 Sep 26 14:37 testdatei
```



Eigentlich soll der Befehl `touch` die Zeit der letzten Änderung und des letzten Zugriffs auf die aktuelle Zeit stellen. Existiert die angegebene Datei nicht, dann wird einfach eine neue Datei erstellt.

Ein neues Verzeichnis zu erstellen ist ebenso einfach. Der Befehl heißt hier `mkdir`.

```
ole@defiant:~> mkdir test
ole@defiant:~> ls
Documents public_html test testdatei
ole@defiant:~> mkdir beruf hobby
ole@defiant:~> ls
Documents beruf hobby public_html test testdatei
```

Sie müssen sich beim Befehl `mkdir`, wie auch bei `touch`, nicht auf einen Parameter beschränken. Sie können ruhig mehrere Dateien bzw. Verzeichnisse angeben, die erstellt werden sollen.

Auch eine Angabe des Verzeichnis mit seinen Elternverzeichnissen ist möglich. Allerdings müssen diese vorher existieren.

```
ole@defiant:~> cd /
ole@defiant: /> mkdir /home/ole/test/neuerOrdner
ole@defiant: /> ls /home/ole/test
neuerOrdner
```

## Löschen

Was man erstellt, muß man auch löschen können. Dies gilt für Verzeichnisse wie auch für Dateien. Der Befehl `rmdir` (*remove directory*) sorgt für das Löschen von Verzeichnissen.

```
ole@defiant:~> rmdir beruf
ole@defiant:~> rmdir test
rmdir: »test«: Das Verzeichnis ist nicht leer
```

Allerdings löscht `rmdir` nur leere Verzeichnisse. Sobald sich nur eine Datei oder ein Verzeichnis darin befindet, wird das Löschen des Verzeichnis verweigert.

Dateien entledigt man sich dagegen durch den Befehl `rm` (*remove*).

```
ole@defiant:~> rm testdatei
ole@defiant:~> rm test
rm: »test« ist ein Verzeichnis
```

Mit Verzeichnissen tut sich `rm` dagegen schwer und verweigert das Löschen mit einem entsprechenden Hinweis.

Dies ist auch die normale Arbeitsweise der Shellkommandos. Klappt ein Befehl, so erscheint in der Regel kein Kommentar, sondern der Prompt fordert den Benutzer auf wieder tätig zu werden. Nur wenn Fehler aufgetreten sind, melden sich die Kommandos noch einmal beim Benutzer.

Sie können aber `rm` überreden auch Verzeichnisse zu löschen. Mit der Option `-r` (*recursive*) löscht `rm` Verzeichnisse *mit* ihrem Inhalt.

```
ole@defiant:~> rm -r test
```

Manchmal fragt `rm` auch nach, ob eine Datei gelöscht werden soll. Ist einem das zu lästig, gerade wenn ganze Verzeichnisstrukturen gelöscht werden sollen, dann sorgt der Schalter `-f` für Ruhe. Er sorgt dafür, daß die Dateien ohne Nachfrage gelöscht werden.

Sie sollten aber mit dem Befehl vorsichtig sein. Stellen Sie sich doch einfach mal vor was passiert, wenn der Administrator folgendes eingibt:

```
defiant:~$ rm -rf /
```

## Der Editor vi

Der Editor, den Sie auf jedem Linux/Unix-System vorfinden ist der **vi** oder einer seiner Klone. Seine Benutzerführung ist etwas eigenwillig und gewöhnungsbedürftig. Da er aber auf jedem System vorhanden ist, sollten Sie sich doch mit ihm etwas auskennen.

Eine Datei zum Bearbeiten rufen Sie ganz einfach auf, indem Sie den Namen hinter dem Befehl **vi** schreiben.

```
ole@defiant:~> vi meinedatei
```

Existiert die Datei nicht, so geht der **vi** von einer leeren neuen Datei aus, wie sie auch in der Statuszeile ganz unten sehen können.

```
"meinedatei" [Neue Datei]                                0,0-1      Alles
```

Der **vi** besitzt verschieden Modi für die Bearbeitung von Dokumenten. Beim Start landen Sie im Bearbeitungsmodus, der spezielle Funktionen zur Bearbeitung der Dokumente zur Verfügung stellt. Diese Funktionen werden nicht wie von den meisten Editoren gewohnt über ein Menü zur Verfügung gestellt, sondern durch Tastaturkürzel.

Sie müssen, wenn Sie Text schreiben wollen, in den Eingabemodus wechseln. Dazu drücken Sie im Bearbeitungsmodus die Taste **<I>**. Die Statuszeile zeigt nun **EINFÜGEN** bzw. **INSERT** an.

```
-- EINFÜGEN --                                           0,1      Alles
```

Hier können Sie nun Text eingeben, mit den Richtungstasten sich durch den Text bewegen und mit der Taste **<ENTF>** bzw. **<DEL>** Zeichen rechts vom Cursor löschen.

Um wieder in den Bearbeitungsmodus zu wechseln, drücken Sie einfach die Taste **<ESC>**. Um sicher zu gehen, können Sie die Taste auch mehrfach drücken. **vi** quittiert das dann mit einem Piepton.

Im Bearbeitungsmodus können Sie das Zeichen unter dem Cursor entweder wie im Eingabemodus mit der Taste **<DEL>** löschen oder indem Sie die Taste **<X>** betätigen. Es gibt aber auch weitere reichende Löschkommandos im Zusammenhang mit der Taste **<D>**. So löscht **<D> <W>** die Zeichen bis zum Wortende inklusive des folgenden Leerzeichens. Mit **<D> <E>** löscht er ebenfalls den Rest vom Wort, ohne aber das Leerzeichen mitzunehmen. Mit der Kombination **d\$** (**<D> <SHIFT>+<4>**) löscht er den Text vom Cursor bis zum Zeilenende. Und um die ganze Zeile, in der sich der Cursor befindet, zu löschen, betätigen Sie einfach zweimal die Taste **<D>** (**<D> <D>**). Sie können auch durch das Voranstellen einer Zahl vor das Kommando angeben, wie oft das Kommando ausgeführt werden soll. So können Sie mit **<4> <D> <D>** vier Zeilen löschen oder mit **<3> <D> <W>** die drei Worte rechts vom Cursor.

Die Mehrfachausführung klappt auch mit dem Eingabemodus. Einfach vorm Einschalten des Eingabemodus eine Zahl eintippen. Nach Beendigung des Eingabemodus wird das frisch Eingegebene mehrfach ausgegeben. So können Sie z. B. mit der Tastenkombination **<7> <5> <I> <-> <ESC>** 75 Minus-Zeichen in den Text einfügen.

Vom Bearbeitungsmodus können Sie auch einen Kommandomodus aufrufen. Dies erfolgt durch den Doppelpunkt (**<SHIFT>+<:>**). Der Doppelpunkt erscheint in der unteren Statuszeile und Sie können dann weitere Befehle eintippen, die Sie mit der **<RETURN>**-Taste abschließen.

```
:q      Beendet den vi
:q!     Beendet den vi ohne Nachzufragen
:wq     Beendet den vi und speichert das Dokument vorher
```

Bei Linux wird der **vi**-Klon **vim** mitgeliefert und durch den Befehl **vi** ausgeführt. Mit dem Befehl **vimtutor** erhalten Sie ein kleines Tutorial mit Übungen zur Bedienung des **vim**.

### 3.2.3 Kopieren, Verschieben und Umbenennen von Dateien

Für das Kopieren, Verschieben und Umbenennen von Dateien brauchen wir hauptsächlich zwei Befehle: **cp** (*copy*) und **mv** (*move*).

## Kopieren

Um eine Kopie einer Datei zu erstellen benutzen Sie den Befehl `cp`. Der Befehl braucht eine Datei, die es zu kopieren gilt, (Quelldatei) und den neuen Namen der Datei (Zieldatei).

```
ole@defiant:~/test> vi themenvorschlag.txt
ole@defiant:~/test> ls -l
insgesamt 4
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
ole@defiant:~/test> cp themenvorschlag.txt themenvorschlag.txt.alt
ole@defiant:~/test> ls -l
insgesamt 8
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
-rw-r--r--  1 ole      users      2123 0kt  9 10:26 themenvorschlag.txt.alt
ole@defiant:~/test>
```

So kann einfach eine Sicherungskopie einer Datei erstellt werden. In den meisten Fällen werden aber eine oder mehrere Dateien in ein anderes Verzeichnis kopiert, wie z. B. in `/floppy` für die Diskette. In diesem Fall wird anstatt der Zieldatei ein Zielverzeichnis angegeben. Erkennt der Befehl `cp`, daß das Ziel ein existierendes Verzeichnis ist, dann kopiert er die Datei ohne den Namen zu ändern in das Verzeichnis. Existiert kein Verzeichnis, so wird die Angabe als Dateiname verstanden und die Datei dementsprechend kopiert. Übrigens: `cp` überschreibt in der normalen Einstellung existierende Dateien gnadenlos.

```
ole@defiant:~/test> mkdir backup
ole@defiant:~/test> ls -l backup/
insgesamt 0
ole@defiant:~/test> cp themenvorschlag.txt backup
ole@defiant:~/test> ls -l backup/
insgesamt 4
-rw-r--r--  1 ole      users      2123 0kt  9 10:40 themenvorschlag.txt
```

Sie können auch mehrere Dateien zum Kopieren angeben, wenn Ihr Ziel ein Verzeichnis ist.

```
ole@defiant:~/test> cp /etc/enscript.cfg /etc/esd.conf backup
ole@defiant:~/test> ls -l backup
insgesamt 16
-rw-r--r--  1 ole      users      5983 0kt  9 10:43 enscript.cfg
-rw-r--r--  1 ole      users        77 0kt  9 10:43 esd.conf
-rw-r--r--  1 ole      users      2123 0kt  9 10:40 themenvorschlag.txt
```

Wenn Sie das aktuelle Verzeichnis als Ziel angeben wollen, machen Sie das einfach durch den Punkt `‘.’`. Genauso können Sie das Elternverzeichnis mit `‘..’` als Ziel angeben. Um zu sehen, welche Dateien kopiert worden sind, benutzen Sie den Befehl `cp` einfach mit dem Schalter `-v` (*verbose*). Diese auch als “Blubberschalter” bekannte Option veranlaßt den Befehl dazu ausführlich über seine Tätigkeit zu berichten. Sie werden bei vielen Shell-Befehlen die Option `-v` wiederfinden. Sie können natürlich auch für die Liste der zu kopierenden Dateien die Jokerzeichen wie den Asterisk `“*”` verwenden.

```
ole@defiant:~/test> cp -v /etc/f* .
>/etc/fam.conf< -> >./fam.conf<
>/etc/fb.modes< -> >./fb.modes<
>/etc/fdprm< -> >./fdprm<
>/etc/filesystems< -> >./filesystems<
>/etc/fstab< -> >./fstab<
>/etc/ftpusers< -> >./ftpusers<
```

## Verschieben und Umbenennen

Der Befehl `mv` arbeitet im Prinzip genau wie der Befehl `cp`. Allerdings existiert die Quelldatei bzw. Quelldateien danach nicht mehr. Im Prinzip ist ein Verschieben und Umbenennen nichts anderes als eine Änderung des Verzeichniseintrags. Nur wenn zwischen zwei Partitionen verschoben werden soll, muß die Datei erst kopiert und dann die alte Datei gelöscht werden. Dadurch ist Verschieben im Normalfall schneller als Kopieren.

Um also eine Datei umzubennen benutze ich den Befehl `mv` unter Angabe von altem und neuem Namen.

```
ole@defiant:~/test> ls -l
insgesamt 12
drwxr-xr-x  2 ole      users      4096 0kt  9 10:43 backup
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
-rw-r--r--  1 ole      users      2123 0kt  9 10:26 themenvorschlag.txt.alt
ole@defiant:~/test> mv themenvorschlag.txt.alt themenvorschlag.old
ole@defiant:~/test> ls -l
insgesamt 12
drwxr-xr-x  2 ole      users      4096 0kt  9 10:43 backup
-rw-r--r--  1 ole      users      2123 0kt  9 10:26 themenvorschlag.old
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
```

Um die Syntax eines Shell-Kommandos kurz und knapp darzustellen, werden Syntaxregeln mit einer bestimmten Symbolik verwendet.

Der `mv` Befehl kann als Syntaxregel so ausgedrückt werden:

```
mv ALTERNAME NEUERNAME
```

Die groß geschriebenen Worte stehen für den Wert, der dort eingetragen werden muß.

Ähnlich wie bei `cp` führt die Angabe eines existierenden Verzeichnis als Ziel dazu, daß die Datei nicht umbenannt, sondern in das Verzeichnis unter Beibehaltung des Namens verschoben wird.

```
ole@defiant:~/test> mv themenvorschlag.old backup
ole@defiant:~/test> ls -l
insgesamt 8
drwxr-xr-x  2 ole      users      4096 0kt  9 11:09 backup
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
ole@defiant:~/test> ls -l backup
insgesamt 8
-rw-r--r--  1 ole      users      2123 0kt  9 10:26 themenvorschlag.old
-rw-r--r--  1 ole      users      2123 0kt  9 10:40 themenvorschlag.txt
```

Wenn Sie ein Verzeichnis als Ziel angeben, können Sie nicht nur eine Datei verschieben, sondern eine ganze Liste von Dateien.

```
ole@defiant:~/test> touch datei1.txt datei2.txt datei3.txt
ole@defiant:~/test> ls -l
insgesamt 8
drwxr-xr-x  2 ole      users      4096 0kt  9 11:10 backup
-rw-r--r--  1 ole      users           0 0kt  9 11:12 datei1.txt
-rw-r--r--  1 ole      users           0 0kt  9 11:12 datei2.txt
-rw-r--r--  1 ole      users           0 0kt  9 11:12 datei3.txt
-rw-r--r--  1 ole      users      2123 0kt  9 10:25 themenvorschlag.txt
ole@defiant:~/test> mv d*.txt backup
ole@defiant:~/test> ls -l backup
insgesamt 8
-rw-r--r--  1 ole      users           0 0kt  9 11:12 datei1.txt
```



```

-c, --no-create           Keine Dateien erzeugen.
-d, --date=ZEICHENKETTE  Lesen der ZEICHENKETTE und statt der momentanen
                          Zeit verwenden.
-f                        (ignoriert)
-m                        Nur Modifikationszeit ändern.
-r, --reference=FILE      Die Zeiten dieser Datei anstatt der momentanen Zeit
                          verwenden.
-t MARKE                  verwenden von [[HH]JJ]MMTSSmm[.ss] statt der
                          momentanen Zeit.
  --time=WORT             Setzen der Zeit die von WORT angegeben wird:
                          access, atime, (wie -a), mtime, modify (wie -m).
  --help                  Anzeigen dieser Hilfe und beenden.
  --version                Ausgabe der Versionsinformation und beenden.

```

Beachten Sie das die drei Zeitformate die von den Optionen -d, -t und dem veraltete Argument erkannt werden alle verschieden sind.

Berichten Sie Fehler an <bug-fileutils@gnu.org>.

Andere Befehle, wie z. B. der Packer zip liefern diese Erklärungen schon, wenn Sie ohne Parameter aufgerufen werden.

```

ole@defiant:~/test> zip
Copyright (C) 1990-1999 Info-ZIP
Type 'zip -L' for software license.
Zip 2.3 (November 29th 1999). Usage:
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
  The default action is to add or replace zipfile entries from list, which
  can include the special name - to compress standard input.
  If zipfile and list are omitted, zip compresses stdin to stdout.
-f  freshen: only changed files  -u  update: only changed or new files
-d  delete entries in zipfile    -m  move into zipfile (delete files)
-r  recurse into directories     -j  junk (don't record) directory names
-O  store only                   -l  convert LF to CR LF (-ll CR LF to LF)
-1  compress faster              -9  compress better
-q  quiet operation              -v  verbose operation/print version info
-c  add one-line comments        -z  add zipfile comment
-@  read names from stdin        -o  make zipfile as old as latest entry
-x  exclude the following names  -i  include only the following names
-F  fix zipfile (-FF try harder) -D  do not add directory entries
-A  adjust self-extracting exe   -J  junk zipfile prefix (unzipsfx)
-T  test zipfile integrity       -X  eXclude eXtra file attributes
-y  store symbolic links as the  link instead of the referenced file
-R  PKZIP recursion (see manual)
-e  encrypt                      -n  don't compress these suffixes

```

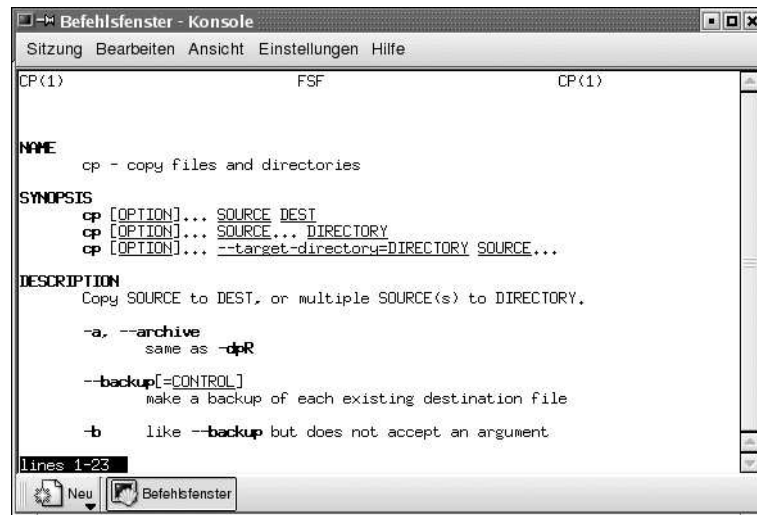
### 3.3.2 Manual-Pages

Für weitergehende Erläuterungen wurden die Manual Pages, oder auch nach Ihrem Verwaltungsprogramm Man-Pages genannt, entwickelt. Die Man-Pages sind praktisch die Bedienungsanleitungen für die Shell-Befehle.

Nach einem `man cp` wird das Dokument entpackt und dann mit dem passenden Pager angezeigt.

Mit den Richtungstasten können Sie nun hin- und herscrollen. Die Betätigung der Taste <Q> beendet dann das Man-Programm.

Die Programme `whatis` und `apropos` basieren auf `man`. `whatis` liefert eine Kurzbeschreibung für einen Befehl, während `apropos` diese Kurzbeschreibungen nach einem Begriff durchsucht.

Abbildung 3.1: Ausgabe des Befehls `man cp`

```

ole@defiant:~/test> apropos login
logname (1)          - print user's login name
/etc/login.defs (5) [login.defs] - Login configuration
login_tty (3)        - tty utility functions
getlogin (3)         - get user name
sulogin (8)          - Single-user login
slogin (1)           - OpenSSH SSH client (remote login program)
nologin (8)          - politely refuse a login
faillog (8)          - examine faillog and set login failure limits
nologin (5)          - prevent non-root users from logging into the system
ssh (1)              - OpenSSH SSH client (remote login program)
utmp (5)             - login records
wtmp (5)             - login records
....
  
```

```

ole@defiant:~/test> whatis ssh
ssh (1)              - OpenSSH SSH client (remote login program)
  
```

Die Manual-Pages enthalten nicht nur Informationen zu Befehlen, sondern auch zu Konfigurationsdateien, Kernelfunktionen u.a.

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....



## III

Arbeitsblatt  
Einführung in die Shell

## 3.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

- 1 Wechseln Sie vom X-Window-System auf die Konsole 2.
- 2 Loggen Sie sich dort als *walter* ein.
- 3 In welchem Verzeichnis befinden Sie sich?
- 4 Auf welchem Terminal arbeiten Sie gerade?
- 5 Führen Sie den Befehl `rm -rf *` aus. Was bewirkt der Befehl?
- 6 Legen Sie eine leere gewöhnliche Datei mit dem Namen *meiohmei* an.
- 7 Legen Sie drei leere gewöhnlichen Dateien mit den Namen *achdugrueneneune*, *achduliebermeinvater* und *achdukannstmichmal* mit einem Befehl an.
- 8 Lassen Sie sich nun den Verzeichnisinhalt anzeigen.
- 9 Lassen sich sich ausführlichere Informationen über den Verzeichnisinhalt anzeigen.
- 10 Lassen Sie sich nun für die vier Dateien die Inode-Nummern anzeigen.
- 11 Legen Sie mit dem *vi* die Datei *wasichschonimmersagenwollte* mit dem folgenden Inhalt an.

```
Linux Unser, Der Du bist im Kernel,
Geöffnet sei Dein Swap-File,
Dein KDE komme,
Dein Bash-Command geschehe,
Wie in Red Hat so auch in Suse.
Unser täglich Login gib uns heute,
Und vergib uns unsere zu kleine Festplatte,
Wie auch wir vergeben Dir Deiner Installation.
Und führe uns nicht in die Kernel-Panik,
Sondern erlöse uns von Microsoft.
Denn Dein ist das Netz Und die Festplatte Und die Stabilität
In Ewigkeit
Enter!
Anonymous
```

- 12 Legen Sie von der Datei *wasichschonimmersagenwollte* eine Kopie namens *ichhabsgesagt* an.
- 13 Lassen Sie sich die Inode Nummern der Dateien anzeigen.
- 14 Benennen Sie nun *ichhabsgesagt* nach *ichsagte* um.
- 15 Lassen Sie sich die Inode Nummern der Dateien wiederum anzeigen und vergleichen Sie das Ergebnis mit Aufgabe 13. Was sehen sie?
- 16 Legen Sie das Verzeichnis *abfall* an.
- 17 Lassen Sie sich den Inhalt des aktuellen Verzeichnis ausführlich anzeigen. Wie unterscheidet sich das Verzeichnis von den gewöhnlichen Dateien?
- 18 Lassen Sie sich ausführlich den Inhalt des Verzeichnis *abfall* anzeigen.

## III

Arbeitsblatt  
Einführung in die Shell

## 3.2

- 19 Kopieren Sie die Datei `ichsagte` in das Verzeichnis `abfall`.
- 20 Kopieren Sie die Dateien, die mit `achdu` beginnen in das Verzeichnis `abfall`. Lassen Sie sich dabei die Aktionen des Programms anzeigen.
- 21 Lassen Sie sich wieder ausführlich den Inhalt des Verzeichnis `abfall` anzeigen.
- 22 Kopieren Sie alle Dateien aus dem Verzeichnis `etc`, die mit `c` beginnen, in das Verzeichnis `abfall`. Welche Fehler treten dabei auf?
- 23 Versuchen Sie über die Hilfe herauszubekommen, ob mit dem `cp` es prinzipiell möglich ist Verzeichnisse zu kopieren.
- 24 Wechseln Sie in das Verzeichnis `abfall` und lassen Sie sich das aktuelle Verzeichnis anzeigen.
- 25 Wechseln Sie zur Verzeichniswurzel und lassen Sie sich den Inhalt des Verzeichnis ausführlich anzeigen. Um was für Dateien handelt es sich hier?
- 26 Wechseln Sie in das Verzeichnis `root`. Was passiert?
- 27 In welchem Verzeichnis sind Sie nun?
- 28 Wechseln Sie in das Verzeichnis `etc`.
- 29 Wann wurde das Verzeichnis `cron.daily` erstellt?
- 30 Was bewirkt der Schalter `-G` beim Befehl `ls`
- 31 Testen Sie den Schalter `-G` bei der ausführlichen Anzeige des Inhalts des Verzeichnis `cron.daily`.
- 32 Öffnen Sie die Datei `do_mandb` aus `cron.daily` im `vi`. Welchen Hinweis bekommen Sie.
- 33 Beenden Sie den `vi` ohne zu speichern.
- 34 Wechseln Sie wieder in das Heimatverzeichnis von `walter`.
- 35 Öffnen Sie die Datei `wasichschonimmersagenwollte` mit dem `vi`.
- 36 Löschen Sie das Wort `KDE`.
- 37 Schreiben Sie an der Stelle jetzt `Gnome` hinein.
- 38 Gehen Sie an den Anfang des Dokuments und fügen Sie dort eine Leerzeile ein.
- 39 Fügen Sie nun in dieser Zeile 75 mal den Asterisk ein.
- 40 Löschen Sie die Zeile "Dein Bash-Command geschehe,"
- 41 Verlassen Sie den `vi` ohne das Dokument zu speichern.
- 42 Wenn Sie noch Zeit haben, dann starten Sie das Lernprogramm für den `vi` und arbeiten es durch, soweit Sie kommen.

# Kapitel 4

## Die Shell I

### 4.1 Was ist eine Shell?

Die Shell ist nur ein Linux-Programm. Sie dient als Verbindungsstelle zwischen dem Benutzer und dem Betriebssystem. Ihre Aufgabe ist es, die eingegebene Kommandozeile so umzuformen, daß das Betriebssystem sie interpretieren kann. Daher gibt es auch nicht nur eine Shell, sondern eine Anzahl von Programmen, die diese Vermittlungsaufgabe übernehmen können.

Die älteste, heute noch in der Praxis benutzte Shell ist die von Stephen L. Bourne Mitte der 70er Jahre für "Unix Version 7" entwickelte *Bourne-Shell*. Heute ist sie nur noch selten in ihrer ursprünglichen Form zu finden.

Die *C-Shell* wurde an der University of California in Berkeley entwickelt. Sie wurde an die Programmiersprache C angelehnt. Allerdings sind die Möglichkeiten der C-Shell kaum vergleichbar mit denen von C.

Eine zur Bourne-Shell weitgehend kompatible Shell ist die von David Korn entwickelte *Korn-Shell*. Sie wurde aber mit einem größeren Funktionsumfang ausgestattet.

Die Linux-Standard-Shell ist die im Rahmen des GNU-Projekts entwickelte *Bash*, die Fähigkeiten der Korn- und C-Shell miteinander verbindet. Auf diese Shell werde ich noch genauer eingehen, da Sie auch Prüfungsstoff der ersten LPI-Prüfung ist.

Eine Übersicht über die verschiedenen Shells liefert Tabelle 4.1.

Im Normalfall, wenn man sich auf einem Linux-Rechner einloggt, benutzt man `/bin/bash`. Die Standard-Shell eines Benutzers wird durch einen Eintrag in der `/etc/passwd` festgelegt. Der Name der Login-Shell wird in der Umgebungsvariablen `SHELL` gespeichert. Die Variable enthält nicht zwingend, was viele glauben, den Namen der aktuellen Shell. Den Namen der aktuellen Shell können Sie über die Variable `0` ermitteln. Den Inhalt der Umgebungsvariablen können sie mit dem Befehl `echo` ausgeben lassen.

```
ole@enterprise:~> echo $SHELL
/bin/bash
ole@enterprise:~> echo $0
/bin/bash
```

Weitere Erläuterungen zu den Umgebungsvariablen finden Sie im Verlauf dieses Skripts im Abschnitt 5.2.

Um eine andere Shell zu starten, geben sie einfach das Shellkommando mit seinem Pfad ein. Dies startet eine Kindprozess in dem die neue Shell läuft. Um die Shell zu beenden, geben Sie `exit` ein.

Im folgenden Beispiel wird durch die Eingabe `ksh` zur *Public Domain Korn Shell*<sup>1</sup> gewechselt. Dies können Sie an der veränderten Prompt-Einstellung sehen, da die Korn-Shell die Abkürzung `~` fürs Heimatverzeichnis nicht kennt. Trotzdem zeigt die Variable `SHELL` immer noch die Bash als Shell an. Die Variable `0` hingegen zeigt die aktuelle Shell an.

```
ole@enterprise:~> echo $SHELL $0
```

---

<sup>1</sup>Das Paket *pdksh* muß eventuell nachinstalliert werden.

Aufruf	Shell	Kommentar
<b>sh</b>	Bourne Shell	Oft ein Verweis auf die Bash
<b>bash</b>	Bourne Again Shell	
<b>ksh</b>	Public Domain Korn Shell	Freie Version der Korn-Shell
<b>csh</b>	C Shell	Oft ein Verweis auf die Tenex C Shell
<b>tcsh</b>	Tenex C Shell	Verbesserte Version der C Shell

Tabelle 4.1: Übersicht über die häufigsten Shells

```

/bin/bash /bin/bash
ole@enterprise:~> ksh
ole@enterprise:/home/ole> echo $SHELL $0
/bin/bash ksh
ole@enterprise:/home/ole> exit
ole@enterprise:~> echo $SHELL $0
/bin/bash /bin/bash

```

Es gibt drei verschiedene Situationen, in denen eine Shell arbeiten kann. Als *interaktive Login-Shell*, als *interaktive Shell* und als *nichtinteraktive Shell*.

Eine interaktive Login-Shell, meistens nur Login-Shell genannt, wird direkt nach dem Einloggen gestartet und Sie können direkt mit ihr Arbeiten.

Eine interaktive Shell ermöglicht ebenfalls das direkte Arbeiten. Allerdings wurde sie nicht durch einen Login-Vorgang gestartet. Dies kann z. B. durch die Eingabe des Shellprogramms in einer anderen Shell geschehen, wie Sie im obigen Beispiel sehen konnten, oder z. B. durch das Starten einer Terminalemulation auf der graphischen Oberfläche.

Mit einer nichtinteraktiven Shell können Sie nicht am Prompt arbeiten. Sie wird nur zum Ausführen eines Shell-Skripts gestartet und beendet sich nach dem Ende des Skripts automatisch. Damit dient diese Shell nur als eigenständige Umgebung für die Ausführung von vordefinierten Shell-Befehlen ohne Möglichkeiten für den Benutzer direkt einzugreifen.

#### 4.1.1 chsh

Das Kommando **chsh** ändert dauerhaft die Login-Shell.

```
chsh [-s LOGINSHELL] [USER]
```

Der Benutzer kann nur die Shell für sich selbst ändern, während **root** das für jeden Benutzer machen kann. Dabei ist aber der Benutzer auf die Shells eingeschränkt, die in der Datei `/etc/shells` aufgelistet werden. Wird die Option `-s` nicht angegeben, so erfolgt die Änderung der Daten im interaktiven Modus. Vor jeder Änderung wird das Kennwort des Benutzers zur Sicherheit abgefragt.

##### Optionen

`-s LOGINSHELL` Die neue Login-Shell

##### Beispiel

Der Benutzer will in Zukunft die Korn-Shell verwenden. Er ändert dies mit

```
chsh -s /usr/bin/ksh
```

## 4.2 Die Bash

Die erste programmierbare Shell wurde von Steve L. Bourne Mitte der 70er Jahre entwickelt und war eine der ersten UNIX-Shells. Als aufwärtskompatibler Nachfolger wurde später die *Bourne-Again-Shell* oder

kurz *Bash* entwickelt. Sie wurde von Brian Fox und Chet Ramey als Bestandteil des GNU-Betriebssystems entworfen, für das allerdings bisher nur der Compiler, Tools wie z.B. Editoren und die Shell existieren. Der wichtigste Teil des Betriebssystems, der Betriebssystemkern, fehlt allerdings noch. Da alle Teile freie Software sind, sind sie Bestandteile der freien UNIX-Varianten Linux und Free-BSD, die im Prinzip hauptsächlich aus dem Betriebssystemkern mit Treibern bestehen und die anderen Teile, die für ein funktionierendes System notwendig sind, wie eben Compiler, Shell, Editoren, Tools zum Dateihandling oder ähnliches, aus anderen Quellen beziehen. So ist die Bash die Standard-Shell von Linux, wobei auch hier dem Benutzer die Wahlfreiheit gelassen wird, und man jede andere UNIX-Shell verwenden kann.

Im Prinzip ist die Bash nichts weiter als ein normales Programm, was die Kommandos des Benutzers an das Betriebssystem und andere Programme weiterleitet und deren Ausgaben wieder dem Benutzer zur Verfügung stellt.

Als allererstes sollten Sie bei der Bash beachten, daß zwischen Groß- und Kleinschreibung peinlich genau unterschieden wird. So sind mit den Namen `ahrschlecker.txt`, `AHRSCHECKER.TXT` und `AhrSchlecker.txt` im Gegensatz zu DOS/Windows drei verschiedene Dateien gemeint. Da die meisten Befehle auch nur ausführbare Dateien sind, gilt es natürlich auch für diese. In meinen Schulungen habe ich am Anfang oft Teilnehmer über dieses Verhalten schimpfen hören. Aber dieses Verhalten hat schon seinen Sinn. Schauen Sie sich doch mal folgendes Motto einer Tierschützerin an:

Ich bin gut zu Vögeln.

Meinen Sie jetzt noch immer, daß die Verwendung von Groß- und Kleinschreibung überflüssig ist?

Der Prompt zeigt nicht nur die Bereitschaft der Shell an ein neues Kommando zu empfangen. Er kann auch mit Informationen ausgestattet werden. Bei SuSE wird voreingestellt der eingeloggte Benutzer, der Rechner und das Verzeichnis angezeigt. Verantwortlich für das Aussehen des Prompts sind die Variablen `PS1` und `PS2`.

```
tapico@defiant:/etc/news >
```

Unvollständige Befehlszeilen quittiert die Shell (Bourne und Bash) durch Ausgabe des Hilfsprompts, der durch die Variable `PS2` festgelegt wird. In der Regel handelt es sich um ein Größer-Zeichen gefolgt von einem Leerzeichen. Häufigste Ursache dafür ist ein fehlendes zweites Anführungszeichen. Durch Eingabe dieses Zeichens wird die Befehlszeile vervollständigt und der Befehl kann ausgeführt werden.

```
ole@enterprise:~> echo "Hallo Welt
> "
Hallo Welt
```

Häufig ist es nötig einen Befehl zu wiederholen oder einen Befehl leicht abzuwandeln. Unter Linux steht eine Eingabewiederholung ähnlich dem `DOSKEY` unter Windows zur Verfügung: die History. Diese wird im wesentlichen über die Cursortasten Auf und Ab gesteuert.

Sollten Sie mal mit dem Platz in einer Zeile nicht auskommen, so kann eine Eingabezeile problemlos verlängert werden, indem man anstatt `<RETURN>` zu drücken unmittelbar vor `<RETURN>` einen Backslash `\` eingibt.

Gerade am Anfang können Sie ungewollt in Programme raten, mit denen Sie sich nicht auskennen. Hier ein paar Tips, wie Sie solche Programme beenden können.

- Die Taste `<Q>` oder `<Q><RETURN>` für "quit" führen oft zum Erfolg. (z. B. `less` (4.5.10), `more` (4.5.9) und `top` (12.8.3))
- Die Kombination `<STRG>+<D>` steht für Ende-der-Datei. Damit können Sie vor allem Programme beenden, die auf eine Eingabe warten. (z. B. `cat` (4.5.2), wenn es von der Standardeingabe liest, und `at` (13.1.1)). Sollten Sie sich aber direkt in der Shell befinden, dann werden Sie damit ausgeloggt.
- Eine etwas härtere Methode ein Programm zu beenden ist die Kombination `<STRG>+<C>`. Hiermit wird das laufende Programm aufgefordert, sich sofort zu beenden.
- Sollten Sie sich im Editor `vi` verfangen haben, kommen sie durch die Tastenfolge `<ESC><:;><Q><!><RETURN>` heraus.

- Der Editor `joe` ist da viel leichter zu beenden. Er reagiert auf die Kombination `<STRG>+<C>` als normalen Ende-Befehl.

### 4.2.1 Kommandosyntax

Ein Kommando ist eine Folge von Zeichenketten, die durch ein oder mehrere Leerzeichen oder Tabulatoren getrennt werden. Die ersten Zeichenkette legt den Namen des Kommandos fest. Die weiteren Zeichenketten werden als Parameter bezeichnet.

KOMMANDO [PARAMETER1] [PARAMETER2] ... [PARAMETERn]

Dabei gehören die Leerzeichen und Tabulatoren nicht zu den Parametern, sondern werden nur als Trennzeichen gewertet. Soll der Parameter hingegen Leerzeichen oder Tabulatoren enthalten, so muß der Ausdruck in einfache (') oder doppelte (") Anführungszeichen gesetzt werden.

#### Beispiel

```
echo Ich          bin          da!
echo "Ich         bin         da!"
```

Bei den Parametern wird zwischen Argumenten und Optionen unterscheiden. Optionen werden durch ein - in der Kurzform<sup>2</sup> oder -- in der langen Form eingeleitet.

**Das Kommando legt fest *was* gemacht werden soll. Die Argumente bestimmen *womit* und die Optionen *wie*.**

## 4.3 Arbeiten mit Verzeichnissen

Nach dem Einloggen befinden Sie sich immer in Ihrem Heimatverzeichnis<sup>3</sup>. Dies wird generell abgekürzt durch das Tilde-Zeichen `~`. Die folgenden Arbeitsweisen des Tilde-Zeichens werden als Tilde-Ausdehnung (engl. *tilde expansion*) bezeichnet.

```
~           Heimatverzeichnis des aktuellen Benutzers
~walter     Heimatverzeichnis von dem Benutzer walter
~+          Das aktuelle Arbeitsverzeichnis
~-          Das alte Arbeitsverzeichnis
```

Ihre Funktionsweise können Sie im folgenden Beispiel verfolgen.

```
ole@enterprise:~/test> echo ~
/home/ole
ole@enterprise:~/test> echo ~walter
/home/walter
ole@enterprise:~/test> echo ~+
/home/ole/test
ole@enterprise:~/test> cd ../Documents/
ole@enterprise:~/Documents> echo ~-
/home/ole/test
```

### 4.3.1 pwd

Der Befehl `pwd` (*Print Working Directory*), der Bestandteil der Bash ist, zeigt das Verzeichnis an, in dem Sie gerade arbeiten.

```
pwd
```

---

<sup>2</sup>Nur ein Buchstabe lang.

<sup>3</sup>„Homedirectory“. Verwechseln Sie dieses Verzeichnis nicht mit dem Verzeichnis `/home`, in dem sich die Heimatverzeichnisse befinden.

### 4.3.2 cd

Um in ein anderes Verzeichnis zu wechseln benutzt man das Kommando `cd` (*Change Directory*). Dabei ist `cd` kein eigenständiges Programm sondern wie `pwd` ebenfalls Bestandteil der Bash.

`cd [DIRECTORY]`

Das Zielverzeichnis kann dabei auf absolute oder relative Weise angegeben werden. Bei der absoluten Darstellung geht man immer von der Wurzel (*root*) aus. Diese Pfadangaben beginnen immer mit einem Slash `/`. Bei der relativen Pfadangabe wird dagegen von dem aktuellen Verzeichnis als Startpunkt ausgegangen. Jedes Verzeichnis enthält zwei besondere Verzeichnisse. Das Verzeichnis `.` steht für das Verzeichnis selber, während `..` für das übergeordnete Verzeichnis (Elternverzeichnis) steht.

Um wieder zurück ins Heimatverzeichnis zu kommen gibt es mehrere Wege. Sie können zum einen den absoluten oder relativen Pfad zum Verzeichnis angeben. Kürzer geht es aber mit den Befehlen `cd ~` oder ganz kurz nur mit `cd`.

### 4.3.3 ls

Eine Reihe von Befehlen ermöglicht es, sich den Inhalt eines Verzeichnisses anzusehen. Der am häufigsten benutzte Befehl ist `ls` (*LiSt*).

`ls [OPTIONEN] [DATEINAME]`

Für `DATEINAME` können Namen von Dateien oder Verzeichnissen<sup>4</sup> verwendet werden. Dabei kann, um eine Menge von Dateinamen zu bilden, Metazeichen (Joker) verwendet werden. Die bekanntesten Joker sind dabei das Fragezeichen `?`, das für ein einzelnes beliebiges Zeichen steht, und der Asterisk `*`, der für eine beliebige Anzahl beliebiger Zeichen steht.

#### Optionen

<code>-a</code>	Anzeige aller Dateien, auch derjenigen, die mit <code>.</code> beginnen.
<code>-A</code>	Anzeige aller Dateien außer <code>.</code> und <code>..</code> .
<code>-c</code>	Sortiert zusammen mit <code>-t</code> Dateien nach dem Datum der letzten Änderung der Inode (Verwaltungsinformationen). Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit <code>-l</code> . ( <code>ls -ctl</code> )
<code>-d</code>	Verzeichnisse in der Liste der Argumente werden wie andere Dateien behandelt. Unterdrückung der Durchsuchung des Inhalts von Verzeichnissen.
<code>-l</code>	Anzeige ausführlicher Dateiinformatoren. Viele Schalter haben nur im Zusammenhang mit <code>-l</code> eine Bedeutung, da sie nicht automatisch die Information anzeigen sondern nur vorbereiten (z. B. <code>-e</code> oder <code>-k</code> ).
<code>-r</code>	Ausgabe in umgekehrter Sortierreihenfolge
<code>-t</code>	Sortiert Dateien nach dem Datum der letzten Änderung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des Datums erfolgt zusammen mit <code>-l</code> .
<code>-x</code>	Spaltenweise Ausgabe der Dateinamen, im Gegensatz zu <code>-C</code> jedoch waagerecht geordnet.
<code>-m</code>	Ausgabe der Dateinamen als durch Komma getrennte Liste.
<code>-F</code>	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen: <code>/</code> für Verzeichnisse, <code>*</code> für ausführbare Dateien, <code>@</code> für Links, <code> </code> für FIFOs und <code>=</code> für Sockets.
<code>-R</code>	Rekursive Anzeige, d. h. es werden nicht nur Verzeichnisse nach ihrem Inhalt durchsucht, sondern auch darin enthaltene (Unter-) Verzeichnisse. Nicht zusammen mit <code>-d</code> anwendbar.

Eine Auflistung von weiteren Schaltern finden Sie in Tabelle 4.2. Es gibt eine Reihe von alternativen Optionen. Z. B. ist `--tabsize=zahl` eine Alternative zu `-T zahl`. Diese Optionen sind selbsterklärend (wenn man Englisch kann und Phantasie hat) und werden stets mit doppeltem Minuszeichen eingeleitet, da sie sonst wegen ihrer Länge mit Zusammenfassungen der herkömmliche Schalter verwechselt werden können. Man kann sie sich mit `ls --help` anzeigen lassen.

Über die Ausgabe des `ls`-Befehls sollten sie noch wissen:

- Die standardmäßig vorgegebene Reihenfolge der Ausgabe beruht auf einer lexikalischen Sortierung, wobei erst nach Ziffern, dann nach Großbuchstaben und zuletzt nach Kleinbuchstaben sortiert wird.

<sup>4</sup>In Linux sind Verzeichnisse im Prinzip auch nur Dateien.

<b>-b</b>	Oktale Darstellung nichtdruckbarer Zeichen im Dateinamen (wichtiger Notnagel, wenn <b>ls</b> ein merkwürdiges Verhalten zeigt).
<b>-e</b>	Zusammen mit <b>-l</b> Ausgabe von Datum und Uhrzeit. (Nicht S.u.S.E.)
<b>-f</b>	Unsortierte Ausgabe (Dateien werden in der Reihenfolge angezeigt, in der sie im Verzeichnis eingetragen sind). Beinhaltet automatisch <b>-a</b> .
<b>-g</b>	existiert nur aus Kompatibilitätsgründen. Wird ignoriert.
<b>-i</b>	Anzeige der Inode-Nummer vor dem Dateinamen.
<b>-k</b>	Zusammen mit <b>-l</b> Anzeige der Dateigröße in Kilobyte (aufgerundet) vor dem Dateinamen (siehe auch <b>-s</b> )
<b>-n</b>	Ersetzt (in Verbindung mit <b>-l</b> ) die Namen von Benutzer und Gruppe durch die entsprechenden Nummern (UID und GID).
<b>-o</b>	Keine farbliche Kennzeichnung der Dateitypen
<b>-p</b>	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen. Ausführbare (gewöhnliche) Dateien werden im Gegensatz zu <b>-F</b> nicht gesondert gekennzeichnet.
<b>-q</b>	Nichtdruckbare Zeichen im Dateinamen werden als <b>?</b> dargestellt (vgl. auch <b>-b</b> ).
<b>-s</b>	Zusammen mit <b>-l</b> Angabe der Dateigröße in Blocks vor dem Dateinamen. Ist die Umgebungsvariable <b>POSIXLY_CORRECT</b> definiert, ist die Blockgröße 512 Byte, ansonsten (i. a.) ein Kilobyte (vgl. auch <b>-k</b> ).
<b>-u</b>	Sortiert zusammen mit <b>-t</b> Dateien nach dem Datum der letzten Benutzung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit <b>-l</b> . ( <b>ls -utl</b> )
<b>-w zahl</b>	Legt die maximale Breite ( <i>width</i> ) der Ausgabezeile auf <b>zahl</b> fest. Wird durch <b>-l</b> aufgehoben.
<b>-B</b>	Keine Anzeige von Backup-Dateien (Enden mit <b>~</b> )
<b>-C</b>	Spaltenweise Ausgabe der Dateinamen, senkrecht geordnet (dies ist im übrigen die Standardeinstellung. Sie wird durch <b>-l</b> aufgehoben).
<b>-D</b>	Erzeugt eine für einen bestimmten emacs-Modus geeignete Ausgabe.
<b>-G</b>	Unterdrückt bei <b>-l</b> die Anzeige der Gruppe
<b>-I muster</b>	Unterdrückt die Ausgabe von Informationen über Dateien, deren Name <b>muster</b> entspricht. <b>muster</b> kann Joker enthalten. Der Schalter kann mehrfach (mit unterschiedlichen Mustern) benutzt werden. Beachten Sie, daß auch hier Verzeichnisse besonders behandelt werden.
<b>-L</b>	Zeigt im Zusammenhang mit <b>-l</b> und <b>-I</b> bei symbolischen Links die Informationen über die Originaldatei und nicht die Daten des symbolischen Links an.
<b>-N</b>	Dateien mit Sonderzeichen im Namen werden nicht durch Anführungszeichen gekennzeichnet. (Voreinstellung)
<b>-Q</b>	Dateien mit Sonderzeichen im Namen werden durch Anführungszeichen gekennzeichnet. Die Sonderzeichen selbst werden in der aus der Programmiersprache C bekannten Form dargestellt.
<b>-S</b>	Sortierung der Dateinamen nach der Größe der Datei
<b>-T zahl</b>	Definiert den Abstand von Tabstopps (Tabulatoren).
<b>-U</b>	Unsortierte Ausgabe (im Gegensatz zu <b>-f</b> wird nicht der Schalter <b>-a</b> aktiviert).
<b>-X</b>	Sortiert die Ausgabe nach Dateieindung! Erstes Sortierkriterium ist der Namensteil nach dem letzten <b>.</b> im Namen. Z. B. werden Dateien, die mit <b>.c</b> enden, vor Dateien angezeigt, die mit <b>.o</b> enden. Dateien "ohne Endungen" werde zuerst angezeigt.
<b>-1</b>	Angabe der Dateinamen in nur einer Spalte (war in älteren UNIX-Systemen voreingestellt).

Tabelle 4.2: Weitere Optionen des Befehls **ls**



Kennzeichen	Bedeutung
-	gewöhnliche Datei
d	Verzeichnis (directory)
c	zeichenorientierte Gerätedatei (character device)
b	blockorientierte Gerätedatei (block device)
p	FIFO-Pipeline(named pipe)
l	symbolischer Link (symbolic link)
s	Netzwerksocket (socket)

Tabelle 4.3: Die Kennzeichnung der Dateiartern durch `ls -l`

- Die Benutzung von `ls` ohne Schalter ergibt eine Liste in vertikal sortierten Spalten.
- Dateien, die mit einem Punkt `.` beginnen, werden als versteckte Dateien bezeichnet. Sie werden nur unter Verwendung der Schalter `-a` und `-A` angezeigt. Im Allgemeinen handelt es sich hier um Konfigurationsdateien.
- Die Farben bei der Ausgabe von `ls` werden in der Datei `/etc/DIR_COLORS` festgelegt. Um die Farben selbst bestimmen zu können, muß die Datei als `.dir_colors` in das Heimatverzeichnis kopiert werden.

**Beispiel**

Ausgabe des Befehls mit der Option `-Fl`

```
tapico@defiant:~/test > ls -Fl
total 4
lrwxrwxrwx  1 ole      users          9 May  9 22:00 hallo -> hallowelt*
-rwxr-xr-x  1 ole      users          36 May  9 21:56 hallowelt*
-rw-r--r--  2 ole      users         401 May  9 22:04 liste
-rw-r--r--  2 ole      users         401 May  9 22:04 myliste
drwxr-xr-x  2 ole      users        1024 May  9 21:55 mytest/
prw-r--r--  1 ole      users           0 May  9 21:54 testpipe|
```

Dabei enthalten die Spalten von links nach rechts gelesen: Dateityp und Rechte, Anzahl der Hardlinks, Besitzer, Gruppe, Größe in Byte, Monat, Tag, Uhrzeit, Dateiname und Dateityp (durch `-F`). Den Dateityp aus der ersten Spalte können Sie aus der Tabelle 4.3 entnehmen.

**4.3.4 dir**

Der Befehl `dir` wird in den einzelnen Distributionen unterschiedlich behandelt. Ursprünglich ein eigenständiges Programm ist er heute entweder ein Alias für `ls` oder für `ls -l`.

```
ole@enterprise:~> type dir
dir is aliased to 'ls -l'
```

**4.3.5 vdir**

Das Kommando `vdire` ist fast identisch mit dem Kommando `ls`. Ohne Schalter eingegeben wirkt es wie `ls -l`. Die meisten Schalter sind mit denen von `ls` identisch. Mehr Informationen entnehmen Sie bitte der Manual-Page oder der Texinfo zu `vdire`.

**4.3.6 mkdir**

Der Befehl `mkdir` wird dazu benutzt neue Verzeichnisse zu erstellen.

`mkdir [OPTIONEN] [VERZEICHNISPFADLISTE]`

Wird die Option `-p` nicht mit angegeben, so müssen die Elternverzeichnisse für das neue Verzeichnis existieren.

#### Optionen

`-p` Erzeugt auch die nötigen Elternverzeichnisse, wenn diese nicht existieren.  
`-m RECHTE` Erlaubt gleich das Setzen der Rechte für das neue Verzeichnis wie durch `chmod`

#### Beispiele

Dieses Kommando legt das Verzeichnis `tex/linux` an. Dabei muß das Verzeichnis `tex` existieren.

```
mkdir tex/linux
```

Dieser Befehl legt einen kompletten Verzeichnispfad an. Dabei werden alle Verzeichnisse erzeugt, wenn sie noch nicht existieren.

```
mkdir -p tex/linux/kurs/material
```

Auf die neuen Verzeichnisse bekommt nur der Besitzer Rechte.

```
mkdir -m 700 tagebuch adressen
```

### 4.3.7 rmdir

Um leere Verzeichnisse wieder zu löschen, wird der Befehl `rmdir` verwendet.

`rmdir [OPTIONEN] [VERZEICHNISPFADLISTE]`

Dabei wird immer das letzte Verzeichnis in der angegebenen Verzeichnishierarchie gelöscht. Der Schalter `-p` ermöglicht das Löschen ganzer Hierarchien.

#### Optionen

`-p` Löschen aller leeren Verzeichnisse in der Verzeichnishierarchie.

#### Beispiele

Dieses Kommando löscht alle Verzeichnisse des Elternverzeichnisses `linux`.

```
rmdir tex/linux/*
```

Soll die ganze Hierarchie gelöscht werden, also auch die Verzeichnisse `tex/linux` und `tex` gelöscht werden, so lautet der Befehl:

```
rmdir -p tex/linux/*
```

Dabei dürfen die Verzeichnisse aber keine weiteren Verzeichnisse oder Dateien enthalten.

## 4.4 Der Linux-Verzeichnisbaum

Den Linux-Verzeichnisbaum gibt es eigentlich nicht. Jede Distribution hat eine individuellen Verzeichnishierarchie für sich entwickelt. Allerdings gibt es einige Verzeichnisse, die eigentlich überall vorhanden sein sollten. Als Leitlinie für den Baum gilt der sogenannte Dateisystemstandard *FSSTND* (Abkürzung für *File System STaNdard*). Dieser wird auch als *Filesystem Hierarchy Standard* (kurz *FHS*) bezeichnet. Eine Übersicht über die wichtigsten Verzeichnisse liefert Abbildung 4.1.

/ Dies ist Basis des Verzeichnisbaums und wird als *Wurzel* oder auch als *root* bezeichnet. Das Verzeichnis ist das Elternverzeichnis für alle anderen Verzeichnisse. Die folgenden Verzeichnisse müssen nach dem FSSTND als Hard- oder Softlink vorhanden sein: `/bin`, `/boot`, `/dev`, `/etc`, `/lib`, `/mnt`, `/opt`, `/sbin`, `/tmp`, `/usr` und `/var`. Die Verzeichnisse `/home` und `/root` sind optional, sollten aber vorhanden sein, wenn das passende Untersystem installiert ist.

`/bin` Dieses Verzeichnis enthält diverse wichtige Kommandos, die für den Systemadministrator und für normale Benutzer wichtig sind. Dieses Verzeichnis darf nach FSSTND keine Unterverzeichnisse enthalten und folgenden Programme sollten enthalten sein. `cat` (4.5.2), `chgrp` (9.2.2), `chmod` (9.2.3), `chown`

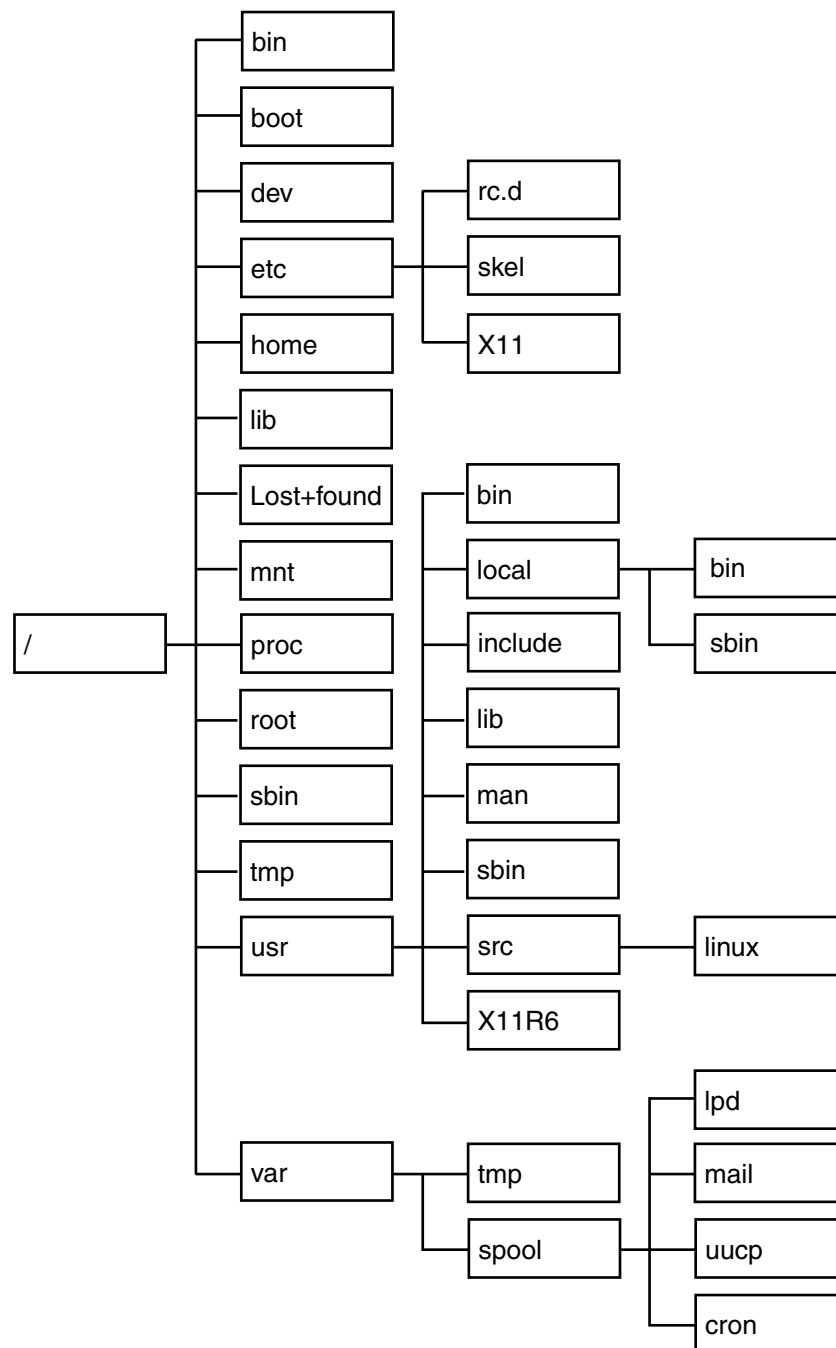


Abbildung 4.1: Der Linux-Dateibaum

(9.2.1), `cp` (4.5.3), `date` (4.6.3), `dd` (4.5.4), `df` (10.8.2), `dmesg` (12.1.3), `echo` (4.6.5), `false` (8.2.3), `hostname` (18.2.4), `kill` (12.9.7), `ln` (10.7.3), `login` (), `ls` (4.3.3), `mkdir` (4.3.6), `mknod` (), `more` (4.5.9), `mount` (10.3.1), `mv` (4.5.6), `ps` (12.8.1), `pwd` (4.3.1), `rm` (4.5.7), `rmdir` (4.3.7), `sed` (7.7.6), `sh` (8.6.3), `stty` (5.1.3), `su` (8.1.2), `sync` (10.3.4), `true` (8.2.3), `umount` (10.3.3) und `uname` (14.4.1).

**/boot** Enthält den Kernel und für das Booten wichtige Dateien.

**/dev** Das Verzeichnis für die Gerätedateien.

**/etc** Konfigurations- und Informationsdateien, darunter `magic`, `DIR_COLORS`, `input_rc` u. v. m.

**/etc/rc.d** Enthält die für den Bootvorgang wichtigen Skripte und die `rc`-Verzeichnisse.

**/etc/skel** Aus diesem Verzeichnis werden die Konfigurationsdateien (*skeleton user files*) für einen neuen Benutzer in dessen Heimatverzeichnis kopiert.

**/etc/X11** Die Konfigurationsdateien des X-Window-Systems liegen hier.

**/home** beherbergt die Heimatverzeichnisse, in denen normalerweise keine Rechte die Arbeit des Benutzers einschränken. Es ist somit für den normalen Benutzer von existentieller Bedeutung. In größeren LINUX-Systemen sollte für dieses Verzeichnis entsprechender Platzbedarf einkalkuliert werden, da hier die Benutzerdaten abgelegt werden. Häufig wird es dann auf einen Extra-Datenträger ausgelagert, der beim Systemstart separat gemountet wird.

**/lib** Hier findet man die Bibliothek des C-Compilers, sowie sogenannte Shared Libraries, Programmbibliotheken (vereinfacht gesagt Programmteile), die von verschiedenen Anwendungen benutzt werden können. Diese Programmteile werden erst zur Laufzeit in den Arbeitsspeicher geladen, wenn das betreffende Programm sie benötigt. Dadurch und durch die Tatsache, daß verschiedene Programme darauf zugreifen können (shared) wird der Arbeitsspeicher wesentlich entlastet. Sind Sie Windows-Benutzer kennen Sie sicherlich .DLL-Dateien. Diese werden unter Windows entsprechend genutzt.

**/Lost+found** Dieses Verzeichnis wird für wiederhergestellte Dateien verwendet.

**/mnt** Hilfsverzeichnis zur Aufnahme von Dateisystemen, die in den LINUX-Verzeichnisbaum gemountet werden sollen.

**/proc** ist ein virtuelles Dateisystem für Informationen über die laufenden Prozesse. Der Befehl `ps` (process status) liefert seine Angaben auf der Basis des Inhalts dieses Verzeichnisses. Jeder Prozeß erhält ein eigenes Unterverzeichnis in dem die Informationen über ihn gespeichert werden.

**/root** ist das Heimatverzeichnis der Superusers (**root**).

**/sbin** Hier liegen die Befehle zur Systemverwaltung wie `adduser`, mit dessen Hilfe neue Benutzer definiert werden, `fsck` zur Überprüfung von Dateisystemen oder `shutdown`, mit dem das System heruntergefahren werden kann. Programme für Systemstart wie `init`, mit dem diverse Systemdienste (z. B. Drucker- und Zeitdienste) initiiert werden, darunter auch `getty` (ebenfalls in **/sbin** zu finden). Über `getty` erfolgt das Anmelden an der Konsole oder an Terminals, die via serieller Leitung mit dem LINUX-Rechner verbunden sind.

**/tmp** Verzeichnis zum Zwischenlagern von Dateien. Hier darf jeder schreiben und lesen.

**/usr** Enthält alle wichtigen Programme und Daten wie etwa das Online-Manual, die nicht zum Booten des Systems benötigt werden. Es kann ein Verzeichnis sein, das sich physikalisch auf einem anderen Datenträger befindet als das Bootmedium, auf einer CD-ROM etwa oder an einem entfernten NFS-Server. In dieser Situation müßte dieser Teil des Dateisystems nach dem Booten erst verfügbar gemacht werden. Für den Fall, daß es dabei zu Problemen kommen sollte, stehen die allernötigsten Befehle in verschiedenen anderen Verzeichnissen, z. B. **/bin**, zur Verfügung.

**/usr/bin** enthält die Befehle, die allen Teilnehmern zur Verfügung stehen sollten. Ein Teil dieser Benutzer-Befehle befindet sich aus den oben genannten Gründen (vgl. Eintrag bei **/usr**) in **/bin**.

**/usr/local** wird für Programme verwendet, die nicht Bestandteil des Betriebssystems sind, wie z. B. das Dokumentenverzeichnis für den Webserver.

**/usr/local/bin** Hier werden die Binärdateien von diesen Programmen gespeichert. Dieses Verzeichnis sollte im Suchpfad des Benutzers enthalten sein.

**/usr/local/sbin** ist für lokal installierte Administrationstools zuständig.

**/usr/include** enthält die Standard-C/C++-Header-Dateien.

**/usr/lib** Die statischen Programmbibliotheken sowie die Unterverzeichnisse für Bibliotheken verschiedener Programmiersprachen sind hier zu finden. Enthält auch Links zu X-Window-Dateien.

**/usr/man** ist eins der Verzeichnisse für das Online-Manual. Eine unerschöpfliche Informationsquelle für den Benutzer.

**/usr/sbin** Analog zu **/usr/bin** gibt es dieses Verzeichnis für den Superuser (**root**).

**/usr/src** enthält die Quellcodes für die Systemprogramme.

**/usr/src/linux** In diesem Unterverzeichnis ist der Quellcode des Systemkerns zu finden.

**/usr/X11R6** Enthält Dateien für das X-Window-System. Oft auch nur ein Link auf ein anderes Verzeichnis.

**/var** Informationsdateien für das System sind hier abgelegt. Dies sind Tabellen, die von LINUX laufend verändert werden, z. B. die Dateien **utmp** und **wtmp**, die Informationen über die eingeloggten Benutzer enthalten. (Standard-UNIX: Die Tabellen befinden sich zum größten Teil in **/etc**.)

**/var/tmp** ist das Ziel der temporären Dateien, die von Anwendungen aus verschiedenen Gründen während eines Programmlaufs angelegt und verändert, zum Programmende aber wieder gelöscht werden. Editoren oder Textverarbeitungen speichern unter Umständen den vom Anwender geschriebenen Text zur Sicherheit in einer temporären Datei. Temporäre Dateien sind kurzlebig und ständiger Änderung unterworfen. Sie sind ein sprechendes Beispiel für die allgemeine Bedeutung des Verzeichnisses **/var**.

**/var/spool** ist der Ort der Warteschlangen (*Queues*) diverser Spoolprogramme. Dies sind Programme, die im Hintergrund arbeitend Aufträge der verschiedenen Benutzer entgegennehmen und in einer vernünftigen Reihenfolge abwickeln. Da mehrere Benutzer (beinahe) gleichzeitig z. B. einen Druckauftrag an den Druckspooler geben können, sprich drucken wollen, muß dieser dafür sorgen, daß die einzelnen Aufträge sich nicht vermischen und etwa am Drucker zwei Texte zu einem - bunt gemischten - werden. als Hilfsmittel dienen ihnen Warteschlangen, in die die einzelnen Aufträge nacheinander eingereiht und (nach dem Wirkungsprinzip einer FIFO) nach und nach abgearbeitet werden.

**/var/spool/lpd** Warteschlangen des Druckerspoolers

**/var/spool/mail** Warteschlange für die elektronische Post

**/var/spool/uucp** Warteschlange für den Kommunikationsdienst uucp (unix-to-unix-copy)

**/var/spool/cron** ist für die Abwicklung von Zeitdiensten über den cron-Dämonenprozeß wichtig, der in regelmäßigen Abständen Tabellen in **/var/spool/crontabs** daraufhin überprüft, ob sie Anweisungen enthalten, die zum gegenwärtigen Zeitpunkt auszuführen sind.

## 4.5 Arbeiten mit Dateien

### 4.5.1 touch

Das Kommando **touch** ändert die Zeit des letzten Zugriffs und der letzten Änderung auf die aktuelle Zeit. Existiert die Datei nicht, so wird eine neue Dateien erstellt.

**touch** [OPTIONEN] [DATEILISTE]

### 4.5.2 cat

Das Kommando `cat` (`concatenate file`) kann dazu benutzt werden neue Dateien zu erstellen. Es wird aber hauptsächlich dazu verwendet Dateien zusammenzufügen und sie auf dem Bildschirm oder einem anderen Gerät auszugeben.

```
cat [OPTIONEN] [DATEILISTE]
```

`cat` wird vor allem als Tool zum Anzeigen und Zusammenfügen von Dateien verwendet und als schneller Mini-Editor.

#### Optionen

<code>-b</code>	Nummeriert alle nichtleeren Zeilen durch
<code>-e</code>	Gleich mit <code>-vE</code>
<code>-n</code>	Nummeriert alle Zeilen durch ( <code>-b</code> hat Vorrang)
<code>-E</code>	Fügt am Ende jeder Zeile ein <code>\$</code> ein
<code>-s</code>	Faßt aufeinanderfolgende Leerzeilen zu einer Leerzeile zusammen
<code>-v</code>	Zeigt alle nichtdruckbaren Zeichen durch Metazeichen an
<code>-T</code>	Zeigt Tabulatoren als <code>␣</code> an
<code>-A</code>	Zeigt alle nichtdruckbaren Zeichen, Tabulatoren als <code>␣</code> und das Zeilenende als <code>\$</code> an (vgl. <code>-vET</code> )

#### Beispiel

Fügt die Dateien `t1` und `t2` zusammen und schreibt sie in die Datei `t3`.

```
cat t1 t2 > t3
```

Ein einfacher Editor kann mit `cat` realisiert werden.

```
cat > text.txt
```

Dieser Befehl liest die Daten nicht aus einer Datei sondern von der Standardeingabe und schreibt sie dann in die Datei `text.txt`. Die Eingabe wird mit `<CTRL>+<D>` (EOF-Zeichen) beendet.

### 4.5.3 cp

Um Kopien von einer Datei oder einem Verzeichnis zu erstellen wird das Kommando `cp` verwendet.

```
cp [OPTIONEN] QUELLDATEI ZIELDATEI
cp [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS
```

Wird ein Dateiname für die Quelle angegeben, so kopiert `cp` die Datei in eine zweite Datei (`ZIELDATEI`). Ist der letzte Parameter ein Verzeichnis, dann wird die Datei in das Verzeichnis kopiert. Listen von Quelldateien können nur in ein Verzeichnis kopiert werden. Allerdings kann `cp` in der Normaleinstellung keine Verzeichnisse kopieren.

#### Optionen

<code>-a</code>	Kopiert die Dateien unter Beibehaltung von Struktur und Attributen
<code>-b</code>	Erzeugt von jeder überschriebenen Datei eine Sicherheitskopie
<code>-f</code>	Überschreiben von vorhandenen Zieldateien
<code>-i</code>	Nachfragen vorm Überschreiben von vorhandenen Zieldateien
<code>-l</code>	Legt Hardlinks anstatt Kopien der Dateien an
<code>-s</code>	Legt Softlinks anstatt Kopien der Dateien an
<code>-p</code>	Überträgt Besitzer, Gruppe, Rechte und Zeitmarken an die neue Datei
<code>-P</code>	Kopieren der Dateien inklusiver ihrer Verzeichnisstruktur
<code>-r</code>	Kopiert rekursiv Dateien aus Verzeichnissen
<code>-R</code>	Wie <code>-r</code> , aber anstatt die Inhalte der Dateien zu kopieren, werden die Datei wie sie sind kopiert
<code>-u</code>	Kopiert nur Dateien, die jünger sind als die Zieldateien
<code>-v</code>	Zeigt die Namen der kopierten Dateien an

#### Beispiel

Die folgende Kommandosequenz kopiert alle Dateien des aktuellen Verzeichnis unter Beibehaltung ihrer Strukturen.

```
cp -dpRP * \backup
```

#### 4.5.4 dd

Das Programm *dd* (*Device to Device copy*) ist ein spezielles Kopierprogramm. Es wird in aller erster Linie dazu genutzt um Dateien von einem Gerät zu einem anderen zu kopieren.

```
dd [OPTIONEN]
```

##### Optionen

<b>if=DATEI</b>	Die Eingabedatei
<b>of=DATEI</b>	Die Ausgabedatei
<b>bs=BLOCKGRÖSSE</b>	Anzahl der Bytes, die auf einmal gelesen bzw. geschrieben werden.

##### Beispiel

Dieser Befehl legt eine Kopie einer Diskette in der Datei BackupDisk an.

```
dd if=/dev/fd0 bs=512 of=BackupDisk
```

#### 4.5.5 Erstellung der Bootdisketten mit dd

Unter Linux können sie ebenfalls, wie unter DOS (Abschnitt 1.4.1) die Bootdisketten erstellen. Hierzu werden die gleichen Images im Verzeichnis `/disks` der CD 1 verwendet. Die Disketten müssen low-level-formatiert sein. Dazu kann der Befehl `fdformat` (Abschnitt 10.4.2) verwendet werden.

Nachdem die CD auf das Verzeichnis `/cdrom` gemountet worden ist, können mit den Befehlen

```
dd if=/cdrom/disks/bootdisk of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules1 of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules2 of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules3 of=/dev/fd0 bs=8k
```

die Bootdisketten erzeugt werden.

#### 4.5.6 mv

Um Dateien zu verschieben oder umzubenennen wird das Kommando `mv` verwendet.

```
mv [OPTIONEN] ALTERDATEINAME NEUERDATEINAME
mv [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS
```

Werden zwei Dateinamen als Parameter vergeben, dann wird die Datei umbenannt. Ist der letzte Parameter ein Verzeichnis, dann werden die angegebenen Dateien in dieses Verzeichnis verschoben.

##### Optionen

<b>-b</b>	Erzeugt von jeder überschriebene Datei eine Sicherheitskopie
<b>-f</b>	Überschreiben von vorhandenen Zieldateien
<b>-i</b>	Nachfragen vorm Überschreiben von vorhandenen Zieldateien
<b>-u</b>	Verschiebt nur Dateien, die jünger sind als die Zieldateien

##### Beispiel

Die folgende Kommandosequenz benennt die Datei `megabox.txt` in die Datei `MEGABOX.txt` um.

```
mv megabox.txt MEGABOX.txt
```

Mit dem folgenden Kommando werden alle Dateien aus dem aktuellen Verzeichnis in das Verzeichnis `/home/harald` verschoben und eventuell vorhandene Zieldateien überschrieben.

```
mv -f * /home/harald
```

### 4.5.7 rm

Der Befehl **rm** löscht die angegebenen Dateien. In seiner Standardeinstellung löscht er keine Verzeichnisse.

**rm** [OPTIONEN] DATEILISTE

Um eine Datei zu löschen, benötigt man das Schreibrecht (**w**) auf das Verzeichnis, aber nicht auf die Datei. Sollte ein Datei nicht das Schreibrecht besitzen, so erbittet **rm** nur eine Bestätigung des Löschbefehls, wenn kein **-f** oder wenn ein **-i** gesetzt ist.

#### Optionen

<b>-d</b>	Löscht ein Verzeichnis durch Entfernen des Hardlinks. Volle Verzeichnisse werden auch gelöscht. Da die enthaltenen Dateien nicht mehr referenziert werden, ist es ratsam ein <b>fsck</b> danach auszuführen. (Nur Superuser)
<b>-f</b>	Löscht alle Dateien ohne explizites Nachfragen. Überlagert die Option <b>-i</b>
<b>-i</b>	Fragt vor jedem Löschen einer Datei um Bestätigung
<b>-r</b>	Löscht Verzeichnisse und deren Inhalt rekursiv

#### Beispiel

Der folgende Befehl löscht alle Dateien, die mit **.temp** enden.

```
rm *.temp
```

Vorsicht! Dieser Befehl versucht die gesamte Dateistruktur ohne Nachfragen zu löschen. Als **root** ausgeführt ist das Ergebnis fatal.

```
rm -rf /
```

### 4.5.8 Dateibezeichnung mit Jokerzeichen

Bei der Arbeit mit der Shell kommen Sie oft in Situationen, in denen Sie nicht nur einzelne Dateien oder alle Dateien in einem Verzeichnis ansprechen wollen, sondern nur eine Gruppe von Dateien (z. B. alle HTML-Dateien) bearbeiten möchten. Für diesen Zweck besitzt die Shell die sogenannten Jokerzeichen (*Wildcards*).

Wenn diese Jokerzeichen eingesetzt werden, gibt die Shell nicht einen einzelnen Dateinamen zurück, sondern eine Liste von Dateinamen, auf die das Muster zutrifft. So erhalten Sie mit dem Muster **\*.jpg** eine Liste aller Dateinamen, die mit **.jpg** enden.

```
tapico@defiant:~> echo *.jpg
helm35.jpg helm37.jpg helm38a.jpg helm59.jpg
tapico@defiant:~> cp *.jpg ~/bilder
```

Denken Sie daran. Die Shell und nicht der Befehl interpretiert das Kommando. Deshalb müssen die Jokerzeichen in manchen Fällen maskiert werden, damit sie unbeschadet an den Befehl übergeben werden können. Probieren Sie den unteren Befehl mal ohne Backslash aus.

```
tapico@defiant:~> echo \* Hallo \*
* Hallo *
```

Eine Reihe von Jokerzeichen steht Ihnen zur Verfügung.

#### Asterisk \*

Der Asterisk bedeutet im Prinzip alles oder nichts. Er steht für eine beliebige Anzahl beliebiger Zeichen. Wobei beliebige Anzahl auch kein Zeichen bedeuten kann. Z. B. stimmt das Muster **lk\*** mit folgenden Dateinamen überein.

```
tapico@defiant:~> ls lk*
lk-aufbau-ext2.eps      lk-shell.aux           lk.log
```



lk-dateibaum.eps	lk-shell.tex	lk.pdf.gz
lk-dateisystem.aux	lk-shell.tex.bck	lk.ps
lk-dateisystem.tex	lk-vorwort.aux	lk.tex
lk-dateisystem.tex.bck	lk-vorwort.tex	lk.tex.bck
lk-installation.aux	lk-vorwort.tex.bck	lk.toc
lk-installation.tex	lk.aux	lktex
lk-installation.tex.bck	lk.dvi	

### Fragezeichen ?

Das Fragezeichen steht für *genau ein* beliebiges Zeichen.

```
tapico@defiant:~> ls lk.*
lk.aux lk.idx lk.ind lk.pdf.gz lk.tex lk.toc
lk.dvi lk.ilg lk.log lk.ps lk.tex.bck
tapico@defiant:~> ls lk.???
lk.aux lk.dvi lk.idx lk.ilg lk.ind lk.log lk.tex lk.toc
tapico@defiant:~> ls lk.??
lk.ps
```

### Menge [ZEICHEN]

Mit den eckigen Klammern ist es möglich eine Menge von Zeichen vorzugeben, die an dieser Stelle stehen können. So steht [Aa] für ein Zeichen, daß entweder ein großes oder ein kleines A sein kann. Ein Ausrufezeichen in der eckigen Klammer negiert die Bedeutung der Menge. So steht [!aeiou] für ein Zeichen, daß kein Vokal ist.

Innerhalb der Klammern können nicht nur einzelne Zeichen angegeben werden, sondern auch Bereiche. So steht z. B. [a-z] für alle Kleinbuchstaben und [0-9] für alle Zahlen.

```
tapico@defiant:~> ls lk.???
lk.aux lk.dvi lk.idx lk.ilg lk.ind lk.log lk.tex lk.toc
tapico@defiant:~> ls lk.[it]??
lk.idx lk.ilg lk.ind lk.tex lk.toc
tapico@defiant:~> ls lk.[!it]??
lk.aux lk.dvi lk.log
tapico@defiant:~> ls [Hh]*
Home.txt Hurra.jpg hans.txt himmel.jpg
tapico@defiant:~> ls [A-Z]*
Amerika.jpg Home.txt Hurra.jpg Qualle.jpg
tapico@defiant:~> ls *[0-9][0-9].jpg
helm35.jpg helm37.jpg helm38.jpg helm59.jpg
```

### Klammerexpansion {WORT1,WORT2,WORT3,...}

Die Klammerexpansion (*brace expansion*) ist im eigentlichen Sinne kein Jokerzeichen. Sie funktioniert nämlich auch ohne existierende Dateien. Die Funktion ist einfach. Der Ausdruck wird jeweils mit dem in den geschweiften Klammern angegebenen und durch Kommata getrennten Zeichenketten ausgegeben.

```
tapico@defiant:~> echo "Ich bin "{gut,besser,super}""
Ich bin gut. Ich bin besser. Ich bin super.
tapico@defiant:~> echo {1,2,3}{1,2,3}
11 12 13 21 22 23 31 32 33
```

Trotzdem können Sie die Klammerexpansion natürlich auch auf Dateioperationen anwenden.

```
tapico@defiant:~> ls lk-{kernel,shell}.*
```

```
lk-kernel.aux  lk-kernel.tex.bck  lk-shell.tex
lk-kernel.tex  lk-shell.aux      lk-shell.tex.bck
tapico@defiant:~> ls lk-*.tex,eps
lk-administration.tex  lk-dateibaum.eps    lk-installation.tex  lk-shell.tex
lk-aufbau-ext2.eps     lk-dateisystem.tex  lk-kernel.tex        lk-vorwort.tex
```

### 4.5.9 more

Der Befehl `more` gehört zur Gruppe der Befehle, mit denen man sich den Inhalt von Dateien anschauen kann (*pager*). Im Gegensatz zu `cat` (4.5.2) zeigt er die Daten aber seitenweise an.

`more [OPTIONEN] DATEILISTE`

#### Befehle

<LEERTASTE>	Eine Seite nach unten scrollen
<f>	Eine Seite nach unten scrollen
<b>	Eine Seite nach oben scrollen
<RETURN>	Eine Zeile nach unten scrollen
<s>	Eine Zeile nach unten scrollen
</>MUSTER	Durchsuchen des Textes nach dem regulären Ausdruck MUSTER
<n>	Weitersuchen nach unten
<q>	Beenden

### 4.5.10 Der Pager less

Der Befehl `less` ist die Weiterentwicklung von `more`<sup>5</sup>.

`less [OPTIONEN] DATEILISTE`

Neben den Möglichkeiten von `more` bietet `less` die Möglichkeit mit den Richtungstasten zu scrollen und mit Lesezeichen, Zeilennummern und prozentualen Textpositionen zu arbeiten. Auch wird `less` nicht am Ende der Datei beendet.

#### Zusätzliche Befehle

<j>	Eine Zeile nach unten scrollen
<k>	Eine Zeile nach oben scrollen
<d>	Einen halben Bildschirm nach unten scrollen
<u>	Einen halben Bildschirm nach oben scrollen
<f>	Einen Bildschirm nach unten scrollen
<b>	Einen Bildschirm nach oben scrollen
<g>	An den Anfang der Datei scrollen
<G>	Ans Ende der Datei scrollen
</> MUSTER	Durchsuchen des Textes nach dem regulären Ausdruck <MUSTER> nach unten
<?> MUSTER	Durchsuchen des Textes nach dem regulären Ausdruck <MUSTER> nach oben
<n>	Weitersuchen nach unten
<N>	Weitersuchen nach oben
<q>	Programm beenden
<:;><n>	Nächste Datei in der Liste anzeigen
<:;><p>	Vorherige Datei in der Liste anzeigen

Wichtig ist beim Befehl `less` die Suche innerhalb eines Textes. Dazu stehen die Kommandos `</>` und `<?>` zur Verfügung. Nach Betätigen der jeweiligen Tasten erscheint in der untersten Zeile eine Eingabeaufforderung, wo Sie Ihr Suchmuster eingeben können. Den Zugriff auf alte Suchmuster erhalten Sie durch das Betätigen der Cursortaste für "hoch" und "runter".

Das Programm `less` besitzt viele Schalter und Kommandos. Eine Übersicht über die Möglichkeiten von `less` erhalten Sie durch die Manualpage `less(1)`.

<sup>5</sup>Und weil er mehr kann als `more` wird er `less` genannt ;-)

## Das Konfigurationsprogramm `lesskey`

Das Programm `lesskey` erlaubt eine individuelle Konfiguration der Steuerbefehle von `less`.

### 4.5.11 `file`

Der Befehl `file` analysiert Dateien und gibt ihren Typ aus.

`file DATEILISTE`

`file` prüft zunächst anhand der Inode-Informationen, um welchen Datentyp es sich handelt, prüft aber zusätzlich den Anfang der Datei anhand der Informationen, die in der Datei `/etc/magic` enthalten sind.

#### Optionen

<code>-b</code>	Ausgabe ohne Angabe der Dateinamen
<code>-f DATEINAME</code>	Die Dateiliste wird aus der Datei <code>DATEINAME</code> entnommen (Ein Dateiname pro Zeile).
<code>-z</code>	Bearbeitet auch den Inhalt von gepackten Dateien

#### Beispiel

```
ole@defiant:~ > ls -Fl
total 5
-rw-r--r--  1 ole      users      468 May  9 22:28 194.195.155.105
lrwxrwxrwx  1 ole      users         9 May  9 22:00 hallo -> hallowelt*
-rwxr-xr-x  1 ole      users      36 May  9 21:56 hallowelt*
-rw-r--r--  2 ole      users     401 May  9 22:04 liste
-rw-r--r--  2 ole      users     401 May  9 22:04 myliste
drwxr-xr-x  2 ole      users    1024 May  9 21:55 mytest/
prw-r--r--  1 ole      users       0 May  9 21:54 testpipe|
ole@defiant:~ > file *
194.195.155.105: ASCII text
hallo:           symbolic link to hallowelt
hallowelt:       perl commands text
liste:           ASCII text
myliste:         ASCII text
mytest:          directory
testpipe:        fifo (named pipe)
```

### 4.5.12 `/etc/magic`

Die Datei `/etc/magic` ist eine Liste von sogenannten "Magic Numbers". Dies Magic Numbers sind die Erkennungsmelodien verschiedener Dateiformate, die meistens am Anfang einer Datei stehen.

Die Datei wird von verschiedenen Programmen gebraucht, wie z. B. `file` (4.5.11).

Der Eintrag z. B. für eine Bilddatei im PNG-Format ist:

```
# PNG [Portable Network Graphics, or "PNG's Not GIF"] images
# (Greg Roelofs, newt@uchicago.edu)
# (Albert Cahalan, acahalan@cs.uml.edu)
#
# 137 P N G \r \n ~Z \n [4-byte length] H E A D [HEAD data] [HEAD crc] ...
#
0      string      \x89PNG          PNG image data,
>4     belong      !0x0d0a1a0a      CORRUPTED,
>4     belong      0x0d0a1a0a
>>16   belong      x                %ld x
>>20   belong      x                %ld,
```

```
>>24    byte          x          %d-bit
>>25    byte          0          grayscale,
>>25    byte          2          \b/color RGB,
>>25    byte          3          colormap,
>>25    byte          4          gray+alpha,
>>25    byte          6          \b/color RGBA,
#>>26   byte          0          deflate/32K,
>>28    byte          0          non-interlaced
>>28    byte          1          interlaced
1        string        PNG        PNG image data, CORRUPTED
```

Die Datei kann sich auch im Verzeichnis `/usr/share/misc/magic` befinden. Dann ist `/etc/magic` nur ein Link auf diese Datei. Weitere Informationen können Sie den Manualpages (`man 4 magic`) entnehmen.

## 4.6 Weitere Befehle

### 4.6.1 clear

Dieser Befehl löscht den Bildschirm.

```
clear
```

### 4.6.2 cal

Das Kommando `cal` zeigt einen Kalender an.

```
cal [[MONAT] JAHR]
```

Für `JAHR` sind Werte zwischen 1 und 9999 erlaubt und für `MONAT` zwischen 1 und 12.

#### Optionen

```
-m          Zeigt Montag als ersten Tag der Woche an
-j          Benutzt das Julianische Datum
-y          Zeigt einen Kalender für das aktuelle Jahr
```

### 4.6.3 date

Diese Befehl zeigt und setzt das aktuelle Datum und die Uhrzeit. Allerdings kann nur `root` die Zeit setzen.

#### Zeit anzeigen

Die Anzeige der aktuellen Zeit erfolgt über

```
date [+FORMATSTRING]
```

`FORMATSTRING` ist ein beliebiger Text, der, falls er Sonderzeichen enthält, in Anführungszeichen zu setzen ist. Sinn macht dieser Text allerdings erst, wenn er Platzhalter enthält, die Datum oder Zeit anzeigen. Eine Liste der Platzhalter finden Sie in Tabelle 4.4

Durch den Formatstring kann das Ausgabeformat bestimmt werden.

```
tapico@defiant:~> date +"Es ist der %d.%m.%y"
Es ist der 19.01.03
tapico@defiant:~> date +"Es ist %T Uhr"
Es ist 13:47:35 Uhr
```

%a	Abgekürzter Wochentagname nach lokaler Spracheinstellung
%A	Voller Wochentagname nach lokaler Spracheinstellung
%b	Abgekürzter Monatsname nach lokaler Spracheinstellung
%B	Voller Monatsname nach lokaler Spracheinstellung
%c	Datum und Zeit nach lokaler Spracheinstellung
%C	Das Jahrhundert (00..99)
%d	Tag im Monat (01..31)
%D	Datum im Format mm/dd/yy
%e	Tag des Monats mit führendem Leerzeichen ( 1..31)
%F	Datum im Format %Y-%m-%d
%g	Zweistellige Jahresangabe im Zusammenhang mit der Angabe %V
%G	Vierstellige Jahresangabe im Zusammenhang mit der Angabe %V
%H	Stunde (00..23)
%I	Stunde (01..12)
%j	Tag des Jahres
%m	Monat (01..12)
%M	Minuten (00..59)
%n	Zeilenumbruch
%N	Nanosekunden (000000000..999999999)
%p	AM/PM nach lokaler Spracheinstellung
%P	am/pm nach lokaler Spracheinstellung
%r	Uhrzeit im 12-Stunden-Format (hh:mm:ss am/pm)
%R	Uhrzeit im 24-Stunden-Format (hh:mm)
%s	Sekunden seit 01.01.1970 00:00:00 Uhr UTC
%S	Sekunden (00..60); Die 60 ist für Schaltsekunden notwendig
%t	Tabulator
%T	Zeit (hh:mm:ss)
%u	Tag der Woche (1..7) 1=Montag
%U	Kalenderwoche für Wochenanfang Sonntag (00..53)
%V	Kalenderwoche für Wochenanfang Montag (01..53)
%w	Tag der Woche (0..6) 0=Sonntag
%W	Kalenderwoche für Wochenanfang Montag (01..53)
%x	Datum nach lokaler Spracheinstellung
%X	Zeit nach lokaler Spracheinstellung
%y	Jahr (zweistellig)
%Y	Jahr (vierstellig)
%z	Zeitzone nach RFC-822 in numerischer Darstellung
%Z	Zeitzone

Tabelle 4.4: Platzhalter beim Befehl `date`

Einige der Platzhalter orientieren sich bei der Ausgabe an den lokalen Spracheinstellungen des Systems. Diese ist für eine laufende Sitzung in der Umgebungsvariablen `LANG` gespeichert. Eine Veränderung dieser Variablen führt daher auch zu einer veränderten Ausgaben des Befehls `date`. Meist ist beim Benutzer `root` die Variable nicht gesetzt. Default ist dann Englisch.

```
ole@enterprise:~> echo $LANG
de_DE@euro
ole@enterprise:~> date +%x
14.05.2004
ole@enterprise:~> date +%c
Fr 14 Mai 2004 10:53:42 CEST
ole@enterprise:~> date +"Ein %A im %B."
Ein Freitag im Mai.
ole@enterprise:~> LANG=en; export LANG
ole@enterprise:~> date +%x
05/14/04
ole@enterprise:~> date +%c
Fri May 14 10:54:42 2004
ole@enterprise:~> date +"Ein %A im %B."
Ein Friday im May.
```

## Zeit stellen

Der Systemverwalter kann auch Zeit und Datum über diesen Befehl ändern.

```
date MMTTSSmm[[HH]JJ][.ss]
```

Wobei `MM` für Monat, `TT` für Tag des Monats, `SS` für Stunde, `mm` für Minute, `HH` fürs Jahrhundert, `JJ` fürs Jahr und `ss` für die Sekunden stehen.

Die Änderung der Systemzeit hat keine Auswirkungen auf die Hardware-Uhr. Nach dem Neustart des Systems sind die Einstellungen wieder verloren.

Um die Zeit zu setzen brauchen Sie die Berechtigung des Systemadministrators. Wenn Sie als normaler Benutzer eingeloggt sind, wechseln Sie zuerst mit dem Befehl `su` (8.1.2) ihre Identität zu `root`. Danach können Sie das Datum einstellen.

```
tapico@defiant:~> su
Password:
root@defiant:/home/tapico # date 10221735
Mon Okt 22 17:35:00 CEST 2001
```

Für weitere Informationen zu Schaltern und Platzhaltern geben Sie `man date` ein.

## 4.6.4 hwclock

Der Befehl `date` beschäftigt sich nur mit der Systemzeit. Zum Auslesen und Setzen der Hardware-Uhr können Sie den Befehl `hwclock` verwenden. Dieser Befehl steht nur dem Systemadministrator zur Verfügung.

```
hwclock [OPTIONEN]
```

Für die Anwendung und Möglichkeiten dieses Befehls konsultieren Sie bitte die Manualpage (`man 8 hwclock`).

Um neben der Systemzeit auch die Hardware-Uhr zu setzen, gehen Sie bitte wie folgt vor.

1. Wechseln Sie ihre Identität zu `root`.

2. Setzen Sie mit dem Befehl `date` (4.6.3) die Systemzeit.
3. Setzen Sie nun mit den Befehl `hwclock` die Systemzeit und Hardware-Uhr gleich.

Das kann dann so aussehen:

```
ole@defiant:~> su
Password:
defiant:/home/ole # date 01191418
Son Jan 19 14:18:00 CET 2003
defiant:/home/ole # hwclock --systohc
```

#### 4.6.5 echo

Der Befehl `echo` wirkt wie ein Echo, indem er einfach den angegebenen Text wieder auf den Bildschirm ausgibt. Seine Haupteinsatz findet er daher in den Shellskripten, der Ausgabe von Texten in Dateien und dem Anzeigen der Inhalte von Variablen.

`echo [OPTIONEN] [TEXT]`

##### Optionen

<code>-e</code>	erlaubt die Benutzung von Sonderzeichen im Text, z. B:
	<code>\n</code> Zeilenumbruch
	<code>\a</code> Piepton (ASCII-Zeichen 7)
	<code>\b</code> Backspace
	<code>\f</code> Seitenvorschub
	<code>\ooo</code> beliebiges ASCII-Zeichen in oktaler Notation
<code>-n</code>	unterdrückt Zeilenumbruch nach Ausgabe des Textes

Texte mit Sonderzeichen (z. B. Leerzeichen) sollten in Anführungszeichen stehen. `echo` ohne Angabe eines Textes bewirkt die Ausgabe einer Leerzeile.

#### 4.6.6 Farbcode fürs Terminal

Um Terminals zu konfigurieren werden Codesequenzen benutzt. Sie werden als Escape-Sequenzen bezeichnet, da sie mit dem Escape-Zeichen (0x1B)<sup>6</sup> beginnen. So kann die Farbe der Ausgabe festgelegt werden. Mit der Sequenz `<ESC>[{attr};{fg};{bg}m` kann nun die Farbe vorgegeben werden.

So bewirkt der Befehl

```
echo "~[[0;31;40mIch sehe rot"
```

daß der Text 'Ich sehe rot' in roter Farbe ausgegeben wird. Das erste Zeichen ist das Escape-Zeichen. Es wird als `~[` dargestellt. Um es zu schreiben drücken Sie `<STRG>+<V>` und dann `<ESC>`. Um wieder den Normalzustand herzustellen, müssen Sie für die Konsole den Befehl

```
echo "~[[0;37;40m"
```

eingeben.

Eine Liste der möglichen Farben und Attribute liefert Tabelle 4.5. Diese Werte werden auch in den Dateien `/etc/DIR_COLORS` und `.dir_colors` für die Farbcodierung verwendet. Eine vollständige Liste der Konsolencodes liefert die Manpage `console_codes` (4).

```
ole@enterprise:~> man 4 console_codes
```

#### 4.6.7 logout

Dieses `bash`-Kommando beendet die Sitzung eines Benutzers.

`logout`

---

<sup>6</sup>Zahlen, die mit `0x` beginnen, sind in hexadezimaler Schreibweise dargestellt. `0x1B` ist somit die Zahl 27 (16 + 11).

{attr}		{fg}		{bg}	
0	Normaler Modus	30	Schwarz	40	Schwarz
1	Hell oder Fett	31	Rot	41	Rot
2	Dunkel	32	Grün	42	Grün
3	Unterstrichen	33	Gelb	43	Gelb
5	Blinkend	34	Blau	44	Blau
7	Invertiert	35	Margenta	45	Margenta
8	Versteckt	36	Türkis	46	Türkis
		37	Weiß	47	Grau
				48	Weiß

Tabelle 4.5: Farbcodes fürs Terminal

Die Möglichkeit der Darstellung nach den oben gezeigten Farbcodes hängt stark vom darstellenden Terminal ab. Gerade Attribute wie Unterstrichen und Blinkend funktionieren meistens nicht.

#### 4.6.8 script

Der Befehl `script` sorgt dafür, daß alle Ausgaben auf den Bildschirm gleichzeitig in einer Datei mitgeloggt werden. Dabei wird alles mitgeschrieben bis der Benutzer sich wieder ausloggt. Dieses Kommando eignet sich sehr gut zum Protokollieren von Aktionen.

```
script [OPTIONEN] [DATEI]
```

Wird kein Dateiname angegeben, so wird automatisch die Datei `typescript` angelegt.

#### 4.6.9 telnet

Natürlich gibt es auch eine Möglichkeit eine Linux-Rechner als Terminal zu verwenden. Das Programm `telnet` ermöglicht den Zugang zu anderen Rechnern. Im Gegensatz zu Windows sehen Sie im Textterminal keine Unterschied ob Sie direkt an der Konsole oder über `telnet` arbeiten.

```
telnet HOST [PORT]
```

Achtung! Aus Sicherheitsgründen können Sie sich nicht als *root* über Telnet an einem Rechner anmelden. Sie müssen sich erst als normaler Benutzer einloggen und können dann einen Identitätswechsel zum Superuser vollziehen.

Weitere Informationen über das Programm `telnet` finden Sie in den Manual-Pages über den Befehl `man telnet` und im Abschnitt 18.3.4.

#### 4.6.10 tty

Die Ausgabe eines Befehls landet immer auf der Konsole oder dem Terminal, auf dem es gestartet wurde. Aber auf welcher Konsole bin ich denn überhaupt? Da hilft der Befehl `tty` weiter.

```
tty
```

##### Beispiel

```
tapico@defiant:~ > tty
/dev/pts/0
```



## IV

Arbeitsblatt  
Shell 1

## 4.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

- 1 Loggen Sie als Benutzer `walter` auf Konsole 1 ein!
- 2 Führen Sie den Befehl `rm -rf *` aus. Was bewirkt der Befehl?
- 3 Legen Sie eine leere gewöhnliche Datei mit dem Namen `ogottogottogott` an.
- 4 Wechseln Sie ins Elternverzeichnis (Nach `..` also)
- 5 In welchem Verzeichnis sind Sie jetzt?
- 6 Stellen Sie fest, welche Dateien hier sind!
- 7 Löschen Sie den Bildschirm. (Nicht mit Wasser!)
- 8 Wechseln Sie in das Verzeichnis mit Ihrem Benutzernamen!
- 9 Lassen Sie sich den Text "Moin, Moin" begleitet von einem Piepton auf dem Bildschirm ausgeben.
- 10 Welche Dateien gibt es im aktuellen Verzeichnis?
- 11 Wechseln Sie zur Dateisystemwurzel! Welche Dateien gibt es hier?
- 12 Wechseln Sie in das Verzeichnis `/boot`!
- 13 Welche Dateien finden Sie hier?
- 14 Wechseln Sie in das Verzeichnis `/dev`?
- 15 Prüfen Sie den Inhalt des Verzeichnisses! Kommen Ihnen einige Namen bekannt vor?
- 16 Wechseln Sie ins übergeordnete Verzeichnis (nach `..` also).
- 17 Wechseln Sie ins Verzeichnis `/bin` !
- 18 Stellen Sie fest, welche Dateien mit dem Anfangsbuchstaben `l` (wie Ludwig) vorhanden sind.
- 19 Lassen Sie sich ausführliche Informationen über die Dateien im Verzeichnis `/etc` geben!
- 20 Stellen Sie fest, in welchem Verzeichnis Sie sind! Wechseln Sie, falls notwendig, ins Verzeichnis `/bin` !
- 21 Legen Sie ein Verzeichnis mit Ihrem Benutzernamen in `/bin` an!
- 22 Wechseln Sie in Ihr Heimatverzeichnis!
- 23 Legen Sie im Heimatverzeichnis die Verzeichnisse `test`, `org`, `org/hh` und `org/ki` an!
- 24 Stellen Sie fest, wem die Verzeichnisse `test` und `org` in Ihrem Heimatverzeichnis gehören!
- 25 Erstellen Sie (mittels `echo text > datei`) eine Datei namens `keks` in ihrem Heimatverzeichnis (Inhalt beliebig)!
- 26 Wechseln Sie nach `org/hh` !
- 27 Stellen Sie fest, in welchem Verzeichnis Sie sind!
- 28 Legen Sie (wie in Aufgabe 25) eine Datei `bonbon` im aktuellen Verzeichnis an!
- 29 Lassen Sie sich den Inhalt des aktuellen Verzeichnisses anzeigen!

## IV

Arbeitsblatt  
Shell 1

## 4.2

- 30 Lassen Sie sich den Inhalt Ihres Heimatverzeichnisses anzeigen!
- 31 Wechseln Sie ins Heimatverzeichnis!
- 32 Löschen Sie die Verzeichnisse `org` und `test` mit dem Befehl `rmdir`!
- 33 Stellen Sie fest, welche Dateien in `/usr/bin` sind!
- 34 Lassen Sie sich alle Dateien/Verzeichnisse des Heimatverzeichnisses anzeigen außer den der Standard-einträge `.` und `..` !
- 35 Lassen Sie sich die Inode-Nummern der Dateien des Verzeichnisses `/usr/bin` anzeigen!
- 36 Wann wurde zuletzt auf die Datei `/usr/share/doc/packages/shadow/HOWTO` zugegriffen?
- 37 Wie groß (Byte) ist die Datei `/etc/passwd` ?
- 38 Wem gehört die Datei `/boot` ?
- 39 Wieviele Links gibt es für Ihr Heimatverzeichnis?
- 40 Lassen Sie sich die gesamte Verzeichnisstruktur (Unterverzeichnisse) von `/usr` anzeigen!
- 41 Welche Dateien in dem Verzeichnis `/boot` beginnen mit `?`?
- 42 Gibt es in `/dev` ausführbare Dateien?
- 43 Vergleichen Sie die Inode-Nummern der Dateien `.` und `..` in der Wurzel!
- 44 Welche Inode-Nummern haben die Dateien `mv`, `cp` und `ln` in `/bin` ?
- 45 Lassen Sie sich ausführliche Informationen über das aktuelle Verzeichnis anzeigen, nicht aber über die darin enthaltenen Dateien!
- 46 Kopieren Sie alle Dateien aus `/usr/share/doc/packages/shadow` ins aktuelle Verzeichnis!
- 47 Prüfen Sie die Zeit der letzten Änderung von `HOWTO` im aktuellen Verzeichnis und in `/usr/share/doc/packages/shadow`!
- 48 Erstellen Sie die Verzeichnisse `PRIMA`, `PrimA` und `prima` im aktuellen Verzeichnis!
- 49 Verschieben Sie `HOWTO` aus dem aktuellen Verzeichnis nach `prima` !
- 50 Kopieren Sie die Dateien `README` und `ogottogottogott` aus dem aktuellen Verzeichnis in einem Arbeitsgang nach `prima` !
- 51 Wechseln Sie nach `prima` !
- 52 Benennen Sie `ogottogottogott` um in `meingottwalter` !
- 53 Benennen Sie `README` um in `.juhe` !
- 54 Kopieren Sie alle Dateien des aktuellen Verzeichnisses in einem Zug ins Heimatverzeichnis! Wechseln Sie danach ins Heimatverzeichnis !
- 55 Wann wurde die Datei `ogottogottogott` zum letzten Mal geändert?
- 56 Setzen Sie die Zeit des letzten Zugriffs und der letzten Änderung auf die aktuelle Zeit.
- 57 Löschen Sie das komplette Verzeichnis `prima`, unterdrücken Sie dabei Rückfragen!
- 58 Lassen Sie sich den Text "Fertig" in roter blinkender Schrift auf dem Bildschirm ausgeben. Achten Sie dabei darauf, daß der Prompt wieder dem ursprünglichem Schema entspricht.
- 59 Löschen Sie den Bildschirm.

# Kapitel 5

## Die Shell II

### 5.1 Die gewissen Extras

Um dem Benutzer die Arbeit mit dem Rechner zu erleichtern, besitzt die Shell einige sehr nützliche Funktionen. Das fängt mit dem Editor für die Eingabe der Befehle an, umfaßt Hilfen zur Eingabe langer Dateinamen und vieles mehr.

#### 5.1.1 Readline Library

Wenn Sie auf der Eingabezeile arbeiten, benutzen Sie den Editor Readline Library der Free Software Foundation. Standardmäßig arbeitet er im `emacs`-Modus. D. h. er verwendet zur Steuerung die gleichen Befehle wie der Editor `emacs`.

Eine Übersicht über die Steuerungsbefehle finden Sie in Tabelle 5.1.

Anstatt der Readline Library kann man auch einen anderen Editor für die Eingabezeile benutzen. Um z. B. den `vi` zu benutzen, geben Sie ein:

```
set -o vi
```

Zurück kommen Sie mit

```
set -o emacs
```

#### 5.1.2 inputrc

Die Konfigurationsinformationen für die Eingabezeile befinden sich in der Datei `/etc/inputrc`. Um diese Einstellungen individuell zu ersetzen, kann im Heimatverzeichnis die Datei `.inputrc` erstellt werden. Die Informationen dieser Konfigurationsdatei werden nur beim Einloggen ausgelesen.

So kann z. B. die Tastaturbelegung selbst definiert werden. Die Codesequenzen, die der Shell für die Tasten übermittelt werden, können von der Eingabeoberfläche abhängen. So steht die Sequenz `\e[11~` für die Taste `<F1>` bei einem X-Window-Terminal (`xterm`, `kvt`) oder einem Telnet-Terminal. Dagegen übermittelt die normale Linuxkonsole (`linux`) die Codesequenz `\e[[A` für das Drücken der Taste `<F1>`.

Um die Taste `<F1>` mit dem Befehl `ls -l` zu belegen, muß folgender Eintrag in die `.inputrc` gemacht werden.

```
"\e[11~": "ls -l"
```

oder

```
"\e[[A": "ls -l"
```

Jedesmal, wenn die Taste `<F1>` gedrückt wird, erscheint nun `ls -l` am Prompt. Soll der Befehl dann auch noch sofort ausgeführt werden, dann muß nur noch ein `\n` hinten angehängt werden.

```
"\e[11~": "ls -l\n"
```

Soll eine andere Datei für die Konfiguration verwendet werden, so muß diese mit ihrem Pfad in der Umge-



```
dozent@linux37:~> ls -l /dev/tty[12]
crw--w----    1 dozent    tty          4,    1 2004-07-01 14:25 /dev/tty1
crw-rw----    1 root      tty          4,    2 2004-07-01 08:06 /dev/tty2
dozent@linux37:~> stty -F /dev/tty1
speed 38400 baud; line = 0;
lnext = <undef>; min = 1; time = 0;
-icrnl
-icanon -echo
dozent@linux37:~> stty -F /dev/tty2
stty: /dev/tty2: Keine Berechtigung
```

Sie können auch über eine Umleitung ein Terminal ansprechen.

```
linux37:/home/dozent # stty -F /dev/ttyS0
speed 9600 baud; line = 0;
-brkint -imaxbel
linux37:/home/dozent # stty < /dev/ttyS0
speed 9600 baud; line = 0;
-brkint -imaxbel
```

Übrigens ist auch eine serielle Schnittstelle ein Terminal.

Interessant wird es, wenn `stty` zum Setzen von Eigenschaften verwendet wird. Für ein Terminal kann eine Vielzahl von Werten gesetzt werden. Diese setzen Zeichen für verschiedene Aktionen fest, verändern die Kontrollfunktionen und die Ein- und Ausgabeeigenschaften. Eine Übersicht aller Einstellungen finden Sie in der Manualpage `stty(1)`. Hier nur ein paar Beispiele:

Sie können z. B. die Geschwindigkeit der ersten seriellen Schnittstelle auf 38400 bit/s festlegen.

```
stty -F /dev/ttyS0 38400
```

Wenn der letzte Prozess das Terminal schließt, wird mit der folgenden Anweisung das Terminal angewiesen ein SIGHUP-Signal an alle angeschlossene Prozesse zu senden. Damit sollten sich die Prozesse eigentlich beenden.

```
stty hupcl
```

Ein Minuszeichen vor der Einstellung kehrt die Bedeutung um.

```
stty -hupcl
```

Sie können sich auch die Größe des Terminals in Zeilen und Zeichen ausgeben lassen.

```
linux37:/home/dozent # stty size
24 80
```

Nach einer neuen Zeile soll möglichst auch ein Wagenrücklauf erfolgen, sprich der Cursor sollte am Anfang der Zeile stehen. Der folgende Befehl führt zu etwas Chaos auf dem Bildschirm. Können Sie ihn auch wieder rückgängig machen?

```
stty -onlcr
```

Alle Zeichen können für die Ausgabe in Großbuchstaben umgewandelt werden.

```
linux37:/home/dozent # stty olcuc
LINUX37:/HOME/DOZENT # STTY -OLCUC
linux37:/home/dozent #
```

Und es gibt noch viele Einstellungen mehr. Es ist sehr unwahrscheinlich, daß Sie etwas mit diesen Einstellungen zu tun haben sollten. Aber Sie sollten wissen, daß man mit `stty` die Einstellungen von Terminals abrufen und ändern kann.

### 5.1.4 Automatische Kommandoergänzung

Die Shell ist konzipiert worden, um dem Benutzer eine komfortable Umgebung zum Arbeiten zu schaffen. Ein Tool dafür ist die automatische Kommandoergänzung. Oft reicht es aus, nur die ersten Zeichen eines Befehls oder Dateinamens einzugeben um diesen eindeutig zu identifizieren. Durch Betätigen der Tabulator-Taste (<TAB>) versucht die Shell diesen Namen zu vervollständigen.

So erweitert die Shell nach Betätigung der Tabulator-Taste die Buchstaben `ec` zu dem Befehl `echo`.

Ist der Befehl der Dateiname nicht eindeutig über die ersten Zeichen definiert, so gibt die Shell ein Signalton zurück. Wird die Tabulator-Taste noch einmal betätigt, zeigt die Shell eine Liste von Kommandos, die mit den entsprechenden Zeichen beginnen.

So erhalten Sie beispielsweise nach Eingabe von `mk` und zweimaligem Betätigen der Tabulator-Taste folgende Liste<sup>1</sup>.

```
mkdir      mkfontdesc  mkmanifest  mktemp      mktexpk
mkdirhier  mkfontdir   mknod       mktexlsr    mktexftm
mkfifo     mkindex     mksusewmrc  mktexmf
```

Auf diese Art und Weise können Sie sich auch eine Liste aller zur Verfügung stehenden Befehle ausgeben lassen. Drücken Sie einfach am leeren Prompt zweimal die Tabulator-Taste. Da es nicht gerade wenig Befehle sind, werden Sie dann erst einmal von der Bash gefragt, ob Sie auch wirklich alle Treffer anzeigen lassen möchten.

```
tapico@defiant:~><TAB><TAB>
Display all 2192 possibilities? (y or n)
```

Genauso kann man sich auch schnell durch das Dateisystem bewegen. Durch die Eingabe von

```
cd /h<TAB>t<TAB>mo<TAB>l<TAB>
```

erhält man z. B. die Kommandosequenz `cd /home/tapico/moebius/linux`.

Auch wenn Sie den Pfad nicht kennen, hilft die Tabulatortaste weiter. Einfach zweimal die Tabulatortaste drücken und alle in Frage kommenden Verzeichnisse werden angezeigt.

```
tapico@defiant:~> cd /var/<TAB><TAB>
X11R6      games      log         opt         squid      yp
adm         lib         mail        run         tmp
cache      lock        named       spool       ucd-snmp
tapico@defiant:~> cd /var/
```

Und was für Verzeichnisse gilt, gilt auch für alle anderen Dateien. Dann natürlich nicht mit `cd`, sondern einem dateibezogenen Befehl wie z. B. `less`.

```
tapico@defiant:~> less /etc/cron<TAB><TAB>
cron.d      cron.hourly  cron.weekly  crontab.old
cron.daily  cron.monthly crontab
tapico@defiant:~> less /etc/cron
```

### 5.1.5 Gruppierung von Kommandos

Unter Linux ist es möglich, mehrere Kommandos in einer Zeile einzugeben. Dabei werden die einzelnen Kommandos durch das Zeichen `;` getrennt.

```
tapico@defiant:~/test/ > date; echo "Homeverzeichnis"; ls ~
Die Sep 19 23:55:00 MEST 2000
Homeverzeichnis
```

---

<sup>1</sup>Die Liste ist natürlich von System zu System unterschiedlich, je nachdem welche Programme installiert sind.

Desktop	datum	public_html	test.pl	who
Mail	datum.old	public_html_old	test1.pl	whoami
cgi-bin	manpages	test	test2.pl	zulu.txt

Damit die Ausgabe aller Befehle in eine Datei umgeleitet werden kann, müssen die Befehle gruppiert werden. Denn die Befehlssequenz

```
date; echo "Homeverzeichnis"; ls ~ > dir.txt
```

würde nur die Ausgabe des `ls`-Befehls in die Datei umlenken. Richtig muß die Kommandosequenz lauten:

```
(date; echo "Homeverzeichnis"; ls ~) > dir.txt
```

Die Klammern gruppieren die Kommandos zu einem Befehl. Deshalb wird dies als Kommandogruppierung bezeichnet.

### 5.1.6 Bedingte Ausführung

Die Zeichen `&&` und `||` verknüpfen zwei Befehle miteinander, wobei der zweite Befehl in Abhängigkeit vom Erfolg des ersten Befehls ausgeführt wird.

```
less index.htm || less index.html
```

bewirkt, daß nur wenn der erste Befehl keinen Erfolg hat, weil z. B. die Datei nicht existiert, der zweite Befehl ausgeführt wird.

```
cat index.html && cp index.html /backup
```

bewirkt, daß nur wenn der erste Befehle erfolgreich war, der zweite Befehl ausgeführt wird.

### 5.1.7 Substituierung von Kommandos

Oft ist es sinnvoll die Ausgabe eines Kommandos in einem anderen Kommando zu verwerten. Das Einschließen eines Kommandos in `$(...)` oder in `'...'` bewirkt, daß die Ausgabe des Kommandos verwendet wird.

```
tapico@defiant:~ > echo $(pwd)
/home/tapico
tapico@defiant:~ > echo 'pwd'
/home/tapico
```

Die obere Methode ist die neue Syntax der `bash`. Die zweite zeigt die Syntax der Bourne Shell. Mindestens eine der beiden Methode wird in der `bash` funktionieren.

**Beispiel** Sie wollen den Kalender des aktuellen Monats in eine gleichnamigen Datei speichern.

```
tapico@enterprise:~/cal> cal -m > kalender-$(date +"%Y-%m").txt
tapico@enterprise:~/cal> ls -l kalender-*
-rw-r--r--  1 tapico  users          129 Apr 14 12:57 kalender-2002-04.txt
```

## 5.2 Variablen

Um bestimmte, immer wiederkehrende Zeichenfolge zu speichern, werden in der Shell sogenannte *Variablen* verwendet. Jede Variable besitzt einen Namen, über den auf diese Zeichenfolge zugegriffen werden kann. Einer Benutzervariable kann ein Name und ein Wert zugewiesen werden, während die Namen der Umgebungsvariablen festgelegt sind. Eine besondere Form der Benutzervariablen sind die *Aliase*.

Name	Bedeutung
?	Fehlercode des letzten ausgeführten Kommandos (0 für "OK")
CDPATH	Suchpfad für das <code>cd</code> -Kommando
EDITOR	Name des Editors, den man verwenden möchte
EUID	Effektive Benutzerkennung (ro)
FCEDIT	Editor für den Befehl <code>fc</code>
HISTFILE	Name und Pfad für die History-Datei
HISTFILESIZE	Max. Größe der History-Datei
HISTSIZE	Max. Anzahl von Befehlen zur Speicherung in der History-Datei
HOME	Heimatverzeichnis des Benutzers
HOSTNAME	Name des Rechners
INPUTRC	Konfigurationsdatei für Kommandozeileneditor
PATH	Suchpfad für Kommandos
PS1	Aktueller Prompt
PS2	Prompt für die Eingabe fehlender Parameter
PWD	Name des aktuellen Verzeichnisses
RANDOM	Zufallszahl
SECONDS	Sekunden nach Starten der Shell
SHELL	Name der benutzten Login-Shell
TERM	Art des Linux-Terminals
UID	Benutzerkennung (ro)
USER	Login-Name des Users

Tabelle 5.2: Auszug aus den Umgebungsvariablen

### 5.2.1 Benutzervariablen

Benutzervariablen werden meistens in Skripten verwendet. Die Variablennamen dürfen aus Buchstaben, Ziffern und dem Unterstrich bestehen. Dabei darf das erste Zeichen keine Ziffer sein.

Den Variablen wird ein Wert durch folgende Sequenz zugewiesen.

```
VARIABLENNAME=WERT
```

Um eine Variable zu löschen, reicht es ihr einen Null-Wert zu geben.

```
VARIABLENNAME=
```

Die Variablen können nur von dem erzeugenden Prozess und seinen Kinderprozessen verwendet werden. Es sind lokale Variablen.

### 5.2.2 export

Um eine Variable überall zugänglich zu machen, wird das bash-Kommando<sup>2</sup> `export` verwendet.

```
export [OPTIONEN] [NAME[=WERT]]
```

`export` ohne die Angabe von Optionen zeigt eine Liste aller exportierten Variablen.

#### Optionen

- f Die Variable wird als Funktion interpretiert
- n Löschen einer Variablen aus der Liste
- p Liste aller exportierten Variablen (standard)

#### Beispiel

Folgende Kommandos erstellen eine Variable und machen sie global zugänglich.

```
ILove=Linux; export ILove
```

---

<sup>2</sup>built-in command: Befehl, der direkt in die Shell eingebaut ist und nicht als eigenständiges Programm vorliegt.



### 5.2.3 Umgebungsvariablen

Aussehen und Funktionsweise der Shell werden durch die Umgebungsvariablen festgelegt. Die Festlegung der Umgebungsvariablen erfolgt in verschiedenen Dateien, die in `/etc` oder im Heimatverzeichnis `~` liegen. So findet man z. B. die Umgebungsvariablen der `bash` in der Datei `/etc/profile` wieder. Einen Auszug von Umgebungsvariablen gibt Tabelle 5.2 wieder. Ausführlichere Informationen über die Umgebungsvariablen liefert das Kommando `man bash`.

#### Beispiel

Schauen wir uns als Beispiel die Variable `PWD` an, die den Namen des aktuellen Verzeichnisses enthält.

```
tapico@defiant:~/test > echo $PWD
/home/tapico/test
```

Der Variablen `PWD` kann ein neuer Wert zugewiesen werden.

```
tapico@defiant:~/test > PWD=/home/tapico
tapico@defiant:~/ > echo $PWD
/home/tapico
tapico@defiant:~/ > pwd
/home/tapico/test
tapico@defiant:~/ > echo $PS1
\u@\h:\w >
```

Nach der Änderung der Umgebungsvariable `PWD` ändert sich auch das Aussehen des Prompts, da dieser die Umgebungsvariable nutzt. Wie aber der Befehl `pwd` zeigt, hat sich das aktuelle Verzeichnis nicht geändert.

### 5.2.4 printenv

Um sich die angelegten Umgebungsvariablen ansehen zu können, wird der Befehl `printenv` verwendet.

```
printenv [VARIABLE]
```

Wird keine spezielle Variable angegeben, so zeigt `printenv` eine Liste aller Variablen an.

### 5.2.5 env

`env` startet ein Kommando mit anderen Umgebungsvariablen.

```
env [OPTIONEN] [VARIABLE=WERT] [BEFEHL [ARG1] ...]
```

Wird kein `BEFEHL` angegeben, so listet `env` die resultierende Umgebung auf. Es arbeitet dann wie `printenv` (5.2.4).

### 5.2.6 Suchpfad

Die Umgebungsvariable `PATH` enthält eine Liste von Verzeichnissen, die nach einem Kommando durchsucht werden. Die einzelnen Verzeichnisnamen werden durch einen Doppelpunkt `:` getrennt. Im Gegensatz zu DOS wird das aktuelle Verzeichnis `.` nicht immer durchsucht, sondern muß auch in der `PATH`-Variable angegeben werden.

```
tapico@defiant:~/ > echo $PATH
/bin:/usr/bin:/usr/local/bin:.
```

Im Normalfall ist das aktuelle Verzeichnis sogar das letzte in der Suchreihenfolge. Das kann zu Problemen führen. So wird beim Ausprobieren von Skripten z. B. sehr gerne der Name `test` als Dateiname verwendet. `test` ist aber ein Programm im Verzeichnis `/usr/bin`. Daher wird beim Aufruf von `test` nicht das Skript im aktuellen Verzeichnis, sondern das Programm in `/usr/bin` gestartet, weil es dort zuerst gefunden wurde.

Die Variable `PATH` wird in der Datei `/etc/profile` definiert. Um die Datei für sich selber zu ändern, sollte die Änderung in den Datei `~/.bash_profile`, `~/.profile` oder `~/.bashrc` erfolgen. Welche Datei als Beste dafür geeignet ist, ist von Distribution zu Distribution unterschiedlich.

Bsp.: `PATH=$PATH:$HOME/bin`

### 5.2.7 Prompt

Der Prompt steht immer am Anfang einer Kommandozeile. Er zeigt an, daß die Shell auf eine Eingabe wartet. Das Aussehen des Prompts wird durch die Umgebungsvariable `PS1` festgelegt.

#### Beispiel

```
tapico@defiant:~ > echo $PS1
\u@\h:\w >
```

`\u`, `\h` und `\w` sind besondere Zeichen, die ausführliche Informationen enthalten. Für eine Liste der speziellen Zeichen siehe Tabelle 5.3.

Wie der Name `PS1` schon vermuten läßt, existiert noch eine Umgebungsvariable `PS2`. Diese steuert das Aussehen des zweiten Prompts. Dieser Prompt taucht immer dann auf, wenn Sie einen Befehl noch nicht vollständig eingegeben und doch die Eingabetaste gedrückt haben. Er zeigt an, daß noch etwas fehlt.

Im folgenden Beispiel fehlt das schließende Anführungszeichen.

```
ole@enterprise:~> echo $PS2
>
ole@enterprise:~> PS2="ergänzen >"
ole@enterprise:~> echo $PS2
ergänzen >
ole@enterprise:~> echo "Hallo
ergänzen >Welt "
Hallo
Welt
ole@enterprise:~> echo $PS2
ergänzen >
```

### 5.2.8 alias

Für häufig wiederkehrende Befehle kann ein Alias angelegt werden. Dieses Alias ist vergleichbar mit einem kleinen Skript, wird aber wesentlich schneller ausgeführt. Um z. B. für den Befehl `cd ..` den Alias `..` zu setzen, wird folgendes Kommando eingegeben.

```
alias ..="cd .."
```

Die allgemeine Syntax für den Befehl `alias` lautet:

```
alias [NAME[=KOMMANDO]]
```

Dabei gibt das Kommando `alias` unter Angabe des Aliasnamen das dazugehörige Kommando aus. Der Aufruf ohne Parameter gibt eine Liste aller definierten Aliase mit ihren Kommandosequenzen aus.

Der Befehl `alias` sowie der Gegenbefehl `unalias` sind fester Bestandteil der Bash und direkt eingebaut.

Zeichen		Bedeutung
\a	<i>alert</i>	Piepton
\d	<i>date</i>	Datum Format: Wochentag Monat Tag
\e	<i>escape</i>	ESC-Zeichen
\h	<i>host</i>	Hostname bis zum ersten .
\H	<i>host</i>	Der ganze Hostname
\n	<i>newline</i>	Neue Zeile
\r	<i>return</i>	Wagenrücklauf
\s	<i>shell</i>	Name der Shell
\t	<i>time</i>	Zeit Format: 24 Stunden HH:MM:SS
\T	<i>time</i>	Zeit Format: 12 Stunden HH:MM:SS
\@		Zeit Format: 12 Stunden a.m./p.m.
\u	<i>user</i>	Username des aktuellen Benutzers
\v	<i>version</i>	Version der <b>bash</b>
\V	<i>version</i>	Ausgabennummer der <b>bash</b> (Version + Patchstufe)
\w	<i>working</i>	Aktuelle Arbeitsverzeichnis
\W		Der Basisname des aktuellen Verzeichnisses
\!		Die Nummer in der History für den aktuellen Befehl
\#		Die Nummer des Kommandos für den aktuellen Befehl
\\$		Wenn die effective UID 0 ist, dann #, sonst \$
\nnn		Zeichen mit dem oktalen ASCII-Code nnn
\\		Backslash
\[		Beginn einer Folge von nichtdruckbaren Zeichen
\]		Ende einer Folge von nichtdruckbaren Zeichen

Tabelle 5.3: Spezielle Zeichen für den Prompt. Die kursiven Worte sind Eselsbrücken zum Merken

### 5.2.9 unalias

Um die ganze Alias-Geschichte rückgängig zu machen, wird der Befehl **unalias** verwendet. Damit kann der normale Benutzer Aliase wieder entfernen, die der Systemverwalter für alle Benutzer durch Eintragen in die `/etc/.profile` angelegt hat.

```
unalias [-a] [NAMENSLISTE]
```

Wenn Sie den Schalter `-a` verwenden, arbeitet **unalias** sehr gründlich: Alle Alias-Definitionen werden gelöscht.

## 5.3 History-Liste

Die History-Liste umfaßt die letzten eingegeben Kommandos (**events**). Diese Liste kann dazu benutzt werden um alte Befehle wieder zu benutzen, sie zu bearbeiten oder sie zu analysieren. Dabei wird jedes Element von der Shell mit einer Nummer versehen.

Beim Starten der Shell wird die History-Liste aus der Datei `.bash_history` im Heimatverzeichnis erstellt. Die Umgebungsvariable `HISTFILE` enthält den Namen dieser Datei. Die Größe der Datei in Zeilen wird durch `HISTFILESIZE` bestimmt. Beim Ausloggen wird eine Anzahl von Zeilen, die in der Variablen `HISTSIZE` definiert worden ist, an die Datei `.bash_history` angehängt oder die komplette Datei wird durch diese ersetzt, je nachdem ob das `histappend`-Attribut der Shell gesetzt ist oder nicht. Mit den Shellvariablen `HISTCONTROL` und `HISTIGNORE` kann in Abhängigkeit von im Befehl enthaltenen Zeichen bestimmt werden, welcher Befehl in der History gespeichert wird und welcher nicht.

### 5.3.1 history

Das **bash**-Kommando **history** zeigt den Inhalt der History-Liste an.

**history** [OPTIONEN] [ZAHL]

Ohne Angabe von Parametern wird die gesamte History-Liste angezeigt. Durch Angabe einer Zahl werden die letzten ZAHL Zeilen der History-Liste angezeigt.

**Optionen**

**-c**                      Löschen der History-Liste

### 5.3.2 fc

Wie auch **history** ist **fc** ein **bash**-Kommando zur Arbeit mit der History-Liste. Es kann zur Bearbeitung und Wiederbenutzung von Kommandos verwendet werden.

**fc -l** [ANFANG] [ENDE]

**fc -s** [NUMMER]

Der Schalter **-f** in Kombination mit zwei Zahlen zeigt die Zeilen von ANFANG bis ENDE an. ANFANG und ENDE können auch Zeichenketten sein. Dann werden alle Zeichen vom ersten Auftauchen von ANFANG bis zum ersten Auftauchen von ENDE angezeigt.

**Optionen**

**-l**                      Zeigt ohne Parameter die letzten 16 Kommandos an.

**-s**                      Führt ohne Parameter das letzte Kommando aus. Ansonsten das Kommando mit der angegebenen Nummer.

**-e**                      Ermöglicht das Bearbeiten der History-Liste

Standardmäßig verwendet **fc** den Editor **vi** zum Bearbeiten. Durch setzen der Umgebungsvariablen **FCEDIT** kann auch ein anderer Editor dafür eingesetzt werden. So setzt

```
tapico@defiant:~> FCEDIT=/usr/bin/joe
```

```
tapico@defiant:~> export FCEDIT
```

den Editor **joe** als Standardeditor ein.

## 5.4 Umleitungen der Eingaben und Ausgaben

Normalerweise kommen die Daten für ein Programm von der Standardeingabe oder, wenn es das Programm direkt unterstützt, aus einer Datei. Die Ausgabe des Programms landet in der Standardausgabe und die Fehlermeldungen in der Standardfehlerausgabe. Im Prinzip sind Standardeingabe, Standardausgabe und Standardfehlerausgabe keine Ziele sondern Datenkanäle zu Zielen. Die Standardeingabe ist normalerweise mit der Tastatur verbunden und Standardausgabe und Standardfehlerausgabe stellen ihre Daten auf dem verbundenem Terminal dar. Diese Datenkanäle können aber auch auf ganz andere Ziel zeigen. Dies wird durch Umleitungen und Pipelines ermöglicht.

### 5.4.1 Umleitung der Eingabe und Ausgabe

Standardmäßig kommen die Eingaben von der Tastatur und die Ausgabe des Programms und seiner Fehlermeldungen landen auf dem Terminal und damit auf dem Bildschirm.

**Ausgabe**

**Ausgabe in eine Datei** Mit dem Zeichen **>** oder **1>** wird die Ausgabe des Programm in die angegebene Datei umgelenkt. Ob das Eingabeziel vor dem Befehl oder nach dem Befehl steht ist relativ egal. Als Konvention hat sich der Platz hinter dem Befehl eingebürgert.

```
ole@enterprise:~/test> cat *.txt > alle.txt
```

```
ole@enterprise:~/test> > nochmal.txt cat *.txt
```

Dabei wird jedesmal eine neue Datei erstellt und der Inhalt der alten Datei gelöscht.

**Anhängen an eine Datei** Soll die Ausgabe nicht die alte Datei löschen, dann müssen die Daten an die Datei angehängt werden. In diesem Fall verwendet man >>.

```
ole@enterprise:~/test> date >> blubb.log
ole@enterprise:~/test> >> blubb.log echo $USER $SHELL $0
```

In diesem Fall wird die blubb.log nicht durch den zweiten Befehl überschrieben sondern ergänzt.

**Ausgabe von Fehlermeldungen** Obwohl die Fehlermeldungen auch an das Terminal gehen, werden sie durch ihren eigenen Datenkanal transportiert. Dies muß natürlich auch im Umleitungszeichen ausgedrückt werden. Mit 2> wird die Standardfehlerausgabe umgeleitet. Oft stören die Fehlermeldungen, wenn sie z. B. nur melden, daß keine Zugriffsrechte auf ein Verzeichnis bestehen.

```
ole@enterprise:~/test> ls -Rl 2> /dev/null
ole@enterprise:~/test> 2> fehler.log ls -Rl
```

Die Gerätedatei ist praktisch das Nichts von Linux. Alle Daten, die in diese Datei geleitet werden, gehen verloren und werden nie wieder gesehen. Also der ideale Ort für unsinnige Warnungen, überflüssige Fehlermeldungen oder einfach zum Testen. Schade, daß ich die Ausgabe mancher Menschen nicht in ein solches Gerät umleiten kann.

**Beide Ausgaben umleiten** Um beide Ausgaben (Standardausgabe und Standardfehlerausgabe) gemeinsam in eine Datei zu leiten, gibt es mehrere Möglichkeiten. Schauen Sie sich einfach mal die Beispiele für den Befehl grep (7.7.1), mit dem Dateien durchsucht werden können.

```
ole@enterprise:~/test> grep -r enterprise /etc/ &> blubb.txt
ole@enterprise:~/test> grep -r enterprise /etc/ >& blubb.txt
ole@enterprise:~/test> grep -r enterprise /etc/ > blubb.txt 2>&1
ole@enterprise:~/test> grep -r enterprise /etc/ 2> blubb.txt 1>&2
```

Bei den letzten Kommandos kommt es auf die Reihenfolge an. Denn die folgenden Kombinationen führen nicht zum gewünschten Erfolg.

```
ole@enterprise:~/test> grep -r enterprise /etc/ 2>&1 > blubb.txt
ole@enterprise:~/test> grep -r enterprise /etc/ 1>&2 2> blubb.txt
```

## Eingabe

**Daten aus einer Datei** Heute unterstützen die meisten Programme direkt das Lesen aus einer Datei. Allerdings ist dies nicht zwingend notwendig, da die Aufgabe des Lesens einer Datei auch die Bash übernehmen kann. Mit dem Zeichen < liest das Programm nicht von der Standardeingabe sondern holt seine Daten aus der angegebenen Datei. Es ist dabei völlig egal, ob die als Eingabe dienende Datei vor oder hinter dem befehl definiert wird.

```
ole@enterprise:~/test> cat < input.txt
In a world without walls and fences,
who needs windows and gates?
ole@enterprise:~/test> < input.txt cat
In a world without walls and fences,
who needs windows and gates?
```

**Simulation einer Datei zur Eingabe** Sie können auch den Einsatz einer Datei mit der Tastatur simulieren. Mit den Zeichen << und einer Endemarkierung können Sie auf der Tastatur so viele Zeilen eingeben, wie sie wollen. Abgeschlossen wird der Vorgang, wenn Sie in eine Zeile nur die Endemarkierung hineinschreiben.

```
ole@enterprise:~/test> sort <<ENDE
> Leif Neid
> Thore Wart
> Axel Schweiß
> Erk Lasse
> Walter Meingott
> ENDE
Axel Schweiß
Erk Lasse
Leif Neid
Thore Wart
Walter Meingott
```

Die Ein- und Ausgabeumleitungen können natürlich wild miteinander kombiniert werden. Das kann zu manchmal merkwürdigen Befehlen führen, die aber völlig in Ordnung sind.

### 5.4.2 Pipelines

Pipelines ermöglichen es, die Ausgabe eines Kommandos direkt an die Eingabe eines anderen Kommandos zu senden. Realisiert wird dies durch ein zwischengeschaltetes Pipe-Zeichen |.

```
ole@enterprise:~/test> ls -l /dev | less
```

Eine Philosophie von Linux/Unix ist die Verwendung kleiner spezialisierter Programme anstatt großer universeller "Eierlengender Wollmilchsäue". Die einzelnen Kommandos werden dann mit Pipelines miteinander verbunden. Beispiele für die Anwendung der Pipelines finden Sie ausführlich in Kapitel 7 *Textfilter*.

### 5.4.3 tee

Das Kommando **tee** sendet gleichzeitig die Ausgabe an eine Datei und an die Standardausgabe, die der Bildschirm oder eine weitere Pipeline sein kann.

```
tee [OPTIONEN] DATEILISTE
```

#### Optionen

-a	Die Daten werden an eine vorhandene Datei angehängt
-i	Ignoriert Interrupt-Signale

#### Beispiele

Diese Kommandosequenz sendet den Inhalt des Verzeichnisses in die Datei **schmeckt** und auf den Bildschirm.

```
ls -al | tee schmeckt
```

Sie können auch mehrere Dateien gleichzeitig mit dem Inhalt des Verzeichnisses füllen. In diesem Fall sind es die Dateien **schmeckt** und **gut**.

```
ls -al | tee schmeckt gut
```

Hier wird der Inhalt der Datei **amAngelhaken** an die Datei **angeln.txt** angefügt und gleichzeitig mit **less** angezeigt.

```
cat amAngelhaken | tee -a angeln.txt | less
```

#### 5.4.4 xargs

Der Befehl `xargs` ermöglicht die Weitergabe von Argumenten von einem Kommando zu einem anderen.

`xargs` KOMMANDO

Dabei ordnet `xargs` die übergebenen Daten als durch Leerzeichen getrennte Reihe von Worten an.

##### Beispiele

Dieser Befehl ist ein gutes Beispiel für die Verarbeitung der übergebenen Daten.

```
ls -l | xargs echo
```

Dieses Beispiel faßt die in der Datei `liste` aufgeführten Dateien zu einer Datei `myFiles` zusammen und gibt sie gleichzeitig mit `less` auf dem Bildschirm aus.

```
cat liste | xargs cat | tee myFiles | less
```

Oder wollten Sie nicht schon mal wissen, was eigentlich die Programme im Verzeichnis `/bin` so machen. Der Befehl `whatis` (6.3.3) gibt eine Kurzbeschreibung aus.

```
ls /bin | xargs whatis | less
```

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....



V

Arbeitsblatt  
Shell 2

5.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

Vergewissern Sie sich vor der Bearbeitung der Aufgaben, daß die Korn-Shell installiert wurde.

- 1 Loggen Sie sich als Walter auf der Konsole 1 ein.
- 2 Führen Sie das Kommando `rm -rf *` aus.
- 3 Ermitteln Sie den Namen der aktuellen Shell.
- 4 Starten Sie jetzt die Korn-Shell.
- 5 Wiederholen Sie den Befehl aus Aufgabe 3. Welches Ergebnis erhalten Sie diesmal?
- 6 Kehren Sie zur ursprünglichen Shell zurück.
- 7 Ändern Sie für Ihren Benutzer die Standardshell auf die Korn-Shell.
- 8 Loggen Sie sich aus.
- 9 Loggen Sie sich wieder ein.
- 10 Wiederholen Sie den Befehl aus Aufgabe 3. Welches Ergebnis erhalten Sie diesmal?
- 11 Machen Sie die Shell-Änderung wieder rückgängig.
- 12 Loggen Sie sich wieder aus und dann wieder ein.
- 13 Lassen Sie sich das Tagesdatum in der in Deutschland gebräuchlichen Form (Tag.Monat.Jahr) anzeigen.
- 14 Welche Tastenkombination unterbricht ein laufendes Programm?
- 15 Was für ein Wochentag war der 18. März 1970?
- 16 Was war das besondere am September 1752?
- 17 Speichern Sie den Kalender für das Jahr 2001 in der in Deutschland gebräuchlichen Form in der Datei `Kalender`.
- 18 Wie spät ist es auf dem Linux-Rechner?
- 19 Mit welchem Terminaltyp arbeiten Sie gerade?
- 20 Erstellen Sie eine persönliche `inputrc`-Datei. Belegen Sie die Tasten `<F1>` mit dem Befehl für den aktuellen Jahreskalender `<F2>` mit dem Befehl für die Uhrzeit und `<F3>` mit dem Befehl für den Editor `vi`.
- 21 Testen Sie die neuen Tastenbelegungen.
- 22 Lassen Sie sich die Datei `/etc/DIR_COLORS` anzeigen.
- 23 Kopieren Sie die Datei `/etc/DIR_COLORS` als `.dir_colors` in Ihr Heimatverzeichnis.
- 24 Ändern Sie den `ls`-Farbcode für Verzeichnis auf rot und unterstrichen.
- 25 Testen Sie die Änderung der Farbcodes.
- 26 Lassen Sie sich die Umgebungsvariable für den aktuellen Prompt anzeigen.

V

Arbeitsblatt  
Shell 2

5.2

**27** Ändern Sie den Prompt zu: "BENUTZERNAME @ HOSTNAME SHELL SHELLVERSION+PATCHLEVEL am DATUM : VERZEICHNIS >"

Bsp.: tapico@defiant bash 2.03.0 am Fri Dec 29: ~ >

**28** Loggen Sie sich aus und wieder ein. Wie sieht der Prompt nun aus?

**29** Richten Sie nun die Promptänderung so ein, daß Sie bei jedem Einloggen gilt.

**30** Lassen Sie sich alle Umgebungsvariablen anzeigen.

**31** Welche Bedeutung hat die Variable \_.

**32** Geben Sie die Liste der Umgebungsvariablen zusammen mit dem Datum und dem Text "Umgebungsvariablen" in die Datei mit dem Namen `variablenJJ-MM-TT.txt` aus. Dabei steht JJ-MM-TT für das aktuelle Datum.

**33** Speichern Sie die Dateinamen aus dem Verzeichnis `/usr/share/doc/packages/cron` in der Datei `dateiliste`.

**34** Kopieren Sie alle Dateien aus `/usr/share/doc/packages/cron` und `/usr/share/doc/packages/yast2` in Ihr Heimatverzeichnis.

**35** Fassen Sie den Inhalt der Dateien, die Sie aus `/usr/share/doc/packages/cron` kopiert haben und die sich nun in Ihrem Heimatverzeichnis befinden, zu einer Datei `alleAnleitungen.txt` zusammen.

**36** Wechseln Sie zur Konsole 2.

**37** Loggen Sie sich als **root** ein.

**38** Legen Sie das Verzeichnis `/kurs/bin` an und wechseln Sie in das Verzeichnis.

**39** Erstellen Sie eine Datei mit dem Namen `hurra` und dem Befehl  
`echo -e "Hurra\n"`  
als Inhalt.

**40** Geben Sie den Befehl `chmod 755 hurra` ein. Die Datei wird damit ausführbar.

**41** Wechseln Sie wieder zur Konsole 1.

**42** Starten Sie das Programm `hurra`. Klappt das?

**43** Starten Sie nun das Programm `hurra` mit Angabe des Pfades `/kurs/bin`. Klappt das?

**44** Überprüfen Sie, ob das Verzeichnis `/kurs/bin` in der Pfadliste enthalten ist.

**45** Fügen Sie das Verzeichnis `/kurs/bin` der Verzeichnisliste hinzu.

**46** Starten Sie nun wieder das Programm `hurra`. Klappt das?

**47** Legen Sie das Alias `adir` mit `ls -al | less` an.

**48** Legen Sie das Alias `werbinich` an. Es soll den Namen ausgeben, mit dem Sie sich eingeloggt haben.

**49** Testen Sie die Aliase.

**50** Loggen Sie sich aus und wieder ein.

V

Arbeitsblatt  
Shell 2

5.3

- 51** Testen Sie wieder die Aliase.
- 52** Legen Sie in Ihrem Heimatverzeichnis eine Datei `.alias` ein und tragen Sie dort die beiden Aliase ein.
- 53** Loggen Sie sich aus und wieder ein.
- 54** Testen Sie wieder die Aliase.
- 55** Legen Sie ein Alias `myenv` in der Datei `.alias` ein. Das Alias soll für den Befehl aus Aufgabe 32 stehen.
- 56** Führen Sie die Datei `.alias` aus.
- 57** Testen Sie den neuen Alias.
- 58** Führen Sie wieder den Befehl `hurra` aus. Klappt das? Wenn nicht, warum nicht?

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## Kapitel 6

# Hilfe und Dokumentation

*Ein Mathematiker, ein Physiker und ein Computeruser werden getrennt von einander eingeschlossen. Jeder erhält zwei Glaskugeln. Nach einer Stunde schaut man was diese Leute damit machen. Der Mathematiker sitzt dort und berechnet das Volumen und die Oberfläche der Kugeln. Der Physiker hält die Kugeln gegen das Licht und berechnet Brechzahl und Absorptionskoeffizient. Als letztes schaut man beim Computerbenutzer herein und stellt fest, daß eine Kugel weg ist und das Fenster zerbrochen. Auf die Frage, was denn passiert sei, zuckt der User nur mit den Achseln und sagt: "Ich hab' nix gemacht!..."*

*Anonymous*

Ein nicht unerheblicher Nachteil der Shell gegenüber der graphischen Benutzeroberfläche ist eindeutig der Umstand, daß man die Befehle des Systems kennen muß. Bei der graphischen Oberfläche sind die meisten Einstellmöglichkeiten und Aktionen gut beschriftet. Sie müssen als nur Lesen können und es natürlich auch tun. Zum Glück gibt es im Linux-System und auch im Internet umfangreiche Hilfen und Anleitungen, damit Sie nicht alle Befehle auswendig lernen müssen. Dieses Kapitel stellt Ihnen die wichtigsten Hilfe-Quellen vor.

### 6.1 Lokale Dokumentation

Es gibt eine Reihe von lokalen Dokumentationen in einem Linux-System. Diese befinden sich alle nach dem FHS unter dem Verzeichnis `/usr/share` bzw. bei älteren Distributionen unter dem Verzeichnis `/usr/share`. Tabelle 6.1 zeigt die Typen und ihre Position im Dateisystem.

#### 6.1.1 Das Handbuch: Manual-Pages

Die *Manual-Pages*, auch oft *Man-Pages* genannt, sind praktisch das Handbuch zu Linux und den dazugehörigen Tools. Für fast jedes Programm der Shell gibt es eine solche "Handbuchseite". Neben Programmen gibt

Dokumentation	Verzeichnis	altes Verzeichnis
Manual-Pages	<code>/usr/share/man</code>	<code>/usr/man</code>
Texinfo	<code>/usr/share/info</code>	<code>/usr/info</code>
HOWTOs	<code>/usr/share/doc/howto</code>	<code>/usr/doc/HOWTO</code>
Frequently Ask Questions (FAQ)	nicht mehr aktuell	<code>/usr/doc/FAQ</code>
Programm Dokumentationen	<code>/usr/share/doc/packages/PAKETNAME</code>	<code>/usr/doc/PAKETNAME</code>

Tabelle 6.1: Die lokalen Dokumentationen und ihre Verzeichnisse

Abschnitt	Beschreibung
NAME	Name und kurze Beschreibung des Kommandos
SYNOPSIS	Anwendung und Syntax des Kommandos
DESCRIPTION	Beschreibung des Kommandos und seiner Optionen
OPTIONS	Liste aller Optionen mit Kurzbeschreibung
SEE ALSO	Manual-Pages zu verwandten Inhalten
DIAGNOSTICS	Beschreibung von Fehlermeldungen
FILES	Liste der Dateien, die von dem Kommando verwendet werden
BUGS	Bekannte Probleme
HISTORY	Meilensteine der Programmentwicklung
AUTHOR	Autor des Programms und Mitarbeiter

Tabelle 6.2: Die Abschnitte in den Manual-Pages

es auch Man-Pages für Konfigurationsdateien, Systemaufrufe u.a. Diese Dokumentationen sind nicht gerade benutzerfreundlich, da sie mehr als Referenz für die Syntax gedacht sind und nicht als Online-Tutorial.

Für die Verwendung der Manual-Pages wird das Tool `man` (6.1.2) verwendet. Dieses Tool sucht den passenden Hilfetext heraus und stellt ihn durch einen Textbetrachter dar. In den meisten Fällen ist `less` der Pager.

Die Hilfetexte sind in mehrere Texte aufgeteilt, wie in Tabelle 6.2 zu sehen. Es muß aber nicht unbedingt jeder Abschnitt in dem Text vorkommen.

Die Handbuchseiten liegen als gezippte ASCII-Code vor, die in einem speziellen Format geschrieben wurden. Das Programm `groff` kann diese Seiten für die Ausgabe aufbereiten. Die Dateien für die Manual-Pages können in verschiedenen Pfaden liegen. Generell ist dies der Pfad `/usr/share/man`. Bei älteren Distribution liegen die Hilfetexte im Verzeichnis `/usr/man`. Die Umgebungsvariable `MANPATH` enthält die Pfade zu den Hilfetexten. Für die Ermittlung der Verzeichnisse kann auch der Befehl `manpath` verwendet werden.

```
tapico@defiant:~ > echo $MANPATH
/usr/local/man:/usr/share/man:/usr/X11R6/man:/opt/gnome/man
```

In diesen Verzeichnissen existieren mehrere Unterverzeichnisse für die neun "Kapitel" der Manual-Pages. Am wichtigsten für Administratoren sind die Kapitel 1, 5 und 8. Eine Übersicht über die Kapitel liefert die Tabelle 6.3. Da einige Themen in mehreren Kapiteln vorkommen können, wird die Kapitelnummer bei Verweisen aufs Handbuch in Klammern hinter das Thema geschrieben. Sie finden also die Informationen zu dem Befehl `passwd` in der Manual-Page **passwd(1)** und zu der Datei `/etc/passwd` in der Manual-Page unter **passwd(5)**. Es können durch Programme aber noch weitere Kapitel hinzugefügt werden, wie Sie es im folgenden Beispiel sehen. Für die SSL-Verschlüsselung gibt es auch ein Passwort-Tool.

```
enterprise:~ # whatis passwd
passwd (1ssl)      - compute password hashes
passwd (1)         - change user password
passwd (5)         - The password file
```

Neben den normalen Benutzerbefehlen (`man1`), den Systemverwalterbefehlen (`man8`) und den Dateiformaten (`man5`), gibt es noch weitere wichtige Kapitel. So enthält Kapitel 7 auch Informationen über die Implementierung der Netzwerkprotokolle.

```
enterprise:~ # whatis tcp udp ip arp
tcp (7)           - TCP protocol.
udp (7)           - User Datagram Protocol for IPv4
ip (7)            - Linux IPv4 protocol implementation
ip (8)            - show / manipulate routing, devices, policy routing and tunnels
arp (7)           - Linux ARP kernel module.
arp (8)           - manipulate the system ARP cache
```

Verzeichnis	Beschreibung
<code>man1</code>	Ausführbare Programm oder Shell-Kommandos für den Benutzer
<code>man2</code>	Systemaufrufe (Kernelfunktionen)
<code>man3</code>	Bibliotheksaufufe (Funktionen mit Systembibliotheken)
<code>man4</code>	Spezielle Dateien (meistens Gerätedateien in <code>/dev</code> )
<code>man5</code>	Dateiformate und Konfigurationsdateien
<code>man6</code>	Spiele
<code>man7</code>	Diverses, Makropakete, Konventionen, Netzwerkprotokolle
<code>man8</code>	Systemadministrationskommandos
<code>man9</code>	Kernelroutinen (kein Standard)

Tabelle 6.3: Die neun “Kapitel” der Manual-Pages

Wenn nun über den Befehl `man` eine Seite zu einem Thema angefordert wird, wird die erste Seite angezeigt, die gefunden wird. Dabei werden die Kapitel in der Reihenfolge

1, 8, 2, 3, 4, 5, 6, 7, 9

durchsucht. Deshalb findet `man` z. B. bei dem Aufruf von `passwd` immer das gleichnamige Tool. Um die Beschreibung der Benutzerliste `/etc/passwd` zu bekommen, muß explizit die Kapitelnummer angegeben werden.

### 6.1.2 `man`

Dieses Programm dient zum Anzeigen der Manual-Pages zu einem bestimmten Thema.

```
man [OPTIONEN] [SEKTION] [THEMA]
```

Dabei wird die erste passende Seite zu dem Thema angezeigt.

#### Optionen

- `-a` Zeigt alle passenden Seiten hintereinander an
- `-f` Zeigt eine Kurzbeschreibung zu den passenden Seiten an.
- `-k` Stichwortsuche in der Hilfe nach den Kurzbeschreibungen
- `-K` Volltextsuche in der Hilfe (Achtung: Nicht in jeder Version von `man` implantiert)
- `-t` Gibt die angeforderte Man-Page als Postscript-Datei aus.
- `-L SPRACHE` Gibt die Sprache für die Manual-Page an.

Die Konfigurationsdatei für `man` ist `/etc/man.config` bzw. `/etc/manpath.config`. Die Umgebungsvariable `PAGER` enthält den Namen des Textbetrachters, den `man` verwendet. Dies ist im Normalfall das Programm `less` (Abschnitt 4.5.10).

```
ole@enterprise:~> echo $PAGER
less
```

Die meisten Manual-Pages liegen in Englisch vor. Manche Seiten sind auch in Deutsch und anderen Sprachen erhältlich. Verantwortlich für die Auswahl der Sprache durch `man` ist u.a. die Variable `LANG`.

```
ole@enterprise:~> echo $LANG
de_DE@euro
```

Um Themen aus verschiedenen Kapiteln anzeigen zu lassen, muß die Kapitelnummer vor dem Thema angegeben werden.

```
ole@enterprise:~> man -f groff
groff (1)          - front end for the groff document formatting system
groff (7)          - a short reference for the GNU roff language
ole@enterprise:~> man 1 groff
```

```
Formatiere groff(1) neu, bitte warten...
ole@enterprise:~> man 7 groff
Formatiere groff(7) neu, bitte warten...
```

Die Manual-Page für `man` wird bei SuSE in einer deutschen Übersetzung angezeigt. Will man die englische Version der Seite bekommen, so muß die Sprache explizit angegeben werden.

```
ole@enterprise:~> man -L en man
```

Größere Man-Pages sind oft schlecht am Bildschirm zu lesen. Manchmal möchte man die Dateien auch ausgedruckt haben. Die folgende Befehlsfolge gibt die Manual-Page als Postscript-Datei aus, die an die Druckerwarteschlange `lp` weitergeleitet und ausgedruckt wird.

```
ole@enterprise:~> man -t cat | lpr -Plp
```

### 6.1.3 manpath

Dieses Programm versucht den aktuellen Suchpfad der Manual-Pages aus den Standardeinstellungen und der Umgebungsvariable `PATH` zu ermitteln.

```
manpath [OPTIONEN]
```

Wenn ein Verzeichnis nicht in der Konfigurationsdatei `/etc/manpath.config` aufgelistet ist, sucht `manpath` nach den Unterverzeichnissen `man` und `MAN`. Wenn diese existieren, werden sie dem Suchpfad hinzugefügt.

Der Befehl `manpath` wird von `man` benutzt um den Suchpfad zu bestimmen. Daher ist es für einen Benutzer im Normalfall nicht notwendig, die Umgebungsvariable `MANPATH` direkt zu setzen.

Weitere Informationen über Anwendung und Optionen lesen Sie bitte in den Manual-Pages **manpath(1)** nach.

### 6.1.4 TexInfo: Informationen mit info

TexInfo<sup>1</sup>, die GNU Hypertext Dokumentation, ist einfacher zu benutzen als die Manual-Pages und oft auch ausführlicher. Für die Darstellung der Dokumentation wird das Kommando `info` verwendet.

```
info [KOMMANDO]
```

Wenn Sie nur den Befehl `info` eingeben, landen Sie auf einem Inhaltsverzeichnis, das die wichtigsten Themen der TexInfo auflistet. Sie können mit den Cursortasten durch das Menü scrollen. Um in einem Menüpunkt zu wechseln, die sind durch einen vorangestellten Asterisk gekennzeichnet, müssen Sie nur mit dem Cursor auf die Zeile wechseln und `<Return>` drücken. Alternativ können Sie auch die Taste `<m>` drücken. In diesem Fall erscheint in der untersten Zeile der Text "Menu item:". Hier können Sie den Menüpunkt direkt eintippen. Dabei sind nur so viele Zeichen nötig, wie zum einwandfreien Erkennen des Menüpunkt notwendig sind. Auch diese Aktion beenden Sie mit der Taste `<Return>`.

Schauen wir uns das ganze an einem Beispiel an. Geben Sie auf der Konsole den Befehl `info` ein. Sie landen nun im Hauptmenü der TexInfo. Sie können nun mit den Cursortasten den Cursor auf den Eintrag `dd` bewegen und dort `<Return>` drücken. Oder Sie geben die Tastenfolge `<m><d><d><Return>` ein.

Nun befinden Sie sich im Eintrag für den Befehl `dd`, der schon in Abschnitt 4.5.4 behandelt wurde. In der obersten Zeile finden Sie nun mehrere Informationen. Da diese oft ausführlicher sind, setzen sie sich meistens in der zweiten Zeile fort.

**File: coreutils.info** : In dieser Datei ist der Text gespeichert, den Sie gerade sehen.

**Node: dd invocation** : Auf dieser Seite bzw. diesem Thema befinden Sie sich gerade.

---

<sup>1</sup>Das Paket `texinfo` gehört nicht mehr zur Standardinstallation ab SuSE 8.0 und muß daher nachinstalliert werden.



**Steuerung**

<q>	beendet das Programm
<STRG>+<h>	ruft die Hilfe auf
<n>	zum nächsten Knoten (Menüpunkt)
<p>	zum vorherigen Knoten (Menüpunkt)
<u>	zum aufwärtigen Knoten (Menüpunkt)
<l>	zum letzten dargestellten Knoten (auch die Hilfe ist ein Knoten)
<m>	Menüpunkt mit Namen aufrufen
<1> ... <9>	Menüpunkt mit dieser Nummer folgen

Tabelle 6.4: Steuerkommandos von `info`

**Next: install invocation** : Hier finden Sie den Namen des nächsten Themas. Durch betätigen der Taste <n> gelangen Sie dorthin.

**Prev: cp invocation** : Auch das vorherige Thema in der Reihenfolge von TexInfo wird angegeben. Sie können durch die Taste <p> dorthin wechseln.

**Up: Basic operations** : Die TexInfo ist in Abschnitte gegliedert, die wiederum Unterabschnitte u.s.w. Mit der Taste <u> können Sie in das Menü des Abschnitts wechseln, in dem die aktuelle Seite liegt.

Wechseln Sie nun mit der Taste <u> einen Abschnitt höher. Sie befinden sich nun im Abschnitt *Basic Operations*, der die Befehle `cp`, `dd`, `install`, `mv`, `rm` und `shred` behandelt.

Um wieder in den Abschnitt *dd invocation* zu wechseln können Sie die Taste `1` betätigen. Sie entspricht in ihrer Funktion des Zurück-Buttons im Webbrowser. Sie können natürlich auch die Menüfunktionen nutzen um in den gewünschten Abschnitt zu wechseln. Weil hier nur ein Menüeintrag mit dem Buchstaben *d* beginnt, brauchen Sie nur die Tastenfolge <m><d><Return> einzugeben.

Eine Übersicht über die Steuerkommandos von `info` liefert die Tabelle 6.4.

Bei Linux befinden sich die TexInfo-Dateien in den Verzeichnissen `/usr/local/info`, `/usr/share/info` und `/usr/info`.

## 6.1.5 help

Die Bash enthält eine Reihe von eingebauten Befehlen, die auch als *Builtins* bezeichnet werden. Die Hilfe zu diesen Befehlen erhalten Sie über den Befehl `help`, der ebenfalls Bestandteil der Bash ist.

```
help [KOMMANDO]
```

Der Befehl ohne Parameter zeigt eine Liste der Bash-internen Befehle an.

```
dozent@linux37:~> help
GNU bash, version 2.05b.0(1)-release (i586-suse-linux)
These shell commands are defined internally.  Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.
```

A star (\*) next to a name means that the command is disabled.

```
%[DIGITS | WORD] [&]          (( expression ))
. filename                     :
[ arg... ]                     [[ expression ]]
alias [-p] [name[=value] ... ] bg [job_spec]
bind [-lpvsPVS] [-m keymap] [-f fi break [n]
builtin [shell-builtin [arg ...]] case WORD in [PATTERN [| PATTERN] .
cd [-L|-P] [dir]               command [-pVv] command [arg ...]
compgen [-abdefgjkusv] [-o option complete [-abdefgjkusv] [-pr] [-o
```

```

continue [n]
dirs [-clpv] [+N] [-N]
echo [-neE] [arg ...]
eval [arg ...]
exit [n]
false
fg [job_spec]
for (( exp1; exp2; exp3 )); do COM
getopts optstring name [arg]
help [-s] [pattern ...]
if COMMANDS; then COMMANDS; [ elif
kill [-s sigspec | -n signum | -si
local name[=value] ...
popd [+N | -N] [-n]
pushd [dir | +N | -N] [-n]
read [-ers] [-u fd] [-t timeout] [
return [n]
set [--abefhkmnptuvxBCHP] [-o opti
shopt [-pqsu] [-o long-option] opt
suspend [-f]
time [-p] PIPELINE
trap [arg] [signal_spec ...] or tr
type [-afptP] name [name ...]
ulimit [-SHacdfilmnpstuv] [limit]
unalias [-a] [name ...]
until COMMANDS; do COMMANDS; done
wait [n]
{ COMMANDS ; }

declare [-afFirtx] [-p] name[=valu
disown [-h] [-ar] [jobspec ...]
enable [-pnds] [-a] [-f filename]
exec [-cl] [-a name] file [redirec
export [-nf] [name[=value] ...] or
fc [-e ename] [-nlr] [first] [last
for NAME [in WORDS ... ;] do COMMA
function NAME { COMMANDS ; } or NA
hash [-lr] [-p pathname] [-dt] [na
history [-c] [-d offset] [n] or hi
jobs [-lnprs] [jobspec ...] or job
let arg [arg ...]
logout
printf format [arguments]
pwd [-PL]
readonly [-anf] [name[=value] ...]
select NAME [in WORDS ... ;] do CO
shift [n]
source filename
test [expr]
times
true
typeset [-afFirtx] [-p] name[=valu
umask [-p] [-S] [mode]
unset [-f] [-v] [name ...]
variables - Some variable names an
while COMMANDS; do COMMANDS; done

```

### 6.1.6 HOWTOs

Die Linux HOWTOs sind ausführliche Dokumente, die die Bearbeitung bestimmter Aufgaben in Linux beschreiben. Die HOWTO-Dokumente behandeln meist komplexere Aufgaben und sind daher länger. Für einfachere Aufgaben wie Lilo oder Drucken gibt es kürzere Texte, die *mini-HOWTOs*.

Die HOWTO-Texte befinden sich meistens in komprimierter Form im Verzeichnis `/usr/doc/HOWTO`, `/usr/doc/howto` bzw. `/usr/share/doc/howto`. Bezugsquelle für die HOWTOs sind  
<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>  
<http://sunsite.unc.edu/LDP/HOWTO/>  
<http://www.linuxdoc.org/HOWTO/>  
bzw. einer Ihrer Spiegelserver.

### 6.1.7 FAQ

Die FAQs (Frequently Asked Questions) sind eine Sammlung von Dokumenten im Zusammenhang mit den HOWTOs. Sie sind in der Form Frage-Antwort aufgebaut, wobei die Fragen gesammelt werden, die die Benutzer am häufigsten stellen.

Die FAQs sind im HTML-, PostScript- und Text-Format vorhanden und befinden sich im Verzeichnis `/usr/doc/FAQ`. Die Online-Quelle für die neuesten FAQs ist  
<http://www.linuxdoc.org/FAQ/>.

Seit SuSE 8.0 sind die FAQs kein Bestandteil mehr der Distribution. Das ist aber zu verschmerzen, da ihr Informationswert nicht sehr groß ist.

### 6.1.8 Programmdokumentation

Dokumentationen über spezielle Programme finden Sie im Verzeichnis `/usr/doc` in einem Unterverzeichnis mit dem Namen des Programms. Bei der SuSE-Distribution finden Sie weitere Informationstexte zu den Programmen der Softwarepakete im Verzeichnis `/usr/share/doc/packages` und dessen Unterverzeichnissen. Die meisten Texte können mit `less` oder `more` gelesen werden. Andere Dokumentationen sind als  $\text{\LaTeX}$ -DVI, PostScript, HTML oder PDF zugänglich. Manche Programme legen aber auch bei Ihrer Installation eine Manual-Page an.

## 6.2 Internetquellen

Das Internet war und ist für Linux die Basis seiner Entwicklung. Der Quellcode für Linux ist übers Internet erhältlich. Desweiteren gibt es auch zahlreiche Dokumentationen an verschiedenen Orten. Die erste Quelle für Informationen zu Linux ist das **Linux Documentation Project** auf <http://tldp.org/>.

Auch gibt es durch dieses Projekt einige Bücher, die spezielle Themen abdecken. Diese können von <http://www.tldp.org/guides.html> heruntergeladen werden. U. a. sind dies

- Goldt, Sven; Meer, Sven van der; Burkett, Scott; Welsh, Matt. *The Linux Programmers Guide*
- Greenfield, Larry. *The Linux User's Guide*
- Johnson, Michael K.. *The Linux Kernel Hacker's Guide*
- Kirch, Olaf. *The Network Administrator's Guide*
- Pomerantz, Ori. *The Linux Kernel Module Programming Guide*
- Rusling, David. *The Linux Kernel*
- Welsh, Matt. *Installation and Getting Started Guide*
- Wirzenius, Lars. *The Linux System Administrators Guide*

Ein weiteres Projekt für eine mehr benutzerfreundliche Dokumentation ist Linuxnewbie.org (<http://www.linuxnewbie.org/>).

Die Linuxgazette (<http://www.linuxgazette.com/>) ist ein monatlich erscheinendes Online-Magazin zum Thema Linux. Es ist Mitglied des Linux Documentation Projects. Es gibt auch eine deutsche Ausgabe (<http://www.linuxgazette.de/>) in der einige Artikel der englischen Ausgabe ins Deutsche übersetzt wurden.

Besonders für SuSE-Nutzer ist die SuSE-Seite (<http://www.suse.de>) mit ihrer Supportdatenbank (<http://sdb.suse.de/sdb/de/html/>) zu empfehlen, da dort die gängigsten Probleme und ihre Lösungen beschrieben werden.

Ein allgemeines Portal über Linux finden Sie unter <http://www.linux.org/> wie auch unter <http://www.linux.de/>.

Auch ein Wiki gibt es für Linux. Unter der URL <http://www.linuxwiki.de/> ist eine freie Textdatenbank zu finden, die sich mit dem Themen Linux und GNU beschäftigt.

Ein teilweise kostenpflichtiger Dienst ist <http://www.lwn.net/>. Hier erhalten Sie eine tägliche Übersicht über die neuesten Entwicklungen, Sicherheitslücken, Produkte usw.

Eine Zusammenfassung der wichtigsten Nachrichten und Schlagzeilen zu Linux aus verschiedenen Quellen finden Sie auf <http://www.linuxkp.org/>, dem Linux Knowledge Portal.

Natürlich soll auch die Seite <http://www.fibel.org/> nicht vergessen werden, wo Sie dieses Skript herunterladen können. ;-)

### 6.2.1 Newsgroups

Eine weitere Möglichkeit um an Informationen über Linux heranzukommen sind die Newsgroups. Alle Gesamtheit aller Newsgroups wird auch als *Usenet* bezeichnet. Allerdings sollten Sie zuerst alle anderen Möglichkeiten ausgeschöpft haben, bevor Sie eine Frage in einer Newsgroup stellen. D. h. konsultieren Sie zuerst Manual-Pages, TexInfo, HOWTOs, FAQs, Programm- und Paketdokumentationen und auch die FAQs, die zu den Newsgroups gehören.

Für das Lesen der Newsgroups benötigen Sie einen Newsreader wie er in Netscape oder Mozilla integriert ist oder ein separates Programm wie z. B. KNode (2.4.6). Die Suchmaschine Google (<http://www.google.de>) hat das Newsgrouparchiv von <http://www.deja.com> übernommen. Sie ist eine gute Alternative um die Newsgroups mit dem Browser durchsuchen und lesen zu können.

**comp.os.linux.announce** ist eine moderierte Newsgroup, die sich mit den neuesten Informationen zu Software-Updates, neuen Ports, Benutzergruppentreffen und kommerziellen Produkten beschäftigt.

**comp.os.linux.answers** enthält alle FAQs, HOWTOs und andere wichtige Dokumentationen. Die HOWTOs werden immer am Anfang des Monats gepostet.

**comp.os.linux.advocacy** ist das Diskussionsforum für juristischen Fragen zu Linux.

**comp.os.linux.alpha** legt seinen Themenschwerpunkt auf Fragen zu Linux für Alpha-Systeme.

**comp.os.linux.development.apps** beschäftigt sich mit der Entwicklung von Anwendungsprogrammen für Linux.

**comp.os.linux.development.system** behandelt den Bereich der Entwicklung von Linux.

**comp.os.linux.embedded** ist eine Newsgroup für Embedded-Geräte unter Linux.

**comp.os.linux.hardware** enthält Fragen zur Zusammenarbeit von Linux und der Hardware.

**comp.os.linux.m68k** legt seinen Themenschwerpunkt auf Fragen zu Linux für Apple-Prozessoren.

**comp.os.linux.networking** kümmert sich um den Netzbereich von Linux.

**comp.os.linux.portable** deckt den Bereich Linux für Notebooks ab.

**comp.os.linux.powerpc** legt seinen Themenschwerpunkt auf Fragen zu Linux für den PowerPC.

**comp.os.linux.security** ist die richtige Newsgroup, wenn es um Sicherheitsfragen zu Linux geht.

**comp.os.linux.setup** beschäftigt sich mit der Administration und Konfiguration von Linux.

**comp.os.linux.x** ist der graphischen Oberfläche X-Window gewidmet.

**comp.os.linux.misc** deckt die Themen ab, die nicht zu den anderen Newsgroups passen.

### 6.2.2 KNode

Unter KDE ist das Programm KNode zum Arbeiten mit den Newsgroups vorgesehen.

Wenn Sie KNode zum ersten Mal starten, erscheint gleich das Einstellungsfenster. Sie können es aber auch nachträglich unter *Einstellungen/KNode einrichten...* wieder aufrufen.

Als erstes tragen Sie unter Identität die persönlichen Angaben ein, wie Sie es auch für die eMail machen.

Unter *Zugang/News* tragen Sie die zu verwendenden Newsserver ein. Diese halten die Artikel der einzelnen Newsgroups vor und sorgen auch für den Abgleich der Server untereinander. Entweder haben Sie von Ihrem Provider einen Newsserver benannt bekommen oder Sie müssen sich einen freien Newsserver suchen. Für freie Newsserver gibt es im Netz mehrere Suchmaschinen. Eine Liste dieser Suchmaschinen finden Sie unter [http://directory.google.com/Top/Computers/Usenet/Public\\_News\\_Servers/](http://directory.google.com/Top/Computers/Usenet/Public_News_Servers/)

Eine dieser Suchmaschinen ist *Newzbot* (<http://www.newzbot.com/>). Geben Sie einfach den Namen der Newsgroup oder einen Teil des Namens an und Newzbot ermittelt die passenden Newsserver dafür.

Über den Button *Neu* im Server-Dialog von KNode bekommen Sie jetzt ein Dialogfenster für den neuen Newsserver. Tragen Sie einen Namen für die Anzeige und den Namen des Newsservers ein. Unter dem Reiter *Identität* können Sie individuell für jeden Newsserver sich eine Identität zulegen.

Weitere Einträge brauchen Sie zuerst nicht. Im linken Rahmen von KNode ist nun der eingetragene Newsserver aufgetaucht. Sie müssen nun eine Newsgroup von dem Server abonnieren. Dazu klicken Sie mit der rechten Maustaste auf den Servernamen und wählen den Punkt *Newsgruppen abonnieren...* aus. Nun holt KNode vom Server die vorhandenen Gruppen. Das kann bei manchen Servern sehr lange dauern. Die Gruppen werden Ihrer Hierarchie gemäß in Baumform angezeigt.

Wählen Sie nun die gewünschten Gruppen (z. B. alle Linuxgruppen) aus, indem Sie auf die schwarz umrandeten Kästchen klicken. Bestätigen Sie die Auswahl mit *OK*. Die Gruppen sind nun unter dem Newsserver-symbol erschienen. Wenn Sie nun auf eine der Gruppen klicken, dann zeigt das linke Fenster die Zahl der vorhandenen Artikel an, während eine Liste der Artikel im rechten oberen Rahmen erscheint. Klicken Sie dort auf einen Eintrag, so erscheint der gewählte Artikel im rechten unteren Rahmen.

### 6.2.3 Mailinglisten

Neben den Newsgroups gibt es auch einige Mailinglisten zum Thema Linux. In einer Mailingliste senden Sie eine Mail an die eMail-Adresse der Liste. Diese eMail wird dann an alle Mitglieder weitergeleitet. Aus Sicherheitsgründen werden viele Mailinglisten moderiert und jede eMail auf Spam und Viren überprüft.

#### **comp.os.linux.announce**

Um einen Spiegel dieses Forums zu bekommen, senden Sie eine eMail an `Linux-Announce-Request@NEWS-DIGESTS.MIT.EDU` mit dem Wort **subscribe** im Nachrichtenrumpf.

#### **majordomo@vger.rutgers.edu**

Hier werden einige Mailinglisten zum Thema Linux gehostet. Um eine Liste aller Mailinglisten zu bekommen, senden Sie einfach eine Nachricht an diese Adresse mit dem Wort **list** im Nachrichtenrumpf. Steht das Wort **help** im Nachrichtenrumpf, dann bekommen Sie eine eMail mit der Hilfe zu diesen Mailinglisten.

#### **SuSE Mailinglisten**

Auch SuSE bietet Mailinglisten an. Diese können Sie unter <http://www.suse.de/de/private/support/maillinglists/index.html> abonnieren. Stellen Sie sich aber auf einige hundert eMails am Tag ein, wenn Sie das tun.

## 6.3 Suchen nach Informationen

### 6.3.1 whereis

Das Kommando **whereis** lokalisiert die Programmdatei, Quellcodedatei und die Online-Hilfe (Manual-Pages) für ein Kommando.

**whereis** [OPTIONEN] KOMMANDO

Die Suche ist dabei auf eine limitierte Anzahl von bekannten Verzeichnissen beschränkt. Wenn die gesuchte Datei nicht gefunden wurde, wird nur der gesuchte Name zurückgegeben.

#### **Optionen**

<b>-b</b>	Suche nach Programmdatei (binary)
<b>-s</b>	Suche nach Quellcodedatei (source file)
<b>-m</b>	Suche nach Online-Hilfe (manpages)
<b>-u</b>	Suche nach anderen Dateien (unusal entries)

Voreinstellung ist `-bmsu` zur Suche nach allen möglichen Dateien.

### 6.3.2 which

Das Kommando `which` zeigt an, welches Programm mit welchem Pfad bei Eingabe eines Kommandos gestartet wird.

`which KOMMANDO`

`which` durchsucht die Verzeichnisse, die in der Umgebungsvariablen `PATH` angegeben sind, und gibt den Pfad des ersten Fundorts des gesuchten Programms an.

### 6.3.3 whatis

Zeigt die Kurzbeschreibung der Online-Hilfe (Manual-Pages) an.

`whatis [OPTIONEN] THEMA`

Innerhalb jeder Manualseite ist eine Kurzbeschreibung vorhanden. `whatis` sucht Schlüsselwort in den Kurzbeschreibungen der Indexdatenbank. Falls es eine solche nicht im Manualpfad gibt, durchsucht es die `whatis`-Datenbank, die sich in `/usr/man/man1` befindet, nach dem Schlüsselwort.

`whatis` verhält sich wie `man -f`.

### 6.3.4 mandb

Erzeugt die von `whatis` (6.3.3) benutzte Indexdatenbank für die Manualpages.

`mandb [OPTIONEN] [PFAD]`

In der Regel ist diese Datenbank die Datei `/var/cache/man/index.db`.

### 6.3.5 apropos

Der Befehl `apropos` ermöglicht eine Stichwortsuche in der Kurzbeschreibung der Hilfe.

`apropos [OPTIONEN] STICHWORT`

Die Suche mit `apropos` ist identisch zu `man -k`.

### 6.3.6 type

Das Kommando `type` zeigt an, wie ein Begriff interpretiert wird, wenn er als Kommando benutzt wird.

`type [OPTIONEN] BEGRIFFE`

#### Optionen

- `-t` Ausgabe eines einzelnen Wortes ('alias', 'keyword', 'function', 'builtin', 'file' oder '')
- `-p` Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn `-t 'file'` liefert.)
- `-a` Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn `-t 'file', 'alias' oder 'function'` liefert.)

Die Schalter `-t`, `-p` und `-a` können nicht zusammen verwendet werden.

## 6.4 Dokumentation und Support

Neben den reinen Kenntnissen über das Betriebssystem, die Software und die Hardware gehören für die Tätigkeit als Systemadministrator auch weiche Fähigkeiten (Soft-Skills) dazu. Zu diesen Soft-Skills gehört das Schreiben von Dokumentationen und der Benutzer-Support.

### 6.4.1 Dokumentationen schreiben

Eine der Aufgaben, denen sich die Leute am liebsten entziehen, ist die Dokumentation. Trotzdem ist gerade dieser Bereich der EDV besonders wichtig, weil hier Erfahrungen dauerhaft gespeichert werden können.

**Systemübersicht** Für jeden Rechner sollte eine Systemübersicht existieren. Diese sollte umfassen: Art des Festplattencontroller, Anzahl und Art der Festplatten, Partitionsübersicht, Hostname, IP-Adresse, Peripheriegeräte, BIOS-Einstellungen u. v. m.

**Problemlösungen** Wenn ein Problem aufgetreten ist, sollte man seine Lösung ausführlich dokumentieren. Wenn später der Fall wieder auftritt, braucht man nur nachzuschlagen.

**Änderungen am System** Jegliche Änderungen am System sollte auch festgehalten werden. Dies umfaßt die Änderungen an der Hardware, Konfigurationsdateien, Programmen u. s. w. Dies sollte ausreichen, um das System innerhalb kürzester Zeit wieder zu rekonstruieren.

**Dokumentation für Benutzer** Neben der Dokumentation für sich, den Administrator, sollten Sie auch in der Lage sein, Dokumentationen für die Benutzer zu schreiben. So können Sie z. B. erklären wie der Benutzer sich einloggt oder seine eMail abholen kann. Daneben sollten auch andere Administratoren in der Lage sein mit Hilfe der Dokumentation Ihr System zu warten<sup>2</sup>.

Jetzt kommt es noch darauf an, welche Art der Dokumentation man wählen soll. Es gibt zwei Arten: die elektronische Dokumentation und die Dokumentation auf Papier.

**Elektronische Dokumentation** Die Vorteile der elektronischen Dokumentation liegen auf der Hand. Sie ist leicht auf dem neuesten Stand zu halten und leicht zu durchsuchen. Kommandoausgaben und Screenshots können leicht integriert werden. Außerdem ist sie schnell zu kopieren und kann vielen zugänglich gemacht werden. Der Nachteil ist allerdings, sie steht nur zur Verfügung wenn die Maschine läuft. Auch läßt sie sich schlecht herumtragen.

**Dokumentation auf Papier** Die alte Methode der Dokumentation auf Papier hat auch heute noch ihre Vorteile. Sie ist auch zugänglich wenn das System unten ist. Dank der modernen Textverarbeitung können auch Screenshots und Kommandoausgaben eingearbeitet werden. Vor allem kann man sie überall hinnehmen, wo man Sie braucht. Der Nachteil dieser Form der Dokumentation liegt in ihrer schlechten Wartbarkeit. Eine Papierdokumentation auf dem neuesten Stand zu halten ist aufwendig. Daneben kann diese Form der Dokumentation schnell zu Chaos führen. Das Entziffern der Notizen, gerade bei handschriftlichen Dokumentationen, ist auch ein nicht zu unterschätzender Nachteil.

### 6.4.2 Benutzer-Support

Der Benutzer-Support ist eine Aufgabe, die besondere Fähigkeiten verlangt. Neben dem fachlichen Wissen sind hier vor allem der Umgang mit Menschen und die Kommunikationsfähigkeit gefragt. Sie müssen in der Lage sein Informationen so an den Benutzer weiterzugeben, daß er sie verstehen und anwenden kann.

---

<sup>2</sup>Als Administrator unersetzlich zu sein hat nämlich nicht nur Vorteile.

Obwohl der Benutzer-Support sehr aufwendig ist, sollten Sie dem Benutzer nie das Gefühl geben seine Anfrage wäre Zeitverschwendung. Diese Anfragen können auf viele Arten und Weisen erfolgen. Ihr Ziel sollte es aber immer sein den Benutzer produktiver werden zu lassen.

- **Zeit** Sie sollten alle Anfragen innerhalb eines kurzen Zeitrahmens beantworten. Wenn Sie keine Zeit haben, informieren Sie den Benutzer darüber und wann Sie wieder für ihn Zeit haben.
- **Analyse des Problems** Bevor Sie ein Problem lösen können, müssen Sie es erstmal analysieren. Fragen Sie genau nach wann, wo und wie das Problem aufgetreten ist. Es könnte ja auch kein Fehler sondern eine Anfrage für eine neue Funktion sein.
- **Dokumentation** Wenn Sie das Problem erkannt haben, konsultieren Sie Ihre Dokumentation. Wenn Sie Glück haben, ist es dort mit der Lösung verzeichnet.
- **Telefonsupport** Hier kommt es darauf an die richtigen Fragen zu stellen. Aber Vorsicht! DAUs<sup>3</sup> können ihnen den letzten Nerv abverlangen.
- **eMail-Support** Bei schriftlichen Anfragen, wie z. B. per eMail, werden nicht immer alle wichtigen Einzelheiten aufgeführt. Fragen Sie also wieder nach!
- **Sprache** Beim Kontakt mit dem Endanwender vermeiden Sie bitte den EDV-Slang. Nicht jeder ist so ein Abkürzungslexikon wie Sie. Versuchen Sie dem Benutzer alles mit möglichst einfachen Worten zu erklären.
- **Allgemeines Wissen** Solange Dokumentationen keine sicherheitsrelevanten Bereiche betreffen, sollten Sie sie den Benutzern zur Verfügung stellen. Je mehr der Benutzer weiß, desto weniger wird er Sie fragen müssen.
- **Ende** Das Problem ist erst dann gelöst, wenn der Benutzer meint, es ist gelöst. Sollte sich die Problemlösung verzögern, dann informieren Sie den Benutzer darüber und wann er mit den Informationen rechnen kann. Sie sollten dem Benutzer das Gefühl geben ein Teil auf dem Weg zur Lösung zu sein.

.....  
*Notizen:*

.....  

---

<sup>3</sup>Dümmster Anzunehmender User



## VI

Arbeitsblatt  
Dokumentation und Hilfe

## 6.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

**Teil 1**

- 1 Loggen Sie sich als Walter auf der Konsole 1 ein.
- 2 Führen Sie den Befehl `rm -rf *` aus.
- 3 Lassen Sie sich die Online-Hilfe zum Befehl `man` anzeigen und informieren Sie sich über den Befehl.
- 4 Was sagt `TexInfo` über `man` aus?
- 5 Was ist Ihnen aufgefallen?
- 6 Suchen Sie nach dem Stichwort “man” in der Hilfe (Manualpages).
- 7 Stellen Sie fest was `man` ist?
- 8 Rufen Sie die Manual-Page des Befehls `which` auf.
- 9 Lassen Sie sich den Ort der Binärdatei zu `which` zeigen.
- 10 Um welchen Typ handelt es sich bei `which`?
- 11 Nutzen Sie die oberen Informationen, um die wahre Funktion von `which` zu ermitteln.
- 12 Sorgen Sie dafür, daß `which` seine ursprüngliche Aufgabe dauerhaft wieder erledigen kann. (Nutzen Sie dafür das Ergebnis aus Aufgabe 9.)
- 13 Was ist `dir` ?
- 14 Zu welchem Typ gehört `dir` ?
- 15 Was bewirkt der Parameter `-P` beim Befehl `cp` ?
- 16 Unter welche Gruppe fällt `cp` nach `info`?
- 17 Welche Befehle fallen noch in die Gruppe und was tun sie?
- 18 Was ist `passwd` und wo befindet es sich?
- 19 Zu welchem Zweck dient der Befehl `rman`?

.....  
*Notizen:*

.....

## Teil 2

Für das Archivieren und platzsparende Packen von Dateien werden Packprogramme verwendet. Verwenden Sie die Hilfen des Linux-Betriebssystems um die folgenden Fragen zu beantworten.

- 1 Loggen Sie sich als **walter** ein oder wechseln Sie das Heimatverzeichnis von **walter**!
- 2 Führen Sie den Befehl `rm -rf *` aus.
- 3 Beschreiben Sie mit ein paar Sätzen jeweils die Funktion der folgenden Programme. (Anwendung, Anwendungsgebiet, Verhalten.)
  1. `tar`
  2. `gzip`
  3. `compress`
  4. `zcat`
  5. `zip`
- 4 Erläutern Sie die Schalter `-d`, `-l`, `-r`, `-S`, `-t` und `-v` von `gzip`.
- 5 Kopieren Sie aus `/usr/share/man/man1` die Dateien `zip.1.gz`, `zipgrep.1.gz` und `zipinfo.1.gz` in Ihr Heimatverzeichnis.
- 6 Fassen Sie den (entpackten) Inhalt der Dateien in der neuen Datei `zip.all` zusammen.
- 7 Lassen Sie sich den Inhalt der Datei `zip.1.gz` mit `less` anzeigen.
- 8 Entpacken Sie die gepackten Dateien.
- 9 Packen Sie alle `‘.1’`-Dateien zum Archiv `zip.zip` zusammen.
- 10 Packen Sie die Dateien noch einmal jeweils mit den Parametern `-1` und `-9` und vergleichen Sie die entstandenen Zip-Archive `zip1.zip` und `zip9.zip` miteinander.
- 11 Löschen Sie alle Dateien mit der Endung `.1`.
- 12 Welche Funktion hat das Programm `unzip`?
- 13 Lassen Sie sich Informationen über die Dateien im Zip-Archiv `zip.zip` anzeigen.
- 14 Entpacken Sie aus dem Archiv `zip.zip` die Datei `zipgrep.1`.
- 15 Entpacken Sie aus dem Archiv `zip1.zip` alle Dateien außer `zipgrep.1`.
- 16 Am 02.01.2001 wurde zum letzten Mal ihr Heimatverzeichnis gesichert. Wie sichern Sie nun alle neuen und geänderten Dateien in das Zip-Archiv `bup.zip`?
- 17 Wandeln Sie die Manpage für `zip` (`zip.1`) in eine HTML-Datei namens `zip.html` um.
- 18 Lassen Sie sich die HTML-Datei im Browser anzeigen.
- 19 Löschen Sie alle Zip-Archive im Heimatverzeichnis.

# Kapitel 7

## Textfilter

### 7.1 Ausgabe von ganzen Dateien

Um ganze Dateien auszugeben haben wir schon die Pager `less` (4.5.10) und `more` (4.5.9) sowie den Verbindungsbefehl `cat` (4.5.2) kennengelernt. Aber es gibt noch weitere Befehle, die sich mit der Ausgabe der ganzen Datei beschäftigen.

#### 7.1.1 `tac`

Das Kommando `tac` dreht die Reihenfolge der Datensätze (Voreinstellung: Zeilen) in Dateien um. Diese werden dann zusammengefügt und ausgegeben.

```
tac [OPTIONEN] [DATEILISTE]
```

Werden keine speziellen Datensatztrenner (Separatoren) angegeben, wird die Zeile als Datensatz interpretiert.

##### Optionen

<code>-r</code>	Separator wird als regulärer Ausdruck interpretiert
<code>-s SEP</code>	Neuen Separator eingeben (Voreinstellung: NeueZeile)

Es werden zwar die Datensätze in einer Datei umgekehrt ausgegeben, aber die Reihenfolge der angegebenen Dateien wird nicht geändert.

```
ole@enterprise:~/test> cat mann
Axel Schweiß
Bert Rahm
Carlos Calvados
ole@enterprise:~/test> cat frau
Ada Bsurdum
Betti Marsch
Claire Grube
ole@enterprise:~/test> tac mann frau
Carlos Calvados
Bert Rahm
Axel Schweiß
Claire Grube
Betti Marsch
Ada Bsurdum
```

#### 7.1.2 `nl`

Mit diesem Kommando werden der Ausgabe einer Textdatei Zeilennummern hinzugefügt.

**nl** [OPTIONEN] [DATEILISTE]

**nl** fügt für jede Zeile einer Textdatei eine Zeilennummer hinzu. Die Nummerierung beginnt bei jeder neuen Seite wieder neu.

Eine Seite gliedert sich in drei Abschnitte: header, body und footer (Kopf, Rumpf und Fuß). Die Abschnitte werden durch Zeilen eingeleitet die nur folgende Zeichenfolgen enthalten:

```
\:\:\: für den Kopf
\:\:   für den Rumpf
\:     für den Fuß
```

#### Optionen

<b>-b</b> STYLE	Wählt die Nummerierungsart für den Rumpf aus. Wird eine Zeile nicht nummeriert, so wird der Zeilenzähler nicht erhöht. Das Trennzeichen wird aber eingefügt. Als Stilarten stehen zu Verfügung
<b>-a</b>	alle Zeilen nummerieren
<b>-n</b>	keine Zeilen nummerieren (header, footer)
<b>-t</b>	alle Zeilen mit Inhalt nummerieren (body)
<b>-p</b> REGEXP	nur Zeilen, die den Ausdruck REGEXP enthalten, nummerieren
<b>-h</b>	Wie <b>-b</b> , aber für den Kopf.
<b>-f</b>	Wie <b>-b</b> , aber für den Fuß.
<b>-d</b> CD	Setzt die Abschnittsmarkierung neu (Voreinstellung \:)
<b>-i</b> N	Schrittweite der Nummerierung. Für jede Zeile wird N hinzuaddiert. (Voreinstellung: 1)
<b>-l</b> N	Zählt N aufeinanderfolgende leere Zeilen als eine (Voreinstellung: 1)
<b>-n</b> FORMAT	Setzt das Format für die Zeilennummerierung fest
	<b>ln</b> links ausgerichtet ohne führende Nullen
	<b>rn</b> rechts ausgerichtet ohne führende Nullen
	<b>rz</b> rechts ausgerichtet mit führenden Nullen
<b>-p</b>	Startet die Nummerierung am Anfang einer logischen Seite nicht neu
<b>-s</b> STRING	STRING ist die Trennzeichenkette zwischen Nummerierung und Zeile (Voreinstellung: TAB-Zeichen)
<b>-v</b> N	Startet die Nummerierung mit N (Voreinstellung: 1)
<b>-w</b> N	Anzahl der Zeichen für die Nummerierung (Voreinstellung: 6)

### 7.1.3 od

Das Kommando **od** gibt den Inhalt einer Datei in Oktal-, Dezimal-, Hexadezimal- oder ASCII-Darstellung auf der Standardausgabe aus.

**od** [OPTIONEN] [DATEILISTE]

Dieser Befehl wird meistens zur Ausgabe von Binärdateien eingesetzt.

#### Optionen

<b>-b</b>	Byteweise Ausgabe in oktaler Form
<b>-c</b>	Byteweise Ausgabe in ASCII-Format
<b>-d</b>	Dezimale Ausgabe von je zwei Byte
<b>-x</b>	Hexadezimale Ausgabe von je zwei Byte

## 7.2 Textformatierung

Die Befehle **fmt**, **pr** und **fold** formatieren den Dateitext um ihn z. B. für Ausgaben zu optimieren.

### 7.2.1 fmt

Das Kommando **fmt** erzeugt durch Trennen und Zusammenfügen Ausgabezeilen mit vorgegebener Zeilenlänge. Dabei liest **fmt** aus Dateien oder von der Standardeingabe und gibt das Ergebnis auf der Standardausgabe aus.

**fmt** [OPTIONEN] [DATEILISTE]

### Optionen

<b>-s</b>	Erlaubt nur das Zerlegen von Zeilen und nicht das Zusammenfügen von zu kurzen Zeilen
<b>-u</b>	Reduziert die Anzahl der Leerzeichen zwischen Wörtern auf ein und zwischen Sätzen auf zwei Leerzeichen
<b>-WIDTH</b>	Gibt Zeilen mit einer Breite von WIDTH Zeichen aus. (Voreinstellung: 75)
<b>-w WIDTH</b>	
<b>-p PREFIX</b>	Es werden nur Zeilen verarbeitet, die mit der Zeichenfolge PREFIX beginnen. Führende Leerzeichen werden ignoriert. Führende Leerzeichen und PREFIX werden entfernt, die Zeilen werden umgebrochen und dann werden vor jeder Zeile die Leerzeichen und der PREFIX wieder angefügt. Einsatzgebiet: Umbrechen von Kommentaren in Programmen, ohne daß der Programmcode verändert wird.

### Beispiel

Diese Kommandosequenz entfernt aus einer HTML-Datei alle überzähligen Leerzeichen.

```
fmt -u index.html > index.neu.html
```

Hier werden die Kommentare in einem Perl-Skript auf eine Zeilenlänge von 70 Zeichen umgebrochen.

```
fmt -p \# -w 70 sort.pl > sort.new.pl
```

## 7.2.2 pr

Das Kommando **pr** formatiert eine Textdatei entsprechend festgelegter Optionen.

**pr** [OPTIONEN] [DATEILISTE]

Wenn keine Optionen ausgewählt werden, werden die Dateien seitenweise formatiert. Jede Seite besteht aus 66 Textzeilen: einem fünfzeiligen Kopf, einem Textbereich und einem fünfzeiligen Fuß. Die Kopfzeile enthält Seitenzahl, Dateiname, Datum und Uhrzeit. Alternativ kann der Text auch in Spalten ausgegeben werden.

### Optionen

<b>+ANFANG[:ENDE]</b>	Ausgabe von Seite ANFANG bis Seite ENDE
<b>-SP</b>	Gibt den Text in SP Spalten aus (Voreinstellung: 1)
<b>-c</b>	Ausgabe von nichtdruckbaren Zeichen ( <b>show-control-char</b> )
<b>-d</b>	Fügt Leerzeilen ein ( <b>double-space</b> )
<b>-f</b>	Verwendet Zeilenvorschub anstatt NeueZeile
<b>-F</b>	Kopf- und Fußbereich werden auf drei Zeilen reduziert ( <b>form-feed</b> )
<b>-h HEADER</b>	Ersetzt den Dateinamen im Kopf durch die Zeichenkette HEADER
<b>-l LENGTH</b>	Setzt die Seitenlänge (Voreinstellung: 66 Zeilen) ( <b>page-length</b> )
<b>-m</b>	Fügt Dateien spaltenweise zusammen ( <b>merge</b> )
<b>-n[SEP[ZAHL]]</b>	Nummeriert die Zeilen durch SEP Trennungszeichen ZAHL Anzahl der Ziffern
<b>-N ANFANG</b>	Beginnt Nummerierung der Zeilen mit ANFANG (nur im Zusammenhang mit <b>-n</b> verwendet)
<b>-s SEP</b>	Trennzeichen für Spalten
<b>-S SEP</b>	Trennzeichenkette für Spalten
<b>-t</b>	Führt keine Seitenformatierung durch
<b>-w WIDTH</b>	Einstellung Textbreite auf WIDTH Zeichen

### Beispiele

Gibt die Datei zweispaltig von Seite 4 bis Seite 10 aus.

```
pr -2 +4:10 linux.tex
```

Gibt die Datei mit Zeilennummerierung (Trennzeichen ':', 4 Ziffern) auf dem Drucker aus.

```
pr -n:4 hallo.pl > lpr
```

### 7.2.3 fold

Das Kommando **fold** gibt den Inhalt einer Datei aus, wobei die Zeilen auf eine definierte Länge umgebrochen werden.

```
fold [OPTIONEN] [DATEILISTE]
```

Im Normalfall ermittelt **fold** die Bildschirmspalten und bricht die Zeilen auf diese Länge um. Ein Tabulatorzeichen kann mehrere Spalten umfassen.

#### Optionen

<b>-b</b>	Zählung nach Bytes und nicht nach Spalten ( <b>bytes</b> )
<b>-s</b>	Umbruch erfolgt wenn möglich nur an Leerzeichen ( <b>spaces</b> )
<b>-w WIDTH</b>	Stellt Textbreite auf WIDTH Spalten/Bytes ein ( <b>width</b> )

#### Beispiel

Die Zeilen der Datei werden auf maximal 70 Spalten umgebrochen. Der Umbruch erfolgt nur an Leerzeichen und die Ausgabe erfolgt über **less** auf der Standardausgabe.

```
fold -s -w 70 kn20000904.txt | less
```

## 7.3 Teilen von Texten

Diese Befehlsgruppen zeigt nur Teile des Textes (**head** und **tail**) oder zerlegt ihn in mehrere Teile (**split**).

### 7.3.1 head

Das Kommando **head** gibt den Anfang einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe auf.

```
head [OPTIONEN] [DATEILISTE]
```

Werden mehrere Dateien angegeben, so fügt **head** den Dateinamen in der Form

```
==> DATEINAME <==
```

als Kopf vor dem Ausgabetext ein. Der Befehl **tail** (siehe 7.3.2) arbeitet ähnlich.

#### Optionen

<b>-c B</b>	Gibt anstatt der ersten 10 Zeilen, die ersten B Zeichen aus.
<b>-n N</b>	Gibt anstatt der ersten 10 Zeilen, die ersten N Zeilen aus.
<b>-N</b>	
<b>-q</b>	Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
<b>-v</b>	Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe

#### Beispiel

Diese Kommandosequenz gibt die ersten 5 Zeilen der Datei mit dem Dateinamen als Kopf aus.

```
head -n 5 -v links.html
```

### 7.3.2 tail

Das Kommando **tail** gibt das Ende einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe auf.

```
tail [OPTIONEN] [DATEILISTE]
```

Werden mehrere Dateien angegeben, so fügt **tail** den Dateinamen in der Form

```
==> DATEINAME <==
```

als Kopf vor dem Ausgabetext ein. Der Befehl **head** (siehe 7.3.1) arbeitet ähnlich.

**Optionen**

<b>-c B</b>	Gibt anstatt der letzten 10 Zeilen, die letzten B Zeichen aus.
<b>-n N</b>	Gibt anstatt der letzten 10 Zeilen, die letzten N Zeilen aus.
<b>-N</b>	
<b>-q</b>	Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
<b>-v</b>	Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe
<b>-f</b>	Mit dieser Option überwacht <b>tail</b> kontinuierlich das Ende einer oder mehrerer Dateien. Wird an die Datei etwas angehängt, so werden die Änderungen ausgegeben. Die Überwachung wird mit STRG+C abgebrochen.

**Beispiel**

Diese Kommandosequenz gibt die letzten fünf Zeilen der Dateien ohne den Dateinamen als Kopf aus.

```
tail -n 5 -q seite1.html seite2.html seite3.html
```

Mit der folgenden Sequenz werden mehrere Dateien auf Veränderung überwacht.

```
tail -fn 15 /etc/passwd /etc/group /tmp/mylogfile
```

**7.3.3 split**

Das Kommando **split** zerlegt Dateien in mehrere kleinere Dateien.

```
split [OPTIONEN] [DATEI [PREFIX]]
```

Wird keine Datei angegeben bzw. “-” als Dateiname, dann wird von der Standardeingabe gelesen.

**split** zerlegt eine große Datei in kleinere Dateien vordefinierter Größe (Voreinstellung: 1000 Zeilen). Die dabei entstehenden Dateien beginnen mit einem Prefix (Voreinstellung: x) und werden zweistellig mit Buchstaben durchnummeriert. (xaa, xab, xac, ...).

**Optionen**

<b>-LINES</b>	Größe der Ergebnisdateien in Zeilen
<b>-l LINES</b>	
<b>-b BYTES</b>	Größe der Ergebnisdateien in Bytes
<b>b</b>	angehängt bedeutet BYTES x 512
<b>k</b>	angehängt bedeutet BYTES x 1024
<b>m</b>	angehängt bedeutet BYTES x 1048576

**7.4 Textstatistik**

Die folgenden Befehle analysieren den Text (**wc**) oder ermitteln eine Checksumme (**sum** und **chsum**) für die Fehlerkontrolle.

**7.4.1 wc**

Das Kommando **wc** zählt die Zeichen, die durch Leerzeichen getrennten Worte und die Zeilen einer Eingabe.

```
wc [OPTIONEN] [DATEILISTE]
```

**wc** gibt pro Datei die Anzahl der Zeilen, die Anzahl der Worte, die Anzahl der Zeichen und den Namen der Datei. Werden mehrere Dateien angegeben, so wird am Schluß eine Zusammenfassung aller Dateien angezeigt.

**Optionen**

<b>-c</b>	Anzahl der Zeichen ausgeben
<b>-w</b>	Anzahl der Worte ausgeben
<b>-l</b>	Anzahl der Zeilen ausgeben

**Beispiel**

Gibt die Anzahl der Worte und Zeilen in den drei Dateien an.

```
wc -lw artikel1.txt artikel2.txt artikel3.txt
```

Bestimmt die Anzahl der Dateien im Verzeichnis `/etc`. `ls /etc | wc -l`

**7.4.2 sum**

Das Kommando `sum` ermittelt eine 16-bit-Checksumme und die Größe (in 1 k-Blöcken) einer Datei.

```
sum [OPTIONEN] [DATEILISTE]
```

**7.4.3 cksum**

Das Kommando `cksum` ermittelt eine CRC-Checksumme und die Größe (in Bytes) einer Datei.

```
cksum [OPTIONEN] [DATEILISTE]
```

Diese Funktion benutzt einen stabileren Algorithmus (CRC: cyclic redundancy check) als `sum` (siehe 7.4.2).

**7.5 Sortieren**

Daten werden meistens erst dann richtig lesbar, wenn Sie sortiert werden. Die Tools `sort`, `comm` und `uniq` kümmern sich um diesen Bereich.

**7.5.1 sort**

Der Befehl `sort` durchsucht, verbindet oder vergleicht alle Zeilen einer Datei.

```
sort [OPTIONEN] [DATEILISTE]
```

`sort` kennt drei Arten der Funktion. Sortieren, Verbinden und Prüfen auf Sortierung.

**Optionen**

<code>-c</code>	Prüft, ob die Datei sortiert ist (check sort)
<code>-m</code>	Verbindet mehrere Dateien (die nicht sortiert werden, sondern als Gruppe behandelt werden) miteinander
<code>-b</code>	Führende Leerzeichen eines Schlüsselfeldes werden ignoriert
<code>-d</code>	Sortieren als Telefonverzeichnis: Nur Buchstaben, Ziffern und Leerzeichen werden berücksichtigt
<code>-f</code>	Unterscheidet nicht zwischen Groß- und Kleinbuchstaben
<code>-g</code>	Sortiert nach Zahlen anstatt lexikalisch
<code>-n</code>	Die Optionen unterscheiden sich in der Konvertierung der Zahlen zum Vergleich
<code>-o DATEI</code>	Ausgabe in die Datei <code>DATEI</code> umlenken
<code>-r</code>	Umkehrung der Sortierung
<code>-t SEP</code>	Trennzeichen für die Felder definieren
<code>-u</code>	Unterdrückung von gleichen Zeilen
<code>+POS1[-POS2]</code>	Als Schlüsselfelder werden die Felder ab <code>POS1</code> bis <code>POS2</code> (oder bis zum Zeilenende) verwendet.

**Beispiel**

Die Zählung der Spalten beginnt bei 0! Daher sortiert dieser Befehl die Datei `mai.log` nach der sechsten Spalte. Trennzeichen ist das Leerzeichen.

```
sort +5 -t " " mai.log > hits.log
```



Dieser Kommandosequenz sortiert eine Datei nach Telefonbuchmethode und unter Vernachlässigung der Groß- und Kleinschreibung.

```
sort -fd telefon.buch | less
```

### 7.5.2 comm

Das Kommando `comm` vergleicht zwei sortierte Dateien miteinander.

```
comm [OPTIONEN] DATEI1 DATEI2
```

`comm` gibt drei Spalten aus. In den ersten beiden Spalten stehen die Zeilen aus `DATEI1` und `DATEI2`, die nicht in der jeweiligen anderen Datei vorkommen. In der dritten Zeile werden die gleichen Zeilen ausgegeben.

### 7.5.3 uniq

Das Kommando `uniq` entfernt aus einer sortierten Datei die doppelten Zeilen.

```
uniq [OPTIONEN] [INPUT [OUTPUT]]
```

#### Optionen

<code>-c</code>	Gibt an, wie oft die Zeile vorkommt
<code>-i</code>	Ignoriert Groß- und Kleinschreibung
<code>-d</code>	Gibt nur die doppelten Zeilen aus
<code>-u</code>	Gibt nur die nicht doppelten Zeilen aus

#### Beispiel

Dieser Befehl extrahiert die siebente Spalte aus der Datei `mai.log`, sortiert sie, entfernt die doppelten Zeilen und gibt an, wie oft die doppelten Zeilen vorkamen.

```
cut -f 7 -d " " mai.log | sort | uniq -c | less
```

## 7.6 Zeilenoperationen

Um einen Text Zeile für Zeile zu analysieren und zu bearbeiten sind die folgenden Befehle programmiert worden.

### 7.6.1 cut

Das Kommando `cut` gibt ausgewählte Felder (Spalten) aus jeder Zeile einer Datei auf der Standardausgabe aus.

```
cut [OPTIONEN] [DATEILISTE]}
```

#### Optionen

<code>-b BYTES</code>	Zeige die angegebenen BYTES an.
<code>-f NR</code>	Zeige das Feld (die Spalte) Nummer NR (field)
<code>-d ZEICHEN</code>	ZEICHEN ist der Spaltentrenner (delimiter)
<code>-s</code>	Zeilen ohne Trennzeichen (delimiter) werden nicht ausgegeben

Das Tabulatorzeichen ist als Trennzeichen voreingestellt. Das Trennzeichen kann ein Leerzeichen oder ein anderes Sonderzeichen sein. Dann sollte es in Anführungsstriche gesetzt (oder allgemeiner maskiert) werden.

#### Beispiel

Diese Kommandosequenz gibt die Benutzernamen aus der Datei `/etc/passwd` aus.

```
cut -f 1 -d : /etc/passwd
```

Gibt die ersten 10 Zeichen jeder Zeile aus.

```
ls -l | cut -b -10
```

### 7.6.2 join

Das Kommando `join` gibt die Zeilen aus zwei Dateien aus, die identische Vergleichsfelder besitzen.

```
join [OPTIONEN] DATEI1 DATEI2
```

#### Optionen

<code>-j FIELD</code>	Vergleichsfeld aus DATEI1 und DATEI2 angeben
<code>-j1 FIELD</code>	Vergleichsfeld aus DATEI1 angeben
<code>-j2 FIELD</code>	Vergleichsfeld aus DATEI2 angeben
<code>-t CHAR</code>	Das Zeichen CHAR ist Feldtrennzeichen
<code>-i</code>	Ignoriert Groß- und Kleinschreibung

#### Beispiel

Dieser Befehl vergleicht das 2. Feld in `t1.txt` mit dem 3. Feld in `t2.txt` und gibt bei Gleichheit die passenden Zeilen aus. `join -j1 2 -j2 3 t1.txt t2.txt`

### 7.6.3 paste

Das Kommando `paste` fügt Dateien spaltenweise zusammen. Jede Datei erhält eine eigene Spalte.

```
paste [OPTIONEN] [DATEILISTE]
```

`paste` ohne Angabe von Dateien funktioniert wie `cat` (siehe 4.5.2). Ein `-` in der Dateiliste führt zu einem Lesen von der Standardeingabe.

`paste` eignet sich zum Zusammenführen von Daten aus mehreren Dateien.

#### Optionen

<code>-d ZEICHEN</code>	ZEICHEN ist das Spaltentrennzeichen
<code>-s</code>	Verknüpft die Zeilen einer Datei zu einer einzigen langen Zeile

#### Beispiel

Die Dateien werden zeilenweise zusammengefügt und die einzelnen Spalten mit einem `;` getrennt.

```
paste -d ; sonnenwind.dat magnetfeld.dat teilchen.dat > new.dat
```

Diese Kommandosequenz liest von der Standardeingabe und schreibt die Eingaben mit einem `+` getrennt in eine Zeile in die Datei `punkte.dat`.

```
paste -d + -s > punkte.dat
```

## 7.7 Suchen und Ersetzen

Um bestimmte Stellen aus Texten zu extrahieren oder zu ändern werden die Tools `grep`, `tr`, `expand`, `sed`, `diff` und `patch` verwendet.

### 7.7.1 grep

Der Befehl `grep` (*Global Regular Expression Print*) durchsucht eine Textdatei nach bestimmten Mustern und gibt die Zeilen, in denen das Muster vorkommt, auf der Standardausgabe aus.

```
grep [OPTIONS] MUSTER [DATEILISTE]
```

**Optionen**

-G	Interpretiert das MUSTER als regulären Ausdruck; Standardeinstellung. Nicht zusammen mit -F und -E verwenden
-E	Interpretiert das MUSTER als erweiterten regulären Ausdruck. Nicht zusammen mit -F und -G verwenden
-F	Interpretiert das MUSTER als einfache Zeichenkette. Nicht zusammen mit -F und -E verwenden
-c	Zeigt nur die Zeilennummern der gefundenen Zeilen an
-n	Zeigt zusätzlich zur Zeile auch die Zeilennummer an
-v	Zeigt die Zeilen an, die nicht dem MUSTER entsprechen
-f DATEINAME	Die Liste der zu bearbeitenden Dateien
-h	Unterdrückt die Ausgabe des Dateinamens bei Verwendung einer Dateiliste.
-i	Unterscheidet bei der Suche nicht nach Groß- und Kleinschreibung
-w	MUSTER wird als ganzes Wort und nicht als Teil des Wortes betrachtet
-l	Zeigt den Namen der Datei an, wenn die Zeile darin gefunden wurde
-s	Fehlermeldungen unterdrücken
-r	Durchsucht auch die Unterverzeichnisse

**Beispiele**

Durchsucht die Datei `kuno.txt` nach Zeilen mit dem Wort Bruno.

```
grep Bruno kuno.txt
```

Durchsucht alle Dateien im Verzeichnis nach der Zeichenkette 'midnight' ohne Berücksichtigung der Groß- und Kleinschreibung und gibt die Namen der Dateien aus, die die Zeichenkette enthalten.

```
grep -il midnight *
```

**7.7.2 egrep**

Das Kommando `egrep` entspricht dem Befehl `grep -E`. Es verwendet erweiterte reguläre Ausdrücke.

**7.7.3 fgrep**

Das Kommando `fgrep` entspricht dem Befehl `grep -F`. Es benutzt als Suchmuster nur reine Zeichenketten.

**7.7.4 tr**

Das Kommando `tr` sendet die Standardeingabe zur Standardausgabe, wobei es mehrere Operationen auf die Daten ausführt. Mögliche Operationen sind:

- Verändern von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.
- Löschen von Zeichen, die wiederholt werden.
- Löschen von Zeichen.
- Löschen von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.

```
tr [OPTIONEN] ZEICHENKETTE1 [ZEICHENKETTE2]
```

**Optionen**

-d	Löscht die angegebenen Zeichen
-s	Löscht doppelt vorkommende Zeichen

**Beispiele**

So wandelt `tr` mit dem Befehl

```
cat stundenplan.txt | tr m M > stundenplan.neu
```

jedes Vorkommen des Buchstabens 'm' in den Buchstaben 'M' um.

Das folgende Kommando löscht die Zeichen ‘m’ und ‘y’.

```
cat etwas.txt | tr -d my
```

Doppelt vorkommende Zeichen werden mit dem Schalter -s gelöscht.

```
tr -s le < harry.txt
```

Dabei wird dann aus der ‘Allee’ ganz schnell ‘Ale’.

Um den einfachen ROT13 Verschlüsselungsalgorithmus zu verwenden, reicht die folgende Sequenz.

```
cat harry.txt | tr ' [A-M] [N-Z] [a-m] [n-z] ' ' [N-Z] [A-M] [n-z] [a-m] '
```

oder

```
cat harry.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
```

Dabei wird der erste Buchstabe mit dem 13. Buchstaben, der zweite mit dem 14. Buchstaben, u. s. w. getauscht. Wendet man ROT13 wieder auf verschlüsselten Text an, so erhält man den Originaltext.

### 7.7.5 expand und unexpand

Das Kommando **expand** liest aus Dateien oder von der Standardeingabe, wandelt die Tabzeichen in Leerzeichen und gibt das Ergebnis auf der Standardausgabe aus. Wird nichts anderes angegeben, dann wird ein Tabulator durch acht Leerzeichen ersetzt.

```
expand [OPTIONEN] [DATEILISTE]
```

#### Optionen

-TAB1[,TAB2[,...]] TABx ist die Anzahl der Leerzeichen, durch die das jeweilige Tabulatorzeichen ersetzt werden soll.

TAB1[,TAB2[,...]]

-i Tabulatoren ohne vorhergehende Leerzeichen werden nicht konvertiert.

Wird nur für das erste Tabulatorzeichen ein Wert angegeben, so gilt dieser Wert für alle Tabulatoren. Werden zwei oder mehr Werte eingegeben, so gelten Sie für die jeweiligen Tabulatoren. Tabulatoren, für die kein Wert angegeben wurde, werden durch einfache Leerzeichen ersetzt.

Die Gegenoperation können Sie mit dem Befehl **unexpand** durchführen, der Leerzeichen wieder versucht in Tabulatoren umzuwandeln.

#### Beispiel

Diese Kommandosequenz ersetzt den ersten Tabulator jeder Zeile der Datei `helloworld.java` durch 6 Leerzeichen, den zweiten durch 8 Leerzeichen und alle folgenden Tabulatoren durch ein Leerzeichen. Das Ergebnis wird in die Datei `helloworld.txt` geschrieben. `expand -6,8 helloworld.java > helloworld.txt`

### 7.7.6 sed

**sed** ist die Abkürzung für **S**treaming **E**ditor. Eigentlich ist das Kommando kein Editor sondern ein Textfilter, der bestimmte Zeichenkombinationen sucht und ersetzt. Die Verarbeitung eines Streams oder Datenstroms bedeutet, daß von der Standardeingabe (`stdin`) Daten angenommen werden und auf die Standardausgabe (`stdout`) wieder ausgegeben werden. Als Datenquelle kann neben der Standardeingabe auch eine Datei fungieren.

```
sed [OPTIONEN] [KOMMANDO] [DATEILISTE]
sed [OPTIONEN] [-f SCRIPTFILE] [DATEILISTE]
```

Es werden grundsätzlich zwei Möglichkeiten genutzt um **sed** aufzurufen. Im ersten Fall wird das **KOMMANDO** auf der Befehlszeile eingegeben und auf die angegebenen Dateien angewendet. Im zweiten Fall stehen die Anweisungen in einer externen Skriptdatei. Diese wird abgearbeitet und die darin enthaltenen Kommandos auf den Inhalt der Dateien angewendet. Sollte keine Datei angegeben sein, so liest **sed** von der Standardeingabe. Dabei wird jede gelesene Zeile mit den **sed**-Kommandos bearbeitet und in einen Puffer geschrieben. Dessen Inhalt wird dann zum Schluß auf der Standardausgabe ausgegeben.

**Optionen**

<b>-V</b>	Versionsnummer
<b>-h</b>	Hilfe
<b>-e SCRIPT</b>	Zusätzliche Skriptanweisung zur Bearbeitung
<b>-f SCRIPTFILE</b>	Datei, die die Skriptbefehle enthält
<b>-n</b>	Ausgabe erfolgt nur bei Benutzung des 'p' Kommandos

Jeder Befehl kann durch einen Bereich, für den er gültig ist, eingeschränkt werden. Die Bereichsangabe erfolgt im Format

VON

oder

VON, BIS.

Dabei können die Werte für VON und BIS

- die Zeilennummern,
- das Zeichen \$ für das Ende der Datei oder
- ein Suchmuster, begrenzt durch das Zeichen /, z. B. /unix/, sein.

**Ersetzen** Die folgenden Sequenzen weisen `sed` an, eine Zeichenkette durch eine andere zu ersetzen.

```
sed s/SUCHEN/ERSETZEN/
sed s/SUCHEN/ERSETZEN/g
sed s/SUCHEN/ERSETZEN/p
sed -n s/SUCHEN/ERSETZEN/gp
sed s3/SUCHEN/ERSETZEN/
```

Die Zeichenkette zwischen dem ersten und dem zweiten Schrägstrich gibt das Suchmuster an, während die Zeichenkette zwischen dem zweiten und dem dritten Schrägstrich den Ersetzungstext enthält.

Um das Suchmuster effektiver zu gestalten, können reguläre Ausdrücke<sup>1</sup> verwendet werden. Diese bestehen normalerweise aus zwei Komponenten: Die Angabe, nach welchem Zeichen gesucht wird, und die Angabe, wie oft das Zeichen auftreten darf.

Die hinten angestellten Optionen verändern die Arbeitsweise von `sed`. Dabei bewirkt `g`, daß alle Zeichenketten in der Zeile durch die neue Zeichenkette ersetzt werden (Sonst nur die erste Zeichenkette). Das `p` führt dazu, daß die Zeilen ausgegeben werden, in denen eine Ersetzung erfolgte. Sind beide Optionen gesetzt, so wird bei einer mehrfachen Ersetzung in einer Zeile die Zeile auch mehrfach angezeigt. Verhindert wird dies durch den Schalter `-n`. Das `s3` sorgt dafür, daß das dritte Vorkommen von `SUCHEN` in der Zeile ersetzt wird.

**Suchen** Die Sequenz

/SUCHEN/p

im Zusammenhang mit dem Schalter `-n` bewirkt, daß nur die Zeilen, die das Suchmuster enthalten, ausgegeben werden.

**Zeilen löschen** Wenn der Schalter `-n` nicht gesetzt ist, werden alle Zeilen nach der Bearbeitung ausgegeben. Um zu verhindern, daß bestimmten Zeilen ausgegeben werden, wird die Sequenz

/SUCHEN/d

verwendet.

**Dateien einfügen** Durch die Sequenz

/SUCHEN/r DATEINAME

werden alle Zeilen, die das Suchmuster enthalten, durch den Inhalt der Datei `DATEINAME` ersetzt.

---

<sup>1</sup>Weitere Informationen bekommen Sie unter `man 7 regex`.

**Auswahl in Datei schreiben** Durch die Sequenz

/SUCHEN/w DATEINAME

werden alle Zeilen, die das Suchmuster enthalten, in die Datei DATEINAME geschrieben.

### Beispiele

Ersetzt jedes Wort "UNIX" durch das Wort "Linux" in der Datei `einleitung.tex`.

```
sed 's/UNIX/Linux/g' einleitung.tex
```

Ersetzt jedes Wort "paragraph" durch das Wort "subsubsection" in der Datei `linux.tex` und schreibt das Ergebnis in die Datei `linuxneu.tex`.

```
sed 's/paragraph/subsubsection/g' linux.tex > linuxneu.tex
```

Einen interessanten Effekt hat das Zeichen '&' in der Ersetzungszeichenkette. Die gesuchte Zeichenkette wird nicht ersetzt, sondern die Ersetzungszeichenkette wird hinten angefügt.

```
sed 's/</&lt;/g' index.html
```

Damit die schließende spitze Klammer ersetzt wird, muß das kaufmännische Und auskommentiert werden.

```
sed 's/</\&lt;/g' index.html
```

Ersetzt alle Zeichenfolgen "man" durch "frau" in den Zeilen 1 bis 3 in der Datei `einleitung.tex`. Dabei werden nur die Zeilen ausgegeben, in denen die Änderung erfolgte.

```
sed -n '1,3s/man/frau/p' einleitung.tex
```

Gibt alle Zeilen aus, die entweder "man" oder "frau" enthalten.

```
sed -n -e '/man/p' -e '/frau/p' einleitung.tex
```

Das folgende Kommando löscht alle Zeilen, die die Zahl '0' enthalten.

```
sed '/0/d' einleitung.tex
```

Alle Zeilen, die die Zeichenkette "include" enthalten, werden durch den Inhalt der Datei `include.txt` ersetzt.

```
sed '/include/r include.txt' einleitung.txt
```

Alle Zeilen, die die Zeichenkette "section" enthalten, werden in die Datei `inhalt.txt` geschrieben.

```
sed '/section/w inhalt.txt' einleitung.tex
```

## 7.7.7 diff

Das Kommando `diff` vergleicht zwei Textdateien miteinander. Als Ergebnis wird eine Liste aller Zeilen ausgegeben, die voneinander abweichen. Dabei erkennt das Programm auch eingefügte Zeilen und arbeitet danach reibungslos weiter. Im Gegensatz zu `comm` (siehe 7.5.2) kann `diff` auch auf unsortierte Dateien angewendet werden. Der Befehl wird hauptsächlich dazu verwendet die Abweichungen zwischen zwei Versionen eines Programmlistings rasch zu dokumentieren.

```
diff [OPTIONEN] DATEI1 DATEI2
```

### Optionen

<b>-b</b>	Betrachtet mehrfache Leerzeichen und Leerzeilen als einfache Leerzeichen bzw. Leerzeilen
<b>-c</b>	Ausgabe als 'Context Diffs'
<b>-e</b>	Ausgabe als Skript für den Befehl <code>ed</code>
<b>-r</b>	Vergleicht den Inhalt zweier Verzeichnisbäume
<b>-u</b>	Ausgabe als 'Unified Diffs'
<b>-w</b>	Ignoriert Leerzeichen und Leerzeilen ganz

Die Ausgabe von `diff` besteht aus sogenannten 'hunks', was man frei als 'Stücke' übersetzen kann. Ein Hunk besteht aus bis zu vier Teilen: einer Informationszeile, dem alten Text, einer Trennzeile und dem neuen Text. Beim Einfügen und Löschen von Zeilen fehlt die alte oder neue Textinformation und die dann überflüssige Trennzeile fällt auch weg.

In der Informationszeile steht, an welcher Stelle in der jeweiligen Datei die Änderung stattgefunden hat und welche Operationen nötig sind um die erste Eingabedatei `DATEI1` in die zweite Eingabedatei `DATEI2` zu verwandeln. Dabei stehen die Buchstaben 'a', 'c' und 'd' für 'add' (Zeilen hinzufügen), 'change' (Zeilen verändern) und 'delete' (Zeilen löschen). Die Zahlen bzw. Zahlenbereiche links und rechts von den Buchstaben geben die betroffenen Zeilen an.

Alle Hunks zusammengekommen bilden den Patch, der die erste Datei in die zweite Datei verwandelt. Es geht auch umgekehrt, da Patches dieser Art symmetrisch sind und rückwärts angewendet werden können ('reverse patching');

### Beispiel

Um zwei Perlscripte miteinander zu vergleichen, ohne mehrfache Leerzeichen zu berücksichtigen, gibt man das folgende Kommando ein.

```
diff -b div.pl.old div.pl
```

Soll die Ausgabe für den Editor `ed` verarbeitbar sein, dann muß das Kommando lauten:

```
diff -be div.pl.old div.pl
```

Die Anwendung von `diff` fürs Patchen von Programmen beschreibt der Abschnitt 7.7.9.

## 7.7.8 patch

Da das Patchen von Programmen mit `ed` einige Probleme aufweist hat Larry Wall das Programm `patch` entwickelt, das später vom GNU-Projekt weiterentwickelt wurde. Es ist in der Lage, alle gängigen `diff`-Formate zu verstehen.

```
patch [OPTIONEN] [ORIGINALDATEI [PATCHDATEI]]
```

### Optionen

<code>-c</code>	Interpretiere den Patch als Context Diff
<code>-e</code>	Interpretiere den Patch als ed-Skript
<code>-u</code>	Interpretiere den Patch als Unified Diff

## 7.7.9 Praxisbeispiel: Patchen von Programmen

Das Patchen von Programmen hat viele Vorteile. So muß z. B. nicht das ganze Programm heruntergeladen oder per eMail versandt werden. Außerdem enthält der Patch nur die Änderungen im Programm. So muß ein Mitautor nicht erst mühselig im Programm die Änderungen suchen, sondern sie werden ihm komprimiert übergeben.

Als Beispiel schauen wird uns zwei Versionen eines Perl-Skripts an.

### Listing 7.1 *div.pl*

```
1: #!/usr/bin/perl
2:
3: # Dieses Programm teilt zwei Zahlen durcheinander
4:
5: # Eingabe
6: print "Zähler: ";
7: $z = <STDIN>;
8: chomp($z);
9: print "Nenner: ";
10: $n = <STDIN>;
11: chomp($n);
12:
13: # Verarbeitung
14: $erg = int($z / $n);
15: $rest = "Rest " . ($z % $n);
16:
```

```

17: # Ausgabe
18: print "$z / $n = $erg $rest\n";
19:
20: exit(0);

```

### Listing 7.2 *div2.pl*

```

1: #!/usr/bin/perl
2:
3: # Dieses Programm teilt zwei Zahlen durcheinander
4: # Version 2
5:
6: # Eingabe
7: print "Zähler: ";
8: $z = <STDIN>;
9: chomp($z);
10: print "Nenner: ";
11: $n = <STDIN>;
12: chomp($n);
13:
14: # Verarbeitung
15: # Verarbeitung nur, wenn der Nenner ungleich Null ist
16: if ($n != 0) {
17:     $erg = int($z / $n);
18:     $rest = "Rest ". ($z % $n);
19: } else {
20:     $erg = "unendlich";
21:     $rest = "";
22: }
23:
24: # Ausgabe
25: print "$z / $n = $erg $rest\n";
26:
27: exit(0);

```

diff liefert uns nun die Unterschiede der beiden Skripte.

```

tapico@defiant:~/perl > diff div.pl div2.pl
3a4
> # Version 2
14,15c15,22
< $erg = int($z / $n);
< $rest = "Rest ". ($z % $n);
---
> # Verarbeitung nur, wenn der Nenner ungleich Null ist
> if ($n != 0) {
>     $erg = int($z / $n);
>     $rest = "Rest ". ($z % $n);
> } else {
>     $erg = "unendlich";
>     $rest = "";
> }

```

Manuell können mit diesen Informationen nun die Änderungen an der Datei *div.pl* durchgeführt werden.

- Füge hinter der dritten Zeile als vierte Zeile den Kommentar ein.
- Lösche die Zeilen 14 und 15 und füge als Zeilen 15 bis 22 die angegebenen Zeilen ein.



Da dies bei größeren Projekten doch problematisch ist, kann auch ein Skript für den Editor `ed` erstellt werden.

```
tapico@defiant:~/perl > diff -e div.pl div2.pl
14,15c
# Verarbeitung nur, wenn der Nenner ungleich Null ist
if ($n != 0) {
    $erg = int($z / $n);
    $rest = "Rest ". ($z % $n);
} else {
    $erg = "unendlich";
    $rest = "";
}
.
3a
# Version 2
.
```

Wenn Sie die Ausgabe als Datei gespeichert haben, kann nun jemand anders mit dem Editor `ed` die alte Datei (`div.pl`) mit dem Patch (hier die Datei `div.patch`) bearbeiten und daraus die neue Datei (`div2.pl`) erstellen.

```
barclay@enterprise:~/projekt > (cat div.patch; echo w div2.pl) | ed - div.pl
```

Dies setzt aber voraus, daß an der alten Datei nichts geändert worden ist. Daher gibt es zwei andere Formate, die mehr Sicherheit beim Patchen versprechen.

**Context Diff**

```

tapico@defiant:~/perl > diff -c div.pl div2.pl
*** div.pl Sun Sep 23 14:36:48 2001
--- div2.pl Sun Sep 23 14:36:12 2001
*****
*** 1,6 ***
--- 1,7 ----
    #!/usr/bin/perl

    # Dieses Programm teilt zwei Zahlen durcheinander
+ # Version 2

    # Eingabe
    print "Zähler: ";
*****
*** 11,18 ***
    chomp($n);

    # Verarbeitung
! $erg = int($z / $n);
! $rest = "Rest ". ($z % $n);

    # Ausgabe
    print "$z / $n = $erg $rest\n";
--- 12,25 ----
    chomp($n);

    # Verarbeitung
! # Verarbeitung nur, wenn der Nenner ungleich Null ist
! if ($n != 0) {
!     $erg = int($z / $n);
!     $rest = "Rest ". ($z % $n);
! } else {
!     $erg = "unendlich";
!     $rest = "";
! }

    # Ausgabe
    print "$z / $n = $erg $rest\n";

```

**Unified Diff**

```

tapico@defiant:~/perl > diff -u div.pl div2.pl
--- div.pl Sun Sep 23 14:36:48 2001
+++ div2.pl Sun Sep 23 14:36:12 2001
@@ -1,6 +1,7 @@
    #!/usr/bin/perl

    # Dieses Programm teilt zwei Zahlen durcheinander
+# Version 2

    # Eingabe
    print "Zähler: ";
@@ -11,8 +12,14 @@
    chomp($n);

```

```

# Verarbeitung
-$erg = int($z / $n);
-$rest = "Rest ". ($z % $n);
+# Verarbeitung nur, wenn der Nenner ungleich Null ist
+if ($n != 0) {
+  $erg = int($z / $n);
+  $rest = "Rest ". ($z % $n);
+} else {
+  $erg = "unendlich";
+  $rest = "";
+}

# Ausgabe
print "$z / $n = $erg $rest\n";

```

Diese beiden Formate, wie auch die anderen, können mit dem Befehl `patch` die Veränderung durchführen.

```
barclay@enterprise:~/projekt > patch < div.c.patch
```

Die im Patch angegebene Datei (`div.pl`) ist nun verändert worden.

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## VII

Arbeitsblatt  
Textverarbeitung und Textfilter

## 7.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

- 1 Loggen Sie sich als Walter ein!
- 2 Führen Sie den Befehl `rm -rf *` aus.
- 3 Kopieren Sie aus dem Verzeichnis `/usr/share/man/man1` die Datei `less.1.gz` ins Heimatverzeichnis.
- 4 Erzeugen Sie aus der Datei `less.1.gz` die HTML-Datei `less.html`.
- 5 Lassen Sie sich den Inhalt der Datei `less.html` seitenweise anzeigen!
- 6 Erzeugen Sie mit dem `vi` eine Textdatei, die mindestens 3 Zeilenvorschübe beinhaltet (4 Zeilen lang)! Name der Datei: `funny` !
- 7 Kopieren Sie die Datei `/usr/share/man/man1/more.1.gz` in Ihr Heimatverzeichnis und machen Sie wie in Aufgabe 4 eine HTML-Datei daraus.
- 8 Lassen Sie sich Inhalt von `more.html` und `funny` in Ihrem Heimatverzeichnis nacheinander seitenweise auf dem Bildschirm anzeigen, dazu ist nur ein Befehl einzugeben!
- 9 Erstellen Sie mit einem einzigen Befehl aus den Dateien `more.html` und `funny` eine neue Datei namens `hummer`.
- 10 Lassen Sie sich den Inhalt von `hummer` anzeigen.
- 11 Löschen Sie die Dateien `funny` und `hummer` in Ihrem Heimatverzeichnis!
- 12 Wie lauten die letzten (10) Zeilen der Datei `less.html`.
- 13 Lassen Sie sich den Inhalt der Datei `less.html` in oktaler Form anzeigen!
- 14 Kopieren Sie alle Dateien, die mit 'mc' beginnen, aus dem Verzeichnis `/usr/share/man/man1/` in Ihr Heimatverzeichnis.
- 15 Wandeln Sie wie in Aufgabe 4 die kopierten Dateien in HTML-Dateien.
- 16 Lassen Sie sich die ersten 5 Zeilen der Datei `mcop.html` anzeigen.
- 17 Wieviele Wörter und Zeilen enthält die Datei `more.html`?
- 18 Welche Datei im Heimatverzeichnis enthält die Zeichenkette 'Midnight'?
- 19 Schauen Sie sich die Datei `/etc/passwd` an.
- 20 Wieviele Zeilen enthält sie?
- 21 Lassen Sie sich die ersten 20 Zeilen der Datei `/etc/passwd` numeriert anzeigen.
- 22 Geben Sie die erste Spalte der Datei `/etc/passwd` aus.
- 23 Geben Sie sortiert die erste Spalte der Datei `/etc/passwd` aus.
- 24 Geben Sie sortiert die erste Spalte der Datei `/etc/passwd` aus und speichern Sie das Ergebnis in der Datei `~/passwd.sort`.

## VII

Arbeitsblatt  
Textverarbeitung und Textfilter

## 7.2

- 25** Schreiben Sie den Inhalt der Datei `/etc/passwd` in umgekehrter Zeilenfolge in die Datei `~/passwd.reverse`.
- 26** Vergleichen Sie die CRC-Checksummen von `/etc/passwd`, `~/passwd.sort` und `~/passwd.reverse`.
- 27** Lassen Sie sich die Logdatei für Warnmeldungen des Systems `/var/log/warn` anzeigen. Welche Bedeutung haben wohl die einzelnen Spalten?
- 28** Lassen Sie sich alle Zeilen von heute ausgeben.
- 29** Lassen Sie sich alle Zeilen vom letzten Arbeitstag ausgeben.
- 30** Wieviele Warnmeldungen gab es?
- 31** Geben Sie die beiden ersten Spalten aus.
- 32** Geben Sie die beiden ersten Spalten aus, entfernen Sie alle doppelten Zeilen.
- 33** Lassen Sie sich anzeigen, wieviele Warnmeldungen es pro Tag gab.
- 34** Sortieren Sie diese Liste absteigend nach der Häufigkeit.
- 35** Verfassen Sie mit dem Editor `vi` eine Botschaft an Ihren Nachbarn und speichern Sie diese in der Datei `gutenachricht.txt`.
- 36** Verschlüsseln Sie den Inhalt der Datei mit dem ROT13-Verfahren und speichern Sie das Ergebnis in der Datei `gutenachricht.krypto`.
- 37** Senden Sie Ihrem Nachbarn die Datei zu. (eMail, FTP, Turnschuhnetz etc.)
- 38** Entschlüsseln Sie die Nachricht Ihres Nachbarn.
- 39** Was bedeutet der Satz "Qvr Nhstnora fvaq sregvt"?

.....  
*Notizen:*

.....

# Kapitel 8

## Benutzerverwaltung

### 8.1 Benutzer

Nach der Installation von Linux existiert nur ein Benutzerkonto. Dieses Benutzerkonto ist *root*, das Konto für den Superuser. Daher muß der Administrator (Superuser) für die Arbeit weitere Konten anlegen.

#### 8.1.1 Der Superuser root

Das Konto *root* wird nicht umsonst als Superuser bezeichnet. Der Benutzer unter diesem Konto darf im System alles. Er hat vollen Zugriff auf alle Verzeichnisse, Dateien und Geräte im System. Damit kann er auch alles löschen. Diese Aktion kann aber zu erheblichen Schäden am System führen, somit sollte man als Superuser sehr vorsichtig handeln.

Der Superuser *root* besitzt die UID 0 (*User IDentification*) und wird wie alle anderen Benutzer in der Datei */etc/passwd* definiert. Da das System die Benutzer nicht nach ihrem Namen identifiziert, sondern nach ihrer UID, kann der Superuser auch einen anderen Lognamen bekommen. Da Linux wie das Christentum nur einen Gott kennt, nämlich den Benutzer mit der UID 0, ist es ratsam für diesen Benutzer weitere Namen anzulegen. Wenn Sie nämlich das Paßwort für *root* vergessen, wünsche ich Ihnen viel Spaß ;-).

Um die Sicherheit ihres Systems zu erhöhen, sollten Sie mehrere Benutzer anlegen, die spezielle Aufgaben erledigen können. Diese Benutzer erhalten spezielle eingeschränkte Rechte für die verschiedenen administrativen Aufgaben. Wie das geht, wird in Abschnitt 8.3.6 beschrieben.

Für die alltäglichen Aufgaben sollten Sie sich nicht als *root* einloggen, da dies ein nicht zu unterschätzendes Risiko darstellt. Im normalen Betrieb sollten Sie als einfacher Benutzer arbeiten. Nur für administrative Aufgaben loggen Sie sich als *root* ein und erledigen die Aufgaben, um danach als einfacher Benutzer weiterzuarbeiten. Um diesen zeitaufwendigen Prozeß des Aus- und Einloggens abzukürzen, wurde der Befehl *su* entwickelt.

#### 8.1.2 su

Das Kommando *su* erlaubt einen Wechsel der Benutzeridentität bzw. UID während einer Sitzung.

```
su [OPTIONEN] [BENUTZER]
```

Die einfache Eingabe von *su* ohne einen Benutzernamen bewirkt einen Wechsel zur Superuseridentität nach der Eingabe des Superuserkennworts.

Der Wechsel der Identität wird durch das Starten einer neuen Shell realisiert. Im Gegensatz zum Einloggen, bei dem auch eine Shell gestartet wird, bleibt beim Wechsel der Identität mit *su* die Umgebung des alten Benutzers erhalten. Nur die Umgebungsvariablen *HOME* und *SHELL* werden auf die entsprechenden Werte des neuen Benutzers geändert.

Spalte	Bezeichnung	Inhalt
1	Benutzername	Einzigartig um den Benutzer bei Einloggen zu identifizieren
2	Kennwort	Das verschlüsselte Kennwort (heute in <code>/etc/shadow</code> )
3	UID	Eindeutige Nummer zur Identifizierung des Benutzers
4	GID	Eindeutige Nummer zur Identifizierung der Gruppe
5	Kommentar	Voller Name des Benutzers und andere Angaben
6	Heimatverzeichnis	Arbeitsverzeichnis nach dem Einloggen
7	Login-Kommando	Das Kommando wird nach dem Einloggen ausgeführt; meist eine Shell

Tabelle 8.1: Die Felder der Datei `/etc/passwd`.

Jeder Benutzer kann seine Identität wechseln, solange er das Kennwort des neuen Benutzers kennt. Nur der Superuser kann seine Identität wechseln ohne ein Kennwort eingeben zu müssen.

Mit dem Befehl `exit` oder drücken der Tastenkombination `<STRG>+<D>` kehren Sie zur alten Identität zurück.

#### Optionen

- `-, -l` Die Konfigurationsdateien des Benutzers werden ausgeführt
- `-c KOMMANDO` Führt das angegebene Kommando unter dem neuen Benutzer aus

#### Beispiele

`su`

mit anschließender Kennworteingabe läßt den Benutzer nun als Superuser arbeiten.

`su tapico`

mit anschließender Kennworteingabe läßt den Benutzer nun unter der UID des Benutzers `tapico` arbeiten. Der Superuser benötigt keine Kennworteingabe um die Identität zu wechseln.

`su - tapico`

führt im Gegensatz zum obigen Beispiel dazu, daß die Konfigurationsdateien von `tapico` mit ausgeführt werden. Daher ist z. B. das Heimatverzeichnis von `tapico` jetzt das aktuelle Arbeitsverzeichnis.

## 8.2 Das Benutzerkonto

Als erstes nach der Installation des Systems, selbst wenn Sie der einzige Benutzer sind, sollten Sie sich einen eigenen einfachen Benutzer zulegen. Dazu müssen Sie für jeden anderen Benutzer ein eigenes Konto einrichten. Dies erhöht die Sicherheit im System, da so jeder Benutzer individuelle Privilegien erhält. Außerdem ist eine Verfolgung der Benutzeraktivitäten möglich.

### 8.2.1 `/etc/passwd`

Alle Informationen über das Benutzerkonto werden in der Datei `/etc/passwd` gespeichert. Es enthält u. a. den Loginnamen, die UID, das verschlüsselte Kennwort<sup>1</sup> und den vollen Namen. Ein Übersicht über den Inhalt gibt Tabelle 8.1.

Die Datei `/etc/passwd` gehört `root` mit dem Recht `rw-`. Alle anderen eingeschlossen die Gruppe haben nur das Lese-Recht.

Die Datei könnte z. B. folgende Einträge enthalten:

```
root:x:0:0:Herr und Meister:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
lp:x:4:7:lp daemon:/var/spool/lpd:/bin/bash
news:x:9:13:News system:/etc/news:/bin/bash
```

<sup>1</sup>Falls das Shadow-Paßwort-System installiert ist, was inzwischen bei den meisten Distributionen Standard ist, werden die Kennwörter nicht in `/etc/passwd` sondern in der getrennten Datei `/etc/shadow` gespeichert.



```
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
at:x:25:25:./var/spool/atjobs:/bin/bash
wwwrun:x:30:65534:Daemon user for apache:/tmp:/bin/bash
squid:x:31:65534:WWW proxy squid:/var/squid:/bin/bash
ftp:x:40:2:ftp account:/usr/local/ftp:/bin/bash
firewall:x:41:31:firewall account:/tmp:/bin/false
named:x:44:44:Nameserver Daemon:/var/named:/bin/bash
tapico:x:501:100:Thomas Prizzi:/home/tapico:/bin/bash
luke:x:502:100:Lukas Himmelsgeher:/home/luke:/bin/bash
lori:x:503:100:Lori Kalmar:/home/lori:/bin/bash
grayson:*:504:100:Grayson Death Carlyle:/home/grayson:/bin/bash
nobody:x:65534:65534:nobody:/tmp:/bin/bash
```

Die Felder werden durch Doppelpunkte (:) voneinander getrennt. Daher kann auch ein Feld, wie bei dem Benutzer *at* zu sehen, leer bleiben. Dieses Format wird bei fast allen UNIX-Systemen verwendet.

### Feld: Benutzername

Das erste Feld ist der Benutzername. Er muß einmalig sein, da er für die Identifikation beim Einlogvorgang verwendet wird. Der Benutzername kann bis zu 8 Zeichen lang sein. Er kann aus Buchstaben, Ziffern, dem Unterstrich, dem Minuszeichen und anderen speziellen Zeichen bestehen. Bei Verwendung der nichtalphanumerischen Zeichen kann es zu Problemen mit Programmen kommen, die den Benutzernamen verwenden. Auch hier wird zwischen Groß- und Kleinschreibung unterschieden. So handelt es sich bei *tapico* und *Tapico* um zwei verschiedene Benutzer. Natürlich sollte die Anlage von zwei Benutzern, die sich nur in Groß- und Kleinschreibung unterscheiden, vermieden werden.

Wenn Sie viele Benutzer in ihrem System einrichten wollen, ist es wichtig sich eine Strategie zur Namensvergabe zu überlegen. In den meisten Fällen wird es eine Kombination aus Vorname und Nachname sein. So kann der Loginname der Benutzerin Angela Mahnke z. B. lauten: *mahnke*, *amahnke*, *aMahnke*, *A\_Mahnke* oder *angelam*. Beim Benutzer Ole Vanhoefer wären es z. B.: *vanhoefer*, *ovanhoefer*, *oVanhoefer*, *o\_Vanhoefer* oder *olevan*.

### Feld: Kennwort

Das nächste Feld im Datensatz ist das durchs Betriebssystem verschlüsselte Kennwort. Sollte der Eintrag beschädigt werden, so kann sich der Benutzer nicht mehr einloggen und der Administrator muß für den Benutzer ein neues Kennwort setzen.

Um einen Benutzer am Einloggen zu hindern, kann der Superuser hier ein Asterisk (\*) einsetzen. Um einen Benutzer nur vorübergehend auszusperrern, weil er im Urlaub ist oder den Superuser geärgert hat, ist es üblich an den Anfang des verschlüsselten Kennworts ein Asterisk zu setzen. Dabei ist darauf zu achten, daß die restlichen Zeichen des verschlüsselten Kennworts nicht verändert werden.

Ein leeres Feld an dieser Stelle bewirkt, daß Sie sich unter dem Benutzernamen ohne Eingabe eines Kennworts einloggen können. Solche Benutzerkonten sollten aber stark in ihren Rechten eingeschränkt werden, um nicht hier eine gewaltige Sicherheitslücke zu öffnen. Die Kennwortlänge beträgt bei den heutigen Distributionen zwischen fünf und acht Zeichen. Dies kann aber individuell konfiguriert werden.

### Feld: UID

Das dritte Feld eines Eintrags in der */etc/passwd* enthält die UID (*User IDentification*). Dies ist eine Zahl, mit der das System den Benutzer identifiziert. So wird beim Starten eines Prozesses durch einen Benutzer dessen UID dem Prozess zugeordnet.

Der Zahlenbereich kann den Typ eines Kontos schon einer bestimmten Gruppe zuordnen. UIDs unter 100 sind Systemkonten zugeordnet. Die UID 0 ist dem Superuser *root* vorbehalten. Normale Benutzer bekommen eine UID größer als 100 zugewiesen. Dabei ist die Grenze von Distribution zu Distribution unterschiedlich. Bei S.u.S.E. liegt die Grenze bei 500. In der Datei */etc/login.defs* (Abschnitt 8.2.6) kann der Wertebereich der UID mit den Parametern *UID\_MIN* und *UID\_MAX* genau festgelegt werden.

Benutzerkonto	Einsatzgebiet
root	Superuser des Systems
daemon	Besitzt und setzt die Berechtigungen auf die Systemprozesse
bin	Besitzer der ausführbaren Dateien
sys	Besitzer der ausführbaren Dateien
adm	Besitzer der Log-Dateien
uucp	Benutzt für den UUCP-Zugriff

Tabelle 8.2: Besonder Benutzerkonten

**Feld: GID**

Die GID (*Group IDentification*) steht in Feld Nummer vier. Sie wird wie die UID vom System zur Verwaltung benutzt. Die Gruppen dienen zur besseren Organisation der Benutzer und deren Privilegien. Die hier angegebene Gruppe wird als Login-Gruppe des Benutzers bezeichnet. Die Gruppen werden in der Datei `/etc/group` (Abschnitt 8.4.1) definiert.

Wie auch bei der UID gibt die GID durch ihren Zahlenbereich schon Aufschluß über die Verwendung. Der Bereich 0 bis 49 ist den Systemgruppen vorbehalten. Die Nummern ab 50 bekommen die Benutzergruppen. Auch dies ist abhängig von der Distribution und wird ebenfalls in der Datei `/etc/login.defs` (Abschnitt 8.2.6) festgelegt. Als Standardgruppe für Benutzer werden meistens die Gruppe *group* oder *users* angelegt. Diesen Gruppen werden die Benutzer beim Erstellen standardmäßig zugeordnet.

**Feld: Kommentar**

Das Kommentarfeld kann leer gelassen werden. Normalerweise enthält es den vollen Namen des Benutzers. Es kann aber auch Abteilung, Telefonnummer, eMail oder sonstige Informationen enthalten. Der Befehl `finger` benutzt z. B. die hier enthaltenen Informationen. Auch nutzen einige eMail-Programme diesen Eintrag für den Absender.

**Feld: Heimatverzeichnis**

Dieser Eintrag bestimmt das Heimatverzeichnis des Benutzers. Hier findet er sich nach dem Einloggen wieder. Sollte der Eintrag ein nichtexistierendes Verzeichnis beinhalten, so schlägt der Einlogvorgang fehl.

**Feld: Login-Kommando**

In dem letzten Feld steht der Befehl, der nach dem Einloggen zuerst ausgeführt werden soll. Grundsätzlich handelt es sich hier um den Aufruf der zu benutzenden Shell. Dadurch kann der Benutzer in der Nutzung des System eingeschränkt werden. Ein leerer Eintrag führt dazu, daß die als Standard vorgegebene Shell benutzt wird.

Der Eintrag kann über den Befehl `chsh` (Abschnitt 4.1.1) geändert werden.

**8.2.2 Besondere Benutzerkonten**

Wenn Sie in die Datei `/etc/passwd` schauen, sehen Sie im Bereich der niedrigen UIDs die Systemkonten. Diese Konten sind virtuelle Benutzer unter dessen Identifikation die Systemprozesse laufen. Dies dient vor allen Dingen zur Sicherheit des Systems. Früher liefen die meisten Prozesse unter dem Konto *root*. Dies führte aber bei Sicherheitslücken dazu, daß der Angreifer unter der Identität von *root* arbeiten konnte. Wenn ihm heute ein Durchbruch gelingt, so kann er nur als einer der Systembenutzer arbeiten, die natürlich nur mit eingeschränkten Rechten versehen sind. Eine Liste solcher Systembenutzer zeigt Tabelle 8.2.

Neben diesen in der Tabelle aufgeführten Systembenutzer gibt es solche wie *mail*, *ftp*, *gopher*, *news*, *wwwroot* u.s.w. Denken Sie daran. Ändern Sie nie die Einträge für diese Besitzer, denn das kann verheerende Folgen für ihr System haben.

Vielleicht ist Ihnen auch aufgefallen, daß die Accounts im Kennwortfeld ein Asterisk (\*) stehen haben. Es ist also nicht möglich sich interaktiv als Systembenutzer einloggen.

### 8.2.3 Benutzer am Einloggen hindern

Manchmal braucht man ein Benutzerkonto für eine bestimmte Aufgabe. Trotzdem soll sich keiner interaktiv mit diesem Konto auf dem Rechner einloggen können. Dies sind zum einen Systembenutzerkonten, zum anderen sind dies Konto, für die FTP- oder POP3-Zugriff möglich ist, aber eben kein direkter Shell-Login.

#### **/bin/false**

Wenn Sie sich die Datei `/etc/passwd` genauer anschauen, werden Sie bei vielen Systemkonten den Befehl `/bin/false` als Login-Shell eingetragen finden. Eigentlich ist `false` keine Shell, sondern ein Befehl der nichts tut und dann sich auch noch mit einem Statuscode beendet, der einen Fehler signalisiert. Das Ergebnis ist simpel. Der Benutzer loggt sich ein und sieht sofort wieder den Login-Prompt.

Bei Verwendung des Befehls `su` können Sie die Fehlermeldung besser sehen.

```
ole@enterprise:~> su martin
Password:
This account is currently not available.
ole@enterprise:~>
```

Es kann aber auch in Abhängigkeit von der Distribution vorkommen, daß keine Fehlermeldung erscheint.

#### **/bin/true**

Das Gegenteil des Befehls `false` ist der Befehl `true`. Dieser Befehl tut auch nichts, liefert aber immer den Fehlercode 0 für eine erfolgreiche Ausführung zurück. Dies können Sie im folgenden Beispiel sehen.

```
ole@enterprise:~> false; echo $?
1
ole@enterprise:~> true; echo $?
0
```

Obwohl es ungewöhnlich ist, können Sie auch diesen Befehl als Loginshell eintragen. Er verhindert ebenfalls erfolgreich das Einloggen.

#### **/sbin/nologin**

Die Shell `/sbin/nologin` hat nur eine Aufgabe. Einem sich einloggenden Benutzer freundlich zu sagen, daß er sich nicht einloggen darf und ihn dann eiskalt aus dem System rauszuwerfen. Die übliche Nachricht wird durch den Inhalt der Datei `/etc/nologin.txt` ersetzt.

Beim normalen Login-Prompt auf der Konsole bekommen Sie nichts davon mit, da der neue Login-Prompt vorher den Bildschirm löscht. Versuchen Sie sich dagegen mit dem Befehl `su` einzuloggen, können Sie die Fehlermeldung gut sehen.

```
ole@enterprise:~> su martin
Password:
Hallo Fremder,
Du darfst Dich hier nicht einloggen.

Gruß Root
ole@enterprise:~>
```

Ohne die Datei `/etc/nologin.txt` ist die Fehlermeldung lange nicht so schön.

```
ole@enterprise:~> su martin
Password:
This account is currently not available.
ole@enterprise:~>
```

### 8.2.4 passwd

Um ein Kennwort zu ändern wird der Befehl `passwd` verwendet. Der Benutzer kann nur sein eigenes Kennwort ändern, während der allmächtige Superuser alle Kennwörter ändern kann.

```
passwd [BENUTZER]
```

Wenn ein Benutzer sein Kennwort ändern will, dann braucht er natürlich keinen Benutzernamen anzugeben. Er muß aber mit seinem alten Kennwort den Vorgang bestätigen und danach zweimal das neue Kennwort, was nicht auf dem Bildschirm angezeigt wird, eingeben. Dies soll eine Falscheingabe des neuen Kennworts durch Vertippen verhindern.

Die Richtlinien für die Kennwörter werden in der Datei `/etc/login.defs` (Abschnitt 8.2.6) festgelegt. Das kann die Mindestlänge des Kennworts, das Auslaufdatum des Kennworts und die Warnung davor sein. Außerdem finden Sie hier den Eintrag für das Verzeichnis der Mailboxen.

Bei der Kennwortvergabe ist darauf zu achten, daß gebräuchliche Worte aus Lexika (Brutal Force Attack) und Worte aus dem Umfeld der Person (scharf nachdenken) relativ leicht geknackt werden können. Die Kombination aus zwei Worten, Groß- und Kleinschreibung sowie Buchstaben und Zahlen, die mindestens eine Länge von sechs Zeichen aufweisen sollten, sind ein idealer Schutz.

### 8.2.5 chpasswd

Das Tool `chpasswd` erlaubt eine skriptähnliche Änderung von mehreren Kennwörtern gleichzeitig.

```
chpasswd [OPTIONEN]
```

Dabei liest `chpasswd` eine Datei von der Standardeingabe. Die Datei enthält pro Zeile den Benutzernamen und das Kennwort getrennt durch einen Doppelpunkt (`tapico:zorro67`). Das Kennwort steht dabei im Klartext. Ist das Kennwort bereits verschlüsselt, so muß der Schalter `-e` mitgegeben werden.

#### Beispiele

```
cat newpass.txt | chpasswd
```

Diese Kommandosequenz ändert die Kennwörter wie in der Datei `newpass.txt` beschrieben. Die Kennwörter stehen im Klartext in der Datei.

### 8.2.6 /etc/login.defs

Die Datei `/etc/login.defs` enthält die Einstellungen für den Login-Vorgang. Alle Einträge sind optional. Fehlen diese, so gelten sie als nicht gesetzt. Wie in den meisten Konfigurationsskripten üblich werden Kommentarzeilen (beginnen mit einem '#') und leere Zeilen ignoriert.

Hier ein Beispiel für die `/etc/login.defs`. Wie in guten Installationen üblich ergeben sich die Bedeutungen der Einträge aus den vorangestellten Kommentaren.

```
# Delay in seconds before being allowed another attempt after a login failure
#
FAIL_DELAY 3

# Enable logging and display of /var/log/faillog login failure info.
#
FAILLOG_ENAB yes
```

```

# Enable display of unknown usernames when login failures are recorded.
#
LOG_UNKFAIL_ENAB no

# Enable logging and display of /var/log/lastlog login time info.
#
LASTLOG_ENAB yes

# Enable additional checks upon password changes.
#
OBSCURE_CHECKS_ENAB yes

# If defined, ":" delimited list of "message of the day" files to
# be displayed upon login.
#
MOTD_FILE /etc/motd
#MOTD_FILE /etc/motd:/usr/lib/news/news-motd

# If defined, file which maps tty line to TERM environment parameter.
# Each line of the file is in a format something like "vt100  tty01".
#
TTYTYPE_FILE /etc/ttytype

# If defined, login failures will be logged here in a utmp format.
# last, when invoked as lastb, will read /var/log/btmp, so...
#
#BTMP_FILE /var/log/btmp

# If defined, file which inhibits all the usual chatter during the login
# sequence. If a full pathname, then hushed mode will be enabled if the
# user's name or shell are found in the file. If not a full pathname, then
# hushed mode will be enabled if the file exists in the user's home directory.
#
#HUSHLOGIN_FILE .hushlogin
HUSHLOGIN_FILE /etc/hushlogins

# The default PATH settings.
#
ENV_PATH /usr/local/bin:/usr/bin:/bin

# The default PATH settings for root:
#
ENV_ROOTPATH /sbin:/bin:/usr/sbin:/usr/bin

# Terminal permissions
#
# TTYGROUP Login tty will be assigned this group ownership.
# TTYPERM Login tty will be set to this permission.
#
# If you have a "write" program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP to the group number and
# TTYPERM to 0620. Otherwise leave TTYGROUP commented out and assign
# TTYPERM to either 622 or 600.
#
TTYGROUP tty
TTYPERM 0620

# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable password length.

```

```
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

# If compiled with cracklib support, where are the dictionaries
#
CRACKLIB_DICTPATH /usr/lib/cracklib_dict

# Min/max values for automatic uid selection in useradd
#
UID_MIN 100
UID_MAX 60000

# Min/max values for automatic gid selection in groupadd
#
GID_MIN 100
GID_MAX 60000

# Max number of login retries if password is bad
#
LOGIN_RETRIES 3

# Max time in seconds for login
#
LOGIN_TIMEOUT 60

# Maximum number of attempts to change password if rejected (too easy)
#
PASS_CHANGE_TRIES 3

# Warn about weak passwords (but still allow them) if you are root.
#
PASS_ALWAYS_WARN yes

# Number of significant characters in the password for crypt().
# Default is 8, don't change unless your crypt() is better.
# Ignored if the "md5" option is given to the pam_pwcheck module.
#
PASS_MAX_LEN 8

# Require password before chfn/chsh can make any changes.
#
CHFN_AUTH yes

# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone). If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
#
CHFN_RESTRICT rwh

# Should login be allowed if we can't cd to the home directory?
# Default is yes.
#
DEFAULT_HOME yes
```

## 8.3 Einrichten eines Benutzers

Damit ein Benutzer sich einloggen kann, benötigt er mindestens einen Benutzernamen und ein Heimatverzeichnis. Alles andere, sogar das Kennwort, werden aus anderen Gründen eingerichtet.

Es gibt mehrere Möglichkeiten einen Benutzer anzulegen. Sie werden hier die direkte Methode und den Befehl `useradd` kennenlernen. Viele Distributionen bieten weitere, meist komfortablere Tools an.

### 8.3.1 Bearbeiten der Datei `/etc/passwd`

Der ursprünglichste Weg (Hardcore-Administration) einen Benutzer einzurichten ist der direkte Eingriff in die Datei `/etc/passwd`. Bevor Sie aber dies tun, sollten Sie sich auf jeden Fall eine Sicherungskopie der Datei anlegen, denn ohne eine funktionstüchtige `passwd`-Datei können Sie sich selbst als *root* nicht mehr einloggen.

Um jetzt einen neuen Benutzer anzulegen, müssen Sie nur eine neue Zeile mit den entsprechenden Informationen hinzufügen. Denken Sie bitte daran, daß der Editor die Datei in reinem ASCII-Code (z. B. `vi` oder `joe`) abspeichert. Sie tragen ins erste Feld den Lognamen des neuen Benutzers ein. Das zweite Feld mit dem Kennwort lassen Sie leer, da es hier in verschlüsselter Form stehen muß. Für das dritte Feld müssen Sie sich erst eine Übersicht über die vorhandenen UIDs verschaffen, damit Sie dann die nächste freie UID hier eintragen können. Ein solcher Eintrag könnte dann lauten:

```
tapico::523:100:Thomas Prizzi:/home/tapico:/bin/bash
```

Nach dem Eintrag und dem Abspeichern der Datei müssen Sie den Befehl `passwd` verwenden um das Kennwort des Benutzers festzulegen.

Jetzt sollte das Heimatverzeichnis des Benutzers angelegt und die Konfigurationsdateien kopiert werden. Auf diesen Vorgang gehen wir später genauer ein. Denken Sie daran den Benutzer als neuen Besitzer und seine Gruppe als Gruppe für das Heimatverzeichnis einzutragen. Dazu können Sie das `chown`-Kommando (Abschnitt 9.2.1) benutzen.

### 8.3.2 `useradd`

Ein komfortablerer und weniger fehleranfälliger Weg einen Benutzer einzurichten ist der Befehl `useradd`.

`useradd [OPTIONEN] [BENUTZER]`

#### Optionen

<code>-D</code>	Zeigt die Standardeinstellung des Systems
<code>-u UID</code>	Erlaubt eine UID vorzugeben
<code>-g GID</code>	Erlaubt eine GID vorzugeben
<code>-o</code>	Erlaubt die mehrfache Verwendung von UIDs
<code>-m</code>	Das Heimatverzeichnis wird automatisch erstellt
<code>-d VERZEICHNIS</code>	Angegebenes Verzeichnis als Heimatverzeichnis verwenden
<code>-k VERZEICHNIS</code>	Angegebenes Verzeichnis als Skeleton-Verzeichnis verwenden
<code>-c KOMMENTAR</code>	Inhalt des Kommentarfeldes
<code>-s SHELL</code>	Name der LOGIN-Shell
<code>-e MM/DD/YY</code>	Verfallsdatum des Kontos
<code>-f TAGE</code>	Zeitraum zwischen Auslaufen des Paßworts und schließen des Kontos

Im Normalfall benutzt `useradd` die nächste freie UID und weist dem neuen Benutzer die aktuelle Shell als Login-Kommando des Kontos zu. Ein Heimatverzeichnis wird zwar in der `/etc/passwd` eingetragen, aber nicht erstellt. Erst der Schalter `-m` bewirkt das Anlegen des Heimatverzeichnis. Dies ist normalerweise `/home/BENUTZERNAME`. Durch den Schalter `-d` kann aber auch ein anderes Verzeichnis angegeben werden. Wenn das Heimatverzeichnis angelegt wird, werden automatisch die Dateien aus dem Skeleton-Verzeichnis (meistens `/etc/skel`) in das neue Heimatverzeichnis kopiert. Der Befehl `useradd` sorgt auch dafür, daß für alle angelegten Verzeichnisse und Dateien die richtigen Besitzer, Gruppen und Rechte eingetragen werden.

Mit dem Schalter `-e MM/DD/YY` kann ein Verfallsdatum für das Konto angegeben werden. Nach dem Auslaufen des Kennworts kann ein Konto auch gesperrt werden. Der Schalter `-f TAGE` gibt dabei an, wann dies erfolgen soll. Eine Null steht dabei für sofort, während `-1` dafür steht, daß das Konto nicht gesperrt wird, wenn das Kennwort abgelaufen ist.

## Beispiele

Die Verwendung des Schalters `-D` ohne Angabe eines Benutzernamens zeigt die Standardeinstellungen des Systems.

```
root@defiant:/home > useradd -D
GROUP=100
HOME=/home
INACTIVE=0
EXPIRE=10000
SHELL=/bin/bash
SKEL=/etc/skel
```

```
useradd -m luke
```

legt den Benutzer *luke* zusammen mit seinem Heimatverzeichnis an.

### 8.3.3 usermod

Nachdem Sie einen Benutzer angelegt haben, können Sie mit dem Befehl `usermod` alle Felder in der `/etc/passwd` ändern.

```
usermod [OPTIONEN] BENUTZER
```

Der Benutzername kann nicht geändert werden, wenn der Benutzer gerade eingeloggt ist. Wird der Benutzername geändert, dann sollten Sie auch das Heimatverzeichnis entsprechend umbenennen. Dies ist allerdings nicht notwendig, denn solange das Heimatverzeichnis existiert, kann sich der Benutzer einloggen.

#### Optionen

<code>-l</code> BENUTZERNAME	Neuer Benutzername
<code>-u</code> UID	Neue UID
<code>-g</code> GID	Neue GID
<code>-c</code> "KOMMENTAR"	Neuer Kommentareintrag
<code>-d</code> VERZEICHNIS	Neues Heimatverzeichnis
<code>-s</code> SHELL	Neue Login-Shell
<code>-o</code>	Erlaubt mehrfache Nutzung von UIDs
<code>-m</code>	Neues Heimatverzeichnis wird angelegt und alte Dateien rüberkopiert

Wird die UID des Benutzers geändert, dann erhalten auch alle Dateien des Benutzers in seinem Heimatverzeichnis die neue UID zugeordnet. Dateien außerhalb des Heimatverzeichnisses werden nicht geändert. Auch die `crontab`- und `at`-Jobs müssen manuell geändert werden.

### 8.3.4 Entfernen eines Benutzers

Wenn ein Benutzer aus dem System entfernt werden soll, muß nicht nur sein Eintrag in der `/etc/passwd` gelöscht werden, sondern auch seine Hinterlassenschaften entweder gelöscht oder anderen Benutzern übertragen werden. Bei einer manuellen Entfernung sollten Sie folgenden Schritte ausführen:

1. Entfernen Sie die Zeile des Benutzers aus der `/etc/passwd`.
2. Löschen Sie das Heimatverzeichnis des Benutzers.
3. Suchen Sie nach Dateien des Benutzers, die sich außerhalb seines Heimatverzeichnisses befinden und löschen Sie diese.
4. Löschen Sie die Mail-Dateien des Benutzers und jedes Mail-Alias
5. Entfernen Sie jeden Job (`crontab` oder `at`), die der Benutzer eingerichtet hat.

Ein leichter Weg einen Benutzer zu löschen ist der Befehl `userdel`.



### 8.3.5 userdel

Der Befehl `userdel` dient dazu einen Benutzer zu entfernen.

```
userdel [OPTIONEN] BENUTZER
```

#### Optionen

`-r` Löscht auch das Heimatverzeichnis des Benutzers

Die Ausführung des Befehls ohne Optionen löscht den Benutzer aus der Datei `/etc/passwd` und aus anderen Systemdateien, entfernt aber nicht sein Heimatverzeichnis. Dies hingegen bewirkt der Schalter `-r`. Dateien des Benutzers, die sich außerhalb seines Heimatverzeichnis befinden, werden nicht gelöscht.

### 8.3.6 Einrichten eines zweiten Superusers

Wie schon am Anfang des Kapitels erwähnt, ist nur der Benutzer mit der UID 0 Gott im System und darf alles. Wenn Sie nun das Paßwort von dem Benutzer vergessen oder unabsichtlich verstellen, was manchen meiner Teilnehmer schon mal passiert ist, ist das System nicht mehr administrierbar. Für diesen Fall sollte man sich einen zweiten (oder auch dritten) Superuser einrichten. Wie bekommt aber dieser nun die Rechte des Superusers. Das Einstellen in die Gruppe `root` bringt da nicht viel.

Die Lösung liegt auf der Hand. Dem zweiten Superuser wird auch die UID 0 zugewiesen. D. h. es gibt zwei Benutzer mit der gleichen UID. Eigentlich gibt es nur einen Benutzer, der sich aber unter zwei verschiedenen Namen einloggen kann. Eigentlich dürfen keine zwei Benutzer die gleiche UID verwenden. Wenn Sie mit `useradd` den zweiten Superuser einrichten wollen, dann wird dies mit einer Fehlermeldung quittiert. Erst die Verwendung des Schalters `-o` ermöglicht eine doppelte Vergabe von UIDs.

```
root@defiant:~> useradd -u 0 -g 0 -m -c "2. Superuser" admin
useradd: uid 0 is not unique
root@defiant:~> useradd -u 0 -g 0 -o -m -c "2. Superuser" admin
root@defiant:~> passwd admin
New password:
Re-enter new password:
Password changed
```

In Ihrer Firma sollten zumindest drei Superuserkonten vorhanden sein. `root` oder das umbenannte Konto für Sie, eins für Ihren Stellvertreter und eins, dessen Daten in einem versiegelten Umschlag im Tresor aufbewahrt werden. Denn wenn Ihrem Stellvertreter etwas zustößt, während Sie gerade in Ihrem Urlaub das australische Outback erforschen, ist das Firmennetz ohne zugänglichem Superuserkonto nur noch schwer administrierbar.

## 8.4 Gruppen

Gruppen werden dazu verwendet um die Benutzer und ihre Rechte zu organisieren. Gruppen werden erstellt um bestimmten Benutzern Rechte auf ausgewählte Verzeichnisse oder Geräte zu erteilen.

Wenn ein Benutzer auf eine Datei zugreifen will, dann prüft das Betriebssystem, ob er der Besitzer der Datei ist. Ist er es nicht, dann prüft es, ob er der Datei zugewiesenen Gruppe angehört. Trifft das zu, dann erhält er den hier definierten Zugang zur Datei. Gehört der Benutzer auch nicht zur Gruppe, dann werden ihm nur die Privilegien eingeräumt, die auch alle anderen bekommen.

Die Benutzer können Mitglieder vieler Gruppen sein. Es ist aber nur immer eine Gruppe mit dem Benutzer assoziiert. Diese Gruppe ist in der Datei `/etc/passwd` festgelegt.

Um eine Gruppe einzurichten kann entweder die Datei `/etc/group` (Abschnitt 8.4.1) um einen Eintrag erweitert oder der Befehl `groupadd` (Abschnitt 8.4.7) verwendet werden. Änderungen der Gruppeneigenschaften werden entweder in der Konfigurationsdatei selber ausgeführt oder über den Befehl `groupmod` (Abschnitt 8.4.9). Das Entfernen der Gruppe erfolgt entweder wieder über die Konfigurationsdatei oder über den Befehl `groupdel` (Abschnitt 8.4.11). Bei der Verwendung des Shadow-Paßwort-Systems (Abschnitt 8.5) sollten möglichst nur diese Tools verwendet werden.

### 8.4.1 /etc/group

Die Datei `/etc/group` ist die Konfigurationsdatei für die Gruppe und ihre Mitglieder. Sie ist ähnlich wie die Datei `/etc/passwd` aufgebaut.

```
root::0:root
bin::1:root,bin,daemon
daemon::2:
sys::3:
tty::5:
wwwadmin::8:
mail::12:
news::13:news
uucp::14:uucp,fax,root,ole
shadow::15:root
dialout::16:root,ole,tapico,grayson
at::25:at
firewall::31:
public::32:
named::44:named
users::100:
greydeath::101:root,grayson
nogroup::65534:root
```

Die durch Doppelpunkte abgetrennten Felder sind von links nach rechts Gruppenname, Gruppenkennwort, GID und die Mitgliederliste.

#### Feld: Gruppenname

In dem ersten Feld steht der Gruppenname. Er muß einmalig sein und darf bis zu acht Buchstaben lang sein. Für die Wahl der Zeichen gelten die gleichen Regeln wie bei den Benutzernamen.

#### Feld: Gruppenkennwort

Das zweite Feld ist für das Gruppenkennwort vorgesehen. Dieses Kennwort ist für den Wechsel in eine Gruppe gedacht. Normalerweise ist dieses Feld leer oder enthält nur einen Asterisk (\*).

#### Feld: GID

Die Group Identification wird vom System benutzt und muß einmalig sein. Die Identifikation der Gruppe vom System her erfolgt nur über diese Nummer.

#### Feld: Mitgliederliste

Im letzten Feld werden die Mitglieder (Benutzer) der Gruppe eingetragen. Dabei werden die einzelnen Einträge durch Semikola getrennt.

### 8.4.2 Besondere Gruppen

Vom System werden einige Systemgruppen wie z. B. *bin*, *mail*, *sys* oder *adm* angelegt. Sie sollten auf keinem Fall einen Benutzer zu den Gruppen hinzufügen, da er sonst weitreichenden Rechte bekommen würde. Diese Gruppen sollten nur Systemkonten (Abschnitt 8.2.2) als Mitglieder enthalten. Eine Übersicht über die wichtigsten Systemgruppen liefert Tabelle 8.3.

Systemgruppe	Einsatzgebiet
<i>root</i>	Ist den meisten Systemdateien zugeordnet.
<i>daemon</i>	Ist der Mail, den Druckern und anderer Systemsoftware zugeordnet.
<i>kmem</i>	Verwaltet den direkten Zugriff zum Kernelspeicher.
<i>sys</i>	Ist einigen Systemdateien, Swap-Dateien und Speicherabbildungen zugeordnet.
<i>nobody</i>	Ist der Software zugeordnet, die keine speziellen Rechte braucht.
<i>tty</i>	Ist den Gerätedateien zugeordnet, die die Terminals kontrollieren.

Tabelle 8.3: Systemgruppen

### 8.4.3 id

Um herauszufinden welche GID einem Benutzer zugeordnet ist, wird der Befehl `id` verwendet.

`id [OPTIONEN] [BENUTZERNAME]`

#### Optionen

<code>-g</code>	Gibt nur die Haupt-GID aus
<code>-G</code>	Zeigt nur die GIDs aller Gruppen
<code>-n</code>	Gibt die Namen aus (für <code>-u</code> , <code>-g</code> und <code>-G</code> )
<code>-r</code>	Gibt die reale ID aus anstatt der effektiven ID (für <code>-u</code> , <code>-g</code> und <code>-G</code> )
<code>-u</code>	Gibt nur die UID aus

Der Befehl zeigt die UID und die mit dem Benutzer assoziierte GID an. Zusätzlich werden auch die Gruppen angezeigt, in denen der Benutzer Mitglied ist.

#### Beispiel

```
tapico@defiant:~ > id
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath)
tapico@defiant:~ > groups
users greydeath defiant
```

### 8.4.4 groups

Der Befehl `groups` entspricht dem Befehl `id -Gn` und zeigt die Namen der Gruppen an, in denen der Benutzer Mitglied ist.

`groups [BENUTZER]`

### 8.4.5 newgrp

Um die assoziierte Gruppe zu wechseln, wird der Befehl `newgrp` benutzt.

`newgrp [NEUEGRUPPE]`

Die Angabe des Befehls ohne einen Gruppennamen, läßt die GID zur in der `/etc/passwd` eingetragenen Gruppe zurückwechseln.

#### Beispiel

```
tapico@defiant:~ > id
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath),102(defiant)
tapico@defiant:~ > newgrp defiant
tapico@defiant:~ > id
uid=501(tapico) gid=102(defiant) Gruppen=100(users),101(greydeath),102(defiant)
```

### 8.4.6 Anlegen von Gruppen

Sie können Gruppen auf zwei Arten anlegen. Entweder Sie editieren die Datei `/etc/group` direkt oder Sie benutzen ein Tool wie `groupadd`. Im Zusammenhang mit dem Shadow-Paßwort-System ist die Benutzung des Tools vorzuziehen, da sonst auch manuelle Eintragungen in der Datei `/etc/gshadow` nötig sind.

#### 8.4.7 groupadd

Der Befehl `groupadd` legt eine neue Gruppe an.

```
groupadd [OPTIONEN] GRUPPENNAME
```

Die Gruppe wird mit einem leeren Gruppenkennwort eingerichtet und enthält noch keine Mitglieder. Die können mit dem Befehl `gpasswd` hinzugefügt werden.

##### Optionen

<code>-g GID</code>	GID für die neue Gruppe vorgeben
<code>-o</code>	Mehrfache Nutzung der GID zulassen
<code>-r</code>	Systemgruppen einrichten (nicht S.u.S.E)

#### 8.4.8 gpasswd

Das Tool `gpasswd` wird zur Mitglieder- und Kennwortverwaltung der Gruppen benutzt.

```
gpasswd [OPTIONEN] GRUPPE
```

##### Optionen

<code>-R</code>	Verhindert die Benutzung des Befehls <code>newgrp</code> für diese Gruppe
<code>-a BENUTZER</code>	Fügt neues Mitglied hinzu
<code>-d BENUTZER</code>	Entfernt eine Mitglied aus der Gruppe
<code>-A BENUTZER</code>	Legt das Mitglied als Gruppenadministrator fest
<code>-r</code>	Entfernt das Kennwort der Gruppe

##### Beispiel

```
root@defiant:/ > groupadd defiant
root@defiant:/ > gpasswd -a tapico defiant
Adding user tapico to group defiant
root@defiant:/ > id tapico
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath),102(defiant)
root@defiant:/ > grep defiant /etc/group
defiant:x:102:tapico
```

#### 8.4.9 groupmod

Um die Eigenschaften einer Gruppe zu ändern wird der Befehl `groupmod` verwendet.

```
groupmod [OPTIONEN] GRUPPE
```

##### Optionen

<code>-n NAME</code>	Neuer Gruppenname
<code>-g GID</code>	Neue GID
<code>-o</code>	Mehrfache Nutzung einer GID zulassen

Bei der Änderung der GID sollte die `/etc/passwd` überprüft werden, ob dort nicht die alte GID einem Benutzer zugeordnet wurde. Diese muß natürlich dann auf die neue GID geändert werden.

### 8.4.10 Löschen einer Gruppe

Um eine Gruppe zu löschen, entfernen Sie einfach die entsprechende Zeile aus der `/etc/group`. Davor sollten Sie noch folgenden Schritte durchführen.

1. Prüfen Sie die Datei `/etc/passwd` und vergewissern Sie sich, daß kein Benutzer die GID der gelöschten Datei als Login-Gruppe verwendet. Sollte dies der Fall sein, so ändern Sie die GID auf eine existierende Gruppe. Tun Sie das nicht, dann kann der Benutzer sich nicht mehr einloggen.
2. Prüfen Sie im Dateisystem nach, ob Verzeichnisse oder Dateien mit der gelöschten GID assoziiert sind. Dies könnte den Zugriff auf die Dateien erschweren oder sogar gänzlich unmöglich machen.
3. Prüfen Sie, ob nicht die Gruppe eine Systemgruppe ist. Das Entfernen einer solchen Gruppe könnte zu erheblichen Schäden am System führen.

### 8.4.11 groupdel

Einfacher geht das Löschen einer Gruppe mit dem Befehl `groupdel`.

```
groupdel GRUPPE
```

Sollten Sie diesen Befehl auf eine Gruppe anwenden, die einem Benutzer als Login-Gruppe zugewiesen ist, wird der Befehl nicht ausgeführt.

Auch wie beim manuellen Löschen müssen Sie sich selber davon überzeugen, daß keine Verzeichnisse oder Dateien noch mit der gelöschten GID assoziiert sind.

## 8.5 Shadow-Paßwort-System

Die Speicherung des Kennwort, wenn auch in einer verschlüsselten Form, in einer von allen lesbaren Datei ist ein Sicherheitsrisiko. Der Einsatz von Entschlüsselungsprogrammen ermöglicht es relativ leicht an die Kennwörter zu gelangen. Deswegen ist der Einsatz des Shadow-Paßwort-Systems bei den heutigen Linux-Distributionen üblich. In der `/etc/passwd` steht für das Kennwort nur noch ein 'x'. Die Kennwörter stehen in der nur für `root` zugänglichen `/etc/shadow`.

### 8.5.1 /etc/shadow

Die `/etc/shadow` enthält nicht nur den Namen und das Kennwort des Benutzers, sondern noch weitergehende Informationen zum Konto.

```
root@defiant:/home/ole > grep tapico /etc/shadow
tapico:9Fd4SU/BbEP52:11317:0:10000:::
```

Die Felder von links nach rechts sind:

1. Der Benutzername
2. Das verschlüsselte Kennwort.
3. Die Anzahl von Tagen zwischen dem 01.01.1970 und der letzten Kennwortänderung.
4. Die Zeit in Tagen, die zwischen zwei Kennwortänderungen liegen muß.
5. Die Zeit in Tagen, wie lange ein Kennwort gültig ist.
6. Die Zeit in Tagen, wie lange der Benutzer vor dem Auslaufen des Kennworts gewarnt wird.
7. Die Zeit in Tagen bis das Konto nach dem Auslaufen des Kennworts gesperrt wird.

8. Auslaufen des Kontos in Tagen seit dem 01.01.1970.

Im Gegensatz zur `/etc/passwd` darf das Kennwortfeld nicht leer sein. Um einen Benutzer daran zu hindern sein Kennwort zu ändern, brauchen Sie nur das minimale Kennwortalter größer zu machen als das maximale Kennwortalter.

### 8.5.2 `/etc/gshadow`

Genau wie das Kennwort kann auch das Gruppenkennwort durch das Shadow-Paßwort-System besser geschützt werden. Hier ist die Datei `/etc/gshadow` dafür verantwortlich. Ist das System eingerichtet, wird das Kennwortfeld in der `/etc/group` durch ein 'x' ersetzt und das Kennwort in der `/etc/gshadow` eingetragen. Auch auf die `/etc/gshadow` sollte nur `root` Zugriffsrechte besitzen.

Sie sollten auf keinen Fall die Dateien `/etc/shadow` oder `/etc/gshadow` selbst editieren. Benutzen Sie lieber dafür die vorgesehen Tools wie `useradd`, `usermod` oder `gpasswd`.

### 8.5.3 `pwconv`

Um das Shadow-Paßwort-System zu nutzen, müssen Sie die Kennwörter aus der `passwd`-Datei in eine neue Datei umgewandelt werden. Der Befehl `pwconv` erzeugt aus Ihrer `/etc/passwd` eine `/etc/shadow`-Datei und ersetzt die Kennwörter durch 'x'.

```
pwconv
```

*Um diesen Befehl erfolgreich ausführen zu können, darf kein Paßwortfeld in der `/etc/passwd` leer sein.*

### 8.5.4 `pwunconv`

Um das Shadow-Paßwort-System zu deaktivieren, wird der Befehl `pwunconv` verwendet.

```
pwunconv
```

Der Befehl bringt die Datei `/etc/passwd` auf den neuesten Stand und löscht die `/etc/shadow`.

### 8.5.5 `grpconv`

Was `pwconv` für die `/etc/passwd` ist `grpconv` für `/etc/group`.

```
grpconv
```

### 8.5.6 `grpunconv`

Wie bei den Benutzern kann auch das verbesserte System zu Kennwortspeicherung auch bei den Gruppen rückgängig gemacht werden.

```
grpunconv
```

Wenn Sie die Dateien `/etc/passwd` und `/etc/group` per Hand editieren müssen, ist es oft notwendig, daß Sie die Shadow-Dateien updaten. Dazu können Sie die Befehle `pwconv` und `grpconv` verwenden. In der Regel ist es am besten erst mit `pwunconv` und `grpunconv` das Shadow-Paßwort-System abzuschalten, die Änderungen vorzunehmen und dann mit `pwconv` und `grpconv` wieder zu aktivieren.

## 8.6 Startdateien des Benutzers

Das Aussehen und die Funktion der Shell ist durch die Umgebungsvariablen festgelegt. Diese Parameter sind in verschiedenen Konfigurationsdateien im Verzeichnis `/etc` und im Heimatverzeichnis des Benutzers festgelegt bzw. werden bei jedem Start und Einloggen definiert.

### 8.6.1 `/etc/profile`

Die Standardeinträge für die Umgebungsvariablen für alle Benutzer, die sich interaktiv einloggen, befinden sich in der `/etc/profile`. Hier das Beispiel für eine solche Datei.

```
# /etc/profile

PROFILEREAD=true

umask 022

# adjust some limits (see bash(1))
ulimit -Sc 0 # don't create core files
ulimit -d unlimited
fi
ulimit -m unlimited

# make path more comfortable
#
MACHINE='test -x /bin/uname && /bin/uname --machine'
PATH=/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin
test "$UID" = 0 || PATH="$PATH:."
export PATH

# set some environment variables
#
if test -n "$TEXINPUTS" ; then
    TEXINPUTS=":$TEXINPUTS:~/TeX:/usr/doc/TeX"
else
    TEXINPUTS="~/TeX:/usr/doc/TeX"
fi
export TEXINPUTS
PRINTER='lp'
export PRINTER

LESSCHARSET=latin1
export LESSCHARSET
LESS="-M -S -I"
export LESS
LESSKEY=/etc/lesskey.bin
export LESSKEY
LESSOPEN="|lesspipe.sh %s"
export LESSOPEN

MANPATH=/usr/local/man:/usr/share/man:/usr/man:/usr/X11R6/man
export MANPATH

# set INFOPATH to tell xemacs where he can find the info files
#
INFODIR=/usr/local/info:/usr/share/info:/usr/info
INFOPATH=$INFODIR
```

```

export INFODIR INFOPATH

# do not save dupes in the bash history file
HISTCONTROL=ignoredups
export HISTCONTROL

alias dir='ls -l'
alias la='ls -la'
alias l='ls -alF'
alias ls-l='ls -l'
alias o='less'
alias ..='cd ..'
alias ...='cd ../../'
alias rd=rmdir
alias md='mkdir -p'
alias beep='echo -en "\x07"'
alias umount='echo "Error: Try the command: umount"'

function startx { /usr/X11R6/bin/startx $* 2>&1 | tee ~/.X.err ; }

PAGER=less
export PAGER

if [ "$TERM" = "" -o "$TERM" = "unknown" ]; then
    TERM=linux
fi

# set prompt
#
set -p
if test "$UID" = 0 ; then
    PS1="\h:\w # "
else
    PS1="\u@\h:\w > "
fi
PS2='> '
ignoreeof=0
export PS1 PS2 ignoreeof

```

Um systemweite Änderungen in den Umgebungen der Benutzer durchzuführen, sollten Sie diese Datei editieren. S.u.S.E.-Linux erstellt die `/etc/profile` selber. Für die Änderungen des Administrators ist die `/etc/profile.local` vorgesehen, die aus der `profile` aufgerufen wird.

Sie können die `bash` auch aufrufen, ohne daß die `/etc/profile` abgearbeitet wird. Dies bewirkt der Schalter `--noprofile`.

Wenn der Benutzer sein eigenes Umfeld gestalten will, kann er das in der Datei `.profile` machen, die in seinem Heimatverzeichnis liegt. Wenn ein Benutzer sich einloggt, dann liest das Betriebssystem erst die `/etc/profile` aus. Dann sucht es nach den Dateien `.bash_profile`, `.bash_login` und `.profile` im Heimatverzeichnis des Benutzers.

Beim Ausloggen aus einer interaktiven Shell wird, wenn vorhanden, die Datei `.bash_logout` im Heimatverzeichnis ausgeführt.

### 8.6.2 /etc/bashrc

Wenn die `bash` eine interactive Shell aufruft, die nicht die Login-Shell ist, wie es z. B. bei der Eingabe des Befehls `bash` vorkommt, dann wird die Datei `/etc/bashrc` abgearbeitet anstatt die `/etc/profile`.



Durch die Option `-norc` kann die Abarbeitung der Konfigurationsdatei verhindert werden. Der Schalter `-rcfile DATEINAME` ermöglicht sogar die Angabe einer alternativen Konfigurationsdatei.

Kommt es zum Aufruf einer nicht-interactiven Shell, z. B. wenn ein Shell-Skript ausgeführt wird, dann schaut das Betriebssystem erst einmal in der Umgebungsvariablen `BASH_ENV` nach, ob dort der Name einer Konfigurationsdatei steht. Ansonsten wird dann auch die `/etc/bashrc` ausgeführt.

Auch diese Konfigurationsdatei gibt es als individuelle Version für den Benutzer als `~/.bashrc`

### 8.6.3 sh

Der Befehl `sh` ist im Prinzip nur ein anderer Namen für den Befehl `bash`. Allerdings wertet er beim Aufruf die Umgebungsvariable `ENV` aus. Dort steht der Name der Konfigurationsdatei, die ausgeführt werden soll. Andere Konfigurationsdateien werden im Gegensatz zu `bash` nicht ausgeführt.

## 8.7 Das “Who-is-Who” der Benutzer

### 8.7.1 who

Wer ist denn zur Zeit eingeloggt? Diese Frage stellt sich dem Administrator immer dann, wenn er den Rechner herunterfahren will. Aber auch da kann Linux weiterhelfen. Der Befehl `who` zeigt alle eingeloggten Benutzer mit ihren Terminals an.

`who [OPTIONEN]`

#### Optionen

<code>-H</code>	Zeigt eine Überschriftenzeile
<code>-i, -u</code>	Zeigt an, wie lange ein Benutzer nicht mehr gearbeitet hat
<code>-m</code>	Zeigt den aktuellen Benutzer an
<code>-q</code>	Zeigt die eingeloggten Benutzer und ihre Anzahl an

Für die Ausgabe des Befehls wertet er die Dateien `/var/run/utmp` und `/var/log/wtmp` aus.

Der Befehl kann aber auch auf eines der wichtigsten Probleme im Leben eines Administrators eine Antwort geben: “Wer bin ich?”. Einfach `who am I` eingeben und die Antwort ist da. Wer es liebevoller mag, dem sei die Version `who mom likes` an Herz gelegt. Im Endeffekt sind aber diese Erweiterungen nur andere Namen für `who -m`.

#### Beispiel

```
tapico@defiant:~ > who
tapico  tty1      Feb 10 13:26
tapico  tty3      Feb 10 15:28
tapico  tty4      Feb 10 15:26
grayson pts/0      Feb 10 14:05
tapico@defiant:~ > who -Hi
BENU    LEIT      LOGIN-ZEIT  RUHIG VON
tapico  tty1      Feb 10 13:26 05:29
tapico  tty3      Feb 10 15:28 00:15
tapico  tty4      Feb 10 15:26 00:30
grayson pts/0      Feb 10 14:05 .
root    pts/1      Feb 10 18:49 00:05
```

### 8.7.2 whoami

Der Befehl `whoami` sagt zwar auch, wer ich gerade bin. Aber er macht das wesentlich kürzer.

`whoami`

### 8.7.3 logname

Der Befehl `logname` erzählt dem Benutzer unter welchem Namen er sich ursprünglich mal eingeloggt hat.

`logname`

Frisch nach dem Einloggen zeigen die Befehle `whoami` und `logname` keine Unterschiede. Erst wenn ein Identitätswechsel vollzogen wurde, werden die Unterschiede der beiden Befehle deutlich.

```
ovidio@defiant:~ > whoami
ovidio
ovidio@defiant:~ > logname
ovidio
ovidio@defiant:~ > su tapico
Password:
tapico@defiant:/home/ovidio > whoami
tapico
tapico@defiant:/home/ovidio > logname
ovidio
```

### 8.7.4 w

Der Befehl `w` zeigt wie `who` die eingeloggten Benutzer an. Er liefert allerdings weitergehende Informationen wie Einlogzeitpunkt, CPU-Nutzung und was der Benutzer gerade ausführt.

`w [OPTIONEN]`

#### Beispiel

```
tapico@defiant:~ > w
 4:23pm up 2:00, 6 users, load average: 0.00, 0.00, 0.00
USER   TTY     FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
tapico  tty1    -              2:24pm  1:58m 44.56s  0.04s  sh /usr/X11R6/bin/startx
root    tty2    -              4:22pm  22.00s 0.12s  0.06s  -bash
lori     tty3    -              4:23pm  14.00s 0.08s  0.05s  -bash
tapico  tty5    -              4:21pm  1:33   0.08s  0.04s  -bash
tapico  pts/0   -              4:21pm  0.00s  0.08s  0.03s  w
tapico  pts/1   -              3:37pm  2:05   2.32s  0.10s  bash
```

### 8.7.5 finger

Auch der Befehl `finger` zeigt die eingeloggten Benutzer an. Dabei kann er eine Übersicht über die Benutzer liefern oder ausführliche Informationen über einzelne Benutzer.

`finger [OPTIONEN] [BENUTZER]`

Der Befehl `finger` wertet für seine Ausgabe die Datei `/etc/passwd` sowie die Dateien `.plan` und `.project` im Heimatverzeichnis des jeweiligen Benutzers aus. Diese beiden Dateien geben Auskunft über zukünftige Pläne und Projekte.

#### Beispiele

```
lori@defiant:~ > finger
Login      Name           Tty      Idle   Login Time   Where
lori       Lori Kalmar     3         5      Sun 16:23
```

```

root      Herr und Meister      2          5      Sun 16:22
tapico    Thomas Prizzi             1          2:04   Sun 14:24
tapico    Thomas Prizzi          pts/0      -       Sun 16:21
tapico    Thomas Prizzi          pts/1      7       Sun 15:37
tapico    Thomas Prizzi          5          6       Sun 16:21
lori@defiant:~ > finger tapico
Login: tapico                      Name: Thomas Prizzi
Directory: /home/tapico           Shell: /bin/bash
On since Sun Feb 11 16:21 (MET) on tty5, idle 0:12
No Mail.
Plan:
Bin vom 15. bis 17. Februar nicht in der Firma.

```

### 8.7.6 last

Der Befehl `last` zeigt die Benutzer an, die sich zuletzt eingeloggt haben. Er durchsucht dafür die Datei `/var/log/wtmp`.

```
last [OPTIONEN] [BENUTZER]
```

Daneben zeigt er auch an, seit wann die Datei `/var/log/wtmp` Informationen enthält.

```

tapico@defiant:~ > last -10
lori      tty3                      Sun Feb 11 16:23   still logged in
root      tty2                      Sun Feb 11 16:22   still logged in
tapico    tty5                      Sun Feb 11 16:21   still logged in
lori      pts/0                     Sun Feb 11 16:21   still logged in
tapico    pts/1                     Sun Feb 11 15:37   still logged in
tapico    pts/0                     Sun Feb 11 14:47 - 15:54 (01:07)
root      tty1                      Sun Feb 11 14:24   still logged in
reboot    system boot 2.2.14          Sun Feb 11 14:23   (04:32)
lori      pts/2                     Sat Feb 10 19:09 - 01:05 (05:55)
tapico    pts/1                     Sat Feb 10 18:49 - 01:05 (06:15)

wtmp begins Mon Feb  5 20:18:53 2001

```

### 8.7.7 lastlog

Mit dem Befehl `lastlog` ist es möglich den Zeitpunkt des letzten Einloggens eines Benutzer zu bestimmen.

```
lastlog [OPTIONEN]
```

#### Optionen

```

-t N          beschränkt die Ausgabe auf die letzten N Tage. (time)
-u NAME       erlaubt die direkte Anzeige der Daten eines Benutzers. (user)

```

Für die Auswertung greift der Befehl auf die Logdatei `/var/log/lastlog` zurück.

```

tapico@defiant:~ > lastlog -u tapico
Username      Port      From      Latest
tapico        tty5      Son Feb 11 16:21:59 +0100 2001
tapico@defiant:~ > lastlog -t 1
Username      Port      From      Latest
root          tty2      Son Feb 11 16:22:16 +0100 2001
tapico        tty5      Son Feb 11 16:21:59 +0100 2001
lori          tty3      Son Feb 11 16:23:18 +0100 2001

```

.....  
*Notizen:*

.....

## VIII

Arbeitsblatt  
Benutzerverwaltung

## 8.1

- 1 Loggen Sie sich als *root* im Terminal 1 ein.
- 2 Richten Sie den Benutzer *grayson* mit den Default-Bedingungen ein.
- 3 Welche Einträge sind für den Benutzer in */etc/passwd* und */etc/shadow* gemacht worden?
- 4 Ist ein Home-Verzeichnis für *grayson* angelegt worden?
- 5 Wechseln Sie zu Terminal 2 und loggen Sie sich als *grayson* ein.
- 6 Wechseln Sie zu Terminal 1 und ändern das Paßwort von *grayson* auf 'werner'.
- 7 Welches Paßwort steht in */etc/shadow* für *grayson*? Vergleichen Sie den Eintrag mit dem Eintrag ihres Nachbarn.
- 8 Wechseln Sie zu Terminal 2 und loggen Sie sich als *grayson* ein.
- 9 Welches ist das aktuelle Verzeichnis?
- 10 Legen Sie den Benutzer *lori* mit folgendem Befehl an!  
`useradd -c "Lori Kalmar" -m -p locust lori`  
Was bewirkt der Befehl?
- 11 Wechseln Sie zu Terminal 3 und versuchen sich als *lori* einzuloggen. Was passiert?
- 12 Wechseln Sie wieder zu Terminal 1 und lassen sie sich die */etc/shadow* anzeigen. Warum konnten Sie sich nicht als *lori* einloggen?
- 13 Ändern Sie das Paßwort von *lori* so, daß Sie sich einloggen können? Testen Sie das.
- 14 Editieren Sie die Datei */etc/shadow*. Kopieren Sie das Paßwort von *lori* in das Paßwortfeld von *grayson*. Versuchen Sie sich nun wieder an Terminal 2 mit dem Paßwort von *lori* als *grayson* einzuloggen.
- 15 Editieren Sie die Datei */etc/shadow* für den Benutzer *grayson*. Fügen Sie vor dem verschlüsselten Paßwort ein '\*' (Asterisk, Stern) ein. Versuchen Sie sich wieder am Terminal 2 als *grayson* einzuloggen. (Eventuell vorher ausloggen.)
- 16 Machen Sie die Änderung in */etc/shadow* rückgängig und versuchen Sie sich wieder am Terminal 2 als *grayson* einzuloggen.
- 17 Wie müssen Sie vorgehen um ein Konto fürs Einloggen zu deaktivieren? Dabei darf das Paßwort nicht unbrauchbar gemacht werden, da es für einen FTP-Zugang gebraucht wird. Beschreiben Sie das Verfahren und testen Sie es am Benutzer *grayson*.
- 18 Wechseln Sie zu Terminal 1 und legen Sie die Gruppe *greydeath* an.
- 19 Lassen Sie sich alle Gruppen anzeigen in denen *root* Mitglied ist. Benutzen Sie dazu die Befehle `groups` und `groups root`. Erklären Sie den Unterschied.
- 20 Fügen Sie *root* zur Gruppe *greydeath* hinzu.
- 21 Lassen Sie sich alle Gruppen anzeigen in denen *root* Mitglied ist. Was ist das Problem? (Testen Sie die unterschiedliche Arbeitsweise von `groups` und `groups root`.)
- 22 Lassen Sie sich die Datei */etc/group* anzeigen. Ist *root* eingetragen bei der Gruppe *greydeath*?
- 23 Loggen Sie sich einmal aus und wieder als *root* ein. Überprüfen Sie wiederum die Gruppenzugehörigkeit.

## VIII

Arbeitsblatt  
Benutzerverwaltung

## 8.2

- 24** Legen Sie die Datei `/trellwan` (Inhalt: ein schöner Spruch) an. Ändern Sie die Gruppenzugehörigkeit auf `greydeath` und die Rechte mit dem Befehl `chmod 640 /trellwan`. Damit besitzt der Besitzer der Datei Schreib- und Leserecht und die Gruppe `greydeath` das Leserecht.
- 25** Können `grayson` und `lori` auf `/trellwan` zugreifen?
- 26** Fügen Sie als `root` `grayson` zur Gruppe `greydeath` hinzu. Lassen Sie sich als `root` und als `grayson` die Gruppenzugehörigkeit von `grayson` anzeigen.
- 27** Lassen Sie sich nun als `grayson` den Inhalt von `/trellwan` anzeigen.
- 28** Loggen Sie sich als Benutzer `lori` aus. Löschen Sie den Benutzer `lori` mit seinem Heimatverzeichnis.
- 29** Löschen Sie die Gruppe `greydeath`.
- 30** Versuchen Sie nun den Benutzer `grayson` zu löschen. Was passiert?
- 31** Loggen Sie als `grayson` aus und löschen den Benutzer. Was ist mit seinem Heimatverzeichnis passiert?
- 32** Welche Informationen bekommen Sie über die Datei `/trellwan`?
- 33** Löschen Sie die Datei `/trellwan`.
- 34** Sie sollen auf der Applikationsebene einen Fehler für einen Benutzer beheben. Leider kennen Sie das Passwort des Benutzers nicht, wollen es aber auch nicht permanent ändern. Beschreiben Sie ihre Vorgehensweise.

.....  
*Notizen:*

.....

## Kapitel 9

# Rechte im Linux-Dateisystem

### 9.1 Die Rechte

Basierend auf der Art, wie auf Dateien zugegriffen werden kann, unterscheidet Linux drei Rechte. Sie werden abkürzend mit **r**, **w** oder **x** bezeichnet. Die einzelnen Rechte sind an die jeweilige Datei gebunden und werden in der Inode der Datei gespeichert. Dabei ist besonders wichtig in welcher Stellung Sie sich zu der Datei befinden. Sind Sie der Besitzer, gehören Sie zur Gruppe, die Zugriff auf die Datei hat, oder gehören Sie zum Rest der Welt.

Wie vom Befehl `ls -l` bekannt, besteht die Rechtetabelle aus 9 Angaben, z. B. `rwxr--xr--`. Die ersten drei angegebenen Rechte gelten dabei für den Dateibesitzer, die folgenden drei für die Besitzergruppe und die letzten für den Rest der Welt.

#### 9.1.1 Rechte auf Dateien

Die Wirkung der Rechte ist bei Dateien und Verzeichnissen unterschiedlich. Bei Dateien arbeiten Sie wie folgt:

**r (read = lesen)** Der Benutzer kann den Inhalt der Datei einsehen, d. h. unter anderem, er kann sie am Bildschirm anzeigen lassen, sie drucken oder kopieren.

**w (write = schreiben)** Der Benutzer kann die Datei verändert unter dem bisherigen Namen speichern.

**x (execute = ausführen)** Die Datei kann als Programm gestartet werden. Dies setzt natürlich voraus, daß die Datei ein Programm ist.

Angenommen also, Sie haben an der Datei `uschistagebuch` kein **r**-Recht, wollen Sie aber trotzdem auf dem Bildschirm sehen. Die Eingabe von

```
cat uschistagebuch
```

führt dann zu einer Fehlermeldung der Art:

```
uschistagebuch: permission denied
```

Viele Anwender, die aus einer Einplatzumgebung zu Linux stoßen, überschätzen allerdings die Bedeutung der Rechte, die man an einer (gewöhnlichen) Datei hat. Dies mag daran liegen, das MS-DOS und viele seiner Verwandten den Zugriff auf Dateien durch Attribute zu regeln versuchen. Die vielzitierte "schreibgeschützte Datei" gibt es unter Linux nicht.

Der Dateischutz wird ausschließlich über die Rechte gewährleistet. Hierbei gilt eine glasklare Logik, die sich im wesentlichen aus dem Dateisystem ergibt: Wird in einer Befehlszeile ein Pfad angegeben, auf den der Befehl zugreifen soll, werden die einzelnen Bestandteile des Pfades daraufhin überprüft, ob hinreichende Rechte bestehen. Ist dies nicht der Fall, wird der Zugriff verweigert.

### 9.1.2 Rechte auf Verzeichnisse

Verzeichnisse sind für Linux nur eine spezielle Art von Dateien. Dies macht sich hier daran bemerkbar, daß die Rechte `r` und `w` für Verzeichnisse genauso gehandhabt werden, wie für Dateien. Daraus ergeben sich weitreichende Konsequenzen für die darin enthaltenen Dateien!

Das `r`-Recht an einem Verzeichnis zu besitzen heißt, man kann den Inhalt des Verzeichnisses lesen, mit anderen Worten, man kann den Befehl `ls` für dieses Verzeichnis einsetzen.

Was aber ist nun der Inhalt eines Verzeichnisses? Ein Verzeichnis ist nichts anderes als eine kleine Tabelle. Jede Zeile dieser Tabelle hat zwei Einträge: Den Dateinamen und die zugehörige Inode-Nummer. Wie Sie bereits wissen, ist der Dateiname für Linux nicht das entscheidende Kriterium für die Bearbeitung einer Datei. Linux orientiert sich an der Inode.

Was bedeutet in diesem Zusammenhang das `r`-Recht für ein Verzeichnis?

Ganz einfach: Mit dem `r`-Recht am Verzeichnis können darin Dateien gefunden werden (der Befehl `ls` liefert die Namen der im Verzeichnis enthaltenen Dateien), zusätzlich kann man die Inode-Nummern dieser Dateien abfragen (`ls -li`).

Da in der Inode selbst die Verwaltungsinformationen (darunter die Rechtetabelle für die Datei) gespeichert sind, ist ein Zugriff auf eine Datei also erst dann möglich, wenn man ihre Inode kennt! Das heißt also, wenn Ihnen das `r`-Recht an einem Verzeichnis fehlt, haben Sie keinen Zugang zu den darin enthaltenen Dateien! Insbesondere sind auch Unterverzeichnisse dem Zugriff des Anwenders entzogen!

Das für das Verständnis der Linux-Rechte wichtigste Recht ist wohl das `w`-Recht für Verzeichnisse. Dieses Recht beinhaltet die Möglichkeit ein Verzeichnis verändert abzuspeichern. Anders ausgedrückt: Ein Benutzer, der an einem Verzeichnis das `w`-Recht hat, kann dieses verändern.

Nun ist wie gesagt ein Verzeichnis eine Liste von Dateinamen (plus zugehöriger Inode-Nummer). Besitzen Sie das `w`-Recht an einem Verzeichnis, können Sie einen Verzeichniseintrag löschen, das Verzeichnis wird verändert gespeichert, der Dateiname und die Inode-Nummer existieren nicht mehr. Einen Verzeichniseintrag löschen heißt also den Zugang (Link) zu einer Datei löschen! Dies geschieht allerdings nicht mit einem Editor sondern mit Hilfe des Befehls `rm`.

Verzeichnisse sind insofern "schreibgeschützt", als Linux aus Konsistenzgründen schreibende Zugriffe auf Verzeichnisse nur mit dafür geeigneten Befehlen erlaubt. Das Bearbeiten mit einem Editor würde viele Benutzer überfordern, denn es muß Rücksicht auf die Struktur des Verzeichnisinhalts genommen werden. Wird diese aber zerstört, sind die Daten im Verzeichnis nicht mehr verfügbar!

Nehmen wir nun einen extremen Fall an. Eine Datei, an der Sie keinerlei Rechte besitzen, befindet sich in einem Verzeichnis, in dem Sie das `w`-Recht haben. Diese Datei können Sie, auch ohne Rechte an ihr selbst, löschen, denn Sie können das Verzeichnis manipulieren! Sie entfernen mit dem Befehl `rm` den Eintrag der Datei im Verzeichnis und somit löschen Sie den Zugang zu den Daten.

Gleiches gilt für das Umbenennen (ändert den Eintrag in einem Verzeichnis), Verschieben (löscht den Eintrag in einem Verzeichnis) oder Erstellen (erzeugt einen neuen Eintrag in einem Verzeichnis) einer Datei. Auch hierzu ist das `w`-Recht am Verzeichnis von entscheidender Bedeutung.

Wie Sie sehen hängen sehr viele Dateioperationen von Verzeichnisrechten ab. Vereinfachend gesagt, Verwaltungsoperationen mit Dateien, wie z. B. Löschen, Verschieben oder Umbenennen sind Eingriffe in die Verzeichnisstruktur und werden deshalb über Rechte an Verzeichnissen geregelt. Die inhaltliche Manipulation von Dateien ist dagegen nur möglich, wenn Sie Rechte direkt an der Datei haben. Es kann sein, daß Sie eine Datei löschen, umbenennen oder verschieben können, ohne ihren Inhalt auch nur sehen zu können. Zur Manipulation der Dateiinhalte benötigt Sie zumindest das `r`-Recht an ihr, alle anderen genannten Operationen erfordern das `w`-Recht am Verzeichnis.

Alles in allem kann man feststellen, daß Verzeichnisse bezüglich der Rechte `r` und `w` wie gewöhnliche Dateien behandelt werden. Da man aber mit einem Verzeichnis eine Liste von Dateinamen bearbeitet, bedeutet das Ändern von Verzeichnissen sehr häufig ein Löschen, Umbenennen oder Verschieben von Dateien.

Bis jetzt haben sich Verzeichnisse wie Dateien verhalten. Beim `x`-Recht gibt es aber einen signifikanten Unterschied. `x` steht für das Recht ein Programm ausführen zu können. In diesem Sinne kann also ein Verzeichnis nicht ausführbar sein.

Das `x`-Recht an einem Verzeichnis bedeutet nun, daß Sie



- mit `cd` in das Verzeichnis wechseln dürfen und
- Zugang zu den Inodes aller darin enthaltenen Dateien erhalten.

Rechte an einem Verzeichnis betreffen die darin enthaltenen Dateien. Wollen Sie sich den Inhalt der Datei `/home/tapico/.profile` mit `cat /home/tapico/.profile` ansehen, so setzt dies folgende Rechte voraus:

- Das `r`-Recht an der Datei selbst.
- Das `r`-Recht am Verzeichnis `/home/tapico`.
- Das `x`-Recht an diesem Verzeichnis.

`cat` muß zunächst den Inhalt von `/home/tapico` lesen, um zu prüfen, ob die Datei überhaupt existiert. Dies erlaubt das `r`-Recht am Verzeichnis. Weiter benötigt `cat` die Inode-Nummer der Datei, um über diese die Inode selbst finden, Zugang dazu erhält `cat` über das `x`-Recht an `/home/tapico`. In der Inode sind nun aber alle relevanten Informationen, die `cat` braucht, um den Dateiinhalt endlich lesen zu können. Hier ist die Rechttabelle, die über das `r`-Recht die Erlaubnis zum Lesen gewährt und schließlich sind hier die Informationen abgelegt, mit deren Hilfe die Daten auf der Platte gefunden werden können.

In der Praxis erhält man mit dem `r`-Recht an einem Verzeichnis stets auch das `x`-Recht. Natürlich kann man das `x`-Recht wieder nehmen (sofern man dazu die Erlaubnis hat), dann schafft man aber eine skurile Situation: Das Leserecht am Verzeichnis erlaubt es dann zum Beispiel `ls`, Dateien zu finden, weitere Informationen werden aber verweigert. Man erhält in jedem Fall eine Fehlermeldung wie die folgende

```
/bin/ls: /home/tapico/.profile: Permission denied
```

`ls` meldet, daß man auf die Inode von `.profile` im Verzeichnis `/home/tapico` nicht zugreifen darf: `Permission denied`.

An den meisten Verzeichnissen im Dateisystem hat der gewöhnliche Benutzer die Rechte `r` und `x`, am Heimatverzeichnis aber alle Rechte.

Neben den Abkürzungen `r`, `w` und `x` gibt es auch noch eine oktale<sup>1</sup> Notation der Rechte. Dies ist eine dreistellige Zahl, bei der die erste Ziffer für den Besitzer gilt, die zweite für die Gruppe, die dritte für alle anderen.

Ziffer	Bedeutung
4	<code>r</code> -Recht
2	<code>w</code> -Recht
1	<code>x</code> -Recht
0	kein Recht

Die Rechtekombinationen ergeben sich durch Addition! So bedeutet `754` dasselbe wie `rw-r--r--`!

Anstelle der Kennzeichnung `x` treten hin und wieder ein `s` oder ein `t` auf.

### 9.1.3 Spezielle Rechte: SUID und SGID

SUID (*Set User ID*) und SGID (*Set Group ID*) werden durch ein `s` anstelle eines `x` bei den Eigentümer- und bei den Gruppenrechten ausgedrückt. Das `s`-Recht an einer ausführbaren Datei bedeutet, daß der Benutzer, der es startet, während des Programmlaufes die UID des Dateibesitzers bzw. die GID der Besitzergruppe erhält. Das hängt davon ab, ob das `s` beim Besitzer oder der Besitzergruppe steht. Im Falle des Programms `/bin/passwd` wird man z. B. während des Programmlaufs zum Superuser!

```
tapico@defiant:~ > ls -l /etc/passwd /etc/shadow
-rw-r--r--  1 root    root      3810 Dec 26 19:56 /etc/passwd
-rw-r-----  1 root    shadow    2422 Dec 26 19:56 /etc/shadow
tapico@defiant:~ > ls -l /usr/bin/passwd
-rwsr-xr-x  1 root    shadow    27920 Mar 11  2000 /usr/bin/passwd
```

---

<sup>1</sup>Auf der Basis der Zahl 8!

Wie Sie sehen, hat nur der Superuser Schreibrechte auf die Dateien `/etc/passwd` und `/etc/shadow`. Das `s`-Recht auf den Befehl `/usr/bin/passwd` bewirkt, daß der Befehl immer unter der UID des Superuser ausgeführt wird. So kann auch ein normaler Benutzer sein Kennwort ändern.

Wird das SGID-Recht auf ein Verzeichnis gegeben, so hat es eine andere Funktion. In diesem Fall erhalten alle Dateien, die in diesem Verzeichnis erstellt werden, automatisch die gleiche Gruppe, die auch dem Verzeichnis zugeordnet ist.

### Beispiel

Nehmen wir mal an, der User `tapico`, dem die Hauptgruppe `users` zugeteilt wurde, ist auch Mitglied der Gruppe `web`. Für das Verzeichnis `/home/wwwrun` mit der zugeordneten Gruppe `web` wurde das SGID-Recht gesetzt. Legt `tapico` in einem Verzeichnis z. B. seinem Heimatverzeichnis eine Datei an, so wird der Datei seine Hauptgruppe zugeordnet. Legt er aber in dem Verzeichnis `/home/wwwrun` eine Datei an, so bekommt die Datei die Gruppe des Verzeichnis zugewiesen.

```
tapico@defiant:~> touch screen
tapico@defiant:~> ls -l screen
-rw-r--r-- 1 tapico users 0 Nov 18 09:07 screen
tapico@defiant:~> cd /home/wwwrun
tapico@defiant:/home/wwwrun> ls -ld .
drwxrwsr-x 2 wwwrun web 35 Nov 18 08:58 .
tapico@defiant:/home/wwwrun> touch mahal.html
tapico@defiant:/home/wwwrun> ls -l
insgesamt 0
-rw-r--r-- 1 tapico web 0 Nov 18 09:03 mahal.html
```

#### 9.1.4 Skript: Formatieren von Disketten

Wie schon im Abschnitt 10.4.1 erwähnt, kann nur der Benutzer `root` Dateisystem einrichten. Sollen nun normale Benutzer Disketten formatieren können, so kann dies durch ein Skript erfolgen, was durch das Setzen des SUID-Rechtes unter der Identität von `root` läuft.

Beispiel für das Skript:

```
root@defiant:/usr/bin> ls -l fd0format
-rwsr-xr-x 1 root users 124 Okt 19 22:33 fd0format
root@defiant:/usr/bin> cat fd0format
#Formatiert eine Diskette Low-Level und erzeugt das Dateisystem ext2
#SUID für das Skript setzen, da /sbin nur für root zugänglich

/usr/bin/fdformat /dev/fd0
/sbin/mkfs -t ext2 /dev/fd0
```

#### 9.1.5 Spezielle Rechte: Sticky Bit

Das `t`-Recht, auch *Sticky Bit* oder *Save Text Bit* genannt, wirkt sich auf Verzeichnisse und Dateien unterschiedlich aus.

Das Sticky Bit bei ausführbaren Dateien veranlaßt Linux, den Programmcode des gestarteten Programms nicht wieder aus dem Arbeitsspeicher zu entfernen. Dies hat den Vorteil, daß bei einem weiteren Start des Programms dieses nicht erst in den RAM geladen werden muß. Dieses Merkmal ist besonders nützlich, wenn ein Programm häufig eingesetzt wird; denn der Programmcode muß nur einmal im Speicher sein, er kann aber von mehreren Benutzern gleichzeitig genutzt werden.

Bei Verzeichnissen hingegen ermöglicht das Sticky Bit das Anlegen von gemeinsam genutzten Verzeichnissen. Normalerweise muß ein solches Verzeichnis die Rechte `drwxrwxrwx` besitzen. Jeder kann nun in dieses Verzeichnis schreiben, allerdings auch jede Datei darin löschen. Durch das Setzen des Sticky Bits wird es nur dem Eigentümer einer Datei erlaubt, diese zu löschen.

Das t-Recht lässt sich nur vom Superuser vergeben.

Oktal werden die neuen Rechte durch eine weitere vorangestellte Ziffer ausgedrückt. So werden z. B. die Rechte `rwrxrwxrwt` durch `1777` ausgedrückt.

Ziffer	Bedeutung
4	SUID
2	GUID
1	Sticky Bit

## 9.2 Verwalten und Setzen von Rechten

Das Verwalten und Setzen von Rechten übernehmen Programme wie `chown`, `chmod`, `chgrp` und `umask`.

### 9.2.1 chown

Mit dem Befehl `chown` ist es möglich der Datei einen neuen Besitzer und eine neue Gruppe zu geben.

```
chown [OPTIONEN] [BENUTZER] [:GRUPPE] DATEILISTE
```

Allerdings ist es nur dem Superuser gestattet, den Eigentümer einer Datei zu ändern. Die Änderung der Gruppe ist für den Besitzer und für den Superuser erlaubt. Der Besitzer darf allerdings seiner Datei nur eine Gruppe zuordnen, in der er selber Mitglied ist.

#### Optionen

- c                    Informationen über alle geänderten Dateien werden angezeigt
- v                    Informationen über alle Aktionen werden ausgegeben
- f                    Fehlermeldungen werden nicht ausgegeben
- R                    Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

#### Beispiele

```
chown tapico pampelmuse
```

ändert den Besitzer der Datei `pampelmuse` auf `tapico`.

```
chown -R tapico:users /home/uschi
```

ändert den Besitzer auf `tapico` und die Gruppe auf `users` für das Verzeichnis `/home/uschi` und seinem Inhalt.

```
chown :xtra xtra*
```

ändert die Gruppe für alle Dateien, die mit "xtra" beginnen auf `xtra`.

### 9.2.2 chgrp

Ändert für eine Datei die zugehörige Gruppe.

```
chgrp [OPTIONEN] GRUPPE DATEILISTE
```

Die Änderung der Gruppe ist für den Besitzer und für den Superuser erlaubt. Der Besitzer darf allerdings seiner Datei nur eine Gruppe zuordnen, in der er selber Mitglied ist.

#### Optionen

- c                    Informationen über alle geänderten Dateien werden angezeigt
- v                    Informationen über alle Aktionen werden ausgegeben
- f                    Fehlermeldungen werden nicht ausgegeben
- R                    Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

#### Beispiel

```
chgrp users pampelmuse
```

ändert die Gruppe der Datei `pampelmuse` auf `users`.

### 9.2.3 chmod

Der Befehl `chmod` erlaubt das Ändern der Rechte für eine Datei.

`chmod [OPTIONEN] RECHTE DATEILISTE`

Die Rechte können dabei in der Form *WerWieWas* oder *Maske* angegeben werden.

*Wer* ist dann `u` (user/Besitzer), `g` (group/Gruppe), `o` (others/Sonstige), eine Kombination daraus oder `a` (all user/alle Benutzer).

*Was* ist eines der Rechte `r`, `w`, `x`, `s`, `t` oder eine Kombination daraus. Wobei sich `x`, `s` und `t` sich gegenseitig ausschließen.

*Wie* ist = (Rechte werden exakt so gesetzt wie in *Was* angegeben), + (Rechte aus *Was* werden zusätzlich vergeben) oder - (Rechte aus *was* werden entzogen)

*WerWieWas* kann mehrfach, dann durch Komma getrennt angegeben werden. In der ganzen Liste der *WerWieWas* darf kein Leerzeichen sein, da alles nach diesem Leerzeichen als Dateiname interpretiert wird!

*Maske* ist eine dreistellige (Superuser auch vierstellige) oktale Zahl wie in Abschnitt 9.1.2 und 9.1.5 beschrieben.

#### Optionen

<code>-c</code>	Informationen über alle geänderten Dateien werden angezeigt
<code>-v</code>	Informationen über alle Aktionen werden ausgegeben
<code>-f</code>	Fehlermeldungen werden nicht ausgegeben
<code>-R</code>	Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

#### Beispiele

```
chmod u+x memo*.txt
```

fügt für den Besitzer das `x`-Recht hinzu.

```
chmod 755 makeitso
```

setzt die Rechte auf `rwxr-xr-x`.

```
chmod a-rwx,u=rwx meinsein
```

entzieht allen alle Rechte und gibt dem Besitzer alle Rechte.

### 9.2.4 umask

Für die Erstellung neuer Dateien gibt es für die Rechte eine Maske, die bestimmt welche Rechte vergeben werden. Für diese Maske ist der Befehl `umask` (*User's creation MASK*) zuständig.

`umask [MASKE]`

Die Eingabe von `umask` ohne Parameter gibt die aktuell eingestellte Maske wieder. Die Maske selber ist eine dreistellige oktale Zahl. Die Bedeutung der Zahlen ist identisch mit derer für die Rechte aus Abschnitt 9.1.2. Allerdings gibt `umask` nicht an, welche Rechte gegeben werden, sondern welche Rechte entzogen werden.

Allerdings verhält sich die Rechtevergabe doch anders als erwartet. Manche Systeme liefern auch eine vierte Ziffer (an erster Stelle stehend) zurück. Diese beschreibt das besondere Verhalten der Rechtevergabe für neue Dateien, wie aus der Tabelle 9.1 zu ersehen ist. Dabei verhält sich die `bash` wie durch den Wert 0 gegeben. Die für die Benutzergruppen angegebenen Rechte werden von dieser Grundeinstellung abgezogen.

Beim Kopieren von Dateien hat `umask` auch seine Finger im Spiel. Als Vorgabe gelten die Rechte, die die Quelldatei besitzt. Diese werden dann mit der Einstellung von `umask` gefiltert.

Der durch `umask` eingestellte neue Wert gilt nur für die Dauer einer Sitzung. Eine permanente Änderung erreichen Sie, wenn Sie den Befehl `umask` in die Konfigurationsskripte `.profile` oder `.bashrc` eintragen.

#### Beispiel

```
umask 022
```

setzt die Defaultmaske auf `rwxr-xr-x`.

Wert	Bedeutung
0	rw für Dateien, rwx für Verzeichnisse
1	rw für Dateien und Verzeichnisse
2	r für Dateien, rx für Verzeichnisse
3	r für Dateien und Verzeichnisse
4	w für Dateien, wx für Verzeichnisse
5	w für Dateien und Verzeichnisse
6	x für Dateien und Verzeichnisse
7	Keine Rechte für Dateien und Verzeichnisse

Tabelle 9.1: Besondere Berechtigungen für umask

## 9.3 Mit anderen Rechten arbeiten

Manchmal ist es notwendig, daß ein normaler Benutzer ein Kommando ausführen soll, daß ansonsten nur `root` zugänglich ist. Bei normalen Skripten ist dies weniger ein Problem, als bei den Systemprogrammen. Diese besitzen meistens intern noch weitere Prüfroutinen auf den Benutzer. Leider sind diese meistens so restriktiv, daß man sogar mit dem SUID- und SGID-Recht nicht weiterkommt.

Im Prinzip können Sie Ihre Identität mit dem Befehl `su` wechseln. Sie können auch einen Befehl direkt mit `su` starten.

```
ole@enterprise:~> su -c "mount -t auto /dev/hda1 /mnt"
Password:
ole@enterprise:~>
```

Allerdings setzt diese Operation die Kenntnis des Paßworts desjenigen voraus, dessen Identität man annehmen will. Wenn Sie allerdings dies kennen, dann können Sie alles unter dieser Identität machen.

Um dieses Problem zu erledigen wurde der Befehl `sudo` entwickelt.

### 9.3.1 sudo

Der Befehl `sudo` erlaubt die Ausführung eines Programms unter fremder Identität. Dabei wird in der Konfigurationsdatei `/etc/sudoers` festgelegt, welcher Benutzer auf welchem Host was tun darf.

```
sudo [OPTIONEN] Kommando
sudo -v
```

Dabei wird die reale und die effektive UID und GID so gesetzt, daß sie mit den Daten des angegebenen Benutzers übereinstimmen. Die Standardeinstellung ist, daß der Benutzer sich vor der Ausführung mit seinem eigenen Paßwort identifizieren muß. Wenn sich ein Benutzer authentifiziert hat, dann wird ein Zeitstempel aktualisiert. Der Benutzer kann nun für 5 Minuten `sudo` benutzen, ohne sein Paßwort neu eingeben zu müssen. Die Zeitstempel werden im Verzeichnis `/var/run/sudo` gespeichert.

### Optionen

-V	Informationen über Version und Einstellungen (nur <i>root</i> ( <i>version</i> ))
-l	Zeigt die möglichen Befehle eines Benutzers an ( <i>list commands</i> )
-L	Zeigt die möglichen Standardoptionen der Datei <i>/etc/sudoers</i> an ( <i>list options</i> )
-h	Zeigt Hilfe zu <i>sudo</i> an ( <i>help</i> )
-v	Aktualisiert den Zeitstempel, damit es zu keinem Timeout kommt ( <i>validate</i> )
-k	Setzt den Zeitstempel auf eine ungültige Zeit, beim nächsten <i>sudo</i> -Befehl wird das Paßwort wieder abgefragt ( <i>kill</i> )
-K	Löscht den Zeitstempel völlig, beim nächsten <i>sudo</i> -Befehl wird das Paßwort wieder abgefragt ( <i>sure kill</i> )
-b	Startet das Kommando als Hintergrundprozeß ( <i>background</i> )
-p PROMPT	Erlaubt die Definition eines eigenen Paßwortprompts (%u für Benutzernamen und %h für Rechnername stehen als Variablen zur Verfügung) ( <i>prompt</i> )
-u BENUTZER	Das Kommando wird unter dem angegebenen Benutzer ausgeführt, Standard ist <i>root</i> ; um eine UID anstatt einem Benutzernamen zu benutzen, wird ein Schweinegatter davorgestellt (#512) ( <i>user</i> )
-s	Benutzt die in der Variablen SHELL angegebene Shell ( <i>shell</i> )
-H	Setzt die ansonsten unveränderte Variable HOME auf den Wert des angegebenen Benutzers ( <i>home</i> )
-P	Benutzt die Gruppen des Benutzers anstatt der Gruppen des angegebenen Benutzers ( <i>preserve</i> )
-S	Das Password wird von der Standardeingabe gelesen ( <i>stdin</i> )

### Beispiele

Was kann der aktuelle Benutzer ausführen?

```
ole@enterprise:~> sudo -l
User ole may run the following commands on this host:
    (root) /usr/sbin/useradd
    (root) /usr/sbin/userdel
    (root) /usr/bin/passwd
```

Danach sollte der Benutzer einen Benutzer anlegen, sein Passwort ändern und ihn auch wieder löschen können.

```
ole@enterprise:~> /usr/sbin/useradd theodor
useradd: unable to lock password file
ole@enterprise:~> sudo /usr/sbin/useradd theodor
Password:
ole@enterprise:~> sudo passwd theodor
Changing password for theodor.
New password:
Re-enter new password:
Password changed
ole@enterprise:~> sudo /usr/sbin/userdel theodor
ole@enterprise:~>
```

## 9.3.2 /etc/sudoers

In der Datei */etc/sudoers* wird das Verhalten von *sudo* definiert. Eine Vielzahl von Parametern steuert das Programm. Hier wird nur auf die einfachen Einträge eingegangen.

So könnten eine ganz einfach */etc/sudoers*-Datei aussehen.

### Listing 9.1 /etc/sudoers

```
1: # sudoers file.
```

```

2: #
3: # This file MUST be edited with the 'visudo' command as root.
4: #
5: # See the sudoers man page for the details on how to write a sudoers file.
6: #
7:
8: # Host alias specification
9:
10: # User alias specification
11:
12: # Cmnd alias specification
13:
14: # Defaults specification
15:
16: # User privilege specification
17: root    ALL=(ALL) ALL
18:
19: ole      ALL=/usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/passwd

```

Ein Eintrag besteht aus mehreren Teilen. Dem Benutzer bzw. Gruppennamen, von welchem Rechner aus das Kommando ausgeführt werden soll, welche Kommandos ausgeführt werden dürfen und als welcher Benutzer etwas ausgeführt werden soll.

Die Benutzer aus der Gruppe *gott* dürfen alles ausführen.

```
%gott ALL=(ALL) ALL
```

Jetzt sollen Sie dies auch tun können ohne sich noch einmal extra identifizieren zu müssen.

```
%gott ALL=(ALL) NOPASSWD: ALL
```

Die Mitglieder der Gruppe *users* sollen den Rechner herunterfahren können. Allerdings nur, wenn sie auch direkt auf dem Rechner eingeloggt sind.

```
%users localhost=/sbin/shutdown -h now
```

Der Benutzer *ole* soll neue Benutzer anlegen, ihr Paßwort ändern und die Benutzer auch wieder löschen können.

```
ole      ALL=/usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/passwd
```

Weitere Informationen entnehmen Sie bitten den Manual-Pages (`man 5 sudoers`).

### 9.3.3 visudo

Natürlich kann man die Datei `/etc/sudoers` auch direkt mit einem Editor bearbeiten. Die Benutzung des Kommandos `visudo` bringt dennoch Vorteile. Es ermöglicht die Bearbeitung der Datei auf einer sicheren Art und Weise. `visudo` sperrt die Datei `/etc/sudoers` gegen gleichzeitigen Zugriff. Außerdem kann es einfache Gültigkeits- und Syntaxtests durchführen.

```
visudo [OPTIONEN]
```

Das Kommando verwendet als Editor den Standard-Editor des Systems. Also in den meisten Fällen den `vi`. Weitere Informationen entnehmen Sie bitten den Manual-Pages (`man 8 visudo`).





## IX

Arbeitsblatt  
Rechte

## 9.1

Zu Beginn der Aufgabe sollten Sie sich im Heimatverzeichnis befinden! Wechseln Sie das Verzeichnis nur, falls Sie dazu ausdrücklich aufgefordert werden. Einige Aufgaben lassen sich nicht ausführen! Versuchen Sie in diesen Fällen die Fehlermeldung zu interpretieren.

- 1 Loggen Sie sich als Benutzer *walter* auf der Konsole 2 ein und versuchen Sie die Datei */boot/vmlinuz* zu löschen.
- 2 Kopieren Sie die Datei *shadow* aus dem Verzeichnis */etc* in ihr Heimatverzeichnis!
- 3 Kopieren Sie die Datei *passwd* aus */etc* ins aktuelle Verzeichnis!
- 4 Wem gehört die Datei *passwd* a) in */etc*, b) im aktuellen Verzeichnis?
- 5 Lassen Sie sich aus der Datei */etc/passwd* nur die Zeile anzeigen, die Ihren Benutzernamen enthält!
- 6 Lassen Sie sich nur die Benutzernamen der Datei */etc/passwd* anzeigen!
- 7 Erstellen Sie mit dem *vi* die Datei *adressen*!

Thomas Prizzi Wall 40 24103 Kiel 0431/978767  
 Martin Lecker Eggerstedtstr. 1 24103 Kiel 0431/978234  
 Norbert Nase Nasa-Str. 23 55111 Nasenhausen 0321/918273  
 Zacharias Ackermann Feldweg 33 12345 Kleindorf 06789/1011

- 8 Lassen Sie sich die Adresse von Norbert Nase aus der Datei *adressen* ausgeben !
- 9 Wieviele Adressen stammen aus Kiel ?
- 10 Geben Sie die Adressen sortiert nach Nachnamen aus !
- 11 Geben Sie nur die Nachnamen aus ?
- 12 Wieviele Buchstaben hat der Nachname von Martin ?
- 13 Stellen Sie fest, welche Voreinstellung für das Erstellen (neuer) gewöhnlicher Dateien besteht, ändern Sie sie gegebenenfalls so, daß alle (*user*, *group*, *other*) das Recht zum Lesen erhalten und sonst keines!
- 14 Kopieren Sie die Datei *vdir* aus */usr/bin* in Ihr Heimatverzeichnis!
- 15 Erstellen Sie mittels *cat* eine Datei namens *indertat* (Inhalt: Ein beliebiges Sprichwort)!
- 16 Prüfen Sie den Inhalt Ihres Heimatverzeichnis und damit die Rechte der Dateien *vdir* und *indertat*. Vergleichen Sie das Ergebnis mit den Voraussetzungen, die Sie in Aufgabe 13 geschaffen haben!
- 17 Verändern Sie die Voreinstellung zum Erstellen neuer gewöhnlicher Dateien so, daß Sie (Besitzer) diese Dateien Lesen und Verändern können, alle sonstigen Teilnehmer (Gruppe, Andere) nur Lesen können.
- 18 Kopieren Sie die Datei *mv* aus */bin* in ihr Heimatverzeichnis.
- 19 Erstellen Sie mit *cat* die Datei *hilfe* (Inhalt: Eine Kurzbeschreibung von 3 Linuxbefehlen z. B.: *cp* kopiert Dateien, je Befehl eine Zeile!)
- 20 Prüfen Sie erneut den Inhalt Ihres Heimatverzeichnis und vergleichen Sie das Ergebnis mit Aufgabe 17 !
- 21 Verändern Sie die Rechtetabelle von *mv* so, daß kein Recht mehr enthalten ist: -----

## IX

Arbeitsblatt  
Rechte

## 9.2

- 22** Versuchen Sie die Rechtetabelle von `mv` so zu ändern, daß Sie als Besitzer die Datei ausführen können, und so wenig wie möglich Rechte gesetzt sind !
- 23** Versuchen Sie die Datei `mv` und `vdir` im Heimatverzeichnis zu löschen! Geht das? Warum geht es bzw. warum nicht?
- 24** Verändern Sie die Rechte der Datei `.bash_history` so, daß nur Sie sie Lesen, Ändern und Ausführen können!
- 25** Verändern Sie die Rechte der Datei `adressen` so, daß die Gruppe und die Anderen Sie nicht Lesen oder Ändern können!
- 26** Prüfen Sie die Rechtetabelle aller Dateien Ihres Heimatverzeichnis!
- 27** Legen Sie ein Verzeichnis `rights` an!
- 28** Prüfen Sie die Rechtetabelle am neuen Verzeichnis!
- 29** Erstellen Sie mittels `cat` eine Textdatei namens `uhr`  
Inhalt: `date +"Es ist %H:%M:%S"`
- 30** Prüfen Sie die Rechtetabelle der neuen Datei! Was fällt Ihnen auf?
- 31** Machen Sie die Datei ausführbar; es ist damit ein selbsterstelltes Shell-Script!
- 32** Kopieren Sie die Datei `uhr` und geben Sie ihr dabei den Namen `bimbam`!
- 33** Verschenken Sie `bimbam` an den Superuser (geben Sie ihr einen anderen Besitzer)!

.....  
*Notizen:*

.....

X

Arbeitsblatt  
Datei-Rechte

10.1

Situation: Sie sind User **tapico** und gehören zur Gruppe **group** !

**1** Annahme: Für das Verzeichnis **/usr** haben Sie die Rechte **x** und **r**, an allen Dateien das Recht **r**. Sie können:

1. ☐ in das Verzeichnis wechseln
2. ☐ Dateien in diesem Verzeichnis löschen
3. ☐ Dateien in diesem Verzeichnis umbenennen
4. ☐ Dateien aus diesem Verzeichnis verschieben
5. ☐ Dateien aus diesem Verzeichnis anschauen
6. ☐ Dateien aus diesem Verzeichnis kopieren

**2** Können Sie im Wurzelverzeichnis **/** Dateien

1. ☐ löschen
2. ☐ neu anlegen
3. ☐ umbenennen

**3** Im Dateisystem gebe es im Verzeichnis **/usr** die (ausführbare) Datei **uhr**. In **/usr** haben Sie die Rechte **x** und **r**. Der Befehl **ls -l uhr** gibt Ihnen folgende Ausgabe:

```
-rwxr--r-- root other 124 Mar 4 16:14 uhr
```

Sie können die Datei

1. ☐ löschen
2. ☐ umbenennen
3. ☐ anschauen
4. ☐ drucken
5. ☐ kopieren
6. ☐ starten

**4** Welche Rechte haben Sie in Ihrem Heimatverzeichnis, welche Rechte andere Mitglieder?

**5** Woran kann es liegen, daß eine Datei nicht löscher ist?

X

Arbeitsblatt  
Datei-Rechte

10.2

**6** In Ihrem Heimatverzeichnis befindet sich die Datei **alpha**. Sie gehört **root** (Gruppe **other**). Die Rech-  
tetabelle ist **-rwxr--r--**. Sie können die Datei

1. ☐ löschen
2. ☐ umbenennen
3. ☐ anschauen
4. ☐ ändern
5. ☐ kopieren
6. ☐ verschieben

**7** Wo befindet sich der physikalische Ort des Dateisystems **proc**?

.....  
*Notizen:*

.....

## XI

Arbeitsblatt  
Fallstudie: Rechte

## 11.1

Die Firma AMOV GmbH möchte ihre Daten auf einem Server speichern. Dazu soll als erstes ein Verzeichnisbaum unter dem Verzeichnis `/daten` aufgebaut werden. Dabei müssen die Rechte auf die einzelnen Bereiche ermittelt werden.

Stellvertretend für die Mitarbeiter mit gleichen Rechten nehmen Sie zur Durchführung der Fallstudie bitte die folgenden Mitarbeiter.

**Dick Tator** ist der Chef. Er hat von internen Abläufen eigentlich keine Ahnung, will aber trotzdem alles wissen. Eigentlich müßte er nur im Bereich der Auftragsbearbeitung arbeiten.

**Petra Pleite-Geier** arbeitet als Geschäftsführerin hart und braucht überall Zugriff um im Notfall alles selbst zu machen.

**Manuela Magersucht** ist die Dame aus der Buchhaltung und auch für die Kundendatenbank sowie für die Auftragsbearbeitung zuständig.

**Maria Krohn** ist ebenfalls in der Buchhaltung tätig, daneben ist sie aber auch für die Lohnabrechnung zuständig.

**Carlo Calvados** ist der feurige Spanier aus der Forschungsabteilung. Außer Manuela Magersucht kennt er nur seine Arbeit, die er zusammen mit seinen Kollegen im Forschungsordner speichert.

**Karl Lagerhalle**, der Lagerverwalter, braucht den Zugriff auf die Lagerdatenbank.

**Michael Schrauber** braucht als Produktionsleiter Zugriff auf den Lagerbestand, die Produktionsdaten und die Aufträge.

**Peter Proxy** als Systemadministrator hat als `root` eigentlich überall seine Nase drin, arbeitet als normaler User aber nur in dem Dokumentationsbereich, den alle Lesen dürfen.

Alle Mitarbeiter sollen Zugriff auf den Pool haben um Daten auszutauschen.

- 1 Erstellen Sie die Liste der Bereiche mit den zugehörigen Verzeichnissen und Gruppen.
- 2 Erstellen Sie eine Liste der Benutzer mit ihren UIDs, Heimatverzeichnissen und Gruppenzugehörigkeiten. (UIDs ab 700).
- 3 Legen Sie die Benutzer an.
- 4 Legen Sie die Gruppen an.
- 5 Erstellen Sie aus Ihren Überlegungen die Verzeichnisstruktur und vergeben Sie die passenden Rechte.
- 6 Testen Sie Ihre Installation.

.....  
*Notizen:*

.....

# Kapitel 10

## Partitionen und Dateien

*Nur eine gelöschte Datei ist eine gute Datei. Und nur eine formatierte Platte ist eine gute Platte.*

*Martin Bethke*

### 10.1 Partitionen

Partitionen ermöglichen das Aufteilen in einer Festplatte in mehrere unabhängige Einheiten. In Abschnitt 1.3.3 bin ich schon auf die Partitionen eingegangen. Der folgende Abschnitt wird sich mit der Einrichtung und Pflege von Partitionen beschäftigen.

#### 10.1.1 fdisk

Um Partitionen einzurichten wird das Programm `fdisk` verwendet. Sie sollten sich bei der Arbeit mit `fdisk` immer Notizen machen.

```
fdisk [-l] [-u] [PLATTE]
fdisk -s PARTITION
```

##### Optionen

<code>-u</code>	Größenangabe in Sektoren anstatt in Zylindern
<code>-l</code>	Ausgabe der Partitionstabelle
<code>-s</code>	Ausgabe der Größe der angegebenen Partition

Ein Zylinder (*cylinder*) ist die Menge aller Spuren, die sich an der gleichen Stelle auf jeder Platte befinden. Als Spur (*track*) bezeichnet man die konzentrischen Kreise auf einer Platte der Festplatte.

Sie können Linux auf jeder Festplatte und jeder primären Partition installieren. Wenn Sie keine Festplatte angeben, wird standardmäßig das Gerät `/dev/hda` genommen.

Wenn Sie `fdisk` benutzen, können Sie mit der Taste `<m>` sich eine Übersicht über die Hilfe anzeigen lassen.

```
root@defiant:/ > fdisk /dev/hda
```

Kommando (m für Hilfe): m

Kommando    Bedeutung

- |   |   |
|---|---|
| a | (De)Aktivieren des bootbar-Flags            |
| b | »bsd disklabel« bearbeiten                  |
| c | (De)Aktivieren des DOS Kompatibilitätsflags |
| d | Eine Partition löschen                      |
| l | Die bekannten Dateisystemtypen anzeigen     |

```

m   Dieses Menü anzeigen
n   Eine neue Partition anlegen
o   Eine neue leere DOS Partitionstabelle anlegen
p   Die Partitionstabelle anzeigen
q   Ende ohne Speichern der Änderungen
s   Einen neuen leeren »Sun disklabel« anlegen
t   Den Dateisystemtyp einer Partition ändern
u   Die Einheit für die Anzeige/Eingabe ändern
v   Die Partitionstabelle überprüfen
w   Die Tabelle auf die Festplatte schreiben und das Programm beenden
x   Zusätzliche Funktionen (nur für Experten)

```

Kommando (m für Hilfe):

Um nun eine Partition anlegen zu können, müssen Sie drei Schritte durchlaufen.

1. Größe der Partition angeben.
2. Den Typ der Partition festlegen.
3. Die Veränderung in die Partitionstabelle schreiben.

Für jede Festplatte, die partitioniert werden soll, muß `fdisk` einmal gestartet werden.

Wenn Sie logische Laufwerke anlegen wollen, müssen Sie nach der Einrichtung der erweiterten Partition den Rechner erst wieder neu starten. Mit `fdisk` wird nämlich nur die Partitionstabelle geschrieben. Das Einrichten der Partitionen erfolgt erst beim Neustarten des Rechners.

Denken Sie daran, `fdisk` für Linux erstellt oder löscht nur Partitionen für das Linux-Betriebssystem. Andere Betriebssysteme können diese Partitionen meistens nicht sehen oder mit Ihnen arbeiten.

Nachdem Sie die Partitionen angelegt haben, muß die Swap-Partition aktiviert werden, die für die Wurzel vorgesehene Partition ausgewählt und die Partitionen formatiert werden.

## Beispiele

Der folgende Befehl zeigt den Zustand der Festplatte `/dev/hda` an.

```
root@defiant:/ > fdisk -l
```

```
Festplatte /dev/hda: 255 Köpfe, 63 Sektoren, 592 Zylinder
Einheiten: Zylinder mit 16065 * 512 Bytes
```

Gerät	boot.	Anfang	Ende	Blöcke	Id	Dateisystemtyp
/dev/hda1	*	1	261	2096451	6	FAT16
/dev/hda2		262	392	1052257+	5	Erweiterte
/dev/hda3		393	576	1477980	83	Linux
/dev/hda4		577	592	128520	82	Linux Swap
/dev/hda5		262	392	1052226	6	FAT16

Wenn das Programm `fdisk` läuft, zeigt der Befehl `l` zeigt die Liste der Dateisystemtypen.

0	Leer	1c	Verst. Win95 FA	65	Novell Netware	bb	Boot Wizard hid
1	FAT12	1e	Verst. Win95 FA	70	DiskSecure Mult	c1	DRDOS/sec (FAT-
2	XENIX root	24	NEC DOS	75	PC/IX	c4	DRDOS/sec (FAT-
3	XENIX usr	39	Plan 9	80	Old Minix	c6	DRDOS/sec (FAT-
4	FAT16 <32M	3c	PartitionMagic	81	Minix / old Lin	c7	Syrinx
5	Erweiterte	40	Venix 80286	82	Linux Swap	da	Non-FS data
6	FAT16	41	PPC PReP Boot	83	Linux	db	CP/M / CTOS / .
7	HPFS/NTFS	42	SFS	84	OS/2 verst. C:-	de	Dell Utility



8	AIX	4d	QNX4.x	85	Linux erweitert	df	BootIt
9	AIX bootfähig	4e	QNX4.x 2nd part	86	NTFS volume set	e1	DOS access
a	OS/2 Bootmanage	4f	QNX4.x 3rd part	87	NTFS volume set	e3	DOS R/0
b	Win95 FAT32	50	OnTrack DM	8e	Linux LVM	e4	SpeedStor
c	Win95 FAT32 (LB	51	OnTrack DM6 Aux	93	Amoeba	eb	BeOS fs
e	Win95 FAT16 (LB	52	CP/M	94	Amoeba BBT	ee	EFI GPT
f	Win95 Erw. (LBA	53	OnTrack DM6 Aux	9f	BSD/OS	ef	EFI (FAT-12/16/
10	OPUS	54	OnTrackDM6	a0	IBM Thinkpad hi	f0	Linux/PA-RISC b
11	Verst. FAT12	55	EZ-Drive	a5	FreeBSD	f1	SpeedStor
12	Compaq Diagnost	56	Golden Bow	a6	OpenBSD	f4	SpeedStor
14	Verst. FAT16 <3	5c	Priam Edisk	a7	NeXTSTEP	f2	DOS secondary
16	Verst. FAT16	61	SpeedStor	a9	NetBSD	fd	Linux raid auto
17	Verst. HPFS/NTF	63	GNU HURD / SysV	b7	BSDI fs	fe	LANstep
18	AST SmartSleep	64	Novell Netware	b8	BSDI swap	ff	BBT
1b	Verst. Win95 FA						

Mit `fdisk` kann der Dateisystemtyp einer Partition unter dem Menüpunkt `t` nachträglich geändert werden. Der Dateisystemtyp wird auch beim Formatieren in der Partitionstabelle eingetragen. Wichtig ist diese Funktion dann, wenn eine zerstörte Partitionstabelle rekonstruiert werden soll.

## 10.2 Master Boot Record

Nach einer im BIOS festgelegten Suchreihenfolge (siehe dazu auch Abschnitt 16.1.1) werden die Datenträger des Rechners nach ihren *Master Boot Record* durchsucht. Der *Master Boot Record* ist der erste Sektor auf jedem Datenträger.

### 10.2.1 Aufbau des MBR

Da der MBR genau einen Sektor umfaßt ist er normalerweise 512 Bytes groß. Er besteht aus drei Teilen: Dem *Mini-Startprogramm*, der *Partitionstabelle* und der *Magischen Nummer*.

#### Mini-Startprogramm

Dieses kleine Programm, das in den ersten 446 Bytes des MBR liegt, sorgt dafür das ein Betriebssystem geladen wird. Als Standardeinstellung steht dort ein Programm, das die Ausführung einfach an den Boot-Record der als aktiv gekennzeichneten Partition übergibt. Hier kann sich auch der erste Teil von Bootloadern wie GRUB (12.2) oder LILO (12.3) befinden.

#### Partitionstabelle

Die Partitionstabelle nimmt insgesamt 64 Bytes ab Speicherstelle `0x1BE` ein. Hier können maximal 4 Partitionen eingetragen werden für die jeweils 16 Byte zur Verfügung stehen. Dabei teilen sich die 16 Byte wie in Tabelle 10.1 zu sehen auf.

Das *Bootflag* kennzeichnet, ob die Partition aktiv ist und daß ihr Boot-Record einen Bootloader enthält. Normalerweise steht dort der Wert `0x00`. Ist die Partition aktiv dann steht an dieser Stelle der Wert `0x80`. Natürlich kann, was zwar falsch aber trotzdem möglich ist, eine Partition als aktiv gekennzeichnet sein, obwohl sie keinen funktionsfähigen Boot-Record besitzt.

In der Partitionstabelle wird auch festgehalten, für welches Dateisystem die Partition gedacht ist bzw. ob es sich um eine erweiterte Partition handelt. So steht z. B. der Wert `0x83` für eine Linuxpartition, egal ob `ext2`, `ext3`, ReiserFS oder andere. Dagegen steht `0x82` für die Swap-Partition.

Die 6 Bytes für den Partitionsanfang und das Partitionsende nach der CHS-Zählung sind heute veraltet. Mit dieser Notations sind maximal Platten von 8 GB Größe verwaltbar, wenn man davon ausgeht, daß diese 256 Leseköpfe haben. Da Partitionen heute sowieso nur nach an Zylindergrenzen festgemacht werden, haben

Größe	Typ	Name	Erklärung
1 Byte	char	boot_flag	Kennzeichnung für Bootpartition
1 Byte	char	begin_head	Partitionsanfang nach CHS
1 Byte	char	begin_sector	Partitionsanfang nach CHS
1 Byte	char	begin_cylinder	Partitionsanfang nach CHS
1 Byte	char	partition_type	Partitionstyp
1 Byte	char	end_head	Partitionsende nach CHS
1 Byte	char	end_sector	Partitionsende nach CHS
1 Byte	char	end_cylinder	Partitionsende nach CHS
4 Byte	long	start_sector	Beginn der Partition nach LBA
4 Byte	long	sector_count	Größe der Partition

Tabelle 10.1: Aufbau eines Partitionseintrags im MBR

diese Werte keine weitere Bedeutung. Am Ende des Partitionseintrags liegen 8 Byte, die Anfang und Anzahl der Sektoren der Partition anzeigen. Mit dieser Methode sind Platten bis zu 2 TB (2048 GB) verwaltbar.

Übrigens fehlt bei Disketten die Partitionstabelle im MBR. Wer würde auch schon auf die Idee kommen eine Diskette zu partitionieren?

### Magische Nummer

Die letzten zwei Bytes bilden die magische Nummer des Master Boot Records. Hier steht immer der Wert 0xAA55 drin.

### 10.2.2 MBR anschauen

An einem Beispiel wird es deutlicher. Schauen wir uns doch mal den Master Boot Record einer Festplatte genauer an. Dazu müssen wir ihn von der Platte auslesen. Das direkte Auslesen und Schreiben von Datenträger erlaubt der Befehl `dd` (4.5.4). Die Sektor- und damit die physikalische Blockgröße auf einer Diskette und einer Festplatte ist 512 Bytes. Der MBR umfaßt nur einen Sektor und zwar den ersten auf Platte.

Wir lesen also von der ersten IDE-Platte (`if=/dev/hda`) und schreiben die gelesenen Daten in die Datei `mbr.dmp` (`of=mbr.dmp`). Die Blockgröße beträgt 512 Bytes (`bs=512`) und es wird genau ein Block (`count=1`) gelesen.

```
defiant:~ # dd if=/dev/hda of=mbr.dmp bs=512 count=1
1+0 Records ein
1+0 Records aus
```

In der Datei `mbr.dmp` befindet sich nun eine direkte Kopie des MBR. Diese Datei ist eine Binärdatei. Im Gegensatz zu einer Textdatei, die hauptsächlich nur lesbare Zeichen enthält, können alle Werte eines Byte – zwischen 0 und 255 – als Zeichen vorkommen. Da dadurch auch viele Steuerzeichen vorkommen, sollten Sie solche Dateien niemals mit `cat` oder einem anderen Textbetrachter anzeigen lassen.

Speziell zur Betrachtung solcher Binärdateien wurde der Befehl `od` (7.1.3) entwickelt. Dieser stellt die Bytes der Datei in verschiedenen Formen (dezimal, octal, hexadezimal oder auch als Zeichen und Escape-Sequenzen) dar.

Für diesen Fall lassen wir uns den Dateiinhalt in hexadezimaler Notation anzeigen und zwar Byte für Byte (`-t x1`). Da pro Zeile 16 Byte ausgegeben werden, sollten Sie die Ausgabe an den Pager `less` weiterleiten um sich die Ausgabe genau anschauen zu können.

```
defiant:~ # od -t x1 mbr.dmp | less
```

Eigentlich interessieren uns erst die Bytes ab Nummer 446 (Erstes Byte ist die Nummer 0). Die Nummerierung vor jeder Zeile ist in oktaler Schreibweise verfaßt. Das gesuchte Byte hat also in oktaler Schreibweise

die Nummer 0676. Wir können uns daher bei der Anzeige der Partitionierungsdaten auf die letzten 6 Zeilen beschränken. Hier hilft uns der Befehl `tail` weiter. Dieser zeigt die letzten Zeilen (normalerweise 10) einer Datei oder eines Datenstroms an.

```
defiant:~ # od -t x1 mbr.dmp | tail -6
0000660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 01
0000700 01 00 0b ef 7f b7 3f 00 00 00 41 83 65 00 00 00
0000720 41 b9 83 ef ff 59 90 be 65 00 10 35 60 00 00 00
0000740 c1 5a 0f ef ff ff a0 f3 c5 00 90 be 65 00 00 00
0000760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa
0001000
```

Partition	BF	AK	AS	AC	PT	EK	ES	EC	Start-Sektor	Anzahl Sektoren
1	80	01	01	00	0b	ef	7f	b7	3f 00 00 00	41 83 65 00
2	00	00	41	b9	83	ef	ff	59	90 be 65 00	10 35 60 00
3	00	00	c1	5a	0f	ef	ff	ff	a0 f3 c5 00	90 be 65 00
4	00	00	00	00	00	00	00	00	00 00 00 00	00 00 00 00

Auf der Festplatte sind drei Partitionen eingetragen. Die erste Partition ist aktiv, da der Wert 0x80 im ersten Byte steht. Im Byte für den Partitionstyp steht der Wert 0x0b. Es handelt sich daher um eine "Win95 FAT32"-Partition.

Die Angaben für Start-Sektor und die Anzahl der Sektoren sind als 4-Byte-Wert angegeben. Dabei steht das niedrigwertigste Byte links. Also steht hinter der Bytefolge 3f 00 00 00 die hexadezimale Zahl 0x3f (dezimal 63) und hinter der Bytefolge 41 83 65 00 die Zahl 0x658341 (dezimal 6.652.737). Da jeder Sektor 512 Bytes groß ist hat die Partition eine Größe von ca. 3,2 GB.

Die zweite Partition ist nicht aktiv. Sie ist vom Typ Linux (0x83), beginnt bei Sektor 6.667.920 (0x65be90) und ist ca. 3 GB (0x603510 Sektoren) groß.

Bei der dritten Partition handelt es sich um eine erweiterte Partition (0x0f = "Win95 Erweitert (LBA)"). Sie beginnt bei Sektor 12.972.960 (0xc5f3a0) und ist ca. 3,2 GB (0x65be90 Sektoren) groß.

## 10.3 Mounten

Um mit Dateien zu arbeiten, müssen diese erreichbar sein. Da jede Partition und jedes Speichergerät sein eigenes Dateisystem besitzt, müssen diese an einer zentralen Stelle zusammengeführt werden. Bei Windows wird dies durch die Laufwerksbuchstaben realisiert, die unter *Arbeitsplatz* liegen. Bei Linux muß eine Partition immer als Wurzel / ansprechbar sein. Die anderen Partitionen werden dann in das Verzeichnissystem dieser Partition eingebunden. Das Verzeichnis, das die Wurzel des jeweiligen Dateisystems im Verzeichnissystem repräsentiert, wird als **Mount Point** bezeichnet und der Vorgang des Einbindens als **Mounten**. Dabei ist das Einbinden nicht nur auf die Linux-Dateisysteme wie ext2 beschränkt. Es können auch virtuelle Dateisysteme wie `/proc` eingebunden werden oder sogar Dateisysteme, die sich auf anderen Rechnern im Netz befinden. Eine Liste der möglichen Dateisysteme finden Sie im Abschnitt 10.4.

### 10.3.1 mount

Der `mount`-Befehl bindet ein Dateisystem in den aktuellen Verzeichnisbaum ein.

```
mount [OPTIONEN] [GERÄT] MOUNTPOINT
```

#### Optionen

-a	Mounte alle Dateisysteme, die in <code>/etc/fstab</code> aufgeführt werden.
-f	Überprüfe ob das angegebene Dateisystem gemountet werden kann
-n	Schreibe die Mount-Informationen nicht in die Datei <code>/etc/mtab</code>
-o OPTION	Modifikatoren für den Mountvorgang (siehe Tabelle 10.2)
-r	Mounten nur mit Leseberechtigung (read only)
-t DSTYP	Typ für das zu mountende Dateisystems
-v	Angabe der Mount-Informationen
-w	Mounten mit Schreibberechtigung (Standard)

Option	Aktion
<b>async</b>	Der I/O-Zugriff erfolgt asynchron
<b>atime</b>	Der letzte Zugriff wird in der Inode festgehalten
<b>auto</b>	Kann mit der Option <b>-a</b> gemountet werden
<b>defaults</b>	Entspricht: <b>rw, suid, dev, exec, auto, nouser</b> und <b>async</b> .
<b>dev</b>	Interpretiert spezielle Zeichen- und Block-Geräte
<b>exec</b>	Binärdateien können ausgeführt werden
<b>noatime</b>	Der letzte Zugriff wird nicht in der Inode festgehalten (schnellerer Zugriff)
<b>noauto</b>	Wird nicht automatisch gemountet
<b>nodev</b>	Interpretiert nicht spezielle Zeichen- und Block-Geräte
<b>noexec</b>	Binärdateien werden nicht ausgeführt
<b>nosuid</b>	Das <b>suid</b> - oder <b>sgid</b> -bit wird nicht ausgeführt
<b>nouser</b>	Ein Benutzer kann das Dateisystem nicht mounten
<b>remount</b>	Mountet ein Dateisystem erneut (Wird verwendet um Modifikatoren zu ändern)
<b>ro</b>	Das Dateisystem wird nur lesbar gemountet (read only)
<b>rw</b>	Das Dateisystem wird schreibbar gemountet (read write)
<b>suid</b>	Erlaubt die Benutzung des <b>suid</b> - oder <b>sgid</b> -bits
<b>sync</b>	Alle E/A-Aktionen sollten synchron erfolgen
<b>user</b>	Ein Benutzer darf das Dateisystem mounten

Tabelle 10.2: Modifikatoren für `mount -o` und `/etc/fstab`

Der Aufruf von `mount` ohne Parameter zeigt die Liste aller Dateisysteme an, die in diesem Moment gemountet sind.

```
tapico@defiant:~ > mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
/dev/hda1 on /c type vfat (rw)
/dev/hda5 on /d type vfat (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=0620)
/dev/fd0 on /a type vfat (rw,noexec,nosuid,nodev,user=tapico)
```

Beim Einbinden eines Dateisystems mit `mount` wird die Datei `/etc/fstab` ausgewertet. Wenn das Gerät in der Datei angegeben ist, so reicht für das Mounten die Angabe des Mount Points aus.

```
mount /floppy
```

Außerdem werden die Angaben für dieses Gerät aus der Datei entnommen. Änderungen der gegebenen Modifikatoren kann über den Schalter `-o` erfolgen.

## Beispiele

Ist der Mount Point in der `/etc/fstab` eingetragen, so reicht ein

```
mount /a
aus.
```

Ist dies nicht der Fall, so kann das Gerät über

```
mount -t vfat /dev/fd0 /a
```

gemountet werden. Dies ist aber nur **root** erlaubt.

Um einen Datenträger, der nur lesbar (Read Only) gemountet wurde, auch schreibbar zu machen, muß der Datenträger nicht extra ausgebonden werden. Er kann auch mit anderen Optionen erneut gemountet werden. Dazu wird die Option `remount` benutzt.

```
enterprise:~ # mount -o remount,rw /dev/hda6
```

### 10.3.2 remount

Das erneute Mounten eines Datenträgers mit unterschiedlichen Optionen kann auch über das Kommando `remount` erfolgen. Allerdings ist der Befehl kein Programm sondern nur eine Funktion. (Siehe Abschnitt 15.1.3.)

```
enterprise:~ # type remount
remount is a function
remount ()
{
    /bin/mount -o remount,${1+"$@"}
}
```

Die richtige Form des erneuten Einbinden eines Datenträgers im Schreibmodus wäre dann:

```
enterprise:~ # remount rw /dev/hda6
```

### 10.3.3 umount

Der `umount`-Befehl<sup>1</sup> entfernt ein Dateisystem aus dem aktuellen Verzeichnisbaum.

```
umount [OPTIONEN] [GERÄT] [MOUNTPOINT]
```

#### Optionen

<code>-a</code>	Unmounte alle Dateisysteme, die in <code>/etc/mtab</code> aufgeführt werden.
<code>-n</code>	Schreibe die Mount-Informationen nicht in die Datei <code>/etc/mtab</code>
<code>-t DSTYP</code>	Typ für das zu unmountende Dateisystems

Durch den Befehl `umount` wird auch der Befehl `sync` aufgerufen, damit vor dem Entfernen des Dateisystems dieses mit dem Cache abgeglichen wird.

### 10.3.4 sync

Durch den Befehl `sync` werden die sich im Cache befindlichen Daten auf die Platte geschrieben und damit die Festplatte auf den aktuellen Stand gebracht.

```
sync
```

Der `sync`-Befehl ruft einfach die Kernel-Prozedur `sync` auf. Diese wird auch vom Befehl `umount` zum sauberen Aushängen von Dateisystemen benutzt

### 10.3.5 /etc/fstab

Die Datei `/etc/fstab` wird von den Befehlen `fsck`, `mount` und `umount` benutzt.

```
tapico@defiant:~ > cat /etc/fstab
/dev/hda3 / ext2 defaults 1 1
/dev/hda1 /c vfat defaults 0 0
/dev/hda5 /d vfat defaults 0 0
/dev/hda4 swap swap defaults 0 0
/dev/hdc /cdrom iso9660 ro,noauto,user 0 0
/dev/fd0 /floppy auto noauto,user 0 0
/dev/fd0 /a vfat noauto,user 0 0
proc /proc proc defaults 0 0
```

<sup>1</sup>Das "n" in `umount` ist im Lauf der Zeit aus dem Befehl verschollen.

Die Spalten bedeuten von links nach rechts:

1. Der physikalische Ort des Dateisystems oder Blockgeräts.
2. Der Mount Point. An dieser Stelle wird die Wurzel des Dateisystems in die Verzeichnisstruktur eingebaut.
3. Der Typ des Dateisystems
4. Die Optionen, die beim Mounten des Systems benutzt werden.
5. Die Nummer legt fest, ob das Dateisystem bei einem Backup durch `dump` gesichert werden soll.
6. Die Nummer legt fest, in welcher Reihenfolge `fsck` die Dateisysteme prüft.

### 10.3.6 /etc/mtab

In dieser Datei werden alle gerade gemounteten Dateisysteme aufgelistet. Der Befehl `mount` ohne Parameter benutzt diese Datei für seine Ausgabe. Die Befehle `mount` und `umount` verändern die Datei.

```
tapico@defiant:~ > cat /etc/mtab
/dev/hda3 / ext2 rw 0 0
proc /proc proc rw 0 0
/dev/hda1 /c vfat rw 0 0
/dev/hda5 /d vfat rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=0620 0 0
/dev/fd0 /a vfat rw,noexec,nosuid,nodev,user=tapico 0 0
```

Die Ähnlichkeit mit der Ausgabe des Befehls `mount` ist nicht zu übersehen.

## 10.4 Das Dateisystem

Das Linux-Dateisystem organisiert die Dateien und Verzeichnisse in einer Hierarchie. Es ermöglicht das Speichern von und den zufälligen Zugriff auf Dateien, egal ob sich diese Daten auf der Festplatte, der Floppy, einer CD-ROM oder im Netzwerk befinden. Linux unterstützt mehrere Dateisysteme. Einige von ihnen werden in der folgenden Liste aufgezählt.

- **ext2** - Second Extended Filesystem, das aktuelle Standard-Dateisystem für LINUX
- **ext3** - Third Extended Filesystem, das neue Dateisystem für LINUX mit Journaling-Funktionen
- **bfs** - Das SCO BFS Dateisystem.
- **coherent** - Dateisystem der UNIX-Version Coherent (war lange Jahre eine kostengünstige Alternative zu kommerziellen UNIX-Produkten). Wird wohl in Zukunft nicht mehr unterstützt werden.
- **ffs** - Fast Filesystem, das Dateisystem vom Amiga.
- **hpfs** - High Performance Filesystem, das Dateisystem von OS/2. Es kann zur Zeit nur lesend genutzt werden.
- **iso9660** - ISO-9660-Dateisystem, das für die meisten CD-ROMs benutzt wird.
- **jfs** - Das Journaling-Filesystem von IBM.
- **minix** - Minix Dateisystem, Vorläufer von ext. War das ursprüngliche Dateisystem von LINUX.
- **msdos** - Das FAT16-Dateisystem von MS-DOS.
- **ncpfs** - Das Dateisystem der Novell-Server.

- **nfs** - Network File System (NFS). Gestattet den Zugriff auf Dateien via Netzwerk.
- **ntfs** - Dateisystem für Windows NT (New -ähem- Technology). Windows 2000/XP im Moment leider nur lesend.
- **proc** - Das virtuelle Dateisystem, daß der Linux-Kernel für die Verwaltung der Prozesse verwendet.
- **reiserfs** - Das Reiser-Filesystem mit Journaling-Funktion
- **swap** - Das Linux Swap Dateisystem.
- **sysv** - Dateisystem gemäß System V.
- **ufs** - Uniform Filesystem, das Dateisystem, was BSD, SunOS und NeXTstep verwenden. Es wird nur lesend unterstützt.
- **umsdos** - Dateisystem, mit dessen Hilfe LINUX in einer DOS-Partition installiert werden kann. Es ist allerdings sehr langsam.
- **vfat** - Virtual FAT Filesystem, die Erweiterung der FAT für die Verwendung langer Dateinamen.
- **xenix** - Dateisystem von Xenix (eine Entwicklung von Microsoft für den PC. Wurde in späteren Jahren von der Firma SCO übernommen. Xenix war jahrelang das Standard-UNIX-System für den PC). Wird wohl in Zukunft nicht mehr unterstützt werden.

Seit Kernel 2.1.21 sind diese Dateisysteme nicht mehr dabei.

- **ext** - Extended Filesystem, ein Vorläufer von ext2
- **xiafs** - Ein nach seinem Entwickler benanntes Dateisystem, das ext2 sehr ähnlich ist.

Nachdem Sie also die Partitionen angelegt haben, muß die Partition noch formatiert werden. Dafür wird der Befehl **mkfs** verwendet.

### 10.4.1 mkfs

Das Kommando **mkfs** formatiert eine Partition mit dem angegebenen Dateisystem.

**mkfs** [-t DSTYP] GERÄT [BLOCKS]

#### Optionen

-t DSTYP	Legt den Dateisystemtyp fest
-v	Zeigt alle Kommandos an
-c	Überprüft nach defekten Blöcken vorm Anlegen des Dateisystems
-l DATEINAME	Benutzt die angegeben Datei als Liste für bekannte defekte Blöcke

Normalerweise ruft **mkfs** andere Kommandos auf, die für das jeweilige mit -t angegebene Dateisystem programmiert wurden. Z. B. sind dies **mkfs.ext2**, **mkfs.msdos** und **mkfs.minix**. Sie können diese Programme auch direkt nutzen. Natürlich darf für diesen Vorgang der Datenträger nicht gemountet sein. Normalerweise kann **mkfs** nur von dem Benutzer **root** ausgeführt werden. Um normalen Benutzern trotzdem das Formatieren von Disketten zu erlauben, muß **root** ein passendes Skript (9.1.4) zur Verfügung stellen.

#### Beispiele

Der folgenden Befehl erstellt das ext2-Dateisystem auf der dritten Partition der ersten Festplatte.

```
mkfs -t ext2 /dev/hda3 1477980
```

Heute wird die Anzahl der Blöcke automatisch ermittelt. Es kann aber **mkfs**-Versionen geben, die die Angabe der Blöcke verlangt. Mit dem Befehl **fdisk** kann diese ermittelt werden. Dabei muß man sehr vorsichtig sein, da falsche Angaben zu einem Datenverlust auf einer anderen Partition führen kann.

### 10.4.2 fdformat

Für die Formatierung der Disketten kann auch das Kommando `mkfs` verwendet werden. Allerdings sollte man vorher mit `fdformat` eine Low-Level-Formatierung durchführen, die die Sektoren und Spuren auf der Diskette anlegt.

`fdformat [OPTIONEN] GERÄT`

Dabei wird für das Diskettenlaufwerk A: die Bezeichnung `/dev/fd0` und für Laufwerk B: die Bezeichnung `/dev/fd1` verwendet. Es können über andere Gerätenamen verschiedene Formatierungen durchgeführt werden. Hier eine Liste von Gerätenamen<sup>2</sup> für das Laufwerk A:.

```
root@defiant:/dev > ls fd0*
fd0          fd0h1440  fd0h360  fd0h880  fd0u1600  fd0u1760  fd0u3200  fd0u720
fd0CompaQ    fd0h1476  fd0h410  fd0u1040  fd0u1680  fd0u1840  fd0u3520  fd0u800
fd0d360      fd0h1494  fd0h420  fd0u1120  fd0u1722  fd0u1920  fd0u360   fd0u820
fd0h1200     fd0h1600  fd0h720  fd0u1440  fd0u1743  fd0u2880  fd0u3840  fd0u830
```

Ein normaler Benutzer kann allerdings nur das Standardformat (`/dev/fd0` und `/dev/fd1`) für die Formatierung verwenden. Außerdem darf das Laufwerk nicht gemountet sein, was bei einer vorformatierten Diskette durchaus möglich ist.

## 10.5 Swap: Der Auslagerungsspeicher

Arbeitsspeicher ist auch heute noch teuer. Deshalb benötigen gerade bei Multitasking-Systemen die Programme oft mehr Speicher als vorhanden ist. Um dem gegenzuwirken arbeiten die Betriebssysteme nicht mit dem physikalischen Arbeitsspeicher sondern mit dem virtuellen Arbeitsspeicher. Der virtuelle Arbeitsspeicher setzt sich aus dem RAM und einem Speicherbereich auf der Festplatte zusammen. Bei Linux wird der Arbeitsspeicher auf der Festplatte durch die Swap-Partition realisiert. Benötigt das Betriebssystem mehr Speicher als vorhanden ist, so lagert es Teile des Arbeitsspeichers auf die Swap-Partition aus.

Dieses Verfahren zur Vergrößerung des Arbeitsspeichers wird als *Paging* bezeichnet. Wenn der physikalische Speicher knapp wird, werden Memory-Pages (Speicherseiten) auf die Festplatte ausgelagert. Diese Speicherseiten haben meistens eine Größe von 4096 Bytes also 4 KByte. Bei Bedarf werden dann die ausgelagerten Seiten wieder in den physikalischen Arbeitsspeicher geladen.

Natürlich kann ein solcher Swap-Speicher keinen physikalischen Speicher ersetzen. Der Zugriff auf die Festplatte ist im Schnitt um den Faktor 10.000 langsamer als der Zugriff auf RAM-Speicher. Seine gute Funktion zeigt der Auslagerungsspeicher, wenn mehrere Programme gleichzeitig laufen. Im Zeitalter der GUI-Programme belegen laufende Programme oft nur Speicher, während Sie die wenig belasten. Dies liegt daran, daß sie die meiste Zeit nur darauf warten, daß der Benutzer endlich etwas tut. Beim Wechsel zwischen den Programmen merkt man deutlich die Verzögerung beim Schreiben- und Rückschreiben der Speicherseiten. Die eigentlich Programmausführung des aktiven Programms ist aber genau so schnell wie auch sonst, da sich seine Speicherseiten im physikalischen Arbeitsspeicher befinden.

Für die Größe der Swap-Partition gibt es eine Faustregel. Der Auslagerungsspeicher sollte immer doppelt so groß sein, wie der eingebaute Arbeitsspeicher. Dabei sollte eine Größe von 16 MB für die Swap-Partitione nicht unterschritten werden um eine gute Funktion des Betriebssystems zu gewährleisten.

Früher war die maximale Größe der Swap-Partition auf 128 MB beschränkt. Allerdings können bis zu 16 Swap-Partitionen eingerichtet werden, so daß eine Gesamtkapazität von 2 GB für die Auslagerungsdatei zur Verfügung stand. Seit Kernel 2.1.117 darf die Swap-Partition bis zu 2 GB groß werden. Die Anzahl der Swap-Partitionen kann bei der Kompilierung des Kernels festgelegt werden. Der entsprechende Eintrag beim Kernel 2.4.18<sup>3</sup> in der Kernel-Headerdatei `/usr/src/linux-2.4.18/include/linux/swap.h` lautet:

```
#define MAX_SWAPFILES 32
```

<sup>2</sup>Eine Erläuterung zu den Geräten finden Sie in den Kernelquellen in der Datei `/usr/src/linux/Documentation/devices.txt`.

<sup>3</sup>Nach den Kernelquellen des bei SuSE 8.0 mitgelieferten Originalkernels sowie des SuSE-Kernels.



Danach sollten bis zu 32 Swap Partitionen bei diesem Kernel möglich sein. Maximal unterstützt der Kernel 64 Swap-Partitionen.

Um die Zugriffsgeschwindigkeit zu erhöhen ist es sinnvoll die Swap-Partitionen auf mehrere Platten zu verteilen. Da die einzelnen Partitionen dann zu einem logischen Bereich zusammengezogen werden, bilden sie praktisch ein RAID Level 0. Bei IDE-Platten ist es dann noch von Vorteil, wenn beide Platten an einem separaten Controller hängen.

Wenn Sie meinen, sie haben genügend Speicher, dann können Sie auch mit der Formel "eins zu eins" arbeiten. Dies ist vor allem bei Serversystemen sinnvoll, die mit extra viel Speicher bestückt worden sind um Festplattenzugriffe zu verringern.

Wenn Sie allerdings meinen zu wenig Platz zu haben, dann können Sie die Swap-Partition auch auf die dreifache Arbeitsspeichergröße setzen.<sup>4</sup>

Sollten Sie auf der Festplatte nicht genügend Speicher oder Partitionsplatz besitzen, so können Sie auch eine Swap-Datei (*swap file*) einrichten. Dies ist allerdings nicht so effektiv wie eine separate Partition. Sollte ihr Rechner 4 MB oder weniger Arbeitsspeicher besitzen, so benötigen Sie auf jeden Fall eine Swap-Partition.

### Beispiele

Ihr System besitzt 8 MB Arbeitsspeicher. Sie legen daher eine Swap-Partition von 16 MB an. Der zur Verfügung stehende virtuelle Speicher liegt dann bei 24 MB.

Sie besitzen einen Rechner mit 128 MB RAM. Nach der Formel sollten Sie einen Swap-Bereich von 256 MB einrichten. Da eine Swap-Partition aber nur 128 MB groß sein darf, legen Sie zwei Partitionen mit jeweils 128 MB an.

Ein Datei-Server wurde für den schnellen Dateizugriff mit 512 MB RAM bestückt. Da Zugriffe auf die Festplatte möglichst vermieden werden sollen, reichen vier Swap-Partitionen á 128 MB.

Ihr alter 486er Rechner soll als Router eingesetzt werden. Da er nur 4 MB RAM besitzt, müssen Sie eine Swap-Partition einrichten um Linux installieren zu können. Dabei sollte die Swap-Partition mindestens eine Größe von 16 MB aufweisen.

## 10.5.1 Swap-Space einrichten

Der Auslagerungsspeicher kann als Swap-Partition oder als Swap-Datei angelegt werden.

### Anlegen der Swap-Partition

Um eine zusätzliche Swap-Partition einzurichten müssen Sie auf der Festplatte eine neue Partition anlegen. Dies können Sie mit dem Befehl `fdisk` (10.1.1) erledigen.

Nehmen wir einfach mal an, daß Sie eine zweite Festplatte (`/dev/hdb`) in Ihren Rechner eingebaut haben. Um die Swap-Performance zu verbessern, sollten Sie auf dieser zweiten Platte auch eine Swap-Partition einrichten. Starten Sie also das Programm `fdisk`.

```
enterprise:~ # fdisk /dev/hdb
```

```
The number of cylinders for this disk is set to 5005.
```

```
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
```

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
  - (e.g., DOS FDISK, OS/2 FDISK)

```
Befehl (m für Hilfe):
```

---

<sup>4</sup>Bekannt sind alle drei Regeln (einfacher, zweifacher oder dreifacher Arbeitsspeicher). Je nach Autor finden Sie unterschiedliche Angaben.

Mit dem Befehl `n` legen wir nun eine neue Partition an. Und zwar eine primäre Partition mit der Größe von 256 MB.

```
Befehl (m für Hilfe): n
Befehl  Aktion
      1      Logische Partition (5 oder größer)
      p      Primäre Partition (1-4)
P
Erster Zylinder (1-5005) [Standardwert: 1]:
Benutze den Standardwert 1
Letzter Zylinder oder +Größe, +GrößeK oder +GrößeM (1-5005) [Standardwert: 5005]: +256M
```

Mit dem Befehl `p` überprüfen wir den Eintrag in der Partitionstabelle und notieren uns die Werte.

```
Befehl (m für Hilfe): p

Festplatte /dev/hdb: 255 Köpfe, 63 Sektoren, 5005 Zylinder
Einheiten: Zylinder mit 16064 * 512 Bytes

      Gerät boot.  Anfang      Ende    Blöcke   Id  Dateisystemtyp
/dev/hdb1          1          34    273088   82   Linux Swap
```

Nun brauchen wir nur noch mit `w` die Partitionstabelle zu schreiben und das Programm zu verlassen. Danach muß der Rechner neu gestartet werden, damit die neue Partitionstabelle übernommen wird.

### Anlegen einer Swap-Datei

Um eine Swap-Datei zu erstellen, müssen Sie eine Datei öffnen und so viele Bytes hineinschreiben, wie die Swap-Datei groß sein soll. Hierfür können Sie gut den Befehl `dd` (4.5.4) verwenden.

```
enterprise:~ # dd if=/dev/zero of=/swap bs=1024 count=16384
16384+0 Records ein
16384+0 Records aus
enterprise:~ # ls -l /swap
-rw-r--r--  1 root    root      16777216 Nov 17 12:15 /swap
```

Das Gerät `/dev/zero` ist praktisch das Gegenteil von `/dev/null`. Während `/dev/null` praktisch ein WOM (Write Only Memory) ist, der alle Informationen schluckt und Sie nicht wieder hergibt, ist `/dev/null` ein unendlich großer Datenträger, auf dem aber nur Nullen stehen. Der obige Befehl liest praktisch 16384 Blöcke mit 1024 Bytes von diesem Gerät und schreibt Sie in die Datei `/swap`.

Nach dem Anlegen der Datei sollten Sie sicherheitshalber mit dem Befehl `sync` (10.3.4) die Dateisysteme wieder synchronisieren.

Auf keinen Fall darf die Swap-Datei eine Lücke besitzen. Die Blöcke der Datei müssen sich hintereinander auf der Platte befinden.

### Swap-Bereich formatieren

Nach dem Anlegen der Partition bzw. der Datei muß diese für den Einsatz als Auslagerungsspeicher vorbereitet werden, d. h. formatiert werden. Hier kommt der Befehl `mkswap` zum Einsatz.

```
mkswap -c GERÄTENAME
```

`GERÄTENAME` steht für den Namen der Partition bzw. der Datei. Der Schalter `-c` ist optional und bewirkt, daß der Swap-Bereich bei der Formatierung auf fehlerhafte Blöcke untersucht wird.

Nehmen wir als Beispiel mal die oben angelegte Swap-Partition. Mit dem folgenden Befehl wird sie als Swap-Bereich formatiert.

```
enterprise:~ # mkswap -c /dev/hdb1
Swapbereich Version 1 mit der Größe 279642112 Bytes wird angelegt
```

Die Swap-Datei wird analog angelegt. Nur daß anstatt des Gerätenamens, der Dateiname angegeben wird.

```
enterprise:~ # mkswap -c /swap
Swapbereich Version 1 mit der Größe 16773120 Bytes wird angelegt
enterprise:~ # sync
```

Nach dem Einrichten einer Swap-Datei sollten Sie immer das Dateisystem mit **sync** synchronisieren um sicherzugehen, daß die Informationen auch physikalisch auf die Platte übertragen worden sind. Dies ist bei der Swap-Partition natürlich nicht nötig.

### Swap-Bereich aktivieren und deaktivieren

Damit der neue Swap-Bereich auch vom System benutzt wird, muß er aktiviert werden. Dafür ist der Befehl **swapon** verantwortlich. Analog dazu können Sie natürlich mit dem Befehl **swapoff** einen Swap-Bereich auch deaktivieren. Anhand des Befehls **free** (12.8.5) kann die Veränderung des Swap-Bereichs beobachtet werden.

Als Beispiel schauen wir uns doch mal die Einbindung unserer Swap-Datei ins System an.

```
enterprise:~ # free
              total        used        free      shared    buffers     cached
Mem:          513448      466992      46456           0       63500      258672
-/+ buffers/cache:      144820      368628
Swap:         1028120           0      1028120

enterprise:~ # swapon /swap
enterprise:~ # free
              total        used        free      shared    buffers     cached
Mem:          513448      467032      46416           0       63528      258672
-/+ buffers/cache:      144832      368616
Swap:         1044496           0      1044496

enterprise:~ # swapoff /swap
enterprise:~ # free
              total        used        free      shared    buffers     cached
Mem:          513448      467036      46412           0       63536      258672
-/+ buffers/cache:      144828      368620
Swap:         1028120           0      1028120
```

Auf jeden Fall sollten Sie eine Swap-Datei nur dann löschen, wenn Sie sie vorher mit **swapoff** deaktiviert haben.

Wie üblich ist dieser Befehl nicht von Dauer. Nach einem Reboot ist der Swap-Bereich wieder inaktiv. Die Swap-Bereich müssen also bei jedem Start initialisiert werden. Dies geschieht mit dem Befehl **swapon -a** in einer der Startdateien.

**swapon -a** liest die Informationen aus der Datei **/etc/fstab** aus. Wenn dort Informationen des Typs

```
/dev/hda6      swap    swap    sw 0 0
```

bzw.

```
/dev/hdb1      swap    swap    pr=42 0 0
```

dann werden die jeweiligen Swap-Bereiche beim Booten initialisiert. Sie sollten für jeden neu angelegten Swap-Bereich einen Eintrag zur **/etc/fstab** hinzufügen.

### 10.5.2 mkswap

Das Kommando `mkswap` bereitet eine Partition oder eine Datei für den Einsatz als Swap-Bereich vor.

```
mkswap [OPTIONEN] GERÄT [GRÖSSE]
```

Der Parameter `GERÄT` kann die Bezeichnung für eine Partition (z. B. `/dev/hdb3`) oder der Name einer Datei sein.

Die Angabe der Größe des Swap-Bereichs ist heute nicht mehr nötig und existiert nur noch aus Kompatibilitätsgründen. Der Befehl nutzt den zur Verfügung stehenden Platz in der Partition bzw. in der angelegten Datei voll. Eine falsche Angabe der Größe kann zu Schäden auf der Festplatte führen. Also lieber Finger weg von diesem Parameter.

#### Optionen

<code>-c</code>	Testen auf fehlerhafte Blocks vorm Formatieren ( <i>check</i> )
<code>-f</code>	Erlaubt auch die Ausführung mit unsinnigen Parametern ( <i>force</i> )
<code>-p GRÖSSE</code>	Größe der verwendeten Speicherseiten ( <i>page</i> )
<code>-v0</code>	Erstellt einen Swap-Bereich im alten Format
<code>-v1</code>	Erstellt einen Swap-Bereich im neuen Format (Standard)

### 10.5.3 swapon

Das Kommando `swapon` initialisiert Geräte und Dateien für den Einsatz als Swap-Bereich. Normalerweise wird der Befehl in einer der Startdateien ausgeführt um den Swap-Bereich immer zur Verfügung zu stellen.

```
swapon [OPTIONEN]
swapon [OPTIONEN] SWAPBEREICH
```

#### Optionen

<code>-h</code>	Hilfe ( <i>help</i> )
<code>-V</code>	Versionsnummer ( <i>version</i> )
<code>-s</code>	Übersicht über die Swap-Bereiche ( <i>summary</i> )
<code>-a</code>	Alle Swap-Bereiche aus <code>/etc/fstab</code> mit der Option <code>sw</code> bzw. <code>pri</code> werden initialisiert. ( <i>all</i> )
<code>-p PRIORITÄT</code>	Gibt die Priorität des Swap-Bereichs an (Wert zwischen 0 und 32767), steht seit Kernel 1.3.2 zur Verfügung ( <i>priority</i> )

Die Option sich eine Übersicht über die Swap-Bereiche anzeigen zu lassen steht nur dann zur Verfügung, wenn `/proc/swaps` existiert. Dies ist erst seit Kernel 2.1.25 der Fall.

```
enterprise:/home/ole # ls -l /proc/swaps
-r--r--r-- 1 root root 0 Nov 17 13:38 /proc/swaps
enterprise:/home/ole # cat /proc/swaps
Filename      Type      Size    Used    Priority
/dev/hdb6     partition 1028120 0        42
/swap         file      16376   0        -2
enterprise:/home/ole # swapon -s
Filename      Type      Size    Used    Priority
/dev/hdb6     partition 1028120 0        42
/swap         file      16376   0        -2
```

Um den Befehl jedesmal beim Starten ausführen zu können, wird er in einer der Startdateien des Systems eingetragen. Diese befinden sich meistens im Verzeichnis `/etc/rc.d`. Unter SuSE (8.0) ist `/etc/rc.d` ein Link auf `/etc/init.d`. Dort findet sich der Eintrag in den Dateien `boot.localfs` und `boot.swap`.

```
enterprise:/etc # ls -ld init.d rc.d
drwxr-xr-x 11 root root 4096 Jul 15 21:54 init.d
lrwxrwxrwx 1 root root 6 Mai 2 2002 rc.d -> init.d
```

```

enterprise:/etc # cd init.d/
enterprise:/etc/init.d # grep -l swapon * 2> /dev/null
boot.localfs
boot.swap
enterprise:/etc/init.d # grep -n swapon * 2> /dev/null
boot.localfs:29:      swapon -a &> /dev/null
boot.swap:26:  # .. this should work know with the new swapon behavior(u)r
boot.swap:29:  swapon -a &> /dev/null

```

### 10.5.4 swapoff

Wenn ein Swap-Bereich eingebunden werden kann, dann muß er auch aus dem System entfernt werden können. Dafür ist der Befehl `swapoff` verantwortlich.

```

swapoff [OPTIONEN]
swapoff SWAPBEREICH

```

#### Optionen

-h	Hilfe ( <i>help</i> )
-V	Versionsnummer ( <i>version</i> )
-s	Übersicht über die Swap-Bereiche ( <i>summary</i> )
-a	Alle Swap-Bereiche aus <code>/etc/fstab</code> mit der Option <code>sw</code> bzw. <code>pri</code> werden aus dem System entfernt. ( <i>all</i> )

## 10.6 Verwaltung

Linux unterscheidet folgende Dateiararten

- gewöhnliche Dateien (ordinary files)
- Verzeichnisse (directories)
- Gerätedateien (devices)
- Named Pipes (FIFOs)
- Sockets
- Verweise (Links)

**Gewöhnliche Dateien** sind die Dateien, die mit einer mehr oder weniger gewöhnlichen Anwendung erzeugt werden - der Text, der mit einer Textverarbeitung erstellt wird, die Daten einer Datenbank-, die Tabelle eines Kalkulationsprogramms oder auch die Zeilen eines Shell-Skripts, Tabellen, die von LINUX zu diversen Verwaltungszwecken benutzt werden, wie etwa die Paßwortdatei `/etc/passwd` und nicht zuletzt Programmdateien. Gewöhnliche Dateien sind solche Dateien, deren Inhalt für den einzelnen Anwender direkt oder indirekt von Interesse ist (am Text den man geschrieben hat ist man direkt interessiert, der Inhalt der Datei `/etc/passwd` ist für jeden Anwender deshalb von Belang, weil über ihn der Zugang zum System ermöglicht sowie ein Teil der persönlichen Arbeitsumgebung bereitgestellt wird).

**Verzeichnisse** ermöglichen eine strukturierte Dateiverwaltung. Die Organisation der Daten wird durch sie wesentlich vereinfacht. Verzeichnisse sind Tabellen mit den Namen von Dateien sowie einem Vermerk, über den das System den Zugriff auf die Daten in den Dateien steuert. Die Anzahl der gewöhnlichen Dateien ist schon in einem Einplatzsystem wie DOS oder Windows 95 so groß, daß ein Ordnungsprinzip erforderlich wird. So war ein wesentlicher Grund für die erste Überarbeitung des Betriebssystems DOS (Einführung der Version 2.0) die Verwendung von Festplatten für den IBM PC-XT. Diese ermöglichten die Speicherung großer Datenmengen. Die Hierarchie der DOS-Dateiverwaltung kannte bis dahin in Anlehnung an das System CP/M Laufwerke (Disketten!) und Dateien. Da nun die Anzahl der Dateien sehr hoch sein konnte, führten die Microsoft-Programmierer in DOS die Verzeichnisse als Verwaltungsinstrument hinzu. Somit kennt DOS

drei Hierarchieebenen in der Dateiverwaltung: Laufwerk (aus CP/M geerbt), Verzeichnisse und - natürlich - gewöhnliche Dateien. Daß dabei die Verzeichnisse und ihr Handling im wesentlichen aus Unix übernommen wurden, erleichtert es den heutigen LINUX-Programmierern, den Zugriff auf DOS-Daten zu realisieren; denn Dateien und Verzeichnisse sind in DOS und LINUX prinzipiell sehr ähnlich.

**Geräte-dateien** vereinfachen den Zugriff des Anwenders auf die Hardware. Sie sind im Prinzip Verweise auf die entsprechenden Treiber im LINUX-Kernel. Sollen etwa Daten auf den Bildschirm ausgegeben werden, so kann der Prozeß, der diese Aufgabe zu erledigen hat, die Daten einfach in die zugehörige Geräte-datei, dies könnte z. B. `tty01` sein, kopieren. Dadurch wird der zuständige Gerätetreiber im Kernel zur Ausgabe auf dem Bildschirm veranlaßt.

**Named Pipes** oder **FIFOs** sind eines der vielen Mittel, die LINUX für die Kommunikation zwischen Prozessen (kurz **IPC** für *Inter Process Communication*) bereitstellt. Vereinfacht gesagt ist ein Prozeß ein laufendes Programm. Solch ein Prozeß kann Daten in eine FIFO schreiben, die dann von einem anderen Prozeß gelesen werden können. Eine Variante dieser Möglichkeit des Datenaustausches zwischen Programmen kennen wir schon. Eine Kommandozeile wie

```
ls -l | more
```

dürfte Ihnen in ihrer Bedeutung klar sein.<sup>5</sup> Der Befehl `ls` leitet seine Daten statt auf den Bildschirm an den Pager `more` weiter, der sie so bearbeitet, daß er sie bildschirmseitenweise anzeigt. Bei Linux handelt es sich um temporäre Objekte, die von Prozessen benutzt werden können, die durch ein und dieselbe Befehlszeile gestartet wurden. Named Pipes dagegen werden als besondere Dateien angelegt und können von Prozessen benutzt werden, die unabhängig voneinander sind.

**Sockets** dienen, wenn man so will, ebenfalls der IPC. Dabei geht es aber nicht um Kommunikation zwischen Prozessen in einem LINUX/Unix-Rechner. Sockets ermöglichen Kommunikation über ein Netz hinweg. Dabei kann durchaus einer der beteiligten Kommunikationsteilnehmer ein Nicht-Unix -Programm sein, d.h. Sockets ermöglichen durchaus Kommunikation zwischen einem Prozeß, den ein Benutzer auf einem LINUX-Rechner startet einerseits, und einer DOS-Anwendung auf einem weit entfernten PC andererseits. Da nun DOS als Einplatzsystem keine Prozesse kennt, lassen sich Sockets nicht eindeutig der IPC zuordnen. Im wesentlichen steckt hinter den Sockets jedoch ein dem Pipelining verwandtes Prinzip des verbindungsorientierten Datenaustausches. Sie ermöglichen also die Weitergabe von Daten, in diesem Falle sogar via Netzwerk.

**Links** sind keine eigentlichen Dateien, sondern lediglich Verweise auf solche. Im Zusammenhang mit den Inodes werden wir Links genauer untersuchen können.

### 10.6.1 mkfifo

Erzeugen kann man FIFOs mit dem Befehl `mkfifo` (*make fifo*).

```
mkfifo [OPTIONEN] NAME
```

Die Befehlszeile

```
mkfifo testpipe
```

würde z. B. eine Named Pipe mit Namen `testpipe` erzeugen. Genutzt werden könnte sie dann wie folgt:

```
ls /dev > testpipe & less < testpipe
```

`ls /dev` zeigt die Namen aller Geräte-dateien (genauer: aller Dateien des Verzeichnisses `/dev`). In diesem Fall werden die Daten in die Named Pipe umgeleitet statt auf dem Bildschirm ausgegeben. Der Befehl `less` liest die Daten aus `testpipe` und gibt sie bildschirmseitenweise aus. Das `&` zwischen den Befehlen sorgt dafür, daß beide Teilbefehle nacheinander als Prozesse gestartet werden, allerdings ohne daß auf ihr Ende gewartet werden müßte. Insgesamt werden in diesem Beispiel also die Namen aller Geräte-dateien bildschirmseitenweise angezeigt. Dieses Beispiel stellte die Named Pipes nur prinzipiell vor. Als Benutzer wird man in der gegebenen Situation eine gewöhnliche Pipe vorziehen:

```
ls /dev | less
```

### 10.6.2 Dateinamen

Das Dateisystem von MS-DOS beschränkt die Größe des Namens einer Datei auf acht Zeichen plus einem mit einem Punkt abgetrennten Dateikürzel mit drei Zeichen. Im Gegensatz dazu erlaubte schon das erste

<sup>5</sup>Falls nicht, arbeiten Sie Kapitel 5 noch einmal durch.

UNIX-Dateisystem 14 Zeichen. Das LINUX-Dateisystem ext2 (*extended filesystem, version 2*) erlaubt bis zu 255 Zeichen. Dabei ist der Punkt nur eines von vielen Zeichen und kann beliebig oft angewendet werden.

Im Gegensatz zu MS-DOS unterscheidet LINUX bei Dateinamen zwischen Groß- und Kleinschreibung. TAPIC0, Tapico und tapico sind drei verschiedene Dateinamen. Aus praktischen Gründen werden die Dateinamen meistens klein geschrieben. Bei Pfadangaben werden die einzelnen Verzeichnisse durch einen Schrägstrich / (*slash*) getrennt. Daher ist dieses Zeichen für Dateinamen verboten. Auch das Minuszeichen kann am Anfang einer Datei Probleme bereiten. So stellt sich bei

```
ls -fault
```

die Frage, ob nun die Datei `-fault` oder die Schalter `f`, `a`, `u`, `l` und `t` gemeint sind.

Probleme bereiten auch die nichtdruckbaren (ASCII-)Zeichen 0 bis 32 sowie

```
< > | * ? [ ] { } ( ) $ ' ` " ! ; \ # ^
```

Diese Zeichen sind nicht verboten, können aber große Probleme bereiten und sollten deshalb auch vermieden werden.

### 10.6.3 Die Datei

Was gehört eigentlich zu einer Datei?

1. Der Inhalt der Datei
2. Ein Link
  - (a) Der Name der Datei
  - (b) Die eindeutige Inode-Nummer
3. Der Inhalt der Inode (die eigentliche Verwaltungsinformation)
  - (a) Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei, ...)
  - (b) Rechtetabelle
  - (c) Anzahl der Links (bzw. Dateinamen)
  - (d) Dateibesitzer (uid)
  - (e) Besitzergruppe (gid)
  - (f) Größe in Byte (bei Geräteadressen statt dessen Major und Minor Device number)
  - (g) Datum der letzten Änderung
  - (h) Datum des letzten Zugriffs
  - (i) Datum der letzten Änderung dieser Verwaltungsinformationen
  - (j) Attribute (nur für das Dateisystem ext2)
  - (k) physikalischer Ort der Speicherung (Blöcke in denen die Datei abgelegt ist)

Die Inode speichert alle Verwaltungsinformationen über die Datei. Da die Benutzer in aller Regel besser mit Namen als mit Nummern arbeiten können, bietet Linux folgende Schnittstelle zwischen Benutzer und Dateisystem an. Der Benutzer verwendet in den Befehlen einen Dateinamen. Die benutzte Shell, z. B. die `bash`, ermittelt über den Verzeichniseintrag, den Link, die zugehörige Inode. Jeder Dateiname ist mit genau einer Inode verbunden. Dagegen kann eine Inode durch mehrere Dateiname referenziert werden. Deshalb kann eine Datei nicht nur einen Dateinamen besitzen. Wird die Inode beschädigt, dann kann auf die Datei nicht mehr zugegriffen werden. Um die Inodes der Dateien zu bestimmen, wird der Schalter `-i` des Befehls `ls` eingesetzt.

#### Beispiel

```
tapico@defiant:/ > ls -aFi
 2 ./              1 d/              11 lost+found/    10241 tmp/
 2 ../            38914 dev/         124929 mnt/        165 trellwan
```

305412 a/	2049 etc/	126977 opt/	83969 usr/
59393 bin/	118785 floppy/	1 proc/	12289 var/
51201 boot/	120833 home/	69633 root/	
1 c/	117000 kurs/	55297 sbin/	
112641 cdrom/	65537 lib/	211184 seminare/	

Besitzen zwei Dateinamen die gleiche Inode, so handelt es sich hierbei um ein und dieselbe Datei. Auffällig ist die Inode 1 bei `/proc`, `/c` und `/d`. Bei `proc` handelt es sich um ein virtuelles Dateisystem zur Prozessüberwachung. `/c` und `/d` sind eingebundene FAT16-Dateisysteme von Windows. Ihnen wird daher auch keine "richtige" Inode zugewiesen.

#### 10.6.4 stat

Der Befehl `stat` zeigt die Informationen der Inode an.

```
stat DATEINAME
```

Allerdings ist er nicht auf jedem System zu finden. Bei SuSE muß das Paket explizit installiert werden.

### 10.7 Links

Im Prinzip ist ein Link nur ein weiterer Name für die Datei. Bei den Links wird zwischen dem harten Link und dem symbolischen (weichen) Link unterschieden.

Ein Link kann dazu benutzt werden einem anderen Benutzer Zugriff auf eine Datei zu gewähren. Als Erstes bekommt der andere Benutzer Rechte auf die Datei und evt. auf das Verzeichnis. Danach kann er einen Link auf die Datei in seinem Heimatverzeichnis anlegen. Dadurch erhält er einen vereinfachten Zugriff auf den Inhalt der Datei.

Durch die Benutzung von Links brauchen oft keine Kopien von Dateien erstellt werden. Dies spart Speicherplatz und erleichtert die Aufgabe des Systemadministrators alle Dateien auf dem neuesten Stand zu halten.

Daneben können große Verzeichnisbäume übersichtlicher gestaltet werden. Z. B. landen alle Auswertungsergebnisse in einem Verzeichnis `data`. In den Verzeichnissen `helios` und `ulysses` können dann Links auf die entsprechenden Datendateien im Verzeichnis `data` gemacht werden.

#### 10.7.1 Harte Links

Immer wenn Sie eine Datei erzeugen wird auch ein Link zu dieser Datei erzeugt. Er ermöglicht den Zugriff auf die Datei, weil er direkt auf die Inode der Datei zeigt. Der Befehl `rm` (4.5.7) löscht übrigens nicht direkt die Datei, sondern den Link, der auf die Inode der Datei zeigt.

Einen Link können Sie nur auf eine existierende Datei anlegen. Er ist ein weiterer Name für die gleiche Datei. Dies wird deutlich wenn Sie sich die Links mit `ls -li` anzeigen lassen. Beide Links zeigen auf die gleiche Inode. Sie unterscheiden sich nur im Namen und/oder ihrem Ort im Verzeichnisbaum.

#### Beispiel

Auf die existierende Datei `clarkkent` wird ein harter Link namens `superman` angelegt. Wird nun `superman` durch z. B. den Editor `joe` editiert, so finden sich die Änderungen auch in `clarkkent` wieder.

```
tapico@defiant:~/links > ls -li
insgesamt 2
111099 -rw-r--r--  2 tapico  users      46 Jan 14 13:41 clarkkent
111099 -rw-r--r--  2 tapico  users      46 Jan 14 13:41 superman
```



In der dritten Spalte sehen Sie die Anzahl der harten Links, die auf die Datei zeigen. Lösche Sie nun einen der beiden Links, so bleibt die Datei trotzdem bestehen. Erst wenn alle Links auf die Datei gelöscht sind, die Zahl also auf Null steht, wird die Datei auf wirklich gelöscht.

Es gibt aber auch Einschränkungen für harte Links. So müssen sich alle Links auf dem gleichen Dateisystem befinden. Außerdem ist es einem normalen Benutzer nicht möglich einen Hardlink auf ein Verzeichnis einzurichten. Dies ist nur **root** vorbehalten.

### 10.7.2 Symbolische Links

Im Gegensatz zu dem harten Link zeigt der symbolische Link nur indirekt auf eine Datei, da er auf einen Namen (harten Link) der Datei zeigt. Deshalb ist er auch ein neuer Eintrag mit eigener Inode im Verzeichnisbaum.

Symbolische Links werden dann eingerichtet, wenn die Grenzen der harten Links umgangen werden sollen. Symbolische Links können eingerichtet werden für

- Verzeichnisse durch einen normalen Benutzer.
- Dateien, die nicht existieren.
- Dateien, die sich auf einem anderen Dateisystem befinden.

Wie im Beispiel unten zu sehen, ändert sich das Aussehen des symbolischen Links bei Verwendung des Befehls `ls -l`. Der Linkname wird durch einen Pfeil und den Namen der Zieldatei ergänzt.

```
tapico@defiant:~/links > ls -l
insgesamt 2
111099 -rw-r--r--  2 tapico  users          46 Jan 14 13:41 clarkkent
111101 lrwxrwxrwx  1 tapico  users           5 Jan 14 13:51 kurs -> /kurs
111100 lrwxrwxrwx  1 tapico  users           9 Jan 14 13:50 superheld -> clarkkent
111099 -rw-r--r--  2 tapico  users          46 Jan 14 13:41 superman
```

Die Funktion der Befehle `cd` und `pwd` kann sich bei gelinkten Verzeichnissen je nach Shell unterscheiden. In der `bash` zeigt, wenn wir dem Link `kurs` gefolgt sind, z. B. der Befehl `pwd` das Arbeitsverzeichnis über die verlinkte Struktur an. Der Befehl `cd ..` führt auch in das Verzeichnis mit dem Link zurück.

```
tapico@defiant:~/links > cd kurs
tapico@defiant:~/links/kurs > pwd
/home/tapico/links/kurs
tapico@defiant:~/links/kurs > cd ..
tapico@defiant:~/links > pwd
/home/tapico/links
```

In anderen Shells führt der Link an den Ort der normalen Struktur. Das aktuelle Verzeichnis ist dann nicht mehr durch den symbolischen Link gegeben, sondern durch den harten Link des Verzeichnisses.

### 10.7.3 ln

Das Kommando `ln` legt einen harten oder einen symbolischen Link an.

```
ln [OPTIONEN] DATEINAME NEUERLINK
```

Ist `NEUERLINK` ein Verzeichnis und die Optionen `-d` und `-F` sind nicht gegeben, so wird in dem Verzeichnis ein Link mit dem Namen `DATEINAME` angelegt.

#### Optionen

- `-d` Legt einen Link auf ein Verzeichnis an (Nur Superuser)
- `-F` Legt einen Link auf ein Verzeichnis an (Nur Superuser)
- `-s` Legt einen symbolischen Link an

Einen harten oder symbolischen Link können Sie durch den Befehl `rm` entfernen.

### 10.7.4 Der Befehl cp und die Links

Die Links, egal ob Hard- oder Softlink, sind Wegweiser zu den Inodes. Was passiert, wenn diese kopiert werden sollen? Werfen wir doch mal einen Blick auf den Befehl `cp` und seine Arbeit mit den Links.

Anstatt Kopien von Dateien zu erstellen, kann der Befehl `cp` auch Hard- und Softlinks anlegen. Für eine Hardlink geben Sie die Option `-l` an und für einen Softlink die Option `-s`.

```
ole@enterprise:~/test> ls -ilG
insgesamt 8
147475 -rw-r--r-- 1 ole 78 2004-06-28 21:46 blubb
147479 -rw-r--r-- 1 ole 144 2004-06-28 21:46 fasel
ole@enterprise:~/test> cp -l blubb blubber
ole@enterprise:~/test> cp -s fasel fusel
ole@enterprise:~/test> ls -ilG
insgesamt 12
147475 -rw-r--r-- 2 ole 78 2004-06-28 21:46 blubb
147475 -rw-r--r-- 2 ole 78 2004-06-28 21:46 blubber
147479 -rw-r--r-- 1 ole 144 2004-06-28 21:46 fasel
147481 lrwxrwxrwx 1 ole 5 2004-06-28 21:47 fusel -> fasel
```

Eine besondere Situation tritt auf, wenn Softlinks kopiert werden sollen. Denn hier stellt sich die Frage was denn nun eigentlich kopiert werden soll: Der Link oder die Datei, auf die der Link zeigt.

Wenn der Befehl `cp` ohne Option verwendet wird, wird die verlinkte Datei kopiert.

```
ole@enterprise:~/test> cp fusel fiesel
ole@enterprise:~/test> ls -ilG
147479 -rw-r--r-- 1 ole 144 2004-06-28 21:46 fasel
147483 -rw-r--r-- 1 ole 144 2004-06-28 21:53 fiesel
147481 lrwxrwxrwx 1 ole 5 2004-06-28 21:47 fusel -> fasel
```

Bei der Verwendung des Schalters `-d` werden die Softlinks auch als Links kopiert.

```
ole@enterprise:~/test> ls -ilG
147485 lrwxrwxrwx 1 ole 5 2004-06-28 21:58 dusel -> fasel
147479 -rw-r--r-- 1 ole 144 2004-06-28 21:46 fasel
147483 -rw-r--r-- 1 ole 144 2004-06-28 21:53 fiesel
147481 lrwxrwxrwx 1 ole 5 2004-06-28 21:47 fusel -> fasel
```

Und wie sieht es bei Kopien über ganze Verzeichnisebenen hinweg aus? Der Link wird genau so kopiert, wie er auch dasteht. Relative Links bleiben relative Links und die Pfade werden nicht angepaßt. Sein Ziel muß also entsprechend mitkopiert werden. Bei absoluten Links, wo der Pfad an der Wurzel beginnt, gibt es das Problem nicht. Sie zeigen weiterhin auf die ursprüngliche Datei. Beide Verfahren haben Ihre Vor- und Nachteile.

```
ole@enterprise:~/test> cp -d /usr/bin/bunzip2 entpack
ole@enterprise:~/test> cp -d /usr/X11/bin/X klickibunti
ole@enterprise:~/test> ls -ilG
147475 lrwxrwxrwx 1 ole 5 2004-06-28 22:01 entpack -> bzip2
147479 lrwxrwxrwx 1 ole 16 2004-06-28 22:03 klickibunti -> /var/X11R6/bin/X
```

## 10.8 Festplattennutzung

Die Einrichtung, Formatierung und Einbindung einer Partition in das System hat einen gravierenden Nachteil. Die Benutzer können nun dort Dateien speichern und füllen dort in kürzester Zeit den vorhandenen

Speicherplatz<sup>6</sup>. Damit Linux vernünftig arbeiten kann, sollte zwischen 5 und 30 Prozent der Plattenkapazität frei sein. Es ist also zwingend notwendig den zur Verfügung stehenden Plattenspeicher zu kontrollieren.

### 10.8.1 du

Das Kommando `du` (*Disk Usage*) zeigt den Platz an, den eine Datei oder ein Verzeichnis mit seinen Dateien und Unterverzeichnissen einnimmt.

`du [OPTIONEN] [DATEINAME]`

Wird `du` ohne Dateinamen aufgerufen, dann zeigt es die Daten des aktuellen Arbeitsverzeichnisses an.

#### Optionen

<code>-a</code>	Zeige die Nutzung für einzelne Dateien
<code>-b</code>	Anzeige des Speicherplatzes in Bytes
<code>-c</code>	Zeigt die Summe an
<code>-h</code>	Zeigt durch einen Buchstaben die verwendete Einheit an
<code>-k</code>	Anzeige des Speicherplatzes in Kilobytes (normal)
<code>-m</code>	Anzeige des Speicherplatzes in Megabytes
<code>-l</code>	Zählt auch die Links
<code>-s</code>	Gibt nur die totale Summe aus

Ohne den Schalter `-a` zeigt `du` nur die Verzeichnisse.

#### Beispiel

```
tapico@defiant:~ > du -c test
1259    test/mytest/textfilter
1263    test/mytest
14761   test/log
17453   test
17453   insgesamt
```

### 10.8.2 df

Der Befehl `df` zeigt die Nutzung der Partition an.

`df [OPTIONEN] [DATEINAME]`

Wird eine Datei oder ein Verzeichnis mit angegeben, so zeigt `df` die Daten der Partition an, auf der diese Datei bzw. dieses Verzeichnis liegt. Ansonsten zeigt das Kommando eine Übersicht aller Partitionen an.

#### Optionen

<code>-a</code>	Liefert Informationen über alle Dateisysteme (normal) ( <i>all</i> )
<code>-h</code>	Zeigt durch einen Buchstaben die verwendete Einheit an ( <i>human readable</i> )
<code>-i</code>	Zeigt die Nutzung der Inodes an ( <i>inode</i> )
<code>-k</code>	Anzeige des Speicherplatzes in Kilobytes (normal) ( <i>kilobyte</i> )
<code>-m</code>	Anzeige des Speicherplatzes in Megabytes ( <i>megabyte</i> )
<code>-t DSTYP</code>	Zeigt nur Dateisysteme vom Typ <code>DSTYP</code> ( <i>typ</i> )
<code>-T</code>	Zeigt den Dateisystemtyp für jeden Eintrag ( <i>typ</i> )
<code>-x DSTYP</code>	Zeigt keine Dateisysteme vom Typ <code>DSTYP</code> ( <i>exclude typ</i> )
<code>--sync</code>	Startet erst das <code>sync</code> -Kommando

Als Information liefert `df`:

- Die Größe des Geräts
- Die Anzahl der freien Blöcke

---

<sup>6</sup>Hier sei auf eins der ewigen EDV-Gesetze hingewiesen: Jede Festplatte jeder beliebigen Größe wird innerhalb kurzer Zeit zu 90% belegt sein.

- Die Anzahl der belegten Blöcke
- Den Anteil der freien Blöcke am Speicher in Prozent
- Den Namen des Geräts

## Beispiele

Eine Übersicht über die Plattennutzung aller gemounteten Datenträger liefert der Befehl `df`.

```
tapico@defiant:~ > df
Filesystem      1K-Blöcke  Benutzt Verfügbar Ben% Eingehängt auf
/dev/hda5        5162796   2893372   2007168   60% /
/dev/hda6        3099260    325012   2616816   12% /home
/dev/hda1        4096540   2525980   1570560   62% /windows/C
tmpfs            127828      0       127828    0% /dev/shm
/dev/sda1        127620     55962     71658   44% /media/sda1
```

Die Übersicht über die genutzten Inodes mit dem Befehl `df -i` funktioniert natürlich nur, wenn das Dateisystem auch auf Inodes-Basis arbeitet. Bei anderen Dateisystemen träumt sich Linux etwas im Kaffeesatz. (Hier `/dev/hda1` und `/dev/sda1`.)

```
tapico@defiant:~ > df -i
Filesystem      INodes   IBenut.  IFrei IBen% Eingehängt auf
/dev/hda5       656000   162745   493255   25% /
/dev/hda6       393600    3110   390490    1% /home
/dev/hda1       26619    26603     16   100% /windows/C
tmpfs           31957      1    31956    1% /dev/shm
/dev/sda1        0         0      0     - /media/sda1
```

## 10.8.3 Aufräumen des Systems

Was also tun, wenn das Dateisystem zu voll geworden ist?

- Finden Sie große Dateien, die nicht mehr benötigt werden. Log-, Mail- und News-Dateien können sehr groß werden. Auch die durch Abstürze erstellten Speicherabbildungen (*core dumps*) können sehr groß werden.
- Räumen Sie die temporären Verzeichnisse auf. Das sollte sowieso zu den normalen Wartungsaufgaben gehören.
- Selten genutzte Dateien sollten komprimiert werden und möglichst auf andere Geräte ausgelagert werden.
- Verzeichnisse sollten nicht so viele Dateien enthalten. Kleinere Verzeichnisse sind effektiver.
- Die Verzeichnisse an sich können nur größer werden. Je mehr Dateien vorhanden sind, desto größer wird der Verwaltungsplatz. Werden die Dateien gelöscht, bleibt die Verwaltungsdatei trotzdem so groß wie vorher. Das Verzeichnis muß gelöscht und anschließend wieder erstellt werden, um den Platz zu reduzieren.
- Ist nur eine Partition zu voll, können Dateien auf andere Partitionen ausgelagert werden.
- Eine zweite Möglichkeit ist es die Partition zu vergrößern. Dabei muß von den Daten ein Backup erstellt werden. Dann wird die Partition gelöscht und in dem neuen freien Platz (wahrscheinlich eine neue Platte) neu erstellt. Dann wird das Backup wieder zurückgespielt.

.....  
Notizen:  
.....

.....  
*Notizen:*

.....



## XII

Arbeitsblatt  
Dateisystem 1

## 12.1

**1** Welche Dateinamen sind richtig, problematisch oder falsch?

Dateiname	R	P	F	Dateiname	R	P	F
abc.txt				Dies ist Falsch			
Textdatei				ls -l			
-usr				*.*			
c:\dos				this is right			
pillepallename				\">\$<[!]			
..				...			
../..				calprobedatei1			
katalog_1.2.1				abc/def/ghi			
-123				123			
mc.man.gz				194.195.155.60			
..home				../home			

**2** Loggen Sie sich als *walter* auf der Konsole 1 ein.

**3** Kopieren Sie den Inhalt der Verzeichnisse *yast2*, *mc* und *cron* aus dem Verzeichnis */usr/share/doc/packages* in Ihr Heimatverzeichnis.

**4** Lassen Sie sich den Speicherbedarf der Dateien in Ihrem Heimatverzeichnis anzeigen.**5** Lassen Sie sich eine Übersicht über den freien Platz auf den Partitionen anzeigen.**6** Welchen Platz benötigen die Verzeichnisse */etc*, */home* und */usr/bin*?**7** Loggen Sie sich als *root* auf der Konsole 2 ein.**8** Wiederholen Sie Aufgabe 6 unter *root*. Gibt es Unterschiede im Ergebnis?

**9** Legen Sie im freien Speicherbereich bzw. im freien Bereich der erweiterten Partition eine Partition von 500 MB an und formatieren Sie sie mit dem Dateisystem *ext2*.

**10** Mounten Sie die Platte als *root* auf dem neuangelegten Mountpunkt */usr/extra*.**11** Wie groß ist die Platte laut *fdisk*?

**12** Lassen Sie sich eine Übersicht über den freien Platz auf den Partitionen anzeigen und vergleichen Sie ihr Ergebnis mit dem Ergebnis der vorherigen Aufgabe.

**13** Lassen Sie sich die Liste der Swap-Bereiche anzeigen.

**14** Legen Sie im freien Speicherbereich bzw. im freien Bereich der erweiterten Partition eine Swap-Partition von 256 MB an und aktivieren Sie sie.

**15** Lassen Sie sich die Liste der Swap-Bereiche erneut anzeigen.**16** Wieviel Swap-Speicher steht ihnen laut *free* zur Verfügung?**17** Starten Sie den Rechner neu.

**18** Führen Sie die Aufgaben 12 und 13 erneut durch und vergleichen Sie die jetzigen Ergebnisse mit den vorherigen.

XII

Arbeitsblatt  
Dateisystem 1

12.2

**19** Richten Sie das System so ein, daß die neue Partition und der neue Swap-Bereich permanent zugänglich sind.

**20** Testen Sie die neuen Einstellungen, indem Sie den Rechner neu starten und die Aufgaben 12 und 13 erneut durchführen.

**21** Deaktivieren Sie wieder den neuen Swap-Bereich permanent.

**22** Ein Mitarbeiter steht vor dem Problem, daß er ein-, zweimal die Woche mehr Auslagerungsspeicher für komplizierte Rechenoperationen benötigt. Platz genug ist auf seiner Festplatte. Leider ist aber kein freier Bereich für eine Partition mehr vorhanden. Lösen Sie das Problem so, daß der User eigenständig den neuen Swap-Bereich aktivieren und deaktivieren kann.

.....  
*Notizen:*

.....



## XIII

Arbeitsblatt  
Arbeiten mit Disketten

## 13.1

Sie benötigen für die Aufgaben zwei 3,5"-Disketten.

- 1 Loggen Sie sich als root an Terminal 1 ein.
- 2 Legen Sie den User *ardan* mit Home-Verzeichnis an. Geben Sie ihm das Paßwort *victor*.
- 3 Lassen Sie sich alle möglichen Gerätetypen für das Diskettenlaufwerk A: anzeigen.
- 4 Legen Sie die erste Diskette ins Diskettenlaufwerk ein. Führen Sie eine Low-Level-Formatierung (1440 KB) durch.
- 5 Erstellen Sie ein ext2-Dateisystem auf der Diskette.
- 6 Lassen Sie sich den Inhalt der Datei `/etc/mtab` anzeigen. Welche Laufwerke sind gemountet?
- 7 Mounten Sie die Diskette auf das Verzeichnis `/floppy`.
- 8 Lassen Sie sich erneut den Inhalt der Datei `/etc/mtab` anzeigen. Welche Laufwerke sind gemountet?
- 9 Loggen Sie sich an Terminal 2 als *ardan* ein. Versuchen Sie das Diskettenlaufwerk auf `/floppy` zu mounten.
- 10 Wechseln Sie zu Terminal 1. Unmounten Sie das Diskettenlaufwerk. Wechseln Sie zu Terminal 2 und versuchen Sie erneut das Diskettenlaufwerk zu mounten.
- 11 Lassen Sie sich die Auslastung aller Laufwerke anzeigen.
- 12 Unmounten Sie das Diskettenlaufwerk.
- 13 Wechseln Sie zu Terminal 1 und legen Sie das Verzeichnis `/a` an. Ändern Sie die Rechte des Verzeichnisses so, daß alle volle Rechte besitzen.
- 14 Mounten Sie die Diskette auf das Verzeichnis `/a`.
- 15 Unmounten Sie die Diskette wieder. Wechseln Sie zu Terminal 2. Versuchen Sie nun als *ardan* das Diskettenlaufwerk zu mounten.
- 16 Wechseln Sie wieder zu Terminal 1 und fügen Sie in die Datei `/etc/fstab` einen Eintrag hinzu, so daß jeder das Diskettenlaufwerk auf `/a` mounten kann. Testen Sie dies.
- 17 Formatieren Sie nun die Diskette 2 mit 1440 KB und erstellen ein DOS-Dateisystem darauf.
- 18 Ändern Sie die Datei `/etc/fstab` so, daß auf das Verzeichnis `/a` nur vfat-Dateisysteme gemountet werden können.
- 19 Mounten Sie Diskette 2 auf `/a`.
- 20 Versuchen Sie nun Diskette 1 auf `/a` zu mounten. Was passiert?
- 21 Führen Sie erneut eine Low-Level-Formatierung für Diskette 1 ein durch. Wählen Sie dabei das größtmögliche Format. Erstellen Sie ein ext2-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun auf der Diskette zur Verfügung.
- 22 Erstellen Sie ein minix-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun zur Verfügung?
- 23 Erstellen Sie ein DOS-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun auf der Diskette zur Verfügung.
- 24 Kann ein Windows-Rechner dieses Diskette lesen? Welche Kapazität kann ein Windows-Rechner maximal verkraften?

.....  
*Notizen:*

.....

# Kapitel 11

## Dateisysteme und Disk Quotas

### 11.1 Aufbau des ext2-Dateisystems

Das ext2-Dateisystem besteht aus dem Bootblock und den Blockgruppen. Die Blockgruppen sind wiederum aufgeteilt in einen Superblock, einer Liste der Blockgruppenbeschreibungen, Bitmaps für Blöcke und Inodes, Inode-Tabelle und den Datenblöcken.

#### 11.1.1 Bootblock

Der Bootblock liegt im ersten Block (Block 0) des Dateisystems. Er enthält ein Programm zum Starten und Initialisieren des gesamten Systems. Meistens enthält nur das erste Dateisystem einen Bootblock. Beim ext2-Dateisystem ist der Bootblock 1024 Zeichen groß.

#### 11.1.2 Blockgruppe

Jede Blockgruppe, wie in der Abbildung 11.1 zu sehen, besteht aus sechs Komponenten:

- Superblock
- Liste der Blockgruppenbeschreibungen
- Bitmap für Blöcke
- Bitmap für Inodes
- Inode-Tabelle
- Datenblöcke

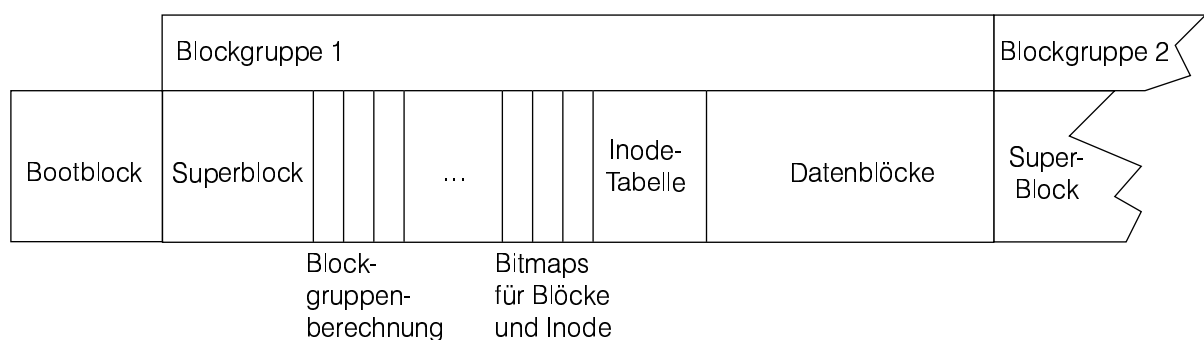


Abbildung 11.1: Aufbau des ext2-Dateisystems

Die Standardgröße für eine Blockgruppe liegt bei 8192 Blöcken und 2048 Inodes, von denen acht reserviert sind.

### 11.1.3 Superblock

Die Informationen über das gesamte Dateisystem liegen im Superblock. Wenn dieser Superblock beschädigt wird, kann das Dateisystem nicht mehr ins Dateisystem eingebunden (gemountet) werden. Deswegen werden in regelmäßigen Intervallen Backup-Kopien dieses Blocks im Dateisystem angelegt. Im Normalfall erfolgt das alle 8192 Blöcke. Das heißt die Daten befinden sich in Block 8193, 16385, 24577 etc. Die Angabe

`Blocks per group:` 8192

des Befehls `dumpe2fs` gibt den für das Gerät aktuellen Wert an.

Die Strukturdefinition des Superblocks finden Sie in der Datei `/usr/include/linux/ext2_fs.h` im Abschnitt `Structure of the super block`. Die wichtigsten Informationen sind:

- Gesamtgröße des Dateisystems in Blöcken bzw. Inodes
- Anzahl der freien Blöcke
- Anzahl der Blöcke für die Inodes
- Anzahl der freien Inodes
- Adresse des ersten Datenblocks
- Größe eines Datenblocks
- Größe eines Teildatenblocks
- Größenbestimmung für eine Blockgruppe
- Zeitpunkt des Einbinden des Dateisystems
- Zeit der letzten Änderung
- Anzahl der Einbindungen (*mount counts*)
- Maximale Anzahl der Einbindungen bevor ein Dateisystemcheck durchgeführt wird
- Versionsnummer des Dateisystems
- Magische Nummer: für ext2 lautet sie `0xEF53`.
- Nummer des einrichtenden Betriebssystems
- Status des Dateisystems: sauber oder mit Fehlern
- Verhalten bei Fehlererkennung
- Zeitpunkt der letzten Prüfung
- Maximaler Zeitraum zwischen zwei Prüfungen

### 11.1.4 dumpe2fs

Der Befehl `dumpe2fs` gibt den Superblock und die Blockgruppeninformationen eines Gerätes aus.

`dumpe2fs` [OPTIONEN] GERÄT

Der Befehl steht nur dem Superuser zur Verfügung. Als Beispiel hier die Ausgabe eines Befehls ohne die Übersicht über die Blockgruppen.

```

root@defiant:/ > dumpe2fs /dev/hda3
Filesystem volume name:   <none>
Last mounted on:         /mnt
Filesystem UUID:         0c7c1f82-510a-11d3-8c57-e447ecb45d66
Filesystem magic number:  0xEF53
Filesystem revision #:    0 (original)
Filesystem features:      (none)
Filesystem state:         not clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              370688
Block count:              1477980
Reserved block count:     73899
Free blocks:              203986
Free inodes:              269515
First block:              1
Block size:               1024
Fragment size:            1024
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         2048
Inode blocks per group:   256
Last mount time:          Mon Jan  8 20:45:40 2001
Last write time:          Mon Jan  8 21:19:10 2001
Mount count:              20
Maximum mount count:      20
Last checked:             Mon Dec 18 20:44:00 2000
Check interval:           15552000 (6 months)
Next check after:         Sat Jun 16 21:44:00 2001
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)

```

### 11.1.5 tune2fs

Die Einstellungen für das Dateisystem können über den Befehl `tune2fs` eingestellt werden.

`tune2fs [OPTIONEN] GERÄT`

#### Optionen

- c MOUNTCOUNTS      Stellt die Anzahl der maximalen Mountcounts zwischen zwei Dateisystemchecks ein
- C MOUNTCOUNTS      Trägt die Anzahl der erfolgten Mountcounts ein

Die weiteren Schalter entnehmen Sie bitte den Man-Pages.

#### Beispiele

```
root@defiant:/> tune2fs -c 200 /dev/hda5
```

stellt die Anzahl der maximalen Mountvorgänge zwischen zwei Dateisystemchecks für das Gerät `/dev/hda5` auf 200 ein.

### 11.1.6 Liste der Blockgruppenbeschreibungen

Hier werden die wichtigsten Kenndaten der Blockgruppe festgehalten. Dies sind

- Blocknummer für Block-Bitmap
- Blocknummer für Inode-Bitmap

- Blocknummer für Inode-Tabelle
- Anzahl der freien Blöcke
- Anzahl der freien Inodes
- Anzahl der Verzeichnisse

Damit die Verzeichnisse gleichmäßig über die Blockgruppen verteilt werden, werden neue Verzeichnisse in dem Block angelegt, der am wenigsten Verzeichnisse enthält.

### 11.1.7 Block- und Inode-Bitmap

Die Bitmaps für die Verwaltung der Blöcke und Inodes sind auf einen logischen Block beschränkt. Für jeden Block wird ein Bit benötigt. Daher ist eine Blockgruppe auf  $1024 \cdot 8 = 8192$  Blöcke beschränkt.

### 11.1.8 Inode-Liste

Die Inode-Liste besteht aus vielen aneinandergereihten Inodes, die alle exakt die gleiche Größe haben. Im Standardfall sind es 2048 Stück, die innerhalb einer Blockgruppe gespeichert werden. Die Inode ist der Dreh- und Angelpunkt des Dateizugriffs.

- Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei, ...)
- Rechtetabelle
- Anzahl der Links (bzw. Dateinamen)
- Dateibesitzer (uid)
- Besitzergruppe (gid)
- Größe in Byte (bei Geräteadressen statt dessen Major und Minor Device number)
- Datum der letzten Änderung
- Datum des letzten Zugriffs
- Datum der letzten Änderung dieser Verwaltungsinformationen
- Attribute (nur für das Dateisystem ext2)
- physikalischer Ort der Speicherung (Blöcke in denen die Datei abgelegt ist)

### Adressierung des Speicherblocks

In der Inode befinden sich auch die Informationen über die für die Datei verwendeten Datenblöcke. Diese Tabelle besitzt 15 Einträge.

Die ersten zwölf Einträge verweisen direkt auf jeweils einen Datenblock. Im Normalfall ist dieser logische Block 1024 Bytes groß. Es können also direkt 12 KB adressiert werden. Der Zugriff ist aufgrund der Mechanik des Caches sehr schnell.

Der dreizehnte Eintrag weist auch auf einen Datenblock. Dieser Datenblock enthält aber wiederum bis zu 256 Adressen von Datenblöcken. Es sind also nun 268 KByte adressierbar.

Der vierzehnte Eintrag weist auf einen Datenblock der 256 Einträge vom Typ des dreizehnten Eintrags enthält. Er adressiert hier also  $256 \cdot 256$  Blöcke mit 64 MByte Daten.

Der fünfzehnte Eintrag weist auf einen Datenblock mit 256 Einträgen vom Typ des vierzehnten Eintrags. Hiermit werden also  $256 \cdot 256 \cdot 256$  Blöcke mit ca. 16,6 GByte adressiert.

## 11.2 Pflege des Dateisystems

Nachdem Sie nun mit viel Mühe die Partitionen und Ihr System eingerichtet haben, sollten Sie Ihr Dateisystem auch pflegen, damit es gesund bleibt. Wie beim Menschen ist Vorsorge besser als Nachsorge.

### 11.2.1 fsck

Das wichtigste Programm für die Wartung des Dateisystems ist **fsck**. Wie auch **mkfs** (10.4.1) ist **fsck** ein Frontend zu dem Systemcheck-Programm des jeweiligen Dateisystemtyps. Es wird zur Überprüfung des Dateisystems und zur Behebung von Unstimmigkeiten verwendet.

In den meisten Fällen basiert ein solcher Dateisystemfehler auf einem Systemabsturz. Der Kernel ist dann nicht mehr in der Lage die im Cache zwischengespeicherten Daten auf die Platte zu schreiben.

```
fsck [OPTIONEN] [-t DSTYP] [GERÄT]
```

#### Optionen

<b>-a</b>	Das Programm läuft ohne Rückfragen ab (Vorsicht!)
<b>-c</b>	Untersucht auf defekte Blöcke

#### Beispiel

Um das ext2-Dateisystem der zweiten Partition der ersten IDE-Platte zu testen geben Sie den folgenden Befehl ein.

```
fsck -t ext2 /dev/hda2
```

Dies führt dazu, daß das Programm **e2fsck** aufgerufen wird.

Ist dieses Dateisystem das Wurzel-Dateisystem, dann fragt **e2fsck** nach, ob der Check auch durchgeführt werden soll. Beim Check werden getestet:

- Inodes
- Blöcke
- Dateigröße
- Verzeichnisstruktur
- Links

### 11.2.2 e2fsck

Das Tool **e2fsck** ist das Dateisystem-Checkprogramm für das ext2-Dateisystem.

```
e2fsck [OPTIONEN] GERÄT
```

#### Optionen

<b>-b BLOCK</b>	Die zu benutzende Kopie des Superblocks
<b>-f</b>	Führt den Check auch durch, wenn das System sauber erscheint.

Zum Testen können alle Dateisysteme außer dem Wurzel-System unmountet werden. Das Wurzel-System kann in folgenden Situationen getestet werden.

- Im Modus "Single-User" und "Read-Only"
- Booten mit der Diskette
- Während des normalen Systemstarts

### Reparatur des Superblocks

Sollte der Superblock beschädigt sein, so kann das Dateisystem nicht gemountet werden. Für den Dateisystemcheck geben wir deshalb eine Kopie des Superblocks an. Für ein ext2-Dateisystem lautet der Befehl dann:

```
e2fsck -f -b 16385 /dev/hda3
```

Die Option `-f` muß gesetzt werden, da das Programm sonst die Kopie des Superblocks für den echten hält und, da dieser ja in Ordnung ist, den Test beendet.

### 11.2.3 badblocks

Mit dem Tool `badblocks` können Sie ein Gerät, in den meisten Fällen eine Festplatte oder Diskette, auf defekte Datenblocks untersuchen.

```
badblocks [OPTIONEN] GERÄT [LETZTER-BLOCK] [ERSTER-BLOCK]
```

Mit der Angabe von `ERSTER-BLOCK` und `LETZTER-BLOCK` können sie auch einen Bereich auf dem Gerät bestimmen, welcher untersucht werden soll. Falls nichts angegeben wird, wird vom ersten bis letzten Block alles getestet.

Das Programm `badblock` sollte möglichst nur auf nicht gemounteten Geräten gestartet werden.

#### Optionen

<code>-b GROESSE</code>	Größe der Datenblöcke in Bytes
<code>-c ZAHL</code>	Anzahl der Blocks, die auf einmal getestet werden
<code>-f</code>	Erlaubt auch die Durchführung bei gemounteten Geräten. Vorsicht!
<code>-i DATEI</code>	Liest Liste mit bekannten defekten Blocks ein
<code>-o DATEI</code>	Schreibt gefundene defekte Block in die angegebene Datei
<code>-p VERSUCHE</code>	Anzahl der Wiederholungen, wenn kein defekter Block gefunden wurde. Defaultwert ist 0
<code>-n</code>	Nicht zerstörender Lese-und-Schreibtest, Defaulteinstellung
<code>-s</code>	Zeigt Fortschritt in Form der abgearbeiteten Blocks
<code>-v</code>	“Blubbermodus”
<code>-w</code>	Schreibmodus-Test: dieser Test löscht vorhandene Daten

#### Beispiel

Um eine Diskette zweimal auf defekte Sektoren zu untersuchen wird folgender Befehl eingegeben.

```
enterprise:~ # badblocks -p 2 -s /dev/fd0
Checking for bad blocks (read-only test): done
Checking for bad blocks (read-only test): done
```

## 11.3 Auf der Suche nach der verlorenen Datei

Wie Sie schon in Abschnitt 4.4 und Abbildung 4.1 gesehen haben, kann in diesem weit verzweigtem Verzeichnisbaum eine Datei leicht mal verloren gehen. Um diese verlorene Datei wieder zu finden, stehen Ihnen einige Tools zur Verfügung. Die Tools `whereis` (6.3.1) und `which` (6.3.2) haben Sie schon in dem Kapitel über die Hilfe kennengelernt. Neben diesen Befehlen gibt es noch `find` und `locate`.

### 11.3.1 find

Dieses Tool wird dazu benutzt das angegebene Verzeichnis und seine Unterverzeichnisse nach Dateien zu durchsuchen, die den mitgegebenen Kriterien entsprechen.

```
find [PFAD] [SUCHKRITERIEN]
```



Kriterien	Aktion
<code>-atime ±N</code>	Findet Dateien auf der Basis des letzten Zugriffs
<code>-ctime ±N</code>	Findet Dateien auf der Basis der letzten Änderung des Verzeichniseintrags
<code>-exec BEFEHL</code>	Ausführen eines BEFEHL für jede gefundene Datei
<code>-group GRUPPE</code>	Findet Dateien die zu einer bestimmten GRUPPE (GID) gehören
<code>-inum N</code>	Findet Dateien mit der Inode N
<code>-links N</code>	Findet Dateien mit N Links
<code>-mount</code>	Findet Dateien nur im gleichen Dateisystem
<code>-mtime ±N</code>	Findet Dateien auf der Basis der letzten Änderung
<code>-name MUSTER</code>	Findet Dateien, auf deren Dateiname das MUSTER zutrifft.
<code>-newer DATEI</code>	Findet Dateien, die jünger als die angegebene DATEI sind.
<code>-ok BEFEHL</code>	Ausführen eines BEFEHL für jede gefundene Datei mit vorheriger Sicherheitsabfrage
<code>-perm NNN</code>	Findet Dateien, auf die Rechtemaske NNN zutrifft.
<code>-print</code>	Zeigt die gefundenen Dateien mit ihrem Pfad an. (Standard)
<code>-size N[c]</code>	Findet Dateien in Abhängigkeit von ihrer Größe (N Blocks oder N Zeichen).
<code>-type C</code>	Findet Dateien vom gegebenen Typ: b = Blockgerät, c = Zeichengerät, d = Verzeichnis, l = Link oder f = normale Datei
<code>-user BENUTZER</code>	Findet die Dateien des angegebenen Benutzers
<code>-xdev</code>	Durchsucht nur das aktuelle Dateisystem

Tabelle 11.1: Suchkriterien des `find`-Befehls

Werden keine Kriterien angegeben, so zeigt `find` alle Dateien im Verzeichnis und seinen Unterverzeichnissen an. Wird kein Pfad angegeben, so dient das aktuelle Arbeitsverzeichnis als Startverzeichnis. Um die Anzahl der gefundenen Dateien einzuschränken, können Kriterien für die Eigenschaften der gesuchten Dateien angegeben werden. Tabelle 11.1 liefert einen Überblick über die wichtigsten Kriterien.

Einzelne Suchkriterien sind logisch verknüpfbar, d. h., es können Suchkriterien aus verschiedenen Teilbedingungen zusammengesetzt werden. Numerische Angaben als Bestandteil eines Suchkriteriums lassen sich auf drei Weisen benutzen: `N`, `+N` und `-N`, wobei `N` eine ganze Zahl ist. Das Kriterium `-links` veranlaßt z. B. eine Suche nach Dateien mit einer bestimmten Anzahl von Links und erfordert eine entsprechende numerische Angabe. So bedeuten

- `-links 3` Suche nach Dateien mit 3 Links
- `-links +3` Suche nach Dateien mit mehr als 3 Links
- `-links -3` Suche nach Dateien mit weniger als 3 Links

### Beispiele zu `find`

```
find -name "super"
```

Findet alle Dateien im aktuellen Verzeichnis und darunter, die mit "super" beginnen. Dabei sollte das Suchmuster in Anführungszeichen stehen.

```
find /usr -type d -print
```

sucht nach allen Verzeichnissen, die unterhalb von `/usr` existieren und zeigt ihre Pfadnamen an.

```
find / -name tty* -print
```

liefert die Pfade aller Dateien des gesamten Dateisystems, die mit "tty" beginnen.

```
find / -user 406 -ok rm {} \;
```

sucht im gesamten Dateisystem nach Dateien des Benutzers mit der uid 406. Gefundene Dateien werden bei positiver Beantwortung einer Sicherheitsabfrage gelöscht.

```
find . -newer .version_time -print
```

zeigt alle Dateien des aktuellen Verzeichnisses, die jüngeren Datums sind als die Datei `.version_time`.

```
find / -size +10k -user tapico -print
```

zeigt alle Dateien die größer als 10 kB sind und dem Benutzer **tapico** gehören.

```
find ~ -atime -10 -print
```

findet alle Dateien, auf die innerhalb der letzten 10 Tage zugegriffen wurde.

Bei Angabe von mehreren Kriterien wird immer von einer UND-Verknüpfung ausgegangen.

```
find / -size +10k -user tapico -print
```

zeigt alle Dateien die größer als 10 kB sind und dem Benutzer **tapico** gehören.

Dies kann durch **-a** betont werden.

```
find / -size +10k -a -user tapico -print
```

Eine ODER-Verknüpfung erreicht man durch **-o**.

```
find ~ -name '*.htm' -o -name '*.html' -print
```

sucht nach allen Dateien, die auf ".htm" oder ".html" enden.

Die Negation eines Kriteriums wird durch das **!** erreicht.

```
find / \! -name '*' -print
```

findet alle Dateien, die nicht mit einem Tilde ~ enden. Allerdings stellt die Verwendung des Negationsoperators "!" ein Problem dar, da dieser in der Bash eine besondere Bedeutung hat. Er muß deshalb maskiert werden. Im Zweifelsfall bietet es sich an die längere Version **-not** zu verwenden.

Der Befehl **find** kann auch zur Suche nach weiteren Hardlinks einer Datei verwendet werden.

```
ole@enterprise:~> ls -li /bin/gzip
 926671 -rwxr-xr-x   3 root    root      52964 2003-09-23 18:52 /bin/gzip
ole@enterprise:~> find /bin -inum 926671
/bin/gzip
/bin/zcat
/bin/gunzip
```

### 11.3.2 Suchen mit locate

Eine schneller Suchmöglichkeit als **find** ist der Befehl **locate**. Dieser stützt sich auf eine Datenbank und muß nicht wie **find** das Dateisystem direkt durchsuchen. Allerdings muß die Datenbank immer auf den aktuellsten Stand sein um auch die Dateien zu finden. Bei neueren SuSE-Versionen gehört der Befehl nicht mehr zur Standardinstallation. Bei der Suche nach Programmdateien ist oft der Befehl **whereis** schneller und der Befehl **find** ist weitaus mächtiger. Um den Befehl nutzen zu können, müssen Sie bei SuSE 9.0 das Paket *findutils-locate* installieren.

#### locate

Auf der Basis der Datenbank */var/lib/locatedb* liefert **locate** die Position von Dateien im Verzeichnisbaum.

```
locate [DATEINAME]
```

Der Befehl **locate XYZ** ist wesentlich schneller als **find / -name XYZ**. Allerdings muß auch immer die Datenbank auf dem neuesten Stand sein. Die Anwendung ist sehr einfach, so zeigt der Befehl

```
locate "*.ps"
```

alle Dateien an, die mit ".ps" enden.

#### /var/lib/locatedb

Die Datei */var/lib/locatedb* enthält eine Liste aller Dateien des Dateisystems zu einem bestimmten Zeitpunkt. Sie ist eine große Binärdatei, die Sie mit dem Befehl

```
od -c /var/lib/locatedb | less
```

betrachten können.

Mit dem Befehl **updatedb** wird die Datenbank auf den neuesten Stand gebracht.

## updatedb

Das Kommando `updatedb` bringt die für `locate` wichtige Datenbank `/var/lib/locatedb` auf den neuesten Stand.

`updatedb` [OPTIONEN]

### Optionen

<code>--localpaths</code>	Die lokalen Verzeichnisse werden mit eingebunden
<code>--netpaths</code>	Die Netzwerk-Verzeichnisse werden mit eingebunden
<code>--prunepaths LISTE</code>	Liste der ausgeschlossenen Verzeichnisse
<code>--output DATEI</code>	Name der Ausgabedatei

Für eine generelle Konfiguration des Befehls kann eine Datei `updatedb.conf` benutzt werden.

In vielen Distributionen wird das Update automatisch durch ein mit `cron` gestartetes Skript durchgeführt. Bei SuSE 9.0 ist das Skript `updatedb` im Verzeichnis `/etc/cron.daily` verantwortlich. Die Datei sieht in einer leicht gekürzten Version so aus.

```
enterprise:~ # cat /etc/cron.daily/updatedb
#!/bin/sh
...
# Copyright (c) 2003 SuSE Linux AG, Nuernberg, Germany.
# Author: Burchard Steinbild <bs@suse.de>, 1996
#       Florian La Roche <florian@suse.de>, 1996

# paranoia settings
umask 022

PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH

# get information from /etc/rc.config
if [ -f /etc/sysconfig/locate ] ; then
    . /etc/sysconfig/locate
fi

# update database for locate
if [ -n "$RUN_UPDATEDB" -a "$RUN_UPDATEDB" = "yes" -a \
    -x /usr/bin/updatedb ] ; then

    # avoid error messages from updatedb when using user nobody for find.
    cd /

    PARAMS="test -n "$RUN_UPDATEDB_AS" && \
        fgrep localuser /usr/bin/updatedb > /dev/null && \
        echo --localuser=$RUN_UPDATEDB_AS"

    PARAMS="$PARAMS 'test -n "$UPDATEDB_PRUNEPATHS" && \
        echo --prunepaths=\'$(eval echo $UPDATEDB_PRUNEPATHS)\`'"

    ...

    eval nice -n 19 /usr/bin/updatedb $PARAMS 2> /dev/null
fi
exit 0
```

## 11.4 Disk Quotas

Disk Quotas sind ein Mittel um den Platz, den jeder Benutzer oder Gruppe benutzen kann, einzuschränken. Sie erleichtern dem Administrator die Verwaltung des Systems. Diese Speichernutzungsreglementierung ar-

beitet Partitionsweise. Für die Partitionen können verschiedenen Quoten oder gar keine festgelegt werden. Um diese Funktion nutzen zu können, muß die Disk-Quota-Unterstützung auch in den Kernel einkompiliert werden.

Auch können für einzelne Benutzer oder für Gruppen Quoten festgelegt werden. Diese Quoten sind unabhängig voneinander. Erreicht das Kontingent der Gruppe oder des Benutzers das Limit, kann der Benutzer keine Daten mehr speichern.

Sollte ein Benutzer durch die Gruppenbegrenzung keinen Platz mehr haben, kann er das dadurch umgehen, daß er die Gruppe wechselt.

Im Normalfall sind die Festplattenquoten abgeschaltet. Um sie wieder zu aktivieren ist ein Eintrag für die Partition in `/etc/fstab` notwendig. Also Option wird der Platte der Eintrag `usrquota` für Benutzer und `grpquota` für Gruppen hinzugefügt.

```
/dev/hda1      /          ext2      defaults,grpquota      1 1
/dev/hda2      /home      ext2      defaults,usrquota      1 1
/dev/hda3      /server    ext2      defaults,usrquota,grpquota 1 1
```

Der Superuser muß dann in der Wurzel der Partition eine Datei `quota.user` bzw. `quota.group` einrichten. Diese Dateien sollten nur für den Superuser schreib- und lesbar sein.

```
root@defiant: ~> touch /quota.user /quota.group
root@defiant: ~> chmod 600 /quota.user /quota.group
```

Danach muß das System neu gestartet werden, um die Änderungen im System gültig zu machen. Um die Quoten einzuschalten wird das Kommando `quotaon` und um die Quoten auszuschalten das Kommando `quotaoff` benutzt. Um die Quoten einzurichten, wird der Befehl `edquota` verwendet. Dabei kann die Anzahl der Inodes, der Speicherplatz oder beides als Begrenzung dienen. Dabei spielen drei Parameter eine wichtige Rolle.

**Weiche Limit** (*soft limit*): Die durch das weiche Limit gesetzte Grenze kann für eine bestimmte Zeit überschritten werden.

**Harte Limit** (*hart limit*): Nach Erreichen des harten Limits ist es nicht mehr möglich Dateien zu erstellen. Es gibt keine Gnadenfrist.

**Gnadenfrist** (*grace period*): Für diesen Zeitraum kann das weiche Limit überschritten werden. Standard-einstellung sind sieben Tage.

### 11.4.1 quota

Mit dem Befehl `quota` kann der Superuser die Quoten überprüfen. Der normale Benutzer kann nur seine Quoten und die seiner Gruppe sehen.

```
quota [OPTIONEN] [WER]
```

Normal zeigt `quota` die Quoten der in `/etc/fstab` eingetragenen Dateisysteme.

#### Optionen

<code>-u BENUTZER</code>	Zeigt die Quoten des angegebenen Benutzers
<code>-g</code>	Zeigt neben den persönlichen Quoten auch die Quoten der Gruppen, in denen der Benutzer Mitglied ist
<code>-q</code>	Zeigt nur Dateisysteme an in denen die Quote überschritten wurden.

### 11.4.2 quotaon

Der Befehl `quotaon` startet die Quotenüberwachung.

```
quotaon [OPTIONEN]
```

**Optionen**

<b>-a</b>	Die Quoten werden für alle Dateisysteme mit rw-Recht aus der <code>/etc/fstab</code> aktiviert. Dies sollte jedesmal beim Boot-Vorgang passieren.
<b>-v</b>	Zeigt die Informationen über alle Systeme, auf denen die Quoten aktiviert sind.
<b>-u</b>	Schaltet die Quoten für benannte User an
<b>-q</b>	Schaltet die Quoten für benannte Gruppen an

Die Quoten sollten beim Booten jedesmal aktiviert werden. Deshalb muß das Init-Skript um einen Eintrag erweitert werden.

```
# Check quota and then turn quota on.
if [ -x /usr/sbin/quotacheck ]
then
    echo "Checking quotas. This may take some time."
    /usr/sbin/quotacheck -avug
    echo " Done."
fi
if [ -x /usr/sbin/quotaon ]
then
    echo "Turning on quota."
    /usr/sbin/quotaon -avug
fi
```

**11.4.3 quotaoff**

Der Befehl `quotaoff` beendet die Quotenüberwachung.

`quotaoff` [OPTIONEN]

**Optionen**

<b>-a</b>	Die Quoten werden für alle Dateisysteme mit rw-Recht aus der <code>/etc/fstab</code> deaktiviert. Dies sollte jedesmal beim Shutdown-Vorgang passieren.
<b>-v</b>	Zeigt die Informationen über alle Systeme, auf denen die Quoten deaktiviert sind.
<b>-u</b>	Schaltet die Quoten für benannte User aus
<b>-q</b>	Schaltet die Quoten für benannte Gruppen aus

**11.4.4 edquota**

Das Kommando `edquota` ist für das Anpassen der Quoten zuständig. Es öffnet dafür den Standardeditor (Variable `EDITOR`).

`edquota` [OPTIONEN]

**Optionen**

<b>-u BENUTZER</b>	Editiert die Quote des angegebenen Benutzers
<b>-g BENUTZER</b>	Editiert die Quote der angegebenen Gruppe
<b>-p BENUTZER</b>	Dupliziert die Quote des Benutzers (mit <code>-u</code> ) oder die Quote der Gruppe (mit <code>-g</code> ).
<b>-t</b>	Editiert die weichen Limits

Der Befehl `edquota -u tapico` öffnet den Abschnitt der Datei `quota.user` im Standardeditor.

Quotas for user tapico:

```
/dev/hda3: blocks in use: 2594, limits (soft = 5000, hard = 6500)
          inodes in use: 356, limits (soft = 1000, hard = 1500)
```

Jedes Limit kann dabei drei verschiedene Werte annehmen.

- Die Zahl 0 steht für kein Limit.

- Die Zahl -1 steht für die Standardeinstellungen.
- Jede ganze Zahl größer als Null steht für die Größe in KByte oder Stück.

Die Standardeinstellungen werden in der Datei `quota.h` eingestellt.

### 11.4.5 repquota

Dieser Befehl kann dazu benutzt werden den Status der Quoten zu überprüfen.

`repquota [OPTIONEN] [DATEISYSTEM]`

Dabei werden die Nutzung und die Quoten der Partition angezeigt.

#### Optionen

- a Zeigt alle Dateisysteme an, die in `/etc/fstab` eingetragen sind.
- v Zeigt alle Quoten an, selbst wenn der Benutzer keinen Platz auf der Partition beansprucht.
- g GRUPPE Zeigt die Quoten für die angegebene Gruppe an.
- u BENUTZER Zeigt die Quoten für den angegebenen Benutzer an. Nur der Superuser kann dies für andere Benutzer machen.

### 11.4.6 quotacheck

Der Befehl `quotacheck` aktualisiert die Quoteninformationen, also wieviel Platz von den einzelnen Benutzern und Gruppen belegt wird.

`quotacheck [OPTIONEN]`

Normalerweise wird der Befehl nach einem `fsck` ausgeführt.

#### Optionen

- v Zeigt Informationen über den Verlauf des Programms an
- u Führt einen Benutzerscan aus. (UID als Argument)
- g Führt einen Gruppenscan aus. (GID als Argument)
- a Überprüft alle Dateisysteme, die in der `/etc/fstab` stehen.
- R Im Zusammenhang mit -a verwendet, um alle Systeme außer dem Wurzelverzeichnis zu testen.

### 11.4.7 quotastats

Der Befehl `quotastats` kalkuliert den benutzten Platz für jeden Benutzer und für jede Gruppe. Die Ergebnisse werden in den jeweiligen Dateien `quota.user` und `quota.group` gespeichert.

### 11.4.8 Einrichten von Disk Quotas mit SuSE

Um Disk Quotas nutzen zu können, müssen diese in den Kernel einkompiliert sein, was beim SuSE-Kernel der Fall ist. Die Quota-Utilities befinden sich im Paket *quota*, das auch installiert werden muß.

In diesem Beispiel wird die Partition `/dev/hdb1` mit dem Dateisystem `ext2` am Mountpunkt `/daten` in das Dateisystem eingebunden. Für diese Partition sollen Disk Quotas für Benutzer angelegt werden. Daher muß in der `/etc/fstab` der Eintrag für `/dev/hdb1` um die Option `usrquota` ergänzt werden.

```
/dev/hdb1      /daten  ext2    defaults,usrquota 1 3
```

Im Wurzelverzeichnis der Partition muß nun die Verwaltungsdatei für die Disk Quotas eingesetzt werden. Bei SuSE kommt "aquota" zum Einsatz, daher der abweichende Name. Auf jeden Fall darf nur *root* die Datei lesen und schreiben.

```
voyager:~ # touch /daten/aquota.user
voyager:~ # chmod 600 /daten/aquota.user
```

Nun muß die Verwaltungsdatei eingerichtet werden. Dies übernimmt das Programm quotacheck.

```
voyager:~ # quotacheck -avR
quotacheck: WARNING - Quotafile /daten/aquota.user was probably truncated.
Can't save quota settings...
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 3 directories and 1 files
voyager:~ # quotacheck -avR
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 3 directories and 1 files
```

Jetzt ist es an der Zeit für eine Benutzer Quoten anzulegen. Dies erfolgt über den Befehl edquota, der für die Eingabe der Werte den vi benutzt.

```
voyager:~ # edquota -u walter
```

Un so sollte der Eintrag lauten, wenn der Benutzer *walter* eine Beschränkung von 5 MB als weiches und hartes Limit besitzen soll.

Disk quotas for user walter (uid 510):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hdb1	0	5120	5120	0	0	0

Nach der Einrichtung sollten die Disk Quotas auch aktiviert werden. Danach liefert repquota den derzeitigen Stand der Dinge.

```
voyager:~ # quotaon /daten
voyager:/daten # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
```

		Block limits				File limits			
User		used	soft	hard	grace	used	soft	hard	grace
root	--	32	0	0		4	0	0	
walter	--	0	5120	5120		0	0	0	

```

Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2,000000
```

Zum Testen legen wir uns eine ungefähr 2 MB große Datei an. Um die Datei zu füllen benutzen wir den Befehl dd (4.5.4), der aus dem Gerät /dev/zero Daten liest und sie in die Datei testfile schreibt. Wenn nun die Datei dreimal rüberkopiert wird in den Verzeichnisbaumabschnitt /daten, sollten die Disk Quotas greifen.

```
walter@voyager:~> dd if=/dev/zero of=testfile count=2000000 bs=1
2000000+0 Records ein
2000000+0 Records aus
walter@voyager:~> cp testfile /daten/test/testfile1
walter@voyager:~> cp testfile /daten/test/testfile2
walter@voyager:~> cp testfile /daten/test/testfile3
ide0(3,65): write failed, user block limit reached.
cp: Schreiben von »/daten/test/testfile3«: Der zugewiesene Plattenplatz (Quota)
ist überschritten
```

Mit den Quotas klappt das ja schon ganz gut. Allerdings sind die Quotas nach dem Neustart des Rechners nicht mehr aktiv und müssen per Hand gestartet werden. Abhilfe schafft hier eines der vielen `init`-Skripte. Das Skript `quota` startet die Quoten und `boot.quota` sorgt dafür, daß nach einem unsauberen Beenden des Dateisystems die Quoten überprüft werden. Der Befehl `insserv` sorgt dafür, daß das angegebene Skript nun auch in den im Skript angegebenen Runleveln startet.

```
yoyager:~ # cd /etc/init.d
voyager:/etc/init.d # insserv quota
voyager:/etc/init.d # insserv boot.quota
```

Mit dem Befehl `quota` können Sie sich nun die Quoten der Benutzer anschauen.

```
yoyager:~ # quota
Disk quotas for user root (uid 0): none
voyager:~ # quota walter
Disk quotas for user walter (uid 501):
      Filesystem blocks  quota  limit  grace  files  quota  limit  grace
      /dev/hdb1   5120*  5120   5120           3      0      0
```

Manchmal kann es vorkommen, daß durch ungünstige Umstände die Anzahl der benutzten Ressourcen in der Verwaltungsdatei mit der Wirklichkeit nicht mehr übereinstimmt. Dies passiert z. B. wenn Dateien verändert wurden, während das Quota-System aus war. Der Befehl `quotacheck` ändert diese Information. Allerdings nur unter Verwendung einiger Schalter.

```
yoyager:~ # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
      Block limits
User      used  soft  hard  grace  used  soft  hard  grace
-----
root      --    32     0     0           4     0     0
walter    --   5120  5120  5120           3     0     0
```

```
Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2,000000
```

```
yoyager:~ # quotacheck -avR
quotacheck: Quota for users is enabled on mountpoint /daten so quotacheck might
damage the file.
Please turn quotas off or use -f to force checking.
yoyager:~ # quotacheck -afvR
quotacheck: Cannot remount filesystem mounted on /daten read-only so counted
values might not be right.
Please stop all programs writing to filesystem or use -m flag to force checking.
yoyager:~ # quotacheck -amfvR
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 4 directories and 1 files
yoyager:~ # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
```

```
      Block limits
User      used  soft  hard  grace  used  soft  hard  grace
-----
root      --    32     0     0           4     0     0
walter    --     0  5120  5120           0     0     0
```



Statistics:

Total blocks: 7

Data blocks: 1

Entries: 2

Used average: 2,000000

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## XIV

Arbeitsblatt  
Dateisystem und Disk Quotas

## 14.1

Zu Beginn der Aufgabe sollten Sie sich im Heimatverzeichnis befinden! Wechseln Sie das Verzeichnis nur, falls Sie dazu ausdrücklich aufgefordert werden. Einige Aufgaben lassen sich nicht ausführen! Versuchen Sie in diesen Fällen die Fehlermeldung zu interpretieren.

- 1 Loggen Sie sich als Benutzer *walter* auf der Konsole 1 ein.
- 2 Kopieren Sie die Datei `passwd` aus `/etc` ins aktuelle Verzeichnis!
- 3 Wem gehört die Datei `passwd` a) in `/etc`, b) im aktuellen Verzeichnis?
- 4 Geben Sie der Datei `passwd` den zusätzlichen Hardlink `meingottwalter`!
- 5 Kopieren Sie die Datei `/bin/mv` ins aktuelle Verzeichnis !
- 6 Legen Sie ein Verzeichnis `test` an !
- 7 Kopieren Sie alle Dateien des aktuellen Verzeichnisses ins Verzeichnis `test` und wechseln Sie nach `test`!
- 8 Benennen Sie `passwd` um in `halleluja`!
- 9 Stellen Sie fest, wie viele Hardlinks die Datei `passwd` in `/etc` hat.
- 10 Suchen Sie die Datei `passwd` im gesamten Dateisystem !
- 11 Suchen Sie nach allen Dateien mit Namen `utmp` im gesamten Dateisystem!
- 12 Erstellen Sie in Ihrem Heimatverzeichnis das Verzeichnis `teest2`!
- 13 Geben Sie der Datei `mv` in Ihrem Heimatverzeichnis den zusätzlichen Namen (Softlink) `move`!
- 14 Lassen Sie sich alle Dateien im Dateisystem anzeigen, die Ihnen gehören!
- 15 Lassen Sie sich ausführliche Informationen zu allen Dateien anzeigen, die im Verzeichnis `/usr` liegen und größer als 1 kB sind!
- 16 Wieviele Dateien gibt es im Verzeichnis `/dev` ?
- 17 Wechseln Sie zur Konsole 2 und loggen Sie sich dort als *root* ein.
- 18 Konfigurieren Sie das Heimatverzeichnis so, daß der Benutzer *Willi* dort nur 20 MB Plattenplatz belegen darf. Welchen Dienst nutzen Sie dafür. Beschreiben Sie die Vorgehensweise.

.....  
*Notizen:*

.....

# Kapitel 12

## Bootvorgang und Prozeßverwaltung

Die Art und Weise wie das Betriebssystem geladen wird, hängt davon ab, wie Ihr System eingerichtet ist. Das macht aber nichts, da die grundlegenden Schritte, wie Sie das Betriebssystem starten und zum Laufen bringen, gleich sind. Es ist sehr wichtig den Vorgang und die Schritte des Bootens zu verstehen, um Probleme beim Startvorgang beseitigen zu können.

### 12.1 Bootvorgang und Starten des Betriebssystems

#### 12.1.1 Der Bootvorgang

Jeder Bootvorgang beginnt mit dem Einschalten des Rechners. Um dem Netzteil Zeit zu geben um eine stabile Spannung zu liefern, bleibt der Prozessor erst einmal gestoppt. Sobald das Netzteil meint, es liefert eine stabile Spannung sendet es das sogenannte *Power-Good-Signal* an die Hauptplatine, die dann den Prozessor startet. Alle Intel- und Intel-kompatiblen Prozessoren starten im *Realen Modus* (engl. *real mode*). Im weiteren Verlauf werde ich mich beim Bootvorgang auf diese Prozessoren beschränken. In der Betriebsart *Realer Modus* erfolgt die Adressierung von Speicherstellen über zwei 16-bit-Register, dem *Segmentregister* und dem *Offsetregister*. Aus historischen Gründen, damals glaubte man, daß ein PC nie mehr als 640 KByte Arbeitsspeicher besitzen würde, erfolgt die Startprozedur im ersten Megabyte der Arbeitsspeicher. Um diesen Arbeitsspeicher adressieren zu können werden 20 Bit benötigt.

$$20 \text{ Bit} \equiv 2^{20} \text{ Bytes} = 1048576 \text{ Bytes} = 1 \text{ MByte} \quad (12.1)$$

Da beide verwendeten Register 16 Bit groß sind, werden vom Segmentregister nur 4 Bit benötigt. Der Speicher ist also aufgeteilt in 16 Segmente zu 65.536 Bytes. Die Adresse ergibt sich dann nach folgender Regel: Das Segmentregister wird um 4 Bit nach links verschoben. Das entspricht einer Multiplikation mit 16. Der entstehende Wert wird dann mit dem Offsetregister addiert. Schauen wir uns doch mal als Beispiel das Segmentregister 0xA000 und das Offsetregister 0x3F00 an.

SR um 4 Bit		A0000	
OR	+	3F00	(12.2)
<hr/>			
20 Bit-Adresse		A3F00	

Die Schreibweise für die Adresse lautet dann: 000A:3F00.

Nachdem nun der Prozessor angefangen hat zu arbeiten, startet er den Programmcode an der Speicherstelle 0000F:FFF0. Hier befindet sich direkt nach dem Einschalten der Initialisierungscode des BIOS. Dieser Programmcode sucht nun ab dem zehnten Segment den Speicher nach weiterem BIOS-Code von Geräten (Grafikkarte, SCSI-Controller) durch und führt ihn aus. Dies ist übrigens auch der Grund, warum die Meldung der Grafikkarte als erstes erscheint.

Nach der Initialisierung und Ausführung des Grafikkarten-BIOS startet das BIOS den *Power On Self Test*. Dieser auch als *POST* abgekürzte Test überprüft den Prozessor und die DMA- und Interrupt-Controller.

Danach führt er einen Test des Arbeitsspeichers durch, der als auffälliges Element beim Booten eigentlich jedem Anwender ins Auge fällt.

Am Ende des POST wird dafür gesorgt, daß auftretende Interrupts verarbeitet werden können. Interrupts sind Signale, die ein Programm darüber informieren, daß ein Ereignis eingetreten ist, was nun bearbeitet werden soll. In vielen Fällen wird dazu das Programm angehalten, das aufgetretene Ereignis wird abgearbeitet und danach setzt das Programm seine normale Tätigkeit fort. Denken Sie daran, daß auch das Betriebssystem nur ein Programm ist. Jedenfalls wird nun dafür gesorgt, daß im ersten Speicher-Segment<sup>1</sup> die Interrupt-Vektor-Tabelle initialisiert wird. In ihr steht für jeden Interrupt eine Sprungadresse zu dem passenden Programmcode. Nach dem Abschluß des POST wird als letzte Aktion der Interrupt 0x19<sup>2</sup> ausgelöst, dessen Programmcode nun versucht ein Betriebssystem zu starten. Dafür macht es sich auf die vorhandenen Datenträger nach einem funktionsfähigen *Master Boot Record* zu durchsuchen. Den Aufbau des Master Boot Records habe ich schon in Abschnitt 10.2 beschrieben. Das im MBR enthaltene Mini-Programm wird gestartet. Dies sorgt nun dafür, daß ein Bootloader gestartet wird. Dabei kann es sein, daß das Mini-Programm schon den ersten Teil des Bootloaders darstellt.

### 12.1.2 Linux startet

Linux zu starten ist relativ einfach. Sie müssen dazu nur den Rechner einschalten und einen *Bootloader* bzw. einen Bootmanager verwenden oder von einer Diskette booten. Als Bootmanager kommen z. B. GRUB oder LILO in Frage, die später noch behandelt werden.

Dieser Bootmanager sorgt nun dafür, daß der Kernel gestartet wird. Der Kernel befindet sich entweder auf einer Diskette oder auf der Festplatte im Verzeichnis `/boot`<sup>3</sup>. Der gepackte Kernel ist normalerweise in der Datei `vmlinuz-versionsnummer`. Bei SuSE lautet der Dateiname nur `vmlinuz`. Wie auch immer, der Name und der Dateipfad sind eigentlich unwichtig. Wichtig ist nur, daß der Kernel während des Bootvorgangs lokalisiert werden kann.

Als erstes wird der Kernel in den Speicher geladen. Normalerweise ist der Kernel gepackt. Nur der Code für das Entpacken ist natürlich in ungepackter Form enthalten. Sie können das Programm `gunzip` (13.6.1) zum Entpacken des Kernels verwenden. Der Kernel enthält wichtige Informationen wie z. B. den Ort des Wurzeldateisystems.

Während des ganzen Ladevorgangs gibt der Kernel Meldungen aus und speichert sie in der Datei `/var/log/messages` (Abschnitt 13.3.3). Um sich die Meldungen später anzusehen, können Sie den Befehl `dmesg` verwenden. Er benötigt keine Argumente und gibt die letzten Meldungen aus dem Meldungspuffer des Kernels aus.

Diese Nachrichten enthalten

- den Konsolentyp und -zeichensatz,
- die Erkennung des PCI-Bus und der eingebauten PCI-Karten,
- eine Angabe über die Prozessorgeschwindigkeit,
- den verfügbaren Arbeitsspeicher,
- den CPU-Typ,
- die Versionsnummer des Kernels,
- die geladenen Gerätetreiber,
- den zur Verfügung stehenden Swap-Speicher und
- die Netzwerkadapter mit ihren Einstellungen.

Nachdem der Kernel gestartet wurde, mountet er das Wurzel-Dateisystem, das normalerweise auf der Festplatte liegt. Nach dem Mounten wird die Kontrolle an die Programme auf der Festplatte übergeben, während der Kernel im Speicher bleibt.

---

<sup>1</sup>Adressen 0000:0000 bis 0000:FFFF

<sup>2</sup>Das ist dezimal 25.

<sup>3</sup>Bei älteren Linux-Distributionen kann er sich auch im Wurzelverzeichnis (/) befinden.

### 12.1.3 dmesg

Der Befehl `dmesg` gibt letzten Meldungen des Kernels aus. In den meisten Fällen wird er dazu benutzt um sie die Bootmeldungen nach dem Starten des Systems noch einmal in Ruhe anzuschauen.

`dmesg [OPTIONEN]`

Hier noch ein Beispiel für die vom Kernel gelieferten Meldungen über den Startvorgang.

```
root@defiant:/ > dmesg
Linux version 2.2.14 (root@defiant) (gcc version 2.95.2 19991024) #2 Wed Jul 26 20:49:06 MEST 2000
Detected 365818579 Hz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 730.73 BogoMIPS
Memory: 62800k/65472k available (1280k kernel code, 412k reserved, 936k data, 44k init, 0k bigmem)
Dentry hash table entries: 8192 (order 4, 64k)
Buffer cache hash table entries: 65536 (order 6, 256k)
Page cache hash table entries: 16384 (order 4, 64k)
CPU: Intel Celeron (Mendocino) stepping 0a
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
PCI: PCI BIOS revision 2.10 entry at 0xf0200
PCI: Using configuration type 1
PCI: Probing PCI hardware
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
TCP: Hash tables configured (ehash 65536 bhash 65536)
Initializing RT netlink socket
Starting kswapd v 1.5
Detected PS/2 Mouse Port.
pty: 256 Unix98 ptys configured
apm: BIOS version 1.2 Flags 0x0f (Driver version 1.12)
Real Time Clock Driver v1.09
loop: registered device at major 7
Uniform Multi-Platform E-IDE driver Revision: 6.30
ide: Assuming 40MHz system bus speed for PIO modes; override with idebus=xx
ALI15X3: IDE controller on PCI bus 00 dev 78
ALI15X3: not 100% native mode: will probe irqs later
ALI15X3: simplex device: DMA disabled
ide0: ALI15X3 Bus-Master DMA disabled (BIOS)
ALI15X3: simplex device: DMA disabled
ide1: ALI15X3 Bus-Master DMA disabled (BIOS)
hda: IBM-DBCA-204860, ATA DISK drive
hdc: CD-224E, ATAPI CDRom drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
ide1 at 0x170-0x177,0x376 on irq 15
hda: IBM-DBCA-204860, 4645MB w/420kB Cache, CHS=592/255/63
hdc: ATAPI 24X CD-ROM drive, 128kB Cache
Uniform CDRom driver Revision: 2.56
Floppy drive(s): fd0 is 1.44M
FDC 0 is a National Semiconductor PC87306
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
scsi : 1 host.
scsi : detected total.
Partition check:
  hda: hda1 hda2 < hda5 > hda3 hda4
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 44k freed
```

```
Adding Swap: 128516k swap-space (priority -1)
Serial driver version 4.27 with HUB-6 MANY_PORTS MULTIPORT SHARE_IRQ enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
```

## 12.2 GRUB

*GRUB* ist wie der Name *G*rand *U*nified *B*ootloader schon sagt ein Bootloader. Natürlich ist er mehr als ein Bootloader und dient auch als Bootmanager. Er ist das erste Softwareprogramm was auf einem System läuft und sorgt dafür, daß der gewünschte Betriebssystem-Kernel geladen wird. Ist dies geschehen übergibt GRUB das Kommando an Kernel und beendet sich. Wenn man es genau betrachtet ist GRUB kein Bestandteil von LINUX. GRUB kann eine ganze Reihe von freien Betriebssystemen starten. Auch proprietäre Betriebssysteme können durch den Einsatz von *Chain-Loading*<sup>4</sup> gestartet werden.

GRUB ist sehr flexibel. Der Bootloader versteht verschiedenste Dateisysteme und Kernelformate. So ist er in der Lage Betriebssysteme zu starten ohne sich die physikalische Position des Kernels auf dem Datenträger merken zu müssen. Daher reicht die Angabe des Dateinamens mit Pfad und der Partition um einen Kernel zu starten. Die Eingabe kann dabei über eine Kommandozeile oder über ein Menü erfolgen.

GRUB Ursprung stammt aus dem Jahre 1995, als *Erich Boleyn* das Betriebssystem GNU Hurd mit dem Mach 4 Kernel der University of Utah starten wollte. 1999 wurde GRUB Bestandteil des GNU-Projekts und damit freie Software, an der jeder mitschreiben kann.

### 12.2.1 Bezeichnung von Geräten

Die Bezeichnung von Geräten bei GRUB unterscheidet sich etwas von den üblichen Bezeichnungen von Linux. Schauen wir uns doch mal als erstes Beispiel die Floppy an.

(fd0)

Als erstes sollten Sie sich merken, daß aller Gerätebezeichnungen in runden Klammern stehen. Die Zahl 0 gibt an, daß dies das erste Laufwerk ist, da die Zählung von 0 beginnt. Mit dieser Angabe meint GRUB dann die gesamte Diskette. Bis jetzt also keinen großen Unterschied bei den von Linux und GRUB benutzten Bezeichnungen.

(hd0,1)

Die Buchstaben hd sagen, daß es sich hierbei um eine Festplatte handelt. Die 0 steht wiederum für das erste Laufwerk. Die Zahl hinter dem Komma gibt die Partitionsnummer an. Und hier taucht das erste Problem auf. Im Gegensatz zu Linux beginnt die Nummerierung der Partition bei GRUB auch mit 0. Diese Angabe entspricht als bei Linux dem Gerät /dev/hda2. Genau wie bei Linux haben logische Partitionen auf der erweiterten Partion ihren eigenen Nummerbereich, der bei 4 beginnt. Die folgende Angabe bezeichnet die erste logische Partition auf der zweiten Festplatte.

(hd1,4)

Wenn Sie eine bestimmte Datei im Dateisystem bezeichnen möchten, dann können Sie den Namen und Pfad der Datei direkt hinter die Laufwerksbezeichnung packen.

(hd0,5)/boot/vmlinuz

Diese Angabe bezeichnet die Datei `vmlinuz` im Verzeichnis `boot` auf der zweiten logischen Partition der ersten Festplatte.

---

<sup>4</sup>Chain-Loading ist ein Mechanismus um nicht unterstützte Betriebssysteme zu laden. Dies geschieht durch das Laden eines anderen Bootloaders. Dieses Verfahren wird z. B. bei Dos und Windows angewandt.



## 12.2.2 Die Konfigurationsdatei von GRUB

Neben der Eingabeshell stellt GRUB auch die Möglichkeit eines Menüs bereit. Die dafür zuständige Konfigurationsdatei ist `/boot/grub/menu.lst`. Die Konfigurationsdatei liegt nicht im normalen Konfigurationsverzeichnis von Linux. Dies liegt daran, daß GRUB erstens eigentlich kein Bestandteil von Linux ist und zweitens, daß die Konfigurationsdatei im Bootverzeichnis eines Betriebssystems liegen sollte.

```
# Beispiel für /boot/grub/menu.lst
#
color white/blue black/light-gray
# Starte den ersten Eintrag nach Ablauf einer Frist
default 0
# Warte 8 Sekunden auf eine Entscheidung, dann starte die Standardeinstellung
timeout 8
gfxmenu (hd0,5)/boot/message

# Linux booten
title Linux
    kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 vga=0x317 splash=silent desktop
    hdd=ide-scsi hddlun=0 showopts
    initrd (hd0,5)/boot/initrd

# Windows booten
title Windows
    root (hd0,0)
    chainloader +1

# Von Diskette booten
title Diskette
    root (fd0)
    chainloader +1

# Linux in einem abgesicherten Modus starten
title Failsafe
    kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 showopts ide=nodma apm=off
    acpi=off vga=normal nosmp noapic maxcpus=0 3
    initrd (hd0,5)/boot/initrd
```

## 12.2.3 Installation von GRUB

Am Anfang erst einmal eine Warnung. Das Installieren von GRUB auf diese Weise löscht den von anderen Betriebssystemen benutzten Bootsector. Sie sollten daher vor der Installation eine Sicherheitskopie des Bootsektors der Partition anlegen, in der Sie GRUB installieren wollen. Dies können Sie z. B. mit dem Befehl `dd` (4.5.4) erledigen. Die Sicherung des Master Boot Records ist nicht so wichtig, da dieser einfach neu zu erstellen ist. Sie können dafür z. B. unter DOS den Befehl `"fdisk /mbr"` verwenden.

Sie können GRUB nun auf zwei Arten und Weisen installieren. Sie verwenden entweder den Befehl `grub` direkt oder Sie benutzen das Skript `grub-install`.

### Direkte Installation von GRUB

Für eine direkte Installation erstellen Sie sich eine GRUB-Bootdiskette (beschrieben in Abschnitt 12.2.4) und booten den Rechner mit dieser Bootdiskette. Dann setzen Sie als erstes in der nun aktiven GRUB-Shell die GRUB-Rootpartition auf die Partition des Bootverzeichnis.

```
grub> root (hd0,5)
```

Falls Sie nicht ganz sicher sind, auf welcher Partition das Bootverzeichnis sich befindet, können Sie den Befehl `find` verwenden. Der folgende Befehl sucht nach der Datei mit dem Namen `"/boot/grub/stage1"` und gibt die gefundene Partition aus.

```
grub> find /boot/grub/stage1
(hd0,5)
```

Wenn Sie nun die Wurzelpartition richtig gesetzt haben, können Sie mit dem Befehl `setup` den GRUB-Bootsector installieren. Der folgende Befehl installiert GRUB im MBR der ersten Festplatte.

```
grub> setup (hd0)
```

Wollen Sie dagegen GRUB im Bootsector der ersten Partition auf dem ersten Laufwerk installieren, dann müssen Sie folgenden Befehl eingeben.

```
grub> setup (hd0,0)
```

Nun können Sie GRUB auch ohne Bootdiskette laden.

### Installation mit grub-install

Auch hier als erstes eine Warnung: Die Installation mit `grub-install`, das im Prinzip nur ein Shell-Skript für das eigentliche Programm `grub` ist, gilt als problematisch. In bestimmten Situationen kann die Installation schief gehen. Also halten Sie eine Bootdiskette bereit um den Rechner nach einem Fehler doch nochmal starten zu können.

Ansonsten ist die Verwendung von `grub-install` relativ einfach. Sie müssen im Prinzip nur das Laufwerk angeben, in dessen (Master-)Bootsector GRUB installiert werden soll. So installiert der folgende Befehl GRUB im MBR der ersten Platte.

```
defiant:~> grub-install /dev/hda
```

Sollte sich aber das Bootverzeichnis nicht auf der Rootpartition befinden, dann müssen Sie noch das Verzeichnis angeben. Sie haben z. B. für das Bootverzeichnis eine separate Partition angelegt. Dann muß `grub-install` folgendermaßen verwendet werden.

```
defiant:~> grub-install --root-directory=/boot /dev/hda
```

### 12.2.4 Erstellung einer GRUB-Bootdiskette

Wir brauchen für eine Grub-Bootdiskette eine leere formatierte Diskette. Die Diskette können Sie ganz einfach mit dem Befehl `fdformat` (10.4.2) formatieren.

```
defiant:~ # fdformat /dev/fd0
```

Der Befehl `badblocks` (11.2.3) wird zum Testen von Datenträgern verwendet. Wird `badblocks` mit dem Schalter `-w` zusammen verwendet, so wird ein destruktiver Schreib- und Lesetest des Datenträgers durchgeführt. Danach wissen wir ob die Diskette in Ordnung ist, und daß sie auf jeden Fall keine Daten mehr enthält.

```
defiant:~ # badblocks -w /dev/fd0
```

Wir schreiben zuerst den ersten Teil von GRUB in den MBR der Diskette. Dazu wechseln wir zuerst in das GRUB-Verzeichnis. Das GRUB-Verzeichnis nach der Anleitung `/usr/share/grub/i386-pc`. Bei SuSE finden Sie bis Version 9.0 die GRUB-Dateien im Verzeichnis `/usr/lib/grub/`, ab Version 9.1 im Verzeichnis `/usr/lib/grub/i386-pc`.

```
enterprise:~ # cd /usr/lib/grub/
enterprise:/usr/lib/grub # ls -l
insgesamt 212
-rw-r--r--    1 root    root          8544 2002-09-11 01:56 e2fs_stage1_5
-rw-r--r--    1 root    root          8176 2002-09-11 01:56 fat_stage1_5
-rw-r--r--    1 root    root          7520 2002-09-11 01:56 ffs_stage1_5
-rw-r--r--    1 root    root          9248 2002-09-11 01:56 jfs_stage1_5
-rw-r--r--    1 root    root          7712 2002-09-11 01:56 minix_stage1_5
-rw-r--r--    1 root    root        10432 2002-09-11 01:56 reiserfs_stage1_5
-rw-r--r--    1 root    root           512 2002-09-11 01:56 stage1
-rw-r--r--    1 root    root       105124 2002-09-11 01:56 stage2
-rw-r--r--    1 root    root          7168 2002-09-11 01:56 vstafs_stage1_5
-rw-r--r--    1 root    root        10024 2002-09-11 01:56 xfs_stage1_5
```

Mit dem Befehl `dd` können wir das Images des Grub-MBR (`stage1`) in den MBR der Diskette spielen.

```
enterprise:/usr/lib/grub # dd if=stage1 of=/dev/fd0 bs=512 count=1
1+0 Records ein
1+0 Records aus
```

Der zweite wesentlich größere Teil des Bootloaders (`stage2`) wird in die Blöcke dahinter geschrieben.

```
enterprise:/usr/lib/grub # dd if=stage2 of=/dev/fd0 bs=512 seek=1
205+1 Records ein
205+1 Records aus
```

Wenn wir jetzt den Rechner runterfahren und dann von der Diskette booten, sollte die Grub-Shell erscheinen. Gehen wir mal davon aus, daß der Linux-Kernel auf der zweiten Partition der ersten Platte im Verzeichnis `/boot` liegt. Dann können wir in der GRUB-Shell mit den folgenden Kommandos das Linux-System starten.

```
grub> root (hd0,1)
grub> kernel /boot/vmlinuz root=/dev/hda2
grub> boot
```

Und schon bootet das Linux-System.

Der Windows-Kernel ist leider nicht multibootfähig und daher kann GRUB ihn auch nicht direkt starten. Um trotzdem mit diesen Betriebssystemen arbeiten zu können greift GRUB zu einem Hilfsmittel. Mit einem sogenannten *Chain-Loading* startet GRUB einfach den Bootloader des Betriebssystems und startet damit die Bootsequenz. Als erstes wird das GRUB-Wurzelverzeichnis auf die entsprechende Partition gelegt. Hier ist es die erste Partition der ersten Platte. Dann wird diese Partition aktiv geschaltet. Dann wird der Bootsektor und damit der Bootloader gelesen. Nun besitzt GRUB alle Informationen um den Bootvorgang einzuleiten.

```
grub> rootnoverify (hd0,0)
grub> makeactive (hd0,0)
grub> chainloader +1
grub> boot
```

## 12.3 LILO

Der von Werner Almesberger geschriebene LILO (*Linux LOader*) war wohl der am weitesten verbreitete Bootmanager (*boot loader*) unter Linux. Inzwischen macht ihm der modernere, mächtigere und einfacher zu konfigurierende GRUB diese Position streitig. LILO kann sich auf der Festplatte oder auf einer Bootdiskette befinden, und wird ausgeführt, wenn das System geladen wird. Einige der Eigenschaften von LILO sind:

- LILO arbeitet mit DOS, LINUX, UNIX, OS/2, Windows 95/98 und Windows NT zusammen.

- LILO ersetzt den Masterbootrecord der Festplatte oder Diskette.
- LILO kann bis zu 16 verschiedene Bootimages auf verschiedenen Partitionen verwalten, wobei jedes einzelne durch ein Kennwort geschützt werden kann.
- LILO erlaubt es, daß die Boot-Sektor-Datei, die Map-Datei und das Bootimage sich auf verschiedenen Partitionen befinden kann.

LILO besteht aus zwei Teilen. Der erste Teil befindet sich normalerweise im Master Boot Record (MBR) der Festplatte. Dieser Programmteil wird während des Bootvorgangs vom BIOS gestartet. Seine Aufgabe besteht darin, den größeren zweiten Teil zu starten, der sich irgendwo anders auf einem Medium befinden kann. Der zweite Teil nun bietet dem Benutzer eine Möglichkeit zur Auswahl des gewünschten Betriebssystems, findet den Kernel, lädt ihn in den Speicher und startet ihn. Die Zweiteilung ist notwendig, da der MBR nur sehr wenig Platz (meistens nur 1 Block) bietet. Daher ist der erste Teil sehr kompakt und dient nur als Starter für den größeren zweiten Teil.

Da LILO auch andere Betriebssysteme laden kann, ist er bestens dafür geeignet als Wahlschalter für Systeme mit mehreren installierten Betriebssystemen zu agieren.

Wurde keine Wahlmöglichkeit geschaffen, dann wird das standardmäßig eingestellte Betriebssystem gebootet. Durch das Drücken einer Taste (Alt-, Shift- oder Strg-Taste) oder einem Eintrag in der `/etc/lilo.conf` kann dieser Vorgang abgebrochen und ein Boot-Prompt angezeigt werden.

BOOT:

oder auch

LIL0:

Am Boot-Prompt können Sie den Namen des zu ladenden Betriebssystems angeben. Mit der Tab-Taste oder der Eingabe eines '?' wird Ihnen eine Liste der zu bootenden Betriebssysteme angezeigt. Sollte Sie keine Entscheidung treffen, so bootet LILO nach einer voreingestellten Zeit automatisch das erste Betriebssystem in der Liste.

LIL0: <TAB>

linux\*      windoof      oldlinux      betalinux

Beim Eintippen des Namens des Betriebssystems können Kernelparameter angehängt werden. Dies wird oft dazu benutzt das System im Single-User-Modus zu starten.

Die Einstellungen von LILO werden in seiner Standardkonfigurationsdatei `/etc/lilo.conf` festgelegt. Ein Bootloader wird normalerweise bei der Installation angelegt. Soll der Bootloader später verändert werden, so müssen die Änderungen in der Konfigurationsdatei noch in das System übertragen werden. Diese Aufgabe übernimmt das Programm `lilo`.

### 12.3.1 lilo

Der Befehl `lilo` entnimmt aus der Konfigurationsdatei `/etc/lilo.conf` Informationen über die zu startenden Kernel, Bildschirmereinstellungen, Bootplatte etc. und konfiguriert damit den Bootloader.

`lilo` [OPTIONEN]

`lilo` liest die Konfigurationsdatei und codiert die Informationen auf Basis der physikalischen Festplattenkonfiguration. (Was befindet sich in welchem physikalischem Block.) Dies ist notwendig, da es zum Startzeitpunkt noch kein Dateisystem bzw. keine Dateiverwaltung gibt. Diese Informationen werden in einer Map-Datei (z. B. `/boot/map`) gespeichert.

Um LILO ganz normal zu installieren reicht ein Aufruf ohne Parameter. Es können aber verschiedenste Optionen mit übergeben werden. Eine Übersicht über diese Schalter liefert Tabelle 12.1.

Meistens werden Sie aber eine Konfigurationsdatei bei Ihren Aufrufen von LILO verwenden. Bei der Verwendung einer Konfigurationsdatei müssen Sie Schlüsselworte anstatt Optionen verwenden. Die Tabelle 12.2 zeigt die Optionen mit ihren dazugehörigen Schlüsselworten.

Option	Aktion
-b BOOTDEV	Gibt die Bootplatte an
-C DATEiname	Andere Konfigurationsdatei verwenden
-c	Beschleunigtes Booten durch zusammenfassendes Lesen von benachbarten Sektoren
-D KERNEL	Benutze den Kernel mit der angegebenen Bezeichnung
-d ZENTELSEK	Wartezeit bis zum automatischen Booten
-f DATEiname	Datei für die Plattenparameter (Standard: /etc/disktab)
-I	Fragt nach dem Pfad für den Kernel
-i DATEiname	Name der Bootsectordatei (Standard: /boot/boot.b)
-l	Erzeugt lineare Sektoradressen
-m DATEiname	Name für Map-Datei vorgeben
-P [fix ignore]	Berichtigt oder ignoriert beschädigte Partitionstabellen
-q	Listet die Namen und Orte der zu bootenden Kernel aus der Map-Datei
-R KOMMANDO	Setzt den Standard für den nächsten Bootvorgang
-r VERZEICHNIS	Ändert die Wurzel des Betriebssystems vor allem Anderen; wird für die Fehlerbehebung verwendet
-S DATEiname	Legt fest, wo der alte Bootsektor gespeichert werden soll; alte Sicherungen werden überschrieben
-s DATEiname	Legt fest, wo der alte Bootsektor gespeichert werden soll; alte Sicherungen werden nicht überschrieben
-t	Testen der Konfiguration ohne Installation
-u oder -U	Deinstallation von LILO
-v	LILO liefert ausführliche Informationen über seine Arbeit

Tabelle 12.1: Optionen des Befehls `lilo`

Option	Schlüsselwort
-b BOOTDEV	boot=BOOTDEV
-c	compact
-D KERNEL	default=KERNEL
-d ZENTELSEK	delay=ZEHNTSELSEK
-f DATEiname	disktab=DATEiname
-i DATEiname	install=DATEiname
-l	linear
-m DATEiname	map=DATEiname
-P fix	fix-table
-P ignore	ignore-table
-S DATEiname	force-backup=DATEiname
-s DATEiname	backup=DATEiname
-v	verbose=LEVEL

Tabelle 12.2: Optionen und Schlüsselwörter für LILO

### 12.3.2 /etc/lilo.conf

Die Standardkonfigurationsdatei für lilo ist `/etc/lilo.conf`. Hier ein Beispiel für die Konfigurationsdatei.

```
# LILO Konfigurations-Datei
boot=/dev/hda
vga=normal
read-only
prompt
timeout=100
# End LILO global Section
#
other = /dev/hda1
    label = win
    table = /dev/hda
#
image = /boot/vmlinuz
    root = /dev/hda3
    label = linux
#
image = /boot/vmlinuz.old
    root = /dev/hda3
    label = oldlinux
    password=hamster
```

Es können Parameter wie `append`, `ramdisk`, `read-only`, `read-write`, `root` und `vga` an den Kernel weitergegeben werden. Dies kann global oder auch in den einzelnen Image-Sektionen passieren. So kann z. B. für die Fehlerbehebung ein Kernel in eine RAM-Disk geladen werden. Genauso können Sie Ihr System für den Bootvorgang als Nur-Lesend definieren.

Wenn Sie die Konfigurationsdatei geändert haben, ist es nötig das Kommando `/sbin/lilo` auszuführen, damit die Änderungen auch in den LILO übernommen werden. Führen Sie also das Kommando nach jeder Änderung der Konfigurationsdatei aus. Auch bei Änderungen der Systemkonfiguration, wie z. B. Partitionswechsel und Kernelkompilierung muß der Befehl ausgeführt werden, damit LILO sauber weiter arbeiten kann.

#### Parameter

**boot** Der Parameter `boot` legt fest, auf welchem Datenmedium sich der Bootsektor befindet. Für PCs mit IDE-Festplatten können das Geräte wie z. B. `/dev/hda`, `/dev/hdb` etc. sein.

**read-only** Durch das Setzen dieses Parameters wird das Wurzeldateisystem als “nur lesbar” gemountet. Normalerweise wird es dann durch die weitere Startprozedur als “schreibbar” wiedergemountet.

**prompt** Wird dieser Parameter gesetzt, so wird dem Benutzer beim Start ein Bootmenü angeboten. Fehlt der Parameter, so kann der Benutzer durch Drücken der `<Shift>`-, `<Strg>`- oder `<Alt>`-Taste trotzdem ins Bootmenü gelangen.

**timeout** Der Parameter `timeout` setzt die Zeit in Zehntelsekunden fest, in der der Benutzer sein System auswählen kann, bevor das System mit der Standardeinstellung gestartet wird. Ist der Parameter `prompt` gesetzt, muß `timeout` festgelegt werden um ein automatisches Starten zu ermöglichen.

**map** Der Parameter muß nur gesetzt werden, wenn eine andere Map-Datei als die vom System voreingestellte `/boot/map` verwendet werden soll.

**install** Dieser Parameter legt die Datei fest, die als Bootsektor installiert werden soll. Voreingestellt ist `/boot/boot.b`.

**image** Dieser Parameter legt zusammen mit `label` und `root` die Kernel fest, die im Bootmenü angeboten werden sollen. Für jeden Kernel bzw. Betriebssystem wird ein Eintrag benötigt.

**label** Dieser optionale Parameter legt den Text fest, unter dem der Kernel im Auswahlmenü erscheint.

**root** Mit diesem Parameter, der stets hinter **image** kommt, wird das Wurzelverzeichnis für das Betriebssystem bzw. die Kernelvariante festgelegt.

**other** Dieser Parameter entspricht **image**, wird aber für andere Betriebssystem wie z. B. Windows verwendet.

**password** Es ist manchmal sinnvoll den Zugriff auf einzelne Images zu beschränken. Dies geschieht über die Option **password**. Hiermit können Sie für jedes Image ein individuelles Passwort festlegen. Falls Sie das machen, achten Sie auf jeden Fall darauf, daß die Datei `/etc/lilo.conf` nur für den Benutzer **root** lesbar und schreibbar ist.

**mandatory** Wird diese Option gesetzt, gilt das Passwort für dieses Image auch für alle anderen Images.

**restricted** Das Passwort wird nur abgefragt, wenn der Kernel mit zusätzlichen Parametern gestartet wird.

### 12.3.3 Sicherheit: Zugriff aufs System ohne Passwort

Was macht man, wenn man das Passwort für **root** vergessen hat. Wer schlau war hat sich inzwischen einen zweiten Administrator angelegt (8.3.6). Wenn nicht, kann man über Disketten bzw. CD ein Rettungssystem starten. Dieses ist ein eigenständiges Linux-System. Sie brauchen dann nur die Wurzelpartition (/) mounten und können nun entweder das Passwort in der Datei `/etc/shadow` ändern oder in der Datei `/etc/passwd` einem anderen Benutzer, von dem Sie dann hoffentlich noch das Passwort kennen, die UID 0 zuweisen.

Es geht auch ohne Rettungssystem. Wenn Sie beim Starten des Image den Parameter `init=/bin/bash` an den Imagennamen anhängen, startet der Kernel gleich eine Bash.

```
LILO: linux init=/bin/bash
```

Das Wurzelsystem ist aber nur als Read-Only eingebunden. Es muß daher noch als lesend und schreibend wiedergemountet werden.

```
(none):/# mount -t ext3 -o rw,remount /dev/hda2 /
EXT3 FS 2.4-0.9.17, 10 Jan 2002 on ide0(3,2), internal journal
```

Und schon können die Benutzerdateien von Hand verändert werden. Es gibt nur zwei Nachteile am System. Der Pfad ist etwas sehr dürftig und die englische Tastatur ist eingeschaltet.

Da es so einfach ist das Passwort von **root** zu ändern oder sich einen anderen Benutzer zu **root** zu machen, muß ich mein System schützen. Denken Sie daran, jedes System, an das jemand physikalisch herankommt, ist knackbar. Im einfachsten Fall baut er nur die Festplatte aus.

Das Booten eines anderen Systems über externe Medien (Floppy oder CD) verhindern Sie dadurch, daß in der Bootreihenfolge die Festplatte an erster Stelle kommt. Damit nicht jeder diese Reihenfolge umstellen kann, schützen Sie dann das BIOS noch mit einem Passwort.

Die Möglichkeit über LILO und **init** können Sie verhindern, in dem Sie alle Linux-Images mit einem Passwort schützen. Zusätzlich geben Sie noch den Parameter **restricted** dazu. Dann wird das Passwort nur abgefragt, wenn manuell Parameter beim Booten mit angegeben werden. Ohne Parameter startet das System wie sonst auch.

### 12.3.4 Fehlererkennung

Wenn LILO geladen wird, zeigt er Buchstabe für Buchstaben seinen Namen an. Dabei steht jeder Buchstabe für einen bestimmten Bereich, der abgearbeitet wird. Stürzt LILO beim Laufen ab, so kann man anhand der dargestellten Zeichen erkennen, an welcher Stelle ein Problem auftrat.

**nichts** - LILO wurde nicht geladen. Er ist entweder nicht installiert oder die Partition ist nicht aktiv.

**L FEHLERCODE** - Wahrscheinlich ein Medienfehler.

**LI** - Der erste Teil wurde erfolgreich beendet, aber der zweite Teil wurde nicht ausgeführt. Das kann einem Plattengeometriefehler liegen oder die Datei `/boot/boot.b` wurde nicht gefunden.

**LIL** - Der erste und zweite Teil wurde erfolgreich beendet. Die Map-Datei kann nicht geladen werden. Wahrscheinlich ein Medienfehler.

**LIL?** - Der Loader für den zweiten Teil wurde an eine falsche Adresse geladen. Wahrscheinlich Geometriefehler oder die Datei `/boot/boot.b` konnte nicht gefunden werden.

**LIL-** - Beschädigte Beschreibungstabelle. Wahrscheinlich ein Geometriefehler oder die Map-Datei konnte nicht gefunden werden.

**LILO** - LILO wurde erfolgreich geladen.

Es kann sein, daß Sie hexadezimale Zahle nach dem ersten 'L' sehen. Diese werden zufällig erzeugt und weisen indirekt auf eine temporäres Plattenproblem hin. Wenn LILO lädt, ist das Problem nicht behoben. Sie sollten so bald wie möglich ein `fsck` durchführen.

## 12.4 Der Init-Daemon

Das Letzte, was der Kernel macht, ist den Init-Daemon (`/sbin/initd`) zu starten. Dieser bleibt so lange aktiv, bis das System heruntergefahren wird. Damit ist `init` der erste laufende Prozeß. Alle weiteren Prozesse werden entweder vom Init-Daemon direkt oder indirekt durch Subprozesse von `init` gestartet. Dieser Daemon ist also für die Erstellung der Prozesse verantwortlich, die der Rest des Systems nutzt. Als Beispiel ist hier die Login-Shell zu nennen. Daneben sorgt er dafür, das bestimmte Prozesse nach ihrem Ende wieder neu gestartet werden. Er sorgt z. B. dafür, daß nach dem Logout die Konsole neu gestartet wird, um eine erneutes Einloggen zu ermöglichen. Bei einem Shutdown ist `init` der letzte noch laufende Prozeß, der sich um das korrekte Beenden aller anderen Prozesse kümmert.

Der Init-Daemon und seine Aktionen werden durch die Konfigurationsdatei `/etc/inittab` gesteuert.

### 12.4.1 `/etc/inittab`

Jede Anweisungszeile der Datei `/etc/inittab` besteht aus vier Feldern. Von links nach rechts gesehen sind dies die ID, der Runlevel, die Aktion und der Prozess.

#### Feld: ID

Die ID besteht aus ein oder zwei Zeichen, die den Einstiegspunkt bestimmen. Sie muß eindeutig sein und ist oft der Gerätenamen.

#### Feld: Runlevel

Dies Feld gibt den Runlevel an, für den diese Zeile gültig ist. Hier können mehrere Runlevel angegeben werden. Wird das Feld leer gelassen, dann gilt die Anweisung für alle Runlevels.

#### Feld: Aktion

Mit der Aktion wird angegeben, wie mit dem Eintrag verfahren werden soll. So kann z. B. festgelegt werden, daß der Prozess automatisch neu gestartet wird, wenn er beendet wurde. Eine Übersicht über die möglichen Einträge bietet die Tabelle 12.3. Weitere Informationen erhalten Sie mit `man inittab`.

#### Feld: Prozeß

Im letzten Feld wird das Linux-Kommando oder das Programm aufgeführt, daß in der Situation gestartet werden soll.

Das folgende Listing gibt eine gekürzte Fassung der `/etc/inittab` der SuSE-Linux 6.4 Distribution wieder.



Eintrag	Aktion
boot	Startet, wenn die /etc/inittab zum ersten Mal gelesen wird
bootwait	Startet nach den Einträgen für boot
ctrlaltdel	Startet nach der Tasteneingabe <Strg>+<Alt>+<Entf>
initdefault	Definiert den Standard-Runlevel
once	Startet beim Wechsel eines Runlevels
ondemand	Hält Prozesse in bestimmten Runlevels am Laufen
powerfail	Startet, wenn ein Stromausfallsignal empfangen wurde
sysinit	Startet, bevor die Einträge boot und bootwait ausgeführt werden.
respawn	Startet den Prozeß nach seinem Ende neu
wait	Startet den Prozeß einmal beim Erreichen des Runlevels und init wartet auf sein Ende

Tabelle 12.3: Einträge für das Feld Aktion in der /etc/inittab

```
# /etc/inittab

# default runlevel
id:2:initdefault:

# check system on startup
# first script to be executed if not booting in emergency (-b) mode
si:l:bootwait:/sbin/init.d/boot

# /sbin/init.d/rc takes care of runlevel handling
#
# runlevel 0 is halt
# runlevel S is single-user
# runlevel 1 is multi-user without network
# runlevel 2 is multi-user with network
# runlevel 3 is multi-user with network and xdm
# runlevel 6 is reboot
10:0:wait:/sbin/init.d/rc 0
11:1:wait:/sbin/init.d/rc 1
12:2:wait:/sbin/init.d/rc 2
13:3:wait:/sbin/init.d/rc 3
16:6:wait:/sbin/init.d/rc 6

# what to do in single-user mode
ls:S:wait:/sbin/init.d/rc S
~~:S:respawn:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# what to do when power fails/returns
pf::powerwait:/sbin/init.d/powerfail      start
pn::powerfailnow:/sbin/init.d/powerfail now
po::powerokwait:/sbin/init.d/powerfail    stop

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
1:123:respawn:/sbin/mingetty --noclear tty1
2:123:respawn:/sbin/mingetty tty2
3:123:respawn:/sbin/mingetty tty3
```

```
4:123:respawn:/sbin/mingetty tty4
5:123:respawn:/sbin/mingetty tty5
6:123:respawn:/sbin/mingetty tty6

# end of /etc/inittab
```

### 12.4.2 Die Runlevel

Ein Runlevel definiert einen Satz von Prozessen, der als Teil der Bootsequenz gestartet wird. Das kann von einem minimalen Satz von Prozessen zur Systemadministration bis zu einer Vielzahl von Prozessen für alle eingerichteten Geräte reichen.

Die Nummer, die mit dem Runlevels verknüpft sind, variieren mit den verschiedenen Distributionen. Sie werden in der `/etc/inittab` aufgeführt. So sieht z. B. der Eintrag für SuSE-Linux 6.4 so aus:

```
# runlevel 0 is halt
# runlevel S is single-user
# runlevel 1 is multi-user without network
# runlevel 2 is multi-user with network
# runlevel 3 is multi-user with network and xdm
# runlevel 6 is reboot
```

und der für SuSE-Linux 7.3 so aus:

```
# runlevel 0 is System halt (Do not use this for initdefault!)
# runlevel 1 is Single user mode
# runlevel 2 is Local multiuser without remote network (e.g. NFS)
# runlevel 3 is Full multiuser with network
# runlevel 4 is Not used
# runlevel 5 is Full multiuser with network and xdm
# runlevel 6 is System reboot (Do not use this for initdefault!)
```

Die Einteilung der Runlevel von 7.3 ist auch aktuell (8.1) noch gültig.

Um zu bestimmen, in welchem Runlevel Sie sich gerade befinden, können Sie den Befehl `runlevel` verwenden.

Die Runlevels 0 und 6 sind normalerweise reserviert für die Zustände `halt` und `reboot`. Aber egal wie die Nummernvergabe auch läuft, der Init-Daemon arbeitet nur die Zeilen der `/etc/inittab` ab, die mit dem Runlevel verknüpft sind.

Der Runlevel *Single-User* unterscheidet sich etwas von den anderen Runlevels. Er wird gerne zur Administration und Konfiguration des Systems verwendet, wie z. B. die Reparatur eines beschädigten Dateisystems. Niemand kann sich in das System einloggen, wenn es sich in diesem Modus befindet. Trotzdem können mehrere Prozesse gleichzeitig ausgeführt werden. Dieser Modus ist auch der einzige, in dem der Init-Daemon nicht die Datei `/etc/inittab` ausliest und abarbeitet. Normalerweise starten Sie das Programm `/bin/su` um dann als Superuser zu arbeiten.

Wenn Sie einen höheren Runlevel einstellen als den Single-User-Modus, dann startet das System immer im Multiuser-Betrieb. Dann werden die Zeilen mit den Aktionen `sysinit`, `boot` und `bootwait` ausgeführt. Oft werden in diesen Zeilen auch die Dateisysteme gemountet.

Danach führt der Init-Daemon die anderen Zeilen aus, die mit dem Runlevel verknüpft sind. Der angestrebte Runlevel wird in der Zeile

```
id:2:initdefault:
```

definiert. Denken Sie daran, daß diese Zeile keinen Prozeß startet, sondern den Runlevel festlegt, mit dem gestartet werden soll. Ein Wechsel des Runlevels erfolgt mit den Befehlen `init` und `telinit`.

Im nächsten Abschnitt wird definiert, welche Startskripte ausgeführt werden sollen. Diese Zeile könnte lauten

```
si::sysinit:/etc/rc.d/rc.sysinit
```

oder z. B. bei SuSE

```
si:l:bootwait:/sbin/init.d/boot
```

Die ID `si` am Anfang der Zeile wird intern vom Init-Daemon verwendet. Diese Zeile startet das Skript `rc.sysinit`, welches Aufgaben wie die Aktivierung der Swap-Partition, Starten von `fsck` und Mounten des Dateisystems übernimmt.

Der folgende Abschnitt definiert die Skripte, die für den jeweiligen Runlevel ausgeführt werden sollen. Sie stehen im Verzeichnis `/etc/rc.d` bzw. bei SuSE im Verzeichnis `/etc/init.d`.

```
10:0:wait:/sbin/init.d/rc 0
11:1:wait:/sbin/init.d/rc 1
12:2:wait:/sbin/init.d/rc 2
13:3:wait:/sbin/init.d/rc 3
16:6:wait:/sbin/init.d/rc 6
```

Das `rc`-Skript wird mit einem Parameter, der den Runlevel angibt, aufgerufen. Dabei führt es Skripte aus dem Verzeichnis `rcn.d` aus, wobei `n` für die Nummer des Runlevels steht.

Daneben kann `rc` auch Kernel-Module aufrufen. Dies sind Teile des Kernels, die nicht fest einkompiliert wurden. Sie können bei Bedarf nachgeladen und wieder entladen werden. Meistens handelt es sich hierbei um Gerätetreiber, die nicht oft benötigt werden.

Die Datei `/etc/modules.conf` enthält die Liste der Module mit ihren Parametern. Oft handelt es sich bei den Parametern um I/O-Adressen und Interrupts. Ein Ausschnitt aus der `/etc/modules.conf` könnte so aussehen.

```
# Aliases - specify your hardware

alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none,none

isp16_cdrom_type=Sanyo

options ne                io=0x300

alias block-major-1      rd
alias block-major-2      floppy

alias char-major-4        serial
alias char-major-5        serial
alias char-major-6        lp
alias char-major-9        st
```

Als nächstes werden die Prozesse abgearbeitet, die bei jedem Runlevel ausgeführt werden. Dies ist u. a. der `getty`-Prozeß, der die virtuellen Terminals anlegt.

```
# getty-programs for the normal runlevels
1:123:respawn:/sbin/mingetty --noclear tty1
2:123:respawn:/sbin/mingetty tty2
3:123:respawn:/sbin/mingetty tty3
4:123:respawn:/sbin/mingetty tty4
5:123:respawn:/sbin/mingetty tty5
6:123:respawn:/sbin/mingetty tty6
```

Nach dem Lesen und abarbeiten der `/etc/inittab` bleibt der Init-Daemon aktiv. Er überwacht das System auf Anweisungen den Runlevel zu ändern oder ob ein Prozeß gestoppt wurde. Wenn einer der Kindsprozesse von `init` gestoppt ist, macht der Daemon zwei Sachen.

1. Er liest die `/etc/inittab` wieder. Wenn dort der erneute Start als Aktion eingetragen ist, wird der Prozeß wieder neu gestartet.
2. Dann trägt er in die Dateien `/var/log/wtmp` und `/var/run/utmp` die Tatsache ein, daß der Prozeß beendet wurde und aus welchem Grund.

Genauso liest der Daemon die `/etc/inittab` ein, wenn der Runlevel geändert wurde oder er das Signal über einen Stromausfall bekommt.

Änderungen in der `/etc/inittab` können Sie mit einem einfachen Editor machen während das System läuft. Bevor die Änderungen aber wirksam werden, muß erst wieder die `/etc/inittab` gelesen werden. Dies passiert

- wenn der Runlevel geändert wird,
- wenn ein Stromausfall signalisiert wird und
- wenn Sie das Kommando `init q` eingeben, das direkt das erneute Lesen der Datei bewirkt.

Seien Sie sehr vorsichtig und gründlich, wenn Sie die Datei `/etc/inittab` bearbeiten wollen. Ein Fehler in der Datei kann dazu führen, daß Sie sich nicht mehr am System einloggen können. Legen Sie deshalb vorher eine Kopie der Datei an und erstellen Sie eine Bootdiskette für den Notfall. Es besteht die Möglichkeit, daß Sie noch im Single-User-Modus starten können, darauf sollten Sie aber nicht vertrauen. Auch ist es möglich, daß Sie eine Endlosschleife programmieren, so daß ein Prozeß immer wieder gestartet wird. Für diesen Fall überprüft der Init-Daemon, ob ein Prozeß innerhalb von 2 Minuten 10 mal gestartet wurde. Er gibt dann eine Warnung aus und wartet 5 Minuten, bevor er den Prozeß wiederum startet.

### 12.4.3 runlevel

Der Befehl `runlevel` ermittelt den aktuellen und den vorherigen Runlevel.

`runlevel [UTMP]`

Der Befehl liest dafür die Datei `/var/run/utmp` aus um den Runleveleintrag zu finden und gibt dann den alten und neuen Runlevel getrennt durch ein Leerzeichen aus. Falls es keinen vorherigen Runlevel gegeben hat, wie es nach einem Start der Fall ist, dann wird der Buchstabe N angezeigt.

Ist der Runlevel nicht ermittelbar, dann zeigt `runlevel` das Wort "unknown" und beendet sich mit einem Fehlersignal.

Optional kann der genaue Name der Datei `utmp` mit Pfad angegeben werden, wenn sich z. B. nicht in `/var/run` befindet.

#### Beispiel

Der Befehl `runlevel` zeigt den aktuellen und vorherigen Runlevel an.

```
ole@enterprise:~> /sbin/runlevel
N 5
```

Die Informationen dazu bezieht er dabei aus der Datei `/var/run/utmp`.<sup>5</sup>

```
ole@enterprise:~> od -c /var/run/utmp | grep ^00006
0000600 001  \0  \0  \0  5  N  \0  \0  ~  \0  \0  \0  \0  \0  \0
0000620  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000640  \0  \0  \0  \0  \0  \0  \0  \0  ~  ~  \0  \0  r  u  n  l
0000660  e  v  e  l  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
```

### 12.4.4 init

Der Befehl `init` ermöglicht das Wechseln des Runlevels.

`init [OPTIONEN] [RUNLEVEL]`

---

<sup>5</sup>Beispiel aus SuSE 8.0 und 8.1.

Beim Aufruf des Befehls sendet `init` ein Stoppsignal an alle Prozesse, die nicht für den neuen Runlevel definiert sind. Danach werden die Prozesse "getötet" und die Prozesse für den Runlevel, die noch nicht laufen, werden gestartet.

Die Eingabe des Runlevels 's' oder 'S' bewirkt einen Wechsel in den Single-User-Modus. Die Angabe 'q' führt zu einem erneuten Einlesen und Abarbeiten der `/etc/inittab`.

### Beispiel

Um das X-Window-System mit SAX zu konfigurieren, sollte es möglichst beendet werden. Ist die graphische Anmeldung aktiv, so startet sich das X-Window-System nach einer Beendigung automatisch neu für den Anmeldeschirm. Wenn nun der Windowmanager keine Option zum Wechseln in den Konsolenmodus anbietet, so kann man dies durch folgende Schritte erreichen.

1. Wechseln Sie zur Textkonsole `tty1` (oder auch einer anderen ;-)).
2. Loggen Sie sich als `root` ein.
3. Wechseln Sie mit dem Befehl `init 3` in den Runlevel für die Konsole.
4. Das X-Window-System ist jetzt noch immer aktiv. Wenn Sie es aber jetzt beenden, wird die graphische Anwendung nicht mehr automatisch gestartet.

### 12.4.5 telinit

Das Kommando `telinit` ist im Prinzip nur ein Link auf das Programm `init`.

```
telinit [OPTIONEN] [RUNLEVEL]
```

Der Befehl sorgt dafür, daß `init` erst mit 5 Sekunden Verzögerung ausgeführt wird. Mit dem Schalter `-t SEKUNDEN` kann eine andere Verzögerung definiert werden.

### 12.4.6 Automatisches Starten von Diensten

Soll ein Dienst beim Booten des Systems automatisch gestartet werden, so kann das nur durch den Daemon `initd` erfolgen. Dafür müßte jedes Programm in der `/etc/inittab` zusammen mit den passenden Runleveln eingetragen werden. Dies hat man durch die Einführung des Skriptes `/etc/init.d/rc` erleichtert.

```
enterprise:/etc/init.d # head -15 rc
#!/bin/bash
#
# Copyright (c) 1996-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Florian La Roche <feedback@suse.de> 1996
#        Werner Fink <werner@suse.de> 1994-99, 2000-2001
#
# /etc/init.d/rc    --    The Master Resource Control Script
#
# This file is responsible for starting/stopping services
# when the runlevel changes.  If the action for a particular
# feature in the new run-level is the same as the action in
# the previous run-level, this script will neither start nor
# stop that feature.
```

In dem Verzeichnis `/etc/init.d` stehen die Skripte zum Starten der Programme bzw. Dienste. Für jeden Runlevel existiert ein Verzeichnis in dem sich nun Links auf die Skripte befinden, die in diesem Runlevel gestartet werden sollen.

```

enterprise:/etc/init.d # ls -ld rc*
-rwxr-xr-x    1 root    root          6082 Mär  6 17:56 rc
drwxr-xr-x    2 root    root          4096 Mär 21 01:14 rc0.d
drwxr-xr-x    2 root    root          4096 Mai  2 22:07 rc1.d
drwxr-xr-x    2 root    root          4096 Mai  2 22:07 rc2.d
drwxr-xr-x    2 root    root          4096 Mai  6 20:19 rc3.d
drwxr-xr-x    2 root    root          4096 Mär 21 01:14 rc4.d
drwxr-xr-x    2 root    root          4096 Mai  6 20:19 rc5.d
drwxr-xr-x    2 root    root          4096 Mär 21 01:14 rc6.d
drwxr-xr-x    2 root    root          4096 Mai  2 22:07 rcS.d
enterprise:/etc/init.d # ls -lG rc1.d
insgesamt 8
lrwxrwxrwx    1 root          9 Mai  2 22:07 K03single -> ../single
lrwxrwxrwx    1 root          9 Mai  2 22:07 K12splash -> ../splash
lrwxrwxrwx    1 root          8 Mai  2 22:07 K13fbset -> ../fbset
lrwxrwxrwx    1 root        10 Mai  2 22:07 K16hotplug -> ../hotplug
lrwxrwxrwx    1 root        10 Mai  2 20:25 S07hotplug -> ../hotplug
lrwxrwxrwx    1 root          8 Mai  2 20:33 S10fbset -> ../fbset
lrwxrwxrwx    1 root          6 Mai  2 20:33 S11kbd -> ../kbd
lrwxrwxrwx    1 root          9 Mai  2 20:33 S11splash -> ../splash
lrwxrwxrwx    1 root          9 Mai  2 20:23 S20single -> ../single

```

Die Links, die mit “K” (kill) beginnen, stehen für die Dienste, die beim Verlassen des Runlevels beendet werden sollen und die Links, die mit “S” (start) beginnen, beziehen sich auf Dienste, die in diesem Runlevel zu starten sind. Im dem Bootskript selber steht oft drinnen, in welchen Runleveln dieser Dienst gestartet werden soll, wenn er überhaupt gestartet werden soll.

Um nun einen Dienst beim Booten starten zu können Sie nun entweder per Hand die Skripte mit Links versehen oder sich komfortabler mit dem Programm `insserv` behelfen. Unter YaST2 steht ihnen der Runlevel-Editor für diese Aufgabe zur Verfügung.

### 12.4.7 insserv

Das Programm `insserv` aktiviert installierte Init-Skripte, in dem es die passenden Links in den Runlevel-Verzeichnissen (`/etc/init.d/rcn.d`) setzt.

```
insserv [OPTIONEN] [SKRIPTEN]
```

Dazu wertet das Programm den Kommentarblock des Skriptes aus, wie er z. B. in diesem Skript vorkommt.

```

enterprise:/etc/init.d # head -18 splash
#!/bin/bash
# Copyright (c) 1995-2000 SuSE GmbH Nuernberg, Germany.
#
# Author: Michael Schroeder <feedback@suse.de>
#
# /etc/init.d/splash
# /usr/sbin/rcsplash
#
# System startup script for console splash screens
#
### BEGIN INIT INFO
# Provides: splash
# Required-Start: $remote_fs fbset
# Required-Stop:
# Default-Start: 1 2 3 5 S
# Default-Stop:

```

```
# Description:    Splash screen setup
### END INIT INFO
```

### Optionen

```
-r                Entfernt das Skript aus allen Runlevels
-d                Installiert das Skript in den vorgegebenen Runleveln
```

Konfiguriert wird `insserv` über die Konfigurationsdatei `/etc/insserv.conf`.

## 12.5 Linux beenden

Sie sollten **niemals** Ihr Linux-System einfach abschalten. Es ist üblich, daß der Kernel wichtige Daten wie z. B. die Inode-Tabelle oder frisch geschriebene Daten im RAM vorhält. Diese Daten werden erst dann synchronisiert, wenn der Kernel dafür Zeit hat oder Platz benötigt wird. Wenn Sie nun einfach den Rechner ausschalten, hat der Kernel keine Zeit mehr die Festplatte mit den im RAM gespeicherten Daten zu synchronisieren. Es kann daher zu Datenverlusten kommen bis zu erheblichen Fehlern im Dateisystem. Vor allem kann es zu schwerwiegenden Fehlern kommen, wenn Prozesse gerade aktiv schreibend auf die Festplatten zugreifen.

### 12.5.1 shutdown

Um das laufende System zu beenden wird der Befehl `shutdown` eingesetzt.

```
shutdown [OPTIONEN] WANN [NACHRICHT]
```

Der `shutdown`-Befehl beendet alle Prozesse in dem er das `SIGTERM`-Signal an sie sendet. Als nächstes wird `init` aufgerufen um den Runlevel zu ändern und die Dateisystem zu unmounten.

Der Shutdown-Vorgang kann zu einem bestimmten Zeitpunkt ausgelöst werden. Mit dem Stichwort `now` wird der Vorgang sofort eingeleitet. Die Angabe der Minuten als eine positiven Zahl wie z. B. `+15` löst den Vorgang nach der angegebenen Zeit aus. Es kann aber auch in der Form `hh:mm:ss` ein fester Zeitpunkt vorgegeben werden.

Wahlweise kann eine spezielle Nachricht an alle eingeloggten Benutzer gesendet werden. Wird keine Nachricht angegeben, meldet sich das System bei den Benutzern mit einer Standardnachricht.

### Optionen

```
-c                Beendet einen laufenden Shutdown-Vorgang
-f                fsck wird nach einem Neustart nicht ausgelöst
-F                fsck wird nach einem Neustart ausgelöst
-h                Anhalten des Systems nach dem Herunterfahren; Runlevel 0
-k                Sendet nur die Warnung, fährt das System aber nicht herunter
-n                Führt herunter ohne init aufzurufen
-r                Führt einen Neustart nach dem Herunterfahren aus; Runlevel 6
-t SEKUNDEN      Zeit zwischen dem "Töten" der Prozesse und dem Aufruf von init
```

Nur `root` hat das Recht den Rechner mit dem Befehl `shutdown` herunterzufahren. Wenn die Datei `/etc/shutdown.allow` existiert und der Benutzer befindet sich in der darin enthaltenen Liste, dann kann auch er den Befehl (zusammen mit dem Schalter `-a`) nutzen.

Denken Sie daran: Schalten Sie nie den Rechner aus, wenn der Shutdown-Vorgang noch nicht beendet worden ist. Der Vorgang ist beendet, wenn der Text

```
The system is halted
oder
Runlevel 0 is reached
```

erschien ist.

## 12.5.2 halt und reboot

Neben `shutdown` gibt es noch zwei weitere Befehle um den Rechner herunterzufahren. Dies sind die Befehle `halt` und `reboot`. Ihre Funktion bedarf wohl keiner weiteren Erklärung.

```
halt [OPTIONEN]
reboot [OPTIONEN]
```

Wenn Sie die Befehle `halt` oder `reboot` aufrufen, wird erst der aktuelle Runlevel überprüft. Ist das System im Runlevel 0 oder 6, dann wird der Befehl ausgeführt. Ansonsten wird das Kommando `shutdown -nf` ausgeführt.

Weder `halt` noch `reboot` senden einen Warnhinweis an die eingeloggten Benutzer noch lassen Sie eine Verzögerung bis zur Ausführung zu.

### Optionen

<code>-d</code>	Es wird nicht in die Datei <code>/var/log/wtmp</code> geschrieben
<code>-f</code>	Der Befehl <code>shutdown</code> wird nicht aufgerufen aber ein <code>halt</code> oder <code>reboot</code> versucht
<code>-i</code>	Das Netzwerkinterface wird vor dem Herunterfahren deaktiviert
<code>-n</code>	Der Befehl <code>sync</code> wird nicht vor <code>halt</code> oder <code>reboot</code> ausgeführt
<code>-p</code>	Schaltet die Energie nach dem Herunterfahren aus
<code>-w</code>	Führt das System nicht herunter, sondern schreibt nur in die Datei <code>var/log/wtmp</code>

## 12.6 Prozeßverwaltung

Jede Anfrage, die an die Shell gesendet wird, wird als Prozeß bezeichnet. Da nun Linux ein Multiuser- und Multitasking-Betriebssystem ist, laufen ständig eine ganze Reihe von Prozessen, z. B. für jeden eingeloggten Benutzer die Login-Shell (meist die `bash`) oder Prozesse, die die seriellen Schnittstellen und das Netzwerk dahingehend überprüfen, ob ein Benutzer einloggen will.

Die Verwaltung all dieser Prozesse ist eine der Hauptaufgaben des Systems. Zur Verwaltung werden die sogenannten Prozeßkenndaten verwandt. Die wichtigsten sind:

**PID:** Die Prozeßnummer, über die eine Interprozeßkommunikation erfolgen kann. Das bekanntest Beispiel hierfür ist das Versenden von Signalen. Das Signal 9 etwa tötet einen Prozeß!

**PPID:** Prozeßnummer des Elternprozesses (Parent PID). Jeder Prozeß, bis auf den ersten (besser sollte man sagen den 0.), stammt von einem anderen ab.

**Prozeßzustand** kann z. B. sein *Running* (wird gerade von der CPU bedient), *Waiting* (wartet auf ein Ereignis, z. B. eine Eingabe), *Runnable* (ist bereit, hat aber nicht die CPU zur Verfügung) und *Zombie* (ist eigentlich schon tot, wird aber noch in der Statistik geführt).

**Priorität:** Die Nice-Priorität ist eine Zahl, die dem Prozeß beim Start mitgegeben wird. Diese, sein Zustand, die Zeit, die er auf Bedienung durch die CPU gewartet hat gehen ein in die Berechnung der aktuellen Priorität, die letztlich bestimmt, wann der Prozeß wieder CPU-Zeit erhält. Diese Berechnung erfolgen periodisch.

**UID, GID:** Benutzer- und Gruppennummer, um den Benutzer zu identifizieren. UID und GID gehen normalerweise ein in die

**effektive UID, GID** über die die Rechte des Benutzers kalkuliert werden. Falls die Programmdatei mit dem `s`-Recht versehen ist, wird die effektive UID oder GID die UID oder GID des Besitzers der Programmdatei. Z. B. ist der Besitzer der Programmdatei `/etc/passwd` der Benutzer `root`. Startet ein gewöhnlicher Benutzer dieses Programm erhält dieser Prozeß die effektive UID 0, er wird für den Programmlauf zum Superuser!

**kontrollierendes Terminal:** Terminal von dem aus der Prozeß gestartet wurde. Diese Angabe ist insofern wichtig, da der Prozeß seine Ausgaben/Eingaben über diesen Terminal abwickelt! Eine Reihe von Prozessen wird gewöhnlich nicht von einem Terminal aus gestartet (sondern häufig automatisch beim



Systemstart). Solche Prozesse ohne Terminal nennt man Dämonprozesse. Sie verrichten in der Regel wichtige Systemdienste. Bekannte Dämonen sind `lpd` (Druckerdämon), `crond` (Zeitdienstedämon) oder `inetd` (für das Netzwerk zuständig).

## 12.7 Prozeß- und Systemlastüberwachung

Die Überwachung der Prozesse und der Systemlast ist eine der wichtigsten Aufgaben in der Systemadministration. Während bei Desktop-Systemen, die normalerweise kaum Ihre Ressource ausnutzen, in den meisten Fällen außer Kontrolle geratene Prozesse die meisten Schwierigkeiten verursachen, sind bei den stärker geforderten Servern oft *Flaschenhälse* in Form von ungenügend dimensionierten Ressourcen ein Problem. Als Flaschenhals bezeichnet man die schwächste Systemkomponente, die letztendlich die Geschwindigkeit des Gesamtsystems bestimmt. Solche Flaschenhälse können sein:

- eine zu langsame CPU,
- ein zu kleiner oder zu langsamer Arbeitsspeicher,
- eine zu langsame Festplatte bzw. Festplattenverbindung,
- eine unterdimensionierte Netzwerkverbindung und
- ineffektiv programmierte Anwendungen.

### 12.7.1 Load Average

Der Begriff *Load* bezeichnet die Auslastung eines Rechners, wie auch die Prozesse, die gerade ausgeführt werden. Ein Maß für die Systemauslastung ist das *Load Average*. Es ist ein Maß für die Anzahl der Prozesse, die auf ihre Verarbeitung von der CPU warten oder gerade verarbeitet werden, sich also im Zustand *Runnable* oder *Running* befinden. Natürlich ist durch die aktuelle Anzahl dieser Prozesse schlecht die Auslastung eines Systems abzuschätzen. Daher werden drei Mittelwerte (1 Minute, 5 Minuten und 15 Minuten) über die Anzahl der Prozesse gebildet. Das heißt also, wir können der Load Average entnehmen, wieviele Prozesse in der letzten Minute, den letzten 5 Minuten und den letzten 15 Minuten im Durchschnitt in der Verarbeitungsschlange von der CPU gewartet haben.

Das Load Average wird vom Kernel ermittelt und in der virtuellen Datei `/proc/loadavg` notiert. Neben den drei Mittelwerte wird auch die aktuelle Anzahl der Prozesse in der Verarbeitungsschlange, die Anzahl aller Prozesse und die PID des zuletzt ausgeführten Prozesses angegeben.

```
tapico@enterprise:~> cat /proc/loadavg
4.30 1.44 0.50 2/91 4312
```

Ein Reihe von Tools wie `uptime` (Abschnitt 12.8.4), `top` (Abschnitt 12.8.3), `w` (Abschnitt 8.7.4) und `procinfo` (Abschnitt 12.8.6) lesen diese Datei aus und zeigen das Load Average an.

Was kann nun aber aus dem Load Average geschlossen werden? Normalerweise bedeutet ein hohes Load Average, daß der Rechner stark ausgelastet ist. Ideal ist ein Load Average von 1, da dann immer ein Prozeß in der Verarbeitungsschlange ist und sofort verarbeitet werden kann. Bei einem Load Average unter 1 hat die CPU teilweise nichts zu tun gehabt. Ist der Load Average über 1, dann müssen im Schnitt die Prozesse etwas warten, bevor Sie an die CPU kommen.

Normalerweise geht ein hoher Load Average auch mit einer hohen durchschnittlichen Rechenauslastung der CPU einher. Sind beide Angaben dauerhaft für das 15-Minuten-Mittel hoch, dann wäre es ratsam die Leistung der CPU zu erhöhen. Ein hohes Load Average muß aber nicht unbedingt zusammen mit einer hohen CPU-Auslastung zusammenhängen. Schauen wir uns doch mal das folgende Beispiel an.

**Beispiel:** Ein Mailserver braucht im Schnitt 5 Minuten um eine Mail zu bearbeiten und auszuliefern. Dies ist ein völlig inakzeptable Leistung. Die Ausgabe des Befehls `top` liefert folgende Ausgabe.

```
root@mail-server:~ # top
1:31pm up 2 days, 4:24, 3 users, load average: 17.03, 16.93, 17.28
101 processes: 100 sleeping, 1 running, 0 zombie, 0 stopped
CPU0 states: 1.0% user, 4.1% system, 0.0% nice, 94.0% idle
CPU1 states: 2.0% user, 6.1% system, 0.0% nice, 91.0% idle
Mem: 125844K av, 123272K used, 2572K free, 0K shrd, 5740K buff
Swap: 257000K av, 141920K used, 115080K free 12624K cached
...
```

Auffällig ist die geringe Auslastung der beiden CPUs und das sehr hohe Load Average. Das Problem liegt also nicht in der Geschwindigkeit der CPUs, denn die drehen die meiste Zeit Däumchen. Auffällig ist, daß der Swap-Speicher stark genutzt wird. Dies sollte bei einem Server, der nur als Mail-Server dient, eigentlich nicht vorkommen. Eine Möglichkeit wäre also eine zu geringe Dimensionierung des Arbeitsspeichers. Der Rechner ist hauptsächlich mit der Ein- und Auslagerung von Speicherseiten beschäftigt, die die CPU nicht sehr stark belasten. Allerdings müßte der Mailserver schon sehr stark frequentiert werden um diesen Effekt zu erreichen.

Eine andere Möglichkeit ist eine zu langsame Festplatte. Beim Schreiben und Lesen von der Festplatte wird die CPU wenig gebraucht, allerdings blockiert der schreibende/lesende Prozess die CPU. Dies führt zu einem hohen Load Average aber zu einer kleinen CPU-Auslastung. Aber es geht noch weiter: Da die Daten nicht schnell genug geschrieben werden können, verbleiben Sie im Arbeitsspeicher und sorgen für einen größeren Speicherverbrauch. Die Katastrophe tritt in dem Moment ein, wenn der physikalische Arbeitsspeicher zu klein geworden ist und das System mit dem Auslagern auf die Festplatte beginnt. Die Zugriffe auf die langsame Festplatte erhöhen sich stark und die Verarbeitung einer eMail, die normalerweise nur Sekundenbruchteile braucht, dauert nun 5 Minuten.

Um das Problem zu lösen, sollte die Geschwindigkeit der Platte untersucht und die Zugriffsparameter optimiert werden. Reicht dies nicht aus, muß ein schnelles Plattensystem eingebaut werden. Sicherheitshalber sollte auch der physikalische Arbeitsspeicher erhöht werden um bei kurzfristigen Leistungspitzen besser puffern zu können.

## 12.8 Tools zur Prozeß- und Systemlastüberwachung

Um sich eine Übersicht über die Prozesse zu verschaffen und übelgelaunte Prozesse, die das System belasten, zu identifizieren, können eine Reihe von Tools verwendet werden.

### 12.8.1 ps

Um sich eine Liste der Prozesse und deren PIDs anzeigen zu lassen, wird am häufigsten der Befehl `ps` verwendet.

`ps [OPTIONEN]`

Die Optionen bei `ps` werden mit und ohne führenden Bindestrich geschrieben. Dies funktioniert, da `ps` keine Parameter kennt. Die Optionen mit einem Bindestrich stammen von Unix, während die Optionen ohne Bindestrich ihren Ursprung im BSD-Betriebssystem haben.

Der Befehl `ps` zeigt normalerweise die unter der eigenen UID laufenden Prozesse an. Die Option `-e` wie auch die Kombination `ax` bewirken eine Anzeige aller laufenden Prozesse, wenn auch mit unterschiedlichen Informationen.

Eine Auswahl von Optionen sortiert nach Ihrem Anwendungsgebiet finden Sie im folgenden Abschnitt.

#### Einfache Auswahl von Prozessen

**Optionen**

<b>-e</b>	Zeigt alle Prozesse an
<b>a</b>	Zeigt alle Prozesse, die einem Terminal zugeordnet sind.
<b>x</b>	Zeigt alle Prozesse, die keinem Terminal zugeordnet sind.
<b>T</b>	Zeigt alle Prozesse, die diesem Terminal zugeordnet sind.

**Auswahl von Prozessen durch Listen****Optionen**

<b>-C NAME</b>	Auswahl nach Programmnamen
<b>-G GID</b>	Auswahl nach GID
<b>-p PID</b>	Auswahl nach PID
<b>-t TERMINAL</b>	Auswahl nach Terminal
<b>-U UID</b>	Auswahl nach UID

**Ausgabeformat****Optionen**

<b>-f</b>	Ausführliche Ansicht
<b>l</b>	Sehr ausführliche Ansicht
<b>u</b>	Userorientiertes Format
<b>v</b>	Speicherorientiertes Format

**Ausgabemodifizierer****Optionen**

<b>e</b>	Zeigt die dazugehörigen Umgebungsvariablen.
<b>f</b>	Zeigt die Ausgabe als Baum (Eltern- und Kindsprozesse).
<b>h</b>	Spaltenüberschriften werden unterdrückt.

**Beispiel**

Der Befehl `ps` in normaler Ansicht und in Baumansicht. Die Option `f` erweitert dabei den Umfang der ausgegebenen Prozesse.

```
ole@enterprise:~ > ps
  PID TTY          TIME CMD
 2955 pts/5    00:00:00 bash
 2978 pts/5    00:00:00 kdvi
 2981 pts/5    00:00:37 kviewshell
 4679 pts/5    00:00:00 ps
ole@enterprise:~ > ps f
  PID TTY      STAT   TIME COMMAND
 4537 pts/3      S       0:00 less
 4535 pts/3      Z       0:00 [sh] <defunct>
 2955 pts/5      S       0:00 /bin/bash
 2978 pts/5      S       0:00 \_ /bin/sh /opt/kde3/bin/kdvi lk.dvi
 2981 pts/5      S       0:37 | \_ kviewshell dvi lk.dvi
 4682 pts/5      R       0:00 \_ ps f
 2840 pts/4      S       0:00 /bin/bash
 2296 pts/3      S       0:00 /bin/bash
```

Einzelne Prozesse können über die Prozessnummer mit oder ohne die Option `-p` bzw. `p` ausgegeben werden.

```
ole@enterprise:~> ps -p 2981
  PID TTY          TIME CMD
```

```

2981 pts/5    00:00:37 kviewshell
ole@enterprise:~> ps p 2981
  PID TTY          STAT       TIME COMMAND
 2981 pts/5    S           0:37 kviewshell dvi lk.dvi
ole@enterprise:~> ps 2981
  PID TTY          STAT       TIME COMMAND
 2981 pts/5    S           0:37 kviewshell dvi lk.dvi

```

Auswahl der Prozesse nach dem kontrollierenden Terminal.

```

ole@enterprise:~> ps -t pts/3
  PID TTY          TIME CMD
 2296 pts/3    00:00:00 bash
 4534 pts/3    00:00:00 man
 4535 pts/3    00:00:00 sh <defunct>
 4537 pts/3    00:00:00 less
 4538 pts/3    00:00:00 gzip

```

## 12.8.2 pstree

Das Kommando `ps` gibt einen Baum von allen Prozessen aus. Dadurch wird deutlich, welcher Prozeß von welchem anderen Prozeß gestartet wurde.

`ps`tree [OPTIONEN] [PID]

### Optionen

<code>-a</code>	Zeigt die Kommandozeilenargumente an
<code>-c</code>	Verhindert die kompakte Darstellung von identischen Zweigen
<code>-l</code>	Zeigt lange Zeilen an, sonst werden die Zeilen für die Bildschirmbreite umgebrochen
<code>-n</code>	Sortiert die Prozesse nach PID und nicht nach Name
<code>-p</code>	Zeigt die PIDs zu den Prozessen an
<code>-u</code>	Zeigt an, wenn die UID von Kinds- und Elternprozeß sich unterscheidet

Ohne weitere Parameter gibt `ps`tree den Baum ab dem Prozeß `init` aus. Durch Eingabe einer PID kann die Ausgabe auf einen Teilausschnitt beschränkt werden.

```

tapico@defiant:~> pstree -p 1079
kdeinit(1079) +-kdeinit(1088)
               |-kdeinit(1102) +-bash(1103) +-kdvi(1254) ---kviewshell(1255)
               |               |               ' -pstree(2415)
               |               ' -bash(1186) ---su(1193) ---bash(1194)
               ' -nedit(1207)

```

## 12.8.3 top

Ein anderer Weg die laufenden Prozesse zu betrachten ist der Befehl `top`. Er liefert Prozessinformation wie z. B. PID, Priorität, Größe, Speicher- und CPU-Nutzung. Daneben liefert er gleichzeitig Statistiken über den Speicher und die Swap-Datei, sowie über Systemlaufzeit, CPU-Auslastung und das Load Average. Im Gegensatz zu `ps` gibt `top` nicht nur einen Schnappschuß der Prozesse wieder, sondern erlaubt eine kontinuierliche Beobachtung.

Hier ein Beispiel für eine Ausgabe des Befehls `top`.

```

 8:38pm up 29 min,  3 users,  load average: 0.11, 0.08, 0.09
59 processes: 51 sleeping, 2 running, 6 zombie, 0 stopped
CPU states:  0.7% user,  0.7% system,  0.0% nice, 98.4% idle
Mem:   62844K av,   61308K used,   1536K free,   46324K shrd,   15880K buff

```

Swap: 128516K av, 380K used, 128136K free 17252K cached

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
724	tapico	9	0	1060	1060	876	R	0	0.9	1.6	0:00	top
207	root	10	0	11160	10M	2008	R	0	0.3	17.7	0:16	X
569	tapico	2	0	4272	4272	3100	S	0	0.1	6.7	0:01	kvt
1	root	0	0	200	200	172	S	0	0.0	0.3	0:05	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kupdate
4	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpiod
5	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kswapd
63	bin	0	0	396	396	320	S	0	0.0	0.6	0:00	portmap
67	root	0	0	480	480	400	S	0	0.0	0.7	0:00	rpc.ugidd
75	root	0	0	556	556	468	S	0	0.0	0.8	0:00	syslogd
79	root	0	0	784	784	328	S	0	0.0	1.2	0:00	klogd
105	root	0	0	508	492	396	S	0	0.0	0.7	0:00	rpc.mountd
108	root	0	0	500	484	396	S	0	0.0	0.7	0:00	rpc.nfsd
115	root	0	0	1536	1516	1400	S	0	0.0	2.4	0:00	httpd
116	wwwrun	0	0	1056	912	800	S	0	0.0	1.4	0:00	httpd
122	at	0	0	540	540	460	S	0	0.0	0.8	0:00	atd

Top besitzt eine Reihe von interaktiven Kommandos. Mit der Taste <h> oder <?> wird ein Hilfe-Bildschirm angezeigt.

Proc-Top Revision 1.2

Secure mode off; cumulative mode off; noidle mode off

Interactive commands are:

```
space  Update display
^L     Redraw the screen
fF     add and remove fields
oO     Change order of displayed fields
h or ? Print this list
S      Toggle cumulative mode
i      Toggle display of idle proceses
c      Toggle display of command name/line
l      Toggle display of load average
m      Toggle display of memory information
t      Toggle display of summary information
k      Kill a task (with any signal)
r      Renice a task
N      Sort by pid (Numerically)
A      Sort by age
P      Sort by CPU usage
M      Sort by resident memory usage
T      Sort by time / cumulative time
u      Show only a specific user
n or # Set the number of process to show
s      Set the delay in seconds between updates
W      Write configuration file ~/.toprc
q      Quit
```

Press any key to continue

Weil der Befehl top kontinuierlich seine Informationen auf den neuesten Stand bringt und damit den Bildschirm füllt, ist es ratsam den Befehl auf einer separaten Konsole oder X-Terminal laufen zu lassen. Interes-

sant ist `top` vor allen Dingen, weil er das interaktive “töten” von Prozessen erlaubt, sowie das Ändern von Prozeßprioritäten.

### 12.8.4 uptime

Der Befehl `uptime` zeigt die aktuelle Uhrzeit, die Laufzeit des Systems, die Anzahl der Benutzer und die Durchschnittslast (Load Average über eine, fünf und fünfzehn Minuten) des Systems an. Das Tool besitzt keine nennenswerten Optionen.

```
tapico@enterprise:~> uptime
 7:52am an 1:42, 4 Benutzer, Durchschnittslast: 0,01, 0,05, 0,01
```

### 12.8.5 free

Der Befehl `free` zeigt eine Übersicht über die Auslastung des Arbeitsspeichers.

`free [OPTIONEN]`

#### Optionen

<code>-b</code>	Angabe in Byte ( <i>byte</i> )
<code>-k</code>	Angabe in Kilobyte( <i>kB</i> )
<code>-m</code>	Angabe in Megabyte( <i>MB</i> )
<code>-o</code>	Kein Anzeigen der Puffer/Cache-Korrektur( <i>ohne</i> )
<code>-s ZEIT</code>	Kontinuierliche Anzeige im Abstand von ZEIT Sekunden( <i>switchtime</i> )
<code>-t</code>	Zeigt eine Zeile mit den Werten aller Speicher an( <i>total</i> )
<code>-V</code>	Versionsnummer()

#### Beispiel

In diesem Beispiel zeigt sich deutlich, daß die Programme nur die Hälfte des Speichers (Angaben in MByte) für sich beanspruchen und daß die andere Hälfte durch Puffer und Caches benutzt wird. Die Momentaufnahme entstand während der Kompilierung eines Kernels.

```
tapico@enterprise:~> free -mt
```

	total	used	free	shared	buffers	cached
Mem:	501	492	9	0	55	215
-/+ buffers/cache:		220	280			
Swap:	1004	28	975			
Total:	1505	521	984			

### 12.8.6 procinfo

Ein wesentlich ausführlichere Übersicht über das Gesamtsystem als die oben genannten Befehle liefert das Tool `procinfo`. Es wertet in dem Verzeichnis `/proc` vorhandenen Daten aus und zeigt diese an.

`procinfo [OPTIONEN]`

Bevor wir uns mit den Optionen des Befehls beschäftigen, schauen Sie sich doch erstmal die Informationen der Standardausgabe an.

```
tapico@ds9:~> procinfo
Linux 2.4.19-4GB (root@Pentium.suse.de) (gcc 3.2) #1 1CPU [ds9.]
```

	Total	Used	Free	Shared	Buffers	Cached
Memory:						
Mem:	255004	53904	201100	0	9808	25300
Swap:	530104	0	530104			

```

Bootup: Sun May  2 08:26:17 2004    Load average: 0.10 0.04 0.05 1/42 2962

user  :      0:00:15.06   0.3% page in :      28756   disk 1:      2724r      2500w
nice  :      0:00:00.00   0.0% page out:      28309
system:      0:00:13.13   0.2% swap in :          1
idle  :      1:31:20.36  99.5% swap out:          0
uptime:      1:31:48.55      context :      91434

irq 0:      550855 timer                irq 7:          1
irq 1:          6 keyboard                irq 8:          2 rtc
irq 2:          0 cascade [4]             irq 9:      64241 usb-uhci
irq 3:      17595 eth0                    irq 11:         0 acpi
irq 4:          1                        irq 14:      5936 ide0
irq 5:          0 Intel 82801AA-ICH        irq 15:         0 ide1

```

**Memory:** Die Informationen über die Speicherauslastung sind identisch mit denen des Programms `free` (Abschnitt 12.8.5).

**Bootup** gibt den Zeitpunkt an, an dem das System gestartet wurde.

**Load average** ist etwas ausführlicher als bei anderen Tools. Neben den üblichen 1, 5 und 15 Minuten-Mitteln (hier 0.10 0.04 0.05) der auf Bearbeitung wartenden Prozesse sind weitere Informationen erhalten. Die Anzahl der aktuell wartenden Prozesse wird im Verhältnis zur Anzahl der aktuell laufenden Prozesse (hier 1/42) angezeigt. Dazu wird auch noch die PID des aktuell laufenden Prozesses (hier 2962) angegeben.

**user** gibt die Auslastung des Systems durch Benutzerprozesse an.

**nice** gibt die Auslastung des Systems durch Prozesse mit veränderter Priorität an.

**system** gibt die Auslastung des Systems durch Systemprozesse an.

**idle** zeigt Ihnen die Zeit, in der sich die CPU gelangweilt hat, weil sie nichts zu tun hatte.

**uptime** liefert die Information, wie lange das System schon läuft. Rechnet man die Zeiten von **user**, **nice**, **system** und **idle** zusammen, so kommt man ungefähr auf den Wert von **uptime**.

**page in** Die Anzahl der Speicherseiten, die von der Platte in den Arbeitsspeicher gelesen wurden.

**page out** Die Anzahl der Speicherseiten, die vom Arbeitsspeicher auf die Platte geschrieben wurden.

**swap in** gibt die Anzahl der Speicherseiten an, die aus dem Swapspeicher in den Arbeitsspeicher verlagert wurden, während

**swap out** die Anzahl der Swapseiten angibt, die in den Swapspeicher ausgelagert wurden.

**context** Die Anzahl der *Kontextwechsel* (engl. *context switch*) seit dem Booten des Systems.

**disk 1-4:** Hier finden Sie die Informationen, wie oft auf Ihre Festplatte zugegriffen wurde.

**Interrupts:** Anzahl der Zugriffe auf und die Nutzung der Interrupts werden in diesem Abschnitt dargestellt.

## Optionen

<b>-a</b>	Zeigt alle Informationen an. ( <i>all</i> )
<b>-b</b>	Daten für das I/O-Verhalten der Platten werden in Blöcken angegeben und nicht in Anfragen. ( <i>blocks</i> )
<b>-d</b>	Speicher-, CPU-, Paging-, Swapping-, Platten-, Context- und Interrupt-Daten werden pro Sekunde und nicht total angezeigt. Enthält den Schalter <b>-f</b> .
<b>-D</b>	Wie <b>-d</b> , die Speicherdaten werden aber total angezeigt.
<b>-f</b>	Gibt kontinuierlich Informationen über das System aus wie <b>top</b> . ( <i>force</i> )
<b>-FDATEI</b>	Leitet die Ausgabe in eine Datei in. In den meisten Fällen wird ein Terminal ( <i>tty</i> ) angegeben. ( <i>file</i> )
<b>-h</b>	Zeigt die Hilfe an. ( <i>help</i> )
<b>-i</b>	Zeigt auch ungenutzte Interrupts an. ( <i>irq</i> )
<b>-m</b>	Zeigt Informationen über Module und Geräte an anstatt über CPU und Speicher. ( <i>module</i> )
<b>-nN</b>	Legt den Aktualisierungszeitraum im Zusammenhang mit dem Schalter <b>-f</b> fest. ( <i>new</i> )
<b>-r</b>	Anzeige des wirklich freien Speichers wie in <b>free</b> . ( <i>free</i> )
<b>-v</b>	Version()

## Beispiele

Für eine kontinuierliche Überwachung, wie Sie sie schon bei **top** (Abschnitt 12.8.3) kennengelernt haben, kann der Schalter **-f** verwendet werden. Die Aktualisierung erfolgt normalerweise im Abstand von 5 Sekunden. Dieser Abstand kann über den Schalter **-n** verändert werden. So aktualisiert sich die Anzeige bei folgendem Befehl alle 2 Sekunden.

```
tapico@ds9:~> procinfo -fn2
```

Anstatt der Informationen über CPU und Speicher können Sie sich auch Informationen über die Module und Gerät anzeigen lassen. Dies ermöglicht der Schalter **-m**. Wenn Sie den Schalter **-a** benutzen, werden Informationen über CPU, Speicher, Module und Geräte angezeigt.

```
tapico@ds9:~> procinfo -m
Linux 2.4.19-4GB (root@Pentium.suse.de) (gcc 3.2) #1 1CPU [ds9.]
```

Kernel Command Line:

```
root=/dev/hda5
```

Modules:

8 pppoe	1 *pppox	13 af_packet	18 usbserial
25 *parport_pc	7 lp	22 *parport	18 snd-intel8x0
62 snd-pcm	11 snd-timer	26 snd-ac97-codec	3 snd-mpu401-uar
14 snd-rawmidi	4 snd-seq-device	31 snd	3 soundcore
136 *ipv6	2 *ipt_TCPMSS	1 *ipt_TOS	1 *ipt_MASQUERADE
1 *ipt_state	3 *ipt_LOG	16 ppp_generic	5 slhc
7 printer	14 *pegasus	5 joydev	4 evdev
3 input	21 usb-uhci	55 *usbcore	26 *3c59x
3 *ipt_REJECT	2 *iptables_mangle	2 *iptables_filter	3 ip_nat_ftp
13 *iptables_nat	3 ip_conntrack_f	14 *ip_conntrack	11 *ip_tables
62 lvm-mod	75 *ext3	44 *jbd	

Character Devices:

1 mem	29 fb
2 tty	108 ppp
3 ttyS	109 lvm
4 ttyS	116 alsa
5 cua	128 ptm
6 lp	136 pts
7 vcs	162 raw
10 misc	180 usb
13 input	188 usb/tts/%d

Block Devices:

1 ramdisk
3 ide0
7 loop
9 md
22 ide1
58 lvm



## 14 sound

## File Systems:

ext3	[rootfs]	[bdev]	[proc]
[sockfs]	[futexfs]	[tmpfs]	[shm]
[pipefs]	ext2	[ramfs]	minix
iso9660	[nfs]	[devpts]	[usbdevfs]
[usbfs]			

**Kernel Command Line:** Mit diesen Parametern wurde der Kernel gestartet.

**Modules:** Hier finden Sie die Liste der geladenen Kernelmodule und deren Größe in Kilobyte.

**Character Devices/Block Devices:** Die verfügbaren Geräte werden in dieser Rubrik aufgeführt. Angeben wird neben dem Gerätenamen auch die Major-Device-Number.

**File Systems:** Eine Liste der verfügbaren Dateisysteme wird in diesem Abschnitt dargestellt. Dateisysteme, die kein aktuelles Gerät brauchen, wie z. B. das *procfs*, werden in eckigen Klammern dargestellt.

**Noch zu erledigen:** *-File Redirect output to file (usually a tty). Nice if, for example, you want to run procinfo permanently on a virtual console or on a terminal, by starting it from init(8) with a line like:*  
*p8:23:respawn:/usr/bin/procinfo -biDn1 -F/dev/tty8*

## 12.9 Prozeßadministration

Prozesse können auf verschiedenste Arten und Weisen gesteuert werden. So können Prozesse mit unterschiedlicher Priorität laufen. Sie können auf der Shell direkt ausgeführt werden oder zur Stillarbeit im Hintergrund verdonnert werden. Laufenden Prozesse können mit Signalen gesteuert oder getötet werden. Wobei es beim Killen noch deutliche Nuancen gibt.

Eine Reihe von Shell-Kommandos können zur Administration verwendet werden.

### 12.9.1 nice

Wie oft und schnell ein Programm CPU-Zeit bekommt hängt u. a. von seiner Prozeßpriorität ab. Diese Prozeßpriorität kann mit dem Befehl **nice** verringert werden. Nur der Superuser ist in der Lage einem Prozeß eine höhere Priorität zuzuordnen.

**nice** [OPTION] KOMMANDO

Das Kommando ist ein Befehl oder eine Kette von Befehlen, die mit der angegebenen Priorität ausgeführt werden können. Im Normalfall ist die Prozeßpriorität bei Verwendung von **nice** auf den Wert 10 festgelegt. Mit dem Schalter **-n ZAHL** kann eine Priorität zwischen -20 und 19 angegeben werden. Dabei bedeutet eine kleinere Zahl eine höhere Priorität und eine große Zahl eine niedrige Priorität. Nur der Superuser ist in der Lage einen negativen Wert, und damit eine höhere Priorität als normal, einzustellen.

#### Beispiel

```
nice -n 19 find / -name urmel* -print > urmelliste.txt
```

Dieser langwierige Prozeß bekommt eine sehr niedrige Ausführungspriorität zugewiesen. Er wird praktisch nur ausgeführt, wenn das System genug Zeit hat.

### 12.9.2 renice

Der Befehl **renice** erlaubt eine Änderung der Prozeßpriorität im Gegensatz zu **nice** (12.9.1) für laufende Programme.

## renice PRIORITÄT PROZESS

Dabei kann der Zielprozeß durch Angabe der PID bestimmt werden. Mit den Schaltern `-u` und `-g` kann sich auf alle Prozesse einer Benutzers oder einer Gruppe bezogen werden und mit `-p` weitere PIDs angegeben werden.

### Beispiel

Das Kommando

```
renice +5 4711 -u wwwrun kingkong -p 42
```

erniedrigt die Prozeßpriorität der Prozesse 42 und 4711, sowie aller Prozesse von den Benutzern `wwwrun` und `kingkong`. Normale Benutzer können nur auf die Priorität Ihrer eigenen Prozesse einwirken und dabei die Priorität, wie schon aus `nice` bekannt, verringern.

## 12.9.3 Vorder- und Hintergrundprozesse

Die Shell ist Kommandozeileninterpret und Programmiersprache zugleich. Die Kommandos können einzeln (synchron) oder mehrere parallel nebeneinander (asynchron) ausgeführt werden. Wenn ein synchrones Kommando ausgeführt wird, wartet die Shell bis der Befehl abgearbeitet worden ist, bevor sie weitere Eingaben akzeptiert. Dies bezeichnet man auch als einen Prozess im *Vordergrund* laufen zu lassen. Asynchrone Kommandos laufen ab, während die Shell weitere Kommandos ausführt. Dieses Kommando läuft dann im *Hintergrund*.

Also immer wenn Sie einen Befehl eingeben, läuft dieser Befehl im Vordergrund. Die Shell wartet so lange, bis der Befehl abgearbeitet worden ist. Das kann z. B. beim `find`-Befehl etwas länger dauern. Um das Terminal dabei nicht zu blockieren, kann der Prozeß in den Hintergrund gelegt werden. Dabei sollte natürlich die Ausgabe des Befehls abgefangen und in eine Datei umgeleitet werden. Ein an die Zeile angefügtes kaufmännisches Und (`&`) sorgt für das Verschieben in den Hintergrund.

```
find / -user tapico -exec rm -f {}; 2> /dev/null &
```

Ein laufendes Programm können Sie unter der `bash` mit der Tastenkombination `<Strg>+<z>` stoppen und in den Hintergrund verschieben.

Ein Job ist eine Folge von einem oder mehreren Kommandos. Also immer wenn Sie Linux einen einzelnen oder mehrere miteinander verknüpfte Befehle geben, erstellen Sie einen Job. Die Shell ist in der Lage diese Jobs zu kontrollieren und weißt diesen Jobs daher Identifikationsnummern zu.

## 12.9.4 jobs

Um sich die Jobs anzeigen zu lassen, die sich im Moment im Hintergrund befinden, können Sie den Befehl `jobs` verwenden.

```
jobs [OPTIONEN] [JOB]
```

### Optionen

<code>-r</code>	Zeigt nur laufende Jobs an
<code>-s</code>	Zeigt nur gestoppte Jobs an

```
ole@defiant:~ > jobs
[1]-  Stopped                  tail -f /var/log/warn >warning.log
[2]+  Stopped                  vi warning2.log
[3]   Running                  ping 217.89.70.62 >ping.log &
ole@defiant:~ > jobs -r
[3]   Running                  ping 217.89.70.62 >ping.log &
ole@defiant:~ > jobs -s
[1]-  Stopped                  tail -f /var/log/warn >warning.log
[2]+  Stopped                  vi warning2.log
```

Dabei zeigt `jobs` nur die Hintergrundjobs an, die von dieser Shell gestartet worden sind. Die Jobs in Shells auf anderen Terminals oder sogar in der Elternshell werden nicht dargestellt.

```
ole@defiant:~ > tail -f /var/log/warn > warning.log &
[1] 3085
ole@defiant:~ > jobs
[1]+  Running                  tail -f /var/log/warn >warning.log &
ole@defiant:~ > bash
ole@defiant:~ > tail -f /var/log/warn > warning2.log &
[1] 3108
ole@defiant:~ > jobs
[1]+  Running                  tail -f /var/log/warn >warning2.log &
```

### 12.9.5 fg

Um einen im Hintergrund laufenden Prozeß wieder in den Vordergrund zu holen, kann der Befehl `fg` verwendet werden.

```
fg [JOB]
```

Außerdem geht auch `%` oder `fg %`. Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden.

#### Beispiele

```
fg
```

holt den einzigen im Hintergrund arbeitenden Job in den Vordergrund.

```
fg %4
```

holt den Prozeß mit der Nummer 4 aus dem Hintergrund.

```
fg %joe
```

holt den Job "joe" aus dem Hintergrund hervor.

```
ole@defiant:~ > tail -f ~/.X.err &
[1] 976
ole@defiant:~ > Creating Harddisk icons...
Starting kcontrol -init...Done.
No sound device available. kaudioserver not started.
Starting krootwm...Starting kfm...Done.
Done.
No sound device available. kwmsound not started.
Starting kbgndwm...Done.
Starting kpanel...Done.
startkde: program khotkeys not found!
Max Entries = 23
fg %1
tail -f ~/.X.err

ole@defiant:~ >
```

Der Befehl `tail` wird erst in den Hintergrund geschoben, dann wieder in den Vordergrund gebracht und mit `<Strg>+<c>` abgebrochen.

### 12.9.6 bg

Wenn ein Prozeß mit `<STRG>+<Z>` gestoppt und in den Hintergrund geschoben worden ist, kann er mit dem Befehl `bg` im Hintergrund wieder gestartet werden.

Nr.	Langname	Kurzname	Bedeutung
1	SIGHUP	HUP	Hangup: Reinitialisierung des Prozesses
2	SIGINT	INT	Interrupt (wie <STRG>+<C>)
3	SIGQUIT	QUIT	Quit: Beenden
9	SIGKILL	KILL	Sofortiges Beenden des Prozesses (wird nicht ignoriert)
15	SIGTERM	TERM	Sofortiges Beenden des Prozesses (kann ignoriert werden)

Tabelle 12.4: Signale des Befehls `kill`

`bg [JOB]`

Außerdem geht auch `bg %N` oder `fg %NAME`. Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden.

### 12.9.7 kill

Es ist möglich mit den Prozessen über Signale zu kommunizieren. Das Signal kann eine Unterbrechung, eine illegale Anweisung oder andere Bedingungen anweisen. Das Kommando `kill` kann zur Sendung solcher Signale verwendet werden. Meistens wird es zum “töten” eines Prozesses benutzt.

`kill [OPTIONS] ID`

Das Standardsignal ist das Beenden des Prozesses mit *SIGTERM*. Man spricht in diesem Fall auch von Terminieren. Nur der Superuser oder der Besitzer des Prozesses dürfen Signale senden. Die ID kann die PID, % (Das wäre der aktuelle Job), %N oder %JOBNAME sein.

Eine Übersicht über die Signale liefert Tabelle 12.4.

#### Beispiel

```
kill -9 %joe
```

fordert den Job “joe” zum Selbstmord auf.

```
kill -HUP $(cat /var/run/httpd.pid)
```

führt zum Neustart des Apache-Webservers.

### 12.9.8 killall

Genau wie der Befehl `kill` (12.9.7) leitet der Befehl `killall` Signale an Prozesse weiter. Allerdings wird bei diesem Befehl nicht die PID sondern der Name des laufenden Programm angegeben. Alle Prozesse, die diesem Namen zugeordnet sind, werden dann beendet.

`killall [OPTIONS] ID`

Die sonstige Arbeitsweise ist identisch mit `kill`.

### 12.9.9 nohup

Der Befehl `nohup` startet ein Programm und schützt dieses vor dem Hangup-Signal<sup>6</sup> *SIGHUP*.

`nohup KOMMANDO`

Wenn das `KOMMANDO` die Standardausgabe und Standardfehlerausgabe aufs Terminal benutzt, so wird diese umgeleitet auf die Datei `nohup.out`. Sollte dies nicht klappen wird versucht die Ausgabe in die Datei

---

<sup>6</sup>Auf Deutsch: Auflegen

`$HOME/nohup.out` umzuleiten. Klappt dies wieder erwarten auch nicht, dann verweigert `nohup` die Ausführung des Kommandos. Die Datei `nohup.out` wird nur mit Rechten für den Besitzer erstellt. Bei bereits vorhandenen `nohup.out`-Dateien werden die Rechte nicht angefaßt.

Die Kommandos werden nicht automatisch in den Hintergrund gebracht. Dazu ist ein abschließendes Kaufmännisches Und (&) nötig. Auch die Prozeßpriorität wird nicht verändert. Hier ist zusätzlich der Befehl `nice` (12.9.1) notwendig.

Der Befehl wurde entwickelt um Programme auch nach dem Ausloggen weiter laufen zu lassen. Normalerweise bekommen auch Programme, die in den Hintergrund verschoben wurden und dort laufen, beim Ausloggen das Signal `SIGHUP` gesendet. Die führt, wenn der Programmierer es nicht explizit abfängt, zum Abbruch des Programms. Allerdings kann ich diesen Effekt bei einer SuSE 9.0 nicht nachvollziehen. Auch normal im Hintergrund gestartete Prozesse liefen nach dem Abmelden und Anmelden immer noch. Sie hatten zwar Ihren Elternprozess verloren, waren aber noch immer aktiv.

Als Endestatus bzw. Fehlercode von `nohup` kommen die Werte 126 und 127 in Frage. Bei 126 wurde das `KOMMANDO` zwar gefunden, konnte aber nicht gestartet werden. Dagegen konnte bei 127 das `KOMMANDO` nicht gefunden werden oder `nohup` hat einen Fehler. Ansonsten werden die Fehlercodes des `KOMMANDO` weitergereicht.

## Beispiele

Das Signal `SIGHUP` hat keine Auswirkung auf die `nohup` gestarteten Programme.

```
ole@enterprise:~/test> tail -f /var/log/warn > warn.log &
[2] 6082
ole@enterprise:~/test> kill -s SIGHUP 6082
[2]+  Aufgelegt          tail -f /var/log/warn >warn.log
ole@enterprise:~/test> nohup tail -f /var/log/warn > warn.log &
[2] 6087
ole@enterprise:~/test> kill -s SIGHUP 6087
ole@enterprise:~/test> ps -e | grep tail
 6087 pts/6    00:00:00 tail
```

Die Standardausgabe wird umgeleitet in die Datei `nohup.out`, die mit sehr strengen Rechten versehen ist.

```
ole@enterprise:~/test> nohup tail -f /var/log/warn &
[1] 6108
nohup: appending output to 'nohup.out'
ole@enterprise:~/test> ls -l
insgesamt 4
-rw-----  1 ole      users          655 2004-06-24 23:10 nohup.out
```

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## XV

Arbeitsblatt  
Bootvorgang und Prozesse

## 15.1

- 1 Beenden Sie das X-Window-System und arbeiten Sie nur mit den Terminals.
- 2 Ermitteln Sie die folgenden Daten aus den Bootmeldungen des Kernels.
  1. Konsolentyp:
  2. Prozessortyp:
  3. Prozessorgeschwindigkeit:
  4. PCI-Bios:
  5. Swap-Partition:
  6. Kernelversion:
  7. Kernelerstellungsdatum:
  8. Geräte:
- 3 Für welchen Zweck wird das Programm `fsck` verwendet?
- 4 Loggen Sie sich auf Terminal 1 als *root* ein.
- 5 Lassen Sie sich die Informationen über die Auslastung des Systems (CPU und Speicher) anzeigen.
- 6 Beenden Sie die Anzeige und löschen Sie den Bildschirm!
- 7 Suchen Sie im ganzen Dateisystem nach Dateien, die den Rechnernamen enthalten. Starten Sie den Befehl als Hintergrundprozeß und achten Sie auf Datenumleitung!
- 8 Lassen Sie sich die Prozeßnummer Ihrer Prozesse anzeigen! Finden Sie speziell die Prozeßnummer des Hintergrundprozesses heraus!
- 9 Killen Sie den Hintergrundprozeß!
- 10 Prüfen Sie, ob `grep` trotzdem Informationen in Dateien abgelegt hat!
- 11 Lassen Sie sich die Prozeßnummern aller Prozesse anzeigen! (Welche Prozeßnummer hat der `init`-Prozeß?)
- 12 Lassen Sie sich ausführliche Informationen zu allen laufenden Prozessen anzeigen! Welcher Terminal kontrolliert den `init`-Prozeß, von welchem Prozeß stammt der `init`-Prozeß und gibt es sonst irgendeinen Prozeß der nicht von `init` abstammt?
- 13 Starten Sie nun den Editor `vi`.
- 14 Stoppen Sie `vi` ohne das Programm zu beenden.
- 15 Starten Sie den Befehl aus Aufgabe 7 als Hintergrundprozeß. Merken Sie sich die Job-ID.
- 16 Starten Sie das X-Window-System (`startx`) ohne das die Konsole blockiert wird. Wechseln Sie, falls nötig, zur Konsole zurück (`<Strg>+<Alt>+<F1>`).
- 17 Beenden Sie den in Aufgabe 15 gestarteten Hintergrundprozeß.
- 18 Kehren Sie zu `vi` zurück und schreiben Sie die Job-ID aus Aufgabe 15 in den Text.
- 19 Stoppen Sie `vi` wieder.
- 20 Starten Sie nun den Editor `joe`.
- 21 Stoppen Sie `joe` ohne das Programm zu beenden. Welches Problem tritt auf?

XV

Arbeitsblatt  
Bootvorgang und Prozesse

15.2

- 22** Lassen Sie sich alle Jobs anzeigen.
- 23** Lassen Sie sich alle Prozesse in der Baumansicht anzeigen.
- 24** Kehren Sie zu `joe` zurück, holen Sie den Prozeß in den Vordergrund und beenden Sie das Programm.
- 25** Kehren Sie zu `vi` zurück und schreiben Sie den Satz: "Mit den Aufgaben ist jetzt Schluß." Speichern Sie den Text als `/root/blubber.txt` ab und beenden `vi`.

.....  
*Notizen:*

.....



## Kapitel 13

# Administrative Aufgaben und Datensicherung

Ein großer Teil der Arbeiten des Systemadministrators werden regelmäßig wiederholt oder müssen zu bestimmten Zeiten gestartet werden. So ist es sinnvoll, die tägliche Datensicherung erst in der Nacht zu starten um die normale Tagesarbeit nicht zu behindern. Oder die Daten der letzten Inventur sollen an die Zentrale geschickt werden. Da es sich um einige Megabyte handelt, wäre es sinnvoll die freien Leitungen in der Nacht zu nutzen. Aber warum sollte der Administrator dafür so lange in der Firma bleiben.

Für die automatische Verwaltung täglicher Aufgaben und einmaliger Aufgaben besitzt Linux die entsprechenden Ressourcen.

### 13.1 Ausführung zu anderen Zeiten: Der at-Dämon

Wenn Sie ein Programm zu einem bestimmten Zeitpunkt einmal starten wollen, stellt Ihnen Linux den *at-Dämon* zur Verfügung. Dieser Dämon sorgt dafür, das in Auftrag gegebene *Jobs* zu einem festgelegten Zeitpunkt gestartet werden. Der Zeitpunkt kann auf eine Minute genau festgelegt werden.

Dämonen sind im Prinzip auch nur Programme, die aber nicht über ein Terminal gestartet werden. Beim at-Dämon ist dies das Programm `/usr/sbin/atd`. Sie können den Dämon einfach starten, indem Sie als *root* den Befehl `atd` auf der Shell eingeben. Wie für alle Systemdienste bzw. Dämonen existiert auch für `atd` ein RC-Script. Dieses ist `/etc/init.d/atd` und ist mit dem symbolischen Link `/usr/sbin/rcatd` verknüpft. Sie sollten den Dämon möglichst über dieses Skript starten, neustarten oder beenden.

```
defiant:~ # rcatd start
Starting service at daemon           done
defiant:~ # rcatd restart
Shutting down service at daemon      done
Starting service at daemon           done
defiant:~ # rcatd stop
Shutting down service at daemon      done
```

Allerdings ist der Dämon nach einem Neustart nicht mehr aktiv. Sie müssen, wenn Sie den Dämon jedesmal aktiv haben wollen, ihn in die Runlevel eintragen. Sie können dies bei SuSE entweder über YaST und den Runlevel-Editor erledigen oder Sie benutzen den Befehl `insserv` (12.4.7). Die Runlevel, für die `insserv` das Skript einträgt, stehen in einem Kommentarfeld im Skript.

```
linux37:/etc/init.d # grep Default-Start atd
# Default-Start: 2 3 5
linux37:/etc/init.d # insserv atd
```

Um nun einen Job zu starten benötigen Sie noch ein Werkzeug, daß das entsprechende Programm, Kommando oder Skript in die Warteschlange packt. Dies ist der Befehl `at`.

### 13.1.1 at

Wenn Sie einen Job einmal zu einer bestimmten Zeit starten wollen, benutzen Sie das Kommando **at**.

**at** [OPTIONEN] ZEIT

Verantwortlich für die Ausführung des Befehls zur richtigen Zeit ist der Dämon **atd**, der im Hintergrund läuft und wie ein Wecker zum voreingestellten Zeitpunkt die entsprechenden Maßnahmen einleitet.

#### Optionen

<b>-d</b>	Löscht einen Job ( <i>delete</i> )
<b>-f DATEI</b>	Liest den Job aus der angegebenen Datei ( <i>file</i> )
<b>-l</b>	Zeigt alle Jobs des Benutzers an ( <i>list</i> )
<b>-m</b>	Sendet nach dem Abschluß des Jobs eine eMail ( <i>mail</i> )

Die Benutzung des **at**-Kommandos ist ganz einfach.

1. Geben Sie **at** zusammen mit dem Ausführungszeitpunkt an.
2. Geben Sie die Kommandos ein, die ausgeführt werden sollen.
3. Drücken Sie <Strg>+<d> um den Job abzuschicken.

```
tapico@defiant:~ > at +1 minutes
warning: commands will be executed using /bin/sh
at> who > ~/who.txt
at> <EOT>
job 9 at 2001-02-10 18:25
```

Bei der Zeitangabe stehen Ihnen mehrere Möglichkeiten zur Verfügung.

- Ein Zeitpunkt als Tag, Stunde und Minute.
- Ein Zeitpunkt von jetzt an gemessen.
- Der Wochentag als Zahl oder als Name.
- Die Verwendung von Worten wie **midnight** (00:00 Uhr), **noon** (12:00 Uhr), **teatime** (16:00 Uhr) und **now**.
- Zusätze zu Zeitangaben wie **today**, **tomorrow**, **am** und **pm**.

#### Beispiele für Zeitangaben

**Um 15:30 Uhr** at 15:30

**Am 18.03.2004 um 15:30 Uhr** at 18.03.04 15:30

**Am 18.03.2004 um 15:30 Uhr** at 03/18/04 15:30

**Am 18.03.2004 um 15:30 Uhr** at 031804 15:30

**In 3 Stunden** at +3 hours

**In 5 Minuten** at +5 minutes

**In 3 Tagen um 15:00 Uhr** at 3pm +3 days

**Jetzt** at now

### 13.1.2 atq

Der Befehl `atq` zeigt die `at`-Jobs des aktuellen Benutzers an.

```
atq
```

Die Funktion des Befehls ist identisch zu `at -l`.

### 13.1.3 atrm

Der Befehl `atrm` löscht einen wartenden `at`-Job.

```
atrm JOBNUMMERN
```

Die Funktion des Befehls ist identisch zu `at -d JOBNUMMER`.

### 13.1.4 Jobverwaltung

Jobs benötigen wie jeder Prozess Systemressourcen. Wenn nun viele Benutzer komplexe Jobs laufen lassen, kann dies signifikante Einschränkungen der Systemperformance zur Folge haben.

Mit zwei Dateien ist es möglich festzulegen, wer überhaupt Jobs starten darf. Dies sind die Dateien `/etc/at.allow` und `/etc/at.deny`. Wenn also ein Benutzer einen Job anlegen will, dann prüft das System zuerst, ob der Benutzer dazu berechtigt ist.

Existiert die Datei `/etc/at.allow` nicht, dann wird in der Datei `/etc/at.deny` nachgeschaut, ob der Zugriff für den Benutzer verboten ist.

Existiert die Datei `/etc/at.allow` und der Benutzer ist dort aufgeführt, dann darf der Benutzer den Job anlegen. Die Datei `/etc/at.deny` wird nicht überprüft. Eine leere `/etc/at.deny` erlaubt jedem Benutzer das Anlegen eines Jobs. Existiert hingegen die Datei gar nicht, dann hindert das alle Benutzer am Anlegen von Jobs. Wenn beide Dateien nicht existieren, dann darf nur der Superuser Jobs anlegen.

Auf jeden Fall sollten Sie als Systemadministrator darauf achten, daß das System nicht durch zu viele Job verstopft wird. Benutzen Sie dazu die Kommandos `atq` und `atrm`.

### 13.1.5 batch

Das Kommando `batch` legt einen `at`-Job an, der mit einer geringeren Priorität ausgeführt wird.

```
batch [OPTIONEN] [ZEIT]
```

Der Befehl `batch` entspricht dem Befehl `at -b now`. Der Schalter `-b` steht allerdings unter `at` nicht mehr zur Verfügung. Im Gegensatz zu `at` hat der Job eine sehr niedrige Priorität. Der Befehl untersucht die virtuelle Datei `/proc/loadavg`. Wenn die durchschnittliche Belastung (*average load*) unter 0,8 sinkt, dann wird der Job ausgeführt. Sie haben außerdem eine geringere Priorität als Hintergrundjobs. Im Gegensatz aber zu diesen, werden Sie nicht beim Ausloggen beendet, sondern laufen weiter. Vom Ende des Jobs werden Sie dann per eMail benachrichtigt. Wenn keine Zeit angegeben wird, beginnt der Job sofort.

## 13.2 Regelmäßiges Ausführen von Kommandos: Der cron-Dämon

Der Dämon `atd` erlaubt nur die zeitlich gesteuerte einmalige Ausführung eines Jobs. Allerdings ist es manchmal angebracht ein Job regelmäßig auszuführen. Dazu gehört z. B. die Datensicherung oder das Abgleichen von zwei Datenbanken. Für diese Aufgaben steht der Dämon `crond` zur Verfügung.

### 13.2.1 Der Dämon cron

Die Aufgabe des Ausführens der persönlichen `crontab`-Dateien und der regelmäßigen Systemjobs wird vom Dämon `cron` erledigt, dessen Programmdatei `/usr/sbin/cron` ist.

Das Verzeichnis für den Dämon ist `/var/spool/cron`. Hier schaut der Dämon `cron` im Verzeichnis `tabs` nach ob es persönliche Aufgaben gibt. Die sind in Dateien mit dem Benutzernamen des jeweiligen Users abgelegt. Im folgenden Beispiel hat der User `tapico` mit der Masterdatei `cj` eine Cron-Job angelegt.

```
defiant:/var/spool/cron # ls -l tabs
insgesamt 4
-rw----- 1 root users 214 2004-05-25 15:19 tapico
defiant:/var/spool/cron # cat tabs/tapico
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (cj installed on Tue May 25 15:19:44 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
*/5 * * * * uptime >> ~/uptime.log
```

Die gefundenen Cron-Tabelle (engl. *crontabs*) werden in den Speicher geladen. Daneben konsultiert der Dämon die Datei `/etc/crontab`, die bei SuSE 9.0 so aussieht.

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
#
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly
#
-*/15 * * * * root test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons >/dev/null 2>&1
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 4 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 4 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 4 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

Daneben sorgt das Skript `/usr/lib/cron/run-crons` dafür, daß die Skripte in den Verzeichnissen `/etc/cron.daily`, `/etc/cron.weekly` und `/etc/cron.monthly` regelmäßig ausgeführt werden. Bei anderen Distributionen erfolgt der Aufruf der Skripte durch das Programm `run-parts`, das durch den Daemon aufgerufen wird.

Die Dateien im Verzeichnis `/etc/cron.daily` werden als Erweiterung von `/etc/crontab` bezeichnet. Dies ermöglicht eine feinere Abstimmung von Aufgaben, als wie sie durch die oben erwähnten Cron-Verzeichnisse angeboten werden.

Damit ist also der Dämon `crond` für die Ausführung der regelmäßigen System- und Benutzerjobs zuständig. Er wird normalerweise gestartet, wenn Linux bootet und wacht jede Minute auf um zu überprüfen ob ein Job gestartet werden soll. Dabei überprüft er jedes mal die Cron-Tabelle. Sollte sich das Änderungsdatum einer Tabelle geändert haben, dann wird die Tabelle neu in den Speicher geladen. Es ist daher nicht notwendig den Cron-Dämon nach einer Änderung neu zu starten.

Bei der Ausführung werden alle Ausgaben des jeweiligen Jobs per eMail an den Besitzer der Cron-Tabelle gesendet. Es ist natürlich klar, daß Benutzer nur ihre eigenen Jobs und der Superuser alle Jobs bearbeiten kann. Für weitere Informationen über den Cron-Dämon konsultieren Sie bitte die Manual-Page **cron(8)**.

### 13.2.2 crontab

Für das Anlegen von Cron-Jobs durch eine Cron-Tabelle steht der Befehl `crontab` zur Verfügung.

```
crontab [OPTIONEN] [MASTERDATEI]
```

Feld	Werte
Minute	0 - 59
Stunde	0 - 23
Tag des Monats	1 - 31
Monat	1 - 12 oder Namen
Wochentag	0 - 7 oder Namen (0 oder 7 ist Sonntag)

Tabelle 13.1: Erlaubte Werte für einen Crontab-Job

**Optionen**

<b>-e</b>	Erstellt oder bearbeitet die <b>crontab</b> -Datei des Benutzers
<b>-l</b>	Zeigt die <b>crontab</b> -Datei des Benutzers an
<b>-r</b>	Löscht die <b>crontab</b> -Datei des Benutzers
<b>-u BENUTZER</b>	Legt fest, mit welcher <b>crontab</b> -Datei gearbeitet werden soll (nur Superuser)

Um **crontab** zu nutzen, müssen Sie zuerst eine Masterdatei im Format der oben schon gezeigten **/etc/crontab** anlegen. Den Namen können Sie frei wählen, er sollte aber nicht gerade mit **crontab** lauten. Dort tragen Sie für jeden Cron-Job eine Zeile mit sechs Feldern ein. Diese sechs Felder sind von links nach rechts: Minute, Stunde, Tag des Monats, Monat, Wochentag und Kommando. Die Werte für die Felder entnehmen Sie bitte der Tabelle 13.1.

Zeilen in den Cron-Dateien, die mit einem Schweinegatter '#' beginnen, sind Kommentarzeilen und werden ignoriert. Außerdem werden leere Zeilen und führende Leerzeichen auch ignoriert. Ein Asterisk '\*' in einem Feld repräsentiert jeden möglichen Wert (also vom ersten bis zum letzten) für das Feld.

Das **crontab**-Kommando speichert eine Kopie der Masterdatei unter dem Benutzernamen in dem Verzeichnis **/usr/lib/crontab/tabs**. Sie können auch direkt die Cron-Tabelle bearbeiten, in dem Sie den Befehl **crontab -e** benutzen. Dieser Befehl startet den Editor **vi** mit der aktuellen Cron-Tabelle und sorgt dann nach dem Beenden und Speichern für die Aktualisierung der Cron-Tabellen.

**Beispiele**

Um täglich festzuhalten, wer um 10 Uhr morgens eingeloggt ist, kann folgender Cronjob angelegt werden.

```
00 10 * * * who >> /var/log/benutzerliste.log
```

Bei einer Liste von Werten werden die Einzelwerte durch Kommata voneinander getrennt. Zeitbereich können auch mit einem Bindestrich '-' angegeben werden. So bedeutet '10-12' das Gleiche wie '10,11,12'. Ein Job, der jeden Dienstag und Donnerstag um 12 Uhr ausgeführt werden soll, kann dann so angegeben werden.

```
00 12 * * 2,4 myjob
```

Ein Job, der zu jeder Stunde von 6 bis 18 Uhr ausgeführt werden soll, sieht dann so aus.

```
00 6-18 * * * myjob
```

Soll ein Job innerhalb dieses Zeitraums nur jede zweite Stunde ausgeführt werden, dann kann man den Schrägstrich benutzen.

```
00 6-18/2 * * * myjob
```

Der folgende Befehl veranlaßt alle 5 Minuten einen Ping auf eine Internetadresse. Dies ist z. B. eine Möglichkeit eine Internetverbindung offen zu halten, wenn Sie sonst vom Provider nach 10 Minuten Inaktivität geschlossen werden würde.

```
*/5 * * * * ping -c 1 www.fibel.org &> /dev/null
```

Eine Besonderheit ist die Angabe des Tages. Hier stehen zwei Möglichkeiten zur Verfügung. Sie können den Tag des Monats und den Wochentag angeben. Werden beide Informationen angegeben, dann wird der Job

zu dem Zeitpunkt ausgeführt wenn mindestens eine von den Angaben zutrifft. Der folgende Befehl wird also nicht nur am Freitag, den 13. ausgeführt, sondern er wird an jedem 13. Tag des Monats und an jedem Freitag ausgeführt.

```
00 12 13 * 5 freitag_der_13.sh
```

### 13.2.3 Verwalten von cron-Jobs

Wie auch beim Befehl `at` können zu viele komplexe `cron`-Jobs das System erheblich beeinträchtigen. Deshalb existiert auch für diese Jobart Konfigurationsdateien. Dies sind die Dateien `/etc/cron.allow` und `/etc/cron.deny`. Ihre Funktionsweise ist identisch zu den Zugriffsdateien für `at`. Wenn die Datei `cron.allow` existiert, dürfen nur Benutzer, die in dieser Datei eingetragen sind, Cron-Jobs starten. Existiert hingegen nur die Datei `cron.deny` so dürfen alle Benutzer Cron-Jobs starten, außer denen, die in dieser Datei stehen. Sind beide Dateien nicht vorhanden, dann darf niemand außer `root` Cron-Jobs anlegen.

## 13.3 Logdateien

Eine wichtige Informationsquelle bei Fehlfunktionen des Betriebssystems sind die Logdateien. Systeminformationen werden in verschiedenen Logdateien mitgeschrieben. Diese Logdateien sind entweder speziell für eine Anwendung gedacht oder mehrere Anwendungen schreiben ihre Nachrichten in eine Datei.

### 13.3.1 Der Dämon syslogd

Der Dämon `syslogd` übernimmt die Aufgabe die Systemaktivitäten mitzuloggen. Er wird normalerweise durch einen Teil der `rc`-Konfigurationsdateien gestartet, wenn das System bootet.<sup>1</sup> Wenn der Dämon startet wird die Datei `/etc/syslog.conf` ausgelesen, in der die Optionen für die Ausführung des Dämons `syslogd` stehen.

`syslogd`

#### Optionen

<code>-f DATEI</code>	Gibt alternative Konfigurationsdatei an ( <i>file</i> )
<code>-h</code>	Veranlaßt <code>syslogd</code> dazu, Nachrichten von verbundenen Hosts weiterzuleiten ( <i>hosts</i> )
<code>-l HOST</code>	Die angegebenen HOSTs (Liste mit Doppelpunkt getrennt) werden nur mit einfachem und nicht mit vollem Hostnamen mitgeloggt ( <i>list of hosts</i> )
<code>-m INTERVALL</code>	Zeit zwischen zwei markierten Linien (Standard: 20 Minuten) ( <i>markintervall</i> )
<code>-r</code>	Erlaubt den Empfang von Netzwerknachrichten ( <i>remoteneWS</i> )

Der Dämon `syslogd` kann durch Signale gesteuert werden. Der Befehl `kill` (Abschnitt 12.9.7) kann mit den Signalen dazu verwendet werden, `syslogd` zu starten, stoppen, die Datei `/etc/syslog.conf` auf den neuesten Stand zu bringen u. s. w.

```
kill -SIGNAL $(cat /var/run/syslogd.pid)
```

Die Datei `/var/run/syslogd.pid` enthält die aktuelle PID des `syslogd`-Prozesses. Ein Übersicht über die Signal zeigt Tabelle 13.2.

### 13.3.2 /etc/syslog.conf

Die Datei `/etc/syslog.conf` ist die Konfigurationsdatei für den Daemon `syslogd`. Sie gibt an welche Informationen wohin geschrieben werden. Das folgende Beispiel ist der SuSE 9.0 entnommen.

```
# print most on tty10 and on the xconsole pipe
#
kern.warning;*err;authpriv.none          /dev/tty10
```

---

<sup>1</sup>Siehe dazu auch Kapitel 12 Bootvorgang.

Signal	Aktion
SIGHUP	Reinitialisiert syslogd durch Stoppen des Daemons, Neulesen der Datei /etc/syslog.conf und dem erneuten Starten des Daemons.
SIGTERM	Beendet syslogd.
SIGQUIT	Beendet syslogd.
SIGINT	Beendet syslogd.
SIGUSR1	Wechselt zum Debugging-Modus, wenn mit -d gestartet.
SIGALARM	Setzt eine Markierungslinie

Tabelle 13.2: Signale und ihre Auswirkungen auf syslogd

```
kern.warning;*.err;authpriv.none      |/dev/xconsole
*.emerg                                *

# enable this, if you want that root is informed
# immediately, e.g. of logins
#*.alert                                root

#
# all email-messages in one file
#
mail.*                                  -/var/log/mail
mail.info                              -/var/log/mail.info
mail.warning                           -/var/log/mail.warn
mail.err                               /var/log/mail.err

#
# all news-messages
#
# these files are rotated and examined by "news.daily"
news.crit                              -/var/log/news/news.crit
news.err                               -/var/log/news/news.err
news.notice                            -/var/log/news/news.notice
# enable this, if you want to keep all news messages
# in one file
#news.*                                -/var/log/news.all

#
# Warnings in one file
#
*.=warning;*.=err                      -/var/log/warn
*.crit                                 /var/log/warn

#
# save the rest in one file
#
*.*;mail.none;news.none                -/var/log/messages

#
# enable this, if you want to keep all messages
# in one file
#*. *                                   -/var/log/allmessages

#
# Some foreign boot scripts require local7
```

```
#
local0,local1.*          -/var/log/localmessages
local2,local3.*          -/var/log/localmessages
local4,local5.*          -/var/log/localmessages
local6,local7.*          -/var/log/localmessages
```

Dabei besteht jede Linie aus drei Angaben. Der **Nachrichtenquelle**, dem **Nachrichtentyp** und der **Logdatei**<sup>2</sup>.

Die Nachrichtenquelle ist eines der folgenden Schlüsselworte: `auth`, `authpriv`, `cron`, `daemon`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp` und `local0` bis `local7`.

Der Nachrichtentyp bzw. die Nachrichtenpriorität wird in aufsteigender Reihenfolge durch die Schlüsselworte `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert` und `emerg` ausgedrückt.

Ein Minuszeichen vor dem Namen der Logdatei sorgt dafür, daß die Nachrichten nicht sofort auf Platten geschrieben werden, sondern, wie sonst auch üblich, im RAM gespeichert und erst bei der nächsten routinemäßigen Synchronisation von RAM und Platte geschrieben werden.

Weitere Informationen finden Sie unter `man syslog.conf`.

### Beispiele

Der Stern steht für alle Nachrichtenquellen bzw. Nachrichtentypen. So loggt die folgende Zeile alle Nachrichten mit der Priorität `emerg` mit.

```
*.emerg                root
```

Das Gleichheitszeichen sorgt dafür, daß die genannte Nachrichtenpriorität exklusiv in einer Datei mitgeloggt wird.

```
*.=crit                /var/log/critical
```

Das Ausrufungszeichen `'!'` wird als Negationsoperator verwendet.

```
mail.*;mail.!=info     /var/log/maillog
```

Damit zum Beispiel überhaupt keine Mail-Nachrichten mitgeloggt werden, kann `mail.!` oder `mail.none` gesetzt werden.

### 13.3.3 Verwaltung der Logdateien

Jede Nachricht wird als eine Zeile in eine Logdatei geschrieben. Datum, Quelle und natürlich die Nachricht selber werden in den Logdateien geschrieben. Welche Logdateien Sie verwenden, hängt von der Konfiguration Ihres Systems ab. Es gibt aber ein paar wichtige Systemlogdateien, die überall eigentlich vorhanden sein sollten.

#### `/var/log/messages`

Dies ist die Hauptlogdatei des Systems. Hier laufen die meisten Meldung auf. Dies hängt aber stark davon ab, wieviele weitere Logdateien eingerichtet worden sind.

#### `/var/log/wtmp`

Hier werden die Login-Zeiten und die Login-Dauer der Benutzer festgehalten. Das Kommando `last` (Abschnitt 8.7.6) greift auf diese Logdatei zurück.

#### `/var/run/utmp`

Diese Logdatei enthält die Informationen über die aktuell eingeloggten Benutzer. Auf diese Datei greifen die Befehle `finger`, `w` und `who` zurück.

---

<sup>2</sup>Denken Sie daran: Auch Terminals sind im Prinzip für Linux nur Dateien.



Kommando	Aktion
<code>compress</code>	Benutzt <code>gzip</code> zur Kompression alter Dateien
<code>copytruncate</code>	Kopiert das Log und kürzt dann das alte Log
<code>create</code>	Benutzt die angegebenen Rechte fürs neue Log; sind keine Rechte angegeben, werden die Rechte des alten Logs übernommen.
<code>daily</code>	Logs rotieren täglich
<code>delaycompress</code>	Kompression bei der nächsten Rotation
<code>errors MAILTO</code>	Sendet Fehler an die angegebene Mailadresse
<code>ifempty</code>	Rotiert auch leere Logs
<code>include DATEI</code>	Fügt die angegebene Datei in die Konfigurationsdatei ein
<code>mail MAILTO</code>	Sendet Logs zur angegebenen Mailadresse, wenn sie gelöscht werden.
<code>monthly</code>	Logs rotieren monatlich
<code>nocompress</code>	Alte Logs werden nicht komprimiert
<code>nocopytruncate</code>	Logs werden nicht kopiert und gekürzt
<code>ncreate</code>	Gegenteil von <code>create</code>
<code>nodelaycompress</code>	Alte Logs werden sofort komprimiert
<code>noolddir</code>	Alte Logs werden nicht in ein anderes Verzeichnis verschoben
<code>notifempty</code>	Leere Logs werden nicht komprimiert
<code>olddir VERZEICHNIS</code>	Alte Logs werden in das angegebene Verzeichnis verschoben.
<code>postrotate</code>	Startet Skript nach der Rotation
<code>prerotate</code>	Startet Skript vor der Rotation
<code>rotate N</code>	Gibt die Anzahl der alten Logs an, die archiviert werden.
<code>size N</code>	Rotation, wenn das Log die angegebene Größe erreicht hat. (k für kB und M für MB)

Tabelle 13.3: Kommandos für die Konfigurationsdateien von `logrotate`.

#### `/var/log/lastlog`

In dieser Datei werden die Loginzeitpunkte der Benutzer gespeichert. Der Befehl `lastlog` verwendet diese Datei und ermöglicht so den Zeitpunkt des letzten Einloggens für jeden Benutzer zu sehen.

Vorsicht: Bei den Dateien `/var/log/wtmp`, `/var/run/utmp` und `/var/log/lastlog` handelt es sich um Binärdateien.

#### `/var/log/boot.msg`

Hier werden die gesamten Bootmeldungen gesammelt. Dies sind die Meldungen vom Kernel, von den Bootskripten und vom Starten der Dienste im gewählten Runlevel.

### 13.3.4 Rotation von Logdateien: `logrotate`

Die Natur von Logdateien ist es stetig zu wachsen und das über einen langen Zeitraum. Deswegen müssen diese Dateien wie Hecken regelmäßig zurückgeschnitten werden oder sie werden irgendwann das gesamte System überwuchern.

Für diesen Zweck gibt es den Befehl `logrotate`. Es sorgt dafür, daß ältere Daten aus den Logdateien entnommen und archiviert werden, alte Logdateien gelöscht und neue leere Logdateien erstellt werden.

```
logrotate [OPTIONEN] KONFIG_DATEI
```

Normalerweise ist die Konfigurationsdatei `/etc/logrotate.conf`, es können aber auch andere Dateien verwendet werden. Die Statusinformationen werden in der Datei `/var/lib/logrotate.status` gespeichert.

Die möglichen Kommandos entnehmen Sie bitte der Tabelle 13.3.

Eine Konfigurationsdatei kann globale und lokale Optionen enthalten. *Globale* Optionen gelten für alle Logs, während *lokale* Optionen für ein bestimmtes Log gedacht sind.

Der Eintrag

```
# Globale Optionen
weekly

# Lokale Optionen
# für wtmp
/var/log/wtmp {
    monthly
}
```

bewirkt, daß alle Logs wöchentlich rotieren. Dies ist eine *globale* Option. Die explizite Angabe des Lognamens beim zweiten Eintrag bewirkt, daß die globale Option durch die *lokale* Option überschrieben wird und die Logdatei `wtmp` nur monatlich rotiert.

## 13.4 Datensicherung

Die Datensicherung eines Systems ist eine der wichtigsten Aufgaben eines Administrators. In vielen Fällen kann es sogar die einzige Aufgabe sein.

Eine Datensicherung macht man nur aus einem Grund. Man will in der Lage sein ein zerstörtes System in kürzester Zeit wieder zum Laufen zu bringen. Eine Datensicherung zu besitzen kann einen davor bewahren Tage oder sogar Wochen an der Wiederherstellung des Systems und der Daten zu sitzen.

### 13.4.1 Vorüberlegungen

Mehrere Faktoren spielen bei der Planung von Datensicherungsstrategien eine Rolle.

**Kosten für die Ausfallzeit** Jedes System, sogar Linux, steht zu manchen Zeiten nicht den Benutzer zur Verfügung. Dies kann daran liegen, daß eine neue Programmversion aufgespielt wird, Fehler gepatcht werden, Hardware ausgetauscht wird oder die Systemleistung verbessert wird.

Steht das System dem Benutzer nicht zur Verfügung, so müssen auch die Kosten für die Ausfallzeit der Mitarbeiter mit in die Kosten für das gesamte Projekt einkalkuliert werden. Da dies sehr teuer werden kann, sollte man diesen Posten bei der Abwägung für die Durchführung der Aufgabe genau unter die Lupe nehmen.

**Kosten für die Sicherung** Es gibt eine große Bandbreite von Datensicherungsstrategien. Das reicht von der Speicherung der Daten auf einer Diskette bis zur Bildung eines riesigen Clusters für kritische Dienste die 24 Stunden am Tag und 7 Tage die Woche angeboten werden müssen. Wie Sie leicht sehen sind mit den verschiedenen Strategien auch ganz unterschiedliche Kosten verbunden. Hier muß abgewägt werden, was der Ausfall kosten würde und wieviel die Datensicherung kostet.

**Arbeitslast des System** Die Rolle des Systems ist ein wichtiger Faktor für die benötigte Verfügbarkeit. Während ein Webserver 24 Stunden am Tag aktiv sein muß, reicht es bei den meisten Bürosystem aus, daß sie während der Bürozeiten laufen. Mehrkosten für aufwendigere Datensicherungsstrategien machen sich meist dann bezahlt, wenn das System eine wichtige Rolle einnimmt bzw. einen hohe Verfügbarkeitsstufe besitzt.

### 13.4.2 Strategien

Es gibt eine Vielzahl von Strategien für die Verfügbarkeit und das Backup aus denen Sie die Richtige für sich auswählen können. Das kann z. B. sein:

- Cluster
- Standby- oder Ausfallserver

- Duplizierter oder Backup-Server
- Sicherung und Wiederherstellung

**Cluster** Ein Cluster besteht aus mindestens zwei Rechnern, die gemeinsam den Zugriff auf einen zentralen Datenpool ermöglichen. Jeder der Server kann verschiedene Applikationen ausführen. Wenn nun ein Mitglied des Clusters ausfällt, können die anderen Mitglieder seine Aufgaben übernehmen. Von dieser Maßnahme bemerkt der Benutzer nichts. Wenn Sie eine Verfügbarkeit von 24 Stunden am Tag benötigen, ist dies eine vernünftige Wahl. Neben den Hardwarekosten fallen vor allem hohe Kosten für das Management und Setup an.

**Standby- oder Ausfall-Server** Bei dieser Lösung ist neben dem eigentlichen Server ein identischer Server im Betrieb. Dieser steht im Gegensatz zum Cluster den Anwendern nicht zur Verfügung. Er ist allerdings ständig aktiv und gleicht seine Daten mit dem Hauptserver kontinuierlich ab. Fällt der Hauptserver nun aus, dann springt der Standby-Server ein um die Aufgaben zu erfüllen. Auch dieser Vorgang ist, da der Server die gleiche IP-Adresse und den gleichen Namen hat, für den Anwender nicht nachvollziehbar. Diese Lösung ist dank des einfacheren Managements günstiger als der Cluster, obwohl ein Teil der Hardware brach liegt. Trotzdem sind die Kosten noch so hoch, daß es sich nur für Server mit einer Verfügbarkeit von 24 Stunden am Tag lohnt.

**Duplizierter oder Backup-Server** Auch bei dieser Lösung existiert ein zweiter identischer Server. Dieser ist allerdings nicht im Betrieb. Fällt nun der Hauptserver aus, so muß der Administrator per Hand den Rechner auswechseln und die aktuellen Daten dem neuen Server zur Verfügung stellen. Das kann zum Beispiel durch den Tausch der Datenfestplatte oder durch den Anschluß an das Raid-Archiv geschehen. Bei Medienfehlern ist allerdings das Aufspielen des Backups nötig. Diese Lösung ist wiederum etwas günstiger, da kein kompliziertes Management des Systems nötig ist. Allerdings bedeutet dieses Konzept eine Ausfallzeit, falls der Hauptserver seinen Geist aufgibt.

**Sicherung und Wiederherstellung** Dies ist die häufigste und günstigste Methode der Sicherung. Selbst wenn Sie eine der oberen Lösungen gewählt haben, müssen Sie dieses Verfahren zusätzlich anwenden. Es basiert auf dem einfachen Prinzip, daß die Daten einfach an einem zusätzlichen Ort gespeichert werden. Bei diesem Konzept können sich die Kosten in einem weiten Bereich bewegen. Von der Diskette bis zu einem riesigen Storagetower ragt die Bandbreite.

### 13.4.3 Planung

Als ersten Schritt bei der Planung der Datensicherung sollten Sie sich über vier Fragen klar werden.

1. Was soll gesichert werden?
2. Wie oft soll gesichert werden?
3. Wieviel Zeit steht für die Sicherung zur Verfügung?
4. Wohin soll gesichert werden?

#### Was soll gesichert werden?

Ob eine Datei gesichert werden soll oder nicht läßt sich ganz einfach feststellen. Vergleichen Sie einfach die Zeit, die Sie für das Rekonstruieren der Daten benötigen, und die Zeit, die Sie für die Datensicherung benötigen. In den meisten Fällen brauchen Sie für das Rekonstruieren länger.

Als erstes sollten Sie Ihre Daten unterteilen in jene, die sich ändern, und jene, die sich nicht ändern. Als Regel kann man davon ausgehen, daß die System- und Applikationsdateien sich nicht so häufig ändern. Dies passiert ja nur, wenn das System und die Applikationen auf den neuesten Stand gebracht werden. Daher ist es eine Verschwendung von Ressourcen, wenn man diese Daten täglich sichert. Es ist aber eine gute Idee, nach jedem

Update ein Backup vom aktuellen System zu fahren. So ist eine schnelle Wiederherstellung des laufenden Systems in kürzester Zeit möglich. Eine Installation mit womöglich noch nachträglich zu installierenden Patches und Treiber dauert wesentlich länger, als eine Sicherung wieder auf das System zu spielen.

Für die Datendateien, die sich naturgemäß häufiger ändern, sollten Sie einen regelmäßigen Turnus für die Datensicherung einführen. Da die wichtigsten Dateien sich täglich ändern, sollte auch das Backup täglich durchgeführt werden.

Haben Sie sich entschieden verschiedene Datengruppen in verschiedenen Intervallen zu sichern, so sollten Sie diese Dateien auch an unterschiedlichen Orten, am besten in unterschiedlichen Partitionen, speichern. Dies erleichtert die Datensicherung wesentlich.

So sollte sich das Heimatverzeichnis auf einer separaten Partition befinden um die Sicherung der persönlichen Dateien der Benutzer wesentlich zu vereinfachen. Außerdem würde ein Schaden an der Systempartition keinen Schaden an den Daten verursachen und die Wiederherstellung des Systems wesentlich vereinfachen.

### **Wie oft soll gesichert werden?**

Nachdem Sie sich nun entschieden haben, was Sie sichern wollen, und das kann das gesamte System sein, müssen Sie noch festlegen, wie oft die Daten gesichert werden. Es ist nicht notwendig jede Datei jeden Tag zu sichern. Sie können es aber tun.

Warum sollten Sie Systemressourcen, Platz und Arbeitskraft für ein tägliches Update Ihrer Systemdateien verschwenden, wenn sich diese alle paar Wochen nur ändern. Eigentlich brauchen Sie nur eine Sicherung, wenn Sie Veränderung am System, wie Patches oder Aufspielen neuer Versionen, durchgeführt haben. Ein regelmäßiges Backup in größeren Abständen (z. B. monatlich) sollte aber durchgeführt werden, um sicher zu gehen, daß Sie eine funktionierende Kopie Ihres Systems besitzen.

Wenn es um die Häufigkeit eines Backups geht, dann fragen Sie sich doch einfach, welcher Schaden Ihrem Unternehmen entsteht, wenn die Daten verlorengehen, bevor eine Sicherung gefahren werden konnte. In den meisten Fällen reicht ein tägliches Backup aus. Bei sehr wichtigen sich häufigen ändernden Daten kann es sogar sinnvoll sein eine stündliche Sicherung zu fahren.

### **Wieviel Zeit steht für die Sicherung zur Verfügung?**

Auf jeden Fall sollten Sie berücksichtigen wieviel Zeit für ein Backup zur Verfügung steht. Da ein Backup eine nicht geringe Menge an Systemressourcen für sich beansprucht, sollte man die Sicherung in eine Zeit niedriger Auslastung verlagern. So sollte eine Sicherung für eine Verwaltung am späten Abend beginnen und vor dem nächsten Morgen abgeschlossen sein. Der schlimmste Fall tritt ein, wenn die Dauer der Sicherung länger dauert als das Sicherungsintervall lang ist. Hier muß man dann die Sicherungsstrategie wechseln.

Bei Systemen, die rund um die Uhr ausgelastet sind, muß man Fenster der Inaktivität nutzen um immer wieder einzelne Dateien zu sichern.

### **Wohin soll gesichert werden?**

Die Menge der Daten, die Zeit, die zum Sichern zur Verfügung steht, und die Dauer der Wiederherstellung bestimmen letztendlich die Auswahl des Speichermediums. Bandgeräte sind seit langer Zeit und auch heute noch die erste Wahl der Sicherung. Sie können große Mengen an Daten sicher speichern und sind dabei sehr kostengünstig. Gegen die ausgereiften Bandgeräte spricht die lange Dauer der Sicherung und die Schwierigkeit einzelne Dateien aus einem solchen Backup wieder extrahieren. Ein Bandgerät enthält schließlich nicht ein Dateisystem wie eine Festplatte. Für die Sicherung auf Band existieren viele Applikationen, die die Sicherung managen wie auch beim Wiederherstellen der Daten helfen. Dies ist dann fast so einfach wie ein Festplattzugriff, aber durch die sequentielle Speicherung der Daten kann es sehr lange dauern.

In der heutigen Zeit der billigen Festplatten, CD-Rs und optischen Laufwerken haben Sie ein große Zahl weiterer Medien für ihr Sicherung zur Verfügung.

Eine zweite Festplatte ist ein einfaches und schnelles Sicherungsmedium. Sollen aber mehrere Sicherungen über längere Zeit aufbewahrt werden, wird dieses Prinzip bald unerschwinglich teuer.

Die CD-R ist durch ihren günstigen Preis und ihre lange Haltbarkeit ein weitere gute Lösung. Allerdings macht ihr beschränkter Datenplatz bei größeren Backups Probleme. Daß sie nur einmal verwendbar ist, ist bei dem geringen Medienpreis zu verschmerzen. Die DVD mit ihrer größeren Kapazität und doch relativ kleinen Medienpreisen schließt hier eine Lücke im Angebot und reicht für viele Anwendungen aus.

Wollen Sie aber dennoch ein mehrfach beschreibbares Medium verwenden, dann sollten Sie sich mit den Magneto-Optischen-Medien beschäftigen. Sie stellen mehr Platz als CD-Rs zur Verfügung, der Zugriff auf die Daten ist flexibel wie bei einer Festplatte und sie sind haltbarer als Bänder.

#### 13.4.4 Backuptypen

Beim Backup werden vier Typen unterschieden, die sich darin unterscheiden was gesichert und wie es wiederhergestellt wird.

- Kopie
- Volles Backup
- Partielles Backup
- Inkrementelles oder differenzielles Backup

##### Kopie

Die einfachste und am häufigsten angewandte Methode der Datensicherung ist das Kopieren von Dateien. Obwohl es die am häufigsten angewandte Methode ist, ist es eigentlich kein richtiges Backup.

Wenn man eine Datei an einen anderen Ort kopiert, ist man in der Lage die Datei wiederherzustellen, falls sie beschädigt worden ist. Meisten benutzt man ein wiederbeschreibbares Medium wie Floppy oder Zip-Disk für diese Aufgabe. Hauptanwendungsgebiet ist die Sicherung von Konfigurationsdateien. So kann man bei Fehlern beim Konfigurieren durch einfaches Zurückkopieren der Dateien den alten Zustand des Systems wieder herstellen.

##### Volles Backup

Ein volles Backup umfaßt, wie der Name schon sagt, jede einzelne Datei auf dem System. Man hat also ein Abbild des jetzigen Zustands des Systems. Da alle Dateien gesichert sind der administrative Aufwand für Sicherung und Wiederherstellung minimal. Natürlich hat dieses Backup auch seine Nachteile. Da das gesamte System gesichert werden muß, fällt natürlich eine Menge an Daten an. Daher dauert die Sicherung sehr lange und es ist eventuell sogar nötig während der Sicherung die Medien tauschen zu müssen. Das verhindert natürlich eine automatische Ausführung.

##### Partielles Backup

Beim partiellen Backup hingegen werden nur Teile der Daten gespeichert. Dadurch kann die Häufigkeit der Sicherung auf die verschiedenen Dateiarnten angepaßt werden. Systemdateien werden nur nach Änderungen im System gespeichert, während die Datenpartition z. B. täglich gesichert wird. Es existiert also ein aktuelles Abbild des Systems, was aber zu verschiedenen Zeiten erstellt wurde. Daher werden dann bei der Sicherung weniger Daten übertragen. Die Sicherung ist damit schneller und man kommt mit weniger Medien aus, was dazu führt, daß der Vorgang auch automatisiert ausgeführt werden kann.

##### Inkrementelles oder differenzielles Backup

Das inkrementelle oder differentielle Backup reduziert noch weiter die zu sichernden Daten. Man beginnt mit einem vollen Backup (oder partiellem Backup) aller zu sichernden Daten. In den nächsten Sicherungen werden nur die Dateien gesichert, die sich seitdem geändert haben.

**Beispiel** Sie machen Freitag in der Nacht ein volles Backup. Damit sind alle Daten auf Ihrem Medium. Am Montag bis zum Donnerstag sichern Sie jetzt nur die Daten, die seit der letzten Sicherung sich geändert haben.

Kommt es nun an einem Mittwoch zu einem Systemausfall, dann müssen Sie zuerst das Backup vom Freitag – das ist das volle Backup – einspielen. Danach müssen Sie dann die Backups vom Montag und Dienstag einspielen um die Änderungen vom letzten vollen Backup wiederherzustellen.

### 13.4.5 Lagerung der Backups

Als Administrator wird man häufig gebeten eine Datei wiederherzustellen, die durch einen Fehler gelöscht worden ist. Allerdings merken die meisten Leute diesen Fehler nicht gleich sondern erst nach ein paar Tagen. Es ist daher wichtig die Backups auch einige Zeit aufzubewahren. In Abhängigkeit von den Informationen und wie schnell sie sich ändern, sollten Sie die Lagerhaltung planen.

**Beispiel** Sie wollen die Sicherung acht Wochen lang aufbewahren. Sie benötigen daher für die acht Wochen auch die Bänder. Nach den acht Wochen können Sie die ältesten Bänder wieder benutzen. Daneben wollen Sie das monatliche Backup für ein Jahr aufheben.

1. Jeden Freitag führen Sie ein volles Backup durch. Das Medium können Sie nach acht Wochen wieder benutzen. Sie benötigen dafür diese Sicherung also acht Bänder.
2. Von Montag bis Donnerstag führen Sie ein differentielles Backup durch. Auch diese Bänder müssen acht Wochen aufbewahrt werden, da eine Datei ja am Montag erstellt werden und dann am Donnerstag ausversehen gelöscht kann. Sie brauchen hier vier mal acht Bänder also 32 Stück.
3. An jedem ersten Tag im Monat machen sie ein volles Backup und bewahren die Bandkassette ein Jahr lang auf. Das macht also noch einmal zwölf Bänder.

Summa sumarum brauchen Sie also 52 Bänder um Ihre Backupstrategie durchführen zu können. Vergessen Sie aber auf keinen Fall die Bänder auch sorgfältig zu beschriften, sonst nützt Ihnen auch das vollständigste Backup nichts.

### Lagerplatz und Schutz

Was nützt Ihnen das schönste Backup, wenn Sie Ihre Bänder direkt neben den Server legen. Ein kleines Feuer im Serverraum und nicht nur Ihr Server ist hinüber, sondern auch Ihre Sicherung. Also aufgepaßt. Ihre Sicherung nützt Ihnen nur etwas, wenn Sie die Bänder sicher aufbewahren. Am besten schaffen Sie sich einen feuerfesten Tresor an, wenn Sie die Bänder in der Nähe des Servers aufbewahren. Sinnvoll ist es, einen Teil der Bänder auch außerhalb der Firma aufzubewahren. Denken Sie aber auch daran, daß jeder, der an die Bänder kommt, auch Ihre Daten lesen kann. Ein Tresor zu Hause oder ein Schließfach in Ihrer Bank sind sehr sinnvoll.

### Protokoll

Auch nützt Ihnen eine Sicherung nicht, wenn Sie vergessen ein Protokoll über Ihre Sicherung zu führen. Zumindest die Kassetten sollte durch eine Beschriftung Auskunft geben, was auf Ihnen gespeichert sind. Sonst müssen Sie Band für Band durchgehen, was auf ihm drauf ist. Oder Sie haben die Sicherung auf CD-R gebrannt. Inzwischen ist der Stapel an CD-Rs schon fünf mal umgekippt und alles ist durcheinander.

Machen Sie sich einfach einen Zettel und notieren Sie bei jedem Backup darauf:

- Wann wurde das Backup gefahren?
- Welcher Medientyp wurde verwendet?
- Welcher Bezeichnung hat das Medium?

- Was wurde gesichert?
- Wo wird das Medium aufbewahrt?

## 13.5 Werkzeuge für die Sicherung

Es gibt eine große Anzahl von Programmen für die Datensicherung. Einige davon sind große mächtige kommerzielle Lösungen, andere sind klein und gehören zur Linux-Grundausrüstung. Zwei davon sind **tar** und **cpio**.

### 13.5.1 tar

Das Programm **tar** (*tape archive*) wird dazu benutzt um mehrere Dateien zu einer Archivdatei zusammenzupacken. Dabei wird die Verzeichnisstruktur beibehalten. Obwohl **tar** entwickelt wurde um Daten auf Magnetbänder zu schreiben, kann ein solches **tar**-Archiv auf jedem beliebigen Medium gespeichert werden. Daneben kann **tar** die Archive bei der Erstellung auch gleichzeitig mit **gzip** (13.6.1) komprimieren.

**tar** [OPTIONEN] [TARARCHIV] [DATEILISTE]

In den Optionen bilden die Schalter eine besondere Gruppe. Es kann immer nur ein Schalter zur Zeit verwendet werden, während die anderen Optionen kombinierbar sind.

#### Schalter

<b>A</b>	Hängt das ARCHIV2 and das Ende von ARCHIV1
<b>c</b>	Legt ein neues Archiv an
<b>d</b>	Vergleicht den Inhalt des Archivs mit anderen Dateien
<b>r</b>	Fügt die neuen Dateien an das Ende eines bestehenden Archivs an
<b>t</b>	Zeigt eine Liste aller Dateien im Archiv
<b>u</b>	Fügt nur neue oder veränderte Dateien zum Archiv hinzu
<b>x</b>	Extrahiert Dateien aus dem Archiv

#### Optionen

<b>b <i>N</i></b>	Definiert die Blockgröße
<b>e</b>	Verhindert das Aufsplittern von Dateien über Archiv-Volumes
<b>f <i>DATEI</i></b>	Name des Archivs mit Pfad oder Gerätenamen
<b>j</b>	Benutzt <b>bzip2</b> zur Kompression
<b>m</b>	Übernimmt nicht die Änderungszeit der Datei aus dem Archiv
<b>n</b>	Das Gerät ist kein Bandgerät
<b>p</b>	Die Originalrechte werden übernommen
<b>v</b>	Zeigt die Liste der Dateien an, die bearbeitet werden
<b>w</b>	<b>tar</b> arbeitet interaktiv
<b>z</b>	Benutzt <b>gzip</b> zur Kompression
<b>F <i>Skript</i></b>	Am Ende jedes Mediums das angegeben Skript ausführen
<b>L</b>	Länges des Bandes in kByte
<b>M</b>	Das Archiv ist in mehrere Teile (Volumes) aufgeteilt.
<b>W</b>	Überprüft die Dateien, nachdem sie zum Archiv hinzugefügt worden sind.
<b>X <i>DATEI</i></b>	Die in <i>DATEI</i> aufgeführten Dateien werden nicht berücksichtigt.
<b>Z</b>	Benutzt <b>compress</b> zur Kompression
<b>--exclude <i>DATEI</i></b>	Die <i>DATEI</i> wird nicht berücksichtigt.

#### Anlegen eines tar-Archivs

Erstellung eines Archivs (**archiv.tar**) über alle Dateien im aktuellen Arbeitsverzeichnis und dessen Unterverzeichnissen.

```
tar cf archiv.tar .
```

Als Ziel kann neben einer Datei auch ein Gerät angegeben werden. Der folgenden Befehl speichert das komplette Dateisystem auf dem Gerät **/dev/tape**.

```
tar cf /dev/tape /
```

Dabei wird der Inhalt des Magnetbandes überschrieben. Meistens müssen Sie bei einem Magnetband die Blockgröße mit angeben. Die Blockgröße definiert dabei die Menge an Daten (in Einheiten zu 512 Byte), die zur gleichen Zeit geschrieben werden können. Die Angabe erfolgt durch die Option `b`.

```
tar cfb /dev/tape 20 /
```

Wie Sie sehen, wird der Wert für `b` nicht wie üblich direkt hinter die Option geschrieben, sondern erst werden die Optionen aufgeführt und dann in der Reihenfolge die Werte.

Wenn Sie das Archiv auf mehrere Geräte (z. B. Disketten) aufteilen wollen, müssen Sie die Größe der Archiv-Teile (Volumes) angeben. Dies erfolgt durch den Schalter `M`. So speichert der folgende Befehl den Inhalt des Verzeichnis `/home` auf das Diskettengerät `/dev/fd0` mit einer Größe von 1440 kB.

```
tar cfML /dev/fd0 1440 /home
```

Wenn Sie ein bestehendes Archiv nur auf den neuesten Stand bringen wollen, ohne alle Dateien erneut hineinzupacken, dann benutzen Sie den Schalter `u`.

```
tar uf archiv.tar .
```

Sie können Dateien auch an ein bestehendes Archiv anhängen.

```
tar rf archiv.tar neueDatei
```

Solche Archive können sehr groß werden, wenn Sie z. B. Logdateien speichern. Da es sich hierbei in den meisten Fällen um reine Textdateien handelt, können sehr gut gepackt werden. Mit der Option `z` wird `tar` angewiesen, das entstandene Archiv auch gleich mit `gzip` (Abschnitt 13.6.1) zu packen.

Um sich die Dateiliste des Archivs anzeigen zu lassen, kann man den folgenden Befehl verwenden.

```
tar tf archiv.tar
```

Nicht immer wollen Sie alle Dateien bzw. Unterverzeichnisse mit in das Archiv packen. Hier kommt die Option `-exclude` ins Spiel. Diese Option kann mehrfach vorkommen und gibt an, welche Dateien nicht mit ins Archiv sollen.

```
barclay@enterprise:~/test> tar cvf befehle.tar befehle/
befehle/
befehle/data
befehle/befehle
befehle/befehle.bck
befehle/data.bck
befehle/data.old
barclay@enterprise:~/test> tar cvf befehle.tar --exclude data.old befehle/
befehle/
befehle/data
befehle/befehle
befehle/befehle.bck
befehle/data.bck
barclay@enterprise:~/test> tar cvf befehle.tar --exclude data.old --exclude "*.bck" befehle/
befehle/
befehle/data
befehle/befehle
```

Haben Sie ein paar Dateien mehr, die nicht mit ins Archiv dürfen, dann können Sie die Namen auch in eine Datei schreiben und diese mit dem Schalter `-X` als Informationsquelle verwenden.



```
barclay@enterprise:~/test> cat noarchiv
data.old
*.bck
barclay@enterprise:~/test> tar cvf befehle.tar -X noarchiv befehle/
befehle/
befehle/data
befehle/befehle
```

### Entpacken eines tar-Archivs

Um eine Archiv zu entpacken (Schalter `x`) und dabei auch die Dateinamen zu sehen (Option `v`) können Sie den folgenden Befehl benutzen.

```
tar xvf archiv.tar
```

Vor dem Entpacken, sollten Sie feststellen welche Verzeichnispfad gespeichert wurden. Dies können Sie mit dem Befehl

```
tar tvf archiv.tar
```

feststellen. Die Option `v` erzeugt eine ausführlichere Anzeige als `t` alleine.

Schauen wir uns doch mal folgendes Beispiel an:

Aus dem Wurzelverzeichnis haben Sie die Daten im Verzeichnis `/home` mit folgendem Befehl gespeichert.

```
tar cf home.tar home/*
```

Daher wurde das Verzeichnis `home` in jedem Pfad mit eingefügt. Sind Sie allerdings ins Verzeichnis `/home` gewechselt und haben den Befehl als

```
tar cf home.tar .
```

ausgeführt, so ist das Verzeichnis `home` nicht mehr Bestandteil des Dateipfads.

Bei einem Backup sollten Sie die Dateien immer in dem gleichen Verzeichnis extrahieren in dem Sie auch das Archiv erzeugt hatten.

Um einzelne Dateien aus einem Archiv zu entpacken, können Sie zum einen den interaktiven Modus (Option `w`) von `tar` wählen. Bei dem folgenden Befehl werden Sie bei jeder Datei gefragt, ob Sie die Datei entpacken wollen.

```
tar xvwf archiv.tar
```

Allerdings hat diese Methode einen schwerwiegenden Nachteil. Auf diesem Wege eine Datei aus einem 10.000 Dateien-Archiv zu entpacken ist doch etwas mühselig. Hier bietet sich einfach an, die gewünschte Datei einfach anzugeben.

```
tar xf archiv.tar meineDatei.txt
```

Sie können auch den umgekehrten Weg nehmen und die Dateien angeben, die nicht extrahiert werden sollen. Dies geht analog zum Erstellen des Archivs.

```
tar xvf archiv.tar --exclude "*.bck"
```

### 13.5.2 Was ist ein Tarball?

Diese Frage stellen sich viele Benutzer, die zum ersten Mal ein Programm aus dem Netz laden und nicht von der Distribution installieren wollen. Auf den meisten Seiten findet sich neben Bezeichnungen wie RPM- und GNU Debian-Paketen auch die Möglichkeit das Programm als *Tarball* herunterzuladen.

Neben den schon fertig kompilierten Binärdateien erhalten Sie in der Open Source Gemeinde auch den Quellcode um ihr Programm selbst kompilieren und installieren zu können. Dabei wird der Quellcode nicht nur alleine vertrieben, sondern meist zusammen mit Konfigurationsdateien für die Installation und Dokumentationen zum Programm. Diese Dateien befinden sich meistens in einer Verzeichnisstruktur.

Solche Strukturen werden am besten in einem tar-Archiv gesichert. Damit das entstandene Archiv auch schnell heruntergeladen werden kann, wird es komprimiert. Dabei kommt meistens das Programm `gzip` (13.6.1) zum Einsatz. Aber auch `bzip2` (13.6.2) wird gerne verwendet, während das veraltete `compress` kaum mehr zum Einsatz kommt.

```
barclay@enterprise:~> tar cf superduper_2.1.tar superduper/2.1/
barclay@enterprise:~> gzip superduper_2.1.tar
```

Es ist dann eine Datei mit der Endung `tar.gz` entstanden. Ein solches komprimiertes Archiv wird als *Tarball* bezeichnet.

Beim Auspacken geht man den umgekehrten Weg. Erst entpacken und dann die Dateien aus dem Archiv extrahieren.

```
tapico@defiant:~> gunzip superduper_2.1.tar.gz
tapico@defiant:~> tar xf superduper_2.1.tar
```

Nun ist das Verzeichnis `/superduper/2.1` entstanden, daß alle Dateien enthält. Heute können Sie das Entpacken und Extrahieren in einem Schritt durchführen. Der Schalter `z` beim Befehl `tar` berücksichtigt die GZIP-Kompression. Für die Kompression mit `bzip2` muß der Schalter `j` verwendet werden und bei `compress` der Schalter `Z`.

```
tapico@defiant:~> tar zxf superduper_2.1.tar.gz
tapico@defiant:~> tar Zxf turbogeil_3.2.tar.Z
tapico@defiant:~> tar jxf megastark_0.7.tar.bz2
```

Denken Sie bitte daran, daß Sie beim Erstellen des Archivs die passenden Endungen für die Kompression selbst anhängen müssen.

```
barclay@enterprise:~> tar zcf superduper_2.1.tar.gz superduper/2.1/
barclay@enterprise:~> tar Zcf turbogeil_3.2.tar.Z turbogeil/3.2/
barclay@enterprise:~> tar jcf megastark_0.7.tar.bz2 megastark/0.7/
```

Für die Endung `.tar.gz` kann auch `.tgz` verwendet werden.

### 13.5.3 cpio

Ein weiteres Werkzeug für Arbeit mit Archiven ist `cpio` (*copy in and out*). Sie können damit nicht nur aus mehreren Dateien ein Archiv machen und die Dateien aus einem solchen Archiv wieder extrahieren, sondern auch komplette Verzeichnisstrukturen an einen anderen Ort kopieren.

`cpio` [OPTIONEN]

Wie auch bei `tar` gibt es auch hier Schalter und Optionen.

#### Schalter

- o           Legt ein neues Archiv an
- i           Extrahiert Dateien aus einem Archiv
- p           Kopiert komplette Verzeichnisstrukturen

**Optionen**

<b>-a</b>	Setzt die Zugriffszeit der Dateien nach dem Kopieren zurück.
<b>-d</b>	Erzeugt Verzeichnisse, wenn Sie benötigt werden ( <b>-i</b> und <b>-p</b> )
<b>-E DATEI</b>	Name einer Datei mit zusätzlichen Mustern für die zu entpackenden Dateien
<b>-F ARCHIV</b>	Name des Archivs, was entpackt werden soll ( <b>-i</b> )
<b>-m</b>	Erhält die Änderungszeit der Dateien ( <b>-i</b> )
<b>-r</b>	Fragt nach einem neuen Namen für die Datei vorm Kopieren, wird kein Namen angegeben, wird die Datei nicht kopiert.
<b>-t</b>	Zeigt mit <b>-i</b> die Dateiliste des Archivs an
<b>-u</b>	Überschreibt existierende Dateien
<b>-v</b>	Zeigt die Dateinamen bei der Bearbeitung an

Im Gegensatz zu **tar** arbeitet **cpio** nicht direkt mit den Dateien sondern übernimmt nur die Aufgabe der Archivierung. Um alle Dateien und Verzeichnisse im Verzeichns **/home** zu sichern, müssen die Dateinamen mit **find** übergeben werden und die Ausgabe in eine Datei umgeleitet werden.

```
find /home | cpio -o > home.cpio
```

Um nun zu prüfen welche Dateien in dem Verzeichnis sind, können Sie folgenden Befehl benutzen.

```
cpio -itF home.cpio
```

Um das Archiv zu entpacken, benutzen Sie den Befehl

```
cpio -iF home.cpio
```

Eine einzelne Datei läßt sich genau wie bei **tar** aus einem Archiv extrahieren.

```
cpio -iF home.cpio meineDatei.txt
```

## 13.6 Kompression

Selbst im Zeitalter der heutigen riesigen Festplatten<sup>3</sup> ist nie genug Platz vorhanden. Wichtig wird die Menge der Daten, wenn es um die Übertragung durchs Internet geht. Hier sind die Datenraten noch immer viel zu klein. Da aber die meisten Dateien viel weniger Informationen erhalten, als Daten in ihnen steckt, wurden Algorithmen entwickelt um die Daten auf diese Information zu komprimieren. Dieser Abschnitt behandelt nun die unter Linux gängigen Programme zur Kompression.

### 13.6.1 gzip, gunzip und zcat

Unter den Namen **gzip**, **gunzip** und **zcat** verbirgt sich jeweils das gleiche Programm.

```
ole@enterprise:~> ls -l /bin/{gzip,gunzip,zcat}
926671 /bin/gunzip 926671 /bin/gzip 926671 /bin/zcat
```

**gzip**

Das Tool **gzip** komprimiert den Inhalt einer Datei und erzeugt daraus eine neue Datei mit dem gleichen Namen und einem angehängten **.gz**. Im Gegensatz zu dem Programm **zip**, daß es auch unter verschiedenen Namen unter Windows gibt, kann **gzip** nur eine Datei packen und löscht standardmäßig auch die Originaldatei.

```
gzip [OPTIONEN] [DATEI]
```

---

<sup>3</sup>Meine Plattenkapazität hat sich von 1990 bis 2001 von 40 MB auf 40 GB, also den Faktor 1000, erhöht.

## Optionen

-c	Zeigt den Inhalt an ohne die komprimierte Datei zu löschen; zusammen mit -d ( <i>content</i> )
-d	Dekomprimiert die Datei ( <i>decompress</i> )
-n	Speichert nicht die originalen Zeitstempel und Dateinamen( <i>no name</i> )
-N	Speichert die originalen Zeitstempel (Standard) ( <i>!(no name)</i> )
-q	Unterdrückt Warnmeldungen ( <i>quiet</i> )
-r	Verarbeitet auch die Dateien in Unterverzeichnissen ( <i>recursive</i> )
-t	Test der Datenintegrität( <i>test</i> )
-v	Name und Kompressionsgrad werden ausgegeben( <i>verbose</i> )
-ZAHL	Gibt den Kompressionsgrad mit ZAHL an; Wert zwischen 1 (niedrig) und 9 (hoch).

## Komprimieren

Um die alte Version des Manuskripts zu komprimieren, deren Dateien im Verzeichnis lk-0.3 liegen, kann man wie folgt vorgehen. Beachten Sie, daß jede Datei einzeln komprimiert wurde.

```
tapico@defiant:~ > ls-l lk-0.3
insgesamt 428
-rw-r--r-- 1 tapico users 27494 Okt 17 21:32 lk-backup.tex
-rw-r--r-- 1 tapico users 48607 Okt 17 21:32 lk-bootvorgang.tex
-rw-r--r-- 1 tapico users 42321 Okt 17 21:32 lk-dateisystem.tex
-rw-r--r-- 1 tapico users 40262 Okt 17 21:32 lk-grundbefehle.tex
-rw-r--r-- 1 tapico users 25652 Okt 17 21:32 lk-hilfe.tex
-rw-r--r-- 1 tapico users 13915 Okt 17 21:32 lk-installation.tex
-rw-r--r-- 1 tapico users 13394 Okt 17 21:32 lk-listen.tex
-rw-r--r-- 1 tapico users 30716 Okt 17 21:32 lk-shell.tex
-rw-r--r-- 1 tapico users 39591 Okt 17 21:32 lk-textfilter.tex
-rw-r--r-- 1 tapico users 3427 Okt 17 21:32 lk.tex
tapico@defiant:~ > gzip -v lk-0.3/*
lk-0.3/lk-backup.tex: 65.0% -- replaced with lk-0.3/lk-backup.tex.gz
lk-0.3/lk-bootvorgang.tex: 64.8% -- replaced with lk-0.3/lk-bootvorgang.tex.gz
lk-0.3/lk-dateisystem.tex: 65.2% -- replaced with lk-0.3/lk-dateisystem.tex.gz
lk-0.3/lk-grundbefehle.tex: 67.1% -- replaced with lk-0.3/lk-grundbefehle.tex.gz
lk-0.3/lk-hilfe.tex: 63.7% -- replaced with lk-0.3/lk-hilfe.tex.gz
lk-0.3/lk-installation.tex: 59.6% -- replaced with lk-0.3/lk-installation.tex.gz
lk-0.3/lk-listen.tex: 69.7% -- replaced with lk-0.3/lk-listen.tex.gz
lk-0.3/lk-shell.tex: 65.7% -- replaced with lk-0.3/lk-shell.tex.gz
lk-0.3/lk-textfilter.tex: 69.2% -- replaced with lk-0.3/lk-textfilter.tex.gz
lk-0.3/lk.tex: 58.7% -- replaced with lk-0.3/lk.tex.gz
tapico@defiant:~ > ls-l lk-0.3
insgesamt 164
-rw-r--r-- 1 tapico users 9642 Okt 17 21:32 lk-backup.tex.gz
-rw-r--r-- 1 tapico users 17105 Okt 17 21:32 lk-bootvorgang.tex.gz
-rw-r--r-- 1 tapico users 14729 Okt 17 21:32 lk-dateisystem.tex.gz
-rw-r--r-- 1 tapico users 13261 Okt 17 21:32 lk-grundbefehle.tex.gz
-rw-r--r-- 1 tapico users 9331 Okt 17 21:32 lk-hilfe.tex.gz
-rw-r--r-- 1 tapico users 5648 Okt 17 21:32 lk-installation.tex.gz
-rw-r--r-- 1 tapico users 4087 Okt 17 21:32 lk-listen.tex.gz
-rw-r--r-- 1 tapico users 10556 Okt 17 21:32 lk-shell.tex.gz
-rw-r--r-- 1 tapico users 12230 Okt 17 21:32 lk-textfilter.tex.gz
-rw-r--r-- 1 tapico users 1438 Okt 17 21:32 lk.tex.gz
tapico@defiant:~ >
```

Anstatt eine Datei zu erstellen kann in beiden Richtung (Komprimieren und Dekomprimieren) das Ergebnis auf die Standardausgabe umgeleitet werden. Dabei bleibt die Ursprungsdatei im Gegensatz zur normalen Vorgehensweise erhalten.

```
tapico@defiant:~ > gzip -c lk.tex > lk.tex.gz
```

**gunzip**

Der Befehl `gunzip` sorgt dafür, daß eine mit `gzip` gepackte Datei wieder entpackt wird.

```
gunzip [OPTIONEN] [DATEILISTE]
```

Dabei ist `gunzip` gar kein eigenständiges Programm, sondern nur ein anderer Dateiname für `gzip`.

```
ole@enterprise:~> ls -i /bin/{gzip,gunzip}
926671 /bin/gunzip  926671 /bin/gzip
```

Das Programm `gzip` kann erkennen, unter welchem Namen es aufgerufen wurde. Deshalb ist `gunzip` nichts anderes als `gzip -d`. Aus diesem Grund gelten auch die gleichen Optionen wie bei `gzip`. So ist auch der folgende Befehl, so unsinnig er auch sein mag, vollkommen korrekt und erledigt die vorgesehene Aufgabe.

```
ole@enterprise:~> gunzip -d buch.tex.gz
```

**zcat**

Das Kommando `zcat` arbeitet wie der Befehl `cat` (Abschnitt 4.5.2). Im Gegensatz zu diesem gibt es den Inhalt mit `gzip` und `compress` gepackter Dateien auf den Bildschirm aus.

```
zcat [DATEILISTE]
```

Die gepackte Datei wird durch den Befehl nicht verändert. Auch hier ist `gzip` gemeint und es wird eigentlich der Befehl `gzip -cd` ausgeführt.

**13.6.2 Komprimieren mit bzip2**

Das Tool `bzip2` komprimiert Dateien auf der Basis des “*Burrows Wheeler block sorting text compression*”-Algorithmus. Dieses Verfahren ermöglicht bessere Komprimierungsraten als die der Programme `gzip` oder `compress`.

```
bzip2 [OPTIONEN] [DATEILISTE]
```

Obgleich sich `bzip2` und `gzip` sehr ähnlich sind, sind sie nicht identisch.

**Optionen**

<code>-c</code>	Zeigt den Inhalt an ohne die komprimierte Datei zu löschen; zusammen mit <code>-d</code> ( <i>content</i> )
<code>-d</code>	Dekomprimiert die Datei ( <i>decompress</i> )
<code>-f</code>	Überschreibt vorhandene Dateien ( <i>force</i> )
<code>-n</code>	Speichert nicht die originalen Zeitstempel und Dateinamen ( <i>no name</i> )
<code>-k</code>	Quelldatei wird nicht gelöscht ( <i>keep</i> )
<code>-N</code>	Speichert die originalen Zeitstempel (Standard) ( <i>!(no name)</i> )
<code>-q</code>	Unterdrückt Warnmeldungen ( <i>quiet</i> )
<code>-t</code>	Test der Datenintegrität ( <i>test</i> )
<code>-v</code>	Name und Kompressionsgrad werden ausgegeben ( <i>verbose</i> )
<code>-z</code>	Komprimiert die Datei, Standardeinstellung
<code>-Z AHL</code>	Gibt den Kompressionsgrad mit <code>AHL</code> an; Wert zwischen 1 (niedrig) und 9 (hoch).

Durch den besseren Algorithmus kann `bzip2` deutlich besser komprimieren als `gzip`. Das Beispiel einer großen DVI-Datei (ASCII-Code) zeigt, daß die Kompressionsrate von `bzip2` deutlich größer ist als die von `gzip`.

```
ole@enterprise:~/test> ls -l
insgesamt 2584
-rw-r--r--  1 ole      users      1692688 2004-06-27 07:51 lk.dvi
-rw-r--r--  1 ole      users      409181 2004-06-27 07:59 lk.dvi.bz2
-rw-r--r--  1 ole      users      528016 2004-06-27 07:59 lk.dvi.gz
```

Beim Komprimieren zerlegt **bzip2** die Datei in einzelne Blöcke. Je größer die Blöcke sind, desto besser ist der Komprimierungsgrad. Einen direkten Einfluß auf die Größe der Blöcke können Sie mit den Optionen **-1** bis **-9** ausüben. Diese geben direkt die Größe des Blocks in 100.000 Bytes an. Defaulteinstellung ist **-9**. Die Verwendung von größeren Blocks hat keinen signifikanten Einfluß mehr auf die Komprimierungsrate. Je kleiner die zu komprimierende Datei ist, desto geringer ist auch der Einfluß der Komprimierungseinstellung. Der Komprimierer **bzip2** benutzt eine 32-Bit-CRC-Checksumme um die Datenintegrität des Archivs überprüfen zu können. Die Überprüfung wird durch den Schalter **-t** aktiviert.

Für weitere Informationen, speziell zu den verwendeten Blockgrößen und Speichernutzung, konsultieren Sie die Manualpage **bzip2(1)**.

Wie auch bei **gzip** besitzt **bzip2** weitere Namen, die bestimmte Aktionen von **bzip2** aufrufen. Das Tool **bzip2recover** ist dagegen ein eigenständiges Programm.

```
ole@enterprise:~> ls -lG /usr/bin/{bzip2,bunzip2,bzcat,bzip2recover}
lrwxrwxrwx    1 root          5 2004-04-03 09:17 /usr/bin/bunzip2 -> bzip2
lrwxrwxrwx    1 root          5 2004-04-03 09:17 /usr/bin/bzcat -> bzip2
-rwxr-xr-x    1 root       26464 2003-09-23 17:23 /usr/bin/bzip2
-rwxr-xr-x    1 root       8352 2003-09-23 17:23 /usr/bin/bzip2recover
```

### **bunzip2**

Dieser Name zeigt auch auf das Tool **bzip2**. Unter diesem Namen aufgerufen, aktiviert es den Schalter **-d** zum dekomprimieren.

```
bunzip2 [OPTIONEN] [DATEILISTE]
```

Beim Dekomprimieren werden die Endungen wie folgt umgewandelt. Bei den Endungen **.bz2** und **bz** werden die Endungen vom Dateinamen entfernt. Die Endungen **.tbz2** und **.tbz** weisen auf komprimierte Tar-Archive hin und werden in **.tar** umgewandelt. Handelt es sich um keine der genannten Endungen, dann wird an die entkomprimierte Datei die Endung **.out** angehängt.

### **bzcat**

Auch dieser Name zeigt auf das Tool **bzip2**. Unter diesem Namen aufgerufen, aktiviert es die Schalter **-cd** zum dekomprimieren und gleichzeitiger Ausgabe auf der Standardausgabe.

```
bzcat [OPTIONEN] [DATEILISTE]
```

### **Datenrettung mit bzip2recover**

Sehr selten kommt es vor, daß eine komprimierte Datei z. B. durch ein Bitkipper fehlerhaft wird. In diesem Fall meldet **bunzip2** bzw. **bzip2** einen Fehler und verweigert das Dekomprimieren. Trotzdem ist die Datei nicht vollständig verloren. Da die Komprimierung in unabhängigen Blöcken erfolgt, können diese durch das Tool als einzelne Bzip2-Dateien ausgegeben werden.

```
bzip2recover DATEI
```

Aus der beschädigten Datei erstellt das Tool für jeden Block eine Datei mit dem Namen **rec00001file.bz2**, **rec00002file.bz2**, usw. Die einzelnen Dateien können nun mit **bzip2 -t** auf ihre Integrität überprüft und dann einzeln entpackt werden. Das Verfahren ist natürlich nur sinnvoll bei großen Dateien, die viele Blöcke enthalten. Dateien, die nur aus einem Block bestehen, können nicht wiederhergestellt werden. Wenn Sie die Sicherheit der komprimierten Dateien erhöhen wollen, sollten Sie bei der Komprimierung eine kleine Blockgröße einstellen.

```
ole@enterprise:~/test> bunzip2 lk.dvi.bz2
```

```
bunzip2: Data integrity error when decompressing.
Input file = lk.dvi.bz2, output file = lk.dvi
```

It is possible that the compressed file(s) have become corrupted.  
You can use the `-tvv` option to test integrity of such files.

You can use the `'bzip2recover'` program to attempt to recover data from undamaged sections of corrupted files.

```
bunzip2: Deleting output file lk.dvi, if it exists.
```

```
ole@enterprise:~/test> bunzip2 -tvv lk.dvi.bz2
```

```
lk.dvi.bz2:
[1: huff+mtf rt+rld]
[2: huff+mtf rt+rld]
...
[11: huff+mtf rt+rld]
[12: huff+mtf data integrity (CRC) error in data]
```

You can use the `'bzip2recover'` program to attempt to recover data from undamaged sections of corrupted files.

```
ole@enterprise:~/test> bzip2recover lk.dvi.bz2
```

```
bzip2recover 1.0.2: extracts blocks from damaged .bz2 files.
```

```
bzip2recover: searching for block boundaries ...
```

```
block 1 runs from 80 to 47833
```

```
block 2 runs from 47882 to 85493
```

```
...
```

```
block 17 runs from 3647046 to 3864608
```

```
block 18 runs from 3864657 to 3864696 (incomplete)
```

```
bzip2recover: splitting into blocks
```

```
writing block 1 to 'rec00001lk.dvi.bz2' ...
```

```
writing block 2 to 'rec00002lk.dvi.bz2' ...
```

```
...
```

```
writing block 16 to 'rec00016lk.dvi.bz2' ...
```

```
writing block 17 to 'rec00017lk.dvi.bz2' ...
```

```
bzip2recover: finished
```

```
ole@enterprise:~/test> ls
```

```
lk.dvi.bz2      rec00005lk.dvi.bz2  rec00010lk.dvi.bz2  rec00015lk.dvi.bz2
rec00001lk.dvi.bz2  rec00006lk.dvi.bz2  rec00011lk.dvi.bz2  rec00016lk.dvi.bz2
rec00002lk.dvi.bz2  rec00007lk.dvi.bz2  rec00012lk.dvi.bz2  rec00017lk.dvi.bz2
rec00003lk.dvi.bz2  rec00008lk.dvi.bz2  rec00013lk.dvi.bz2
rec00004lk.dvi.bz2  rec00009lk.dvi.bz2  rec00014lk.dvi.bz2
```

```
ole@enterprise:~/test> bzip2 -t rec*
```

```
bzip2: rec00012lk.dvi.bz2: data integrity (CRC) error in data
```

You can use the `'bzip2recover'` program to attempt to recover data from undamaged sections of corrupted files.

```
ole@enterprise:~/test> rm rec00012lk.dvi.bz2
```

```
ole@enterprise:~/test> bunzip2 -c rec*.bz2 > lk.dvi
```

### 13.6.3 compress und uncompress

Das Tool `compress` ist eines der ältesten Kompressionstools. Die komprimierte Datei endet auf `.Z`.

```
compress [DATEILISTE]
```

Da die Kompressionsraten von `compress` allerdings schlechter sind, als die der heutigen Tools, wird es kaum noch verwendet. Es gehört auch seit geraumer Zeit nicht mehr zum Standardumfang der SuSE-Distribution.

**uncompress**

Um mit `compress` komprimierte Dateien wieder zu entpacken, benutzt man das Tool `uncompress`.

`uncompress` [DATEILISTE]

.....  
*Notizen:*

.....



## XVI

Arbeitsblatt  
Administrative Aufgaben

## 16.1

Stellen Sie sicher, daß Sie auf keiner Konsole eingeloggt sind und auch keine Terminalemulationen auf der graphischen Oberfläche laufen. Führen Sie diese Aufgaben **nur** auf der Textkonsole aus und **nicht** in den Terminalemulationen des X-Window-Systems.

- 1 Loggen Sie sich an der Konsole 1 als *root* ein.
- 2 Legen Sie die Benutzer *ryker* und *troi* mit ihren Heimatverzeichnissen an. Geben Sie *ryker* das Kennwort *terra* und *troi* das Kennwort *betazed*.
- 3 Loggen Sie sich in Konsole 2 als *ryker* ein.
- 4 Loggen Sie sich in Konsole 3 als *troi* ein.
- 5 Auf welchem Terminal arbeiten Sie gerade?
- 6 Loggen Sie sich in Konsole 4 als *ryker* ein.
- 7 Stellen Sie fest, wer eingeloggt ist.
- 8 Stellen Sie fest, an welchem Terminal Sie gerade arbeiten.
- 9 Wechseln Sie zur Konsole 2.
- 10 Schreiben Sie mit dem `echo`-Befehl den Satz "Scott me up, Beamy". Leiten Sie die Ausgabe auf die Konsole 4 um.
- 11 Wechseln Sie zur Konsole 3 und wiederholen die Aufgabe 10. Schauen Sie auf der Konsole 4 nach dem Ergebnis.
- 12 Wechseln Sie zur Konsole 1 und wiederholen die Aufgabe 10. Schauen Sie auf der Konsole 4 nach dem Ergebnis.
- 13 Vergleichen Sie Ihr Ergebnis mit den Rechten auf die Konsole 4.
- 14 Wechseln Sie in die Konsole 3.
- 15 Erstellen Sie die Datei `.plan` mit dem Inhalt: "Heute Kaffekränzchen"
- 16 Lassen Sie sich ausführliche Informationen über den Benutzer *troi* anzeigen. Wurde der Inhalt der Datei `.plan` mit ausgegeben?
- 17 Wechseln Sie zur Konsole 2. Stellen Sie fest, wer Sie nun sind.
- 18 Lassen Sie sich ausführliche Informationen über den Benutzer *troi* anzeigen. Wurde der Inhalt der Datei `.plan` mit ausgegeben?
- 19 Sollte der Inhalt von `.plan` nicht ausgegeben worden sein, dann erläutern Sie das Grund und ändern das System so, daß der Fehler nicht mehr auftritt.
- 20 Sie haben gerade einen Anruf erhalten. Sie sollen den Gesprächspartner in 5 Minuten zurückrufen. Basteln Sie sich mit Bordmitteln einen Wecker, der Sie mit einem Signalton und einer Textnachricht daran erinnert.
- 21 Lassen Sie sich alle aktuellen at-Jobs anzeigen.
- 22 Wechseln Sie Ihre Benutzeridentität zu *root*.
- 23 Pünktlichkeit war noch nie Ihr Stärke. Sie haben nun zum wiederholten Male Ärger bekommen, weil Sie in der Mittagspause durchgearbeitet haben. Richten Sie nun einen Dienst ein, der alle Benutzer täglich um 12:15 Uhr daran erinnert, Mittagspause zu machen.

## XVI

Arbeitsblatt  
Administrative Aufgaben

16.2

- 24** Unter welcher Benutzeridentität arbeiten Sie gerade?
- 25** Unter welchem Namen hatten Sie sich auf dieser Konsole eingeloggt?
- 26** Lassen Sie sich den letzten Einloggzeitpunkt der Benutzer anzeigen. Welche Benutzer haben sich noch nie eingeloggt?
- 27** Welche Benutzer sind zur Zeit eingeloggt?
- 28** Lassen Sie sich den Inhalt der Datei `/var/run/utmp` möglichst effektiv anzeigen und vergleichen Sie den Inhalt mit dem Ergebnis der vorherigen Aufgabe.
- 29** Verbieten Sie dem Benutzer `troi` das Anlegen von `at-Jobs`. Testen Sie den Erfolg der Maßnahme.
- 30** Kehren Sie zur Konsole 1 zurück.
- 31** Verständigen Sie sich mit Ihrem Partner am Nachbarrechner.
- 32** Loggen Sie sich mit SSH auf dem Nachbarrechner als `ryker` ein.
- 33** Auf welchem Terminal arbeiten Sie gerade?
- 34** Wer ist gerade eingeloggt?
- 35** Wie lautet ihr Loginname?
- 36** Loggen Sie sich wieder aus.
- 37** Wo befinden Sie sich nun im System?

.....  
*Notizen:*

.....

## Kapitel 14

# Programminstallation und Kernelmanagement

Bei den heutigen Distributionen ist die Installation der mitgelieferten Programme oder besser gesagt Programmpakete durch die distributionsspezifischen Verwaltungswerkzeuge sehr einfach. In den meisten Fällen liegen die Programmpakete als RPM- (Redhat Package Manager) oder als Debian-Archiv vor. Wie kann der Benutzer nun Programme installieren, die nicht mit der Distribution mitgeliefert wurden?

Das hängt vor allem davon ab, wie das Programm vorliegt: Als ein Archiv mit dem Quellcode, als ein Archiv mit den Binärdateien mit eigenem Installationsprogramm oder als Binärdateien in Installationsarchiven vom Typ Debian oder RPM.

### 14.1 Kompilieren des Quellcodes

Eigentlich alle Open Source Programme sind als Quellcode verfügbar. Der große Vorteil dieser Methode ist die Flexibilität bei der Einpassung des Programms in bestehende Strukturen (Verzeichnisse, Bibliotheken), während die Weitergabe als Binärdateien oft bestimmte Bedingungen an die Form und Ausstattung des Systems stellt.

Die Installation auf diese Art und Weise ist komplizierter, da sie individuell vom Programmierer vorgegeben wird. Es hat sich aber eine bestimmte Vorgehensweise eingebürgert, nach der doch die meisten Programmierer vorgehen. Auf jeden Fall sollten Sie die Dateien `INSTALL` und `README` mit den Anweisungen des Programmierers lesen, **bevor** Sie mit der Installation beginnen.

#### 14.1.1 Entpacken eines Tarballs

In den meisten Fällen werden die Dateien für die Installation in einem Verzeichnisbaum vertrieben. Dies schließt den Quellcode (in der Sprache C), die Make-Datei und zusätzliche Dokumentationen ein. Um diesen Baum einfach vertreiben zu können, werden die Dateien und die Struktur in einem `tar`-Archiv (13.5.1) zusammengefaßt und mit `gzip` (13.6.1) zu einem Tarball (13.5.2) gepackt. Entpackt werden kann das Archiv auf verschiedene Arten:

```
barclay@enterprise:~> gzip -d quellcode.tar.gz
barclay@enterprise:~> tar xvf quellcode.tar
```

```
barclay@enterprise:~> gunzip quellcode.tar.gz
barclay@enterprise:~> tar xvf quellcode.tar
```

```
barclay@enterprise:~> gzip -cd quellcode.tar.gz | tar xv
```

```
barclay@enterprise:~> tar zxvf quellcode.tar.gz
```

### 14.1.2 Aufbau von C-Programmen

Die meisten Programme für Linux sind in der Programmiersprache C bzw. C++<sup>1</sup> geschrieben. Bei größeren Projekten ist es üblich den Quellcode der Programme in mehreren einzelnen Dateien zu erstellen und zu bearbeiten. Diese Dateien mit der Endung `.c` müssen dann einzeln kompiliert werden. Es entstehen sogenannte Objekt-Dateien mit der Endung `.o`. In einem zweiten Arbeitsgang werden dann diese Dateien zu dem eigentlichen Programm zusammengelinkt.

#### Beispiel

An diesem Beispiel für ein zugegeben ziemlich primitives C-Programm kann der Vorgang nachvollzogen werden. Sie werden in den nächsten Abschnitten noch einmal darauf stoßen.

Als Erstes erstellen Sie die C-Datei `start.c`, die die Methode `main()` enthält.

#### Listing 14.1 *start.c*<sup>2</sup>

```
1: int main() {
2:     printHallo();
3: }
```

In der Methode `main()` wird die Methode `printHallo()` aufgerufen. Diese ist weder eine Methode aus den Standard-C-Bibliotheken noch in der gleichen Datei vorhanden. Sie ist nämlich in der Datei `hallo.c` definiert worden.

#### Listing 14.2 *hallo.c*

```
1: #include <stdio.h>
2:
3: void printHallo() {
4:     printf("Hallo Fibelleser!\nEs hat geklappt\n");
5: }
6:
```

Die beiden Quellcode-Dateien müssen nun zuerst kompiliert werden. Dazu verwenden Sie den C-Kompiler `gcc`. Der Schalter `-c` bewirkt, daß die angegebene Datei zu einer Objekt-Datei kompiliert wird. Dann werden die beiden Objektdateien durch `gcc` mit dem Schalter `-o` zu der Programmdatei `gruss` zusammengelinkt. Dann noch schnell die Rechte für das neue Programm geändert und fertig ist der Gruß.

```
barclay@enterprise:~/cprog> gcc -c start.c
barclay@enterprise:~/cprog> gcc -c hallo.c
barclay@enterprise:~/cprog> gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog> chmod 755 gruss
barclay@enterprise:~/cprog> gruss
Hallo Fibelleser!
Es hat geklappt
barclay@enterprise:~/cprog>
```

### 14.1.3 configure

Die meisten größeren Quellcodepakete enthalten ein "configure"-Skript. Dieses Skript muß vom Benutzer weder bearbeitet oder mit Schaltern beim Starten konfiguriert werden. Seine Aufgabe ist es, die Systemkonfiguration auf Compiler, Bibliotheken und andere wichtige Elemente für die Kompilierung zu testen. Auf der Basis der ermittelten Informationen schreibt das Skript eine individuelle Installationskonfigurationsdatei (Make-Datei) passend für das System. Sollte `configure` Fehler wie fehlende Bibliotheken finden, so meldet das Skript das mit einer Fehlermeldung. Erfahrungsgemäß arbeitet `configure` in den meisten Fällen erfolgreich und Sie können mit dem eigentlichen Kompilieren beginnen.

<sup>1</sup>C++ ist die objektorientierte Weiterentwicklung von C.

<sup>2</sup>Die Zahlen am Anfang der Zeilen dienen nur der Orientierung und gehören nicht zum Programm.

#### 14.1.4 make

Das Programm `make` ist ein Hilfsmittel für das Kompilieren. Wenn der Quellcode aus mehreren Dateien besteht, muß jede dieser Dateien einzeln kompiliert und dann alle Dateien zu einem Programm verknüpft werden.

`make` erledigt das, in dem in einer Konfigurationsdatei Ziele und Abhängigkeiten definiert werden. Das Ziel einer solchen Datei ist die Kompilierung eines Programms. Dieses Programm hängt ab von den Objekt-Dateien, die wiederum abhängen von den jeweiligen Quellcode-Dateien. Der Vorteil bei der Verwendung von `make` ist unter anderem, daß die Änderungszeit von bereits bestehenden Objektdateien mit den jeweiligen Quellcode-Dateien verglichen werden. Normalerweise sind die Objektdateien immer jünger als die Quellcode-Dateien. In diesem Fall werden die Quellcode-Dateien nicht noch einmal kompiliert. Ist die Quellcode-Datei aber jünger als die Objektdatei, so wurde sie wahrscheinlich seit dem letzten Kompilierungsvorgang verändert und wird deshalb bei diesem Lauf erneut kompiliert. Dadurch sparen Sie enorm viel Zeit, da bei kleinen Änderungen keine komplette Neukompilierung notwendig ist.

Am Beispiel des Programms aus Abschnitt 14.1.2 möchte ich Ihnen den Aufbau einer solchen Make-Datei zeigen.

Das Ziel ist die Erstellung des Programms `gruss`. Dieses besteht aus den Objekt-Dateien `start.o` und `hallo.o`. Wir können also die Beziehung zwischen dem Programm und den Objekt-Dateien so bezeichnen:

```
gruss: start.o hallo.o
```

Die Objektdateien `start.o` und `hallo.o` hängen mit den Quellcode-Dateien `start.c` und `hallo.c` zusammen. In Kurzform ausgedrückt ergibt das:

```
start.o: start.c
hallo.o: start.c
```

Zu diesen Abhängigkeiten gehören die ausführenden Kommandos, die hinzugefügt werden müssen.

```
gcc -c start.c
gcc -c hallo.c
gcc -o gruss start.o hallo.o
```

Um die Make-Datei etwas unabhängiger vom Kompiler zu gestalten, kann sein Name durch eine Variable ersetzt werden.

```
comp=gcc
```

Wenn wir alles kombinieren, erhalten wir unsere Konfigurationsdatei `Makefile`. Achten Sie auf jeden Fall darauf, daß die Einrückung durch einen Tabulator erfolgt.

#### Listing 14.3 Makefile

```
1: comp = gcc
2:
3: gruss: start.o hallo.o
4:         $(comp) -o gruss start.o hallo.o
5:
6: start.o: start.c
7:         $(comp) -c start.c
8:
9: hallo.o: hallo.c
10:        $(comp) -c hallo.c
```

Nach der Erstellung der Datei `Makefile`, nach der `make` im Normalfall sucht, führen wir den Befehl `make` aus. Dabei sollten eventuell vorhandene Dateien `gruss`, `start.o` und `hallo.o` vorher entfernt worden sein.

```
barclay@enterprise:~/cprog> vi Makefile
barclay@enterprise:~/cprog> make gruss
gcc -c start.c
gcc -c hallo.c
gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog> make gruss
make: 'gruss' is up to date.
barclay@enterprise:~/cprog>
```

Beim erneuten Ausführen von `make` erkennt das Programm, daß keine Änderungen vorgenommen worden und meldet dies dem Benutzer. Werden allerdings Quellcode-Dateien bearbeitet, so ist ihr Änderungsdatum jünger als das der Objekt-Datei. In diesem Fall wird neu kompiliert. Allerdings werden nur die Abhängigkeiten abgearbeitet, die von den Änderungen betroffen waren.

```
barclay@enterprise:~/cprog> vi hallo.c
barclay@enterprise:~/cprog> ls -l
insgesamt 48
-rw-r--r--  1 barclay  users      146 Jan 22 22:09 Makefile
-rw-r--r--  1 barclay  users    13719 Jan 22 22:09 gruss
-rw-r--r--  1 barclay  users      94 Jan 22 22:18 hallo.c
-rw-r--r--  1 barclay  users     980 Jan 22 22:09 hallo.o
-rw-r--r--  1 barclay  users      31 Jan 22 21:15 start.c
-rw-r--r--  1 barclay  users     812 Jan 22 22:09 start.o
barclay@enterprise:~/cprog> make gruss
gcc -c hallo.c
gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog>
```

Dieses `Makefile` ist nur ein schwaches Beispiel für die Möglichkeiten, die in dem Programm `make` stecken. Diese Möglichkeiten sprengen aber den Rahmen des Levels 1 der LPI-Prüfung und dieses Kapitels.

### 14.1.5 Installation

Für die meisten Quellcode-Programme wurde ein bestimmter Ort im Verzeichnisbaum für die ausführbaren Dateien festgelegt. In den meisten Fällen landen die Dateien im Verzeichnis `/usr/local/bin`. Um die Installation zu vereinfachen gibt es in manchen `Makefile`-Dateien ein Ziel namens `install`, daß den Installationsort definiert. Durch die Ausführung des Befehls `make install` werden die entstandenen Dateien kopiert und mit den erforderlichen Attributen und Rechten versehen.

**Aber Vorsicht!** Das eingestellte Installationsverzeichnis im `Makefile` kann inkompatibel mit dem Verzeichnisbaum Ihrer Linux-Distribution sein. Auch der Wechsel des Standardinstallationsverzeichnis bei einem Versionswechsel kann zu massiven Problemen führen.

## 14.2 Verwaltung von gemeinsam genutzten Bibliotheken

Ein Programm besteht in der Regel aus vielen immer wieder benutzten Grundfunktionen. Damit der Programmierer diese nicht immer wieder neu schreiben bzw. den Code der Funktionen in seinen Code integrieren muß, werden solche Funktionen in sogenannten Bibliotheken zusammengefaßt. Die meisten Compiler sind mit solchen Grundbibliotheken ausgestattet. Wenn Sie sich unser Beispiel aus Abschnitt 14.1.2 ansehen, so enthält dies die Funktion `printf`. Diese Funktion ist Bestandteil der Bibliotheken des `gcc`-Compilers und wird nun statisch in das Programm eingebunden. Dies ist u.a. ein Grund dafür, daß die Objektdatei deutlich größer ist als die Quellcodedatei.

`printf` ist natürlich eine sehr einfache Funktion. Es gibt auch viel kompliziertere Funktionen, die z. B. die Arbeit mit den Fenstern des X-Window-Systems ermöglichen. Da diese Funktionen natürlich von jedem X-Window-Programm benötigt werden, ist es von Vorteil, wenn sich die Programme diese Bibliotheken

gemeinsam nutzen könnten. Der erste Schritt dahin ist, daß die Bibliotheksfunktion nicht statisch in das Programm eingebunden werden, sondern dynamisch verlinkt werden. D. h. das Programm verweist nur auf die Funktion in der Bibliothek. Wenn es nun eine Trennung von Programm und Bibliothek gibt, können jetzt mehrere Programme auf die gleichen Bibliotheken verweisen. Diese Bibliotheken werden als *Shared Libraries* bezeichnet. Für Windows-Jünger: Die Shared Libraries sind in ihrem Konzept mit den MS-Windows-DLLs (*Dynamic Link Libraries*) zu vergleichen.

Der Nachteil dieser dynamischen Bibliotheken liegt darin, daß diese natürlich installiert sein müssen, wenn ein Programm laufen soll. Um zu überprüfen welche Bibliotheken ein Programm braucht, wird das Tool `ldd` (14.2.2) verwendet. Die Übersicht über die installierten bzw. im System registrierten Dateien liefert der Befehl `ldconfig` (14.2.3), der auch für das Update der Liste verantwortlich ist.

Die dynamische Einbindung von Bibliotheken bietet sich zumeist bei großen Programmen an, die komplexe Standardfunktionen benötigen. Je kleiner und rudimentärer das Programm ist, desto besser ist eine statische Einbindung der Bibliotheksfunktionen.

Die Bibliotheken befinden sich meistens in den speziell dafür vorgesehenen Verzeichnissen. Typische Bibliotheksverzeichnisse sind `/lib`, `/usr/lib`, `/usr/local/lib`, `/usr/X11R6/lib` und `/opt/lib`. Die letzte Ziffer im Namen der Bibliothek ist die Hauptversionsnummer. Oft handelt es sich hierbei aber nur um einen Link, der auf die tatsächlich installierte Version zeigt.

```
tapico@defiant:~> ls -lG /lib/ld*
-rwxr-xr-x    1 root    94543 Sep 20 05:52 /lib/ld-2.2.4.so
lrwxrwxrwx    1 root      11 Nov 22 15:53 /lib/ld-linux.so.2 -> ld-2.2.4.so
```

Als Anwender werden sie mit den Bibliotheken nur dann konfrontiert, wenn Sie fehlen. In den meisten Fällen tritt ein solches Problem nur dann auf, wenn Sie ein Programm nachträglich installieren, das nicht von der Distribution stammt. So besitzen z. B. die Konfigurationsprogramme `yast` und `yast2` von SuSE die Möglichkeit andere benötigte Programmpakete wie z. B. Bibliotheken automatisch mit für die Installation auswählen zu lassen. Es kann vorkommen, daß neuere Programmversionen auch neuere Bibliotheken benötigen, wie es auch vorkommen kann, daß ältere Programme mit neueren Bibliotheken Probleme bekommen.

Als erste Hilfe bei Problemen bietet sich der Befehl `ldd` (14.2.2) an, um überhaupt festzustellen, welche Bibliotheken benötigt werden.

### 14.2.1 Runtime Linker: ld.so

Wie aber findet nun ein Programm die benötigten Bibliotheken. Für die Verbindung zwischen Programmen und Bibliotheken ist der Runtime Linker `ld.so` bzw. `ld-linux.so` zuständig. Damit er nicht ständig alle Verzeichnisse nach Bibliotheken durchsuchen muß, wertet er die Datei `/etc/ld.so.cache` aus. In dieser Datei befindet sich eine Liste der Bibliotheken mit allen relevanten Daten (Versionsnummer, Zugriffspfade etc.). Bei manchen Distributionen wird diese Datei automatisch bei jedem Start mit dem Programm `ldconfig` (14.2.3) aktualisiert. Werden neue Bibliotheken installiert, so muß der Befehl `ldconfig` manuell gestartet werden.

Eine Möglichkeit bestimmte Bibliotheken temporär zur Verfügung zu stellen, bietet die Umgebungsvariable `LD_LIBRARY_PATH`. Hier können Sie ähnlich wie in der Variable `PATH` Verzeichnisse angeben, in denen nach Bibliotheken gesucht wird. Die Variable kann auch von normalen Benutzer gesetzt werden, die keinen Zugriff auf den Cache der Bibliotheken – sprich `ld.so.config` und `ldconfig` – besitzen. Auch Sicherheitsgründen nutzen Programme, die unter SUID- und/oder SGID-Rechten gestartet werden, diese Umgebungsvariable nicht.

### 14.2.2 ldd

Der Befehl `ldd` zeigt für das angegebene Programm die benötigten Bibliotheken (*Shared Libraries*) an.

1dd [OTPIONEN] PROGRAMM|BIBLIOTHEK

## Optionen

-v Ausführliche Informationen

Kleine Programme, wie z. B. `cp` brauchen nur wenige Bibliotheken, während andere Programme, wie z. B. der X-Editor `nedit` da wesentlich anspruchsvoller sind.

```
tapico@defiant:~> ldd /bin/cp
        libc.so.6 => /lib/libc.so.6 (0x40023000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
tapico@defiant:~> ldd /usr/X11R6/bin/nedit
        libm.so.6 => /lib/libm.so.6 (0x40023000)
        libXm.so.2 => /usr/X11R6/lib/libXm.so.2 (0x40045000)
        libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4 (0x401f8000)
        libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x40207000)
        libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0x40215000)
        libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40263000)
        libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x4026e000)
        libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40286000)
        libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0x40369000)
        libXp.so.6 => /usr/X11R6/lib/libXp.so.6 (0x4037f000)
        libc.so.6 => /lib/libc.so.6 (0x40387000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

### 14.2.3 ldconfig

Um schneller seine Arbeit verrichten zu können, wertet der Runtime Linker `ld.so` die Binärdatei `ld.so.cache` aus, die die Liste der installierten Bibliotheken enthält. Um diese Datei zu aktualisieren wird das Programm `ldconfig` verwendet.

```
ldconfig [OPTIONEN] [VERZEICHNIS|BIBLIOTHEK]
```

#### Optionen

```
-p          Zeigt den Inhalt der aktuellen Cache-Datei an, anstatt sie zu erstellen (print)
-v          Ausführliche Informationen während der Ausführung (verbose)
```

Die Liste der zu durchsuchenden Verzeichnis steht in der Datei `/etc/ld.so.conf`. Die Verzeichnisse `/lib` und `/usr/lib` fehlen, da sie immer durchsucht werden. Hier ein Beispiel für die Datei aus SuSE 7.3:

```
tapico@defiant:~> cat /etc/ld.so.conf
/lib-aout
/usr/X11R6/lib/Xaw95
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linux-libc5/lib=libc5
/usr/i486-linux-libc6/lib=libc6
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib
/usr/openwin/lib
/opt/kde/lib
/opt/kde2/lib
/opt/gnome/lib
```

#### Beispiel

Eine neues Programm wurde installiert. Für dieses Programm wurden auch die passenden Bibliotheken installiert. Trotzdem kann das Programm nicht gestartet werden, da die Bibliotheken dem Runtime Linker nicht bekannt sind.

Zuerst wird geprüft, ob das Verzeichnis für die Bibliotheken auch in der Liste für `ldconfig` steht:



```

root@defiant:~> cat /etc/ld.so.conf
/lib-aout
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib

```

Sollte das Verzeichnis nicht in der Liste stehen, so muß es hinzugefügt werden. Danach wird das Programm `ldconfig` gestartet und der Cache des Runtime Linkers aktualisiert. Nun steht einem erfolgreichem Programmstart nichts mehr im Weg.

```

root@defiant:~> vi ld.so.conf
root@defiant:~> ldconfig

```

Um festzustellen, ob eine Bibliothek erfaßt wurde, kann die Option `-p` weiterhelfen.

```

enterprise:~ # ldconfig -p | grep XFree
XFree.so (libc6) => /usr/X11R6/lib/XFree.so

```

### 14.3 Softwareverwaltung mit RPM-Paketen

Der Vertrieb von Programmen in sogenannten Paketen erleichtert die Installation und Verwaltung der Programme erheblich. Dieser Abschnitt beschäftigt sich mit dem *RPM Package Manager* (*RPM*).

Die Installation von Programmen aus dem Quellcode hat einige deutliche Nachteile.

- Es ist schwer den Überblick über die installierte Programme und ihre Versionen zu behalten.
- Es ist praktisch unmöglich herauszufinden, welche Datei zu welchem Programmpaket gehört.
- Updates erfordern einen großen Aufwand, da bei einer Installation durch `make` (14.1.4) auch vorhandene Konfigurationsdateien überschrieben werden. Es existiert keine Beschreibung über die Funktion der Dateien.
- Daneben existieren auch keine Informationen, welche Pakete und Bibliotheken für die Ausführung des Programms benötigt werden. Es kann also Paket A installiert werden, ohne daß das benötigte Paket B auch installiert wurde.

RPM-Paket gibt es meistens im Doppelpack. In der einen Datei (Binärpaket) befinden sich die Binärdateien sowie die für die Installation erforderlichen Informationen und Konfigurationsdateien. Die zweite Datei (Quellpaket) enthält die für die Erstellung des Binärpakets benötigten Dateien mit dem Quellcode. Ein gute Anlaufstelle auf der Suche nach RPM-Paketen ist die Seite <http://www.rpmseek.de>, die eine Suchmaschine für RPM-Pakete zur Verfügung stellt.

Der Dateiname eines solchen Pakets enthält wichtige Informationen: `jdk-1.3.1-1.i386.rpm` bezeichnet das Paket `jdk` mit der Versionsnummer `1.3.1`, rpm-Release `1`. Die Release-Nummer ist meistens `1` oder wird ganz weggelassen. Nur wenn Fehler im RPM-Paket selber behoben, Änderungen durchgeführt oder Dokumentationsdateien hinzugefügt wurden, wird diese Nummer hochgezählt. Die Release-Nummer bezieht sich also auf das RPM-Paket selber, während sich die Versionsnummer auf das eigentliche Programm bezieht.

Die Kennung `i386` weist darauf hin, daß dieses Paket Binärdateien für PC-Prozessoren enthält. Steht dagegen ein `src` dort, dann handelt es sich hier um das Paket mit den Quellcodedateien. Die Informationen aus dem Dateinamen sind nur für den Benutzer bestimmt. Das RPM-Installationsprogramm bezieht alle seine Informationen aus dem RPM selbst. Es ist also für das Installationsprogramm egal, ob die Datei `jdk-1.3.1-1.i386.rpm` oder `hugo.txt` heißt.

Neben den Binärdateien enthält das Paket auch wichtige Informationen:

- Eine kurze Paketbeschreibung

- Weitere Informationen über die Versionsnummer
- Einordnung in die Gruppenhierarchie
- Abhängigkeit von anderen Paketen. Dies ist wichtig, falls das Paket bestimmte Programme (Interpreter, Bibliotheken) voraussetzt.

### Die Datenbank: /var/lib/rpm

Um die Daten verwalten zu können, benutzt der Manager für die RPM-Pakete eine Datenbank für alle installierten Binärpakete. Diese Datenbank besteht aus mehreren Dateien im Verzeichnis /var/lib/rpm. Hier eine alte Version der Datenbank für SuSE 7.3:

```
tapico@defiant:~> ls -l /var/lib/rpm
insgesamt 32908
-rw-r--r-- 1 root root 16384 Nov 22 16:27 conflictsindex.rpm
-rw-r--r-- 1 root root 5230592 Nov 22 16:27 fileindex.rpm
-rw-r--r-- 1 root root 16384 Nov 22 16:27 groupindex.rpm
-rw-r--r-- 1 root root 28672 Nov 22 16:27 nameindex.rpm
-rw-r--r-- 1 root root 28339656 Nov 22 16:27 packages.rpm
-rw-r--r-- 1 root root 159744 Nov 22 16:27 providesindex.rpm
-rw-r--r-- 1 root root 172032 Nov 22 16:27 requiredby.rpm
-rw-r--r-- 1 root root 16384 Nov 22 16:27 triggerindex.rpm
```

Bei den \*.rpm-Dateien handelt es sich nicht um RPM-Pakete, sondern um Binärdateien in einem rpm-spezifischen Format. Sie dürfen auch keinen Fall per Hand verändert werden. Auch die Deinstallation von Paketen darf nicht durch einfaches Löschen erfolgen, sondern muß immer durch einen RPM-Manager erledigt werden.

Inzwischen besitzen die RPM-Datenbankdateien keine Endung .rpm mehr. Das Verzeichnis sieht bei SuSE 9.0 wie folgt aus:

```
dozent@linux37:~> ls -l /var/lib/rpm/
insgesamt 36472
-rw-r--r-- 1 root root 5308416 2004-06-25 12:28 Basenames
-rw-r--r-- 1 root root 12288 2004-06-25 12:28 Conflictname
-rw-r--r-- 1 root root 974848 2004-06-25 12:28 Dirnames
-rw-r--r-- 1 root root 5320704 2004-06-25 12:28 Filemd5s
-rw-r--r-- 1 root root 24576 2004-06-25 12:28 Group
-rw-r--r-- 1 root root 24576 2004-06-25 12:28 Installtid
-rw-r--r-- 1 root root 24576 2004-06-25 12:28 Name
-rw-r--r-- 1 root root 26316800 2004-06-25 12:28 Packages
-rw-r--r-- 1 root root 327680 2004-06-25 12:28 Providename
-rw-r--r-- 1 root root 65536 2004-06-25 12:28 Provideversion
-rw-r--r-- 1 root root 12288 2004-06-25 12:28 Pubkeys
-rw-r--r-- 1 root root 217088 2004-06-25 12:28 Requirename
-rw-r--r-- 1 root root 122880 2004-06-25 12:28 Requireversion
-rw-r--r-- 1 root root 49152 2004-06-25 12:28 Shalheader
-rw-r--r-- 1 root root 49152 2004-06-25 12:28 Sigmd5
-rw-r--r-- 1 root root 12288 2004-06-25 12:28 Triggername
```

### Der Manager: rpm

Das Programm rpm ist der sehr mächtige und umfangreiche Paket-Manager für RPM-Pakete. Ob Installation, Upgrade, Informationen oder Tests, alles läuft über dieses Tool.

rpm [OPTIONEN]

Eine Vielzahl von Funktionen stehen Ihnen beim RPM-Manager zur Verfügung. Es ist etwas irritierend, daß manche Optionen mehrere Bedeutungen haben, je nachdem mit welchen anderen Optionen sie verwendet werden. Daher hier nur eine Übersicht der der Hauptfunktionen.

**Optionen**

<b>-i</b>   <b>--install</b>	Installation von Paketen
<b>-U</b>   <b>--upgrade</b>	Aktualisierung des Pakets auf eine aktuellere Version
<b>-F</b>   <b>--freshen</b>	Aktualisierung des Pakets nur wenn eine alte Version existiert.
<b>-e</b>   <b>--erase</b>	Entfernt installierte Pakete
<b>-q</b>	Liefert Informationen über Pakete
<b>--checksig</b>	Überprüft MD5 Checksumme und PGP-Signatur des Pakets
<b>--import</b>	Importiert einen öffentlichen Schlüssel in die Datenbank
<b>--rebuilddb</b>	Reorganisiert die RPM-Datenbank

*Für die LPI-Prüfung 101 sollten Sie sich genau mit dem RPM Package Manager auskennen, da ca 10 bis 15 Prozent der Fragen in der Prüfung sich mit diesem Thema beschäftigen.*

**14.3.1 Installation und Upgrade**

Der Package-Manager unterscheidet zwischen der Installation eines neuen Pakets und dem Einspielen einer neuen Version des Pakets (Upgrade).

Um ein neues Paket einzuspielen verwenden Sie folgende Befehlskombinationen.

```
rpm -i [OPTIONEN] PAKETLISTE
rpm --install [OPTIONEN] PAKETLISTE
```

Wenn das Paket auf anderen Paketen basiert, die nicht installiert sind, gibt rpm eine passende Fehlermeldung aus. Sie müssen dann die fehlende Pakete zuerst installieren. Hilfreich sind die Schalter **-v** und **-h**. Der Erste läßt rpm gesprächiger werden, während der Zweite einen schönen Fortschrittsbalken generiert.

```
root@defiant:~ # rpm -ivh jdk-1.3.1-fcs.i386.rpm
jdk #####
```

Wollen Sie eine neuere Version eines bereits installierten Pakets installieren, dann geben Sie einen der folgenden Befehle ein.

```
rpm -F [OPTIONEN] PAKETLISTE
rpm --freshen [OPTIONEN] PAKETLISTE
```

Dabei ist darauf zu achten, daß auch eine alte Version des Pakets installiert ist. Ist dies nicht der Fall, dann kommt es zu einer Fehlermeldung.

Wenn Sie nicht so genau wissen, ob es schon eine alte Version gibt oder Sie gar eine ganze Liste von Paketen installieren wollen, dann können Sie noch eine andere Variante wählen.

```
rpm -U [OPTIONEN] PAKETLISTE
rpm --upgrade [OPTIONEN] PAKETLISTE
```

Der Upgrade-Modus ermöglicht es Pakete zu installieren, egal ob es eine alte Version gibt oder nicht. Existiert eine ältere Paketversion, dann wird diese aktualisiert. Existiert keine frühere Paketversion, dann wird eine ganz normale Installation vorgenommen.

```
root@defiant:~ # rpm -U jdk-1.3.1-fcs.i386.rpm
```

Die Installation bzw. Upgrade eines Pakets erfolgt in mehreren Schritten:

- Die Abhängigkeiten des Pakets werden ermittelt und mit den bereits installierten Paketen verglichen.
- Es wird geprüft ob Konflikte mit anderen Paketen vorliegen.
- Vor der Installation der Dateien werden zur Vorbereitung vordefinierte Aufgaben abgearbeitet.

- Die Konfigurationsdateien werden angelegt und es wird entschieden, was mit vorhandenen Konfigurationsdateien gemacht wird.
- Die Dateien werden ausgepackt und an den vorgesehenen Ort im Dateisystem kopiert.
- Nach der Installation der Dateien werden zur Nachbereitung vordefinierte Aufgaben erledigt.
- Die erledigten Aufgaben und Aktionen werden in der Datenbank gespeichert.

Sie können nicht nur Pakete installieren, die sich auf der lokalen Platte befinden, sondern Sie können die Pakete auch aus dem Internet über die Protokolle *http* und *ftp* beziehen. In diesem Fall brauchen Sie nur die passende URL eingeben.

```
defiant:~ # rpm -ivh ftp://rpm.amov.de/xscorch-0.1.15-334.i586.rpm
Hole ftp://rpm.amov.de/xscorch-0.1.15-334.i586.rpm heraus
Preparing... ##### [100%]
    1:xscorch ##### [100%]
```

Bei der Installation kann es vorkommen, daß es Versionen der Konfigurationsdateien aus dem Paket schon auf dem System gibt. In diesem Fall darf *rpm* diese Konfigurationsdateien nicht einfach löschen, da dort wichtige Informationen enthalten sein können. Stehenlassen darf der Package Manager die Dateien aber auch nicht, da die neue Version eventuell mit der veralteten Konfigurationsdatei nichts anfangen kann. Der RPM Packager Manager löst diese Situation ganz einfach. Die alte Datei wird umbenannt. Dabei wird hinter dem Namen der Datei einfach die Endung *.rpmorig* angehängt. Dann wird die neue Konfigurationsdatei installiert, die auf das neue Paket eingestellt ist.

Natürlich gibt es für die Installation und das Upgrade von Paketen noch spezielle Optionen, die das Verhalten bei diesen Vorgängen steuern. Hier eine Auswahl der wichtigsten Optionen.

#### Optionen

<b>--excludepath <i>PFAD</i></b>	Installiert keine Dateien, die mit <i>PFAD</i> beginnen.
<b>--excludedocs</b>	Installiert keine als Dokumentation gekennzeichneten Dateien (Manual-Pages, Texinfo)
<b>--force</b>	Kombination aus <b>--oldpackage</b> , <b>--replacefiles</b> und <b>--replacepkgs</b> .
<b>-h   --hash</b>	Zeigt einen Fortschrittsbalken von 50 Schweinegattern (#) während der Installation an.
<b>--ignoresize</b>	Überprüft das Dateisystem vor der Installation nicht auf freien Speicherplatz.
<b>--ignorearch</b>	Erlaubt Installation und Upgrade auch dann, wenn die Hardwarearchitektur von Rechner und Binärpaket nicht zusammenpassen.
<b>--ignoreeos</b>	Erlaubt Installation und Upgrade auch dann, wenn das Betriebssystem von Rechner und Binärpaket nicht zusammenpassen.
<b>--includedocs</b>	Installiert die Dokumentation. Dies ist der Standard.
<b>--justdb</b>	Aktualisiert nur die Datenbank und nicht das Dateisystem.
<b>--nodeps</b>	Führt keine Abhängigkeitsprüfung vor der Installation durch.
<b>--nosuggest</b>	Es werden keine Pakete vorgeschlagen, die fehlende Abhängigkeiten enthalten.
<b>--noorder</b>	Die Pakete werden in der angegebenen Reihenfolge installiert. Normalerweise werden Sie so sortiert, daß die Abhängigkeiten erfüllt sind.
<b>--noscripts</b>	Die vor und nach der Installation auszuführenden Skripte werden nicht ausgeführt.
<b>--oldpackage</b>	Erlaubt es der Upgradefunktion ein neueres Paket durch ein älteres zu ersetzen.
<b>--percent</b>	Zeigt den Fortschritt der Installation in Prozent an.
<b>--prefix <i>PFAD</i></b>	Verschiebt
<b>--replacefiles</b>	Installiert das Paket, selbst wenn Dateien anderer installierter Pakete damit überschrieben werden.
<b>--replacepkgs</b>	Installiert die Pakete, selbst wenn einige von ihnen bereits auf dem System installiert sind.
<b>--test</b>	Testet nur und installiert nicht
<b>-v</b>	Ausführliche Informationen über die Aktion
<b>-vv</b>	Debugginginformationen

Schauen wir uns doch mal ein paar Optionen genauer an.

#### Debugginginformationen: -vv

Sehr ausführliche Informationen über die Installation erhalten Sie über den verdoppelten Sabbelschalter **-vv**. Diese Informationen sind sehr sehr ausführlich. Die Verwendung des Pager *less* ist angeraten.

```

defiant:~ # rpm -ivv xscorch-0.1.15-334.i586.rpm
D: ===== xscorch-0.1.15-334.i586.rpm
D: Expected size:      272832 = lead(96)+sigs(264)+pad(0)+data(272472)
D: Actual size:      272832
D: unshared posix mutexes found(38), adding DB_PRIVATE, using fcntl lock
D: opening db environment /var/lib/rpm/Packages create:cdb:mpool:private
D: opening db index      /var/lib/rpm/Packages rdonly mode=0x0
D: locked db index      /var/lib/rpm/Packages
D: opening db index      /var/lib/rpm/Pubkeys rdonly:nofsync mode=0x0
D: read h#              2 Header sanity check: OK
D: ===== DSA pubkey id a84edae89c800aca
D: xscorch-0.1.15-334.i586.rpm: V3 DSA signature: OK, key ID 9c800aca
D: added binary package [0]
D: found 0 source and 1 binary packages
D: ===== +++ xscorch-0.1.15-334 i586/linux 0x0
D: opening db index      /var/lib/rpm/Depends create:nofsync mode=0x0
D: Requires: rpmlib(PayloadFilesHavePrefix) <= 4.0-1      YES (rpmlib provides)
D: Requires: rpmlib(CompressedFileNames) <= 3.0.4-1      YES (rpmlib provides)
D: opening db index      /var/lib/rpm/Providename rdonly:nofsync mode=0x0
D: read h#              29 Header SHA1 digest: OK (08f8fed5fed1b5cebffab60d2a767e333d73c42)
D: Requires: ld-linux.so.2      YES (db provides)
...
D: closed db index      /var/lib/rpm/Depends
D: ===== recording tsort relations
D: ===== tsorting packages (order, #predecessors, #succesors, tree, depth)
D: 0 0 0 0 0 +xscorch-0.1.15-334
D: installing binary packages
...
D: mounted filesystems:
D: i dev bsize bavail iavail mount point
D: 0 0x0305 4096 498950 492677 /
D: 1 0x0002 1024 0 -1 /proc
D: 2 0x0008 1024 0 -1 /dev/pts
D: 3 0x0306 4096 605280 389680 /home
D: 4 0x0009 4096 31957 31956 /dev/shm
D: sanity checking 1 elements
D: opening db index      /var/lib/rpm/Name create:nofsync mode=0x42
D: computing 26 file fingerprints
D: opening db index      /var/lib/rpm/Basenames create:nofsync mode=0x42
...
Preparing packages for installation...
D: computing file dispositions
D: ===== +++ xscorch-0.1.15-334
D: Expected size:      272832 = lead(96)+sigs(264)+pad(0)+data(272472)
D: Actual size:      272832
D: install: xscorch-0.1.15-334 has 26 files, test = 0
xscorch-0.1.15-334
D: ===== Directories not explicitly included in package:
D: 0 /usr/bin/
D: 1 /usr/share/applications/
D: 2 /usr/share/doc/packages/
...
D: =====
D: fini 100755 1 ( 0, 0) 394473 /usr/bin/xscorch;40e92459
D: fini 100755 1 ( 0, 0) 205308 /usr/bin/xscorch-server;40e92459
D: fini 100644 1 ( 0, 0) 2915 /usr/share/applications/xscorch.desktop;40e92459
...
BZDIO: 101 reads, 826824 total bytes in 0.343 secs
D: +++ h# 673 Header SHA1 digest: OK (0330cbaf3b2ae7c8c35ae37468b9045bde8cad84)
D: adding "xscorch" to Name index.
D: adding 26 entries to Basenames index.
D: adding "Amusements/Games/Action/Other" to Group index.
...

```

### Testlauf: --test

Sie können die Installation auch erst einmal testen. Die Option `--test` führt zwar alle Tests durch, die normalerweise bei einer Installation erfolgen, die eigentliche Installation wird aber nicht ausgeführt. Die Fehlermeldungen sind identisch mit denen der normalen Installation.

```
defiant:~ # rpm -i --test gimp-2.0.1-5.i386.rpm
Warnung: gimp-2.0.1-5.i386.rpm: V3 DSA signature: NOKEY, key ID 30c9ecf8
Fehler: Failed dependencies:
    gimp-print >= 4.2.0 is needed by gimp-2.0.1-5
    glib2 >= 2.3.0 is needed by gimp-2.0.1-5
    gtk2 >= 2.3.0 is needed by gimp-2.0.1-5
    libc.so.6(GLIBC_2.3.4) is needed by gimp-2.0.1-5
    libcrlayeng.so.1 is needed by gimp-2.0.1-5
    libcroco.so.1 is needed by gimp-2.0.1-5
    libcrseleng.so.2 is needed by gimp-2.0.1-5
    libpopt.so.0 is needed by gimp-2.0.1-5
    pango >= 1.4.0 is needed by gimp-2.0.1-5
```

### Erneute Installation: --replacepkgs

Normalerweise wird die Installation eines Pakets verweigert, wenn es bereits installiert ist. Dies ist auch nicht weiter verwunderlich. Es kann aber der Fall eintreten, daß Dateien aus einem Paket beschädigt oder entfernt wurden. Der Manager verweigert auch in diesem Fall die Installation, da die Schäden nicht in der RPM-Datenbank verzeichnet sind. Sie können den Prüfmechanismus mit der Option **--replacepkgs** außer Gefecht setzen. Nun ist es möglich ein bereits installiertes Paket noch einmal zu installieren. Im folgenden Beispiel funktioniert das Programm *XScorch* nicht mehr richtig. Eine Überprüfung der Installation ergibt, daß ein komplettes Verzeichnis mit Dateien zu *XScorch* gelöscht wurde. Um die Dateien nachzuinstallieren muß der Schalter **--replacepkgs** verwendet werden, da ansonsten die Installation verweigert wird.

```
defiant:~ # rpm -V xscorch
missing    /usr/share/xscorch
missing    /usr/share/xscorch/accessories.def
missing    /usr/share/xscorch/copying.txt
missing    /usr/share/xscorch/images
missing    /usr/share/xscorch/images/xscorch-icon.xpm
missing    /usr/share/xscorch/images/xscorch-logo.xpm
missing    /usr/share/xscorch/profiles.def
missing    /usr/share/xscorch/weapons.def
defiant:~ # rpm -i xscorch-0.1.15-334.i586.rpm
    package xscorch-0.1.15-334 is already installed
defiant:~ # rpm -i --replacepkgs xscorch-0.1.15-334.i586.rpm
defiant:~ # rpm -V xscorch
defiant:~ #
```

### Überschreiben von Dateien: --replacefiles

Die Installation wird auch abgebrochen, wenn das RPM-Paket Dateien enthält, die schon auf dem System vorhanden sind und aus einem anderen Paket stammen. Das Überschreiben der Dateien kann mit der Option **--replacefiles** erzwungen werden. Dateien, die nicht in der RPM-Datenbank erfaßt sind und damit nicht aus einem RPM-Paket stammen, werden dagegen einfach gelöscht. Was aber passiert, wenn eine Konfigurationsdatei doppelt vorhanden ist. Wird Sie durch die Option **--replacefiles** einfach gelöscht. Natürlich nicht. Die Datei wird umbenannt und erhält die Endung **“.rpmsave“** oder die alte Datei bleibt bestehen und die neue Datei erhält die Endung **“.rpmnew“**.

```
defiant:~ # rpm -i vsftpd-1.2.1-69.i586.rpm
    file /etc/vsftpd.conf from install of vsftpd-1.2.1-69 conflicts with file from package vsftpd-1.2.0-48
    file /usr/sbin/vsftpd from install of vsftpd-1.2.1-69 conflicts with file from package vsftpd-1.2.0-48
    file /usr/share/man/man5/vsftpd.conf.5.gz from install of vsftpd-1.2.1-69 conflicts with file from ...
    file /usr/share/man/man8/vsftpd.8.gz from install of vsftpd-1.2.1-69 conflicts with file from packa...
defiant:~ # rpm -i --replacefiles vsftpd-1.2.1-69.i586.rpm
Warnung: /etc/vsftpd.conf created as /etc/vsftpd.conf.rpmnew
```

Es gibt allerdings ein Problem mit dem Ersetzen von Dateien. Nehmen wir einfach mal an, daß beide Pakete die gleiche Datei `blubb.conf` benötigen. Um beide Pakete installieren zu können, wird das zweite Paket mit der Option `--replacefiles` installiert und die erste Version der Datei wird als `blubb.conf.rpmsave` gespeichert. Noch lebt alles friedlich und einträchtig zusammen. Sollten Sie aber auf die Idee kommen das zweite Paket zu deinstallieren ist plötzlich auch die gemeinsame Datei `blubb.conf` mitsamt der Einstellungen weg. Nur wenn Sie das erste Paket deinstallieren, gibt es kein Problem. Bei der Installation des zweiten Pakets wurde die Datei `blubb.conf` für das erste Paket in der RPM-Datenbank als gelöscht vermerkt. Daher wird es auch beim Deinstallieren nicht gelöscht.

### Downgrade: `--oldpackage`

Manchmal hat man das Bedürfnis auf eine ältere Version eines Pakets zurückzugehen. Dies wird erfolgreich durch den RPM-Manager abgeblockt. Sollten Sie dennoch ein Upgrade bzw. Downgrade auf eine ältere Version machen wollen, sollten Sie den Schalter `--oldpackage` verwenden. Sie können auch richtig brutal vorgehen und die Funktionen von `--oldpackage`, `--replacefiles` und `--replacepkgs` zusammenfassen. Dafür ist die Option `--force` zuständig. Diese sollten Sie aber nur sehr behutsam einsetzen. Ist besser für Sie und Ihr System.

### Keine Abhängigkeiten: `--nodeps`

Ein Vorteil von RPM ist die Überprüfung von Abhängigkeiten vor der Installation. So kann verhindert werden, daß ein Paket installiert wird ohne das die notwendigen anderen Pakete auf dem System vorhanden sind. Das ist auch in 99 Prozent der Fälle gut so. Es kann aber auch vorkommen, daß ein Paket trotzdem installiert werden soll. Dann müssen Sie mit der Option `--nodeps` die Abhängigkeitsüberprüfung abschalten.

```
defiant:~ # rpm -i blubb-6.2-233.i586.rpm
Fehler: Failed dependencies:
        libblubb.so.1 is needed by blubb-6.2-233
defiant:~ # rpm -i --nodeps blubb-6.2-233.i586.rpm
defiant:~ #
```

### Documentationsdateien: `--includedocs` und `--excludedocs`

Normalerweise werden die Dokumentationen (Manual-Pages, TexInfo) bei der Installation mitinstalliert. Das entspricht der Option `--includedocs`. Sollen Sie nicht mitinstalliert werden, dann nehmen Sie die Option `--excludedocs`.

```
defiant:~ # rpm -i --excludedocs blubb-1.2-34.i586.rpm
```

### Verlegen von Dateien: `--prefix PFAD`

Bei einigen Paketen können Sie den Standort der Dateien selbst festlegen. Diese Dateien werden auch als *“Relocatable Packages”* bezeichnet. In diesen Paketen existiert ein Grundpfad für die Installation. Diesen können Sie mit der Option `--prefix` und einer Pfadangabe bei der Installation verändern.

Ein *Relocatable Package* können Sie durch die folgenden Befehle erkennen.

```
enterprise:~ # rpm -qp --queryformat "%{PREFIXES}\n" kernel-source-2.4.21-99.i586.rpm
/usr/src
enterprise:~ # rpm -qpi kernel-source-2.4.21-99.i586.rpm | grep Relocations
Name           : kernel-source           Relocations: /usr/src
```

Ein Paket, das sich nicht verschieben läßt, würde dagegen folgende Informationen liefern.

```
enterprise:~ # rpm -qp --queryformat "%{PREFIXES}\n" xscorch-0.1.15-334.i586.rpm
(none)
```

```
enterprise:~ # rpm -qpi xscorch-0.1.15-334.i586.rpm | grep Relocations
Name          : xscorch                      Relocations: (not relocateable)
```

Die Installation erfolgt dann ganz einfach unter Angabe der Option `--prefix`.

```
enterprise:~ # rpm -i --prefix /tmp/test kernel-source-2.4.21-99.i586.rpm
enterprise:~ # ls -l /tmp/test/
insgesamt 16
drwxr-xr-x    4 root    root          4096 2004-07-05 21:16 .
drwxrwxrwt   29 root    root          4096 2004-07-05 21:16 ..
lrwxrwxrwx    1 root    root              15 2004-07-05 21:16 linux -> linux-2.4.21-99
drwxr-xr-x   19 root    root          4096 2004-07-05 21:17 linux-2.4.21-99
drwxr-xr-x    8 root    root          4096 2004-07-05 21:16 linux-2.4.21-99-include
```

### Skriptausführung unterdrücken: `--noscripts`

Die RPM-Pakete können Skripte enthalten, die vor und/oder nach der Installation ausgeführt werden können. Dies ist notwendig, wenn das RPM-Paket vor oder nach der Installation die Ausführung bestimmter Programme benötigt. Der Einsatz der Option `--noscripts` verhindert die Ausführung dieser Skripte. Auch diese Option ist mit Vorsicht zu genießen, da die Skripte nicht umsonst in das Paket eingefügt worden sind.

```
enterprise:~ # rpm -qp --scripts apache-1.3.28-43.i586.rpm | grep scriptlet
postinstall scriptlet (through /bin/sh):
preuninstall scriptlet (through /bin/sh):
postuninstall scriptlet (through /bin/sh):
enterprise:~ # rpm -i --noscripts apache-1.3.28-43.i586.rpm
```

### Fortschritt: `--hash` und `--percent`

Manchmal dauert die Installation eines Pakets recht lange. In diesem Fall ist es von Vorteil, wenn man den Fortschritt der Installation irgendwie sehen kann. Dazu kann ein Fortschrittsbalken dienen. Mit der Option `--hash` bzw. `-h` können Sie sich einen solchen Balken anzeigen lassen.

```
enterprise:~ # rpm -i --hash kernel-source-2.4.21-99.i586.rpm
##### [100%]
##### ( 50%)
```

Eine weniger gebräuchliche Option ist `--percent`. Sie gibt in mehreren Schritten den aktuellen Fortschritt in Prozent aus. Dies ist weniger für das menschliche Auge als für Programme wie z. B. YaST gedacht, die den RPM Package Manager als Programmbestandteil benutzen und einen Fortschrittsbalken anzeigen lassen wollen.

```
enterprise:~ # rpm -i --percent kernel-source-2.4.21-99.i586.rpm
%% 0,000000
%% 0,000064
%% 0,000125
...
%% 99,997035
%% 99,999944
%% 100,000000
```

### Alternative Verzeichniswurzel: `--root PFAD`

Normalerweise ist die Verzeichniswurzel des Systems auch die Verzeichniswurzel von RPM. Es kann aber Situationen geben, z. B. beim Einsatz eines Rettungssystems, daß die Pakete in einen alternativen Verzeichnisbaum mit einer alternativen Verzeichniswurzel installiert werden müssen. Für diesen seltenen Fall können



Sie mit der Option `--root` eine neue Verzeichniswurzel angeben. Durch diese Angabe verschieben sich alle Verzeichnisse, sogar die RPM-Datenbanken und Konfigurationsdateien. Also passen Sie gut auf bei dieser Option.

```
enterprise:~ # rpm -i --root /mnt/rescue rpm-4.1.1-71.i586.rpm
```

**FTP-Einstellungen:** `--ftpport PORT --ftpproxy HOST`

Für den Zugriff über das Protokoll `ftp` auf ein RPM-Paket können Sie auch Einstellungen vornehmen. Sie können den Port für den Zugriff wählen (Option `--ftpport` und Sie können einen FTP-Proxy bestimmen (Option `--ftpproxy` über den der Zugriff erfolgt).

```
enterprise:~ # rpm -i --ftpport 4242 --ftpproxy proxy.amov.de \
> ftp://rpm.amov.de/blubb-42.42.42-42.i586.rpm
```

**Hardware und Betriebssystem:** `--ignorearch` und `--ignoreos`

In den RPM-Paketen wird festgehalten für welches Hardwareplattform (Architektur) und für welches Betriebssystem das Paket gedacht ist. Mit den jeweiligen Optionen für die Architektur (`--ignorearch`) und das Betriebssystem (`--ignoreos`) kann diese Überprüfung abgeschaltet werden. Von der Verwendung der Funktionen wird abgeraten, es sei denn, Sie wissen genau was Sie tun.

### 14.3.2 Deinstallation

Dieser Modus wird für das Entfernen von installierten Paketen verwendet. Dabei wird natürlich ein Paket nur dann entfernt, wenn kein anderes Paket auf dieses Paket angewiesen ist.

```
rpm -e [OPTIONEN] PAKETLISTE
rpm --erase [OPTIONEN] PAKETLISTE
```

Für die Deinstallation werden heute die Schalter `-e` und `--erase` verwendet. Die Schalter `-u` und `--uninstall` sind veraltet.

```
root@defiant:~ # rpm -uv jdk-1.3.1-fcs
Fehler: -u und --uninstall sind veraltet und funktionieren nicht mehr.
Fehler: Benutzen Sie stattdessen -e oder --erase
root@defiant:~ # rpm -ev jdk-1.3.1-fcs
```

Die Deinstallation eines Pakets erfolgt in mehreren Schritten:

- Die Datenbank wird befragt, ob andere Pakete von diesem Paket abhängen.
- Vor der Deinstallation der Dateien werden zur Vorbereitung vordefinierte Aufgaben abgearbeitet.
- Die Konfigurationsdateien werden überprüft und veränderte Dateien werden als Sicherungskopie abgespeichert.
- Die Datenbank liefert nun die Informationen welche Dateien jetzt gelöscht werden. Wenn Sie nicht zu einem anderen Paket gehören, werden Sie gelöscht.
- Nach dem Löschen der Dateien werden zur Nachbereitung vordefinierte Aufgaben erledigt.
- Alle Spuren des Pakets werden aus der RPM-Datenbank gelöscht.

Auch das Verhalten der Deinstallation kann durch Optionen gesteuert werden. Hier eine Auswahl von möglichen Optionen.

**Optionen**

<code>--nodeps</code>	Führt keine Abhängigkeitsprüfung vor der Installation durch.
<code>--noscripts</code>	Die vor und nach der Installation auszuführenden Skripte werden nicht ausgeführt.
<code>--test</code>	Testet nur und installiert nicht
<code>-v</code>	Ausführliche Informationen über die Aktion
<code>-vv</code>	Debugginginformationen

Die Optionen verhalten sich ähnlich wie bei der Installation. So verhindert `--noscripts` das Ausführen von Deinstallationskripten, `--test` testet nur die Deinstallation und `--nodeps` erlaubt es Pakete zu löschen, die eigentlich von anderen Paketen benötigt werden.

Sollten Sie Veränderungen an den Konfigurationsdateien durchgeführt haben, wollen Sie meistens diese Änderungen nicht verlieren. Aus diesem Grunde werden Konfigurationsdateien nicht gelöscht sondern – mit der Endung `“ .rpmsave ”` versehen – abgespeichert. Dies gilt nur für Konfigurationsdateien.

RPM erleichtert Ihnen die Deinstallation von Programmen. Das kann aber seine Tücken haben. So kann RPM sich ganz schnell selbst deinstallieren.

```
defiant:~ # rpm -e --nodeps rpm
```

Macht ja nichts, das kriegen wir schnell wieder hin. Einfach folgendes eingeben:

```
defiant:~ # rpm -i rpm-4.1.1-71.i586.rpm
-bash: rpm: command not found
```

Ups! Da war doch noch was ...

Viel Spaß macht es auch die Bash zu löschen.

```
defiant:~ # rpm -e --nodeps bash
```

Selbst wenn Sie die Bash selber nicht als Shell nutzen. Viele Skripte geben zur Ausführung die Bourne-Shell (`/bin/sh`) als Interpreter an. Allerdings ist `/bin/sh` nur ein symbolischer Link auf die Bash.

```
defiant:~ # ls -l /bin/sh
lrwxrwxrwx 1 root root 4 May 4 2001 /bin/sh -> bash
```

Also lassen Sie auch davon die Finger. Um sich eine Übersicht zu verschaffen, welche Programme von der Bash abhängen, probieren Sie doch mal folgendes Kommando aus.

```
defiant:~ # rpm -e --test bash
```

Aber denken Sie daran: Vergessen Sie nicht die Option `--test` zu setzen, sonst ...

**14.3.3 Informationen**

Installierte und nicht installierte Pakete können im Informationsmodus (*Query-Mode*) mit dem Befehl `rpm -q` befragt werden. Im einfachsten Fall liefert der Befehl den vollen Paketnamen.

```
defiant:~ # rpm -q xscorch
xscorch-0.1.15-334
```

Auch hier gibt es eine Reihe von Schaltern, die das Verhalten des RPM Package Managers steuern.

**Optionen**

<b>-p</b> <i>DATEI</i>	Bezieht sich auf eine RPM-Datei, anstatt auf ein installiertes Paket
<b>-f</b> <i>DATEI</i>	Pakete, die die angegebene <i>DATEI</i> enthalten.
<b>-a</b>	Alle Pakete
<b>-g</b> <i>GRUPPE</i>	Alle Pakete, die zu einer bestimmten Gruppe gehören
<b>--whatprovides</b> <i>X</i>	Alle Pakete, die eine bestimmte Ressource <i>X</i> enthalten
<b>--whatrequires</b> <i>X</i>	Alle Pakete, die eine bestimmte Ressource <i>X</i> benötigen
<b>-i</b>	Ausführliche Paketinformationen
<b>-l</b>	Liste der Dateien im Paket
<b>-c</b>	Liste der Konfigurationsdateien im Paket
<b>-d</b>	Liste der Dokumentationsdateien im Paket
<b>-s</b>	Liste der Dateien im Paket mit Status
<b>--scripts</b>	Zeigt die im Paket enthaltenden RPM-Skripte an
<b>--queryformat</b>   <b>--qf</b>	Definiert das Ausgabeformat der Daten
<b>--dump</b>	Zeigt alle verifizierbaren Informationen über die Dateien im Paket an
<b>--provides</b>	Zeigt die Ressourcen an, die das Paket bietet
<b>-R</b>   <b>--requires</b>	Zeigt die Ressourcen an, die das Paket benötigt

Schauen wir uns auch hier mal die Optionen genauer an.

**Auswahl der Pakete**

**Befragung einer Paketdatei:** **-p** Normalerweise bezieht RPM die Angabe eines Paketnamen auf die installierten Pakete. Wenn Informationen über eine RPM-Datei anstatt über ein installiertes Paket gefragt sind, kommt der Schalter **-p** ins Spiel.

```
defiant:~ # rpm -qpi jdk-1.3.1.i386.rpm
Name           : jdk                               Relocations: (not relocateable)
Version        : 1.3.1                             Vendor: Sun Microsystems
Release        : fcs                               Build Date: Son 06 Mai 2001 12:46:01 CEST
Install date: (not installed)                      Build Host: lady-linux
Group          : Development/Tools                 Source RPM: jdk-1.3.1-fcs.src.rpm
Size           : 59866441                          License: 1994-2001 Sun Microsystems, Inc.
Packager       : Java Software <j2se-comments@java.sun.com>
URL            : http://java.sun.com/linux
Summary       : Java(TM) 2 Software Development Kit, Standard Edition
Description  :
The Java 2 SDK, Standard Edition includes the Java Virtual Machine,
core class libraries and tools used by programmers to develop Java software
applets and applications. The SDK also provides the foundation for IDE
(Integrated Development Environment) tools such as Sun's Forte for Java,
Community Edition, the Java(TM) 2 Platform, Enterprise Edition (J2EE),
Java-based application servers and more.
The Java 2 Software Development Kit, SDK, is a development
environment for building applications, applets, and components
that can be deployed on the Java platform.
The Java 2 SDK software includes tools useful for developing and
testing programs written in the Java programming language and
running on the Java platform. These tools are designed to be used
from the command line. Except for applet viewer, these tools do not
provide a graphical user interface.
```

**Welches Paket enthält die Datei:** **-f** Sie können sogar herausfinden, aus welchem Paket eine Datei stammt. Dann kommt die Option **-f** ins Spiel. Aber nicht jede Datei wurde aus einem Paket extrahiert.

```
defiant:~ # rpm -qf /bin/cp
fileutils-4.1-51
defiant:~ # rpm -qf /etc/fstab
die Datei »/etc/fstab« gehört zu keinem Paket
```

**Alle Pakete anzeigen:** `-a` Und natürlich können Sie sich alle installierten Pakete anzeigen lassen. Dafür sorgt die Option `-a`. Da es sich um einige hundert Pakete handelt, empfiehlt es sich die Ausgabe zu filtern.

```
defiant:~ # rpm -qa | sort | less
defiant:~ # rpm -qa | wc -l
581
defiant:~ # rpm -qa | grep netscape
netscape-plugins-4.78-14
netscape-4.78-34
```

**Auswahl nach Gruppen:** `-g` Jedes Paket wird zu einer thematischen Gruppe zugeordnet. Alle Pakete einer solchen Gruppe können Sie sich über den Schalter `-g` ausgeben lassen.

```
defiant:~ # rpm -qg System/Shells
sash-3.6-105
ash-0.2-798
bash-2.05b-207
zsh-4.1.1-42
tcsh-6.12.00-285
pdksh-5.2.14-681
```

**Auswahl nach enthaltenen Ressourcen:** `--whatprovides` Sie können sich die Pakete auch in Abhängigkeit von den enthaltenen Ressourcen anzeigen lassen. Hierfür ist die Option `--whatprovides` zuständig.

```
defiant:~ # rpm -q --whatprovides bash
bash-2.05b-207
```

**Auswahl nach benötigten Ressourcen:** `--whatrequires` Auch die gegenteilige Operation ist möglich. Sie können sich auch die Pakete in Abhängigkeit von den benötigten Ressourcen anzeigen lassen. Hierfür ist die Option `--whatrequires` zuständig.

```
defiant:~ # rpm -q --whatrequires less
man-2.4.1-60
defiant:~ # rpm -q --whatrequires rpm
yast2-packagemanager-2.8.36-1
defiant:~ # rpm -q --whatrequires libc.so.6 | wc -l
471
```

**Ausführliche Informationen:** `-i`

Ausführliche Informationen über ein installiertes Paket bekommen Sie mit der Schalterkombination `-qi`.

```
root@defiant:~ # rpm -qi te_latex-1.0.7-285
Name      : te_latex                      Relocations: (not relocateable)
Version   : 1.0.7                        Vendor: SuSE GmbH, Nuernberg, Germany
Release   : 285                          Build Date: Sun 23 Sep 2001 22:43:43 CEST
Install date: Don 22 Nov 2001 16:21:01 CET Build Host: levi.suse.de
Group     : Applications/Publishing/TeX  Source RPM: tetex-1.0.7-285.src.rpm
Size      : 25835581                     License: 1999 - not specified
Packager  : feedback@suse.de
Summary   : All about LaTeX
Description:
This package provides LaTeX, to be exact LaTeX2e, and a huge amount
software around LaTeX. This package is required by the most
(La)TeX documents.
```

## Authors:

-----

```

    Leslie Lamport <lamport@pa.dec.com>
    Johannes Braams
    David Carlisle
    Alan Jeffrey
    Frank Mittelbach <frank.mittelbach@uni-mainz.de>
    Chris Rowley
    Rainer Schöpf
    Members of the LaTeX3 project

```

SuSE series: tex

**Im Paket enthaltene Dateien: -l, -c und -d**

Eine Liste der im Paket enthaltenen Dateien bekommen Sie durch den Schalter -l.

```

defiant:~ # rpm -ql cpio
/bin/cpio
/usr/bin/cpio
/usr/bin/mt
/usr/share/info/cpio.info.gz
/usr/share/man/man1/cpio.1.gz
/usr/share/man/man1/mt.1.gz

```

Bei diesem Schalter können Sie durch den Einsatz des Sabbelschalters -v ausführlichere Informationen über die Dateien bekommen.

```

defiant:~ # rpm -qlv cpio
-rwxr-xr-x    1 root    root           64215 Sep 23  2003 /bin/cpio
lrwxrwxrwx    1 root    root             9 Sep 23  2003 /usr/bin/cpio -> /bin/cpio
-rwxr-xr-x    1 root    root          19445 Sep 23  2003 /usr/bin/mt
-rw-r--r--    1 root    root           6093 Sep 23  2003 /usr/share/info/cpio.info.gz
-rw-r--r--    1 root    root          4235 Sep 23  2003 /usr/share/man/man1/cpio.1.gz
-rw-r--r--    1 root    root           2031 Sep 23  2003 /usr/share/man/man1/mt.1.gz

```

Sie können sich aber auch aus der Liste die Dokumentationsdateien -d herauspicken, oder Konfigurationsdateien -c herauspicken

```

defiant:~ # rpm -qd cron
/usr/share/doc/packages/cron/CHANGES
/usr/share/doc/packages/cron/CONVERSION
/usr/share/doc/packages/cron/FEATURES
/usr/share/doc/packages/cron/MAIL
/usr/share/doc/packages/cron/MANIFEST
/usr/share/doc/packages/cron/README
/usr/share/doc/packages/cron/THANKS
/usr/share/man/man1/crontab.1.gz
/usr/share/man/man5/crontab.5.gz
/usr/share/man/man8/cron.8.gz
defiant:~ # rpm -qc cron
/etc/crontab
/etc/init.d/cron
/var/spool/cron/deny

```

Der Schalter -c hat auch Auswirkungen auf andere Ausgaben. Sie können sich z. B. so die Konfigurationsdatei zu einem Programm ganz einfach über die RPM-Datenbank ausgeben lassen.

```
defiant:~ # rpm -qf /bin/vim
vim-6.2-74
defiant:~ # rpm -qcf /bin/vim
/etc/vimrc
```

### Status einer Datei: -s

Auch der Status einer Datei aus dem Paket kann angezeigt werden. Dabei kann die Datei sich in vier definierten Zuständen befinden.

**normal** Dies ist der häufigste Zustand, der vorkommt. Die Datei wurde von keinem anderem Paket modifiziert, ersetzt oder gelöscht.

**replaced** Dieser Zustand tritt ein, wenn ein anderes Paket diese Datei ersetzt hat.

**not installed** Da es ja möglich ist nur Teile eines Pakets zu installieren, kann auch dieser Zustand auftreten. Meistens ist das der Fall bei Dokumentationsdateien.

**net shared** Die Datei befindet sich auf einem gemounteten Netzwerkverzeichnis.

Im unteren Beispiel können wir sehen, daß bei `hdparm` alles "normal" ist.

```
defiant:~ # rpm -qs hdparm
normal      /sbin/hdparm
normal      /usr/share/doc/packages/hdparm
normal      /usr/share/doc/packages/hdparm/Changelog
normal      /usr/share/man/man8/hdparm.8.gz
```

### Ressourcen, die das Paket zur Verfügung stellt: --provides

Die meisten Pakete stellen sogenannte Ressourcen zur Verfügung. Dabei handelt es sich nicht um Dateinamen.

```
defiant:~ # rpm -q --provides cpio
cpio = 2.5-209
defiant:~ # rpm -q --whatprovides rpm
rpm-4.1.1-71
defiant:~ # rpm -q --provides rpm-4.1.1-71
rpminst
librpm-4.1.so
librpmdb-4.1.so
librpmio-4.1.so
rpm = 4.1.1-71
```

### Ressourcen, die das Paket benötigt: --requires

Ein Paket stellt nicht nur Ressourcen zur Verfügung. In den meisten Fällen benötigt ein Paket auch Ressourcen. Diese benötigten Ressourcen können Sie sich dann unter Angabe der Option `--requires` anzeigen lassen.

```
defiant:~ # rpm -q --requires cpio
info
/bin/sh
/bin/sh
rpmLib(PayloadFilesHavePrefix) <= 4.0-1
rpmLib(CompressedFileNames) <= 3.0.4-1
```

```
ld-linux.so.2
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.2)
libc.so.6(GLIBC_2.2.3)
libnsl.so.1
rpmLib(PayloadIsBzip2) <= 3.0.5-1
defiant:~ # rpm -q --whatprovides libc.so.6
glibc-2.3.2-88
```

### Alle Dateinformationen: --dump

Die Option `--dump` ermöglicht die Anzeige aller Informationen über die installierten Dateien eines Pakets.

```
defiant:~ # rpm -q --dump cpio
/bin/cpio 64215 1064338305 80d0d1c254d7d8b1d26e01c6965e82de 0100755 root root 0 0 0 X
/usr/bin/cpio 9 1064338305 00000000000000000000000000000000 0120777 root root 0 0 0 /bin/cpio
/usr/bin/mt 19445 1064338305 db462cdd66b62145772837206c4deea4 0100755 root root 0 0 0 X
/usr/share/info/cpio.info.gz 6093 1064338304 e4da42e2a3b91e15f382805cfaa039d5 0100644 root root 0 1 0 X
/usr/share/man/man1/cpio.1.gz 4235 1064338304 6e73fade22bf5f797eb1a8f9d228c81f 0100644 root root 0 1 0 X
/usr/share/man/man1/mt.1.gz 2031 1064338304 7bedfee0b7194958f49a88aea65f8ef0 0100644 root root 0 1 0 X
```

Die Informationen werden nicht gerade übersichtlich dargestellt. Gehen wir sie doch einfach mal von links nach rechts am folgenden Beispiel durch.

```
defiant:~ # rpm -q --dump cpio
/bin/cpio 64215 1064338305 80d0d1c254d7d8b1d26e01c6965e82de 0100755 root root 0 0 0 X
...
```

- Der erste Eintrag ist natürlich der Dateiname `“/bin/cpio”`.
- Die Datei ist genau 64215 Bytes groß.
- Und was ist mit 1064338305 gemeint? Dies ist der Zeitpunkt der letzten Veränderung. Er wird gemessen in Sekunden seit dem 01.01.1970 0 Uhr.
- 80d0d1c254d7d8b1d26e01c6965e82de ist die 128 Bit-lange MD5-Checksumme des Dateiinhalts.
- Als nächstes folgt der Dateimodus mit 0100755.
- Der nächste Eintrag bezeichnet den Dateieigentümer.
- Und danach kommt natürlich die zugeordnete Gruppe.
- Die folgende Null zeigt an, daß es sich hierbei um keine Konfigurationsdatei handelt. Hier würde eine 1 stehen, wenn es eine Konfigurationsdatei wäre.
- Auch die nächste Null hat mit dem Dateityp zu tun. Hiermit wird angezeigt, daß es sich hierbei um keine Dokumentationsdatei handelt.
- Eine Null gibt es noch. Hier steht für Gerätedateien die Major und Minor Device Number.
- Hier steht nur ein X. Es handelt sich um eine gewöhnliche Datei. Wenn es sich um einen symbolischen Link handeln würde, stünde hier der Pfad auf den der Link zeigt.

**De- und Installationsskripte: --scripts**

Die Skripte zur Installation und Deinstallation der Pakete können Sie sich mit der Option `--scripts` anzeigen lassen. Es gibt insgesamt 5 Skripte, die aber nicht bei jedem Paket vorhanden sein müssen.

**preinstall** Wird ausgeführt vor der Installation der Dateien.

**postinstall** Wird ausgeführt nach der Installation der Dateien.

**preuninstall** Wird ausgeführt vor dem Löschen der Dateien.

**postuninstall** Wird ausgeführt nach dem Löschen der Dateien.

**verify** Wird während der Überprüfung der Installation ausgeführt.

Hier ein Beispiel für Skripte für das Paket *cpio*.

```
defiant:~ # rpm -q --scripts cpio-2.5-209
postinstall scriptlet (through /bin/sh):

    if test -x sbin/install-info ; then
        sbin/install-info --info-dir=/usr/share/info /usr/share/info/cpio.info.gz
    fi ;
postuninstall scriptlet (through /bin/sh):

ALL_ARGS=(--info-dir=/usr/share/info /usr/share/info/cpio.info.gz)
NUM_ARGS=${#ALL_ARGS[*]}
if test -x sbin/install-info ; then
    if ! test -e "${ALL_ARGS[$((NUM_ARGS-1))]}" ; then
        sbin/install-info --quiet --delete ${ALL_ARGS[*]}
    fi ;
fi ;
```

**Ausgabe formatieren: --queryformat und --querytags**

Sie können viele Informationen aus dem Paket beziehen. Damit diese übersichtlich dargestellt werden können, wird die Option `--queryformat` verwendet. Diese Option hat eine Unmenge von Einstellungsmöglichkeiten, auf die ich hier nicht weiter eingehen will.

Ein einfaches Beispiel für eine Ausgabe.

```
defiant:~ # rpm -q --queryformat "Hallo Welt\n" cpio
Hallo Welt
```

Spannend wird es natürlich erst, wenn Informationen ausgelesen werden. Die Liste der möglichen auslesbaren Felder erhalten Sie durch die Option `--querytags`. Die Liste ist sehr lang.

```
defiant:~ # rpm --querytags
HEADERIMAGE
HEADERI18NTABLE
SIGSIZE
SIGMD5
PUBKEYS
...
```

Nun können Sie Informationen gezielt auslesen.

```
linux37:~ # rpm -q --queryformat "%{NAME}%{VERSION}%{RELEASE}\n" cpio
cpio2.5209
linux37:~ # rpm -q --queryformat "%{NAME}-%{VERSION}-%{RELEASE}\n" cpio
cpio-2.5-209
```



Sie können auch minimale Spaltengrößen für die einzelnen Felder definieren. Die Werte werden dann rechtsbündig ausgegeben. Wenn ein Minus vor der Zahl steht ist das Feld linksbündig.

```
defiant:~ # rpm -qa --queryformat "%-25{NAME} %15{VERSION} %5{RELEASE}\n"
aaa_skel                2003.9.18      4
kdeartwork3-sound       3.1.4        38
mesa                    5.0.1        61
suse-release            9.0           6
ispell-german           1.5          145
...
```

Auch Informationen über verschiedene Zeiten sind in den Paketen enthalten.

```
linux37:~ # rpm --querytags | grep TIME
BUILDTIME
INSTALLTIME
FILEMTIMES
...
```

Sie können sich z. B. den Installationszeitpunkt ausgeben lassen. Normalerweise können Sie mit der üblichen Zeitangabe nichts anfangen. Sie können daher mit einem sogenannten Flag die Ausgabe konvertieren.

```
defiant:~ # rpm -q --queryformat "%-25{NAME} %{INSTALLTIME}\n" cpio
cpio                988974986
defiant:~ # rpm -q --queryformat "%-25{NAME} %{INSTALLTIME:date}\n" cpio
cpio                Fri May  4 13:16:26 2001
```

Was ist aber, wenn Sie mehrere Einträge für einen Quertag haben? Dann packen Sie einfach diese Variable in eckige Klammern.

```
defiant:~ # rpm -q --queryformat "==%{NAME}==\n[%{FILENAMES}]\n" cpio
==cpio==
/bin/cpio
/usr/bin/cpio
/usr/bin/mt
/usr/share/info/cpio.info.gz
/usr/share/man/man1/cpio.1.gz
/usr/share/man/man1/mt.1.gz
```

Die können auch mehrere Werte in einem solchen Array einbinden.

```
defiant:~ # rpm -q --queryformat "==%{NAME}==\n[%{FILENAMES} (%{FILESIZES} Bytes)\n]" cpio
==cpio==
/bin/cpio (64215 Bytes)
/usr/bin/cpio (9 Bytes)
/usr/bin/mt (19445 Bytes)
/usr/share/info/cpio.info.gz (6093 Bytes)
/usr/share/man/man1/cpio.1.gz (4235 Bytes)
/usr/share/man/man1/mt.1.gz (2031 Bytes)
```

Sie können auch alle Werte in das Array packen. Werte, die im Paket nicht mehrfach vorkommen, werden dann mit einem Gleichheitszeichen versehen.

```
defiant:~ # rpm -q --queryformat "[%{=NAME}: %{FILENAMES} (%{FILESIZES} Bytes)\n]" cpio
cpio: /bin/cpio (64215 Bytes)
cpio: /usr/bin/cpio (9 Bytes)
```

```
cpio: /usr/bin/mt (19445 Bytes)
cpio: /usr/share/info/cpio.info.gz (6093 Bytes)
cpio: /usr/share/man/man1/cpio.1.gz (4235 Bytes)
cpio: /usr/share/man/man1/mt.1.gz (2031 Bytes)
```

Haben Sie mal Lust eine große Datei zu erstellen ?

```
defiant:~ # rpm -qa --queryformat "[%{=NAME}: %{FILENAMES} (%{FILESIZES} Bytes)\n]" \
> | sort > allepaketedateien.txt
```

### 14.3.4 Überprüfung

Dateien aus installierten Paketen können mit der im Paket vorgegebenen Konfiguration verglichen werden, wenn der Befehl `rpm -V` verwendet wird.

Dabei werden neun verschiedene Attribute der Dateien überprüft.

- Besitzer
- Gruppe
- Rechte
- MD5-Checksumme
- Größe
- Major Device Number
- Minor Device Number
- Symbolischer Link
- Veränderungszeitpunkt

Wenn veränderte Dateien vorhanden sind, dann werden diese ausgegeben mit einem bestimmten vorgesetzten Code. Jeder Punkt steht für einen bestanden Test. Bei einem fehlerhaften Test werden Zeichen ausgegeben. Die Bedeutung der Zeichen ist wie folgt:

**S** Die Dateigröße wurde verändert.

**M** Die Dateirechte wurden verändert.

**5** Der Inhalt der Datei ist verändert worden und damit die MD5-Checksumme.

**D** Die Major und Minor Device Number wurden verändert.

**L** Der Eintrag für den symbolischen Link wurde verändert.

**U** Der Besitzer der Datei wurde verändert.

**G** Die zugeordnete Gruppe wurde geändert.

**T** Der Änderungszeitpunkt der Datei wurde verändert.

Das kann dann wie folgt aussehen. Fehlende Dateien werden auch aufgeführt.

```
defiant:~ # rpm -V apache
S.5....T c /etc/httpd/httpd.conf
S.5....T c /etc/httpd/suse_addmodule.conf
S.5....T c /etc/httpd/suse_define.conf
S.5....T c /etc/httpd/suse_include.conf
S.5....T c /etc/httpd/suse_loadmodule.conf
S.5....T c /etc/httpd/suse_text.conf
missing    /var/log/httpd/ssl_engine_log
missing    /var/log/httpd/ssl_request_log
missing    /var/run/httpd.pid
```

Diese Überprüfung kann durch eine Reihe von Optionen gesteuert werden.

### Optionen

<b>-p <i>DATEI</i></b>	Bezieht sich auf eine RPM-Datei, anstatt auf ein installiertes Paket
<b>-f <i>DATEI</i></b>	Pakete, die die angegebene <i>DATEI</i> enthalten.
<b>-a</b>	Alle Pakete
<b>-g <i>GRUPPE</i></b>	Alle Pakete, die zu einer bestimmten Gruppe gehören
<b>--nodeps</b>	Führt keine Abhängigkeitsprüfung bei der Überprüfung durch.
<b>--nofiles</b>	Überprüft keine Dateiattribute.
<b>--noscripts</b>	Das bei der Installation auszuführende Skript wird nicht ausgeführt.

### Zuverlässigkeit des Pakets prüfen: `--checksig`

Vor der Installation können Sie die Dateiintegrität und die Seriösität der Quelle eines RPM-Pakets testen. Jeder RPM-Paket enthält eine MD5-Checksumme über das Paket. Dadurch kann festgestellt werden, ob die Daten beschädigt worden sind. Zusätzlich enthält das Paket eine Signatur des Herstellers, die über seinen öffentlichen Schlüssel überprüft werden kann. Dafür wird die Option `--checksig` oder als Kurzform `-K` verwendet.

```
root@defiant:~ # rpm --checksig gimp-1.3.20-21.i586.rpm
gimp2-1.3.20-21.i586.rpm: sha1 md5 gpg OK
root@defiant:~ # rpm --checksig gimp-2.0.1-5.i386.rpm
gimp-2.0.1-5.i386.rpm: (SHA1) DSA sha1 md5 (GPG) NOT OK (MISSING KEYS: GPG#30c9ecf8)
```

Inzwischen erfolgt die Überprüfung auch bei der Installation. Der öffentliche Schlüssel muß sich in der Datenbank von RPM befinden. Er wird dort als Paket abgespeichert.

```
root@defiant:~ # rpm -qa gpg-pubkey*
gpg-pubkey-3d25d3d9-36e12d04
gpg-pubkey-9c800aca-39eef481
linux37:/home/dozent/download # rpm -qi gpg-pubkey-9c800aca-39eef481
Name           : gpg-pubkey                      Relocations: (not relocateable)
Version        : 9c800aca                        Vendor: (none)
Release        : 39eef481                        Build Date: Fr 04 Mai 2001 13:06:43 CEST
Install date:  Fr 04 Mai 2001 13:06:43 CEST      Build Host: localhost
Group          : Public Keys                     Source RPM: (none)
Size           : 0                               License: pubkey
Signature      : (none)
Summary        : gpg(SuSE Package Signing Key <build@suse.de>)
Description    :
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: rpm-4.1.1 (beecrypt-2.2.0)
```

```
mQGIBDnu9IERBACT8Y35+2vv4MGVKiLEM019GdST6MckYS3yEKeueNWc+z/OKvff4JctBsgs
47tjmiI9sl0eHjm3gTR8rItXMN6sJEUHWzDP+YOPFPboMvKx0FX1/A0dM+HFrruGgBlWt6FA
...
```

-----END PGP PUBLIC KEY BLOCK-----

Distribution: (none)

Ein öffentlicher Schlüssel kann über die Option `--import` eingebunden werden. Gelöscht wird er wie ein normales Paket mit der Option `-e`.

### 14.3.5 Allgemeine Optionen

**Alternative Datenbank:** `--dbpath PFAD`

Normalerweise befindet sich die RPM-Datenbank im Verzeichnis `/var/lib/rpm`. Sie können in der Konfigurationsdatei `rpmrc` auch einen anderen alternativen Pfad zur Datenbank angeben. Und natürlich können Sie auch bei der Installation eine andere Datenbank ansprechen. Dies erfolgt über die Option `--dbpath`.

```
enterprise:~ # rpm -i --dbpath /mnt/var/lib/rpm blubb-42.42.42-42.i586.rpm
```

**Alternative Konfigurationsdatei:** `--rcfile KONFDATEI`

Sie können alternativ zu den normalen Konfigurationsdateien `/etc/rpmrc` und `~/rpmrc` auch bei der Installation eine andere Konfigurationsdatei angeben. Dies erfolgt mittels der Option `--rcfile` und der Angabe der neuen Konfigurationsdatei.

```
enterprise:~ # rpm -i --rcfile /etc/rpmrc.spezial spezial-1.2.34-56.i586.rpm
```

**Reparatur der RPM-Datenbank:** `--rebuilddb`

Es kann natürlich auch mal vorkommen, daß die RPM-Datenbanken beschädigt werden. In diesem Fall bekommen Sie Fehlermeldungen, wenn Sie ein Paket installieren oder Upgraden wollen. Als letzte Hilfe können Sie versuchen mit der Option `--rebuilddb` die Datenbank zu reorganisieren. Dieser Vorgang dauert etwas länger.

```
defiant:~ # rpm --rebuilddb
```

Bei einer funktionstüchtigen Datenbank gibt es keine Meldungen vom Programm.

## 14.4 Der Kernel

In Linux ist der *Kernel* der Kern der Software, die Ihr System verwaltet und kontrolliert. Er kontrolliert Hardware, Speicher und verwaltet die Prozesse. Daneben ist er eine Schnittstelle für den indirekten Zugriff der Anwenderprogramme auf die Hardware Ihres Systems.

Die Versionsnummer eines Kernels setzt sich aus mehreren Angaben zusammen: 2.4.16-4GB bedeutet, es handelt sich um die Kernelversion 2.4 mit dem Patchlevel 16<sup>3</sup>, der einen Arbeitsspeicher von 4 GB verwalten kann. Die zweite Ziffer der Kernelversion hat eine besondere Bedeutung. Kernelversionen mit einer geraden Ziffer am Ende (z. B. 2.0, 2.2, 2.4) gelten als stabile Kernels. Ungerade Nummern wie 2.1, 2.3 und 2.5 bezeichnen experimentelle Kernels, in denen neue Funktionen getestet werden. Solche Kernels sollten Sie nicht auf Produktionssystemen einsetzen, da Sie in der Regel instabil sind.

Informationen über den eingesetzten Kernel erhalten Sie über den Befehl `uname`.

---

<sup>3</sup>D. h. die 16. Fehlerkorrektur wurde berücksichtigt.

### 14.4.1 uname

Das Programm `uname` zeigt Informationen über das installierte System und den Kernel an.

```
uname [OPTIONEN]
```

Die Verwendung von `uname` ohne Schalter ist wie `uname -s`.

#### Optionen

<code>-a</code>   <code>--all</code>	Gibt alle Informationen aus
<code>-m</code>   <code>--machine</code>	Gibt den Hardwaretyp aus
<code>-n</code>   <code>--nodename</code>	Gibt den Netzwerkname des Rechners aus
<code>-r</code>   <code>--release</code>	Gibt die Release-Nummer des Systems aus
<code>-s</code>   <code>--sysname</code>	Gibt den Systemnamen aus
<code>-p</code>   <code>--processor</code>	Gibt den Prozessortyp aus
<code>-v</code>	Gibt die Versionsnummer aus

Der Befehl ohne Schalter gibt einfach nur den Namen des Betriebssystems aus.

```
tapico@defiant:~> uname
Linux
```

Werden mehrere Schalter oder der Schalter `-a` verwendet, dann werden die Informationen in der Reihenfolge `-snrvmp` ausgegeben.

```
tapico@defiant:~> uname -a
Linux defiant 2.4.10-4GB #1 Tue Sep 25 12:33:54 GMT 2001 i686 unknown
```

## 14.5 Verwaltung von Kernelmodulen

Die ersten Kernel (Linux 1.x) waren Unix-typische monolithische Kernel, in denen alle Funktionen fest eingekompiliert waren. Dies ergibt einen sehr schnellen und stabilen Kernel. Der Nachteil eines solchen Vorgehens zeigte sich aber, als der Kernel immer mehr Funktionen umfaßte und dadurch immer größer wurde. Außerdem mußte für jede neue Funktion ein neuer Kernel kompiliert werden. So reichte die Verwendung einer neuen Netzwerkkarte z. B. aus um den Kernel neu kompilieren zu müssen. Zwar hätte man alle möglichen Hardwaretreiber in den Kernel einkompilieren können, dann wäre der Kernel aber sehr groß geworden und teilweise verträgt sich bestimmte Hardware mit einigen Treibern nicht.

Um diesen Nachteil zu umgehen wurde der modularisierte Kernel entwickelt. Hier werden die meisten Hardwaretreiber als zusätzliche Objektdateien dem Kernel zur Verfügung gestellt und bei Bedarf im laufenden Betrieb hinzugeladen. Diese Module können im laufenden Betrieb ausgetauscht werden und ermöglichen so ein komfortables Aktualisieren des Systems.

Im Normalfall wird ein Modul dynamisch in den laufenden Kernel eingebunden, wenn es geladen wird. Die meisten Aktionen im Zusammenhang mit den Modulen laufen automatisch ab. Manchmal ist es aber auch nötig Modul manuell zu bearbeiten um z. B. einen neuen Treiber in das laufende System einzubinden. Die Liste der geladenen Module liefert der Befehl `lsmod` (14.5.2). Module werden mit `insmod` (14.5.3) oder `modprobe` (14.5.6) in den Kernel eingebunden und durch `rmmod` (14.5.4) wieder entfernt. Informationen über eingebundene Module bekommen Sie über den Befehl `modinfo` (14.5.5).

### 14.5.1 Die Moduldateien

Die Kernelmodule befinden sich in einem Unterverzeichnis von `/lib/modules` mit dem Namen der Kernelversion. Es können daher auch mehrere Modulverzeichnisse vorhanden sein, wenn mehrere Kernelversionen installiert wurden.

```
root@defiant:/ # uname -r
2.4.10-4GB
```

```

root@defiant:/ # ls -lG /lib/modules/2.4.10-4GB/
insgesamt 271
lrwxrwxrwx    1 root      26 Nov 22 16:49 build -> /usr/src/linux-2.4.10.SuSE
drwxr-xr-x    2 root      341 Nov 22 16:50 dvb
drwxr-xr-x    7 root      134 Nov 22 16:50 kernel
drwxr-xr-x    2 root     4735 Nov 22 16:50 misc
-rw-r--r--    1 root    127579 Jan 23 19:23 modules.dep
-rw-r--r--    1 root      31 Jan 23 19:23 modules.generic_string
-rw-r--r--    1 root     7965 Jan 23 19:23 modules.isapnpmap
-rw-r--r--    1 root      29 Jan 23 19:23 modules.parportmap
-rw-r--r--    1 root    51647 Jan 23 19:23 modules.pcimap
-rw-r--r--    1 root    65049 Jan 23 19:23 modules.usbmap
drwxr-xr-x    2 root      61 Nov 22 16:50 net
drwxr-xr-x    2 root     1018 Nov 22 16:50 pcmcia
drwxr-xr-x    3 root     1433 Nov 22 16:50 pcmcia-external
drwxr-xr-x    2 root     136 Nov 22 16:50 thinkpad

```

In diesem Verzeichnis existieren wiederum Unterverzeichnisse für die verschiedenen Typen von Modulen. Typische Unterverzeichnisse sind:

**block** Module für ein paar blockorientierte Geräte wie RAID-Kontroller oder IDE-Bandlaufwerke.

**cdrom** Treiber für nicht standartisierte CD-ROM-Laufwerke.

**fs** Module für Dateisysteme wie ReiserFS.

**ipv4** Enthält Module, die im Zusammenhang mit IP stehen.

**misc** Hier kommt alles rein, was nicht in die anderen Kategorien paßt.

**net** Treiber für die Netzwerkkarten.

**scsi** Enthält die Treiber für die SCSI-Kontroller.

**video** Für die speziellen Treiber der Grafikkarten.

## 14.5.2 lsmod

Dieser Befehl zeigt Ihnen für jedes geladene Modul den Namen, die Größe, die Anzahl der Benutzungen und eine Liste mit verknüpften Modulen.

```
lsmod
```

Außer Versionsnummer (**-v**) und der Hilfe (**-help**) besitzt **lsmod** keine Schalter.

Der Aufruf von **lsmod** führt zu der einfachen Auflistung aller geladenen Kernelmodule.

```

defiant:/ # lsmod
Module                Size  Used by
parport_pc            19280   1 (autoclean)
lp                    5248    0 (autoclean)
parport               22240   1 (autoclean) [parport_pc lp]
nfsd                  64880   4 (autoclean)
ipv6                 124736  -1 (autoclean)
evdev                 4160    0 (unused)
input                 3072    0 [evdev]
usb-ohci              17680    0 (unused)
iptable_nat           12656    0 (autoclean) (unused)
ip_conntrack          12848    1 (autoclean) [iptable_nat]
iptable_filter         1728    0 (autoclean) (unused)

```

```
ip_tables          10496   4  [iptable_nat iptable_filter]
reiserfs           147920   2
usbcore            47264   1  [usb-ohci]
```

Im Prinzip ist dieses Programm sehr primitiv. Es zeigt einfach den Inhalt der virtuellen Datei `/proc/modules` an.

```
defiant:/ # cat /proc/modules
Module              Size  Used by
parport_pc          19280   1  (autoclean)
lp                  5248    0  (autoclean)
parport              22240   1  (autoclean) [parport_pc lp]
nfsd                 64880   4  (autoclean)
...
```

### 14.5.3 insmod

Das Kommando `insmod` bindet ein Modul in einen laufenden Kernel ein. Das Modul wird automatisch lokalisiert und eingebunden. Diese Funktion steht natürlich nur dem Superuser zur Verfügung.

```
insmod [OPTIONEN] MODUL [MODULOPTIONEN]
```

Hier eine Auswahl von Optionen des Befehls `insmod`. Eine vollständige Liste erhalten Sie über den Befehl `insmod --help`.

#### Optionen

<code>-f   --force</code>	Erlaubt das Laden des Moduls auch bei falscher Kernelversion
<code>-k   --autoclean</code>	Gibt dem Modul den Status "autoclean"
<code>-L   --lock</code>	Verhindert das mehrfache Laden des gleichen Moduls
<code>-m   --map</code>	Generiert eine <i>Load Map</i> um Fehler besser nachverfolgen zu können
<code>-p   --probe</code>	Überprüft ob das Modul mit dem Kernel zusammenarbeitet
<code>-r   --root</code>	Erlaubt <i>root</i> Module zu laden, die nicht <i>root</i> gehören
<code>-s   --syslog</code>	Gibt die Resultate an <code>syslogd</code> weiter, anstatt sie auf dem Terminal auszugeben
<code>-v   --verbose</code>	Liefert ausführliche Informationen

Um ein Modul zu laden muß man nur seinen Namen angeben.

```
root@defiant:/ # insmod umsdos
Using /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
```

Wenn der Objektname ohne Pfadangabe und ohne den Suffix `.o` angegeben wird, sucht `insmod` automatisch in den Modulverzeichnissen des laufenden Kernels.

Schauen wir uns doch mal die Arbeitsweise von `insmod` an. Die Module liegen in Form eines Objektcodes vor (Suffix `.o`). Dieser Objektcode enthält genau die Symbole, die der Kernel in seiner Symboltabelle zum Zugriff auf das jeweilige Gerät verwendet. Der Suchpfad kann durch die Umgebungsvariable `MODPATH` erweitert werden.

Soll das Modul im aktuellen Verzeichnis geladen werden, so müssen Sie den vollständigen Dateinamen inklusive Endung angeben.

Basiert das Modul hingegen auf ein anderes Modul, so wird das Modul nicht geladen und es werden Fehlermeldungen ausgegeben. In diesem Fall müssen natürlich die benötigten Module vorher geladen werden. Aus diesem Grund ist die Verwendung des Programms `modprobe` (14.5.6) vorzuziehen.

### 14.5.4 rmmod

Um ein Modul wieder aus dem laufenden Kernel zu entfernen, wird der Befehl `rmmod` benutzt. Dies klappt allerdings nur, wenn das Modul unbenutzt ist und nicht von einem anderen Modul benötigt wird. Natürlich darf nur der Superuser Module aus dem laufenden Kernel entfernen.

**rmmod [OPTIONEN] MODULE**

#### Optionen

Hier eine Reihe von Optionen zu <b>rmmod</b> .	<b>-a</b>   <b>-all</b>	Entfernt alle unbenutzten Module
	<b>-s</b>   <b>-syslog</b>	Gibt die Resultate an <b>syslogd</b> weiter, anstatt sie auf dem Termin
	<b>-v</b>   <b>-verbose</b>	Liefert ausführliche Informationen

Module werden einfach durch Angabe ihres Namens beim **rmmod**-Befehl aus dem laufenden Kernel entfernt.

```
root@defiant:/ # insmod ntfs
Using /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
root@defiant:/ # lsmod | grep ntfs
ntfs                47216    0  (unused)
root@defiant:/ # rmmod ntfs
root@defiant:/ # lsmod | grep ntfs
root@defiant:/ #
```

Sollte aber der Kernel das Modul benötigen, wird der Versuch mit einer Fehlermeldung quittiert.

```
root@defiant:/ # rmmod usbcore
usbcore: Device or resource busy
```

Um alle unbenutzten Module zu entfernen wird die Option **-a** verwendet. Dabei muß der Befehl aber zwei mal aufgerufen werden. Im ersten Durchgang werden die unbenutzten Module nur markiert. Im nächsten Durchgang werden die unmarkierten unbenutzten Module markiert und die markierten unbenutzten Module gelöscht. Dies soll verhindern, daß nur zeitweise ungenutzte Module entfernt werden.

Übrigens können Sie Module auch mit dem Programm **modprobe** (14.5.6) mit dem Schalter **-r** entfernen.

### 14.5.5 modinfo

Mit **modinfo** bekommen Sie Informationen über das Modul aus der Moduldatei. Allerdings enthalten nicht alle Module ausführliche Informationen. Manche begnügen sich mit einer kurzen beschreibenden Zeile.

**modinfo [OPTIONEN] MODULDATEI**

#### Optionen

<b>-a</b>	Zeigt den Autoren des Moduls an ( <i>author</i> )
<b>-d</b>	Zeigt die Beschreibung des Moduls an ( <i>description</i> )
<b>-p</b>	Zeigt die Parameter des Moduls an ( <i>parameters</i> )

#### Beispiel

Hier sind zwei Beispiel für die Module **umsdos** und **sonypi**.

```
root@defiant:/ # modinfo /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
filename:    /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
description: <none>
author:      <none>
root@defiant:/ # modinfo /lib/modules/2.4.10-4GB/kernel/drivers/char/sonypi.o
filename:    /lib/modules/2.4.10-4GB/kernel/drivers/char/sonypi.o
description: "Sony Programmable I/O Control Device driver"
author:      "Stelian Pop <stelian.pop@fr.alcove.com>"
parm:        minor int, description "minor number of the misc device, default is -1 (automatic)"
parm:        verbose int, description "be verbose, default is 0 (no)"
parm:        fnkeyinit int, description "set this if your Fn keys do not generate any event"
parm:        camera int, description "set this if you have a MotionEye camera (PictureBook series)"
```



### 14.5.6 modprobe

Wie `insmod` (14.5.3) wird `modprobe` dazu benutzt dem Kernel Module hinzuzufügen. Eigentlich ist `modprobe` nur ein Wrapper für `insmod`, stellt aber dem Benutzer wesentlich mehr Funktionen zur Verfügung. So können automatisch neben dem eigentlichen Modul auch die dafür benötigten Module automatisch geladen werden. Auch das Laden aller Module eines Verzeichnis, z. B. um das richtige Modul für eine Hardwarekomponente zu finden, ist möglich.

`modprobe [OPTIONEN] MODULE`

#### Optionen

<code>-a</code>	Lädt alle Module ( <i>all</i> )
<code>-c</code>	Anzeige einer kompletten Modulkonfiguration ( <i>configuration</i> )
<code>-l</code>	Auflisten der vorhandenen Module ( <i>list</i> )
<code>-r</code>	Entfernt Module wie <code>rmmod</code> (14.5.4) ( <i>remove</i> )
<code>-s</code>	Leitet die Ergebnisse an den Protokolldaemon <code>syslog</code> weiter ( <i>syslog</i> )
<code>-t TAG</code>	Bezieht die Aktionen auf die Module des Verzeichnis <code>TAG</code> ( <i>tag</i> )
<code>-v</code>	Erweiterte Bildschirmausgabe ( <i>verbose</i> )

#### Beispiele

Um das Modul `ntfs` zu installieren reicht der folgenden Befehl.

```
root@defiant:/ # modprobe -v ntfs
/sbin/insmod /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
Using /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
Symbol version prefix ''
root@defiant:/ # lsmod | grep ntfs
ntfs                47216      0  (unused)
```

Sollte das zu ladende Modul auf anderen Modulen basieren, so kann `modprobe` dies aus der jeweiligen `modules.dep` entnehmen.

### 14.5.7 Modulabhängigkeiten: modules.dep

Die Informationen über die Abhängigkeiten eines Moduls von anderen Kernelmodulen findet das Programm `modprobe` in der Datei `/lib/modules/Kernel-Version/modules.dep`. Die Zeilen mit den Abhängigkeiten sehen wie folgt aus:

```
modulname.o: Abhängigkeit_1 Abhängigkeit_2 ...
```

Hier ein Beispiel für das `ext3`-Dateisystem aus der `modules.dep`

```
ole@enterprise:/lib/modules/2.4.10-4GB> grep ext3 modules.dep
/lib/modules/2.4.10-4GB/kernel/fs/ext3/ext3.o:
/lib/modules/2.4.10-4GB/kernel/fs/jbd/jbd.o
```

Alle für das System zur Verfügung stehenden Module sind hier aufgelistet mit den Modulen, auf denen sie basieren. Sollte das Modul keine anderen Module benötigen, wird es trotzdem in der Datei aufgeführt. Wenn jetzt mit `modprobe` ein Modul geladen werden soll, so konsultiert das Programm erst diese Datei und lädt die benötigten Module zuerst. Erst wenn alle benötigten Module eingebunden worden sind, dann wird auch das eigentlich angeforderte Modul geladen.

Die Datei `modules.dep` muß immer auf dem neuesten Stand sein. Sollte sich in den Abhängigkeiten etwas geändert haben ohne daß diese Änderung hier erfaßt wurde, dann kann `modprobe` eventuell Module nicht erfolgreich in den Kernel einbinden. Aus diesem Grund wird in den meisten Distributionen diese Datei bei jedem Start des Systems mittels des Befehls `depmod` neu erstellt. Bei SuSE 7.3 erfolgt dies in der Datei `/etc/rc.d/boot` wie im folgenden Ausschnitt zu sehen ist.

```
#
# initialize database for kernel. This should be done earlier, but
# could cause a lot of trouble with damaged file systems.
# restart of kernel will be done by /etc/init.d/kernel
#
MODULES_DIR=/lib/modules/`uname -r`
if test -x /sbin/depmod -a -d $MODULES_DIR ; then
    for i in $MODULES_DIR/* $MODULES_DIR/*/* /etc/modules.conf ; do
        test -e $i || continue
        if test $i -nt $MODULES_DIR/modules.dep ; then
            rm -f $MODULES_DIR/modules.dep
            break
        fi
    done
    if test ! -e $MODULES_DIR/modules.dep ; then
        echo -n Setting up $MODULES_DIR
        /sbin/depmod -a > /dev/null 2>&1
        rc_status -v -r
    fi
fi
```

### 14.5.8 depmod

Das Tool `depmod` findet die Abhängigkeiten zwischen ladbaren Kernelmodulen und schreibt die dafür zuständige Konfigurationsdatei `modules.dep`.

`depmod [OPTIONEN] [MODULLISTE]`

#### Optionen

<code>-a</code>	Sucht nach Modulen in allen Verzeichnissen, die in <code>moduls.conf</code> definiert wurden( <i>all</i> )
<code>-A</code>	Wie <code>-a</code> , führt aber nur ein Update durch( <i>all</i> )
<code>-e</code>	Zeigt alle nicht aufgelösten Symbole für das Modul
<code>-n</code>	Schreibt die Abhängigkeiten auf den Bildschirm und nicht in <code>modules.dep</code> ( <i>not</i> )
<code>-s</code>	Fehlermeldungen gehen an <code>syslog</code> und nicht <code>stderr</code> ( <i>syslog</i> )
<code>-v</code>	Zeigt alle bearbeiteten Module an( <i>verbose</i> )
<code>-q</code>	Arbeitet ohne Meldungen ( <i>quiet</i> )
<code>-V</code>	Zeigt die Versionsnummer an ( <i>Version</i> )
<code>-b</code>	Setzt eine neues Basisverzeichnis fest( <i>Basis</i> )
<b>BASISVERZEICHNIS</b>	
<code>-C KONFIGDATEI</code>	Konfigurationsdatei festlegen ( <i>Config</i> )
<code>-F SYMBOLE</code>	Bei der Bestimmung von Abhängigkeiten für einen nicht laufenden Kernel, müssen hier die passenden Kernelsymbole angegeben werden.

Um z. B. alle Abhängigkeiten auf dem Bildschirm anzuzeigen anstatt in die Konfigurationsdatei zu schreiben, kann folgender Befehl verwendet werden.

```
enterprise:/etc/rc.d # depmod -an
/lib/modules/2.4.10-4GB/kernel/abi/cxenix/abi-cxenix.o:
/lib/modules/2.4.10-4GB/kernel/abi/avr4/abi-avr4.o \
    /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o

/lib/modules/2.4.10-4GB/kernel/abi/ibcs/abi-ibcs.o:
/lib/modules/2.4.10-4GB/kernel/abi/avr4/abi-avr4.o \
    /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o

/lib/modules/2.4.10-4GB/kernel/abi/sco/abi-sco.o:
/lib/modules/2.4.10-4GB/kernel/abi/avr4/abi-avr4.o \
    /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o \
```

```

/lib/modules/2.4.10-4GB/kernel/abi/cxenix/abi-cxenix.o
...

```

### 14.5.9 Die Konfigurationsdatei: modules.conf

Die Datei `/etc/modules.conf`, die auch in alten Systemen unter dem Namen `/etc/conf.modules` zu finden ist, enthält die Liste der Module mit ihren Parametern. Dabei handelt es sich z. B. bei Modulen für Hardware um Parameter wie I/O-Adressen und Interrupts. Das Verhalten des Tools `modprobe` wird durch diese Einstellungen kontrolliert und gesteuert.

Ein Ausschnitt aus der `/etc/modules.conf` könnte so aussehen.

```

alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none,none

options ne                io=0x300

alias block-major-1      rd
alias block-major-2      floppy

ptions bttv      pll=1 radio=0 card=0
post-install bttv /sbin/modprobe "-k" tuner;

options dummy0 -o dummy0
options dummy1 -o dummy1

# ppp over ethernet
# the kernel 2.2 uses pppox
# the kernel 2.4 uses pppoe
if 'kernelversion' == "2.2"
alias char-major-144      pppox
post-install pppox insmod mssclmpfw
pre-remove pppox rmmod mssclmpfw
else
alias char-major-108      ppp_async
alias char-major-144      pppoe
alias net-pf-24           pppoe
endif

# agpgart is i386 only right now
pre-install mga /sbin/modprobe "-k" "agpgart"
pre-install r128 /sbin/modprobe "-k" "agpgart"
pre-install radeon /sbin/modprobe "-k" "agpgart"
options agpgart agp_try_unsupported=1

```

Die folgenden Informationen finden Sie in der `/etc/modules.conf`:

#### Kommentare

Leere Zeilen und Zeilen mit einem `#` am Anfang werden ignoriert.

#### keep

Wenn der `keep` Parameter vor den path-Anweisungen zu finden ist, dann wird der Standardpfad übernommen und zu jeden spezifizierten Pfaden hinzugefügt.

#### depfile=ABSOLUTER\_PFAD

Dieser Eintrag überschreibt die Standardeinstellung für den Ort der Datei für die Modulabhängigkeiten.

#### options MODUL OPT1=WERT1 OPT2=WERT2 ...

Mit dieser Zeile können Optionen für die Module mitgegeben werden. `MODUL` bezeichnet das einzelne Modul ohne die Endung `.so`. Die Optionen werden immer als Paar `Name=Wert` angegeben.

**alias**

Aliase werden dazu benutzt generische Namen speziellen Modulen zuzuordnen. Dies ist nur eine Erleichterung beim Anwenden der Modulbefehle, sondern oft existentiell wichtig. So verbindet der Eintrag

```
alias eth0 e100
```

die Netzwerkschnittstelle `eth0` direkt mit dem Treiber der Netzwerkkarte "EtherExpress 100".

**pre-install MODUL KOMMANDO**

Diese Anweisung führt dazu, daß vor dem Einfügen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

**install MODUL KOMMANDO**

Hiermit kann der vorgesehene Befehl zum Einbinden des Moduls durch einen Shell-Befehl ersetzt werden.

**post-install MODUL KOMMANDO**

Diese Anweisung führt dazu, daß nach dem Einfügen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

**pre-remove MODUL KOMMANDO**

Diese Anweisung führt dazu, daß vor dem Entfernen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

**remove MODUL KOMMANDO**

Hiermit kann der vorgesehene Befehl zum Entfernen des Moduls durch einen Shell-Befehl ersetzt werden.

**post-remove MODUL KOMMANDO**

Diese Anweisung führt dazu, daß nach dem Entfernen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

## 14.6 Kernelkompilierung

In der heutigen Zeit sind die mit den jeweiligen Distributionen mitgelieferten Kernels, dank der individuell integrierbaren Modulen, so universell, daß der Normalanwender keine speziellen Kernel für sich mehr bauen muß. Welche Gründe gibt es also noch heute einen Kernel selbst zu kompilieren.

1. Sie wollen Ihren Bekannten mit Ihrem Insiderwissen, "Ich habe meinen Kernel selbst kompiliert!", beeindrucken.<sup>4</sup>
2. Ihre neue Hardware wird nicht von Ihrem alten Kernel unterstützt.
3. Ihre neue Software benötigt neue Funktionalitäten, die der alte Kernel nicht zur Verfügung stellt.
4. Der alte Kernel ist instabil und/oder enthält Sicherheitslücken.
5. Sie wollen einen optimierten Kernel auf Rechner mit speziellen Funktionen wie Router, Druckserver, Firewall u.s.w einsetzen.
6. Sie sind einfach neugierig, was der neue Kernel kann oder wie das Kernelkompilieren geht.

Um zu erfahren, welche Hardware vom Kernel unterstützt wird, lesen Sie das Hardware-HOWTO, schauen Sie in die `config.in` Dateien in den Kernelquellen oder sehen Sie sich die Ausgaben des Befehls `make config` an.

Bevor Sie mit der Kompilierung des Kernels beginnen, sollten Sie sich die Struktur ihres Dateisystems noch einmal in Erinnerung rufen, damit Sie im Notfall Ihr System mit einer Rettungsdiskette wieder zum Laufen bringen können. Dazu können Sie entweder den Befehl `mount` (10.3.1) oder die Datei `/etc/fstab` (10.3.5) verwenden.

---

<sup>4</sup>Das klappt leider nicht immer, da die meisten Menschen mit dem Begriff Kernel nichts anfangen können.

### 14.6.1 Kernelquellen

Um einen Kernel kompilieren zu können, brauchen Sie den Quellcode. Bei der SuSE-Distribution sind zwei Pakete vorhanden. Das Paket `kernel-source` enthält die Quellen des veränderten SuSE-Kernels, während das Paket `linux` den Quellcode des Originalkernel enthält.<sup>5</sup>

Die Quellen des neuesten Kernels können über Anonymous-FTP von `ftp.funet.fi` bezogen werden; sie befinden sich dort unterhalb des Verzeichnisses `/pub/Linux/PEOPLE/Linus`. Dieser Server wird an vielen Stellen gespiegelt, es lohnt sich also, zunächst mal auf den lokalen Spiegel-Servern nachzusehen. Im angegebenen Verzeichnis befinden sich zumeist unter dem Verzeichnis `kernel` Unterverzeichnisse mit Namen wie `v2.2`, `v2.4`, `v2.6` usw. In den jeweiligen Verzeichnissen stehen dann die Linux-Quellen in Dateien mit den Namen `linux-x.y.z.tar.gz`. `x.y.z` ist dabei die Versionsnummer. Die Datei mit der höchsten Versionsnummer stellt den neuesten Kernel dar. In der Regel sollte dieses auch die beste Version sein.

Sie sollten aber eine schnelle Anbindung ans Internet besitzen oder eine Flatrate und etwas Geduld. In der Version 2.6.6 ist das Tarball-Archiv 41,8 MB groß.<sup>6</sup>

Um die Versionsnummer des aktuellen Kernels zu bestimmen, verwenden Sie den Befehl `uname`. Die Versionsnummer der Quellen können Sie der Datei `Makefile` entnehmen.

```
enterprise:/ # uname -sr
Linux 2.4.10-4GB
enterprise:/ # head -4 /usr/src/linux/Makefile
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 18
EXTRAVERSION = Athlon
```

Der Eintrag `EXTRAVERSION` ist normalerweise leer. Hier können Sie für Ihre spezielle Kernelvariante einen Namen vergeben (z. B. `Athlon` für einen für diesen Prozessortyp optimierten Kernel).

Die Quellen müssen sich im Verzeichnis `/usr/src/linux` befinden. Wenn Sie mit mehreren Kernelversionen parallel arbeiten, bietet es sich an, die jeweiligen Quellen in individuellen Verzeichnissen zu speichern und `/usr/src/linux` als symbolischen Link auf die jeweiligen Verzeichnisse zeigen zu lassen.

```
enterprise:/usr/src # ls -Gl
insgesamt 2
lrwxrwxrwx    1 root          18 Mär 20 15:18 linux -> linux-2.4.18
drwxr-xr-x    15 root        530 Mär 19 10:58 linux-2.4.10.SuSE
drwxrwxrwx     3 root         91 Mär 20 15:18 linux-2.4.18
```

Neben Linus Thorvald, der den Anstoß zu Linux gab, haben noch viele Menschen am Kernel mitgewirkt. Eine Liste dieser Personen mit ihren Beiträgen finden Sie in der Datei `CREDITS` in den Kernelquellen.

### 14.6.2 Konfiguration

Um die Elemente des Kernels festzulegen, muß erst eine Konfigurationsdatei `.config` angelegt werden. Dazu wechseln Sie in das Verzeichnis `/usr/src/linux` und benutzen eines der folgenden Tools.

Die einzelnen Tools werden durch Einträge in der Datei `Makefile` gestartet, wie hier in einem Auszug dieser Datei zu sehen ist.

```
xconfig: symlinks
        $(MAKE) -C scripts kconfig.tk
        wish -f scripts/kconfig.tk

menuconfig: include/linux/version.h symlinks
```

<sup>5</sup>Stand: SuSE 7.3

<sup>6</sup>Bei einer ISDN-Leitung mit 7 kB/s dauert der Download des Archivs ungefähr 100 Minuten. Mit T-DSL und einer Geschwindigkeit von 87 kB/s war das ganze schon in 8 Minuten erledigt.

```

$(MAKE) -C scripts/lxdialog all
$(CONFIG_SHELL) scripts/Menuconfig arch/$(ARCH)/config.in

config: symlinks
$(CONFIG_SHELL) scripts/Configure arch/$(ARCH)/config.in

```

**make config** Der Befehl **make config** startet ein sehr einfaches Tool, daß Sie nacheinander die einzelnen Elemente abfragt. Dies ist vor allem dann sehr ärgerlich, wenn Sie nur ein Element ändern wollen und dann die ganze Prozedur durcharbeiten müssen.

```

enterprise:/usr/src/linux # make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?] N
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]

```

**make menuconfig** Dieses Tool ist ebenfalls für die Textkonsole gedacht. Sie können aber die einzelnen Punkte über ein Menü auswählen. So ist es möglich einzelne Punkte ohne großen Aufwand zu ändern. Das Tool funktioniert aber nur, wenn die ncurses-Bibliotheken installiert sind.

**make xconfig** Dieses Tool ist sehr anspruchsvoll. Es läuft nur unter X und erwartet auch, daß Tcl/Tk installiert ist. Dafür erfolgt hier die Konfiguration sehr angenehm und mit Mausklicks.

Keines der Tools bewahrt Sie davor Lesen zu müssen, was die einzelnen Punkte bedeuten. Ich werde hier in diesem Skript nicht auf die einzelnen Elemente des Kernels eingehen. Dazu konsultieren Sie bitte das Kernel-HOWTO.

**make oldconfig** Falls vorher schonmal eine gültige Konfigurationsdatei erzeugt wurde, kann man mit diesem Tool automatisch die alten Einstellungen benutzt werden und nur bei neu hinzugekommenen Einstellungen wird der Nutzer wie in **make config** gefragt.

**make cloneconfig** Mit diesem Befehl ist es möglich aus dem laufenden Kernel eine Konfigurationsdatei zu erstellen und nur bei neu hinzugekommenen Einstellungen wird der Nutzer wie in **make config** gefragt. Dies ermöglicht den aktuellen Kernel als Vorlage zu nehmen und dann mit den anderen Konfigurationsbefehlen (z.B. **make menuconfig**) diese Einzuladen und nur die benötigten Sachen zu ändern.

### 14.6.3 Aufräumen, Kompilieren, Installieren

#### Aufräumen

Vor dem eigentlichen Kompilieren müssen noch die wechselseitigen Abhängigkeiten der Quell- und Include-Dateien festgestellt werden. Der Befehl

```
make dep
```

erledigt diese Aufgaben. Danach sollten Sie die alten Objektdateien der letzten Kompilierungsrunde auch noch löschen.

```
make clean
```

Sie können die Befehle auch gemeinsam eingeben. Das doppelte kaufmännische Und bewirkt, daß `make clean` nur dann ausgeführt wird, wenn `make dep` erfolgreich ausgeführt wurde.

```
make dep && make clean
```

Hier noch einmal ein Auszug aus dem Makefile zu den beiden oben genannten Befehlen.

```
dep-files: scripts/mkdep archdep include/linux/version.h
    scripts/mkdep -- init/*.c > .depend
    scripts/mkdep -- 'find $(FINDHPATH) -name SCCS -prune -o -follow -name \
$(MAKE) $(patsubst %, _sfdep_%, $(SUBDIRS))
    $(MAKE) update-modverfile
endif

ifdef CONFIG_MODVERSIONS
MODVERFILE := $(TOPDIR)/include/linux/modversions.h
else
MODVERFILE :=
endif
export MODVERFILE

depend dep: dep-files

clean: archclean
    find . \( -name '*.oas' -o -name core -o -name '.*.flags' \) -type f - \
    | grep -v lxdialog/ | xargs rm -f
    rm -f $(CLEAN_FILES)
    rm -rf $(CLEAN_DIRS)
    $(MAKE) -C Documentation/DocBook clean
```

#### Kernel kompilieren

Und jetzt kommen wir zum eigentlichen Kompilieren. Der Befehl

```
make zImage
```

startet das Kompilieren des Kernels.

Soll der Kernel gleich auf einer Diskette untergebracht werden, so können Sie den Befehl

```
make zImage
```

verwenden. Dabei müssen Sie natürlich darauf achten, daß der Kernel nicht zu groß für die Diskette wird.

Apropos großer Kernel: Ab einer gewissen Größe funktioniert eine Kompilieren mit `make zImage` nicht mehr. Hier kommt der Befehl

```
make bzImage
```

zum Einsatz, der größere Kernels erlaubt.

## Kernel installieren

Nach der Installation finden Sie den Kernel im Verzeichnis `/usr/src/linux/arch/i386/boot/`. Die fertige Datei heißt `zImage`. Diese muß in das Verzeichnis `/boot` kopiert werden und eventuell in `vmlinuz` umbenannt werden. Sicherheitshalber sollte man den alten Kernel nicht einfach löschen sondern umbenennen. Danach muß auf jeden Fall der Lilo neu geschrieben werden.

```
enterprise:/ # mv /boot/vmlinuz /boot/vmlinuz.old
enterprise:/ # mv /usr/src/linux/arch/i386/boot/zImage /boot/vmlinuz
enterprise:/ # lilo
```

Wer auf Nummer sicher gehen will, bindet beide Kernels als Bootimages in den Bootloader ein. So kann, falls der neue Kernel nicht funktioniert, mit dem alten Kernel ohne weitere Probleme gebootet werden.

## Module

Enthält der Kernel Module, was sehr wahrscheinlich ist, so müssen auch diese erzeugt und installiert werden. Das Erstellen übernimmt der Befehl

```
make modules
```

und mit dem Befehl

```
make modules_install
```

werden die Moduldateien in das nach der Kernelversion benannte Verzeichnis in `/lib/modules` installiert.

## Kernelkompilierung mit SuSE

Der Kernel der SuSE-Distribution ist nicht identisch mit dem von Linus Thorvald freigegebenen Kernel. SuSE hat Elemente des Kernels weiterentwickelt und an die Distribution angepaßt. Auf der Distribution sind daher auch zwei Pakete mit dem Kernel Quellcode vorhanden. Das Paket *kernel-source* enthält die SuSE-Linux-Kernelquellen und das Paket *linux* enthält den original Linux-Kernel Quellcode.

Die Konfigurationsdatei für den vom System bei der Installation installierten Kernel finden Sie im Verzeichnis `/boot` in der Datei `vmlinuz.config`. Sie können diese Datei z. B. beim Befehl `make menuconfig` importieren und damit die von SuSE vorgegebenen Einstellung als Grundlage für Ihren eigenen Kernel benutzen.

Den neuen SuSE-Kernel können Sie vom SuSE-FTP-Server herunterladen. Sie finden ihn unter der Rubrik `update`.

```
ftp://ftp.suse.com/pub/suse/i386/update/Distributionsnummer/kernel/kernelversion
```

Es ist aber anzuraten nicht vom Original-SuSE-Server die Dateien herunterzuladen, sondern einen auf der SuSE-Seite angegebenen Spiegelserver zu verwenden. Diese sind meist weniger belastet als der Originalserver und das spielt bei einer Downloadgröße von rund 30 MByte schon eine wichtige Rolle.

Das Paket wird mit dem RPM-Manager ausgepackt und erzeugt ein Verzeichnis mit der Versionsnummer und den Link `/usr/src/linux` auf dieses Verzeichnis.

```
enterprise:~/download # rpm -ivh kernel-source-2.4.16.SuSE-31.i386.rpm
kernel-source          #####
enterprise:~/download # cd /usr/src
enterprise:/usr/src # ls -l linux*
lrwxrwxrwx    1 root    root                17 Mär 26 21:35 linux -> linux-2.4.16.SuSE
```



```

drwxr-xr-x   15 root    root          530 Mär 26 21:36 linux-2.4.16.SuSE
enterprise:/usr/src # cd linux
enterprise:/usr/src/linux # make menuconfig
enterprise:/usr/src/linux # ls -l .config
-rw-r--r--    1 root    root          38008 Mär 26 22:12 .config
enterprise:/usr/src/linux # make dep && make clean
.
. (Eine Menge Meldungen)
.
enterprise:/usr/src/linux # make bzImage && make modules && make modules_install
.
. (Eine Menge Meldungen)
.
enterprise:/usr/src/linux # ls -ld /lib/modules/2.4.16-4GB
drwxr-xr-x    5 root    root          313 Mär 26 22:37 /lib/modules/2.4.16-4GB
enterprise:/usr/src/linux # cd /boot
enterprise:/boot # cp /usr/src/linux/arch/i386/boot/bzImage vmlinuz-2.4.16
enterprise:/boot # vi /etc/lilo.conf
enterprise:/boot # lilo

```

Mit dem vi wurde in die /etc/lilo.conf folgender Block eingefügt:

```

image  = /boot/vmlinuz-2.4.16
label  = linux
root   = /dev/hdb7
initrd = /boot/initrd
append = "enableapic vga=0x0317"

```

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## XVII

Arbeitsblatt  
Programminstallation und Kernelkompilierung

## 17.1

- 1 Welche Bibliotheken benötigt das Programm `passwd`?
- 2 Bei der Standardinstallation der SuSE-Distribution wird Java zumeist nur in einer älteren Version installiert. Laden Sie aus dem Internet (<http://www.javasoft.com>) oder Intranet die neueste Version des Java-Development-Kits als RPM-Paket herunter und installieren Sie das Paket. Zum Entpacken der Datei lesen Sie die Instruktionen auf der Webseite.
- 3 Für die Ausführung der Programme ist die "virtuelle Kaffeemaschine" `java` verantwortlich. Sie befindet sich im Unterverzeichnis `bin` des Java-Verzeichnisses. Überprüfen Sie, ob nun die richtige Java-Version läuft. Führen Sie als *normaler* Benutzer den Befehl `java -version` aus. Welche Information bekommen Sie?
- 4 Die Version ist eindeutig nicht richtig. Woran kann es liegen?
- 5 Wo befindet sich die Datei, die durch die Eingabe von `java` gestartet wurde.
- 6 Aus welchem Paket stammt Sie?
- 7 Finden Sie heraus, wohin die Datei `java` des RPM-Paket installiert worden ist.
- 8 Damit nun der "richtige" richtige Java-Interpreter aufgerufen werden kann, muß der Soft-Link `/usr/lib/linux` auf das richtige Paket umgelenkt werden.
- 9 Lassen Sie sich eine alphabetisch sortierte Liste aller installierten Pakete ausgeben.
- 10 Wieviele Pakete sind installiert?
- 11 Welchen Kernel verwendet Ihr System?
- 12 Lassen Sie sich eine Liste aller geladenen Module ausgeben.
- 13 Laden Sie die RPM-Pakete des aktuellen SuSE-Kernels aus dem Internet bzw. Intranet oder installieren Sie sie von der Distributions-CD/DVD.
- 14 Erstellen Sie eine neue Konfiguration auf Basis des aktuellen Kernels und optimieren Sie den Kernel für Ihren Prozessortyp.
- 15 Tragen Sie im `Makefile` bei `EXTRAVERSION` den Prozessornamen ein. Kompilieren Sie auf der Basis der neuen Konfiguration einen neuen Kernel und dessen Module.
- 16 Installieren Sie den Kernel als `vmlinuz-VERSION-PATCHLEVEL-SUBLEVEL-EXTRAVERSION` im Verzeichnis `/boot` sowie die Module ins entsprechende Verzeichnis.
- 17 Tragen Sie für den neuen Kernel einen neuen Eintrag im Bootloader ein.
- 18 Testen Sie das neue System und vergleichen Sie es mit dem alten System.
- 19 Welche Dateisysteme unterstützt der aktuelle SuSE-Kernel als Modul oder fest integriert? Beschreiben Sie mit einem Satz das Einsatzgebiet des Dateisystems.

.....  
*Notizen:*

.....

# Kapitel 15

## Shell-Scripting

*Wir hätten da gerne mal ein Problem.*

*Florian Labs, LPI-Kurs 25.06.2004*

Es ist wichtig für einen Administrator sich mit dem Shell-Scripting auszukennen. Shell-Skripte erlauben eine automatisierte Ausführung von häufig vorkommenden Vorgängen. So können Prozeduren, die umständlich und langwierig sind, in der Ausführung vereinfacht werden.

Wie solche Skripte programmiert werden, hängt von der verwendeten Shell ab. In diesem Kapitel wird das Skripting am Beispiel der Bash erklärt. Die Grundprinzipien sind aber für alle Shells bzw. für die meisten anderen Programmiersprachen gleich. Wer schon in einer anderen Programmiersprache programmiert hat, wird sicherlich keine Probleme mit diesem Kapitel haben. Es gibt einige Eigentümlichkeiten, an die man sich aber schnell gewöhnt.

Auf jeden Fall werden Sie in diesem Kapitel nicht programmieren lernen. Das erfordert ein wesentlich längeres Studium der Materie, als was in diesen wenigen Seiten abgehandelt werden könnte. Es wird Ihnen hier nur die Syntax, die Struktur und die Regeln, der Skript-Programmierung mit der Bash gezeigt. Programmieren umfaßt eine ganze Menge mehr.

### 15.1 Variablen, Aliase und Funktionen

Den Anfang bilden die Speichermöglichkeiten der Shell. In den Variablen können Werte gespeichert werden. Aliase und Funktionen werden dazu benutzt um einzelne oder mehrere Befehle zu speichern.

#### 15.1.1 Variablen

Auf den Einsatz von Shell-Variablen bin ich schon in einem früheren Kapitel eingegangen. Trotzdem möchte ich an dieser Stelle noch einmal auf ihre Verwendung eingehen.

In Variablen können Werte gespeichert werden. Diese Variablen behalten ihren Wert aber nur temporär. Nach einem Logout oder spätestens nach einem Neustart ist der Wert nicht mehr existent. Umgebungsvariablen müssen daher bei jedem Start oder bei jedem Einloggen neu initialisiert werden. Dies erfolgt in den Startskripten. Das Setzen einer Variablen ist einfach. Geben Sie nur den Namen der Variablen gefolgt von dem Gleichheitszeichen und dem Wert an.

```
ole@defiant:~> sinn=42
ole@defiant:~> nachricht="Hallo Welt"
```

Übrigens entspricht das Gleichheitszeichen hier nicht dem Zeichen, was Sie aus der Mathematik kennen. Es bedeutet nicht, daß die Variable gleich dem Wert ist, sondern daß der der Variablen der Wert zugewiesen wird. Es handelt sich also um einen sogenannten Zuweisungsoperator.

Was ist aber ein Operator? Ein Operator ist einfach ausgedrückt eine Regel, wie zwei Werte miteinander verknüpft werden sollen. Einfache Operatoren sind z. B. Addition (+), Subtraktion (-), Multiplikation (\*) und Division (/).

Achten Sie bitte darauf, daß keine Leerzeichen zwischen der Variablen, dem Gleichheitszeichen und dem Wert steht. Sollte ein Leerzeichen im Wert stehen, so muß der Wert in einfache oder doppelte Anführungszeichen gesetzt werden.

```
ole@defiant:~> sinn = 42
bash: sinn: command not found
```

Wenn Sie mit dem Wert der Variablen arbeiten wollen, dann stellen Sie ein Dollarzeichen (\$) vor dem Variablennamen. Dann können Sie sich z. B. mit dem Befehl `echo` (4.6.5) den Inhalt auf dem Bildschirm ausgeben lassen.

```
ole@defiant:~> echo $sinn
42
ole@defiant:~> echo $nachricht
Hallo Welt
```

Allerdings sind die beiden Variablen nur in dieser Shell lokal vorhanden. Um Sie auch Kindershells und anderen Programmen zugänglich zu machen, müssen Sie exportiert werden. Dies geschieht mit Hilfe des Programms `export` (5.2.2). Das folgende Beispiel zeigt den Unterschied.

```
ole@defiant:~> bash
ole@defiant:~> echo $sinn $nachricht

ole@defiant:~>
exit
ole@defiant:~> export sinn
ole@defiant:~> export nachricht
ole@defiant:~> bash
ole@defiant:~> echo $sinn $nachricht
42 Hallo Welt
ole@defiant:~> exit
exit
```

Mit dem Kommando `bash` wird eine neue Shell, in dem Fall die Bash, gestartet. Die Variablen `sinn` und `nachricht` haben keinen Inhalt in dieser Shell. Mit `exit` wird die neue Shell wieder beendet. In der alten Shell werden dann die Variablen exportiert. Jetzt stehen die Variablen auch in der neuen Shell, die aus der alten Shell gestartet wurde, zur Verfügung.

### 15.1.2 Aliase

Auch auf Aliase wurde in diesem Skript schon einmal eingegangen. Trotzdem auch hier noch eine kurze Wiederholung.

Aliase ermöglichen es neue Befehle zu definieren oder einem alten Befehl eine neue Bedeutung zu geben. Verantwortlich dafür ist der Befehl `alias` (5.2.8).

```
ole@defiant:~> alias more=less
ole@defiant:~> alias ls
alias ls='ls $LS_OPTIONS'
ole@defiant:~> echo $LS_OPTIONS
-N --color=tty -T 0
```

Der erste Befehl sorgt dafür, daß beim Aufruf von `more` das Programm `less` gestartet wird. Bei SuSE sind einige Befehle auch durch Aliase verändert worden. Dies kann man schön beim Befehl `ls` sehen, wo noch zusätzlich eine Umgebungsvariable ins Spiel kommt.

Bei der Definition komplexerer Befehle sollte der Ausdruck in Anführungszeichen gesetzt werden. So ergibt das Semikolon im unteren Beispiel nicht das erwartete Ergebnis.

```
ole@defiant:~> alias heute=date;cal
      November 2002
So Mo Di Mi Do Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

ole@defiant:~> alias heute="date;cal"
ole@defiant:~> heute
Mit Nov 20 11:10:40 CET 2002
      November 2002
So Mo Di Mi Do Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

### 15.1.3 Funktionen

Ergänzend zu den Aliase können Sie mit Funktionen neue Befehle definieren. Während Sie mit Aliase nur einfache Befehle definieren, sind Funktionen für komplexere Aufgaben, die normalerweise aus mehreren Befehlen bestehen, ausgelegt.

Die Syntax für die Erstellung einer Funktion lautet:

```
[function] NAME () { KOMMANDOLISTE; }
```

Diese Deklaration definiert eine Funktion `NAME`. Das Wort `function` ist nicht notwendig. Sollten Sie das Schlüsselwort weglassen, sind die runden Klammern nach dem Funktionsnamen Pflicht. In den geschweiften Klammern werden dann die Kommandos entweder durch Semikola oder durch das neue Zeile Zeichen getrennt.

```
ole@defiant:~> werda () { echo Du bist $USER; who; }
ole@defiant:~> werda
Du bist ole
ole      :0      Nov 20 10:03 (console)
ole      pts/0    Nov 20 10:04
tapico   pts/1    Nov 20 10:14
walter   pts/2    Nov 20 10:53
```

Wichtig ist es hier, daß Sie nicht das letzte Semikolon vergessen. Auch der letzte Befehl vor der schließenden geschweiften Klammer muß mit einem Semikolon beendet werden.

Das Schreiben der Befehle in einer Zeile ist bei längeren Befehlsfolgen etwas unpraktisch. Sie können anstatt eines Semikolon auch einen Zeilenumbruch machen, um die Kommandoliste einzugeben.

```
ole@defiant:~> function hallo () {
> echo Hallo $USER
```

```
> date +"Es ist %H:%M:%S Uhr"
> }
ole@defiant:~> hallo
Hallo ole
Es ist 11:49:26 Uhr
```

Bis jetzt wurden die Funktionen wie Aliase verwendet. Im Gegensatz zu diesen können Funktionen Parameter mitgegeben werden. Diese Parameter stehen dann in den Variablen \$1, \$2, \$3 bis \$N. Die Anzahl der übergebenden Parameter steht in der Variablen \$#.

```
ole@defiant:~> function sagmal () {
> echo Es wurden $# Parameter eingegeben.
> echo Der erste Parameter lautet: $1
> }
ole@defiant:~> sagmal Hallo Welt
Es wurden 2 Parameter eingegeben.
Der erste Parameter lautet: Hallo
ole@defiant:~> sagmal "Hallo Welt"
Es wurden 1 Parameter eingegeben.
Der erste Parameter lautet: Hallo Welt
```

Sie können im Beispiel deutlich die Auswirkungen der Anführungszeichen sehen.

```
ole@defiant:~> function werist () {
> finger $1
> ps -aux | grep $1
> }
ole@defiant:~> werist root
Login: root                                Name: root
Directory: /root                          Shell: /bin/bash
Last login Wed Oct  9 08:13 (CEST) on tty1
New mail received Thu Oct 10 11:20 2002 (CEST)
    Unread since Fri May  3 18:15 2002 (CEST)
No Plan.
root      1  0.0  0.0   448   64 ?        S    08:14   0:04 init [5]
root      2  0.0  0.0     0    0 ?        SW   08:14   0:00 [keventd]
root      3  0.0  0.0     0    0 ?        SW   08:14   0:00 [kapmd]
root      4  0.0  0.0     0    0 ?        SWN  08:14   0:00 [ksoftirqd_CPU0]
root      5  0.0  0.0     0    0 ?        SW   08:14   0:00 [kswapd]
...
```

## 15.2 Skripte

So schön Aliase und Funktionen auch sind. Sie haben einen Nachteil. Sie müssen vor der Benutzung erst initialisiert werden. Damit Sie immer zur Verfügung stehen, müssen Sie in den Start- bzw. Login-Dateien definiert werden. Auch sind Sie für größere Projekte nicht so geeignet. Die Kommandos können auch in normalen Textdateien abgelegt werden. Andere Namen für Shell-Skripte in Dateien sind Batch-Dateien oder Stapelverarbeitungsdateien.

Als Beispiel nehmen wir mal die Funktion `werist` aus dem obigen Abschnitt und schreiben die Kommandos in die Datei `werist`. Textabschnitte, die mit einem Schweinegatter `#` beginnen, werden bis zum Ende der Zeile ignoriert. Es handelt sich dabei um Kommentare.

### Listing 15.1 *werist*

```
1: # werda - Zeigt Daten über den Benutzer und seine Prozesse an
```



```
2: #          werda BENUTZERNAME
3:
4: finger $1
5: ps -aux | grep $1
```

### 15.2.1 Ausführen von Skripten

Um Skripte auszuführen gibt es mehrere Methoden. Als erste Methode kann der Bash-Befehl `source` verwendet werden. Er liest die angegebene Datei und führt die in ihr enthaltenen Kommandos aus.

```
ole@defiant:~/skripte> source ./werist ole
```

Die Angabe `./werist` sagt, daß die Datei in dem aktuellen Arbeitsverzeichnis ist. Um dies zu umgehen, können Sie das aktuelle Verzeichnis, dargestellt durch den Punkt ".", in den Pfad einfügen.

```
ole@defiant:~/skripte> echo $PATH
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnome/bin:/opt/kde3/bin:
/opt/kde2/bin:/usr/lib/java/bin:/opt/gnome/bin:.
```

Achten Sie bitte darauf, daß das aktuelle Verzeichnis am Ende der Liste steht. Für den Systemadministrator `root` tragen Sie bitte **nie** das aktuelle Verzeichnis in den Pfad ein.

Es gibt auch eine kürzere Variante um ein Skript zu starten. Verwenden Sie einfach anstatt von `source` den Punkt.

```
ole@defiant:~/skripte> . ./werist ole
```

Sie können das Skript auch starten, indem Sie eine neue Instanz der Bash aufrufen.

```
ole@defiant:~/skripte> /bin/bash ./werist ole
```

Diese Methode verhält sich aber anders als die vorher vorgestellten Methoden. Es wird explizit eine neue Shell gestartet, in der dann das Skript ausgeführt wird. Das hat natürlich zur Folge, daß Variablen, die nicht exportiert worden sind, in dieser Shell nicht zur Verfügung stehen.

#### Ausführbare Datei

Sie kommen aber auch ohne einen zusätzlichen Befehl aus. Machen Sie einfach aus der normalen Textdatei eine ausführbare Datei. Unter Linux ist alles ausführbar, wenn es einen Inhalt besitzt, der durch den Prozessor (nativer Code) oder durch ein anderes Programm (interpretierter Code), wie z. B. die Shell, ausgeführt werden kann.

Um eine Datei ausführbar zu machen, müssen Sie das X-Recht setzen.

```
ole@defiant:~/skripte> ls -l werist
-rw-r--r--  1 ole      users          114 Nov 20 12:48 werist
ole@defiant:~/skripte> chmod a+x werist
ole@defiant:~/skripte> ls -l werist
-rwxr-xr-x  1 ole      users          114 Nov 20 12:48 werist
```

Jetzt können alle (Besitzer, Gruppe und der Rest der Welt) dieses Skript ausführen, indem Sie einfach den Namen eingeben.

```
ole@defiant:~/skripte> werist ole
Login: ole                      Name: Ole Vanhoefer
Directory: /home/ole           Shell: /bin/bash
```

She-Bang	Shell
#!/bin/sh	Die Bourne Shell
#!/bin/bash	Die Bourne Again Shell
#!/bin/ksh	Die Korn-Shell
#!/bin/csh	Die C-Shell
#!/bin/tcsh	Die erweiterte C-Shell
#!/bin/sed	Der Streaming-Editor
#!/usr/bin/awk	Die Programmiersprache awk
#!/usr/bin/perl	Die Programmiersprache Perl

Tabelle 15.1: Eine Auswahl von häufigen Interpretern

```

On since Wed Nov 20 10:03 (CET) on :0, idle 200 days 20:28, from console
On since Wed Nov 20 10:04 (CET) on pts/0, idle 5:09
On since Wed Nov 20 10:14 (CET) on pts/1 (messages off)
On since Wed Nov 20 10:53 (CET) on pts/2, idle 2:38 (messages off)
On since Wed Nov 20 12:45 (CET) on pts/3, idle 2:19 (messages off)
New mail received Thu Oct 10 11:20 2002 (CEST)
    Unread since Mon Jul 15 21:28 2002 (CEST)
No Plan.
root      2049  0.0  1.4  3408 1776 ?      S   10:03   0:00 /usr/X11R6/bin/xconsole -notify
ole       2058  0.0  0.0   2560    0 ?      SW  10:03   0:00 /bin/sh /usr/X11R6/bin/kde
ole       2106  0.0  0.7 19560  924 ?      S   10:03   0:00 kdeinit: Running...
ole       2109  0.0  1.2 19544 1544 ?      S   10:03   0:00 kdeinit: dcopserver --nosid
ole       2112  0.0  2.2 21824 2788 ?      S   10:03   0:00 kdeinit: klauncher
...

```

## She-Bang!

Es gibt einen ganzen Haufen von Skripten auf einem Linux-System und es gibt viele verschiedene Sprachen, in denen sie geschrieben werden können. Wird der Befehl `source`, der Punkt `."` oder eine ausführbare Datei benutzt, dann werden die Kommandos des Skripts in der aktuellen Shell ausgeführt. Bei der Benutzung einer separaten Shell für die Ausführung des Skripts (`/bin/bash meinSkript`) kann die Umgebung selbst gewählt werden.

Es ist ohne Frage von Vorteil, wenn das Skript selber die Information enthalten würde, durch welchen Interpreter die enthaltenen Kommandos ausgeführt werden sollen. In der Bash ist dies durch die Zeichen `"#"` realisiert, die am Anfang der ersten Zeile stehen. Diese Konstruktion wird umgangssprachlich *She-Bang* genannt. Diese Wortschöpfung setzt sich aus den Bezeichnungen *sheepgate* für das Doppelkreuz `"#"` und *bang* für das Ausrufungszeichen zusammen.

So beginnt ein Skript, daß für die Bash geschrieben wurde, mit der folgenden Zeile.

```
#!/bin/bash
```

Die Bash untersucht die erste Zeile des Skripts, startet den gefundenen Interpreter und übergibt das Skript an diesen Interpreter zur Ausführung.

*Ein falsche She-Bang-Anweisung ist ein häufiger Grund für eine fehlerhafte Ausführung des Skripts. In diesem Fall meldet die Bash und nicht der Interpreter einen Fehler.*

In dem im folgenden Beispiel gestarteten Skript wurde ein falscher Interpreter eingetragen. Die Fehlermeldung kommt von der Bash.

```

ole@enterprise:~/test> shebang
bash: ./shebang: bad interpreter: Datei oder Verzeichnis nicht gefunden

```

## Die Umgebung

Wenn Sie ein Skript in einer neuen Shell ausführen, dann wird eine komplett neue Instanz dieser Shell erzeugt. Aus der Eltern-Shell werden alle Variablen übernommen, die exportiert worden sind. Außerdem werden die Konfigurationsskripte der jeweiligen Shell ausgeführt, so daß eventuell Einstellungen überschrieben werden können. Wenn Ihr Skript von einer Variablen abhängt, dann sorgen Sie dafür, daß sie entweder in der Shell-Konfigurationsdatei gesetzt wurde oder als Umgebungsvariable exportiert wurde. Bitte machen Sie nicht beides zur gleichen Zeit.

Eine grundlegende Unix-Regel besagt, daß Kinder-Prozesse die Variablen von Ihrem Eltern-Prozeß erben. Die Variablen des Kinder-Prozeß sind aber nur während seiner Ausführung gültig und werden nicht zur Eltern-Shell zurückgegeben. Also haben Variablenänderungen in diesem Prozeß keine Auswirkungen auf die Variablen im Eltern-Prozeß.

Dies Verhalten läßt sich am folgenden Beispiel nachvollziehen.

```
ole@enterprise:~/test> cat shebang
#!/bin/bash

echo $var
var="Neuer Wert"
echo $var
ole@enterprise:~/test> var="Alt"
ole@enterprise:~/test> export var
ole@enterprise:~/test> shebang
Alt
Neuer Wert
ole@enterprise:~/test> echo $var
Alt
```

### 15.2.2 Rückgabewerte

Für Shell-Skripte ist es sehr wichtig festzustellen, ob die Kommandos ihre Aufgabe erfolgreich abgeschlossen haben. Die meisten Kommandos geben einen *Rückgabewert* oder *Fehlercode* zurück. Dieser Wert ist eine simple Ganzzahl und seine Bedeutung hängt von dem verwendeten Programm ab. Bei allen Programmen bedeutet die Zahl '0', daß der Befehl erfolgreich durchgeführt wurde. Eine von Null verschiedene Zahl hingegen deutet auf ein Problem hin.

Der Fehlercode wird in der Variablen '?' gespeichert. Da sie für jedes Kommando neu gesetzt wird, muß sie direkt nach der Kommandoausführung ausgelesen werden. Sie können dies direkt am Prompt nachvollziehen.

```
ole@enterprise:~/test> cd next
bash: cd: next: Datei oder Verzeichnis nicht gefunden
ole@enterprise:~/test> echo $?
1
ole@enterprise:~/test> echo $?
0
```

Bei der ersten Ausgabe wird die fehlerhafte Ausführung des Kommandos `cd` durch die Zahl 1 angezeigt. Die zweite Ausgabe der Variablen '?' bezieht sich auf den Erfolg des ersten `echo` Befehls.

Befehle können aus mehreren Gründen sich erfolglos beenden. Viele Befehle besitzen daher auch mehr als einen Fehlercode. Schauen Sie sich doch die Verhaltensweise von `grep` (7.7.1) einmal an.

```
ole@enterprise:~/test> grep bash shebang
#!/bin/bash
ole@enterprise:~/test> echo $?
0
ole@enterprise:~/test> grep tapico shebang
```

```
ole@enterprise:~/test> echo $?
1
ole@enterprise:~/test> grep bash sheebang
grep: sheebang: Datei oder Verzeichnis nicht gefunden
ole@enterprise:~/test> echo $?
2
```

Wenn **grep** einen oder mehrere Treffer erzielt, dann wird der Fehlercode 0 ausgegeben. Ist der Befehl soweit richtig, aber **grep** findet keine passende Zeichenkette, dann kommt der Fehlercode 1. Kommt es zu einem richtigen Fehler, wie z. B. einer nichtexistenten Datei, dann wird der Fehlercode 2 zurückgegeben.

## 15.3 Grundstrukturen

Nach all der langen Vorrede geht es nun mit den ersten Skripten los. Dabei geht es erst mal um ganz simple Dinge.

### 15.3.1 Auswerten

In unserem ersten Beispiel geht es um ein Skript mit dem Namen **addiere**, daß zwei als Parameter eingegebene Zahlen miteinander addiert.

#### Listing 15.2 *addiere*<sup>1</sup>

```
1: #!/bin/bash
2:
3: # Addiert zwei als Parameter angegebene Zahlen
4: #   addiere ZAHL1 ZAHL2
5:
6: # Berechnen
7: summe=$(( $1 + $2 ))
8:
9: # Ausgabe
10: echo "Die Summe aus $1 und $2 ist $summe."
```

Zeile 1 kennen Sie schon aus dem vorherigen Abschnitt. Der sogenannte She-Bang sorgt dafür, daß die folgenden Shell-Kommandos durch eine neue Instanz der Bash abgearbeitet werden. Das Schweinegatter ist in der Shell ein Kommentarzeichen. Alles von diesem Zeichen bis zum Ende der Zeile wird als Kommentar aufgefaßt. Skripte sollten Sie immer gut kommentieren, um auch später das Skript verstehen zu können.

In Zeile 7 werden die beiden Parameter 1 und 2 addiert und das Ergebnis in die Variable **summe** geschrieben. Dieser Vorgang wird als Arithmetische Ausdehnung (engl. *Arithmetic Expansion*) bezeichnet. Im Klartext heißt das nichts anderes, als daß der Ausdruck in den Klammern berechnet wird und das Ergebnis der Berechnung zurückgegeben wird. Das Format einer Arithmetischen Ausdehnung ist

```
$( (AUSDRUCK) )
```

Die Ausgabe des Ergebnis in Zeile 10 durch **echo** sollte Ihnen bekannt sein.

Das folgende Beispiel zeigt ein Skript, daß die Zeilen einer Datei durchnummeriert und das Ergebnis in einer HTML-Datei speichert.

#### Listing 15.3 *tohtml*

---

<sup>1</sup>Die Zahlen mit dem Doppelpunkt am Anfang der Zeile dienen nur zur Orientierung und sind nicht Bestandteil des Programms.

Operator	Bedeutung
<code>wert++</code>	Post-Inkrement der Variablen
<code>wert--</code>	Post-Dekrement der Variablen
<code>++wert</code>	Prä-Inkrement der Variablen
<code>--wert</code>	Prä-Dekrement der Variablen
<code>+</code> <code>-</code>	Plus und Minus
<code>!</code> <code>~</code>	Logische und Bitweise Negation
<code>*</code> <code>/</code> <code>%</code>	Mal, Teilen und Modulo
<code>&lt;&lt;</code> <code>&gt;&gt;</code>	linkes und rechtes bitweises Verschieben
<code>&lt;=</code> <code>&gt;=</code> <code>&lt;</code> <code>&gt;</code>	Vergleichsoperatoren
<code>==</code> <code>!=</code>	Gleich und Ungleich
<code>&amp;</code> <code>^</code> <code> </code>	Bitweises UND, exklusives ODER und ODER
<code>&amp;&amp;</code> <code>  </code>	Logisches UND und ODER
<code>AUSDR1?AUSDR2:AUSDR3</code>	Wenn Dann Sonst Ausdruck

Tabelle 15.2: Arithmetische Operatoren

```

1: #!/bin/bash
2:
3: # Gibt ein Skript als zeilennummerierte HTML-Datei zurück
4: #   tohtml DATEINAME
5:
6: # Name der Ausgabedatei
7: out="$1.html"
8:
9: # Lege den Anfang der HTML-Datei an
10: echo -e "<html>\n<head>\n<title>$1</title>\n</head>\n<body>\n" > $out
11: echo -e "<h1 align='center'>Listing: $1</h1>\n<pre>" >> $out
12:
13: # Datei durchnummerieren
14: expand $1 | nl -w 4 -b a -s ": " >> $out
15:
16: # Ende der HTML-Datei
17: echo -e "</pre>\n</body>\n</html>\n" >> $out

```

Einfach dem Skript die gewünschte Datei mitgeben und schon ist eine passende Datei entstanden.

```

ole@enterprise:~/test> tohtml addiere
ole@enterprise:~/test> ls -l addiere*
-rwxr-xr-x    1 ole      users          172 Nov 26 18:52 addiere
-rw-r--r--    1 ole      users          356 Nov 26 20:35 addiere.html

```

Natürlich ist dieses Beispiel sehr einfach und hat viele Schwachstellen. So würden die Spitzklammern in der Datei einfach in die HTML-Datei kopiert und dort dann vom Browser als HTML-Tags interpretiert. Also müssen vorher möglichst alle Spitzen-Klammern in HTML-Sonderzeichen umgewandelt werden. Dies erledigt der Streaming-Editor `sed`.

#### Listing 15.4 *tohtml Version 2*

```

1: #!/bin/bash
...
13: # Datei durchnummerieren und spitze Klammern behandeln
14: sed "s/</\&lt;/g" $1 | sed "s/>/\&gt;/g" \
15:   | expand | nl -w 4 -b a -s ": " >> $out
16:
17: # Ende der HTML-Datei
18: echo -e "</pre>\n</body>\n</html>\n" >> $out

```

### 15.3.2 test

Ein wichtiges Element der Programmierung ist die Möglichkeit sich je nach den gegebenen Fakten unterschiedlich verhalten zu können. Dazu müssen Entscheidungen gefällt werden, die auf wahren oder falschen Aussagen beruhen. Ein Instrument für die Ermittlung von wahren und falschen Aussagen ist der Befehl `test`.

```
test AUSDRUCK  
[ AUSDRUCK ]
```

Der angegebene Ausdruck wird ausgewertet und ein entsprechender Fehlercode 0 für wahr und 1 für falsch wird ausgegeben. Der untere Befehl ist eine Kurzform von `test`. Die öffnende eckige Klammer entspricht dem Befehl `test`. Die schließende eckige Klammer terminiert den Ausdruck. Daher müssen vor und hinter der eckigen Klammer Leerzeichen stehen.

#### Ausdrücke

Der Befehl gibt für die folgenden Ausdrücke wahr zurück, wenn die folgenden Aussagen zutreffen.

`AUSDRUCK` : Der Ausdruck ist wahr.

`! AUSDRUCK` : Der Ausdruck ist falsch.

`AUSDRUCK1 -o AUSDRUCK2` : Einer der beiden Ausdrücke oder beide sind wahr.

`AUSDRUCK1 -a AUSDRUCK2` : Beide Ausdrücke sind wahr.

`-n STRING` : Die Zeichenkette umfaßt mindestens ein Zeichen.

`-z STRING` : Die Zeichenkette enthält keine Zeichen.

`STRING1 = STRING2` : Beide Zeichenketten sind gleich.

`STRING1 != STRING2` : Die Zeichenketten sind unterschiedlich.

`INTEGER1 -eq INTEGER2` : Die beiden Ganzzahlen sind gleich.

`INTEGER1 -ne INTEGER2` : Die beiden Ganzzahlen sind nicht gleich.

`INTEGER1 -gt INTEGER2` : Die erste Zahl ist größer als die Zweite.

`INTEGER1 -lt INTEGER2` : Die erste Zahl ist kleiner als die Zweite.

`INTEGER1 -ge INTEGER2` : Die erste Zahl ist größer oder gleich der Zweiten.

`INTEGER1 -le INTEGER2` : Die erste Zahl ist kleiner oder gleich der Zweiten.

`-e DATEI` : Die Datei existiert.

`-s DATEI` : Die Datei existiert und ist größer als 0 Byte.

`-d DATEI` : Die Datei existiert und ist ein Verzeichnis.

`-f DATEI` : Die Datei existiert und ist eine normale Datei.

`-b DATEI` : Die Datei existiert und ist ein Block-Gerät.

`-c DATEI` : Die Datei existiert und ist ein Zeichen-Gerät.

`-L DATEI` : Die Datei existiert und ist ein symbolischer Link.

`-p DATEI` : Die Datei existiert und ist eine Pipeline-Datei (FIFO).

`-S DATEI` : Die Datei existiert und ist ein Socket.

`-r DATEI` : Die Datei existiert und ist lesbar.  
`-w DATEI` : Die Datei existiert und ist schreibbar.  
`-x DATEI` : Die Datei existiert und ist ausführbar.  
`-u DATEI` : Die Datei existiert und SUID ist gesetzt.  
`-g DATEI` : Die Datei existiert und SGID ist gesetzt.  
`-k DATEI` : Die Datei existiert und Sticky Bit ist gesetzt.  
`-0 DATEI` : Die Datei existiert und sein Besitzer ist gleich der effektiven UID.  
`-G DATEI` : Die Datei existiert und seine Gruppe ist gleich der effektiven GID.  
`DATE1 -ef DATEI2` : Beide Dateien besitzen die gleiche Geräte- oder Inode-Nummer.  
`DATE1 -ot DATEI2` : Die Datei 1 ist älter als die Datei 2.  
`DATE1 -nt DATEI2` : Die Datei 1 ist jünger als die Datei 2.

Mit den Shell-Logik-Operatoren `&&` und `||` können damit einfache Entscheidungen aufgebaut werden. Gerade bei der Ausführung von Skripten wird oft getestet, ob die Skripte überhaupt existieren.

Dieses Beispiel finden Sie in der `~/bashrc`. Die Datei `~/alias`, in der die Alias-Einträge eingetragen werden, wird nur ausgeführt, wenn Sie existiert und nicht leer ist.

```
test -s ~/.alias && . ~/.alias
```

## Beispiel

Ein Verzeichnis mit vier Dateien dient als Übungsgebiet.

```
ole@enterprise:~/test> ls -l
insgesamt 8
-rw-r--r--    1 ole      users      96 Nov 26 22:14 hebong
-rw-r--r--    1 ole      users         0 Nov 26 22:14 leer
drwxr-xr-x    2 ole      users      35 Nov 26 22:13 ordner
-rwxr-xr-x    1 ole      users      96 Nov 25 22:40 shebang
```

Handelt es sich um Verzeichnisse oder nicht. Der Fehlercode verrät es.

```
ole@enterprise:~/test> test -d ordner; echo $?
0
ole@enterprise:~/test> test -d hebong; echo $?
1
```

Der Fehlercode kann auch direkt ausgewertet werden. Die beiden Operatoren `&&` und `||` machen die Ausführung des zweiten Befehls vom Testergebnis abhängig.

```
ole@enterprise:~/test> test -e leer && echo "Die Datei existiert."
Die Datei existiert.
ole@enterprise:~/test> test -s leer && echo "Die Datei ist nicht leer."
ole@enterprise:~/test> test shebang -ot hebong && echo "Älter"
Älter
```

Das ganze geht auch mit Zahlen.

```
ole@enterprise:~/test> [ 42 -gt 12 ] && echo "Richtig"
Richtig
ole@enterprise:~/test> [ 42 -lt 12 ] || echo "Falsch"
Falsch
```

Bei der Kurzschreibweise sollten Sie auf keinen Fall die Leerzeichen vergessen.

```
ole@enterprise:~/test> [ -x shebang ] && echo "Ausführbar"
[: missing `']
```

Der Unterschied zwischen && und || ist deutlich sichtbar.

```
ole@enterprise:~/test> [ -x shebang ] && echo "Ausführbar"
Ausführbar
ole@enterprise:~/test> [ -x hebong ] && echo "Ausführbar"
ole@enterprise:~/test> [ -x hebong ] || echo "Nicht Ausführbar"
Nicht Ausführbar
```

### 15.3.3 if ... then ... elif ... then ... else ... fi

Mit dem Befehl if und seinen Unterbefehlen, kann ein Skript aufgrund der Fakten Entscheidungen treffen. Unser Skript tohtml hat noch ein paar Schwächen. So kommt es zu Fehlermeldungen von sed, wenn ein nichtvorhandene oder nicht lesbare Datei vorliegt. Deshalb soll in Zukunft vor der Ausführung überprüft werden, ob die angegebene Datei lesbar ist. Trifft dies nicht zu, wird eine Fehlermeldung ausgegeben.

#### Listing 15.5 tohtml Version 3

```
1: #!/bin/bash
2:
3: # Gibt ein Skript als zeilennummerierte HTML-Datei zurück
4: #   tohtml SKRIPTNAME
5:
6: # Existiert die Ausgabedatei ?
7: if [ -r $1 ]
8: then
9:   # Name der Ausgabedatei
10:  out="$1.html"
11:
12:  # Lege den Anfang der HTML-Datei an
13:  echo -e "<html>\n<head>\n<title>$1</title>\n</head>\n<body>\n" > $out
14:  echo -e "<h1 align='center'>Listing: $1</h1>\n<pre>" >> $out
15:
16:  # Datei durchnummerieren und spitze Klammern behandeln
17:  sed "s/</\&lt;/g" $1 | sed "s/>/\&gt;/g" \
18:  | expand | nl -w 4 -b a -s " ": " >> $out
19:
20:  # Ende der HTML-Datei
21:  echo -e "</pre>\n</body>\n</html>\n" >> $out
22: else
23:  # Existiert die Datei oder ist sie nur nicht lesbar?
24:  if [ -e $1 ]
25:  then
26:    # Fehlerausgabe auf Fehlerkanal
27:    echo "Die Datei $1 ist nicht lesbar." > /dev/stderr
28:    exit 1      # Beenden mit Fehlercode
29:  else
30:    # Fehlerausgabe auf Fehlerkanal
31:    echo "Die Datei $1 existiert nicht." > /dev/stderr
32:    exit 2      # Beenden mit Fehlercode
33:  fi
34: fi
```



Das Skript ist durch die Struktur `if ... else` in zwei Teile aufgeteilt. Der erste Teil wird ausgeführt, wenn die Datei lesbar ist. Wenn Sie nicht lesbar ist, wird der zweite Teil nach dem `else` ausgeführt.

Die Anweisungen nach dem `then` werden nur ausgeführt, wenn der Ausdruck hinter dem `if` wahr ist. Optional kann nach diesen Anweisungen noch ein `else` erscheinen, das einen zweiten Anweisungsblock einleitet. Diese Anweisungsblock wird nur dann ausgeführt, wenn der Ausdruck hinter dem `if` falsch ist.

Im zweiten Anweisungsblock wird noch unterschieden, ob die Datei nur nicht lesbar ist oder sie gar nicht existiert. Die Meldungen werden auf dem Fehlerausgabe ausgegeben, die normalerweise wie die Standardausgabe auf das ausführende Terminal zeigt. Danach wird das Skript mit unterschiedlichen Fehlercodes beendet. Dieses Verhalten können Sie im folgenden Beispiel sehen.

```
ole@enterprise:~/test> tohtml /etc/shadow ; echo $?
Die Datei /etc/shadow ist nicht lesbar.
1
ole@enterprise:~/test> tohtml blubb ; echo $?
Die Datei blubb existiert nicht.
2
ole@enterprise:~/test> tohtml shebang ; echo $?
0
ole@enterprise:~/test>
```

Weitere Entscheidungsblöcke mit eigenen Testausdrücken können vor dem `else` eingefügt werden, um noch feinere Entscheidungen treffen zu können. So liest das Skript, wenn kein Parameter eingegeben wurde, direkt von der Konsole seine Daten. Dies soll nicht geschehen, sondern in diesem Fall und wenn mehr als ein Parameter eingegeben wurde, soll das Skript einen Hilfetext ausgeben.

#### Listing 15.6 *tohtml Version 4*

```
1: #!/bin/bash
2:
3: # Gibt ein Skript als zeilennummerierte HTML-Datei zurück
4: #   tohtml SKRIPTNAME
5:
6: # Stimmt die Parameteranzahl nicht, dann Hilfetext
7: if [ $# -ne 1 ]
8: then
9:   echo "tohtml - Erzeugt von Dateien Listings im HTML-Format" > /dev/stderr
10:  echo "  Anwendung:" > /dev/stderr
11:  echo "      tohtml DATEINAME" > /dev/stderr
12:  exit 3          # Fehlercode ausgeben
13: # Existiert die Ausgabedatei ?
14: elif [ -r $1 ]
15: then
16:   # Name der Ausgabedatei
17:   ...
28:   echo -e "</pre>\n</body>\n</html>\n" >> $out
29: else
30:   # Existiert die Datei oder ist sie nur nicht lesbar?
31:   ...
40: fi
41: fi
```

Das Skript ist jetzt in drei Teile aufgeteilt. Der erste Teil wird ausgeführt, wenn mehr oder weniger als ein Parameter angegeben wurde. Es wird ein Hilfetext ausgegeben. Der zweite Teil wird nur ausgeführt, wenn ein Parameter angegeben wurde *und* der Parameter eine lesbare Datei bezeichnet. Trifft dies auch nicht zu, dann wird erst der letzte Teil nach dem `else` ausgeführt und die Fehlermeldungen erscheinen. Im Listing sind die Anweisungsblöcke der Teile Zwei und Drei nicht angegeben, da sie mit dem vorherigen Skript identisch sind.

Sie können in eine `if` Struktur so viele `elif ... then ...` Anweisungen einbauen, wie sie wollen.

```
ole@enterprise:~/test> tohtml
tohtml - Erzeugt von Dateien Listings im HTML-Format
Anwendung:
tohtml DATEINAME
```

Noch einmal übersichtlich dargestellt sieht die ganze Konstruktion so aus, wobei die `elif ... then` beliebig oft vorkommen können.

```
if AUSDRUCK1
then
...
elif AUSDRUCK2
then
...
elif AUSDRUCK3
then
...
else
...
fi
```

#### 15.3.4 case

Bei `case` wird ein Wert mit einer Reihe von Strings verglichen. Stimmen sie überein, so wird der entsprechende Abschnitt ausgeführt. Damit ist `case` ein Spezialfall von `if`. In den meisten Fällen wird `case` zur Ausführung von bestimmten Abschnitten eines Skripts verwendet. Insbesondere beim Starten, Neustarten und Stoppen von Diensten kommt es zum Einsatz. In den sogenannten RC-Skripten, die für das Starten und Stoppen von Dämonen und Serverdiensten verwendet werden, wird diese Konstruktion verwendet.

```
ole@enterprise:~> ls /sbin/rc*
/sbin/rcSuSEfirewall2 /sbin/rcgpm /sbin/rcnetwork /sbin/rcsyslog
/sbin/rcdhclient /sbin/rchotplug /sbin/rcportmap
ole@enterprise:~> ls /usr/sbin/rc*
/usr/sbin/rcalsasound /usr/sbin/rckdm /usr/sbin/rcsmbfs
/usr/sbin/rcapache /usr/sbin/rcksysguardd /usr/sbin/rcsmpppd
/usr/sbin/rcapid /usr/sbin/rclpd /usr/sbin/rcsnmpd
/usr/sbin/rcatd /usr/sbin/rcmysql /usr/sbin/rcsplash
/usr/sbin/rcautofs /usr/sbin/rcnfs /usr/sbin/rcsshd
/usr/sbin/rccron /usr/sbin/rcnfsserver /usr/sbin/rcxdm
/usr/sbin/rcfam /usr/sbin/rcnscd /usr/sbin/rcxfs
/usr/sbin/rcfbset /usr/sbin/rcpcscd /usr/sbin/rcxntpd
/usr/sbin/rci4l /usr/sbin/rcpersonal-firewall /usr/sbin/rcypbind
/usr/sbin/rci4l_hardware /usr/sbin/rcpowerfail /usr/sbin/rcyppasswdd
/usr/sbin/rcinetd /usr/sbin/rcrandom /usr/sbin/rcypserv
/usr/sbin/rcisdn /usr/sbin/rcraw /usr/sbin/rcypxfrd
/usr/sbin/rcjoystick /usr/sbin/rcsendmail
/usr/sbin/rckbd /usr/sbin/rcsingle
```

Das folgende Beispiel zeigt das Prinzip, das hinter dieser Idee steckt.

#### Listing 15.7 dienst

```
1: #!/bin/bash
2:
```

```

3: # Steuert einen Dienst. Beispiel für case ... in Struktur
4: #   dienst start|restart|stop
5:
6: # Nach dem Inhalt des Parameters $1 wird entschieden, was gemacht werden soll.
7: case $1 in
8:   start)
9:     # Der Dienst wird gestartet
10:    echo "Der Dienst wird gestartet."
11:    ;;
12:
13:   restart)
14:    # Der Dienst wird neu gestartet
15:    echo "Der Dienst wird neu gestartet."
16:    ;;
17:
18:   stop)
19:    # Der Dienst wird gestoppt
20:    echo "Der Dienst wird gestoppt."
21:    ;;
22:
23:   *)
24:    # Falsches Kommando
25:    echo "Falsches Kommando." > /dev/stderr
26:    echo "Syntax:" > /dev/stderr
27:    echo "  dienst start|restart|stop" > /dev/stderr
28:    exit 1      # Beenden mit Fehlercode
29:    ;;
30: esac

```

Jenachdem ob `start`, `restart` oder `stop` hinter dem Skriptnamen steht, wird ein anderer Abschnitt des Skripts ausgeführt. Sollte es zu keiner Übereinstimmung kommen, wird der Abschnitt mit dem Asterisk ausgeführt, der in diesem Fall einen Hilfetext enthält.

```

ole@enterprise:~/test> dienst start
Der Dienst wird gestartet.
ole@enterprise:~/test> dienst restart
Der Dienst wird neu gestartet.
ole@enterprise:~/test> dienst stop
Der Dienst wird gestoppt.
ole@enterprise:~/test> dienst
Falsches Kommando.
Syntax:
  dienst start|restart|stop

```

Hier eine praktische Anwendung um zwei VNC-Server und einen Viewer für eine Vorführung zu starten und zu beenden. Weitere Informationen über den VNC-Server und seine Konfiguration finden Sie in Abschnitt B.1.

#### Listing 15.8 *vnc*

```

1: #!/bin/sh
2: # Startet zwei VNC-Server (Master und Slave)
3:
4: # Auflösung festlegen
5: GEOM=950x680
6: CDEPTH=16
7:
8: # Variable auswerten

```

```

 9: case $1 in
10:   start)
11:     # Starten der VNC-Server
12:     # Master starten
13:     # Passworddatei ~/.vnc/privat
14:     vncserver :1 \
15:       -geometry $GEOM \
16:       -depth $CDEPTH \
17:       -alwaysshared \
18:       -name master \
19:       -rfbauth $HOME/.vnc/privat
20:
21:     # Slave starten
22:     # Passworddatei ~/.vnc/public
23:     vncserver :2 \
24:       -geometry $GEOM \
25:       -depth $CDEPTH \
26:       -alwaysshared \
27:       -name slave \
28:       -rfbauth $HOME/.vnc/public
29:
30:     # Viewer zur Kontrolle des Master-Servers starten
31:     # Falls in einem X-Terminal als anderer Benutzer gestartet zeigt
32:     # die Display-Variable auf das aktuelle X-Window
33:     # Der Server muß aber vorher mit
34:     # xhost localhost
35:     # freigegeben werden
36:     DISPLAY=:0.0
37:     export DISPLAY
38:     # Viewer starten
39:     vncviewer :1
40:     ;;
41:
42:   stop)
43:     # Beenden des VNC-Servers
44:     vncserver -kill :1
45:     vncserver -kill :2
46:     ;;
47:
48:   *)
49:     # Falsches Kommando
50:     echo "Syntax: vnc start|stop"
51:     exit 1
52:     ;;
53: esac
54:
55: # Ende

```

### 15.3.5 dialog

Um auch auf der Textkonsole ein Gefühl von graphischer Oberfläche zu schaffen, wurde der Befehl `dialog` geschaffen. Er erzeugt verschiedene Arten von Fenstern für Ein- und Ausgabe.

```

dialog --clear
dialog --create-rc DATEI
dialog [OTPIONEN] BOX

```

## Boxtypen

Es stehen die folgenden Boxvariationen zur Auswahl:

--msgbox TEXT HÖHE BREITE

Eine Box in der Größe HÖHE x Breite und einem TEXT wird dargestellt. Durch Betätigen von <RETURN> wird die Box geschlossen.

--yesno TEXT HÖHE BREITE

Eine Box in der Größe HÖHE x Breite und einem TEXT. Es besteht die Möglichkeit mit Ja und Nein zu antworten. Bei Ja wird ein Fehlercode von 0 und bei Nein ein Fehlercode von 1 zurückgegeben.

--infobox TEXT HÖHE BREITE

Die Infobox ist im Prinzip eine Messagebox. Nur muß die Nachricht nicht mit <RETURN> bestätigt werden. Sie bleibt so lange stehen, bis ein neuer Befehl kommt.

--inputbox TEXT HÖHE BREITE [VORGABE]

Die Inputbox erlaubt die Eingabe einer Zeichenkette. Die eingegebene Zeichenkette wird über die Standardfehlerausgabe zurückgegeben.

```
ole@enterprise:~/test> dialog --inputbox Eingabe 20 70 2> in.tmp
ole@enterprise:~/test> echo $(cat in.tmp)
Hallo
```

--textbox DATEI HÖHE BREITE

Die Textbox stellt den Inhalt einer Datei in einem Nachrichtenfenster dar. Im Gegensatz zum normalen Nachrichtenfenster kann hier durch den Text gescrollt werden. Dieses Fenster wird für größere Texte verwendet. Durch Betätigen von <RETURN> wird die Box geschlossen.

--menu TEXT HÖHE BREITE MENÜPUNKTE [PUNKT BEZEICHNUNG] ...

Eine Menübox zeigt eine Auswahl von Punkten an, aus denen der Benutzer einen Auswählen kann. Die Bezeichnung des ausgewählten Punkts wird über die Standardfehlerausgabe zurückgegeben. Ein Beispiel für die Menübox finden Sie in Listing 15.9.

## Eine Auswahlliste für Systembefehle

Die Idee hinter diesem Projekt ist einfach zu beschreiben. Dem Benutzer sollen gewisse Befehle erlaubt werden, ohne daß er Rechte auf andere Befehle bekommt. Hier hilft der Befehl `sudo` (9.3.1) normalerweise weiter. Um das ganze komfortabler zu gestalten, werden alle Befehle, die jemand können muß, in ein Skript geschrieben und können durch ein Auswahldialog ausgewählt werden. Damit das ganze dann auch noch mit `root`-Rechten läuft, wird es erlaubt das Skript mit `sudo` unter `root` zu starten.

Im ersten Schritt wird das Skript von `root` geschrieben und im Verzeichnis `/usr/bin` gespeichert. Die Rechte sind auf 700 gesetzt.

### Listing 15.9 *manager*

```
1: #!/bin/bash
2:
3: # Führt vordefinierte Befehle aus.
4: # Die Ausführung dieses Skripts unter sudo als root erlauben
5:
6: # Temporäre Datei für die Ergebnisse
7: tempdatei=/tmp/gwinternet.tmp.$$
8:
9: # Dialogfenster
10: dialog --menu "Wählen Sie den Dienst" 20 50 10 \
11:         0      "Swap aktivieren" \
12:         1      "Swap deaktivieren" \
13:         2      "Maillogdatei anzeigen" \
```

```
14:          3          "Rechner herunterfahren" \  
15:          4          "Rechner neu starten" \  
16:          2> $tempdatei  
17:  
18: eingabe=$(cat $tempdatei)  
19:  
20: #Auswerten  
21: echo "Sie haben ausgewählt: $eingabe"  
22:  
23: case $eingabe in  
24:   0) # Swap aktivieren  
25:     /sbin/swapon /swap  
26:     if [ $? -gt 0 ]  
27:     then  
28:       echo "Fehler bei Ausführung" > $tempdatei  
29:     else  
30:       /sbin/swapon -s > $tempdatei  
31:     fi  
32:     dialog --textbox $tempdatei 20 70  
33:     ;;  
34:  
35:   1) # Swap deaktivieren  
36:     /sbin/swapoff /swap  
37:     if [ $? -gt 0 ]  
38:     then  
39:       echo "Fehler bei Ausführung" > $tempdatei  
40:     else  
41:       /sbin/swapon -s > $tempdatei  
42:     fi  
43:     dialog --textbox $tempdatei 20 70  
44:     ;;  
45:  
46:   2) # /var/log/mail  
47:     dialog --textbox /var/log/mail 20 70  
48:     ;;  
49:  
50:   3) # Rechner runterfahren  
51:     # Aufräumen  
52:     rm -f $tempdatei  
53:     echo "Der Rechner wird heruntergefahren"  
54:     /sbin/shutdown -h now  
55:     ;;  
56:  
57:   4) # Rechner neu starten  
58:     # Aufräumen  
59:     rm -f $tempdatei  
60:     echo "Der Rechner wird neu gestartet"  
61:     /sbin/shutdown -r now  
62:     ;;  
63:  
64:   *) # Falscher Wert  
65:     echo "Falsche Eingabe!"  
66:     ;;  
67: esac  
68:  
69:  
70: # Aufräumen
```

```
71: rm -f $tempdatei
```

Im zweiten Schritt trägt *root* für den Benutzer mit *visudo* (9.3.3) in die */etc/sudoers* das Recht ein, daß Skript ausführen zu dürfen.

```
ole          ALL= NOPASSWD: /usr/bin/manager
```

Um das Ganze komfortabler zu machen, wird noch ein Alias angelegt.

```
alias manager="sudo /usr/bin/manager"
```

Jetzt kann der Benutzer *ole* das Skript mit Eingabe von *manager* starten und die angebotenen Befehle ausführen.

### 15.3.6 while ... do ... done

Oft müssen Aktionen wiederholt werden. Dazu dienen Schleifen. Eine einfache Schleife ist **while**. Solange der getestete Ausdruck wahr ist, solange wird ein Anweisungsblock ausgeführt.

```
while AUSDRUCK
do
    ....
done
```

Ein Beispiel ist das folgende Skript, daß die Adressen eines C-Netzes nach angeschlossenen durchsucht. Dabei wird die Host-Adresse so lange hochgezählt, bis der Endwert erreicht ist.

#### Listing 15.10 pingall

```
1: #!/bin/bash
2:
3: # Pingt die Rechner eines C-Klasse-Netzes an
4:
5: # Werte festlegen
6: netz='217.89.70.' # Fester Teil der IP-Nummer
7: start=1           # Startwert
8: ende=254          # Endwert
9:
10: # Schleife initialisieren
11: i=$start
12: # Testen, ob der Anweisungsblock noch einmal ausgeführt werden soll
13: while [ $i -le $ende ]
14: do
15:     adresse=$netz$i           # Adresse zusammensetzen
16:     echo -ne "\n$adresse"     # Adresse ausgeben
17:
18:     # Pingen und auf Erfolg testen
19:     ping -w 2 $adresse | grep ' 0% loss' &> /dev/null
20:     if [ $? -eq 0 ]
21:     then
22:         echo -n "    --- Rechner vorhanden ---"
23:     fi
24:
25:     # Host hochzählen
26:     i=$((i+1))
27: done
28:
29: echo -e "\n\nFertig"
```

### 15.3.7 until ... do ... done

Die Until-Schleife ist das Gegenteil von `while`. Während bei `while` die Schleife läuft, wenn der Ausdruck wahr ist, läuft `until` wenn der Ausdruck falsch ist.

```
until AUSDRUCK
do
    ....
done
```

### 15.3.8 for ... in ... do ... done

Diese Schleife arbeitet alle Elemente einer Liste ab. Jedes Element wird in die angegebene Variable gesteckt und dann mit diesem Wert der Anweisungsblock ausgeführt.

```
for VARIABLE in LISTE
do
    ...
done
```

#### Listing 15.11 *tolowercase*

```
1: #!/bin/bash
2:
3: # Benennt alle angegebenen Dateien im aktuellen Verzeich so um,
4: # daß sie klein geschrieben sind
5:
6: for datei in *
7: do
8:     mv $datei $(echo $datei | tr A-Z a-z) 2> /dev/null
9: done
```

### 15.3.9 read

Der Befehl `read` liest eine Zeile von der Standardeingabe und packt jedes Wort einzeln in eine Variable. Sind mehr Worte als Variablen vorhanden, dann landen die restlichen Worte in der letzten Variablen.

```
read VARIABLENLISTE
```

#### Listing 15.12 *saghallo*

```
1: #!/bin/bash
2:
3: # Test für die Eingabe
4: #     saghallo
5:
6: # Ausgabe
7: echo -n "Gegen Sie Vorname und Nachname ein > "
8:
9: # Einlesen
10: read vorname nachname
11:
12: # Ausgabe
13: echo "Guten Tag, $vorname $nachname."
14: echo "Oder darf ich $vorname sagen?"
```



## 15.4 Weitere Builtin-Befehle der Bash

### 15.4.1 shopt

Das Kommando `shopt` zeigt und ändert den Status der Variablen zur Steuerung der optionalen Funktionen der Bash.

`shopt [OPTIONEN] [OPTIONSVARIABLENLISTE]`

#### Optionen

<code>-p</code>	Zeigt den Status der Variablen an ( <i>print</i> )
<code>-s</code>	Aktiviert die Option(en) ( <i>set</i> )
<code>-u</code>	Deaktiviert die Option(en) ( <i>unset</i> )
<code>-q</code>	Gibt den Status einer Optione durch den Fehlercode zurück ( <i>quiet</i> )

Die Liste der Variablen und ihre Bedeutung können Sie der Manual-Page zur Bash (`man 1 bash`) entnehmen.

#### Beispiel

Der Befehl ohne Parameter zeigt den Status der Variablen an.

```
ole@enterprise:~> shopt
cdable_vars      off
cdspell          off
checkhash        off
checkwinsize     on
cmdhist          on
dotglob          off
execfail         off
expand_aliases  on
extglob          on
histreedit       off
histappend       off
histverify       off
hostcomplete     on
huponexit        off
interactive_comments  on
lithist          off
mailwarn         off
no_empty_cmd_completion off
nocaseglob       off
nullglob         off
progcomp         on
promptvars       on
restricted_shell  off
shift_verbose    off
sourcepath       on
xpg_echo         off
ole@enterprise:~> shopt sourcepath
sourcepath       on
```

Mit `-s` und `-u` werden die Variablen gesetzt und wieder abgeschaltet.

```
ole@enterprise:~> shopt -u sourcepath
ole@enterprise:~> shopt sourcepath
sourcepath       off
ole@enterprise:~> shopt -s sourcepath
```

```
ole@enterprise:~> shopt sourcepath
sourcepath      on
```

### 15.4.2 source

Der Befehl **source** dient zur Ausführung eines Scripts. Er liest die Kommandos aus einer Datei aus und führt dieser in der aktuellen Shell-Umgebung aus. Es wird als Fehler der Fehlercode des letzten ausgeführten Befehls genommen.

```
source DATEINAME [PARAMETER]
```

Enthält der angegebene Dateiname keinen Schrägstrich, so werden die Verzeichnisse aus der Variablen **PATH** nach der Datei durchsucht. Die Dateien müssen nicht ausführbar sein.

Sollte sich die Bash nicht im Posix-Modus befinden, so wird das aktuelle Verzeichnis durchsucht, wenn die Suche in den Pfad-Verzeichnissen zu keinem Erfolg geführt hat. Das Durchsuchen der Pfad-Variablen hängt ab von der Option "sourcepath" des **shopt** Befehls (15.4.1). Bei gesetzter Option (**shopt -s sourcepath**) wird der Pfad durchsucht.

## Kapitel 16

# Linux und Hardware

*Manchen Menschen glauben noch immer, daß ein Computer funktioniert. Ein Computer kann nicht funktionieren. Er täuscht es aber sehr gut vor.*

*Anonymous*

Aufgrund seiner offenen Quellen hat sich Linux auf vielen Hardwarearchitekturen etabliert. Vom kleinen PDA bis zum Großrechner reicht die Spanne der Hardwareplattformen. In den meisten Fällen wird aber Linux auf der von IBM und Intel begründeten x86-PC-Architektur laufen. Und ich gehe wohl recht in der Annahme, daß dies auch auf Sie, werter Leser, zutrifft.

Trotz modernster Prozessoren und Komponenten arbeiten wir heute im Prinzip mit einer Technik, die aus dem Jahre 1984 stammt. Wie damals besteht ein PC aus einem oder mehreren Prozessoren (CPUs), die über einen *Chipsatz* mit dem *Hauptspeicher* (RAM) und den Ein- und Ausgabeschnittstellen verbunden ist. Unter Ein- und Ausgabeschnittstellen fallen die Grafikkarte und die Controller für die Massenspeicher. Dazu gehören auch die modernen Kommunikationsschnittstellen wie USB, Firewire und Ethernet sowie die veralteten Schnittstellen Parallel- und Seriellport und die PS/2-Anschlüsse für Maus und Tastatur. Alle diese Komponenten befinden sich auf der *Hauptplatine*, die auch als *Mainbord* oder *Motherboard* bezeichnet wird. Eine ganze Reihe von Firmen bietet solche Hauptplatinen für verschiedene Prozessoren z.B. der Hersteller Intel, AMD oder VIA an. Daneben gibt es große Unterschiede in der Bauform je nachdem ob der Rechner als Server, Workstation oder als Bürorechner dient.

Ein großes Manko von Linux ist die teilweise schlechte Unterstützung mancher Hardwarehersteller. Deshalb ist es angeraten vor dem Kauf der Hardware zu überprüfen, welche Komponenten von Linux gut unterstützt werden oder welche Hersteller Treiber anbieten. Auf jeden Fall hat es sich bewährt nicht die neueste Hardware zu nehmen. Dies ist auch nur in den seltensten Fällen notwendig, da die leistungstärksten Komponenten oft ein schlechtes Preis-Leistungsverhältnis haben. Eine gute Orientierung bietet dabei das "*Linux Hardware Compatibility HOWTO*", das bei Ihrer Distribution mitgeliefert wird oder auf der Webseite des *Linux Documentation Projects*<sup>1</sup> zu finden ist.

### 16.1 Das BIOS

Das *BIOS*, die Abkürzung von *Basic Input Output System*, ist praktisch die einzige Software, die fest im Rechner integriert ist. Es ist in einem Festspeicher (*ROM*, *EEPROM* oder *Flashspeicher*) auf der Hauptplatine gespeichert und organisiert den Bootvorgang, der in Abschnitt 12.1.2 beschrieben wird.

Daneben fungierte das BIOS früher auch als Bindeglied zwischen Betriebssystem und Hardware. Die Software griff nicht direkt auf die Hardware zu sondern bediente sich der Funktionen des BIOS. Heutzutage bringen die Betriebssysteme eigene Routinen für den Hardwarezugriff mit und lassen das BIOS links liegen. Ein gutes Beispiel dafür ist das Handling von zu großen Platten. Es kann vorkommen, daß das BIOS die neue große Platte unterschätzt und viel zu wenig Speicherplatz anzeigt. Dies irritiert im ersten Moment etwas. Beginnt

---

<sup>1</sup><http://www.tldp.org/HOWTO/Hardware-HOWTO/>

man allerdings mit der Installation von Linux, so stellt man fest, daß im Installationsprogramm und später im laufenden Betriebssystem die Platte mit der richtigen Größe angezeigt wird.

Da Linux das BIOS links liegen läßt, ist es auch nicht notwendig in die Tiefen der BIOS-Konfiguration vorzudringen. Diesen Stoff überlasse ich gerne den Hardware-Büchern. Trotzdem sollten Sie einige Einstellungen im BIOS durchführen können. Gerade die Einstellungen für den Bootvorgang und damit für die Installation von Linux sind wichtig. Daneben müssen Sie in der Lage sein, die auf der Hauptplatine vorhandenen Komponenten ein- und ausschalten zu können. Die besten Treiber für eine Infrarotschnittstelle nützen nichts, wenn die Schnittstelle im BIOS deaktiviert ist. Genauso sollten Sie beim Einsatz einer zusätzlichen Grafikkarte darauf achten, daß die integrierte Grafikkarte auch deaktiviert ist. Sonst ist es nicht weiter verwunderlich, wenn der Monitor an der neuen Grafikkarte schwarz bleibt.

### 16.1.1 Konfiguration

Das BIOS besitzt normalerweise eine Oberfläche, die beim Starten des Systems aufgerufen werden kann. Die Art und Weise unterscheidet sich von Hersteller zu Hersteller und auch zwischen Notebook und PC. Wie das BIOS aufgerufen wird, steht meistens auf der Startseite. Dies kann z. B. die Taste <Entf> oder die Taste <F2> sein. Genau wie beim Start unterscheiden sich die BIOS-Varianten in der Oberfläche. Deshalb kann hier kein Kochrezept für die Konfiguration des BIOS gegeben werden. Aber mit ein bißchen Lesen – ich weiß, da war wieder das böse Wort – kann man sich eigentlich recht gut zurechtfinden.

#### Systemzeit

Eine der wichtigsten und einfachsten Einstellung ist die Systemzeit. Hier können Sie das Datum und die Uhrzeit der eingebauten Hardware-Uhr (CMOS-Uhr) einstellen. Sie sollten sich entscheiden, ob sie diese Uhr auf MEZ bzw. MESZ laufen lassen sollten oder die Weltzeit für Ihren Rechner nehmen. Sie müssen Linux dann nur mitteilen, in welcher Zeitzone sich der Rechner nun befindet.

#### Festplatten

Neben der Zeiteinstellung sollte das BIOS auch die Startplatte des Systems kennen, da es seine Aufgabe ist das im dortigen Bootsektor enthaltene Programm zu starten. Alle weiteren Festplatten brauchen im BIOS nicht eingetragen werden, da Linux das BIOS beim Zugriff auf die Festplatten umgeht. Im BIOS werden Daten zu den Festplatten eingetragen. Die erfolgt entweder manuell oder über eine automatische Erkennung der Festplatte. Die Geometrie der Platte wird nach dem BIOS über die Angaben *Zylinder*, *Köpfe* und *Sektoren* eingetragen. Diese Werte entsprechen aber nicht der Wirklichkeit, da bei modernen Platten die äußeren Spuren mehr Sektoren besitzen als die inneren Spuren. Daher werden moderne Platten nicht mehr über das oben beschriebene CHS-System sondern über LBA angesteuert. Trotzdem gibt es für jede Platte einen der Größe entsprechenden CHS-Eintrag um ein älteres BIOS oder ein Betriebssystem wie DOS nicht zu verärgern.

#### Zylinder-1024-Problem

Ein Problem, was mit alten BIOS-Versionen auftreten kann, ist das *Zylinder-1024-Problem*. Bei diesen BIOS-Versionen war die Anzahl der Zylinder auf 1024 (10 Bit) begrenzt. Das BIOS konnte also nur Platten bis 8 GB (1024 Zylinder mal 256 Köpfe mal 64 Sektoren à 512 Byte) vollständig adressieren. Bei größeren Platten konnten nur die ersten 8 GB adressiert werden. Dies hatte zur Folge, daß Linux unter Umständen nicht gebootet hat, wenn der Kernel über Zylinder 1024 lag und der Bootloader deshalb den Kernel nicht vollständig laden konnte. Sollte der Fall auftreten, kann er durch folgende Möglichkeiten gelöst werden.

- Sie schalten die Funktion *LBA* im BIOS ein, falls das BIOS es unterstützt.
- Sie benutzen eine andere kleinere Platte zum Booten.
- Sie legen die Root-Partition als erste ein und machen Sie maximal 8 GB groß.
- Sie legen das Verzeichnis `/boot` auf einer separaten Partition an. Die Partition, die als erste auf der Platte liegen sollte, braucht nur eine Größe von 20 bis 30 MB.

## Bootreihenfolge

Eine weitere wichtige Einstellung ist die Bootreihenfolge. Im BIOS legen Sie fest, welche Datenträger in welcher Reihenfolge nach einem funktionstüchtigen Bootsektor durchsucht werden sollen. So ist es durchaus möglich die Floppy oder das CD-ROM-Laufwerk vor der Festplatte einzutragen. Ist kein Medium in dem entsprechenden Laufwerk vorhanden oder das Medium besitzt keinen Bootloader, dann geht das BIOS zum nächsten Datenträger in der Liste weiter. Neben den bisher genannten Datenträgern sind auch Einstellungen wie SCSI oder USB je nach BIOS-Version möglich.

Für den praktischen Einsatz sollte als erstes in der Bootreihenfolge immer die Festplatte angegeben werden und das BIOS dann durch ein Kennwort geschützt werden. So kann sich niemand durch das einfache Einlegen einer Boot-Diskette, eine Rescue- oder Knoppix-CD zum Systemverwalter machen.

## Peripheriegeräte

Das BIOS verwaltet auch die auf der Hauptplatine eingebauten Peripheriegeräte und -schnittstellen. Heute ist es üblich viele häufig genutzte Komponenten gleich auf der Hauptplatine zu integrieren. Schnittstellen wie Seriell- und Parallelport, wie auch Sound-, Grafik- und Netzwerkadapter waren früher durch separate Steckkarten im Rechner realisiert worden. Heute sind sie nicht immer, aber immer öfter in die Hauptplatine integriert. Um z. B. eine externe Grafikkarte im AGP-Steckplatz betreiben zu können, ist es meistens notwendig die entsprechende interne Grafikkarte zu deaktivieren.

## 16.2 PC-Busarchitekturen

Ein PC-Bus ist im Prinzip nur ein System von parallelen Leitungen zur Übertragung von Daten zwischen den einzelnen Systemkomponenten. Er ermöglicht die Kommunikation zwischen den Prozessoren, dem Hauptspeicher, den Schnittstellen und den Erweiterungskarten. In modernen Systemen sind zwei Klassen von Bussen zu finden. Der System-Bus verbindet die CPU mit dem Hauptspeicher und dem L2-Cache. Daneben gibt es eine Reihe von I/O-Bussen, die eine Vielzahl von Peripheriegeräten mit der CPU verbindet. Die Verbindung erfolgt dabei über eine Bridge, die Bestandteil des Chipsatzes für den Prozessor ist.

In parallelen Bussystem wie PCI oder SCSI finden sie meistens:

**Datenbus** Er regelt die Datenübertragung und kann eine Datenwortbreite von 8, 16, 32 oder 64 Bit besitzen.

**Adreßbus** Dieser Bus ist für die Auswahl der Einzelgeräte und Adressierung innerhalb der Geräte verantwortlich.

**Steuerbus** Er regelt die Busanforderung, die Arbitrierung, die Interrupts, und das Handshaking.

**Versorgungsbus** Für die Stromversorgung und die Taktleitung ist dieser Bus verantwortlich.

Bei seriellen Bussystemen gibt es nur eine Leitung als Busstruktur. Außerdem wird zwischen unidirektionalen und bidirektionalen Leitungssystemen unterschieden.

Für die Schnelligkeit des Datentransports ausschlaggebend ist neben dem Takt die Busbreite (8-, 16-, 32- und 64-Bit). Sie entscheidet wesentlich mit über die Arbeitsgeschwindigkeit des Computers. Für die Busse im PC gibt es verschiedene Standards und Busbreiten: 16 Bit (ISA / AT-Bus), 32 Bit (EISA, Microchannel, Local Bus und PCI-Bus) und 64 Bit (PCI-64, VME-Bus).

Eine der revolutionären Eigenschaften der frühen PCs der 80er Jahre des vorherigen Jahrhunderts war die dokumentierte Schnittstelle für Erweiterungskarten. Der ISA-Bus (*Industry Standard Architectur*) ermöglichte es Drittherstellern ohne großen Aufwand Peripherie für PCs zur Verfügung zu stellen. Der erste ISA-Bus im ersten PC hatte eine Datenbusbreite von 8-Bit und lief mit einem Takt von 4,77 MHz und war damit genau so schnell wie der Prozessor. Der Bus wurde weiterentwickelt und entwickelte sich 1982 im Zusammenhang mit dem IBM PC/AT und dem 80286-Prozessor zu einem 16-Bit-Datenbus. In diesem Stadium wurde er mit 6 MHz, später mit 8 MHz betreten.

Schon bald entwickelte sich der ISA-Bus aufgrund seiner geringen Geschwindigkeit zum Flaschenhals. Seine effektive Datenrate liegt nur bei 5 MB/s. Verschiedene Hersteller versuchten durch Eigenentwicklungen

schnellere Bussysteme im PC zu etablieren. Darunter fielen der MicroChannel (MCA) und EISA. Erst der PCI-Bus am Anfang der 1990er Jahre konnte sich auf dem Markt durchsetzen und ist bis heute mit kleinen Änderungen der Standard. ISA-Steckplätze finden Sie heute nur noch sehr selten in PCs. Allein im Bereich der Embedded-Systeme und der Industrie-PCs sind noch ISA-Steckplätze reichlich vertreten.

### 16.2.1 PCI

Die Firma Intel entwickelte am Anfang der 90er Jahre des vorherigen Jahrhunderts eine neue Spezifikation für einen neuen PC-Bus: PCI. Diese Spezifikationen wurden an eine andere Organisation, die PCI SIG (*Special Interest Group*), übergeben, die 1993 die Spezifikation *PCI Local Bus Revision 2.0* herausbrachte. Durch diese genauen Definitionen war es möglich eine vernünftige Produktion für PCI-Karten aufzubauen. Daher hat sich der PCI-Bus durchgesetzt und hat sich ab 1994 als dominanter PC-Bus etabliert.

Der PCI-Bus ist ein separater, von der CPU isolierter Bus mit Zugang zum Arbeitsspeicher. Über eine Bridge wird der PCI-Bus mit dem Systembus verbunden. Dadurch ist er in der Lage mit einer festen, von der Prozessor- und Systemgeschwindigkeit unabhängigen, Taktrate zu arbeiten. Der PCI-Bus ist beschränkt auf 10 Verbindungen pro Bus. Dabei werden pro externes Gerät zwei Verbindungen verbraucht, während auf der Hauptplatine integrierte Geräte nur eine Verbindung benötigen. Der PCI-Bus unterstützt die Spannung 5 und 3,3 Volt. Durch verschiedene Bauformen der Steckplätze wird verhindert, daß die falsche Karte eingesteckt wird. Der PCI-Bus läuft in seiner Originalfassung mit einer Frequenz von 33 MHz. Mit Spezifikation 2.1 wurde die Taktrate auf 66 MHz erhöht. Da PCI Busbreiten von 32 und 64 Bit unterstützt, kann eine theoretische Datenrate von 524 MB/s erreicht werden. Allerdings sind auch heute noch Karten mit 64 Bit und 66 MHz selten in Rechnern anzutreffen.

### 16.2.2 lspci

Neben den höheren Datenraten unterstützt PCI auch die Hardwareerkennung. Jedes PCI-Gerät enthält einen Code, der seinen Typ, seinen Hersteller und sein Modell angibt. Diese Informationen können über den Befehl `lspci` abgefragt werden.

`lspci [OPTIONEN]`

Die Informationen über die PCI-Busse liefert der Kernel über das virtuelle Verzeichnis `/proc/bus/pci`. Die Schnittstelle existiert seit Kernelversion 2.1.82. Das Verzeichnis enthält für jeden Bus ein Unterverzeichnis mit Informationsdateien für die Geräte und eine Datei `/proc/bus/pci/device`, die eine Liste aller PCI-Geräte enthält.

```
enterprise:~ # lspci
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x [Apollo PR0133x] (rev 02)
00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x [Apollo MVP3/Pro133x AGP]
00:04.0 ISA bridge: VIA Technologies, Inc. VT82C686 [Apollo Super South] (rev 22)
00:04.1 IDE interface: VIA Technologies, Inc. VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235
    PIPC Bus Master IDE (rev 10)
00:04.2 USB Controller: VIA Technologies, Inc. USB (rev 10)
00:04.3 USB Controller: VIA Technologies, Inc. USB (rev 10)
00:04.4 Host bridge: VIA Technologies, Inc. VT82C686 [Apollo Super ACPI] (rev 30)
00:04.5 Multimedia audio controller: VIA Technologies, Inc. VT82C686 AC97 Audio Controller
    (rev 20)
00:0a.0 Ethernet controller: Intel Corp. 82557/8/9 [Ethernet Pro 100] (rev 08)
01:00.0 VGA compatible controller: nVidia Corporation NV5M64 [RIVA TNT2 Model 64/Model 64
    Pro] (rev 15)
```

Am Anfang jeder Zeile ist die PCI-ID angegeben, die die Position des jeweiligen Geräts auf dem PCI-Bus wiedergibt. Hersteller, Typ und Modell der Karte werden in Form von ID-Nummer abgespeichert. Eine Tabelle dieser Nummern und ihrer Zuordnung zu Herstellern, Typen und Modellen befindet sich in der Datei `/usr/share/pci.ids`.

**Optionen**

<b>-b</b>	Darstellung der IRQ und Adressen aus Sicht des PCI-Bus
<b>-d [HID]:[GID]</b>	Zeigt nur die Geräte mit angegeben Hersteller- und Geräte-ID.
<b>-i DATEI</b>	Benutzt eine andere Datei als PCI-ID-Datenbank.
<b>-m</b>	Ausgabe im maschinenlesbaren Format
<b>-n</b>	Zeigt nur die Hersteller- und Geräte-Codes an
<b>-p DIR</b>	Benutzt ein anderes Verzeichnis für die PCI-Bus-Informationen
<b>-s [B:][S][.F]</b>	Zeigt nur die Geräte mit den spezifizierten Bus, Slot und Funktion. Alle Nummern sind hexadezimal.
<b>-t</b>	Darstellung in Baumform
<b>-v</b>	Ausführlichere Informationen
<b>-vv</b>	Sehr ausführliche Informationen
<b>-x</b>	Hexadezimaler Dump der 64 Bytes des PCI-Konfigurationsspeichers

Um ausführlichere Informationen zu bekommen, können Sie den Schalter **-v** verwenden.

```
enterprise:~ # lspci -v
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x [Apollo PR0133x] (rev 02)
        Subsystem: Asustek Computer, Inc.: Unknown device 8023
        Flags: bus master, medium devsel, latency 0
        Memory at e4000000 (32-bit, prefetchable) [size=64M]
        Capabilities: [a0] AGP version 2.0

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x [Apollo MVP3/Pro133x AGP]
        (prog-if 00 [Normal decode])
...
```

Ähnliche Informationen erhalten Sie ebenfalls über eine Schnittstelle aus dem Verzeichnis `/proc`. Die virtuelle Datei `/proc/pci` zeigt Ihnen ebenfalls Informationen aus dem PCI-Bus. Diese sind im Gegensatz zu `/proc/bus/pci/devices` für Menschen lesbar gestaltet.

Für eine bessere Übersicht der PCI-Busse können Sie sich mit dem Schalter **-t** eine baumartige Struktur der Verbindung anzeigen lassen.

```
enterprise:~ # lspci -t
-[00]-+-00.0
        +-01.0-[01]----00.0
        +-04.0
        +-04.1
        +-04.2
        +-04.3
        +-04.4
        +-04.5
        \-0a.0
```

Das obige Beispiel ist etwas langweilig. Allein vom AGP-Port zweigt ein PCI-Bus ab. Dies ist natürlich die Grafikkarte.

Um die Informationen zu sehen, die die PCI-Geräte zurückgeben ohne die Übersetzung aus der PCI-Geräte-Datenbank zu benutzen, können Sie den Schalter **-n** verwenden.

```
enterprise:~ # lspci -n
00:00.0 Class 0600: 1106:0691 (rev 02)
00:01.0 Class 0604: 1106:8598
00:04.0 Class 0601: 1106:0686 (rev 22)
00:04.1 Class 0101: 1106:0571 (rev 10)
00:04.2 Class 0c03: 1106:3038 (rev 10)
00:04.3 Class 0c03: 1106:3038 (rev 10)
00:04.4 Class 0600: 1106:3057 (rev 30)
00:04.5 Class 0401: 1106:3058 (rev 20)
```

```
00:0a.0 Class 0200: 8086:1229 (rev 08)
01:00.0 Class 0300: 10de:002d (rev 15)
```

### 16.2.3 IRQs, I/O-Ports und DMA

Seine volle Einsatzbereitschaft entwickelt der PC nur, wenn alle Komponenten miteinander reden. Gerade die Kommunikation zwischen PC und Peripheriegeräten ist dabei ein wichtiger Faktor, der bis heute nicht gerade wenige Probleme bereitet hat. Die Kommunikation zwischen zwei Partner kann auf verschiedene Arten erfolgen. Wenn ein Peripheriegerät mit dem Prozessor reden will, sendet es ein Unterbrechungssignal, vergleichbar mit dem Klingeln des Telefons, an den Prozessor. Dieses Unterbrechungssignal bezeichnet man auch als *Interrupt* oder *Interrupt-Anforderung*. Abgekürzt wird dies durch die Buchstaben *IRQ*, die vom englischen Ausdruck *Interrupt ReQuest* sich ableiten. Mit diesem Signal signalisiert das Gerät an den Prozessor, daß Daten gesendet oder empfangen werden können.

Umgekehrt kann der Prozessor Geräte steuern, in dem er die sogenannten *I/O-Ports* (engl. *Input/Output-Ports*) verwendet. Daneben können Geräte auch Platz im Adressraums des Hauptspeichers fürs ROM und zur Steuerung belegen. Den für die Steuerung belegten Speicher bezeichnet man auch als *I/O-memory*. Normalerweise erfolgt der Datentransfer innerhalb der Rechner über die CPU und die I/O-Ports. Bei großen Datenmengen führt das dazu, daß der Prozessor nur mit dem Schaufeln von Daten beschäftigt ist. Dies macht sich z. B. beim Abspielen von Videos negativ bemerkbar. Um das Nadelöhr Prozessor zu umgehen, wurde *DMA* (engl. *Direct Memory Access*) entwickelt. Bestimmte ISA-Geräte, die mit großem Datenaufkommen zu kämpfen haben, können durch dieses Verfahren die Daten direkt im Hauptspeicher lesen und schreiben. PCI-Geräte beherrschen dieses Verfahren nicht. Das klingt komisch, macht aber nichts, denn sie beherrschen ein besseres System. Durch das Konzept des *bus mastering* kann das PCI-Gerät temporär den Bus selbst übernehmen und so tun, als wäre es die CPU. Um die Details kümmert sich dabei der Chipsatz auf der Hauptplatine.

Im Laufe der Jahre haben sich feste IRQs und I/O-Ports für bestimmte Peripheriegeräte eingebürgert. So liegt z. B. die beiden IDE-Controller auf den IRQs 14 und 15. Auch die parallele Schnittstelle hält schon seit Jahren die I/O-Ports 0x378 bis 0x37a besetzt. Welcher Interrupt welchem Gerät zugeordnet ist, können Sie der virtuellen Datei `/proc/interrupts` entnehmen.

```
ole@enterprise:~> cat /proc/interrupts
          CPU0
 0:   14118492      XT-PIC  timer
 1:    36197       XT-PIC  keyboard
 2:         0       XT-PIC  cascade
 5:         0       XT-PIC  usb-uhci, usb-uhci
 8:         2       XT-PIC  rtc
10:    72616       XT-PIC  eth0, VIA686A
12:   301546       XT-PIC  PS/2 Mouse
14:    54053       XT-PIC  ide0
15:   144890       XT-PIC  ide1
NMI:         0
LOC:         0
ERR:         0
MIS:         0
```

In jeder Zeile werden die Informationen zu einem IRQ angegeben. Die ersten Spalte enthält den IRQ. Die zweite Spalte zeigt an, wie oft der Interrupt aufgetreten ist. Diese Spalte wird für jeden Prozessor wiederholt. In der letzten Spalte steht dann, für welches Gerät der Prozessor zuständig ist.

Auch die Informationen zu den I/O-Ports finden Sie im Verzeichnis `/proc` wieder. Hierfür müssen Sie nur die virtuelle Datei `/proc/ioports` befragen.

```
ole@enterprise:~> cat /proc/ioports
0000-001f : dma1
0020-003f : pic1
```



```

0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
0378-037a : parport0
...
d800-d80f : VIA Technologies, Inc. VT82C586B PIPC Bus Master IDE
    d800-d807 : ide0
    d808-d80f : ide1
e400-e4ff : VIA Technologies, Inc. VT82C686 [Apollo Super ACPI]
e800-e80f : VIA Technologies, Inc. VT82C686 [Apollo Super ACPI]

```

Letztendlich sollten Sie sich noch die virtuelle Datei `/proc/dma` anschauen. Hier finden Sie die Liste der verwendeten DMA-Kanäle. Bei den heutigen PCI-Systemen sind die Informationen aber eher spärlich. Die DMA-Einstellungen der IDE-Festplatten werden übrigens hier nicht miterfaßt.

```

ole@enterprise:~> cat /proc/dma
4: cascade

```

## Probleme

Da die Ressourcen für IRQs, I/O-Ports und DMA-Kanäle (bei ISA) begrenzt sind, kann es bei intensivem Einsatz von Peripheriegeräten zu Engpässen und Problemen kommen. Eine der guten Eigenschaften von PCI ist es, daß die Geräte mit dem Chipsatz selber aushandeln, welche Ressourcen Sie benutzen können. Dadurch ist es u.a. möglich, daß sich mehrere PCI-Geräte *einen* IRQ teilen. Obwohl Probleme selten sind, ist es doch eine etwas wackelige Angelegenheit. Sollte es zu Problemen kommen, so hilft es meistens die Karten in andere Steckplätze zu stecken. Dabei hat der erste Steckplatz von dem AGP-Port aus gesehen die höchste Priorität und der Steckplatz, der am weitesten Weg ist vom AGP-Port, die niedrigste. Oft wird die Empfehlung ausgesprochen, den Steckplatz neben der AGP-Grafikkarte nicht zu benutzen, da dieser sich meistens einen IRQ mit dem AGP-Port teilt, was manchmal zu Problemen führen kann.

Beim Einsatz von ISA-Geräten, wenn Sie noch welche haben, stehen meist die Ressourcen schon fest. Entweder sind sie durch Jumper und Schalter auf der Karte hardwaremäßig einzustellen oder sie benötigen dazu eine entsprechende Software. Das böse Wort in diesem Zusammenhang ist oft *Jumperless*. In den meisten Fällen liegt die Software auf einer Diskette bereit und läuft natürlich nur unter DOS. Daher ist es auch für Linux-Puristen ratsam eine DOS/Windows-Startdiskette zu besitzen.

## 16.2.4 Plug-and-Pray

Wenn Sie Glück haben, manche würden auch sagen Sie haben Pech, unterstützen Ihre ISA-Geräte ein Konzept namens *Plug-and-Play*. Böse Zungen haben dieses Konzept mit dem Namen *Plug-and-Pray* (Steck rein und bete) versehen. Das Konzept des *ISA-PnP* ermöglicht es dem BIOS und dem Betriebssystem ähnlich wie beim PCI-Bus den Geräten freie Ressourcen zuzuweisen. ISA-PnP wird seit Kernel 2.4 unterstützt. Bei älteren Kernels oder Problemen mit aktuellen Kernels können Sie das Programm `isapnp` verwenden um ISA-Geräte zu initialisieren.

Welche Idee steckt hinter *Plug-and-Play*? Die Geräte melden an, welche IRQs, I/O-Ports, I/O-Speicherbereiche und DMA-Kanäle sie gerne hätten oder unbedingt brauchen. Nun liegt es am BIOS, dem Betriebssystem oder einer Konfigurationssoftware wie `isapnp` den Geräten die passenden Ressourcen zuzuweisen. Ehrlich gesagt, hat die PnP-Unterstützung bei Linux einige Macken und ist noch lange nicht

komplett. So erledigen unter Linux die Gerätetreiber viele Aufgaben, die eine PnP-fähiges Betriebssystem selber erledigen würde. Da aber ISA sowieso am Aussterben ist, ist dieses in Zukunft kein relevantes Problem mehr.

Die heutigen modernen Rechner besitzen ein PnP-fähige BIOS, das zumindest die wichtigsten Komponenten konfigurieren kann. Weitere Konfigurationen übernimmt es nur auf Wunsch. Da Linux kein echtes PnP-Betriebssystem ist, sollten Sie ruhig dem BIOS die Aufgaben überlassen und in den Einstellung des BIOS ruhig die Frage nach einem PnP-fähigem Betriebssystem verneinen. Sollte es dennoch Problem geben, müssen Sie die Einstellung wieder auf "Ja" setzen und ein Tool wie `isapnp` einsetzen.

Um jetzt solche ISA-PnP-Geräte manuell einzurichten, machen Sie sich erst mit dem Tool `pnpdump` ein Bild über die verwendeten ISA-Geräte und ihre Möglichkeiten. Die Ausgabe von `pnpdump` liefert eine fast fertige Vorlage für die Konfigurationsdatei von `isapnp`. Mit dem Aufruf von `isapnp` überschreiben Sie eventuell vorhandene BIOS-Einstellungen. Dies gilt natürlich nur für die aktuelle Sitzung und nicht für den BIOS-Speicher. Für die genaue Funktionsweise und Anwendung konsultieren Sie bitte die Manual-Pages **isapnp(8)**, **isapnp.conf(5)** und **pnpdump(8)**, sowie das "*Plug-and-Play-HOWTO*".

Bei der Verwendung von uralten, nicht PnP-fähigen ISA-Karten können Sie nur noch im BIOS Einstellungen für diese Karten per Hand machen und hoffen, das es so funktioniert. Bei vielen BIOS-Versionen gibt es heute keine Möglichkeiten mehr der manuellen Vergaben von Ressourcen.

## 16.3 Datenfernübertragung

Die Arbeit mit Rechnern macht meistens erst richtig Spaß, wenn diese Daten miteinander austauschen können. Dies kann man besonders deutlich in dem zweiten Anschaffungsboom von privaten Rechner erkennen, der durch das Internet ausgelöst wurde. Der erste wurde bekanntlich durch die Computerspiele ausgelöst. Auf jeden Fall gibt es viele Möglichkeiten Rechner miteinander zu vernetzen. Neben den LAN-Techniken wie Ethernet und WLAN, bei denen es um Vernetzungen auf kleinem lokalen Raum handelt, gibt es Verfahren um Rechner über das Telefonnetz miteinander zu verbinden. Während das Modem im analogen Verfahren und ISDN-Karten im digitalen Verfahren direkt eine Telefonleitung nutzen, bedient sich DSL als Breitbandmedium den von Sprachdiensten nicht genutzten höheren Datenkanälen. Diese drei Datenübertragungsverfahren werden unter Linux unterschiedlich behandelt, obwohl es natürlich auch Gemeinsamkeiten gibt.

### 16.3.1 Analog: Modems

Das Wort *Modem* bezeichnet Geräte, die zum Zwecke der Datenübertragung digitale Daten in analoge Signale (Töne) umwandelt und umgekehrt. Das Wort Modem ist eine Kurzform von *modulator/demodulator*. Um Daten über eine analoge Telefonleitung zu übertragen brauchen Sie immer zwei Modems. Das erste wandelt die Signale des Rechners in Töne um und überträgt sie über das Telefonnetz. Am anderen Hörer bzw. Ende sitzt ein weiteres Modem, daß dann die Töne wieder in digitale Daten umwandelt. Unter günstigen Umständen können Sie auf diese Art und Weise Daten mit einer Geschwindigkeit von bis zu 57600 Bit/s (ungefähr 56 KBit/s) übertragen. Modems können digitale Datenkompression und Fehlerkorrektur bei der Übertragung der Daten verwenden um die Geschwindigkeit und Sicherheit der übertragenen Daten zu erhöhen. Bei Verwendung einer Kompression ist es sogar möglich mehr als die 57600 Bit/s an Daten zu übertragen. Deshalb sollten Sie versuchen Ihr 56k-Modem mit der doppelten Geschwindigkeit (115200 Bit/s) anzusteuern. Auf jeden Fall müssen Sie zwischen der Schnittstellengeschwindigkeit (Modem-Rechner) und der Leitungsgeschwindigkeit (Modem-Modem) unterscheiden. Während erste von den Modems fest eingestellt werden kann, hängt die zweite Geschwindigkeit von der Leitungsgüte und der Tagesform ab.

#### Externe und interne Modems

Im PC-Bereich gibt es im Prinzip zwei Klassen von Modems. Die externen Modems sind separate Geräte, die über ein Kabel mit der seriellen Schnittstelle (oft *RS-232*) oder einem USB-HUB an den Rechner angeschlossen sind. Dies klappt meistens ohne Probleme, da die Leitungsgeschwindigkeit unter der Schnittstellengeschwindigkeit liegt. Während USB-Modems meistens spezielle Treiber benötigen, funktionieren Modems, die über die serielle Schnittstelle angeschlossen sind, meistens ohne besondere Treiber oder Konfiguration. Daher sollten Sie heute beim Einsatz von Linux ein Gerät wählen, daß über die serielle Schnittstelle angeschlossen

wird, oder sich sehr genau informieren. Externe Modems haben zudem meistens LEDs, die über den Zustand des Modems Aufschluß geben und damit bei der Fehlersuche behilflich sein können.

Interne Modems dagegen sind billiger. Sie verbrauchen weniger Strom und keinen zusätzlichen Platz. Und solange Sie einen weiten Bogen um die sogenannten *WinModems* machen, funktionieren sie genauso wie externe Modems. WinModems sind speziell für Windows entwickelt worden und laufen nur mit proprietären Treibern. Durch diese Arbeitsweise können bei der Produktion pro Gerät ein paar Cent eingespart werden. Da sie aber auf den Standard verzichten, werden eben spezielle Treiber benötigt. Inzwischen unterstützt Linux viele Winmodems, bzw. die Hersteller bieten jetzt auch für Linux ihre proprietären, leider nicht im Quellcode verfügbaren Treiber an. Informationen und Treiber für WinModems finden Sie auf der Webseite <http://www.linmodems.org/>. Sollten Sie ein normales PCI-basierendes Modem besitzen, so wird es automatisch von den Treibern der seriellen Schnittstelle erkannt. Interne ISA-Modems können über *Plug-and-Play* konfiguriert werden. Da in einem typischen PC auch heute noch meist zwei serielle Schnittstelle integriert sind (`/dev/ttyS0` und `/dev/ttyS1`), wird das interne Modem meist als dritte Schnittstelle (`/dev/ttyS2`) erscheinen.

## Kommunikation

Es gibt zwei Arten über ein Modem zu kommunizieren. Der erste und ursprüngliche Weg ist es einen anderen Rechner direkt anzuwählen, also eine dedizierte Telefonleitung zu verwenden. Sie können dann auf dem Rechner interaktiv arbeiten, Dateien transferieren, Ihre Post abholen und andere lokale Aktionen ausführen. Dies war die Arbeitsweise in der Mailboxszene der 80er Jahre des letzten Jahrhunderts. Heute ist die Mailboxszene so ziemlich ausgestorben. Für schnelle und sichere Verbindungen zwischen zwei Rechnern oder Netzen werden aber heute noch dedizierte Telefonleitungen verwendet. In den meisten Fällen werden keine Modems mehr verwendet, sondern die digitalen ISDN-Adapter.

Heute ist es üblich seinen Rechner über PPP (engl. *Point-to-Point Protocol*) direkt ins Internet zu integrieren. Dabei bekommt der Rechner vom Provider eine öffentliche IP-Adresse (siehe Abschnitt 17.3) zugewiesen und ist damit für die Dauer der Einwahl vollwertiges Mitglied des Internets mit allen Möglichkeiten und auch Risiken. Sie können im World Wide Web surfen, eMails versenden, sich über ICQ Nachrichten zuschicken, chatten oder irgendeinen der anderen Internet-Dienst nutzen. Sie können sogar selber Dienste anbieten. Allerdings können Sie durch fehlerhafte Implementierungen der Netzwerkprotokolle, falsch konfigurierte oder programmierte Serverdienste sich üble Gesellen (Würmer, Trojaner, Viren) auf Ihren Rechner holen. Sie können auf einen Virus in Ihren eMails hineinfallen und somit selber zum Verbreiten von Viren werden. Allerdings ist dies unter Linux sehr selten, aber trotzdem möglich.

## Direkte Einwahl

Die direkte Einwahl ist primitiv, aber auch leicht zu konfigurieren und setzt eine relativ wenig komplexe Software voraus. Sie brauchen im Prinzip nur ein Terminalprogramm wie `minicom` (siehe Manual-Pages `minicom(1)`).

Die meisten Modems verstehen die Kommandos des *Hayes*-Standard. Dieser Standard ist benannt nach dem Entwickler des ersten PC-Modems. Die Kommandos dieses Standards beginnen immer mit der Zeichenfolge AT. Sie können ein oder mehrere Kommandos einleiten. Die Tabelle 16.1 präsentiert eine Auswahl der gängigen Kommandos. Die meisten Modems heute kennen eine Reihe weitere Kommandos, wie z. B. für Faxfunktionen, die aber nicht standardisiert sind.

Im Terminalprogramm können Sie z. B. mit der Zeichenfolge ATDT gefolgt von der gewünschten Rufnummer eine Verbindung aufbauen. Zum Verbindungsabbau melden Sie sich bei der Gegenstelle ab. In den meisten Fällen baut diese dann die Verbindung ab. Sollte dies nicht der Fall sein, dann tippen Sie dreimal das Pluszeichen ein (+++) und warten einen kurzen Moment. Nach einer guten Sekunde sollte das Modem mit der Meldung "OK" signalisieren, daß es Kommandos annimmt. Sie können dann mit der Zeichensequenz ATH0 die Verbindung beenden.

<b>ATDTNUMMER</b>	Wählen mit Tonwahlverfahren
<b>ATDPNUMMER</b>	Wählen mit Impulswahlverfahren
<b>ATH0</b>	Sitzung beenden
<b>ATL0 ... ATL3</b>	Lautstärke: aus/leise ... laut
<b>ATM0</b>	Lautsprecher aus
<b>ATM1</b>	Lautsprecher an, bis Verbindung aufgebaut
<b>ATM2</b>	Lautsprecher immer an
<b>ATX3</b>	Modem wartet zum Wählen nicht auf Freizeichen (Telefonanlage)
<b>ATZ</b>	Zurückstellen auf gespeicherte Einstellungen

Tabelle 16.1: Die wichtigsten Modemkommandos nach dem Hayes-Standard

### 16.3.2 setserial

Der Befehl **setserial** setzt oder berichtet die aktuellen Konfigurationseinstellungen der seriellen Schnittstellen. Dies können die "normalen" Schnittstellen sein, wie auch Schnittstellen, die zu internen Modems gehören.

```
setserial [OPTIONEN] GERÄT [PARAMETERLISTE]
```

Wenn Sie nur die Geräteschnittstelle angeben, liefert **setserial** aktuelle Informationen zur Schnittstelle aus.

```
enterprise:~ # setserial /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Im einfachsten Fall liefert der Befehl **setserial** den Namen der Schnittstelle, den verwendeten Schnittstellenbaustein, den I/O-Port und den Interrupt der Schnittstelle aus. Der hier angegebene Chip "16550A" ist der am häufigsten verwendete Baustein für die serielle Schnittstelle. Im Gegensatz zu anderen Bausteinen besitzt er einen FIFO-Puffer und muß nicht für jedes gelesene Zeichen einen Interrupt auslösen. Nur so ist es möglich eine Übertragungsrate von 115200 Bit/s zu erreichen.

#### Optionen

<b>-a</b>	Ausführliche Informationen zu der Schnittstelle
<b>-b</b>	Ausgabe in einer lesbareren Form
<b>-g</b>	Informationen über eine Liste von Schnittstellen ausgeben.
<b>-G</b>	Ausgabe der Informationen im Eingabeformat von Setserial
<b>-q</b>	Gibt weniger Zeilen zur Information aus.
<b>-v</b>	Gibt mehr Informationen bei der Arbeit aus.
<b>-z</b>	Setzt die Werte vor der Ausgabe auf den Startwert zurück.

Sie können mit **setserial** auch Werte setzen. Denken Sie bitte daran, daß Sie damit nur die Werte für die Ansprache durch den Kernel ändern. Die hardwareseitige Konfiguration können Sie mit dem Befehl nicht ändern. Dafür müssen Sie entweder das BIOS bemühen oder ein passendes Konfigurationstool verwenden. Um mit **setserial** die Parameter der Schnittstelle setzen zu können, brauchen Sie nur hinter dem Schnittstellennamen die Schlüsselwörter und gewünschten Werte anzugeben.

```
enterprise:~ # setserial /dev/ttyS0 port 0x03f8 irq 4 baud_base 115200 spd_normal
```

So wird z. B. die Schnittstelle **/dev/ttyS0** auf die erste serielle Schnittstelle eingestellt. Dies ist normalerweise auch der Fall. Die wichtigsten Schlüsselwörter sind **port** für den I/O-Port, **irq** für den Interrupt, **baud\_base** für die maximale Datenübertragungsrate des Bausteins und **uart** für den Typ des verwendeten Bausteins.

Die Geschwindigkeit der Schnittstelle können Sie über die Speed-Parameter angeben. Dabei geht es um die Geschwindigkeit, wenn die zugreifende Applikation eine Geschwindigkeit von 38,4 kbit/s anfordert.

```

spd_normal  38,4 kbit/s
spd_hi      57,6 kbit/s
spd_vhi     115 kbit/s
spd_shi     230 kbit/s
spd_warp    460 kbit/s

```

Weitere Parameter und Schalter entnehmen Sie bitte der Manual-Page **setserial(8)**.

Spezielle Konfigurationen erfolgen meistens in einem RC-Skript. Die kann z. B. das Skript `rc.serial` sein. Bei SuSE 9.0 liegt das passende Skript dort, wo sich auch alle anderen RC-Skripte befinden. Das Skript `/etc/init.d/setserial` übernimmt hier die Aufgabe der Konfiguration. Im Normalfall tut dieses Skript nichts, da der Kernel die Standardwerte für die Konfiguration automatisch verwendet.

## Beispiele

Die Ausgabe kann in mehreren Formaten erfolgen.

```

enterprise:~ # setserial /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
enterprise:~ # setserial -a /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
      Baud_base: 115200, close_delay: 500, divisor: 0
      closing_wait: 30000
      Flags: spd_normal skip_test
enterprise:~ # setserial -b /dev/ttyS0
/dev/ttyS0 at 0x03f8 (irq = 4) is a 16550A
enterprise:~ # setserial -G /dev/ttyS0
/dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_normal skip_test

```

Der Schalter `-g` erlaubt es eine Liste von Geräten anzugeben.

```

enterprise:~ # setserial -g /dev/ttyS*
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 3
/dev/ttyS10, UART: unknown, Port: 0x02a8, IRQ: 5, Flags: Fourport
/dev/ttyS11, UART: unknown, Port: 0x02b0, IRQ: 5, Flags: Fourport
/dev/ttyS12, UART: unknown, Port: 0x02b8, IRQ: 5, Flags: Fourport
/dev/ttyS13, UART: unknown, Port: 0x0330, IRQ: 4
/dev/ttyS14, UART: unknown, Port: 0x0338, IRQ: 4
/dev/ttyS15, UART: unknown, Port: 0x0000, IRQ: 0
/dev/ttyS16, UART: unknown, Port: 0x0000, IRQ: 0
/dev/ttyS17, UART: unknown, Port: 0x0100, IRQ: 12
/dev/ttyS18, UART: unknown, Port: 0x0108, IRQ: 12
/dev/ttyS19, UART: unknown, Port: 0x0110, IRQ: 12
/dev/ttyS2, UART: unknown, Port: 0x03e8, IRQ: 4
/dev/ttyS20, UART: unknown, Port: 0x0118, IRQ: 12
/dev/ttyS21, UART: unknown, Port: 0x0120, IRQ: 12
/dev/ttyS22, UART: unknown, Port: 0x0128, IRQ: 12
/dev/ttyS23, UART: unknown, Port: 0x0130, IRQ: 12
/dev/ttyS3, UART: unknown, Port: 0x02e8, IRQ: 3
/dev/ttyS4, UART: unknown, Port: 0xffffffff, IRQ: 0
/dev/ttyS5, UART: unknown, Port: 0x01a0, IRQ: 2, Flags: Fourport
/dev/ttyS6, UART: unknown, Port: 0x01a8, IRQ: 2, Flags: Fourport
/dev/ttyS7, UART: unknown, Port: 0x01b0, IRQ: 2, Flags: Fourport
/dev/ttyS8, UART: unknown, Port: 0x01b8, IRQ: 2, Flags: Fourport
/dev/ttyS9, UART: unknown, Port: 0x02a0, IRQ: 5, Flags: Fourport
/dev/ttySL0: No such device

```

### 16.3.3 isserial

Mit dem Befehl `isserial` kann man feststellen, ob die angegebene Standardeingabe bzw. Terminal eine serielle Schnittstelle ist.

```
isserial [< TERMINAL]
```

Der Fehlercode von `isserial` ist **0**, wenn die Standardeingabe eine serielle Schnittstelle ist. Wenn nicht, wird der Fehlercode **1** zurückgeben. Das Tool wird vor allem in Skripten eingesetzt, um festzustellen ob die Eingaben von einer seriellen Schnittstelle kommen.

#### Beispiele

```
enterprise:~ # isserial; echo $?
1
enterprise:~ # isserial < /dev/tty1; echo $?
1
enterprise:~ # isserial < /dev/ttyS0; echo $?
0
```

### 16.3.4 Digital: ISDN

Der Nachfolger der analogen Telefonnetzes ist ISDN<sup>2</sup> (*Integrated Services Digital Network*). ISDN ist ein digitales Fernsprechnetz, das man wie einen analogen Anschluss zum Telefonieren verwenden kann. Dieses digitale Netz mit seinen integrierten Diensten bietet deutliche Vorteile gegenüber dem analogen Netz. Im Bereich der Kommunikation zwischen Rechnern hat ISDN eindeutig Vorteile gegenüber dem analogen Netz. ISDN stellt bei einem Basisanschluß zwei Datenkanäle (B-Kanäle) mit je 64 KBit/s. Dazu kommt ein Steuerungskanal (D-Kanal) mit 16 KBit/s, der nicht direkt nutzbar ist und für Signalisierungszwecke (unter anderem Uhrzeit und Rufnummer) genutzt werden kann. Durch die Verwendung des D-Kanals funktioniert die Signalisierung bei ISDN Out-of-Band und wird nicht wie beim Mehrfrequenzwahlverfahren im Sprachkanal durchgeführt. Dadurch funktioniert der Verbindungsaufbau störungsfreier und schneller. Die Datenkanäle können unabhängig voneinander für Telefonie, Datenübertragung und andere Dienste verwendet werden. Es ist aber auch möglich die beiden B-Kanäle zu bündeln und damit dann mit doppelter Geschwindigkeit Daten auszutauschen. Das funktioniert natürlich nur, wenn die Gegenstelle diese Funktion unterstützt. Außerdem fallen natürlich die doppelten Verbindungskosten an.

Natürlich gibt es für ISDN wie bei den Modems externe und interne Geräte. Die externen Geräte können über die serielle Schnittstelle oder über USB angeschlossen werden. Sie verhalten sich dabei oft wie normale Modems. In den meisten Fällen kann die bisherige DFÜ-Software weiter eingesetzt werden. Allerdings können Sie nicht alle Funktionen von ISDN damit ausnutzen. In den meisten Fällen sind aber interne Karten die Wahl. Bei diesen Karten, die einen Steckplatz im Rechner belegen, wird zwischen aktiven und passiven Karten unterschieden. Aktive Karten besitzen einen eigenen Prozessor und belasten nicht die CPU des Rechners. So können zeitaufwendige Aktionen wie der Empfang von Faxen ohne Probleme erledigt werden. So gehören aktive ISDN-Karten zur Standardaustattung von Telefonanlagen auf PC-Basis. Die passiven ISDN-Karten nutzen dagegen die CPU des PC-Systems. Sie sind deswegen deutlich billiger als aktive Karten und sind hauptsächlich im privaten Bereich und in Desktop-System zu finden.

Direkte Rechnern zu Rechner Verbindungen werden heute meistens über ISDN realisiert. Durch die dedizierte Leitung meist in Kombination mit Kanalbündelung wird eine bestimmte Datenübertragungsrate garantiert, was z. B. bei medizinischen Operationen überlebenswichtig ist. Auch das Abhören der Verbindung ist wesentlich schwieriger als bei einer Übertragung durch das Internet. Daher übertragen viele Firmen sensible Daten zwischen den Standorten auf eigenen ISDN-Standleitungen.

Gängig ist heute natürlich die Einbindung des Rechners mit ISDN über PPP zu einem Provider ins Internet. Auch hier bekommen Sie vom Provider eine dynamische IP-Adresse zugewiesen und sind damit ein vollwertiges Mitglied des Internets mit allen Möglichkeiten und Risiken.

---

<sup>2</sup>Übrigens ist ISDN auch die Abkürzung für den Medikamentwirkstoff *Iso-Sorbit-DiNitrat*.

### 16.3.5 Breitband: DSL

Die Zukunft der Datenfernübertragung liegt im Breitband. Für den Massemarkt und Endanwender bedeutet dies DSL (*Digital Subscribe Line*). Für DSL werden zwei Modems benötigt, eins in der Vermittlungsstelle des Providers und eins beim Kunden. Die Technik nutzt die Tatsache, daß der herkömmliche analoge bzw. digitale Telefonverkehr im Kupferkabel nur Frequenzen bis 4 kHz belegt. Theoretisch jedoch sind auf Kupferleitungen Frequenzen bis 1,1 MHz möglich. Deshalb kann durch das Aufsplitten der Bandbreite in unterschiedliche Kanäle, z.B. für Daten- und Sprachinformationen, und die Nutzung der bislang verwaisten höheren Frequenzbereiche unter Verwendung der neuesten DSL-Technologien Übertragungsraten von bis zu 52 MBit/s erreicht werden. Dies ist natürlich abhängig von der eingesetzten DSL-Variante.

Bei DSL wird zwischen ADSL, wo der Download wesentlich schneller läuft als der Upload, und SDSL, wo der Download genauso schnell läuft wie der Upload, unterschieden. Das in Deutschland am meisten genutzte ADSL ist "T-DSL" der Telekom. Es erlaubt Geschwindigkeiten von 1000 bis 3000 KBit/s beim Download und 128 bis 384 KBit/s beim Upload.

Die meisten DSL-Modems werden über eine Ethernet-Netzwerkkarte mit dem Rechner verbunden. Daneben gibt es PCI-Karten für den Rechner oder man setzt gleich auf einen Router, der das DSL-Modem bereits integriert hat. Im Prinzip bräuchte man auf der Providerseite nur einen IP-Router aufstellen um den Rechner dann ins Internet zu integrieren. Da die Benutzer sich natürlich bei Provider authentisieren müssen und auch die Nutzung irgendwie abgerechnet werden muß, wird eine Variante des PPP verwandt. Das sogenannte *PPPoE* (*Point to Point Protocol over Ethernet*) wird für die Kommunikation zwischen Modem und Ethernet-Karte verwendet. Die tatsächliche Verbindung zwischen dem Rechner und dem Internet erfolgt über PPP. Für das PPPoE gibt es eine Reihe von Linux-Implementierung, so u.a. das *Roaring Penguin PPPoE*.

## 16.4 Audio: Es gibt was auf die Ohren

Natürlich ist auch das Thema Audio wichtig für Linux. Im Prinzip gibt es zwei Audiotreiber für Linux. Dies ist zum einen das *Open Sound System* (*OSS*). Dabei handelt es sich im Prinzip um eine abgespeckte Version eines kommerziellen Treibers für eine große Anzahl von Audiokarten. Das zweite System ist das *Advanced Linux Sound Architecture* (*ALSA*), das heute von den meisten Distributionen verwendet wird. Es ist wesentlich leistungsfähiger und basiert im Gegensatz zu OSS von Anfang an auf frei verfügbare Quellen. Gerade wenn es um qualitativ hochwertige Soundkarten geht, kommen Sie um den Einsatz von ALSA nicht herum. OSS ist dagegen mehr für die Soundkarten aus dem Konsumentenbereich ausgelegt.

Wie bei aller Hardware ist es gut zu wissen, ob diese auch von den entsprechenden Systemen unterstützt wird. Auf den jeweiligen Homepages der Projekte finden Sie Listen der unterstützten Audiokarten. Dabei ist es oft nützlich, neben dem Hersteller und dem Modell auch den auf der Karte eingesetzten Chipsatz zu kennen. Manche Hersteller neigen dazu schnell und ohne Ankündigung die eingesetzten Chipsätze zu ändern.

Für die Konfiguration stehen mehrere Tools zur Verfügung. So stammt von *Red Hat* das Programm *sndconfig*, mit dem Sie ISA-PnP- und PCI-Soundkarten konfigurieren können. Dieses Programm verwendet das Tool *isapnp* um eine ISA-PnP-Soundkarte zu konfigurieren. Dazu paßt es auch die Datei */etc/modules.conf* an, um die korrekten Treiber zu laden. Andere Distributionen haben ebenfalls ihre Tools entwickelt, sei es für OSS oder für ALSA. Bei SuSE übernimmt YaST die Konfiguration der Soundkarte.

Die Homepage des OSS ist unter der Adresse <http://www.opensound.com/> zu finden. Ein etwas älteres Howto aus dem Jahre 2001 finden Sie auf der Seite des Linux Documentation Projekt (<http://www.tldp.org>). Das "Sound-HOWTO" ist zwar etwas älter, aber noch immer nützlich. Auch für das ALSA-Projekt gibt es eine Homepage. Diese finden Sie unter der Adresse <http://www.alsa-project.org/>.

## 16.5 SCSI

SCSI ist die Abkürzung für *Small Computer System Interface*. Es ist also eine Datenschnittstelle für kleine Computer. Allerdings war im Jahre 1979, als der Magnetplattenhersteller *Shugart* sich Gedanken über ein neues Festplatteninterface machte, ein kleiner Computer etwas was kleiner war als ein Kleiderschrank. Jedenfalls wurde die Spezifikation *SASI* (*Shugart Associates Systems Interface*) von *NCR* aufgegriffen und

Name	Breite	Geräte	Kabellänge	Datenrate
SCSI-1	8 Bit	8	6,0 m	5 MByte/s
SCSI-2 Fast	8 Bit	8	3,0 m	10 MByte/s
SCSI-2 Wide	16 Bit	16	3,0 m	20 MByte/s
SCSI-3 Ultra	8 Bit	4/8	3,0/1,5 m	20 MByte/s
SCSI-3 Ultra HVD	8 Bit	8	25 m	20 MByte/s
SCSI-3 Ultrawide	16 Bit	4/8	3,0/1,5 m	40 MByte/s
SCSI-3 Ultrawide HVD	16 Bit	16	25 m	40 MByte/s
Ultra-2-SCSI LVD	8 Bit	8	12 m	40 MByte/s
Ultrawide-2-SCSI LVD	16 Bit	16	12 m	80 MByte/s
Ultra-160-SCSI	16 Bit	16	12 m	160 MByte/s
Ultra-320-SCSI	16 Bit	16	12 m	320 MByte/s
Ultra-640-SCSI	16 Bit	16	10 m	640 MByte/s

Tabelle 16.2: Die verschiedenen SCSI-Varianten

erweitert. Nach der Namensänderung in SCSI (ausgesprochen wie "Skasi") 1982 wurde die erste Norm 1986 von der *ANSI* abgesegnet.

Heute ist SCSI das System für den professionellen Einsatz. Sie können nicht nur Festplatten und Wechseldatenträger an SCSI betreiben, sondern auch Scanner und andere Geräte sind für den Bus ausgelegt. In den meisten Fällen findet man SCSI in Servern. Desktop-Rechner und Heimrechner verwenden in den meisten Fällen IDE zum Anschluß von Massenspeichern und USB für den Anschluß von externen Geräten. Dies liegt vor allen Dingen daran, daß IDE- und USB-Geräte wesentlich günstiger in der Herstellung sind als SCSI-Geräte. So kostet z. B. eine SCSI-Platte im Schnitt das drei bis vierfache einer vergleichbaren IDE-Platte. Allerdings ist nicht nur die SCSI-Logik für den Aufpreis verantwortlich. Für beiden Systeme wird eine Lebensdauer von fünf Jahren angegeben. Allerdings wird bei SCSI-Systeme von einer Betriebszeit von 24 Stunden am Tag und 7 Tagen in der Woche ausgegangen. Die IDE-Systeme sind aber für den Betrieb in Büro-PCs gedacht mit einer Betriebszeit von 8 Stunden am Tag und 5 Tagen in der Woche. Auf die gesamte Lebenszeit hochgerechnet kommt daher ein SCSI-System auf ca. 43.700 Betriebsstunden, während für ein IDE-System nur mit einer Betriebszeit von 10.400 Stunden gerechnet wird. Diese längere Lebensdauer hat natürlich auch seinen Preis.

Als Linux-Administrator sollten Sie sich immer mit SCSI auskennen. Selbst wenn Sie kein SCSI in Ihrem Netzwerk eingesetzt haben, haben Sie unter Linux mit SCSI zu tun. Viele USB-, IDE- oder FireWire-Geräte werden vom Kernel wie SCSI-Geräte angesprochen und nutzen die dieselbe Infrastruktur.

Um SCSI nutzen zu können, muß ein PC mindestens einen SCSI-Hostadapter haben. Bei manchen Geräten ist dieser auf der Hauptplatine integriert, bei anderen Systemen befindet er sich auf einer Steckkarte. Gute Hostadapter besitzen ein SCSI-BIOS, welches das Booten von SCSI-Geräten ermöglicht. Hier bekommen Sie auch einen Überblick über vergebene SCSI-IDs und angeschlossene Geräte.

Es gibt eine Reihe von SCSI-Varianten, die die Arbeit mit dem System nicht gerade erleichtern. Die unterschiedlichen Varianten unterscheiden sich z. B. in der Anzahl der anschließbaren Geräte. Dabei zählt der SCSI-Hostadapter als Gerät mit. Dazu kommt die Busbreite, die maximale Länge und die Übertragungsgeschwindigkeit. Ein Überblick über die verschiedenen Varianten liefert die Tabelle 16.2.

### 16.5.1 Linux und SCSI

SCSI wird von Linux bestens unterstützt. Dies klappt so gut, daß viele andere Geräte (z. B. USB-Speicher und IDE-CD-Brenner) auch als SCSI-Emulationen verwaltet werden.

#### SCSI-Subsystem

Das *SCSI-Subsystem* basiert auf einem 3-Schichten-Modell, wie Sie es in Abbildung 16.1 sehen können. Die oberste Schicht liegt am nächsten zur Benutzer-/Kernel-Schnittstelle, während die unterste Schicht näher zur Hardware liegt. Die Treiber aus der obersten Schicht werden durch eine Abkürzung, die aus zwei Buchstaben



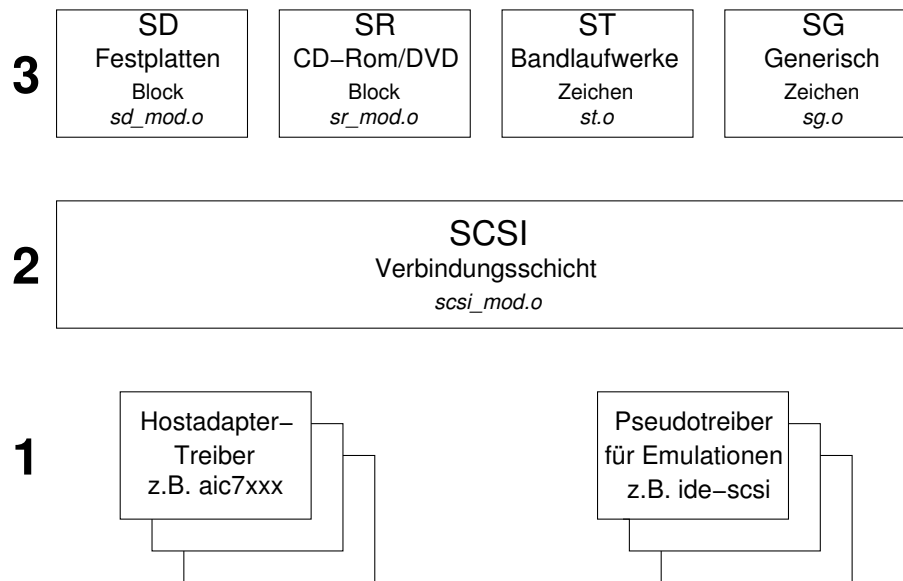


Abbildung 16.1: Die Schichten des SCSI-Subsystems

besteht, bezeichnet. Dort gibt die folgenden vier Schnittstellen:

**SD** ist eine Schnittstelle zu der Familie der allgemeinen Festplatten, wie es *hd* für das IDE-Subsystem ist. Eine der häufigsten verwendeten Aufgaben für blockorientierte Geräte ist das Einhängen von Partitionen ins Dateisystem.

Beispiel: `mount -t ext2 /dev/sda3 /daten`

Natürlich steht auch der Befehl `fdisk` (10.1.1) für SCSI-Festplatten zur Verfügung.

**SR** ist die Schnittstelle zu den CD-ROM- bzw. DVD-Laufwerken. Im Gegensatz zu den Festplatten wird hier in der Regel nicht eine Partition sondern der gesamte Datenträger gemountet.

Beispiel: `mount -t iso9660 /dev/sr0 /media/cdrom`

Natürlich können Sie über diese Schnittstelle auch Audio-CDs lesen. Allerdings wird diese Audio-CD nicht gemountet. Dies ist logisch, da die CD kein Dateisystem enthält.

**ST** ist eine zeichenorientierte Schnittstelle zum Lesen und Schreiben von Bandlaufwerken. So können Sie das Kommando `mt` für den Datentransfer und Kontrolle solcher Geräte verwenden.

**SG** ist eine SCSI-Emulationsschnittstelle auf der Basis eines zeichenorientierten Gerätes. Die Schnittstelle ermöglicht es SCSI-Kommandos an Nicht-SCSI-Geräte zu senden. So können Programme wie *SANE* für Scanner, `cdrecord` und `cdrecord` für CD-Brenner und `cdparanoia` für das digitale Auslesen von Audio-CDs auf diese Geräte zugreifen.

Die Treiber für die passenden Schnittstellen heißen `sd_mod.o`, `sr_mod.o`, `st.o` und `sg.o`. Die unterschiedliche Namensgebung ist historisch bedingt. Sie finden die Treiber in einem der Modulverzeichnisse des Kernels mit dem Namen `scsi`.

```
ole@enterprise:~> find /lib/modules/ -name scsi
/lib/modules/2.4.21-215-athlon/kernel/drivers/scsi
```

## SCSI-Adressierung

Unter Linux wird eine vierstufige Adressierung für ein SCSI-Gerät verwendet. Diese besteht aus

- der Nummer des SCSI-Hostadapters (host),

- der Kanal-Nummer (bus),
- der Geräte-ID-Nummer (target) und
- der logischen Einheit (lun).

In den meisten Fällen besitzt Ihr System nur einen Hostadapter. Die Zählweise beginnt bei 0. Jeder Hostadapter kann einen oder mehr Busse kontrollieren. Im Normalfall gibt es nur einen Bus. Auf jedem Bus können nun, je nach SCSI-Variante 7 oder 15 Geräte angeschlossen sein. Dabei beginnt die Zählweise bei 0. Diese ID ist für die Bootplatte reserviert, während der SCSI-Hostadapter eigentlich immer die ID 7 bekommt. Die Abkürzung *LUN* für *Logical Unit Number* bezeichnet eine Untereinheit in einem SCSI-Geräte. Dies können z. B. die CDs in einem Storage Center sein oder die einzelnen Festplatten in einem SCSI-Storagetower.

## SCSI im Proc-Verzeichnis

Wie schon bei anderen Hardwarekomponenten spiegelt sich die Verwaltung der SCSI-Systeme im Proc-Dateisystem wieder. Der Kernel liefert die Informationen über das Verzeichnis `/proc/scsi`, in dem weitere virtuelle Dateien und Verzeichnisse stehen.

```
enterprise:~ # ls /proc/scsi
aic7xxx ide-scsi scsi sg usb-storage-0
```

Informationen über die angeschlossenen Geräte erhalten Sie über die virtuelle Datei `/proc/scsi/scsi`. So zeigt das untere Beispiel einen Rechner mit drei angeschlossenen SCSI-Geräten. Dabei handelt es sich um ein echtes SCSI-Gerät (IOMEGA) und zwei SCSI-Simulationen, nämlich um einen ATAPI-CD-Brenner und um einen USB-Speicherstift. Aus diesem Grund werden hier gleich drei SCSI-Controller aufgeführt.

```
enterprise:~ # cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 04 Lun: 00
  Vendor: IOMEGA   Model: ZIP 100           Rev: E.02
  Type:   Direct-Access           ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 00 Lun: 00
  Vendor: PLEXTOR  Model: CD-R   PX-W4824A Rev: 1.01
  Type:   CD-ROM           ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor: 3SYSTEM  Model: USB FLASH DISK Rev: 1.00
  Type:   Direct-Access           ANSI SCSI revision: 02
```

Im Verzeichnis `aic7xxx` finden wir in unserem Beispiel die Daten über den SCSI-Controller der Firma Adaptek. Die darin enthaltene Datei 0 beschreibt den ersten SCSI-Controller mit seinen angeschlossenen Geräten.

```
enterprise:~ # ls /proc/scsi/aic7xxx/
0
enterprise:~ # cat /proc/scsi/aic7xxx/0
Adaptec AIC7xxx driver version: 6.2.36
Adaptec 2940 SCSI adapter
aic7870: Single Channel A, SCSI Id=7, 16/253 SCBs
Allocated SCBs: 5, SG List Length: 102
...
Target 3 Negotiation Settings
  User: 10.000MB/s transfers (10.000MHz, offset 127)
Target 4 Negotiation Settings
  User: 10.000MB/s transfers (10.000MHz, offset 127)
  Goal: 3.300MB/s transfers
  Curr: 3.300MB/s transfers
```

```

Channel A Target 4 Lun 0 Settings
    Commands Queued 198
    Commands Active 0
    Command Openings 1
    Max Tagged Openings 0
    Device Queue Frozen Count 0
...

```

Auch die IDE-SCSI-Emulation wird als SCSI-Controller gewertet. Deshalb findet sich auch hier extra ein Verzeichnis. Dieses Verzeichnis enthält ebenfalls eine Datei. Da es sich hierbei um den zweiten SCSI-Controller in der Hierarchie handelt, hat die Datei den Namen 1. Die Informationen hier sind war eindeutig, aber auch ziemlich nutzlos.

```

enterprise:~ # ls /proc/scsi/ide-scsi/
1
enterprise:~ # cat /proc/scsi/ide-scsi/1
SCSI host adapter emulation for IDE ATAPI devices

```

### 16.5.2 scsiinfo

Informationen über angeschlossene SCSI-Geräte liefert der Befehl `scsiinfo`.

`scsiinfo [OPTIONEN] [GERÄT]`

Der Befehl verwendet dazu das SCSI-Kommando INQUERY und die Informationen aus den SCSI-II-Modusseiten. Der Befehl erlaubt es auch bei Festplatten Werte zu verändern und dort permanent zu speichern. Dies kann dazu führen, daß das Gerät nicht mehr funktionsfähig ist und low-level-formatiert werden muß. Den Befehl gibt es auch unter dem Namen `scsi_info`.

#### Optionen

<code>-l</code>	Zeigt eine Liste aller SCSI-Gerätenamen an
<code>-i</code>	Informationen über SCSI-Kommando INQUERY
<code>-s</code>	Ausgabe der Seriennummer
<code>-d</code>	Liste der defekten Blöcke
<code>-a</code>	Zeigt fast alle möglichen Informationen an

Eine schnelle Übersicht über alle verfügbaren SCSI-Geräte erhalten Sie mit dem Schalter `-l`.

```

enterprise:~ # scsiinfo -l
/dev/sda /dev/sdb

```

Die Informationen über das INQUERY-Kommando zeigt auch den Hersteller des jeweiligen Gerätes an.

```

enterprise:~ # scsiinfo -i /dev/sda

```

Inquiry command

-----

Relative Address	0
Wide bus 32	0
Wide bus 16	0
Synchronous neg.	0
Linked Commands	0
Command Queueing	0
SftRe	0
Device Type	0
Peripheral Qualifier	0
Removable?	1

Device Type Modifier	0
ISO Version	0
ECMA Version	0
ANSI Version	2
AENC	0
TrmIOP	0
Response Data Format	2
Vendor:	IOMEGA
Product:	ZIP 100
Revision level:	E.0203/27/96

Weitere Informationen, auch über das Verändern von Werten, entnehmen Sie bitte der Manual-Page zu **scsiinfo(8)**.

### 16.5.3 Die sieben Todsünden

Viele Verstöße gegen die SCSI-Richtlinien werden im ersten Moment oft vom System toleriert. Das kann den Eindruck erwecken, daß man die SCSI-Regeln nicht penibel einhalten muß. Das ist ein Eindruck, den Sie sich schleunigst aus dem Kopf schlagen sollten. Sobald weitere Geräte hinzugefügt werden, reagiert Ihr System oft viel empfindlicher. Oft reicht es schon aus, das Kabel etwas anders im Rechner zu verlegen und das vormals so stabile System meckert und muckt rum wie ein pubertärer Jugendlicher.

**Doppelt vergebene ID** Jeder SCSI-Gerät hat seine eigene im Bus einmalige ID. Je nach SCSI-Variante sind die IDs 0 bis 7 oder 0 bis 15 möglich. Normalerweise erhält der SCSI-Hostadapter die ID 7 und die Bootplatte bekommt traditionell die ID 0. Übrigens haben die IDs nichts mit der Reihenfolge am Bus zu tun.

**Nicht oder falsch terminiert** Der SCSI-Bus hat genau zwei Ende, wobei jedes dieser Enden mit einem am besten aktiven Terminator zu terminieren ist. Das Ende des SCSI-Busse ist immer das physikalische Ende des SCSI-Kabels. Ein lose herabhängendes Ende hinter dem letzten Gerät muß mit einem Terminator beendet werden. Da separate Terminatoren selten zu bekommen sind, setzt man auf den letzten Stecker des Kabels meist ein Gerät mit aktiver Terminierung.

**TermPower fehlt** Über die TermPower-Leitung wird der Terminator mit elektrischer Energie versorgt. Diese Energie kann vom Hostadapter gestellt werden, wie auch von anderen dafür ausgelegten Geräten.

**Zu lange Kabel** Die Längen der Kabel müssen penibel eingehalten werden. Gerade bei den Ultra-Varianten ist die Länge bei drei angeschlossenen Geräte auf 3 m und bei mehr angeschlossenen Geräte auf 1,5 m beschränkt. Da alle Kabellängen, externe wie interne, zusammengezählt werden, kann es beim Einsatz von externen Geräten sehr schnell zu Problemen kommen.

**Ungeeignete Kabel** Bei Kabeln sollten Sie auf Qualität achten. Außerdem ist SCSI-Kabel nicht gleich SCSI-Kabel. Kabel, die mit Fast-SCSI wunderbar funktionieren, können beim Einsatz von Ultra-SCSI die merkwürdigsten Ergebnisse erzielen.

**T-Abzweig** Merken Sie sich bitte, daß SCSI ein Bus ist. Das heißt, alle Geräte werden über ein Kabel miteinander verbunden. Abzweigungen in Form von T- oder Y-Stücken sind absolut verboten.

**Bastlerkabel** Ein SCSI-Kabel besteht immer aus einem Stück. Kommen Sie bitte nicht auf die Idee, falls das Kabel mal zu kurz sein sollte, einfach mit einer Pfostenreihe ein Stück anzuflicken. Das SCSI-System wird es Ihnen danken.

## 16.6 USB

Die große Anzahl von Schnittstellen erleichtert nicht gerade die Bedienung des PCs für den Anwender. Um diesen Mißstand der veralteten Schnittstellen abzuhefen wurde der *Universal Serial Bus* kurz **USB** entwickelt. Gedacht als Ersatz für die alten Schnittstelle bietet er eine Anschlußmöglichkeit für eine Vielzahl

von Geräten. Bis zu 127 Geräte können an einem USB-Bus angeschlossen werden. Da dies natürlich nicht direkt geht, gibt es extra USB-Hubs. Diese Hubs, die meist schon direkt in den PCs integriert sind, ermöglichen normalerweise den Anschluß von 2, 4 oder 8 Geräten. Damit man aber trotzdem auf 127 Geräte kommt, ist es möglich die Hubs zu kaskadieren. D. h. an einem Hub kann wieder ein Hub angeschlossen werden. Um falsche Anschlüsse zu vermeiden ist ein USB-Kabel asymmetrisch aufgebaut. Die Stecker für den PC/Hub-Anschluß unterscheidet sich vom Stecker für den Geräteanschluß. Zu allem dem erlaubt USB die Geräte im aktiven Zustand einzubinden und zu entfernen. Dank dem *Hotplug* braucht der Rechner nicht mehr runtergefahren zu werden um ein Gerät einzustöpseln. Stellen Sie sich doch mal vor, daß Sie für jede Benutzung eines USB-Speicherstifts den Rechner neu starten müßten. Trotzdem sollten Sie mit dem Hotplug vorsichtig sein. So sollten Sie vor dem Entfernen eines USB-Speichermediums das Medium auch aus dem Verzeichnisbaum entfernen (unmounten) und niemals während eines Zugriffs das Gerät entfernen. Hotplug hat eben auch seine Grenzen.

USB-Geräten können mit verschiedenen Geschwindigkeiten mit dem Hub kommunizieren. Bei USB 1.1 wird zwischen der *niedrigen* Geschwindigkeit (1,5 MBit/s) für z. B. Tastaturen, Mäuse, Modems und Audiogeräte, und der *vollen* Geschwindigkeit (12 MBit/s) unterschieden, die z. B. für 10-MBit-Netzwerkkarten, USB-Festplatten, Scanner, Digitalkameras und Kartenleser Verwendung findet. Das USB 2.0 ermöglicht dagegen eine *hohe* Geschwindigkeit von 480 MBit/s. Diese Verbesserung war nötig um unter USB sinnvoll USB-Festplatten, 100-MBit-Netzwerkkarten und Videogeräte betreiben zu können.

Natürlich sind die beiden Standards kompatibel zu einander. Allerdings ist die Zusammenarbeit wie bei SCSI beschränkt. Das langsamste Gerät am Hub gibt den Takt vor. Der Einsatz eines USB-2.0-Geräts an einem USB-1.1-Hub erfolgt nur mit einer Geschwindigkeit von 12 MBit/s. Auch ein USB-1.1-Gerät läuft an einem USB-2.0-Hub auch nur mit der *vollen* Geschwindigkeit. Allerdings sorgt es auch dafür, daß alle anderen USB-2.0-Geräte an diesem Hub mit dieser Geschwindigkeit kommunizieren. Damit nicht gleich auch die Geschwindigkeit auf der Leitung vom Hub zum PC reduziert wird, besitzen Hubs nach USB 2.0 einen sogenannten *Transaction Translator*, der *hohe* Geschwindigkeit zwischen Hub und PC ermöglicht. So wird durch den Einsatz eines Gerätes vom Typ USB 1.1 nicht der gesamte USB-Bus ausgebremst.

### 16.6.1 Linux und USB

Unter Linux wird USB seit der Kernelversion 2.4 unterstützt. Dabei sind zwei Methoden zu unterscheiden. Zum einen das von Compaq entwickelte *Open Host Controller Interface (OHCI)* und das von Intel favorisierte *Universal Host Controller Interface (UHCI)*. Die UHCI-Hardware, damit sind vor allem die USB-Controller gemeint, ist einfacher aufgebaut und damit billiger als OHCI-Hardware. Allerdings benötigt Sie aber aufwendigere Treiber. In den heutigen Distributionen wird normalerweise der Treiber für das jeweilige angewandte Verfahren automatisch erkannt. Sollte dies mal nicht der Fall sein, so können Sie über die PCI-Informationen herausfinden, welcher USB-Typ wohl eingesetzt wird.

```
enterprise:~ # cat /proc/pci
PCI devices found:
...
Bus 0, device 7, function 2:
  USB Controller: VIA Technologies, Inc. USB (rev 22).
  IRQ 10.
  Master Capable. Latency=64.
  I/O at 0xc800 [0xc81f].
Bus 0, device 7, function 3:
  USB Controller: VIA Technologies, Inc. USB (#2) (rev 22).
  IRQ 10.
  Master Capable. Latency=64.
  I/O at 0xcc00 [0xcc1f].
...
```

Im obigen Beispiel finden Sie den Eintrag "I/O at 0xc800". Ein Eintrag solchen Typs deutet auf einen UHCI-Controller hin. Würde dort ein Eintrag wie "32 bit memory at 0xc800" stehen, dann handelt es sich um einen OHCI-Controller. Im allgemeinen finden Sie UHCI auf den Hauptplatinen von Intel und VIA. OHCI findet sich dagegen auf Platinen von Compaq und solchen mit ALI- und SIS-Chipsätzen. Sollten Sie

dennoch nicht wissen, welcher USB-Typ vorliegt, dann probieren Sie einfach alle Treiber aus und ermitteln den Richtigen aus den Fehlermeldungen in den Logfiles.

Unter Linux werden durch die Treiber alle USB-Anschlüsse zu einem virtuellen Hub zusammengefaßt. Dies führt zu einer vereinfachten und damit leichter zu verwaltenden Bustopologie. Die angeschlossenen USB-Geräte bekommen eine Gerätenummer zwischen 1 und 127. Dabei wird auch der *Deskriptor* des Gerätes ausgelesen. Er gibt Auskunft über das Gerät und seine USB-Unterstützung. Auch die Zuweisung zu einer Geräteklasse und -unterklasse erfolgt dort.

Auf jeden Fall muß immer der Treiber `usbcore.o` geladen werden. Er stellt den controllerunabhängigen Teil des USB-Subsystemkerns dar. Bei Verwendung von UHCI haben Sie die Wahl zwischen `uhci.o` und `usb-uhci.o`. Bei OHCI ist der Treiber `usb-ohci.o` für die Funktion verantwortlich. Sollten Sie sogar USB 2.0 einsetzen, dann brauchen Sie noch `ehci-hcd.o`. Auch hier müssen Sie trotzdem einen der UHCI- und OHCI-Treiber parallel laufen lassen, da die Steuerung der alten USB-Geräte hierüber erfolgt.

## 16.6.2 Das usbfs

Informationen und der Zugriff auf die USB-Geräte sind über das *USB Device Filesystem* mit dem Namen *usbfs* möglich. Früher lautete die Bezeichnung *usbdevfs*, was aber zu Verwechslungen mit dem Dateisystem *devfs* führte. Das Dateisystem *usbfs* ist wie *proc* ein rein virtuelles Dateisystem. Es wird unter dem Mount-point `/proc/bus/usb` in den Verzeichnisbaum eingehängt.

```
enterprise:~ # grep usb /etc/fstab
usbdevfs          /proc/bus/usb    usbdevfs    noauto          0 0
```

Das Verzeichnis enthält für jeden Bus durchnummerierte Verzeichnisse und die beiden virtuellen Dateien `devices` und `drivers`.

```
enterprise:~ # ls -l /proc/bus/usb/
insgesamt 0
dr-xr-xr-x    1 root    root          0 2004-06-01 07:50 001
dr-xr-xr-x    1 root    root          0 2004-06-01 07:50 002
-r--r--r--    1 root    root          0 2004-06-01 10:16 devices
-r--r--r--    1 root    root          0 2004-06-01 10:16 drivers
enterprise:~ # ls -l /proc/bus/usb/001
insgesamt 2
-rw-r--r--    1 root    root        18 2004-06-01 07:50 001
-rw-r--r--    1 root    root        18 2004-06-01 07:50 002
-rw-r--r--    1 root    root        18 2004-06-01 07:50 003
-rw-r--r--    1 root    root        18 2004-06-01 07:50 004
-rw-r--r--    1 root    root        18 2004-06-01 10:16 005
```

Eine Beschreibung der vorhandenen Geräte findet sich in der Datei `devices`.

```
enterprise:~ # cat /proc/bus/usb/devices
T:  Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc=  0/900 us ( 0%), #Int=  0, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=cc00
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=  8 Iv1=255ms
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc=222/900 us (25%), #Int=  3, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
```

```
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB UHCI Root Hub
S: SerialNumber=c800
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Iv1=255ms
...
```

Vor den jeweiligen Zeilen in der Datei stehen Buchstaben, die die Information in dieser Zeile genauer definieren.

**T** Einleitung der Gerätedefinition und Position des Geräts in der USB-Topologie.

**B** Die Bandbreite des Geräts. Dieser Wert wird nur für USB-Controller angegeben, die auch als Wurzelhubs fungieren.

**D** Beschreibung des Geräts aus der Geräteinformation

**P** Beschreibung des Herstellers aus der Geräteinformation.

**S** Beschreibung des Geräts im Klartext

**C** Informationen über die Konfiguration. Der Asterisk steht dabei für eine aktive Konfiguration.

**I** Informationen über die Schnittstelle.

**E** Informationen über die Endpunkteigenschaften.

Schauen wir uns doch mal alle Zeilen an, die mit 'T' und 'S' beginnen.

```
enterprise:~ # grep "^[TS]" /proc/bus/usb/devices
T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
S: Product=USB UHCI Root Hub
S: SerialNumber=cc00
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
S: Product=USB UHCI Root Hub
S: SerialNumber=c800
T: Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#= 2 Spd=12 MxCh= 4
S: Product=Standard USB Hub
T: Bus=01 Lev=02 Prnt=02 Port=00 Cnt=01 Dev#= 3 Spd=1.5 MxCh= 0
S: Manufacturer=Microsoft
S: Product=Microsoft IntelliMouse with IntelliEye
T: Bus=01 Lev=02 Prnt=02 Port=02 Cnt=02 Dev#= 4 Spd=1.5 MxCh= 0
S: Manufacturer=AIPTEK International Inc.
S: Product=USB Tablet Series Version 1.04
T: Bus=01 Lev=02 Prnt=02 Port=03 Cnt=03 Dev#= 5 Spd=12 MxCh= 0
```

Die Zeilen mit dem Anfangsbuchstabe 'T' leiten eine neue Gerätedefinition ein. In der Zeile stehen folgende Informationen, die einen genauen Einblick in die Topologie der USB-Geräte ermöglicht.

**Bus** Die Nummer des verwendeten USB-Busses. Im obigen Beispiel besitzt der Rechner 2 USB-Busse vom Typ UHCI.

**Lev** Die Entfernung des jeweiligen Geräts von der Wurzel des USB-Bus. Dabei steht der Wert '00' für den Hub an der Wurzel und der Wert '01' für Geräte, die an diesem Hub direkt angeschlossen sind. Den Wert '02' bekommen alle Geräte, die an einem externen Hub angeschlossen sind, der direkt mit dem Wurzelhub verbunden ist. Und dies geht dann in immer so weiter. Im obigen Beispiel gibt es also zwei Wurzelhubs, einen direkt angeschlossen externen Hub und drei weitere Geräte, die an diesem Hub angeschlossen sind.

**Prnt** Die Angabe des Elterngeräts ermöglicht es eine genaue Verbindungsstruktur des USB-Baums erstellen zu können. Hier steht die Device-Nummer des jeweilig genutzten Hubs drin. Zusammen mit der Busnummer ist die Position des Gerätes in der USB-Hierarchie genau definiert.

**Port** Hier wird die Portnummer des Elterngeräts angegeben, an dem das Gerät angeschlossen ist.

**Cnt** Hiermit wird die Nummer des Geräts auf diesem Topologielevel angegeben.

**Dev#** Jedes Gerät an einem USB-Bus bekommt eine eindeutige Nummer von 1 bis 127 zugewiesen, die hier aufgeführt ist. Geräte mit einer gleichen Nummer müssen daher verschiedene Bus-Nummern besitzen.

**Spd** Die Geschwindigkeit, mit der die jeweiligen Geräte kommunizieren, wird hier auch angegeben. Bei dem im obigen Beispiel dargestellten System handelt sich augenscheinlich um die USB-Version 1.1. Die Geräte arbeiten teilweise mit *niedriger* und teilweise mit *voller* Geschwindigkeit.

**MxCh** An diesem Wert können Sie einen Hub erkennen. Er gibt die Anzahl der USB-Geräte an, die an dieses Gerät angeschlossen werden können. Im obigen Beispiel können an die USB-Wurzelhubs jeweils 2 und an den externen Hub 4 USB-Geräte angeschlossen werden.

### 16.6.3 lsusb

Eine einfachere Übersicht über die angeschlossenen USB-Geräte liefert das Programm **lsusb**.

**lsusb** [OPTIONEN]

Das Programm bedient sich dabei der Daten aus dem USB-Geräte-Dateisystem in `/proc/bus/usb`. Um auf alle Informationen zurückgreifen zu können muß dieser Befehl als *root* ausgeführt werden.

#### Optionen

<b>-v</b>	ausführliche Informationen
<b>-vv</b>	noch ausführliche Informationen
<b>-p PFAD</b>	Alternativer PFAD zum USB-Geräte-Dateisystem
<b>-t</b>	Darstellung der USB-Geräte in einer Baumstruktur

Für weitere Informationen konsultieren Sie bitte die Manual-Page zu **lsusb(8)**.

#### Beispiele

Eine Übersicht über die USB-Geräte zeigt der Befehl ohne Parameter. Dabei wird die USB-Busnummer, die Gerätenummer, die Hersteller- und Produkt-ID und die Beschreibung des Geräts angezeigt. Unter Einsatz der Schalter **-v** und **-vv** wird die Ausgabe noch ausführlicher.

```
enterprise:~ # lsusb
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 002: ID 03eb:3301 Atmel Corp. 4-port Hub
Bus 001 Device 003: ID 045e:0025 Microsoft Corp. IntelliEye Mouse
Bus 001 Device 004: ID 08ca:0010 Aiptek International, Inc. Tablet
Bus 001 Device 005: ID 0c76:0005 JMtek, LLC. USBdisk
```

Auch eine Darstellung in einer Baumstruktur ist mit dem Befehl möglich.

```
enterprise:~ # lsusb -t
Bus# 2
'-Dev# 1 Vendor 0x0000 Product 0x0000
Bus# 1
'-Dev# 1 Vendor 0x0000 Product 0x0000
  '-Dev# 2 Vendor 0x03eb Product 0x3301
    |-Dev# 3 Vendor 0x045e Product 0x0025
    |-Dev# 4 Vendor 0x08ca Product 0x0010
    '-Dev# 5 Vendor 0x0c76 Product 0x0005
```



### 16.6.4 usbmodules

Basierend auf dem Befehl `lsusb` (16.6.3) ermittelt das Programm `usbmodules` die passenden Treiber für ein angeschlossenes USB-Gerät.

```
usbmodules --device /proc/bus/usb/NNN/NNN
```

Der Befehl kann z. B. durch ein Shell-Skript wie `/etc/hotplug/usb.agent` genutzt werden um die Treiber für ein USB-Gerät zu bestimmen. Für die Bestimmung der Treiber benutzt der Befehl die Datei `modules.usbmap` im Modulverzeichnis des Kernels.

Um die möglichen Treiber für ein USB-Gerät zu bestimmen, müssen Sie nur die passende Gerätedatei im USB-Geräte-Dateisystem angeben. Dabei entspricht die Nummer des Verzeichnis der USB-Busnummer und die Nummer der Datei der jeweiligen USB-Gerätenummer.

```
enterprise:~ # lsusb
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 002: ID 03eb:3301 Atmel Corp. 4-port Hub
Bus 001 Device 003: ID 045e:0025 Microsoft Corp. IntelliEye Mouse
Bus 001 Device 004: ID 08ca:0010 Aiptek International, Inc. Tablet
Bus 001 Device 005: ID 0c76:0005 JMTEK, LLC. USBdisk
enterprise:~ # usbmodules --device /proc/bus/usb/001/001
usbcore
enterprise:~ # usbmodules --device /proc/bus/usb/001/002
usbcore
enterprise:~ # usbmodules --device /proc/bus/usb/001/003
hid
enterprise:~ # usbmodules --device /proc/bus/usb/001/004
hid
aiptek
enterprise:~ # usbmodules --device /proc/bus/usb/001/005
usb-storage
```

Ausführlichere Informationen zu dem Befehl und zu seinen weiteren Optionen entnehmen Sie bitte der Manual-Page **usbmodules(8)**.

.....  
*Notizen:*

.....

.....  
*Notizen:*

.....

## XVIII

Arbeitsblatt  
Linux und Hardware

## 18.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem separaten Zettel.

- 1** Starten Sie den Rechner neu und beobachten Sie den Startvorgang.
  1. Welche Grafikkarte ist eingebaut?
  2. Wieviel Arbeitsspeicher hat der Rechner?
  3. Wer hat das BIOS hergestellt?
  4. Mit welcher Tastenkombination können Sie ins BIOS wechseln?
- 2** Starten Sie den Rechner wieder neu und wechseln Sie ins BIOS.
  1. Welche Bootreihenfolge ist eingestellt?
  2. Welche Datenträger sind eingetragen?
  3. Wie spät ist es auf dem Rechner?
  4. Welche Interrupts sind den seriellen Schnittstellen zugeordnet?
- 3** Verlassen Sie das BIOS und fahren Sie den Rechner wieder mit Linux hoch.
- 4** Loggen Sie sich als *walter* auf der Konsole 2 ein.
- 5** Löschen Sie mit dem Befehl `rm -rf *` alle unnötigen Dateien.
- 6** Wechseln Sie ihre Identität zu *root*.
- 7** Lassen Sie sich eine Übersicht über die PCI-Geräte ausgeben und speichern Sie diese in der Datei `pcidevices`.
- 8** Bestimmen Sie aus den Angaben dieser Datei die PCI-ID und den Typ der Grafikkarte.
- 9** Lassen Sie sich nun die PCI-Geräte ohne die Klartextbezeichnungen, sondern mit den Hersteller-, Typ- und Modell-IDs ausgeben.
- 10** Welche Hersteller-, Typ- und Modell-ID hat die Graphikkarte "Erazor III LT" und woher bekommen Sie die Information?
- 11** Lassen Sie sich nun eine Übersicht über die belegten Interrupts, IO-Ports und DMA-Kanäle anzeigen. Wo finden Sie diese Informationen?
- 12** Ermitteln Sie Portnummer und IRQ der ersten seriellen Schnittstelle.
- 13** Lassen Sie sich Informationen über die möglichen seriellen Schnittstellen des Systems anzeigen. Wieviele serielle Schnittstellen sind tatsächlich vorhanden.
- 14** Mit welchem Befehl können Sie feststellen, ob ein Terminal eine serielle Schnittstelle ist?
- 15** Bestimmen Sie den Typ ihres USB-Systems.
- 16** Wieviele USB-Controller enthält der Rechner?

## XVIII

Arbeitsblatt  
Linux und Hardware

18.2

**17** Bestimmen Sie die USB-Topologie eines Rechners aus den folgenden Informationen und zeichnen Sie eine passende Baumstruktur.

```
enterprise:~ # grep "[TS]" /proc/bus/usb/devices
T:  Bus=03 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=  2
S:  Product=USB UHCI Root Hub
S:  SerialNumber=dc00
T:  Bus=03 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh=  0
S:  Manufacturer=EPSON
S:  Product=EPSON Scanner 010F
T:  Bus=03 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  3 Spd=12  MxCh=  0
S:  Manufacturer=ALCATEL
S:  Product=Speed Touch USB
S:  SerialNumber=0090D04A33F6
T:  Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=  2
S:  Product=USB UHCI Root Hub
S:  SerialNumber=cc00
T:  Bus=02 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh=  4
S:  Manufacturer=ALCOR
S:  Product=Generic USB Hub
T:  Bus=02 Lev=02 Prnt=02 Port=00 Cnt=01 Dev#=  3 Spd=12  MxCh=  0
S:  Manufacturer=Iomega
S:  Product=USB Zip 100
S:  SerialNumber=0000000000F404B2
T:  Bus=02 Lev=02 Prnt=02 Port=02 Cnt=02 Dev#=  4 Spd=12  MxCh=  0
S:  Manufacturer=ACOMdATA
S:  Product=Mini-FLASH CF
S:  SerialNumber=41C22993C2
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=  2
S:  Product=USB UHCI Root Hub
S:  SerialNumber=c800
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#=  2 Spd=12  MxCh=  4
S:  Product=Standard USB Hub
T:  Bus=01 Lev=02 Prnt=02 Port=00 Cnt=01 Dev#=  3 Spd=1.5 MxCh=  0
S:  Manufacturer=Microsoft
S:  Product=Microsoft IntelliMouse with IntelliEye
T:  Bus=01 Lev=02 Prnt=02 Port=02 Cnt=02 Dev#=  4 Spd=1.5 MxCh=  0
S:  Manufacturer=AIPTEK International Inc.
S:  Product=USB Tablet Series Version 1.04
T:  Bus=01 Lev=02 Prnt=02 Port=03 Cnt=03 Dev#=  5 Spd=12  MxCh=  0
```

.....  
*Notizen:*

.....

## Kapitel 17

# TCP/IP: Geschichte, Adressen und Subnetting

Richtig Spaß macht das Arbeiten mit Rechnern erst, wenn diese miteinander verbunden werden. Schon zwei Rechner, die miteinander Daten austauschen können, werden als *Netzwerk* bezeichnet. Gerade Linux spielt als Server im Netzwerk seine Stärken aus. Deshalb gehören grundlegende Netzwerkkennnisse zum Handwerkszeug des Linuxadministrators dazu.

Das Größte aller Netzwerke ist das Internet. Es verbindet weltweit eine gigantische Zahl von Rechnern miteinander und ist einer der entscheidenden Motoren der EDV-Entwicklung in den letzten Jahren gewesen. Die Basis des Internet bildet dabei eine Reihe von Protokollen, die unter dem Begriff TCP/IP zusammengefaßt wurden.

### 17.1 Einführung

Eigentlich waren ursprünglich mit TCP/IP die Protokolle TCP (*Transmission Control Protocol*) und IP (*Internet Protocol*) gemeint. Inzwischen steht TCP/IP für eine Familie von Kommunikationsprotokollen zur paketvermittelten Übertragung von Daten zwischen Rechnern und deren Anwendungsdiensten.

Es werden neben den eigentlichen Protokollen zur Datenübertragung (TCP, IP, UDP, ICMP, DNS usw.) oder zur Konfiguration der Netzknoten (SNMP) auch Protokolle für Anwendungen zur Verfügung gestellt, wie z. B. für

- eMail (SMTP, POP3, IMAP),
- Terminaldienste (TELNET, SSH),
- Dateitransfer (FTP, NFS) und
- Webseiten (HTTP).

Weitere Protokolle des Internets (z. B. HTML, CSS, XML) hängend direkt oder indirekt mit TCP/IP zusammen.

#### 17.1.1 Requests for Comments (RFC)

Die Standards und Protokolle des Internets werden in sogenannten *Request for Comments* oder kurz *RFCs* festgelegt.

Die RFCs sind eine Sammlung von Dokumenten von technischen und organisatorischen Festlegungen zum Internet (früher ARPANET). Die einzelnen RFCs diskutieren dabei die vielen Aspekte der Rechnernetzwerke. Dies umfaßt Protokolle, Abläufe, Programme und Konzepte, wie auch Meinungen und manchmal auch Humor.

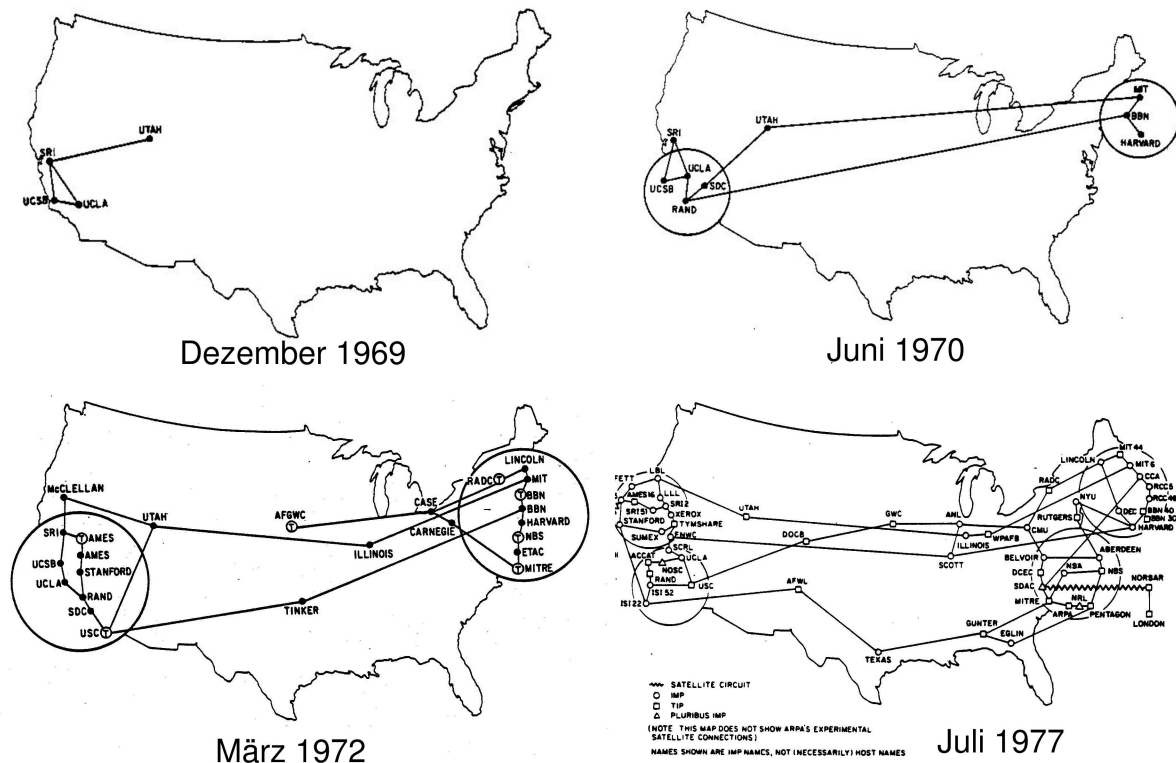


Abbildung 17.1: Ausbreitung des ARPANET von 1969 bis 1977

Die offiziellen Spezifikationen werden von der *Internet Engineering Task Force (IETF)* und der *Internet Engineering Steering Group (IESG)* festgelegt. Diese Spezifikationen werden über RFCs festgelegt und veröffentlicht. Die Art und Weise der Veröffentlichung von RFS hat daher einen großen Einfluß auf den Entwicklungsprozeß der Internet-Standards. So werden RFCs zuerst als *Internet Drafts* veröffentlicht. Drafts sind bereits veröffentlichte Dokumente, die sich aber noch in der Entwicklung befinden und sich daher noch ändern können, wie z. B. auch dieses Dokument.

Verantwortlich für die Veröffentlichung und letzte redaktionelle Überarbeitung der RFCs ist der *RFC-Editor* (<http://www.rfc-editor.org>). Er verwaltet auch die Hauptdatei der RFCs, den sogenannten "RFC Index".

Erhältlich sind die RFCs über die Webseiten und FTP-Server der einzelnen NICs (*Network Information Center*). In Deutschland hat die Aufgabe die DENIC, das NIC für die Toplevel-Domain .de, übernommen. Unter <http://www.denic.de> oder <ftp://ftp.denic.de/pub/rfc> werden die Dokumente zur Verfügung gestellt.

### 17.1.2 ARPA Kadabra: Die Geschichte des Internet

**1957** Die UDSSR startet den ersten Satelliten SPUTNIK in eine Weltumlaufbahn. Die USA reagieren und gründen das ARPA (*Advanced Research Projects Agency*), eine Abteilung des US Department of Defense<sup>1</sup> (DoD). Die Aufgabe der ARPA ist es, neue Technologien im Bereich Kommunikation und Datenübertragung zu entwickeln, um der USA einen technischen Vorsprung gegenüber der UDSSR zu verschaffen.

**1966** Der erste Plan fürs ARPANET entsteht.

**1967** Ein Vortrag von Larry Roberts im April beschreibt das erste Design des entstehenden ARPANET.

**1968** Das Information Processing Techniques Office, eine Abteilung des DoD, startet ein Projekt zur groß-

<sup>1</sup>Das Verteidigungsministerium der USA

räumigen Vernetzung von Rechnersystemen (Wide Area Network WAN). Ziel war es, die Kommunikation im Kriegsfall auch bei Ausfall von Teilen des Netzwerks aufrecht zu erhalten.

**1969** Unter Federführung des ARPA wurde ein weiteres Projekt zur Rechnervernetzung ins Leben gerufen.

Über gemietete Leitungen wird ein paketvermitteltes Netzwerk eingerichtet. Die Kommunikation erfolgt über Knotenrechner, sogenannte Interface Messages Processors (*IMP*). Zuerst wurden vier solcher IMPs an der University of California Los Angeles (*UCLA*), University of California Santa Barbara (*UCSB*), University of Utah (*UTAH*) und dem Stanford Research Institute (*SRI*) eingerichtet.

Das sogenannte ARPANET lief unter dem Network Control Program (*NCP*), einem Vorläufer des TCP/IP. Als Dienste standen Terminalsitzungen (ähnlich *TELNET*) und Dateitransfer (ähnlich *FTP*), der auch die Aufgaben der elektronischen Post übernahm, zur Verfügung.

Das erste Datenpaket versucht im Oktober Charley Kline von *UCLA* nach *SRI* zu senden. Der erste totale Systemabsturz folgt, als er den Buchstaben G im Wort *LOGIN* eingibt.

**1970** Die Eliteuniversitäten Harvard und MIT schließen sich dem ARPANET an.

**1972** Das ARPA wird ins DARPA (*Defence Advanced Research Projects Agency*) umbenannt. Es besteht nun aus 20 IMPs und 50 Hosts.

Zur gleichen Zeit nimmt die Internet Working Group ihre Arbeit auf, um Standards für die Verbindung von Netzen zu erarbeiten.

**1974** Das NCP Protokoll entspricht nicht mehr den Anforderungen des Netzes. Vincet Cerf (Stanford University) und Robert Kahn (DARPA) formulieren die Grundzüge der TCP/IP-Architektur.

Die Arbeit die Protokolle in Berkley Unix einzubauen beginnt.

Es wird versucht das ARPANET zu privatisieren. Durch die Privatisierung hoffte man, die kommerzielle Netzwerkindustrie anzuregen und auch außeruniversitäre und nichtmilitärische Organisationen für die Teilnahme am Netz zu gewinnen. AT&T zeigte an einer Übernahme des ARPANET jedoch kein Interesse, weshalb die ARPA das Netz selbst weiterführte.

**1979** Die National Science Foundation (*NSF*) errichtet in den USA ein landesweites Netz auf der Basis von TCP/IP. Zunächst wurde das *Computer Sciences Net (CSNET)* eingerichtet, das die Informatik-Fakultäten des Landes verband.

**1980** Berkley Unix wird für die neuen DEC-Rechner der VAX-Familie entwickelt.

**1983** Die Unix Version BSD 4.2 (*Berkley System Distribution*) wird mit einer TCP/IP Implementierung vorgestellt. Da sich viele Universitäten VAX-Rechner ohne Betriebssystem angeschafft haben, greifen diese gerne auf das frei kopierbare Betriebssystem zurück. BSD verbreitet sich rasant und TCP/IP setzt sich als de facto Standard durch.

Das Defense Communications Agency (*DCA*) des DoD übernimmt die Verwaltung des ARPANETs, das inzwischen auf 200 IMPs angewachsen ist. Die 160 IMPs des militärischen Teils werden abgetrennt und ins MILNET überführt. Dies Netz wird durch streng kontrollierte Gateways vom restlichen ARPANET getrennt.

**1984** Allmählich begannen auch die Wissenschaftler anderer Disziplinen, in immer größerem Umfang Zugang zum CSNET zu verlangen. Schließlich kulminierten die Netzwerkaktivitäten der NSF dem eigenen landesweiten Forschungsnetz NSFNET.

**1988** Die Kapazität des NSFNET übersteigt die des fast zwei Jahrzehnte älteren ARPANET bei weitem.

**1990** Die bürokratischen und finanziellen Möglichkeiten von ARPA wurden durch das ARPANET immer stärker strapaziert. Die ARPA entschied deshalb, daß das ARPANET überflüssig geworden war, und deinstallierte die ARPANET-Hardware. Das NSFNET übernahm alle Funktionen des ARPANET. Das Internet besteht nun aus 3000 Netzen und 200.000 Hosts.

**1992** Der millionste Host wird angeschlossen.

**1993** Ca. 15 Millionen Benutzer haben Zugang zum Internet.

Binär	Oktal	Dezimal	Hexadezimal
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

Tabelle 17.1: Zahlensysteme im Vergleich

## 17.2 Zahlensysteme

Normalerweise benutzen wir Sterblichen das auf der Zahl 10 basierenden Dezimalsystem. Das liegt einfach daran, daß wir zehn Finger besitzen. Und weil wir es von Kindheit an benutzen, kommt uns das Rechnen in diesem System mehr oder minder einfach vor. Der Computer findet allerdings, da er nur zwischen “An” und “Aus” oder anders gesagt “Eins” und “Null” unterscheiden kann, ein Zahlensystem auf der Basis der Zahl 2 besser. Neben dem Binär-System (Basis 2) treffen Sie in der EDV auch auf das Oktal-System (Basis 8) und das Hexadezimal-System (Basis 16).

### 17.2.1 Dezimalsystem

Das allseits bekannte Dezimalsystem basiert auf der Zahl 10. Es gibt die Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9. Reichen die Ziffern nicht aus, dann nimmt man eine weitere Stelle, die dann das Zehnfache der angegebenen Zahl bezeichnet, hinzu. Die nächste Stelle bedeutet dann das Hundertfache u.s.w. Es gilt dann z. B.

$$531 = 5 \cdot 100 + 3 \cdot 10 + 1 \cdot 1 = 5 \cdot 10^2 + 3 \cdot 10^1 + 1 \cdot 10^0 \quad (17.1)$$

oder

$$1024 = 1 \cdot 1000 + 0 \cdot 100 + 2 \cdot 10 + 4 \cdot 1 = 1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 \quad (17.2)$$

### 17.2.2 Binärsystem

Das Binärsystem basiert auf der Zahl 2. Es gibt also zwei verschiedene Ziffern: 0 und 1. Das hat zur Folge, daß bei der Darstellung der Zahl “2” schon zwei Ziffern benötigt werden: 10. Denn auch hier wird wie in allen hier genannten Zahlensysteme bei der Überschreitung des Wertebereichs einfach eine neue Ziffer links angefügt. Hier bedeutet die nächste Stelle allerdings das Zweifache, dann das Vierfache u.s.w. Es gilt dann z. B.

$$1010_{\text{b}} = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 10 \quad (17.3)$$

oder

$$1001_{\text{b}} = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9 \quad (17.4)$$

In der EDV haben Sie es meistens mit achtstelligen Binärzahlen (Byte oder Oktett) zu tun. Sie sollten sich daher in diesem Zahlenraum auskennen.



Dezimal	128	64	32	16	8	4	2	1	Rechnung
42	0	0	1	0	1	0	1	0	$32 + 8 + 2 = 42$
127	0	1	1	1	1	1	1	1	$64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$
81	0	1	0	1	0	0	0	1	$64 + 16 + 1 = 81$

Das Umrechnen von Binär nach Dezimal ist relativ einfach. Sie müssen nur die jeweiligen “Spaltenzahlen” aufsummieren, wo eine “1” steht. Das Rechnen von Dezimal nach Binär ist etwas schwerer. Versuchen wir es mal mit der Zahl 187.

128	geht in	187	<b>1</b>	mal. Es bleibt ein Rest von	59.
64	geht in	59	<b>0</b>	mal. Es bleibt ein Rest von	59.
32	geht in	59	<b>1</b>	mal. Es bleibt ein Rest von	27.
16	geht in	27	<b>1</b>	mal. Es bleibt ein Rest von	11.
8	geht in	11	<b>1</b>	mal. Es bleibt ein Rest von	3.
4	geht in	3	<b>0</b>	mal. Es bleibt ein Rest von	3.
2	geht in	3	<b>1</b>	mal. Es bleibt ein Rest von	1.
1	geht in	1	<b>1</b>	mal. Es bleibt ein Rest von	0.

Die dezimale Zahl 187 lautet binär dargestellt **10111011**.

### 17.2.3 Oktalsystem

Das Binärsystem basiert auf der Zahl 8. Es gibt also acht verschiedene Ziffern: 0, 1, 2, 3, 4, 5, 6 und 7. Für die Darstellung von Zahlen größer als sieben wird einfach eine neue Ziffer links angefügt. Hier bedeutet die nächste Stelle das Achtfache, dann das Vierundsechzigfache u.s.w. Es gilt z. B.

$$157_{\text{o}} = 1 \cdot 64 + 5 \cdot 8 + 7 \cdot 1 = 1 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 = 111 \quad (17.5)$$

oder

$$234_{\text{o}} = 2 \cdot 64 + 3 \cdot 8 + 4 \cdot 1 = 2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 156 \quad (17.6)$$

Die Umrechnung von Binär nach Oktal und umgekehrt ist einfach, da drei binäre Ziffern immer einer oktalen Ziffer entsprechen.

Dezimal	128	64	32	16	8	4	2	1	Oktal
	2	1	4	2	1	4	2	1	
42	0	0	1	0	1	0	1	0	00b = 0   101b = 5   010b = 2 → 52o
127	0	1	1	1	1	1	1	1	01b = 1   111b = 7   111b = 7 → 177o
81	0	1	0	1	0	0	0	1	01b = 1   010b = 2   001b = 1 → 121o

### 17.2.4 Hexadezimalsystem

Dem Hexadezimalsystem liegt die Basis 16 zugrunde. Da ja nur zehn Ziffern (0, 1, 2, 3, 4, 5, 6, 7, 8 und 9) zur Verfügung stehen, hat man sich für die weiteren Ziffern mit den ersten sechs Buchstaben (A, B, C, D, E und F) des Alphabets beholfen. Auch hier werden zur Darstellung größerer Zahlen weitere Stellen hinzugefügt.

$$42_{\text{h}} = 4 \cdot 16 + 2 \cdot 1 = 4 \cdot 16^1 + 2 \cdot 16^0 = 66 \quad (17.7)$$

oder

$$AF_{\text{h}} = 10 \cdot 16 + 15 \cdot 1 = 10 \cdot 16^1 + 15 \cdot 16^0 = 175 \quad (17.8)$$

Die Umrechnung von Binär nach Hexadezimal und umgekehrt ist genauso einfach wie bei den Oktalzahlen. Da vier binäre Ziffern immer einer hexadezimalen Ziffer entsprechen, eignen sich die hexadezimalen Ziffern sogar noch besser für die Darstellung von einem Byte, dessen Wert sich immer durch zwei hexadezimale Ziffern beschreiben lässt.

	8	16	24
A	0	Netzwerkanteil 7 bit = 128	Hostanteil 24 bit = 16.777.214
B	10	Netzwerkanteil 14 bit = 16.384	Hostanteil 16 bit = 65.534
C	110	Netzwerkanteil 21 bit = 2.097.152	Hostanteil 8 bit = 256
D	1110	Multicast	
E	1111	Reserviert für zukünftige Benutzung	

Abbildung 17.2: Netzwerkklassen

Dezimal	128	64	32	16	8	4	2	1	Hexadezimal
	8	4	2	1	8	4	2	1	
42	0	0	1	0	1	0	1	0	0010 = 2    1010 = 9 → 29h
127	0	1	1	1	1	1	1	1	0111 = 7    1111 = 15 → 7Fh
81	0	1	0	1	0	0	0	1	0101 = 5    0001 = 1 → 51h

## 17.3 Host-Adressierung mit IP

Wenn Sie einen Menschen anrufen wollen, brauchen Sie seine Telefonnummer. Natürlich können Sie eventuell unter dieser Nummer auch seinen Lebensabschnittsgefährten erreichen. Also eine Telefonnummer für zwei Menschen. Beim Telefonanschluß ist die Sache aber eindeutiger. Jeder Telefonanschluß besitzt mindestens eine weltweit eindeutige Nummer (wenn man die Landes- und Ortsvorwahl mit hinzufügt). Und so landen wir auch, wenn wir uns nicht verwählen, bei dem Anschluß den wir haben wollen. Bei den Rechnern im Internet ist es genauso. Auch hier wird jedem Rechner im Internet eine weltweit eindeutige IP-Nummer bzw. IP-Adresse zugewiesen. IP stellt dafür 32 Bit zur Verfügung. Das macht  $2^{32} = 4.294.967.296$  mögliche Adressen.

Allerdings stehen nicht alle diese Adressen auch zur Verfügung. Durch Verwaltung und Unterteilung in Netze fallen einige Adressen weg.

Da kaum einer in der Lage ist 32 binäre Ziffern sich zu merken und die Darstellung als eine Dezimalzahl auch Probleme bereitet, hat man die Adresse in vier Oktette (Bytes) aufgeteilt und stellt diese durch ihre Dezimalzahl dar. Die Dezimalzahlen können dabei Werte zwischen 0 und 255 annehmen.

11011001	01011001	01000110	00111100
217	89	70	60

### 17.3.1 Netzwerkklassen

Um die Hosts besser gruppieren zu können, werden Sie zu Netzen zusammengefügt. Die ersten Bits entscheiden zu welchem Netzwerk der Rechner gehört und die restlichen Bits geben an, welche Rechnernummer er in diesem Netz hat.

Netzwerkanteil				Hostanteil		
11011001	01011001	01000111	00101101			
217	.	89	.	71	.	45

## A bis E

Es wurden dann fünf Klassen (A bis E) von Netzwerken geschaffen. Diese unterscheiden sich in den ersten Bits, den sogenannten charakteristischen Bits.

**A-Klasse** Adressen aus A-Klasse-Netzen beginnen immer mit dem Bit “0”. Für die Adressierung des Netzwerks stehen 8 Bit zur Verfügung und damit 24 Bit für die Adresse des Hosts. Da das erste Bit vorgegeben ist, bleiben für unterschiedliche Adressen 7 Bit über. Von den so nutzbaren 128 Adressen fallen die Netze 0.0.0.0 und 127.0.0.0 wegen besonderer Bedeutung weg. Es stehen also 126 A-Klasse-Netze mit 16.777.214 Adressen zur Verfügung. Stolz Besitzer eines Klasse-A-Netzwerks sind IBM (9.0.0.0), HP (16.0.0.0) und Apple (17.0.0.0).

**B-Klasse** Alle Adressen aus einem B-Klasse-Netz beginnen immer mit der Bitfolge “10”. Für die Adressierung des Netzwerks stehen 16 Bit zur Verfügung, wobei zwei Bit wieder fest vorgegeben sind. Es stehen also  $2^{14} = 16.384$  Netze mit jeweils 65.536 Adressen zur Verfügung.

**C-Klasse** Die Netze mit den wenigsten Rechnern kommen aus dem C-Klasse-Netz. Die Bitfolge “110” ist charakteristisch für ein C-Klasse-Netz. Für den Netzwerkanteil der Adressen sind 24 Bit adressiert. Für den Hostanteil bleiben daher nur 256 Adressen übrig. Allerdings gibt es auch  $2^{21} = 2.097.152$  C-Klasse-Netze.

**D-Klasse** Das D-Klasse-Netz ist nicht für Rechnernetze gedacht sondern dient zur Ansprache von mehreren Geräten gleichzeitig im Netz. Dies wird auch als “multicast” bezeichnet. Alle D-Klasse-Adressen beginnen mit der Bitfolge “1110”.

**E-Klasse** Auch das E-Klasse-Netz ist nicht für Netzwerke gedacht, sondern ist für zukünftige Verwendung und Experimente reserviert. Es umfaßt die restlichen Adressen (Bitfolge “1111”).

## Beispiel

Gegeben ist die IP-Adresse 42.0.8.15. Als Binärzahl ausgedrückt lautet Sie 00101010 00000000 00001000 00001111. Das erste Bit ist “0” und damit gehört die Adresse zu einem A-Klasse-Netz.

Die Adresse 170.42.10.2 lautet in Binärform 10101010 00101010 00001010 00000010. Die ersten beiden Bits sind “10”, damit handelt es sich eindeutig um ein B-Klasse-Netz.

Bei der Adresse 222.222.222.222, die in Binärform als 11011110 11011110 11011110 11011110 dargestellt wird, handelt es sich um ein C-Klasse-Netz, da die ersten drei Bits gleich “110” sind.

## Und noch mal zusammengefaßt

Sie müssen sich aber nicht unbedingt mit Binärzahlen rumschlagen um die Netzwerk-Klasse zu bestimmen. Entscheidend sind nur die Bits des ersten Bytes. Daher können Sie auch schon an dem ersten Oktett erkennen, zu welcher Klasse ein Netz gehört.

Klasse	Bits	Binär	Dezimal
A	0	00000001 – 011111110	1 – 126
B	10	10000000 – 101111111	128 – 191
C	110	11000000 – 110111111	192 – 223
D	1110	11100000 – 111011111	224 – 239
E	1111	11110000 – 111111111	240 – 255

### 17.3.2 Besondere Adressen

Ein paar der IP-Adressen stehen aber nicht für Rechner zur Verfügung sondern haben eine besondere Bedeutung. Ich verwende für die Darstellung dieser Adressen eine neue Art, die in RFC 1700 definiert worden ist.

Eine IP-Adresse wird in der Form { Netzwerkanteil , Hostanteil } dargestellt. Die Zahl 0 bei Netzwerk- und Hostanteil heißt, daß alle Bits auf "0" stehen. Die Zahl "-1" steht für eine Binärzahl, bei der alle Bits auf "1" stehen.

#### Netzwerkadresse

Für die Benennung des Netzwerks wurde die IP-Adresse ausgewählt, bei der alle Host-Bits auf "0" stehen, also die erste Adresse im Netzwerkadressraum.

$$\{ \text{Netzwerkanteil} , 0 \}$$

Das sind z. B. die Adressen 42.0.0.0, 170.42.0.0 und 222.222.222.0.

#### Direkter Broadcast

Damit alle Rechner in einem Netz gleichzeitig angesprochen werden können, wurde der Broadcast entwickelt. Es wurde die letzte Adresse des Netzwerkadressraums ausgewählt bei der alle Bits des Hostanteils auf "1" gesetzt sind.

$$\{ \text{Netzwerkanteil} , -1 \}$$

Das sind z. B. die Adressen 42.255.255.255, 170.42.255.255 und 222.222.222.255.

Damit fallen aus jedem Netzwerk immer zwei Adressen raus. So können in einem C-Klasse-Netzwerk jetzt nur noch 254 Hosts adressiert werden.

#### Limitierter Broadcast: 255.255.255.255

Der limitierte Broadcast ist auch ein Rundruf an alle Rechner eines Netzes. Im Gegensatz zum direkten Broadcast kann aber kein spezielles Netz angesprochen werden, sondern das Ziel sind alle Rechner im Netz des Senderechners. Für den limitierten Broadcast wurde die Adresse **255.255.255.255** gewählt.

$$\{ -1 , -1 \}$$

#### Ohne Adresse: 0.0.0.0

Beim Einsatz eines BOOTP- oder DHCP-Servers im Netz braucht den Hosts keine feste IP-Adresse zugewiesen werden, sondern diese erhalten bei jedem Booten eine Adresse von den Servern zugewiesen. Zum Zeitpunkt der Anfrage an die Server besitzt aber der Host noch gar keine IP-Nummer. Da aber das IP-Protokoll eine Absendeadresse haben möchte, wurde die IP-Nummer **0.0.0.0** als Adresse für Hosts ohne definierte IP-Adresse festgelegt.

$$\{ 0 , 0 \}$$

#### Host im eigenen Netz

Soll ein Host im eigenen Netz angesprochen werden, so braucht das Netzwerk nicht mit angegeben werden und die Bits des Netzwerkanteils können auf "0" gesetzt werden.

$$\{ 0 , \text{Hostanteil} \}$$

Das sind z. B. die Adressen 0.200.100.20, 0.0.8.15 und 0.0.0.42.

## Software Loop-Back

Die Loop-Back Adresse bezeichnet den eigenen Rechner. Bei Angabe dieser Adresse als Zieladresse werden die Daten durch alle Schichten der Netzwerkprotokolle bis auf die Hardware-Schicht geschickt. Dadurch kann ein Netz simuliert werden ohne daß Netzwerkhardware vorhanden ist. Für dieses speziellen Adressen wurde das 127er Netz ausgewählt.

{ 127 , Alle Werte }

Die am meisten benutzte Loop-Back-Adresse ist die 127.0.0.1, die oft mit dem Namen *localhost* verbunden ist..

### 17.3.3 Private Netze

TCP/IP sieht vor, daß jeder Rechner weltweit eine eindeutige Adresse besitzen soll. Damit diese auch wirklich einmalig ist, wird sie von zentralen Vergabestellen vergeben. Eine weltweite eindeutige Adresse ist aber nur wichtig, wenn der Rechner ans Internet angeschlossen ist. In nicht ans Internet angeschlossenen Netzen kann im Prinzip jede Nummer frei vergeben werden. Was ist aber, wenn das Netz irgendwann doch angeschlossen wird. Deshalb wurde in der RFC *Address Allocation for Private Internets* (RFC 1597, RFC 1627 und RFC 1918) bestimmte Nummerbereiche festgelegt, die für private Netze zur Verfügung stehen. Die Router im Internet haben die Anweisung, diese Nummer nicht ins Internet zu routen. Daher kann es zu keinem Konflikt mit anderen bestehenden Adresse kommen. Insgesamt wurden drei Adreßbereiche definiert.

1. 1 A-Klasse-Netz: 10.0.0.0
2. 16 B-Klasse-Netze: 172.16.0.0 - 172.31.0.0
3. 256 C-Klasse-Netze: 192.168.0.0 - 192.168.255.0

Auszug aus der RIPE-Datenbank über das Netz 10.0.0.0

```
inetnum:      10.0.0.0 - 10.255.255.255
netname:      IANA-ABLK-RESERVED1
descr:        Class A address space for private internets
descr:        See http://www.ripe.net/db/rfc1918.html for details
country:      NL
admin-c:      RFC1918-RIPE
tech-c:       RFC1918-RIPE
status:       ALLOCATED UNSPECIFIED
remarks:      Country is really worldwide
remarks:      This network should never be routed outside an enterprise
remarks:      See RFC1918 for further information
```

## 17.4 Subnetting

Oft besteht der Wunsch bestehende Netze aufzuteilen. Dies erfolgt durch das sogenannte *Subnetting*. Subnetting versetzt den Administrator eines großen logischen Netzwerks in die Lage, dieses in kleine logische Netzwerke aufzuteilen. Die Gründe dafür können technischer, administrativer oder organisatorischer Art sein. Da wären z. B.

1. Trennen von Netzwerken unterschiedlicher Topologie
2. Trennen von Netzwerken nach Standorten, Gebäuden und Etagen
3. Trennen von Netzwerken nach Abteilungen und Bereichen
4. Trennen von sensiblen Bereichen vom Hauptnetz

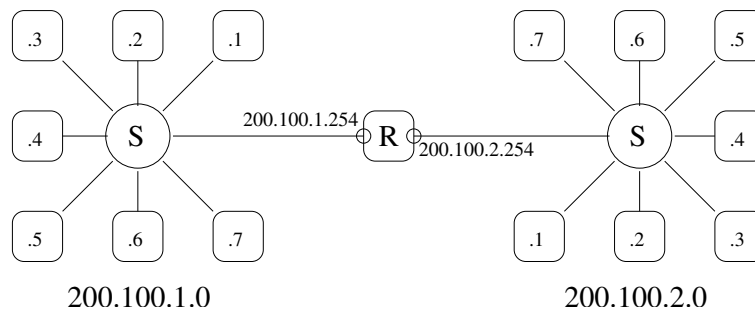


Abbildung 17.3: Zwei Netze werden mit einem Router verbunden.

5. Trennen des Netzwerks in logische Arbeitsgruppen
6. Trennen des Netzwerks zur Reduzierung des Verkehrsaufkommens

Dem Protokoll TCP/IP ist es egal, auf welcher Weise die Daten von einem Rechner zum anderen kommen. Dies kann auf Basis von Ethernet, Fast Ethernet, Gigabit Ethernet, Token-Ring oder durch eine Modem-, ISDN- oder DSL-Verbindung erfolgen. Hier bietet es sich eben an, jeder der Topologien ein eigenes Netz zuzuweisen.

Auch räumliche Faktoren spielen eine Rolle bei der Bildung von Netzen. Die Beschränkung eines Netzes auf einen Firmensitz, ein Gebäude oder sogar ein Stockwerk hängt oft auch mit der eingesetzten Hardware und der Verbindung untereinander zusammen. Zwar könnten diese auch innerhalb eines Netzes erfolgen, ein Problem stellt da die Verwaltung der Routing-Tabellen in den Routern da, die der Wegweiser für die einzelnen Päckchen sind. Hier müßten dann für jeden einzelnen Rechner Einträge gemacht werden, wo er zu finden ist. Befinden sich aber alle Rechner in einem Netz, so reicht die Angabe des Netzes um dem Datenpaket den Weg zu weisen.

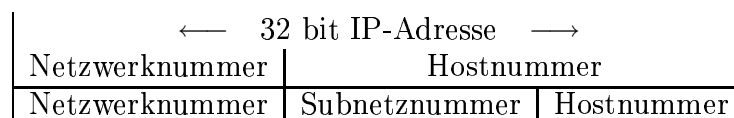
Neben physikalischen Faktoren können auch logische oder organisatorische Gründe vorliegen. So sollen z. B. die Rechner einer Abteilung zusammen in einem Netz sein oder jeder Bereich hat sein eigenes Netz mit eigenem Administrator und Servern. Dadurch ist es auch möglich Zugriffsbeschränkungen einzurichten.

Die Einschränkung des Zugriffs ist z. B. auch ein Grund zum Trennen von Netzen. So bilden z. B. wichtige Server ein eigenes Netz, dessen Zugang durch spezielle Router überwacht und gefiltert wird. Auch die Aufteilung in vom Internet zugängliche und nicht zugängliche Bereiche erhöht die Sicherheit des Firmennetzes.

Ein weiteres Problem ergibt sich beim Ethernet dadurch, daß jeder Rechnern zu jeder Zeit ein Datenpaket senden kann. Dadurch kann es zu Kollisionen kommen, deren Wahrscheinlichkeit sich mit jedem weiteren Rechner im Netz erhöht und damit die effektive Übertragungsrate senkt. Gerade Netze mit einer Verteilung über einen HUB sollten nicht zu groß werden, um die Netzwerklast klein zu halten.

### 17.4.1 Technik

Bisher wurde eine IP-Adresse in einen Netzwerkanteil (Netzwerknummer) und einen Hostanteil (Hostnummer) aufgeteilt. Dabei ergab sich der Anteil der Netzwerknummer an der IP-Adresse durch die Klasse des Netzwerks. Die Idee des Subnettings ist es, den vorhandenen Hostanteil auch noch in zwei Teile aufzuspalten. Nämlich in einen Subnetzanteil (Subnetznummer) und einen Hostanteil (Hostnummer). Dadurch wird praktisch der Netzwerkanteil der Adresse vergrößert und der Hostanteil verkleinert. Es stehen nun mehrere kleinere Netze zur Verfügung.



Um z. B. ein C-Klasse-Netz in 8 Subnetze aufzuteilen, brauchen wir 3 Bit (Zahlen von 0 bis 7) zusätzlich zur Netzwerkadresse. Der Hostanteil reduziert sich dann auf 5 Bit ( $2^5 - 2 = 30$  Rechner). Schauen wir uns doch mal die Adresse 217.89.70.60 als Beispiel für ein solches Netz an.

Größe in Bit	Binär	Dezimal
0	00000000	0
1	10000000	128
2	11000000	192
3	11100000	224
4	11110000	240
5	11111000	248
6	11111100	252
7	11111110	254
8	11111111	255

Tabelle 17.2: Mögliche Nummern für die Subnetzmaske

	← 32 bit IP-Adresse →			
	Netzwerknummer	Subnetznummer	Hostnummer	
Adresse	11011001 01011001 01000110	001	11100	217.89.70.60
Netzwerknummer	11011001 01011001 01000110	000	00000	217.89.70.0
Subnetznummer	00000000 00000000 00000000	001	00000	0.0.0.32
Hostnummer	00000000 00000000 00000000	000	11100	0.0.0.28

### 17.4.2 Subnetzmaske

Aus einer IP-Adresse kann entnommen werden, aus welche Netzwerkkategorie sie stammt. Informationen über die Subnetze enthält sie nicht. Deswegen wird für die Hosts und Router eine weitere Angabe benötigt. Durch die Subnetzmaske (engl. *subnetmask*) wird dem System mitgeteilt, welchen Teil der IP-Adresse es als Netzwerkanteil und welchen Teil als Hostanteil zu werten hat.

Die Subnetzmaske ist eine 32 bit-Zahl. Jedes Bit, das auf 1 steht, bedeutet, daß dieses Bit der IP-Adresse als Netzwerkanteil zu betrachten ist. Umgekehrt folgt natürlich sofort daraus, daß die in der Subnetzmaske auf 0 gesetzten Bits den Hostanteil der IP-Adresse markieren.

Konform zu den IP-Adressen wird sie in vier Dezimalblöcken dargestellt. Nehmen wir das obige Beispiel als Grundlage, so würden die Angaben für ein System lauten:

	← 32 bit Adresse →			
	Netzwerk	Subnetz	Host	
Adresse	11011001 01011001 01000110	001	11100	217.89.70.60
Subnetzmaske	11111111 11111111 11111111	111	00000	255.255.255.224
Netzwerkanteil	11011001 01011001 01000110	001	00000	217.89.70.32
Hostanteil	00000000 00000000 00000000	000	11100	0.0.0.28

Das System hat also die IP-Adresse 217.89.70.60 und die Subnetzmaske 255.255.255.224. Es befindet sich im Netz 217.89.70.32 und ist dort der Host mit der Nummer 28.

Für die Netzwerkklassen A, B und C gibt es die Standardsubnetzmasken.

A-Klasse: 255.0.0.0  
 B-Klasse: 255.255.0.0  
 C-Klasse: 255.255.255.0

Theoretisch könnte es auch Subnetzmasken in der Art 127.33.51.27 geben. In der Praxis werden aber nur Subnetzmasken verwendet, die den Netzwerkanteil links und den Hostanteil rechts haben und die Bits nicht miteinander vermischen. Es werden also nur Nummern verwendet, wie sie in Tabelle 17.2 aufgeführt werden.

### 17.4.3 Beispiele fürs Subnetting

#### Subnetz in einem B-Klasse-Netz

IP-Adresse: 149.18.3.70  
 Subnetzmaske: 255.255.255.0

Es handelt sich bei der Adresse um eine Nummer aus einem B-Klasse-Netzwerk, die sich in einem Subnetz von der Größe eines C-Klasse-Netzes befindet. Dadurch ergibt sich sofort die klassische Netzwerknummer als 149.18.0.0, da die ersten zwei Bytes in einem B-Klasse-Netzwerk das Netzwerk bezeichnen. Aus der Subnetzmaske folgt dann, daß alle Bits des dritten Adressbytes das Subnetz definieren. Also folgt sofort die Subnetznummer mit 0.0.3.0. Beides zusammen ergibt dann die reale Netzwerknummer bzw. Subnetzwerknummer mit 149.18.3.0. Es bleibt als nur noch die Adresse 0.0.0.70 für den Host übrig.

Sie sollten sich merken, daß die Summe aus Netzwerknummer, Subnetznummer und Hostnummer die IP-Adresse ist.

Netzwerknummer		149. 18. 0. 0
Subnetznummer	+	0. 0. 3. 0
Hostnummer	+	0. 0. 0. 70
IP-Adresse		149. 18. 3. 70

Die Broadcast-Adresse ist immer die höchste Hostnummer im Netzwerk. Da das vierte Byte vollständig für den Hostanteil zur Verfügung steht, ist die höchste Nummer 255 und daraus folgt dann 149.18.3.255 als Broadcast-Adresse.

Hier noch einmal die Zusammenfassung aller Adressen.

IP-Adresse: 149. 18. 3. 70  
 Subnetzmaske: 255.255.255. 0  
 Netzwerknummer: 149. 18. 0. 0  
 Subnetznummer: 0. 0. 3. 0  
 Subnetzwerknummer: 149. 18. 3. 0  
 Hostnummer: 0. 0. 0. 70  
 Broadcast: 149. 18. 3.255

#### Subnetz in einem C-Klasse-Netz

IP-Adresse: 217.89.70.109  
 Subnetzmaske: 255.255.255.224

Es handelt sich hier um ein C-Klasse-Netz. Bei einem C-Klasse-Netz geben die ersten drei Bytes (24 Bit) die Netzwerknummer an. Die Adresse befindet sich also im Netz 217.89.70.0. Die Ermittlung der Subnetznummer und der Hostnummer ist hier etwas schwieriger als im vorherigen Beispiel. Als erstes muß ermittelt werden, welche Bits fürs Subnetz verwendet werden. Dazu stellen wir das letzte Byte der Subnetzmaske als eine Binärzahl da. Aus 224 wird dann 11100000. Die ersten drei Bits stehen auf 1 und geben damit den Subnetzanteil an. Das vergleichen wir nun mit der Binärdarstellung der IP-Adresse.

224 = | 111 | 00000  
 109 = | 011 | 01101

Daraus können wir sofort die Subnetznummer als 0.0.0.96 (01100000) und die Hostnummer als 0.0.0.13 (00001101) erkennen. Auch hier gilt, daß die einzelnen Teile zusammen die IP-Adresse ergeben müssen.

Netzwerknummer		217. 89. 70. 0
Subnetznummer	+	0. 0. 0. 96
Hostnummer	+	0. 0. 0. 13
IP-Adresse		217. 89. 70.109

Die Broadcastadresse ist wieder die höchste Nummer im Netz. Das letzte Byte der Adresse sieht in binärer Darstellung wie folgt aus: 01111111. Daher ist die Broadcast-Adresse dann 217.89.70.127.



Hier noch einmal die Zusammenfassung aller Adressen.

IP-Adresse:	217. 89. 70.109
Subnetzmaske:	255.255.255.224
Netzwerknummer:	217. 89. 70. 0
Subnetznummer:	0. 0. 0. 96
Subnetzwerknummer:	217. 89. 70. 96
Hostnummer:	0. 0. 0. 13
Broadcast:	217. 89. 70.127

#### 17.4.4 Die möglichen Subnetze

Die Netzwerke können auch wie im vorherigen Kapitel durch die Schreibweise nach RFC 1700 dargestellt werden. Ein IP-Adresse im Subnetz wird dann in der Form { Netzwerkanteil , Subnetzanteil , Hostanteil } dargestellt. Auch hier steht die "0" dafür, daß alle Bits auf 0 stehen und die "-1", daß alle Bits auf 1 stehen. Wie bei den Hosts gibt es bei den Subnetzen bestimmte Netze mit einer besonderen Bedeutung.

##### Das lokale Subnetz

Wird der Subnetzanteil in der IP-Adresse auf 0 gesetzt, dann ist ein Rechner im aktuellen Subnetz gemeint.

{ Netzwerkanteil , 0 , Hostanteil }

##### Alle Subnetze

Besteht dagegen der Subnetzanteil nur aus gesetzten Bits, dann sind alle Subnetze gemeint.

{ Netzwerkanteil , -1 , Hostanteil }

##### Subnetzwerknummer

Wie auch bei den Netzwerkklassen bedeutet eine IP-Adresse mit einem auf Null gesetzten Hostanteil die Nummer des Netzwerks. Hier natürlich die des "Subnetzwerks".

{ Netzwerkanteil , Subnetzanteil , 0 }

##### Direkter Broadcast

Der direkte Broadcast ist genau so definiert, wie bei den Netzwerkklassen. Auch bei Subnetzen werden alle Rechner des Subnetzes angesprochen, wenn alle Bits des Hostanteils auf 1 stehen.

{ Netzwerkanteil , Subnetzanteil , -1 }

#### 17.4.5 Die belegten Subnetze

Nach der RFC 1219 "*On the Assignment of Subnet Numbers*" sind zwei Subnetznummern analog zu den Hostnummern verboten. Dies sind die Netze { Netzwerkanteil , 0 , 0 } und { Netzwerkanteil , -1 , 0 }. Daher gehen bei der Aufsplittung eines Netzes in verschiedene Subnetze zwei Netze mit ihren Hosts verloren. Viele kommerzielle Router kümmern sich nicht um diese Vorgabe und bearbeiten die Netze trotzdem. Eine Übersicht über mögliche Subnetze eines B-Klasse- und eines C-Klasse-Netzes zeigen die Tabellen 17.3 und 17.4.

Subnetzmaske	Subnetzanteil in Bit	Anzahl Subnetze		Hostanteil in Bit	Anzahl Hosts
		RFC	Kommerz. Router		
255.255.128.0	1	0	2	15	32766
255.255.192.0	2	2	4	14	16382
255.255.224.0	3	6	8	13	8190
255.255.240.0	4	14	16	12	4094
255.255.248.0	5	30	32	11	2046
255.255.252.0	6	62	64	10	1022
255.255.254.0	7	126	128	9	510
255.255.255.0	8	254	256	8	254
255.255.255.128	9	510	512	7	126
255.255.255.192	10	1022	1024	6	62
255.255.255.224	11	2046	2048	5	30
255.255.255.240	12	4094	4096	4	14
255.255.255.248	13	8190	8192	3	6
255.255.255.252	14	16382	16384	2	2

Tabelle 17.3: Subnetzmasken für B-Klasse-Netze

Subnetzmaske	Subnetzanteil in Bit	Anzahl Subnetze		Hostanteil in Bit	Anzahl Hosts
		RFC	Kommerz. Router		
255.255.255.128	1	0	2	7	126
255.255.255.192	2	2	4	6	62
255.255.255.224	3	6	8	5	30
255.255.255.240	4	14	16	4	14
255.255.255.248	5	30	32	3	6
255.255.255.252	6	62	64	2	2

Tabelle 17.4: Subnetzmasken für C-Klasse-Netze

## 17.4.6 Weitere Beispiele

### Aufteilung eines B-Klasse-Netzes in Subklassen

Sie haben das Netzwerk 130.13.0.0 bekommen. Sie benötigen 22 Subnetze in Ihrem Netz. Wie muß die Subnetzmaske lauten und wie die dazugehörigen Netzwerknummern?

Beim Netzwerk 130.13.0.0 handelt es sich um ein B-Klasse-Netzwerk. Die Standardsubnetzmaske dafür lautet 255.255.0.0. Zur freien Verwendung stehen nur die letzten 16 Bit zur Verfügung.

Sie wollen 22 Subnetze bilden. Da immer zwei Subnetze wegfallen brauchen Sie einen Subnetzanteil, der 24 verschiedene Nummern umfassen kann. Bei vier Bit Netzwerkanteil kommen wir auf 16 Nummern. Das ist zu wenig. Bei fünf Bit ( $2^5$ ) reicht der Nummernbereich für 32 Netze. Es stehen also die Nummern 1 bis 30 zur Verfügung. Die Nummern 0 und 31 dürfen nach RFC nicht belegt werden. Nach Tabelle 17.2 stehen bei der Zahl 248 die ersten fünf Bits auf 1 und die restlichen auf 0. Das ergibt 248 für das dritte Byte und 0 für das letzte Byte der Subnetzmaske. Kombinieren Sie das noch mit der Standardsubnetzmaske für ein B-Klasse-Netz, dann lautet die Subnetzmaske 255.255.248.0.

Für den Hostanteil bleiben  $3 + 8 = 11$  Bit über. Das ergibt  $2^{11} - 2 = 2046$  Rechner in einem Subnetz.

Ein Blick in Tabelle 17.3 bestätigt unsere Berechnungen.

Zur Bestimmung der Netzwerknummern müssen wir uns nur das dritten Byte ansehen. Die ersten beiden Bytes sind durch das B-Klasse-Netz vorgegeben. Aufgrund der Subnetzmaske ist das vierte Byte nur Bestandteil des Hostanteils und damit 0 für das Netzwerk.

Subnetzmaske		11111000	248	255.255.248.0
1.	Netz	00001000	8	130.13.8.0
2.	Netz	00010000	16	130.13.16.0
3.	Netz	00011000	24	130.13.24.0
4.	Netz	00101000	32	130.13.32.0
5.	Netz	00110000	40	130.13.40.0
...				
28.	Netz	11100000	224	130.13.224.0
29.	Netz	11101000	232	130.13.232.0
30.	Netz	11110000	240	130.13.240.0

Wir können das ganze auch anders bestimmen. Der Hostanteil im dritten Byte beträgt drei Bit. Mit drei Bit können wir den Zahlenraum zwischen 0 und 7 und damit 8 Zahlen abdecken. Die Netzwerknummer muß sich im dritten Byte in Schritten von 8 erhöhen, wie Sie leicht aus der obigen Tabelle entnehmen können.

### Aufteilung eines C-Klasse-Netz in Subnetze

Sie haben das C-Klasse-Netz 194.195.196.0 für ein Firmennetz zugewiesen bekommen. Sie wollen für jedes der vier Stockwerke des Firmengebäudes ein Subnetz einrichten. In jedem Subnetz müssen mindestens 20 Rechner Platz finden.

In einem C-Klasse-Netz steht nur noch das vierte Byte für den Hostanteil und damit auch fürs Subnetting zur Verfügung. Nach Tabelle 17.4 finden in einem Subnetz mit der Subnetzmaske 255.255.255.224 30 Hosts Platz und es können 6 Subnetze gebildet werden. Die Subnetzmasken 255.255.255.192 und 255.255.255.240 kommen nicht in Frage, da sie entweder zu wenigen Hosts Platz bieten oder zu wenig Subnetze gebildet werden können. Für den Hostanteil stehen also 5 Bit und für den Netzwerkanteil 3 Bit zur Verfügung.

Für die Bestimmung der Netzwerknummern zählen wir uns einfach durch das Netzwerk. Jedes Subnetz umfaßt 32 Nummer. Die Netzwerknummer, 30 Hostnummern und den Broadcast. Wir brauchen also nur in 32er Schritten die Netzwerknummer hochzuzählen. Die Broadcastadresse ist immer eins weniger als die nächste Netzwerknummer.

Netz	Netzwerknummer	Broadcast	
	194.195.196.0	194.195.196.31	Verboten
1.	194.195.196.32	194.195.196.63	1. Stock
2.	194.195.196.64	194.195.196.96	2. Stock
3.	194.195.196.96	194.195.196.127	3. Stock
4.	194.195.196.128	194.195.196.159	4. Stock
5.	194.195.196.160	194.195.196.191	In Reserve
6.	194.195.196.192	194.195.196.223	In Reserve
	194.195.196.224	194.195.196.255	Verboten

.....  
*Notizen:*

.....

XIX

Arbeitsblatt  
Binär-, Oktal-, Dezimal- und Hexadezimalzahlen

19.1

1 Ergänzen Sie die fehlenden Zahlen.

Binär	Oktal	Dezimal	Hexadezimal
		8	
		16	
		42	
			42
	42		
00110101			
10010111			
	30		
		12	
			E8
		67	
01101000			
			A7
		21	
	337		
	104		
		139	
01111110			
	33		
			99
		19	
01100001			
			91
	56		
11010110			
			64

XIX

# Arbeitsblatt Binär-, Oktal-, Dezimal- und Hexadezimalzahlen

19.2

**2** Das Rechte-System von Linux basiert auf drei Benutzergruppen (Besitzer, Gruppe, Rest der Welt) denen jeweils drei Grundrechte (Lesen(r), Schreiben(w), Ausführen(x)) zugeordnet sind. Dargestellt werden die Rechte an einer Datei durch neun Buchstaben bzw. Striche. Der Buchstabe steht für das gegebene Recht, der Strich für ein nicht gegebenes Recht (Bsp.: `rwxr-xr--`). Dies kann natürlich auch in Binärform dargestellt werden (Bsp.: `111101100`). Daneben bieten sich auch oktale Zahlen für die Darstellung an, da eine oktale Ziffer durch drei Binärziffern dargestellt wird (Bsp.: `754`). Daraus ergeben sich zwei gebräuchliche Schreibweisen:

`rwxr-xr-x` und `755`

Ergänzen Sie die fehlenden Einträge.

Buchstaben	Binär	Oktal
-----	000000000	
<b>rwxrw-r--</b>		
<b>rwxr-x---</b>		
<b>rw-rw-r--</b>		
<b>r-xr-----</b>		
		<b>755</b>
		<b>444</b>
		<b>770</b>
		<b>555</b>
		<b>777</b>

**3** Für die Codierung von Farben wird gerne das Format RGB verwendet. Dabei wird die Farbe in drei Grundfarben (Rot, Grün, Blau) aufgeteilt. Die Intensität eines Farbkanals wird dabei durch ein Byte (8 Bit) dargestellt. Der Wert 0 bedeutet keinen Farbanteil, der Wert 255 vollen Farbanteil. Für die Darstellung der Farben eignet sich sehr gut das hexadezimale Zahlensystem, da vier Bits genau einer hexadezimalen Ziffer entsprechen.

Ergänzen Sie die fehlenden Einträge.

Farbe	Dezimal	Hexadezimal
Schwarz	0,0,0	000000
Weiß	255,255,255	FFFFFF
Rot	255,0,0	FF0000
Grün	0,255,0	00FF00
Blau	0,0,255	0000FF
Oliv	<b>128,128,0</b>	
Marine		<b>000080</b>
Silber	<b>192,192,192</b>	
Antikweiß		<b>FAEBD7</b>
Braun		<b>A52A2A</b>
Schokolade	<b>210,105,30</b>	
Kornblumenblau		<b>6495ED</b>
Tomate	<b>255,99,71</b>	
Gelbgrün	<b>154,205,50</b>	
Sattelbraun		<b>8B4513</b>

XX

# Arbeitsblatt IP-Adressierung und Subnetting

20.1

1 Ordnen Sie die folgenden IP-Adressen den richtigen Netzwerkklassen zu.

Hostnummer	Klasse	Hostnummer	Klasse
217.89.70.60		80.134.172.19	
217.227.201.23		170.126.156.149	
73.110.164.102		103.181.30.156	
132.149.220.53		97.135.28.253	
200.178.19.91		29.58.19.11	
174.250.118.190		128.6.222.226	
246.13.169.122		182.100.73.223	
119.12.110.238		163.183.206.65	
205.140.90.5		203.85.95.108	
134.56.69.66		42.70.234.99	
240.139.245.200		139.55.217.3	
211.123.240.208		200.111.190.212	
190.0.2.26		230.253.217.92	
192.216.99.225		239.250.8.127	
56.228.209.208		142.27.134.170	

2 Ergänzen Sie für die Hostnummer die passende Netzwerknummer.

Hostnummer	Netzwerk	Hostnummer	Netzwerk
70.91.209.69		152.103.177.154	
196.131.166.231		223.201.214.89	
179.231.182.85		10.200.254.99	
124.5.186.105		158.152.188.178	
107.142.179.166		217.72.14.148	
37.14.238.53		151.61.11.47	
13.200.114.143		177.108.213.6	
189.26.153.244		178.235.207.35	
65.83.43.15		114.50.130.31	
168.87.136.136		125.100.249.24	
91.230.209.250		214.142.197.148	
34.93.147.247		223.86.9.141	
104.184.101.61		17.206.9.31	
175.79.219.123		53.61.13.36	
107.55.146.207		87.100.76.201	

XX

Arbeitsblatt  
IP-Adressierung und Subnetting

20.2

**3** Ermitteln Sie aus der gegebenen IP-Adresse und Subnetmaske die Subnet-Nummer und Host-Nummer.

IP-Adresse	Subnetmaske	Subnet-Nr.	Host-Nr.
130.51.92.159	255.255.224.0		
9.197.82.175	255.255.192.0		
224.155.133.13	255.255.255.252		
132.51.59.146	255.255.255.0		
213.241.216.6	255.255.255.252		
149.220.7.147	255.255.254.0		
154.129.217.12	255.255.255.224		
158.249.111.14	255.255.255.128		
85.188.84.168	255.255.0.0		
220.161.149.95	255.255.255.192		
209.204.252.238	255.255.255.224		
213.64.202.123	255.255.255.252		
182.26.131.70	255.255.255.0		
108.30.161.16	255.255.255.128		
154.123.28.67	255.255.254.0		
113.139.81.204	255.255.255.224		
170.165.117.44	255.255.255.0		
168.12.140.68	255.255.255.224		
220.89.51.35	255.255.255.240		
148.252.157.238	255.255.255.192		

**4** Mit welcher Begründung

1. darf der Hostanteil nicht nur aus Nullen
2. und nicht nur aus Einsen bestehen?

**5** Welche Bedeutung haben folgende Adressen und dürfen deshalb bei der IP-Adressierung nicht verwendet werden?

1. 0.0.0.0
2. 127.0.0.1
3. 255.255.255.255

**6** Warum hat man 1985 so dringend das Subnetting gemäß RFC 950 im Internet eingeführt?

**7** Ihnen ist die Netzwerkadresse 135.42.0.0 zugewiesen worden. Wieviele Binärstellen werden benötigt, um 14 Subnetze zu erhalten. Wie lautet die Subnetmaske und das erweiterte Netzwerk-Präfix?



XX

# Arbeitsblatt IP-Adressierung und Subnetting

20.3

**8** Nehmen Sie das Beispiel aus Aufgabe 7 und drücken Sie die 14 Subnetze in Binär- und Dezimalform aus. Sortieren Sie die Zahlen von der kleineren Nummern zur größeren.

	Binärform	Dezimalform
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

- 9** Geben Sie den erlaubten Adreßbereich der Hosts des dritten Subnetzes aus Aufgabe 8 an.
- 10** Wie lautet die Broadcast-Adresse für Subnetz Nr. 5 aus Aufgabe 8?
- 11** Gehen Sie von der Netzwerkadresse 217.89.70.0 aus.
1. Wie lautet das erweiterte Netzwerk-Präfix um mindestens 20 Hosts im Subnetz adressieren zu können?
  2. Was ist die maximale Anzahl von Hosts pro Subnetz?
  3. Was ist die maximale Anzahl von Subnetzen, die definiert werden können?
- 12** Im Internet werden bestimmte Adressbereiche nicht veröffentlicht, sind aber gut für Intranets geeignet. Geben Sie die Bereiche für B-Klasse-Netze und C-Klasse-Netze an.
- 13** Sie sollen für eine Firma ein Netz einrichten. Sie haben von Ihrem Provider das C-Klasse-Netz 217.89.50.0 zugewiesen bekommen. Jede Abteilung soll ein eigenes Subnetz besitzen, die durch Router miteinander verbunden sind.

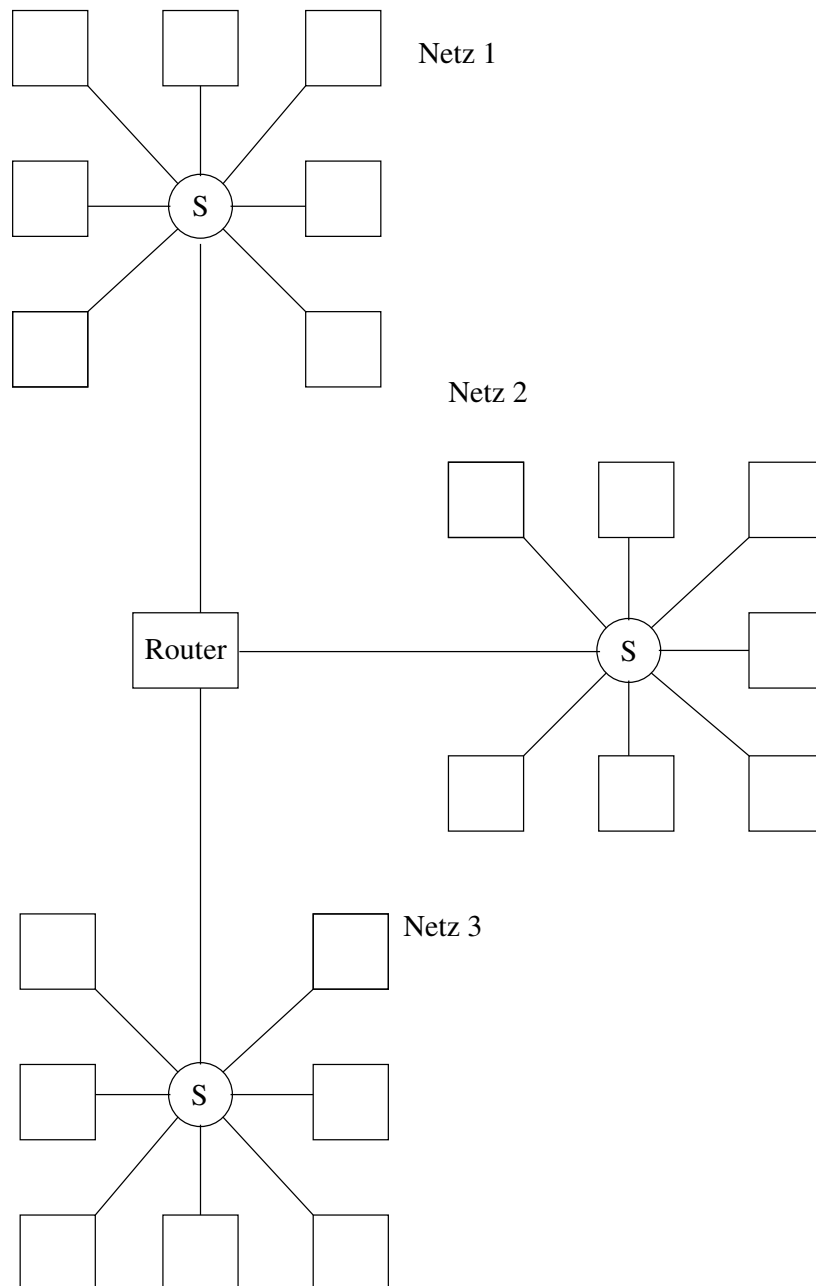
Abteilung	Hosts	Subnetmaske	Subnetzwerk
Buchhaltung	9		
Versand	13		
Call Center I	48		
Call Center II	31		
Lager	10		
Werkstatt	6		
Firewall	5		
Entwicklung	29		

XX

Arbeitsblatt  
IP-Adressierung und Subnetting

20.4

**14** Eine Universität erhält die B-Klasse-Netzwerknummer 180.76.0.0 vom Provider. Sie planen als Administrator ein "Campus-Netzwerk" mit weniger als 10 internen Netzwerken. Vergeben Sie alle IP-Adressen im abgebildeten Netzwerk. Geben Sie auch die Subnetzadressierung an. Tragen Sie bitte die Adressen mit Bleistift in die Skizze ein.



# Kapitel 18

## Linux im Netzwerk

Dieses Kapitel beschäftigt sich mit der TCP/IP-Konfiguration von Linux. Wie nicht anders zu erwarten übernimmt der Kernel die Aufgabe die Netzwerkschnittstellen zu steuern. In den meisten Fällen liegen die dafür benötigten Kernelteile bzw. Treiber als Kernelmodule vor, die bei Bedarf geladen werden. Im Gegensatz zu der üblichen Linuxstruktur gibt es für Netzwerkschnittstellen keine Gerädateien. Die Ansprache der Schnittstellen erfolgt über sogenannte *Interfaces* bzw. Schnittstellennamen, die vom Kernel bereitgestellt werden. So lauten z. B. für Ethernet-Karten die Schnittstellennamen `eth0`, `eth1` u.s.w.

### 18.1 Konfiguration

Zur TCP/IP-Konfiguration gehört das Vergeben einer IP-Adresse an eine Netzwerkschnittstelle, wie auch das Einrichten von Routingtabelle. Jede Netzwerkschnittstelle benötigt eine IP-Adresse, eine Subnetmaske und eine Broadcast-Adresse.

#### 18.1.1 Konfiguration der Netzwerkschnittstellen: `ifconfig`

Das Kommando `ifconfig` wird benutzt um Netzwerkschnittstellen zu konfigurieren. Es wird meistens beim Booten des Systems verwendet, um die Schnittstellen zu initialisieren. Im laufenden Betrieb wird es üblicherweise nur zur Fehlersuche oder zur Änderung der aktuellen Systemkonfiguration verwendet.

```
ifconfig [SCHNITTSTELLE]
ifconfig SCHNITTSTELLE [ADR-FAMILIE] [OPTIONEN] [ADRESSEN]
```

Wenn keine Argumente angegeben werden, dann zeigt `ifconfig` den Zustand der augenblicklich aktiven Netzwerkschnittstellen. Wird ein einzelnes Schnittstellenargument angegeben, so zeigt es nur den Zustand der angegebenen Netzwerkschnittstelle an.

Ist das erste Argument hinter der Schnittstelle der Name einer unterstützten Adressfamilie, so wird diese Adressfamilie dazu benutzt um alle Protokolladressen zu dekodieren und darzustellen. Zur Zeit werden u. A. folgende Adressfamilien unterstützt: `inet` (TCP/IP, standard), `inet6` (IPv6), `ax25` (AMPR Packet Radio), `ddp` (Appletalk Phase 2), `ipx` (Novell IPX) und `netrom` (AMPR Packet radio).

Denken Sie daran, daß der Befehl natürlich nur *root* zur Verfügung steht und seine Wirkung auf das laufende System beschränkt ist. Nach einem erneuten Booten stehen die Einstellungen nicht mehr zur Verfügung. `ifconfig` muß daher in den Startskripten aufgerufen werden. Inzwischen steht `ip` als weiteres Tool zur Konfiguration zur Verfügung. Es wird bei SuSE zusammen mit dem Skript `ifup`, daß die Informationen im Verzeichnis `/etc/sysconfig/network` auswertet, benutzt um die Netzwerkschnittstellen beim Start zu konfigurieren.

**Optionen**

<b>up</b>	Schnittstelle wird aktiviert, wird automatisch bei Adresszuweisung gesetzt.
<b>down</b>	Schnittstelle wird deaktiviert
<b>[-]promisc</b>	Promiscuous-Modus, alle Pakete vom Netzwerk werden empfangen (- davor deaktiviert den Modus)
<b>[-]allmulti</b>	All-Multicast-Modus, alle Multicast-Pakete werden empfangen (- davor deaktiviert den Modus)
<b>metric N</b>	Dieses Argument setzt den Metrik-Wert für die Schnittstelle auf N
<b>mtu N</b>	Dieses Argument setzt die Maximum Transfer Unit (MTU), das größte Paket, daß gesendet werden kann
<b>irq IRQ</b>	Legt einen neuen Interrupt für die Karte fest, nicht alle Karten unterstützen diese Funktion
<b>io_addr IOBA</b>	Setzt die I/O-Basisadresse für dieses Gerät
<b>netmask ADR</b>	Setzt die Netzwerkmaske
<b>[-]broadcast ADR</b>	Setzt die Broadcastadresse und setzt/löscht das IFF_BROADCAST Flag

Wird ein einzelne Option **-a** angegeben, zeigt **ifconfig** den Zustand aller Schnittstellen an, selbst wenn diese inaktiviert sind.

```
defiant:~ # ifconfig -a
eth0      Protokoll:Ethernet  Hardware Adresse 00:E0:18:1B:21:55
          inet Adresse:10.0.1.3  Bcast:10.0.1.255  Maske:255.255.255.0
          inet6 Adresse: fe80::2e0:18ff:fe1b:2155/10  Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:0 (0.0 b)  TX bytes:288 (288.0 b)
          Interrupt:9 Basisadresse:0x7c00

lo        Protokoll:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128  Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:1956 (1.9 Kb)  TX bytes:1956 (1.9 Kb)

sit0      Protokoll:IPv6-nach-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Mit den Schaltern **up** und **down** wird die Schnittstelle aktiviert oder deaktiviert.

```
defiant:~ # ifconfig eth0 down
defiant:~ # ifconfig eth0 up
```

Das Setzen der IP-Adressen inklusive Netzwerkmaske und Broadcastadresse geht so:<sup>1</sup>

```
defiant:~ # ifconfig eth0 217.89.70.60 netmask 255.255.255.224 \
> broadcast 217.89.70.63
defiant:~ # ifconfig eth0
eth0      Protokoll:Ethernet  Hardware Adresse 00:E0:18:1B:21:55
          inet Adresse:217.89.70.60  Bcast:217.89.70.63  Maske:255.255.255.224
          inet6 Adresse: fe80::2e0:18ff:fe1b:2155/10  Gültigkeitsbereich:Verbindung
```

---

<sup>1</sup>Der Backslash in der Anweisung sagt nur, daß der Befehl in der nächsten Zeile weitergeschrieben werden kann.

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
Kollisionen:0 Sendewarteschlangenlänge:100
RX bytes:0 (0.0 b) TX bytes:576 (576.0 b)
Interrupt:9 Basisadresse:0x7c00
```

Sie können auch auf eine Schnittstelle weitere IP-Adressen legen. In diesem Fall hängen Sie an den Schnittstellennamen einfach einen Doppelpunkt und eine Zahl an.

```
linux37:~ # ifconfig eth0:1 10.0.0.1
linux37:~ # ifconfig eth0
eth0      Protokoll:Ethernet Hardware Adresse 00:D0:B7:B2:51:43
          inet Adresse:217.89.70.37 Bcast:217.89.70.63 Maske:255.255.255.224
          inet6 Adresse: fe80::2d0:b7ff:feb2:5143/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:20102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18259 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:13770745 (13.1 Mb) TX bytes:2171618 (2.0 Mb)
          Interrupt:10 Basisadresse:0xa400 Speicher:df800000-df800038
linux37:~ # ifconfig eth0:1
eth0:1    Protokoll:Ethernet Hardware Adresse 00:D0:B7:B2:51:43
          inet Adresse:10.0.1.1 Bcast:10.255.255.255 Maske:255.0.0.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          Interrupt:10 Basisadresse:0xa400 Speicher:df800000-df800038
```

Die Angaben zu einer virtuellen Schnittstelle sind knapper als die zu einer physikalischen Schnittstelle. Die Verkehrsstatistiken, die natürlich für beide Schnittstellen identisch sind, werden ausgelassen.

Ein ganzes A-Klasse-Netz ist uns zu groß. Wir wollen uns mit einem Netz mit 256 Adressen begnügen.

```
linux37:~ # ifconfig eth0:1 10.0.1.1 netmask 255.255.255.0
linux37:~ # ifconfig eth0:1
eth0:1    Protokoll:Ethernet Hardware Adresse 00:D0:B7:B2:51:43
          inet Adresse:10.0.1.1 Bcast:10.255.255.255 Maske:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          Interrupt:10 Basisadresse:0xa400 Speicher:df800000-df800038
```

Die Netzwerkmaske stimmt, aber die Broadcastadresse wurde nicht geändert. Hier ist also Handarbeit angesagt.

```
linux37:~ # ifconfig eth0:1 10.0.1.1 netmask 255.255.255.0 broadcast 10.0.1.255
linux37:~ # ifconfig eth0:1
eth0:1    Protokoll:Ethernet Hardware Adresse 00:D0:B7:B2:51:43
          inet Adresse:10.0.1.1 Bcast:10.0.1.255 Maske:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          Interrupt:10 Basisadresse:0xa400 Speicher:df800000-df800038
```

Eine Abfrage der Routing-Tabelle im Kernel zeigt, daß auch schon die richtige Route für die neue Schnittstelle gesetzt worden ist.

```
linux37:~ # route -n
Kernel IP Routentabelle
```

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
217.89.70.32	0.0.0.0	255.255.255.224	U	0	0	0	eth0
10.0.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
0.0.0.0	217.89.70.62	0.0.0.0	UG	0	0	0	eth0

## Loopback-Interface

Eine ganz besondere Netzwerkschnittstelle ist das *Loopback-Interface*. Es stellt eine Netzwerkverbindung zum eigenen Rechner her. Normalerweise besitzt jedes Linux-System durch dieses Interface ein Netzwerk, selbst wenn keine physikalischen Netzwerkschnittstellen vorhanden sind. Das Loopback-Interface kann zum Testen der Netzwerkkomponenten verwendet werden. Wichtig ist es auch in den Fällen, wenn Client-Server-Anwendungen auf dem gleichen Rechner laufen sollen und keine Netzwerkhardware verbaut wurde.

Auch das Loopback-Interface muß explizit angelegt werden. In den meisten Fällen wird es durch die Startskripte aktiviert. Mit folgendem Befehl können Sie selbst ein Loopback-Interface erstellen.

```
ifconfig lo 127.0.0.1
```

### 18.1.2 Konfiguration der Kernel-Routing-Tabelle: route

Pakete können über das Internet in alle Welt verbreitet werden. Die Ansprache der Rechner im gleichen Netz ist kein Problem. Für alle anderen Rechner müssen Regeln aufgestellt werden, wie das Paket zu diesem Rechner kommt. Im einfachsten Fall wird durch die Regel festgelegt, welcher Rechner das Paket näher ans Ziel bringt. Diese Regeln der Weiterleitung bezeichnet man als *Routen*.

Jeder Rechner besitzt mindestens zwei Routen. Eine ist für das Loopback-Interface bestimmt, die andere Route gilt für die Netzwerkschnittstelle. Dabei können statische und dynamische Routen verwendet werden. Für dynamische Routen werden noch Routing-Daemonen benötigt. Geläufige Routing-Daemonen sind **gated** und **routed**. Auf das dynamische Routing, was nicht gerade frei von Tücken ist, will ich hier nicht weiter eingehen. Für den Anfang reicht das statische Routing.

Der Befehl **route** wird zum Verändern der IP-Routing-Tabelle des Kernels verwendet. In den meisten Fällen wird es dazu verwendet statische Routen zu Rechnern und Netzwerken anzuzeigen oder festzulegen.

```
route [OPTIONEN]
route add [-net|-host] ZIEL [OPTIONEN]
route del [-net|-host] ZIEL [OPTIONEN]
```

Im einfachsten Fall zeigt der Befehl **route** die Routingtabelle an. So sieht die Routing-Tabelle für einen einfachen Netzwerkrechner aus.

```
enterprise:~ # route
Kernel IP Routentabelle
```

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
10.0.1.0	*	255.255.255.0	U	0	0	0	eth0
default	10.0.1.11	0.0.0.0	UG	0	0	0	eth0

Die Routing-Tabelle für einen DSL-Router unterscheidet sich schon etwas von der des Netzwerkrechners.

```
ds9:~ # route
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
217.5.98.62	*	255.255.255.255	UH	0	0	0	ppp0
10.0.1.0	*	255.255.255.0	U	0	0	0	eth0
default	217.5.98.62	0.0.0.0	UG	0	0	0	ppp0

Schauen wir uns doch mal die Bedeutung der 8 Spalten von links nach rechts an.

1. **Destination:** Als Zieladresse können Netz- oder Hostadressen eingetragen sein. Neben den individuellen Routen gibt es noch die *Standardroute* (engl. *default route*). Sie wird immer dann verwendet, wenn keine der anderen Routen auf die Paketadresse zutrifft. In der Spalte Destination wird die Standardroute mit *default* oder mit *0.0.0.0* bezeichnet.

2. **Gateway:** Ist das angegebene Ziel nur über einen Router zu erreichen, dann wird hier seine Adresse eingegeben. Sollte kein Router benötigt werden, dann steht hier der Asterisk (\*).
3. **Genmask:** In dieser Spalte wird die Subnetmaske für die Zieladresse angegeben. Handelt es sich bei dem Ziel um einen Host, dann ist die Subnetmaske 255.255.255.255. Die Standardroute hat dagegen die Subnetmaske 0.0.0.0.
4. **Flags:** Eine Beschreibung der Route erfolgt über die Flag-Spalte. In dieser Spalte können mehrere der folgenden Buchstaben stehen.
  - U** Die Route ist aktiv.
  - G** Es handelt sich dabei um eine Route, die auf einen Router zeigt.
  - H** Die Route ist nicht für ein Netz sondern für einen Host bestimmt.
5. **Metrics:** Nur für dynamisches Routing wichtig.
6. **Ref:** Nur für dynamisches Routing wichtig.
7. **Use:** Hiermit wird angegeben, wie oft die Route schon verwendet wurde.
8. **Iface:** Sobald mehrere Schnittstellen im Rechner vorhanden sind, sollten die Routen auch die verwendeten Schnittstellen aufführen.

Eine Übersicht über die Optionen von `route` bietet die folgende Tabelle.

#### Optionen

<b>-A</b> <i>ADRESSFAMILIE</i>	Benutzt die angegebenen <i>ADRESSFAMILIE</i> , z. B. inet oder inet6
<b>-e</b>	Ausgabeformat von <code>netstat</code> für Routentabellen
<b>-ee</b>	Ausgabe aller Routenparameter aus der Routentabelle
<b>-F</b>	Zeigt die FIB Routentabelle des Kernels an
<b>-C</b>	Zeigt den Routencache des Kernels an
<b>-n</b>	Zeigt numerische Adressen an, es erfolgt keine Namensauflösung
<b>-net</b>	Das Ziel ist ein Netzwerk
<b>-host</b>	Das Ziel ist ein Rechner
<b>-V</b>   <b>--version</b>	Version
<b>-v</b>	Ausführliche Ausgabe
<b>add</b>	Setzt eine neue Route
<b>del</b>	Löscht eine Route
<b>netmask</b> <i>NN</i>	Die Netzwerkmaske der Route
<b>gw</b> <i>ROUTER</i>	Das zum Erreichen des Ziels benutzte Gateway.
<b>metric</b> <i>M</i>	Setzt das Metric-Feld der Routentabelle auf <i>M</i>
<b>mss</b> <i>M</i>	Setzt den MSS-Wert (Maximum Segment Size) für TCP-Verbindungen fest (Angabe in Bytes)
<b>reject</b>	Installiert eine Blockaderoute, die einen Abbruch der Suche nach einer Route bewirkt.
<b>dev</b> <i>Schnittstelle</i>	Verbindet die Route mit der angegebenen Schnittstelle.

Benutzte Routen werden für eine gewisse Zeit im Routencache des Kernels gespeichert. Diesen Cache können Sie sich mit dem Schalter `-C` anzeigen lassen.

```
linux37:~ # route -C
```

```
Kernel IP Routencache
```

Ziel	Ziel	Genmask	Flags	Metric	Ref	Ben	Iface
linux37.amov.de	www-proxy.HH1.s	217.89.70.62		0	0	96	eth0
linux37.amov.de	resolv-H.DTAG.D	217.89.70.62		0	0	1	eth0
linux37.amov.de	217.6.176.11	217.89.70.62		0	0	1	eth0
217.89.70.34	217.89.70.63	217.89.70.63	ib1	0	0	4	lo
linux37.amov.de	search.ebay.de	217.89.70.62		0	0	0	eth0
217.6.176.11	linux37.amov.de	linux37.amov.de	1	0	0	15	lo
217.6.176.24	linux37.amov.de	linux37.amov.de	1	0	0	7	lo
linux37.amov.de	banners.ebay.co	217.89.70.62		0	0	0	eth0
linux37.amov.de	217.89.70.255	217.89.70.62		0	0	23	eth0

```

www-proxy.HH1.s linux37.amov.de linux37.amov.de 1      0      0      96 lo
resolv-H.DTAG.D linux37.amov.de linux37.amov.de 1      0      0      1 lo
...

```

Eine Route zu einem Netzwerk können Sie mit dem folgenden Befehl setzen. Die Angabe der Option `-net` ist dabei optional, da davon ausgegangen wird, daß ein Route sich auf ein Netz bezieht.

```
route add -net 10.0.0.0
```

Handelt es sich bei dem Netz um ein Subnetz, dann muß auch die Subnetzmaske mit angegeben werden.

```
route add -net 10.0.1.0 netmask 255.255.255.0
```

Ist das Netzwerk nur über einen Router zu erreichen, muß dieser auch mit angegeben werden.

```
route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.254
```

Besitzt der Rechner mehrere Schnittstellen, so muß bei den Routen auch mit angegeben werden, welche Schnittstelle benutzt werden soll.

```
route add -net 10.0.1.0 netmask 255.255.255.0 dev eth0
route add -net 10.0.2.0 netmask 255.255.255.0 dev eth1
```

Es ist auch möglich eine Route nur zu einem Rechner zu legen.

```
route add -host 217.89.70.193 dev eth2
```

Das Löschen der Routen ist meistens einfacher.

```
route del -host 217.89.70.193
```

Bei Überschneidungen kann es vorkommen, daß eventuell auch noch Subnetzmaske oder Interface angegeben werden muß.

Sie sollten immer daran denken, daß Netzwerkeinstellungen über die Befehle `ifconfig` und `route` nur temporär sind. Um die Einstellungen dauerhaft zu machen, müssen die Befehle in den Startskripten aufgeführt werden.

### 18.1.3 Konfiguration unter SuSE

Unter SuSE kann die Netzwerkkonfiguration unter YaST ausgeführt werden. Diese Änderungen der Netzwerkkonfiguration werden permanent abgespeichert. Dabei wird im Verzeichnis `/etc/sysconfig/network` für jede Schnittstelle eine Datei abgelegt. Diese Datei beginnt mit der Zeichenfolge `ifcfg-` gefolgt von dem Schnittstellennamen.

```

linux37:/etc/sysconfig/network # dir
insgesamt 68
drwxr-xr-x  6 root    root      4096 2001-05-04 14:22 .
drwxr-xr-x  5 root    root      4096 2004-06-04 09:07 ..
-rw-r--r--  1 root    root      5172 2003-09-23 18:51 config
-rw-r--r--  1 root    root      6343 2004-05-04 10:25 dhcp
-rw-r--r--  1 root    root       184 2004-07-06 16:17 ifcfg-eth0
-rw-r--r--  1 root    root       126 2003-09-23 18:51 ifcfg-lo
-rw-r--r--  1 root    root      6192 2003-09-23 18:51 ifcfg.template
drwxr-xr-x  2 root    root      4096 2003-09-23 20:02 if-down.d
drwxr-xr-x  2 root    root      4096 2003-09-23 20:02 if-up.d
drwx----- 2 root    root      4096 2003-09-23 20:02 providers
-rw-r--r--  1 root    root       26 2001-05-04 14:22 routes
drwxr-xr-x  2 root    root      4096 2004-05-04 10:25 scripts
-rw-r--r--  1 root    root     5471 2004-05-04 10:25 wireless

```



In diesen Konfigurationsdateien befinden sich die notwendigen Parameter zur Schnittstellenkonfiguration. Die Datei ist ziemlich selbsterklärend.

```
linux37:/etc/sysconfig/network # cat ifcfg-eth0
BOOTPROTO='static'
BROADCAST='217.89.70.63'
IPADDR='217.89.70.37'
MTU=''
NETMASK='255.255.255.224'
NETWORK='217.89.70.32'
REMOTE_IPADDR=''
STARTMODE='onboot'
UNIQUE='bSAa.HVgIlg0rmpC'
linux37:/etc/sysconfig/network # cat ifcfg-lo
# Loopback (lo) configuration
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
STARTMODE=onboot
```

Auch die Routen werden in einer Datei gespeichert. Die Datei trägt den Namen `routes`.

```
linux37:/etc/sysconfig/network # cat routes
default 217.89.70.62 - -
```

Verantwortlich für die Konfiguration des Netzwerks ist dann das RC-Skript `network`. Dieses bedient sich der Skripte aus dem Verzeichnis `scripts` des Netzwerkkonfigurationsverzeichnis `/etc/sysconfig/network` sowie des Skriptes `ifup`, das eine SuSE-Entwicklung ist. Durch RC-Skript können Sie das Netzwerk auch jederzeit starten und stoppen.

```
linux37:~ # rcnetwork stop
Shutting down network interfaces:
    eth0                                     done
linux37:~ # rcnetwork start
Setting up network interfaces:
    lo                                     done
    eth0      IP/Netmask: 217.89.70.37 / 255.255.255.224      done
```

Um ein neue Schnittstellenkonfiguration hinzuzufügen müssen Sie nur eine passende Datei erstellen. Dazu können Sie eine der anderen Dateien als Vorlage verwenden. Die genaue Erklärung der enthaltenen Direktiven können Sie der Datei `ifcfg.template` entnehmen. Weitere Erläuterungen finden Sie in der Manualpage `ifup(8)`.

```
linux37:/etc/sysconfig/network # cp ifcfg-eth0 ifcfg-eth0:1
linux37:/etc/sysconfig/network # vi ifcfg-eth0:1
linux37:/etc/sysconfig/network # rcnetwork restart
Shutting down network interfaces:
    eth0                                     done
Setting up network interfaces:
    lo                                     done
    eth0      IP/Netmask: 217.89.70.37 / 255.255.255.224      done
    eth0:1     IP/Netmask: 10.0.1.1 / 255.255.255.0            done
```

### 18.1.4 Einrichten eines Routers

Für Rechner mit nur einer Netzwerkschnittstelle nach außen sind Routingtabellen relativ langweilig. Interessant wird es erst bei zwei oder mehr Netzwerkschnittstellen nach außen und wenn das Gerät beide Netze

miteinander verbinden soll. Ein Rechner unter dem Betriebssystem kann als Router fungieren, wenn die entsprechende Kernel-Funktion einkompiliert ist. Bei den heutigen Kernels ist dies normalerweise der Fall, die Funktion ist aber aus Sicherheitsgründen normalerweise abgeschaltet. Als ersten Schritt der Konfiguration des Routers müssen Sie daher das IP-Forwarding einschalten. Im Prinzip bedeutet dies, daß der Linux-Rechner ankommende Pakete analysiert und Sie bei Bedarf an andere Rechner weiterleitet. Seit SuSE 8.0 müssen Sie für die Aktivierung des IP-Forwarding die Direktive `IP_FORWARDING` in der Datei `/etc/sysconfig/sysctl` auf `yes` setzen.

```
linux37:~ # cat /etc/sysconfig/sysctl
...
## Path:          Network/General
## Description:   forward/route IP(v4) packets
## Type:          yesno
## Default:       no
#
# Runtime-configurable parameter: forward IP packets.
# Is this host a router? (yes/no)
#
IP_FORWARD="yes"
...
```

Ob das IP-Forwarding auch aktiv ist, können Sie der Datei `/proc/net/snmp` entnehmen. Dies ist dann der Fall, wenn die erste Zahl in der zweiten Reihe eine '1' ist.

```
linux37:~ # cat /proc/net/snmp
Ip: Forwarding DefaultTTL InReceives InHdrErrors InAddrErrors ForwDatagrams ...
Ip: 1 64 33999 0 0 0 0 25813 22942 0 0 0 12257 4088 0 0 0 0
Icmp: InMsgs InErrors InDestUnreachs InTimeExcds InParmProbs InSrcQuenchs In...
Icmp: 1 0 1 0 0 0 0 0 0 0 0 0 107 0 107 0 0 0 0 0 0 0 0
Tcp: RtoAlgorithm RtoMin RtoMax MaxConn ActiveOpens PassiveOpens AttemptFail...
Tcp: 1 200 120000 -1 442 198 0 56 0 20216 17215 47 0 10
Udp: InDatagrams NoPorts InErrors OutDatagrams
Udp: 5177 107 0 5637
```

Nun müssen Sie noch die zwei oder mehr Netzwerkschnittstellen konfigurieren. Als Beispiel nehmen wir einen Router an, der über die Netzwerkschnittstelle `eth0` das Netz `10.0.1.0/24` und über `eth1` das Netz `10.0.2.0/24` miteinander verbinden soll.

```
ds9:~ # ifconfig eth0 10.0.1.254 netmask 255.255.255.0 broadcast 10.0.1.255
ds9:~ # ifconfig eth1 10.0.2.254 netmask 255.255.255.0 broadcast 10.0.2.255
```

Jetzt müssen noch die Routen für die Schnittstellen gesetzt werden.

```
ds9:~ # route add 10.0.1.0 netmask 255.255.255.0 dev eth0
ds9:~ # route add 10.0.2.0 netmask 255.255.255.0 dev eth1
```

Wenn wir jetzt noch davon ausgehen, daß eine Verbindung zu weiteren Netzen (z. B. dem Internet) über das erste Netz erfolgt, müssen wir noch die Standardroute setzen. Als Gateway nehmen wir mal den Rechner mit der IP-Nummer `10.0.1.253`.

```
ds9:~ # route add default gw 10.0.1.253
```

Und fertig ist der Router. War doch gar nicht so schwer.

### 18.1.5 Netzwerkeinstellungen mit DHCP

Neben der oben gezeigten statischen Konfiguration des Netzwerks kann die Netzwerkkonfiguration auch dynamisch erfolgen. Heute bedient man sich meistens dem *Dynamic Host Configuration Protocol*, kurz *DHCP*, daß ein direkter Nachfahre des *BOOTP* (engl. *Bootstrap Protocol*) ist. Genau definiert wird DHCP im RFC 2131.

DHCP ermöglicht eine dynamischer Vergabe von Konfigurationsparametern auf Zeit. Dazu gehören die IP-Adressen und Netzwerkmasken der Schnittstellen, wie auch das Standardgateway und die verwendeten DNS-Server. Nicht mehr benötigte Ressourcen können nach einer gewissen Wartezeit wieder freigegeben werden.

Um DHCP auf einem Linux-Client benutzen zu können, müssen die entsprechenden Dämonen installiert sein. Die kann z. B. `dhcpcd`, `dhclient` oder `pump` (Red Hat) sein. Für diese Dämonen existieren im Normalfall keine RC-Skripte, da diese über die Netzwerkskripte gestartet werden.

Normalerweise verwendet SuSE den `dhcpcd`. Allerdings kann auch der `dhclient` verwendet werden. Mehr Informationen über den `dhcpcd` können Sie der Manualpage **`dhcpcd(8)`** entnehmen.

Als Server dient das Programm `dhcpd`, dessen Konfigurationsdatei `dhcpd.conf` im Verzeichnis `/etc` finden. Die Konfiguration eines DHCP-Servers ist Stoff der zweiten LPI-Levels.

## 18.2 Clientseitige Namensauflösung

IP-Adressen sind Zahlen und Zahlen sind etwas für Computer. Mit diesem Satz können wir das Problem des Anwenders mit IP-Adressen (oder auch Telefonnummern) zusammenfassen. Der normale Mensch kann sich Worte einfach viel besser merken als Zahlen, da er diese mit Gegenständen oder Ereignissen in Verbindung bringen kann. Zahlen sind dagegen etwas abstraktes. Oder können Sie sich unter der Adresse 217.160.110.81 etwas sinnvolles vorstellen? Dagegen sorgt der Begriff `www.fibel.org` schon für einen Wiedererkennungseffekt.

Viele Netzwerkkaplikationen erwarten, daß der Hostrechner einen sinnvollen Hostnamen besitzt. In vielen Fällen wird der einfach kurze Hostname benutzt, der nur aus einem Wort besteht. In meinem Netz gibt es daher die Rechner *enterprise*, *voyager*, *defiant*, *ds9* und *test*. Für ein normales Netz mag das reichen, aber auf die ganze Welt übertragen gäbe es doch schnell eine Knappheit an Namen. Deshalb ist das Namenssystem hierarchisch aufgeteilt worden. So lautet der Netzname meines Netzes *ole.tux*. Da es keine First-Level-Domain *tux* gibt, brauche ich mir auch keine Sorgen um Namenskollisionen mit den regulären Domainnamen zu machen. Den vollständigen Hostnamen aus kurzem Hostnamen und dem Netzwerknamen bzw. Domainnamen bezeichnet man auch als *Full Qualified Domain Name (FQDN)*. Die FQDNs meines Netzes lauten dann also *enterprise.ole.tux*, *voyager.ole.tux*, *defiant.ole.tux*, *ds9.ole.tux* und *test.ole.tux*

### 18.2.1 Übersicht

Für die Namensauflösung stehen mehrere Verfahren zur Verfügung, die nicht nur alleine sondern auch hintereinander ausgeführt werden können.

**Die Datei `/etc/hosts`** Die wohl älteste und einfachste Methode ist die Aufzählung der Adressen und Namen in der Datei `hosts`.

**NIS** Das auf RPC (Remote Procedure Calls) basierende NIS (Network Information System) stellt anderen Rechner die lokalen Informationen eines anderen Rechners zur Verfügung. Dies kann nicht nur die Namensauflösung sondern auch z. B. die Benutzerverwaltung sein.

**DNS** Das heute am häufigsten verwendete System der Namensauflösung ist DNS. Die Clients befragen einfach einen DNS-Server, der die Namensliste vorhält oder, wenn er den jeweiligen Namen nicht identifizieren kann, einfach andere DNS-Server nach zugehörigen IP-Nummer des Namens fragt.

Die Namensauflösung muß von den Programmentwicklern nicht selbst geschrieben werden. Unter Linux steht in der Bibliothek *libc* ein sogenannter *Resolver* zur Verfügung, der diese Aufgabe übernimmt. Welche

Namensauflösung von dem Resolver in welcher Reihenfolge benutzt werden soll, können Sie selbst in einer Konfigurationsdatei einstellen. Im Moment sind je nach eingesetzter Software zwei Dateien aktuell.

#### Resolverkonfiguration: /etc/host.conf

Das alte System bediente sich der Datei `/etc/host.conf`. Diese Datei wird auch heute noch benötigt, wenn Sie Programme benutzen, die auf den alten Versionen 4 und 5 der libc-Bibliothek basieren.

```
defiant:~ # cat /etc/host.conf
#
# /etc/host.conf - resolver configuration file
#
# The following option is only used by binaries linked against
# libc4 or libc5. This line should be in sync with the "hosts"
# option in /etc/nsswitch.conf.
#
order hosts, bind
#
# The following options are used by the resolver library:
#
multi on
```

Die Direktive `order` gibt die Reihenfolge der verwendeten Namensauflösungsverfahren an. Diese möglichen Werte könne sein `bind` für DNS, `nis` für NIS und `hosts` für die Datei `/etc/hosts`.

Wenn es mehrere Einträge für einen Host in `/etc/hosts` gibt, dann können Sie mit der Direktive `multi` entscheiden, ob alle gültigen Adressen (`on`) oder nur die erste gültige Adresse (`off`) zurückgeliefert werden sollen.

Weitere Informationen zu `host.conf` können Sie der Manual-Page **host.conf(5)** entnehmen.

#### Resolverkonfiguration: /etc/nsswitch.conf

Wenn Sie die GNU C Library 2.x (`libc.so.6`) für die Programme verwenden, dann erfolgt die Resolverkonfiguration über die Datei `nsswitch.conf`. In dieser Datei werden nicht nur die Namensauflösungsverfahren festgelegt, sondern auch für andere Konfigurationen die Quellen festgelegt. Ursprünglich lagen alle Konfigurationen in Dateiform vor. Durch den Einsatz von NIS und DNS konnten die Konfigurationen aber auch aus dem Netz bezogen werden. So können Sie z. B. auch die Benutzer und Gruppen übers Netz beziehen. Das Schema wird als *Name Service Switch (NSS)* bezeichnet.

```
defiant:~ # cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Legal entries are:
#
#      compat          Use compatibility setup
#      nisplus         Use NIS+ (NIS version 3)
#      nis             Use NIS (NIS version 2), also called YP
#      dns             Use DNS (Domain Name Service)
#      files           Use the local files
#      db              Use the /var/db databases
#      [NOTFOUND=return] Stop searching if not found so far

passwd: compat
group:  compat

hosts:      files dns
networks:   files dns
```

```

services:      files
protocols:     files
rpc:           files
ethers:        files
netmasks:     files
netgroup:      files
publickey:     files

bootparams:    files
automount:     files nis
aliases:       files

```

Wichtig für die Namensauflösung sind die Direktiven **hosts** und **networks**. Mit der Angabe **files** werden die Dateien **/etc/hosts** und **/etc/networks** benutzt. Die Angabe **dns** bezeichnet natürlich eine Namensauflösung über DNS und die Angabe **nis** sorgt dafür, daß das Network Information System benutzt wird. Weitere Informationen entnehmen Sie bitte der Manual-Page **nsswitch.conf(5)**.

### 18.2.2 Namensauflösung mit /etc/hosts und /etc/networks

Die ursprüngliche Form der Namensauflösung ist die Datei **/etc/hosts**. Das Prinzip ist relativ simpel. Es wird eine Tabelle benutzt, in der die IP-Nummer, der Full Qualified Domain Name und diverse Alias-Namen stehen. Eine Zeile steht für eine Kombination aus IP-Nummer und FQDN. Dabei kann die IP-Nummer mehrfach vorkommen. Am Anfang jeder dieser Zeilen steht die IP-Adresse. Danach folgt der FQDN. Im Anschluß an den FQDN können mehrere Alias-Namen für den Rechner vergeben werden. Es ist üblich hier auch den kurzen Hostnamen des Rechners einzutragen. Zeilen, die mit einem Schweinegatter (#) beginnen, sind wie üblich Kommentarzeilen. Schauen Sie sich doch das folgende Beispiel mal an.

```

linux37:~ # cat /etc/hosts
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1      localhost

217.89.70.36   linux36.amov.de      linux36 walter
217.89.70.37   linux37.amov.de      linux37 willi
217.89.70.38   linux38.amov.de      linux38 manuela
217.89.70.39   linux39.amov.de      linux39 peter
217.89.70.40   linux40.amov.de      linux40 kai

217.89.70.38   mail.amov.de         mail

```

Für das Netz der Firma AMOV sind hier ein paar Namen eingetragen worden. Schauen wir doch mal am Beispiel des Befehls **ping** wie die Namensauflösung so funktioniert.

```

linux37:~ # ping linux37.amov.de
PING linux37.amov.de (217.89.70.37) 56(84) bytes of data.
...
linux37:~ # ping linux37
PING linux37.amov.de (217.89.70.37) 56(84) bytes of data.
...
linux37:~ # ping willi
PING linux37.amov.de (217.89.70.37) 56(84) bytes of data.
...

```

Anhand des FQDN oder eines Alias-Namen wird der entsprechende Eintrag identifiziert, die passende IP-Nummer und der FQDN ermittelt. Dies können Sie schön im folgenden Beispiel sehen.

```
linux37:~ # ping manuela
PING linux38.amov.de (217.89.70.38) 56(84) bytes of data.
...
linux37:~ # ping mail
PING mail.amov.de (217.89.70.38) 56(84) bytes of data.
...
```

Es gibt nicht nur eine Namensauflösung für Hostnamen sondern auch eine Namensauflösung für Netzwerkadressen. Für die Netzwerke und ihre Namen ist die Datei `/etc/networks` zuständig. Jede Zeile bezeichnet eine Verbindung zwischen Netzwerkname und Netzwerkadresse. Am Anfang der Zeile steht der Netzwerkname, dem die Netzwerkadresse folgt. So könnte z. B. eine `/etc/networks` aussehen.

```
linux37:~ # cat /etc/networks
# networks      This file describes a number of netname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.

loopback        127.0.0.0

amov             217.89.70.0
sub1.amov       217.89.70.32
sub2.amov       217.89.70.64
```

### 18.2.3 Namensauflösung mit DNS

Wenn Sie für die Namensauflösung der Resolver-Bibliotheken DNS benutzen, dann müssen Sie der Bibliothek mitteilen, welche Nameserver benutzt werden sollen. Dafür zuständig ist die Datei `resolv.conf` im Verzeichnis `/etc`. Fehlt die Datei oder ist sie leer, dann nimmt die Resolver-Bibliothek an, daß der Nameserver auf dem lokalen Rechner läuft.

```
linux37:~ # cat /etc/resolv.conf
nameserver 217.237.150.225
nameserver 194.25.2.129
nameserver 194.25.0.60
search amov.de itdozent.info fibel.org
```

Die wichtigste Direktive in dieser Datei ist `nameserver`. Hier wird die Adresse des Nameservers eingetragen. Sie können die Direktive mehrfach eingeben. Die Nameserver werden in der Reihenfolge der Einträge befragt. Daher sollten Sie den zuverlässigsten Nameserver an erster Stelle eintragen. Gegenwärtig werden bis zu drei Nameservereinträge in der `resolv.conf` unterstützt. Alle weiteren Einträge werden ignoriert.

Oft ist praktisch Rechner mit einem Kurznamen zu bezeichnen anstatt immer den FQDN einzugeben. Für Rechner, die in der `/etc/hosts` geführt werden, ist das kein Problem, da es dort eine extra Rubrik für solche Namen gibt. Eine Anfrage nach dem Rechner `www` würde aber beim DNS keinen Erfolg haben. Durch die Direktive `domain` können Sie eine Defaultdomain eintragen lassen. Geben Sie nun einen Kurznamen ein, dann erweitert die Resolver-Bibliothek den Namen automatisch um diesen Eintrag. Dies können Sie schön an folgendem Beispiel sehen.

```
linux37:~ # cat /etc/resolv.conf
...
domain amov.de
```

Und so reagiert dann die Namensauflösung bei einem Kurznamen.

```
linux37:~ # ping www
PING www.amov.de (193.254.185.32) 56(84) bytes of data.
...
```

Die Direktive hat den Nachteil, daß Sie nur eine Domain angeben können. Mit der Direktive `search` können Sie mehrere Domainnamen angeben, die solange durchsucht werden, bis ein Kurzname zur Domain paßt.

```
linux37:~ # cat /etc/resolv.conf
...
search amov.de itdozent.info fibel.org
```

Hier geht die Suche weiter, wenn ein Kurzname mal nicht passt.

```
linux37:~ # ping www
PING www.amov.de (193.254.185.32) 56(84) bytes of data.
...
linux37:~ # ping linux
PING linux.fibel.org (193.254.185.30) 56(84) bytes of data.
...
```

Die beiden Direktiven `search` und `domain` schließen sich gegenseitig aus und dürfen nur einmal verwendet werden. Seien sie aber vorsichtig bei der Anwendung der Direktive `search`. Bei Vertippern kann es vorkommen, daß plötzlich die falsche Domain beim Kurznamen übereinstimmt.

### 18.2.4 hostname

Die Anzeigen und Setzen des Hostnamens übernimmt das Programm `hostname`.

```
hostname NEUERHOSTNAME
hostname [OPTIONEN]
```

Ohne Parameter zeigt der Befehl den kurzen Namen des Rechners an. Dies ist der Teil des FQDN, der links vom ersten Punkt steht.

```
defiant:~ # hostname
defiant
```

Wird nur ein Argument an den Befehl übergeben, so wird der kurze Name neu gesetzt.

```
defiant:~ # hostname reliant
defiant:~ # hostname
reliant
defiant:~ # bash
reliant:~ #
```

Lassen Sie sich nicht irritieren. Der neue Namen wird in den Prompt erst aufgenommen, wenn die Bash neu gestartet wird. Der Hostnamen wird nur beim Starten eine Shell ausgelesen.

Natürlich gibt es Optionen, die das Verhalten von `hostname` steuern.

#### Optionen

<code>-a   --alias</code>	Alias-Namen des Rechners ausgeben falls vorhanden
<code>-d   --domain</code>	Zeigt und legt DNS Domain-Namen fest
<code>-F   --file</code>	Den Namen aus der angegebenen Datei auslesen
<code>-f   --fqdn   --long</code>	Zeigt den Full Qualified Domain Name an
<code>-h   --help</code>	Die Hilfe
<code>-i   --ip-address</code>	IP-Nummer des Rechners anzeigen
<code>-s   --short</code>	Den kurzen Rechnernamen anzeigen (Standard)
<code>-v   --verbose</code>	Ausführlichere Informationen
<code>-y   --yp   --nis</code>	Der NIS/YP-Domainname

Schauen wir uns doch mal die Ausgaben des Befehls `hostname` auf der Basis der Datei `/etc/hosts` aus Abschnitt 18.2.1 an. Der kurze Hostname ist im System gespeichert. Der FQDN ergibt sich aus der Befragung der Datei `/etc/hosts`, wie auch die aufgeführten Alias-Namen.

```
linux37:~ # hostname
linux37
linux37:~ # hostname -f
linux37.amov.de
linux37:~ # hostname -a
linux37 willi
```

Schön zeigt folgendes Beispiel, wie der FQDN ermittelt wird. Solange der gesetzte Hostname eine Entsprechung in der `/etc/hosts` hat, solange kann auch der FQDN ermittelt werden. Dabei reicht auch einer der gesetzten Alias-Namen schon für die Namensauflösung aus. Sollte der Hostname nicht in der Datei vorhanden sein, kommt es zu einer Fehlermeldung.

```
linux37:~ # hostname willi
linux37:~ # hostname -f
linux37.amov.de
linux37:~ # hostname falscher
linux37:~ # hostname -f
hostname: Unknown host
```

Es gibt zwei symbolische Links auf das Programm `hostname`. Unter diesen beiden Namen aufgerufen verhält sich das Programm anders als unter seinem echten Namen.

```
defiant:~ # ls -lG /bin/{host,domain,dnsdomain}name
lrwxrwxrwx    1 root          8 2001-05-04 13:11 /bin/dnsdomainname -> hostname
lrwxrwxrwx    1 root          8 2001-05-04 13:11 /bin/domainname -> hostname
-rwxr-xr-x    1 root       9968 2003-09-23 19:20 /bin/hostname
```

## dnsdomainname

Den Domainnamen nach DNS können Sie sich mit dem Befehl `dnsdomainname` anzeigen lassen. Dabei ist `dnsdomainname` nur ein symbolischer Link auf `hostname` und entspricht in seiner Ausführung der Option `-d` von `hostname`.

```
defiant:~ # dnsdomainname
local
defiant:~ # hostname -d
local
```

Ein Setzen des Domainnamens ist nicht möglich. Dies kann nur in dem jeweiligen Namensauflösungsverfahren (Bind, NIS oder `/etc/hosts`) erledigt werden.

```
defiant:~ # dnsdomainname -d neue.domain.tux
dnsdomainname: Mit diesem Program kann der DNS Domainname nicht geändert werden
```

Wenn Bind oder NIS nicht zur Hostnamensauflösung benutzt werden, kann der DNS Domainname (welcher Teil des FQDN ist) in der Datei `/etc/hosts` geändert werden.

```
defiant:~ # hostname -d neue.domain.tux
hostname: Mit diesem Program kann der DNS Domainname nicht geändert werden
```

Wenn Bind oder NIS nicht zur Hostnamensauflösung benutzt werden, kann der DNS Domainname (welcher Teil des FQDN ist) in der Datei `/etc/hosts` geändert werden.

## domainname

Den Domainnamen nach NIS können Sie sich mit dem Befehl `domainname` anzeigen lassen. Dies führt gerne zu einigen Verwechslungen mit `dnsdomainname`. Auch `domainname` ist nur ein symbolischer Link auf `hostname`



und sorgt eben dafür, daß statt des Hostnamens der NIS-Domainname ausgegeben wird. Das ist identisch mit der Option `-y` von `hostname`.

```
defiant:~ # domainname
mydomain
defiant:~ # hostname -y
mydomain
```

Das Setzen des NIS-Domainnamens ist analog zum Hostnamen möglich.

```
defiant:~ # domainname ypforme
defiant:~ # domainname
ypforme
```

## 18.3 Diagnosewerkzeuge

Nicht immer läuft in einem Netzwerk alles glatt. In diesem Fall können Sie auf eine Reihe von Diagnosewerkzeuge zurückgreifen, die Ihnen helfen das Problem einzukreisen.

### 18.3.1 Bist du da? ping

Das wohl einfachste Diagnosewerkzeug ist das Programm `ping`. Es sendet mehrere kleine Datenpakete an einen entfernten Rechner. Dann überprüft es ob das Ziel antwortet und wie lange es unterwegs war. Der Name des Programm leitet sich vom Sonar ab, mit dem durch akustische Signale Objekte (z. B. U-Boote) unter Wasser lokalisiert werden sollen.

Das Programm benutzt das ICMP-Protokoll für seine Aktionen. Es sendet ein Paket vom Typ *ECHO-REQUEST* an den Zielrechner. Ist dieser erreichbar und nicht gerade von einer Firewall geschützt, so antwortet dieser mit einem Paket vom Typ *ECHO-REPLY*. Ist der Zielrechner nicht erreichbar, so meldet sich der letzte erreichbare Router auf dem vermeintlich richtigen Weg zum Zielrechner mit einem Paket vom Typ *DESTINATION-UNREACHABLE*.

```
ping [OPTIONEN] HOST
```

Normalerweise sendet `ping`, wenn es einmal gestartet ist, alle Sekunde ein Paket an den Zielrechner. Den Vorgang können Sie ganz einfach mit der Tastenkombination `<Strg>+<c>` unterbrechen. Wenn Sie die Anzahl der Pings begrenzen wollen, können Sie Option `-c` verwenden.

```
defiant:~ # ping -c 3 www.fibel.org
PING www.fibel.org (193.254.185.32) 56(84) bytes of data.
64 bytes from vincent.netbeat.de (193.254.185.32): icmp_seq=1 ttl=52 time=110 ms
64 bytes from vincent.netbeat.de (193.254.185.32): icmp_seq=2 ttl=52 time=72.7 ms
64 bytes from vincent.netbeat.de (193.254.185.32): icmp_seq=3 ttl=52 time=69.8 ms

--- www.fibel.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 69.826/84.206/110.062/18.323 ms
```

Als Ausgabe zeigt der Befehl `ping` die Größe des Pakets an (56 Bytes). Durch den 8 Byte großen ICMP-Kopf wird das Antwortpaket auf 64 Bytes erhöht. Für jeden einzelnen Ping wird die Adresse des Zielhosts, die laufende Pingnummer (`icmp_seq`), die *Time To Live* (TTL) und die Zeit für eine Rundreise angeben. Zum Abschluß der Ausgabe macht der Befehl noch eine Statistikkzusammenfassung. Sie können die Anzahl der gesendeten und empfangenen Pakete sehen. Dazu wird auch noch die Verlustrate in Prozent angegeben und die Zeit, die für das Senden und Empfangen der Pakete insgesamt benötigt wurde. Die letzte Zeile gibt eine Statistik der Rundreisezeiten (round trip) aus. Das Minimum, arithmetische Mittel, das Maximum und die Standardabweichung werden hier in Millisekunden angegeben.

Natürlich kennt der Befehl `ping` eine Reihe von Optionen. Eine Auswahl von Optionen finden Sie in der folgenden Liste.

### Optionen

<code>-b</code>	Erlaubt Ping an eine Broadcastadresse
<code>-c ANZAHL</code>	Legt die Anzahl der gesendeten Pakete fest
<code>-f</code>	Flood ping, so viele Pings wie möglich pro Zeiteinheit
<code>-i INTERVALL</code>	Wartezeit zwischen den Paketen in Sekunden
<code>-I INTERFACE</code>	Ping geht über das angegebene Interface
<code>-l ANZAHL</code>	Sende eine Anzahl von Paketen ohne auf eine Antwort zu warten, nur <i>root</i> kann mehr als 3 angeben.
<code>-n</code>	Es erfolgt keine Namensauflösung für IP-Adressen
<code>-q</code>	Nur die Zusammenfassungszeilen werden ausgegeben
<code>-s GROESSE</code>	Die Größe des zu sendenden Pakets in Bytes
<code>-t TTL</code>	Setzt die Time To Live fest
<code>-v</code>	Ausführliche Ausgabe
<code>-w TIMEOUT</code>	Dauer der Prüfung in Sekunden
<code>-W TIMEOUT</code>	Maximale Wartezeit auf ein Paket

Um herauszufinden welche Rechner in einem Netz aktiv sind, können Sie einen Ping auf die Broadcast-Adresse senden. Die Option `-b` erlaubt Ihnen diesen sonst verbotenen Vorgang.

```
dozent@linux37:~> ping -c 2 -b 217.89.70.63
WARNING: pinging broadcast address
PING 217.89.70.63 (217.89.70.63) 56(84) bytes of data.
64 bytes from 217.89.70.37: icmp_seq=1 ttl=64 time=0.136 ms
64 bytes from 217.89.70.62: icmp_seq=1 ttl=128 time=0.244 ms (DUP!)
64 bytes from 217.89.70.60: icmp_seq=1 ttl=255 time=0.248 ms (DUP!)
64 bytes from 217.89.70.56: icmp_seq=1 ttl=64 time=0.253 ms (DUP!)
64 bytes from 217.89.70.50: icmp_seq=1 ttl=64 time=0.256 ms (DUP!)
64 bytes from 217.89.70.42: icmp_seq=1 ttl=64 time=0.277 ms (DUP!)
64 bytes from 217.89.70.49: icmp_seq=1 ttl=64 time=0.281 ms (DUP!)
64 bytes from 217.89.70.38: icmp_seq=1 ttl=64 time=0.286 ms (DUP!)
64 bytes from 217.89.70.41: icmp_seq=1 ttl=64 time=0.303 ms (DUP!)
64 bytes from 217.89.70.39: icmp_seq=1 ttl=64 time=0.307 ms (DUP!)
64 bytes from 217.89.70.35: icmp_seq=1 ttl=64 time=0.326 ms (DUP!)
64 bytes from 217.89.70.46: icmp_seq=1 ttl=64 time=0.331 ms (DUP!)
64 bytes from 217.89.70.47: icmp_seq=1 ttl=64 time=0.335 ms (DUP!)
64 bytes from 217.89.70.40: icmp_seq=1 ttl=64 time=0.352 ms (DUP!)
64 bytes from 217.89.70.61: icmp_seq=1 ttl=255 time=1.89 ms (DUP!)
64 bytes from 217.89.70.37: icmp_seq=2 ttl=64 time=0.131 ms

--- 217.89.70.63 ping statistics ---
2 packets transmitted, 2 received, +14 duplicates, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.131/0.372/1.890/0.396 ms
```

Einen ganze Flut von Pings können Sie mit der Option `-f` auf ein Ziel loslassen. Diese Funktion, die so viele Pings wie möglich absendet, steht nur dem Benutzer *root* zu Verfügung. Im Gegensatz zum normalen Ping werden die einzelnen Pakete nicht als Zeile aufgeführt, sondern als einzelner Punkt. Für jedes empfangene Paket wird ein Punkt aus der Liste entfernt. Je länger die Reihe der Punkte ist, desto schlechter ist die Laufzeit der Pakete.

```
dozent@linux37:~> ping -f www.fibel.org
PING www.fibel.org (193.254.185.32) 56(84) bytes of data.
ping: cannot flood; minimal interval, allowed for user, is 200ms
dozent@linux37:~> su
Password:
linux37:/home/dozent # ping -f www.fibel.org
PING www.fibel.org (193.254.185.32) 56(84) bytes of data.

--- www.fibel.org ping statistics ---
```

```
76 packets transmitted, 76 received, 0% packet loss, time 4914ms
rtt min/avg/max/mdev = 69.847/130.882/307.052/86.131 ms, pipe 21, ipg/ewma 65.528/249.178 ms
```

Weitere Optionen und Funktionen können Sie der Manualpage **ping(8)** entnehmen.

### 18.3.2 Wegverfolgung: traceroute

Das Programm **traceroute** ist im Prinzip eine erweiterte Form des Befehls **ping**. Es ermöglicht den Weg eines Pakets über die Router zu seinem Ziel zu verfolgen. Im Gegensatz zu **ping**, daß auf ICMP beruht, benutzt **traceroute** das Protokoll UDP.

Die Arbeitsweise ist recht einfach. Der Befehl sendet einfach 3 Pakete auf willkürlich ausgewählte Ports des Zielrechners. Dabei wird inständig gehofft, daß auf mindestens einem der Port kein Server lauscht. Um den Weg zu verfolgen, wird der ersten Paketserie nur eine *Time To Live* von 1 mitgegeben. Beim ersten Router wird dieser Wert auf 0 gesetzt und der Router sendet ein Paket vom Typ *ICMP-TIME-EXCEEDED* als IP-Paket zurück. Dieses Paket enthält die IP-Nummer des Routers, die dann ausgegeben wird. Die zweite Paketserie bekommt eine TTL von 2 und wird daher erst beim zweiten Router verworfen. Die dritte Serie bekommt eine TTL von 3 und so geht der ganze Spaß weiter bis der Zielrechner erreicht wurde. Dieser sendet nun ein Paket vom Typ *PORT-UNREACHABLE* zurück und zeigt damit an, daß das Ziel erreicht wurde.

```
traceroute [OPTIONEN] HOST
```

Der einfache Aufruf von **traceroute** mit eine Zieladresse zeigt die Liste der Router, über die ein Paket zum Zielhost geleitet wird.

```
linux37:~ # traceroute www.fibel.org
traceroute to www.fibel.org (193.254.185.32), 30 hops max, 40 byte packets
 1  217.89.70.62  0.247 ms   0.117 ms   0.116 ms
 2  217.89.70.125 0.287 ms   0.274 ms   0.271 ms
...
```

Eine Auswahl von Optionen, die **traceroute** unterstützt, finden Sie in der folgenden Liste.

#### Optionen

<b>-f</b> <i>TTL</i>	TTL für die erste Paketserie (Standard: 1)
<b>-I</b> <i>INTERFACE</i>	Kommunikation erfolgt über das angegeben Interface
<b>-m</b> <i>MAXHOPS</i>	Maximale Anzahl von getesteten Sprüngen
<b>-n</b>	Es erfolgt eine Namensauflösung für die Router
<b>-p</b> <i>PORT</i>	Startnummer für die zu testenden Ports (Standard: 33434)
<b>-w</b> <i>SEKUNDEN</i>	Wartezeit zwischen den Paketserien
<b>-q</b> <i>PAKETE</i>	Anzahl der zu sendenden Pakete (Standard: 3)

Die zu testenden Ports werden durch eine einfache Regel bestimmt. Dabei wird ein Anfangsport genommen (Standard: 33434) und auf den Port wird die Nummer des Hops addiert und eins abgezogen. Diese Form der Portbestimmung kann zu Problemen führen, wenn der ausgewählte Port vom Rechner benutzt wird. Sie können daher mit der Option **-p** die Anfangsportnummer festsetzen.

```
linux37:~ # traceroute -p 25000 www.fibel.org
traceroute to www.fibel.org (193.254.185.32), 30 hops max, 40 byte packets
 1  217.89.70.62  0.247 ms   0.117 ms   0.116 ms
 2  217.89.70.125 0.287 ms   0.274 ms   0.271 ms
...
```

Sollte es zu einem Timeout durch eine Zeitverzögerung des Pakete kommen, wird anstatt der Zeit eine Asterisk (\*) angezeigt. Dies kann durch zu hohe Netzlast oder durch eine Firewall verursacht werden. Einige Router antworten auch nicht mehr auf **traceroute**-Pakete.

### 18.3.3 netstat

Der Befehl `netstat` liefert Ihnen ein Überblick über den Zustand und die Konfiguration der Netzwerkfunktionalität eines Rechners. Netzwerkverbindungen, Routingtabellen, Schnittstellenstatistiken und vieles mehr können Sie sich mit dem Befehl ausgeben lassen.

`netstat [OPTIONEN]`

Wenn Sie keine Optionen angeben, zeigt der Befehl `netstat` den Zustand der offenen Sockets an. Dabei werden die offenen Sockets aller konfigurierbaren Adressfamilien ausgegeben.

```
dozent@linux37:~> netstat
Aktive Internetverbindungen (ohne Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 linux37.local:33324     195.71.11.67:www-http   VERBUNDEN
tcp        0      0 linux37.local:33321     jules.netbeat.de:ftp    VERBUNDEN
tcp        0      0 linux37.local:33320     linux40.amov.de:ssh     VERBUNDEN
tcp        0      0 linux37.local:33328     195.71.11.67:www-http   TIME_WAIT
tcp        0      0 linux37.local:33330     195.71.11.67:www-http   VERBUNDEN
tcp        0      0 linux37.local:33325     195.71.11.67:www-http   TIME_WAIT
tcp        0      0 linux37.local:33320     linux40.amov.de:ssh     VERBUNDEN
udp        0      0 linux37.local:33617     www-proxy.HH1.sr:domain VERBUNDEN
Aktive Sockets in der UNIX Domäne (ohne Server)
Proto RefZäh Flaggen      Typ      Zustand      I-Node Pfad
unix   2      [ ]          DGRAM           1332  /var/lib/named/dev/log
unix  11      [ ]          DGRAM           1329  /dev/log
unix   3      [ ]          STREAM    VERBUNDEN    40208  /tmp/.ICE-unix/dcop2042
-1089785143
...
```

Mit der Option `-p` werden zusätzlich die verwendeten Programme und deren PID mit angegeben.

```
linux37:~ # netstat -p
Aktive Internetverbindungen (ohne Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp        0      0 linux37.local:32779     217.89.70.41:ssh        VERBUNDEN  2790/ssh
udp        0      0 localhost:filenet-nch   localhost:filenet-nch   VERBUNDEN  1431/postmaster
Aktive Sockets in der UNIX Domäne (ohne Server)
Proto RefZäh Flaggen      Typ      Zustand      I-Node PID/Program name      Pfad
unix   2      [ ]          DGRAM           1332  935/syslogd           /var/lib/named/dev/log
unix  12      [ ]          DGRAM           1329  935/syslogd           /dev/log
unix   2      [ ]          DGRAM           26140  2815/su
unix   3      [ ]          STREAM    VERBUNDEN    17056  2264/kdeinit: dcops  /tmp/.ICE-unix/dcop2264-1089872880
```

Hier eine Auswahl der möglichen Optionen von `netstat`. Weitere Informationen und Optionen entnehmen Sie der Manualpage `netstat(8)`.

**Optionen**

<b>-a</b>	Zeigt alle wartenden Verbindungen/Sockets an
<b>-t   --tcp</b>	Zeigt alle offenen TCP-Verbindungen an
<b>-u   --udp</b>	Zeigt alle offenen UDP-Verbindungen an
<b>-x   --unix</b>	Zeigt alle offenen Unix-Sockets an
<b>-w   --raw</b>	Zeigt alle offenen RAW-Sockets an
<b>--ip   --inet</b>	Zeigt alle offenen TCP- und UDP-Verbindungen an (wie <b>-u</b> und <b>-t</b> )
<b>--ax25</b>	Zeigt alle offenen AX.25-Verbindungen an
<b>--ipx</b>	Zeigt alle offenen IPX-Sockets an
<b>--netrom</b>	Zeigt alle offenen NET/ROM-Verbindungen an
<b>-r   --route</b>	Gibt die Routingtabelle aus (siehe <b>route</b> )
<b>-i</b>	Liefert Statistik über die Schnittstellen
<b>-n   --numeric</b>	Es erfolgt keine Namensauflösung
<b>-p   --programms</b>	PID und Programmname werden angezeigt
<b>-c   --continuous</b>	Wiederholt die Anfrage in Sekundenabständen

Mit der Option **-a** können Sie sich alle lauschenden Sockets anzeigen lassen. So zeigt der folgende Befehl alle lauschenden TCP-Ports an.

```
linux37:~ # netstat -ta
```

Aktive Internetverbindungen (Server und stehende Verbindungen)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	*:sunrpc	*:*	LISTEN
tcp	0	0	*:x11	*:*	LISTEN
tcp	0	0	*:ftp	*:*	LISTEN
tcp	0	0	*:ipp	*:*	LISTEN
tcp	0	0	*:postgresql	*:*	LISTEN
tcp	0	0	localhost:smtp	*:*	LISTEN
tcp	0	0	localhost:953	*:*	LISTEN
tcp	0	0	linux37.local:32779	217.89.70.41:ssh	VERBUNDEN
tcp	0	0	*:www-http	*:*	LISTEN
tcp	0	0	*:domain	*:*	LISTEN

Mit der Option **-r** verhält sich **netstat** wie der Befehl **route** (18.1.2).

```
linux37:~ # netstat -r
```

Kernel IP Routentabelle

Ziel	Router	Genmask	Flags	MSS	Fenster	irtt	Iface
sub1.amov	*	255.255.255.224	U	0	0	0	eth0
10.0.1.0	*	255.255.255.0	U	0	0	0	eth0
default	217.89.70.62	0.0.0.0	UG	0	0	0	eth0

Statistiken über die vorhandenen Schnittstelle liefert die Option **-i**.

```
linux37:~ # netstat -i
```

Kernel Schnittstellentabelle

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	2663	0	0	0	1767	0	0	0	BMRU
lo	16436	0	72	0	0	0	72	0	0	0	LRU

### 18.3.4 telnet

Dieses Kommando erlaubt es sich interaktiv auf einen anderen Rechner anzumelden. Sie können sogar noch mehr machen. Im Prinzip können Sie mit diesem Befehl mit jedem TCP-Port Kontakt aufnehmen. Als Dienst für den Fernzugriff sollten Sie **telnet** nicht mehr verwenden, da die Verbindung nicht verschlüsselt ist und alle Kommandos, auch die Passwörter, unverschlüsselt übertragen werden. Deshalb war der Zugriff von **root** über **telnet** blockiert.

Trotzdem hat das Client-Programm **telnet** noch immer eine wichtige Bedeutung. So können Sie einfach über die Angabe eine IP-Adresse und des Ports mit beliebigen TCP-Ports verbinden.

telnet ADRESSE [PORT]

So können Sie z. B. eine Webseite über Telnet aufrufen. Sie senden einfach über `telnet` das normale HTTP-Kommando (GET).

```
linux37:~ # telnet www.leichtlernen.de 80
Trying 193.254.185.24...
Connected to www.leichtlernen.de.
Escape character is '^]'.
GET http://www.leichtlernen.de/index.html
<html>
<head>
<title>Leicht Lernen</title>
</head>
<body>
<h1>Leicht Lernen</h1>
...
</body>
</html>
Connection closed by foreign host.
```

Wie Sie sehen, ist das ganz einfach, aber nicht gerade komfortabel.

### 18.3.5 tcpdump

Bei dem Programm `tcpdump` handelt es sich um einen sogenannten *Netzwerk-Sniffer*. Solche Netzwerk-Sniffer analysieren alle Pakete, die bei einer Netzwerkschnittstelle ankommen. Normalerweise reagieren Netzwerkschnittstellen nur auf Pakete, die auch für sie bestimmt sind. Die Schnittstellen können aber auch in den sogenannten *Promiscuous Modus* geschaltet werden. In diesem Fall werden alle ankommenden Pakete akzeptiert und ausgewertet. Nur `root` kann eine Netzwerkschnittstelle in den Promiscuous Modus schalten. Daher kann der Befehl `tcpdump` auch nur von `root` verwendet werden.

`tcpdump` [OPTIONEN]

Da ein vernünftiger Einsatz von `tcpdump` tiefere Netzwerkkennntnisse voraussetzt, hier nur ein paar Optionen des Befehls.

#### Optionen

<code>-a</code>	Es wird versucht Netzwerk- und Broadcastadressen in Namen umzuwandeln
<code>-c PAKETE</code>	Hört nach der angegebenen Anzahl von Pakete mit dem Sniffen auf.
<code>-i INTERFACE</code>	Horcht auf die angegebene Schnittstelle
<code>-n</code>	Führt keine Namensauflösung durch
<code>-v</code>	Ausführlichere Ausgabe

Ein Flushing-Ping auf Ihren Rechner können Sie leicht mit dem Befehl `tcpdump` feststellen. Eine Vielzahl von ICMP-Paketen vom Typ "echo request" innerhalb kurzer Zeit deuten daraufhin.

```
linux37:/etc # tcpdump
tcpdump: listening on eth0
10:36:15.576001 217.89.70.46 > linux37.local: icmp: echo request (DF)
10:36:15.576019 linux37.local > 217.89.70.46: icmp: echo reply
10:36:15.576352 217.89.70.46 > linux37.local: icmp: echo request (DF)
10:36:15.576370 linux37.local > 217.89.70.46: icmp: echo reply
10:36:15.576589 217.89.70.46 > linux37.local: icmp: echo request (DF)
10:36:15.576610 linux37.local > 217.89.70.46: icmp: echo reply
...
```

## XXI

Arbeitsblatt  
Routing

## 21.1

Für diese Aufgaben bilden Sie bitte Gruppen mit drei Personen und drei Rechnern.

- 1** Lassen Sie sich für alle drei Rechner die Netzwerkschnittstellen mit ihren Daten anzeigen. Welche Informationen erhalten Sie?
- 2** Wie sind die Routen auf den Rechnern angelegt?
- 3** Notieren Sie sich die IP-Nummer, Netzwerkmaske, Broadcastadresse und Gateway für die drei Rechner.
- 4** Sorgen Sie dafür, daß auf jedem Rechner die Namen "rechner1", "rechner2" und "rechner3" auf die passenden IP-Nummern verweisen. Als Domainname wählen Sie bitte "test.tux".
- 5** Pingen Sie nun jeweils von jedem Rechner die anderen Rechner an. Probieren Sie es mit der IP-Nummer und dem Rechnernamen. Können alle Rechner alle sehen?
- 6** Wie lautet die Adresse des privaten A-Klasse-Netzes?
- 7** Wie lautet die Netzwerkmaske für ein A-Klasse-Netz und wie muß sie lauten, wenn ich es in Netze von der Größe der C-Klasse unterteilen will?
- 8** Wählen Sie bitte nach Absprache mit den anderen Arbeitsgruppen ein Subnetz (C-Klassen-Größe) der Form 10.1.X.0, wobei X der wählbare Teil ist. Wie lautet nun die Netzwerknummer, Netzwerkmaske, Broadcastadresse und der Bereich der für Hosts zur Verfügung stehenden Nummern?
- 9** Ändern Sie die Einstellung der Netzwerkkarte für Rechner 1 auf die IP-Nummer 10.1.X.1. Überprüfen Sie nun mit `ifconfig` ob Netzwerkmaske und Broadcast-Adresse richtig gesetzt worden sind.
- 10** Ändern Sie die Einstellung der Netzwerkkarte für Rechner 1 auf die IP-Nummer 10.1.X.1 und die passende Netzwerkmaske. Ist auch die Broadcast-Adresse nun richtig?
- 11** Richten Sie nun die Netzwerkkarte so ein, daß alle Angaben richtig sind. Ändern Sie bitte auch die Namensauflösung entsprechend.
- 12** Welche Routen gelten nun auf Rechner 1?
- 13** Versuchen Sie die beiden anderen Rechner von Rechner 1 aus mit einem Ping zu erreichen. Klappt das?
- 14** Um die beiden Rechner zu erreichen, tragen Sie eine Route mit der Netzwerknummer der beiden Rechner ein. Warum können Sie die Rechner immer noch nicht erreichen?
- 15** Was benötigen Sie damit zwei Rechner in verschiedenen Netzen sich unterhalten können?
- 16** Richten Sie nun auf dem Rechner 2 eine Alias-IP-Nummer 10.1.X.2 mit der richtigen Netzwerkmaske und Broadcast-Adresse ein.
- 17** Wie sehen die Routen auf Rechner 2 aus?
- 18** Können Sie nun von Rechner 1 aus den Rechner 2 über die neue Nummer anpingen?
- 19** Versuchen Sie Rechner 2 über die alte IP-Adresse zu erreichen. Warum klappt das nicht?
- 20** Legen Sie bitte nun auf Rechner 1 eine Route an, die den Rechner 2 auch über seine alte IP-Adresse erreichbar macht. Testen Sie dies.
- 21** Ist der Rechner 3 jetzt auch von Rechner 1 erreichbar? Wenn nicht, was muß auf Rechner 2 noch gemacht werden, damit dies möglich ist?
- 22** Führen Sie Ihre Überlegungen auf Rechner 2 aus und testen Sie ob Ihre Änderungen erfolgreich waren.

.....  
*Notizen:*

.....



# Kapitel 19

## X-Window

*Warum die Shell benutzen? Klicki-Klicki-Bunti-Bunti ist doch viel schöner.*

*André Helsberg, LPI-Kurs 2004*

Für die meisten Aufgaben ist die Shell ein nützliches und praktisches Element. Aber auch das beste Werkzeug hat seine Grenzen. Denn selbst eingefleischte Kommandozeilenhacker müssen zugeben, daß Bildbearbeitung, Spiele, Raytracing und andere bildlich orientierten Anwendungen auf einer graphischen Oberfläche einfach mehr Spaß machen. Selbst die Hardcore-Shellbenutzer nutzen das X-Window-System – um nämlich eine Menge X-Terminals gleichzeitig offen zu haben.

Da sind wir auch schon beim Thema. Die graphische Standardoberfläche für Linux ist das *X-Window-System*. Es wird auch *X-Window* genannt oder sogar nur mit *X11* oder *X* bezeichnet. Beachten Sie bitte, daß es X-Window und nicht X-Window<sub>s</sub> heißt. Diese graphische Oberfläche wurde eben nicht in Redmond sondern in den Jahren 1985 bis 1987 am *MIT* (*Massachusetts Institute of Technology*) entwickelt.

Eine Implementierung des X-Window in der *Version 11 Release 6* (X11R6) ist *XFree86*. Diese quelloffene Software wurde speziell für Intel-kompatible Rechner entwickelt, was sich in der Zahl 86 im Namen ausdrückt. Im Moment sind zwei Versionen im Umlauf: Die alte Version 3, die noch für alte Grafikkarten eingesetzt wird und die aktuelle Version 4 für neue Grafikkarten. Andere graphische Oberflächen haben unter Linux praktisch keine Bedeutung.

Den Informationswilligen stellt SuSE (jedenfalls in der Version 9.0) erst einmal vor ein Problem. Die Manualpages zu den für X11 relevanten Programmen und Konfigurationsdateien müssen erst einmal nachinstalliert werden. Dazu müssen Sie über YaST das Paket *XFree86-man* installieren.

### 19.1 Aufbau

Die graphische Oberfläche X11 basiert auf dem *X-Protokoll*. Dieses Protokoll ermöglicht die Übertragung von graphischen Grundoperationen über ein Netzwerk. Das X11 ist also ein Client-Server-System, wobei sich meistens aber Client und Server auf einem Rechner befinden.

Der *X-Server* erledigt die Darstellung der Graphik auf der Workstation an der der Benutzer sitzt. Daneben verarbeitet er die Informationen, die über Maus, Tastatur oder andere Eingabegeräte eintreffen. Diese sendet er an die laufenden Anwendungsprogramme.

Ein Anwendungsprogramm, das X11 nutzt, wird als *X-Client* bezeichnet. Es kann auf der Workstation laufen oder auf einem anderen Rechner und seine Graphikbefehle übers Netzwerk an den X-Server senden. Die Graphikbefehle legen fest, was durch den X-Server dargestellt wird.

Die Stärke des X11 liegt in der möglichen räumlichen Trennung von X-Client und X-Server.

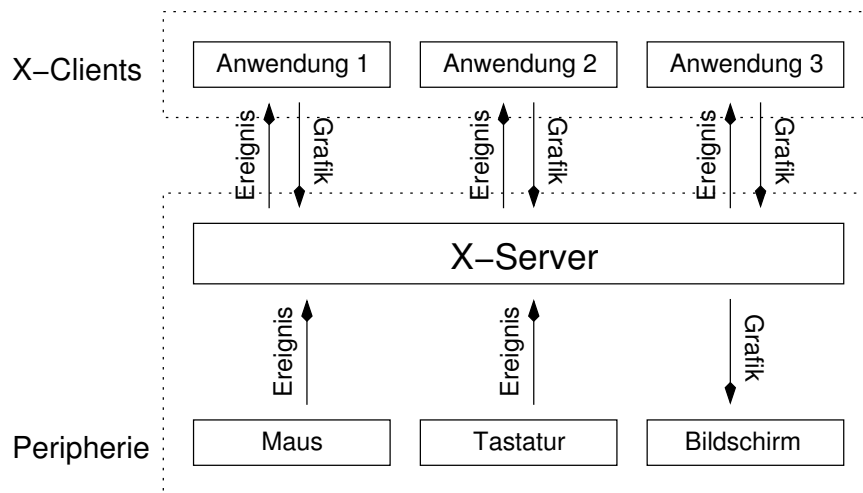


Abbildung 19.1: Das X-Window-System als Client-Server-System

### 19.1.1 X-Protokoll

Das in den 80er Jahren des vorherigen Jahrhunderts entwickelte *X-Protokoll* stellt Operationen und Funktionen für eine graphische Oberfläche zur Verfügung. Die genaue Definition des Protokolls finden Sie in dem *RFC 1013 "X Window System Protocol, version 11: Alpha"*<sup>1</sup> aus dem Jahre 1987.

Die Operationen des X-Protokolls sind ziemlich primitiv. Einfache Grafikoperationen wie das Zeichnen von Punkten und Linien, sowie einfache geometrische Objekte wie Kreise und Rechtecke sind möglich. Dazu kommt natürlich auch noch die Darstellung von Zeichenketten. Daneben stellt das Protokoll auch Funktionen zur Verwaltung von Fenstern zur Verfügung. Im Prinzip sind Fenster Flächen, auf die sich bestimmte Ereignisse beziehen. Weitere Funktionen des Protokolls dienen zur Verwaltung der internen Organisation.

Schauen wir uns doch mal ein einfaches Beispiel an. Das Anwendungsprogramm (X-Client) bietet ein Aufklappmenü an. Bewegt nun der Benutzer die Maus über das Aufklappmenü und betätigt dann die linke Maustaste, dann sendet der X-Server dieses Ereignis an das Anwendungsprogramm. Da nach einem Mausklick auf ein solches Menü weitere Menüpunkte in einem Menüfenster auftauchen soll, sendet der X-Client Anweisungen an den X-Server, wie dieses neue Menüfenster auszusehen hat. Jede weitere Mausbewegung wird vom X-Server an den X-Client weitergeleitet, der wiederum auf diese Ereignisse mit graphischen Darstellungen reagiert. Hat der Anwender endlich seinen Menüpunkt ausgewählt, dann wird der folgende Mausklick wieder vom X-Server an den X-Client übertragen. Das Anwendungsprogramm weiß nun, welche Aktion erfolgen soll und startet den entsprechenden Programmteil.

### 19.1.2 Fenstermanager

Das X-Window-Server ist allein für die Darstellung und die Weiterleitung von Ereignissen verantwortlich. Nicht mehr und nicht weniger. Um mit dem System komfortabel arbeiten zu können, fehlt noch ein wichtiges Element: Der *Fenstermanager*. Der Fenstermanager (engl. *window manager*) ermöglicht es dem Anwender die Größe und die Position von Fenstern festzulegen. Daneben sorgt er noch einen dekorativen Fensterrahmen, der meistens weitere Funktionen (z. B. Verkleinern (Icon), Vollbild und Schließen) zur Verfügung stellt. Dabei sind die Fenstermanager gewöhnliche X-Clients, wie auch die Shell ein normales Konsolenprogramm ist.

#### twm

Der Name **twm** steht für *Tab Window Manager*. Ebenfalls üblich ist die Bezeichnung *Thom's Window Manager* nach seinem Entwickler *Tom LaStrange*. Dieser Fenstermanager unterstützt Fensterrahmen mit Titelleisten und Icon-Management. Fenster können durch Anklicken oder durch die Position des Mauszeigers

<sup>1</sup>Den Text des RFC 1013 (Request For Comments) erhalten Sie auf der Homepage des "RFC Editors" (<http://www.rfc-editor.org/>).

den Tastaturfokus erhalten. Er ermöglicht auch die individuelle Definition von Tastaturkombinationen und Maustasten.

Inzwischen ist der `twm` veraltet und wird in seinen Möglichkeiten von anderen aktuellen Fenstermanager weit hinter sich gelassen. Früher war er für eine Zeit die einzige sinnvolle Wahl für einen Fenstermanager. Daher basieren viele moderne Fenstermanager auf ihm. Trotz seines Alters hat er einen Vorteil, da er eigentlich auf fast allen Systemen installiert wird.

### **fvwm, fvwm2 und fvwm95**

Der von *Robert Nation* entwickelte `fvwm` war einst ein dominanter Fenstermanager in der Linuxgemeinschaft. Er basiert stark auf dem Fenstermanager `twm`. `fvwm` benötigt nur geringen Speicherbedarf und bietet einen einfachen 3-D-Look. Eine wichtige Eigenschaft ist die Unterstützung einfacher virtueller Desktops. Unter dem Namen `fvwm2` kursierte lange die Version 2 des Fenstermanagers. Heute ist die Bezeichnung nicht mehr gängig, sondern Sie finden auch die zweite Version unter der Bezeichnung `fvwm`. Eine besondere Form war der `fvwm95`. Diese gepatchte Version des `fvwm` simulierte das Look-And-Feel von Windows 95.

### **ctwm**

Der von *Claude Lecommandeur* entwickelte Fenstermanager `ctwm` ist auch eine Erweiterung von `twm`. Er unterstützt bis zu 32 virtuelle Bildschirme, die als *Workspaces* bezeichnet werden. Dabei können Sie zwischen den einzelnen Workspaces durch das Betätigen einer Schaltfläche im Workspace-Manager wechseln oder durch den Aufruf einer Funktion. Jeder Workspace kann individuell eingerichtet werden. Der Fenstermanager unterstützt die 3D-Darstellung von Rahmen und Titelzeilen. Eine Karte ermöglicht das einfache Verschieben von Fenstern zwischen den Workspaces. Dabei kann ein Fenster ohne weiteres auch über mehrere Workspaces sich ausdehnen.

### **wm2**

Einer der einfachsten, schnellsten und kleinsten Fenstermanager ist `wm2`. Als einzige Funktion versieht er die Anwendungsfenster mit einem Rahmen. Da er nicht den Ballast weiterer Funktionen wie andere Fenstermanager mitschleppen muß ist er ideal für den Einsatz in ressourcenschwachen Systemen. Trotz der mageren Funktionen sind die Rahmen dennoch stylisch gut ausgearbeitet.

### **Window Maker**

Der von *Alfredo Kojima* entwickelte *Window Maker* ist ein Fenstermanager, der die NeXT-Oberfläche emuliert. Er besitzt ein Dock für Applikationen, das auch Drag-and-Drop unterstützt. Neben der Unterstützung für mehrere Sprachen und von GNUstep-Anwendungen, zeichnet er sich durch vor allem durch graphische Elemente aus.

## **19.1.3 Displaymanager**

Neben den Fenstermanagern spielen auch die *Displaymanager* eine wichtige Rolle. Immer dann, wenn das X11 nicht von einem Benutzer aus der Shell heraus gestartet wird, sondern sofort nach dem Booten zur Verfügung stehen soll, kommen diese Programme zum Einsatz. Sie sind wie die Fenstermanager auch nur einfache X-Clients. Nach der Konvention des LSB wird das X11 im Runlevel 5 gestartet. Eines der Init-Skripte dieses Runlevels startet das X-Windows-System und gleichzeitig für dieses System einen Displaymanager. Dieser bietet nun dem Anwender ein graphisches Login-Fenster. Loggt sich der Anwender ein, dann wird das X11 beendet und für den Benutzer ein neues X11 mit seinen Einstellungen (Fenstermanager, Applikationen) gestartet. Sobald dieses System beendet wird, startet wieder ein X11 mit dem Displaymanager.

Der Displaymanager ist nicht beschränkt auf den lokalen Rechner. Der Displaymanager kann auch Displays auf anderen Rechnern verwalten. Damit wird der lokale Rechner zu einem X-Terminal degradiert und ist nun nicht mehr für die Ausführung sondern nur noch für die Darstellung der Anwendungen zuständig. Dies

ermöglicht auch den Einsatz von leistungsschwächeren Systemen für anspruchsvollere Software, die dann auf einem leistungsstarken Rechner laufen kann.

## 19.2 Konfiguration und Einrichtung

### 19.2.1 Installation

Auf die Installation des X11 möchte ich in diesem Skript nicht genau eingehen. Die meisten Distributionen bringen ein funktionstüchtiges X-Window-System bereits mit, daß in den meisten Fällen ohne Probleme installiert werden kann. Beim Einsatz von Flachbildschirmen sollten Sie bei der Installation auf die richtige Konfiguration der Monitore achten, da zu hohe Frequenzen zur Abschaltung des Monitors führen können. Führen Sie daher auf jeden Fall am Ende der X11-Einstellungen immer einen Test aus. Sollte der Test fehl schlagen und Sie nur noch einen schwarzen Monitor sehen können, dann können Sie den Test mit der Tastenkombination <Strg>+<Alt>+<←> beenden.

Die Mindestanforderung für ein X11-System sind mindestens 12 MB Arbeitsspeicher, einer VGA-Grafikkarte mit einer Auflösung von 800x600 Pixel und möglichst eine Maus. Wie gesagt, dies ist eine Minimalausstattung. Sollten Sie auf die Idee kommen Desktopumgebungen wie KDE oder Gnome einzusetzen, heißt die Devise ganz einfach: "So viel Speicher wie möglich!" Denn erst dann macht die Arbeit mit dem X-Window-System richtig Spaß.

Die X11-Pakete der Distributionen enthalten die X-Server und ihre Komponenten, die Clients und Client-Bibliotheken, sowie Schriften und Konfigurationsdateien. Auf Server-Systemen ist eine Installation eines X-Servers meistens unnötig. Sie können aber die Clients und die dazugehörigen X-Bibliotheken installieren. Dann ist es möglich die Anwendungen auf dem Server auszuführen und die Darstellung auf einen anderen Rechner zu übertragen.

### 19.2.2 Die Konfigurationsdatei XF86Config

Die Konfigurationsdatei für den X-Server finden Sie im Verzeichnis `/etc/X11` unter dem Namen `XF86Config`. In dieser Datei finden Sie die Konfigurationsanweisungen in Abschnitte aufgeteilt, die mit dem Schlüsselwort `Section` und der Bezeichnung des Abschnitts beginnen und mit dem Schlüsselwort `EndSection` enden. Das folgende Beispiel zeigt einen Ausschnitt aus dem Abschnitt "File".

```
Section "Files"
    FontPath      "/usr/X11R6/lib/X11/fonts/misc:unscaled"
    ...
    InputDevices  "/dev/pointer3"
EndSection
```

Die meisten Abschnittstypen können auch mehrfach vorkommen. Auf die Groß- und Kleinschreibung brauchen Sie außer bei Dateinamen nicht zu achten. Schauen wir uns doch mal die verschiedenen Abschnitte an.

#### Files

Der Abschnitt `Files` enthält Pfadangaben zu Dateien, Verzeichnissen und Geräten, die der X-Server benötigt. Die folgenden Direktiven können in diesem Abschnitt auftauchen:

**FontPath** Die Pfade zu den Schriftenverzeichnissen bzw. die Adresse eines Fontservers können hier eingetragen werden. Sollte diese Direktive nicht vorhanden sein, dann werden die folgenden Standardverzeichnisse verwendet.

```
/usr/X11R6/lib/X11/fonts/misc/
/usr/X11R6/lib/X11/fonts/Speedo/
/usr/X11R6/lib/X11/fonts/Type1/
/usr/X11R6/lib/X11/fonts/CID/
```

```
/usr/X11R6/lib/X11/fonts/75dpi/
/usr/X11R6/lib/X11/fonts/100dpi/
```

**InputDevices** Die möglichen Geräteschnittstellen für Eingabegeräte werden hier angegeben.

**ModulePath** Diese Direktive legt das Verzeichnis für zusätzliche X-Module fest. In den meisten Fällen liegen die Module im Verzeichnis `/usr/X11R6/lib/modules`.

**RgbPath** Hiermit wird der Pfad zur RGB-Farbtabelle angegeben. Sollte die Direktive nicht angegeben werden, so wird der fest einkompilierte Pfad `/usr/X11R6/lib/X11/rgb` verwendet.

#### Section "Files"

```
FontPath      "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
FontPath      "/usr/X11R6/lib/X11/fonts/Type1"
FontPath      "/usr/X11R6/lib/X11/fonts/truetype"
InputDevices  "/dev/ttyS0"
InputDevices  "/dev/mouse"
InputDevices  "/dev/usbmouse"
ModulePath    "/usr/X11R6/lib/modules"
RgbPath       "/usr/X11R6/lib/X11/rgb"
```

EndSection

#### ServerFlags

Der Abschnitt **ServerFlags** kontrolliert das Verhalten des X-Servers. Das hier auftretende Schlüsselwort ist "Option". Die hier festgelegten Werte können im Abschnitt **ServerLayout** oder beim Starten des X-Servers überschrieben werden.

#### Section "ServerFlags"

```
Option        "AllowMouseOpenFail"
Option        "DontVTSwitch" "on"
```

EndSection

Hier eine Auswahl von möglichen Optionen.

**DontVTSwitch** Deaktiviert die Funktion der Tastensequenz `<Strg>+<Alt>+<Fn>`, die normalerweise das Hin- und Herschalten zwischen den Konsolen ermöglicht. Mögliche Werte sind "on" und "off", wobei "off" die Standardeinstellung ist.

**DontZap** Deaktiviert die Funktion der Tastensequenz `<Strg>+<Alt>+<←>`, die normalerweise den X-Server beendet. Mögliche Werte sind "on" und "off", wobei "off" die Standardeinstellung ist.

**DontZoom** Deaktiviert die Funktion der Tastensequenz `<Strg>+<Alt>+<Nummernfeld-Plus>` und `<Strg>+<Alt>+<Nummernfeld-Minus>`, die normalerweise einen Wechsel zwischen den verschiedenen Auflösungsmodi ermöglichen. Mögliche Werte sind "on" und "off", wobei "off" die Standardeinstellung ist.

**AllowMouseOpenFail** Erlaubt dem X-Server zu starten, selbst wenn kein Mausgerät geöffnet oder initialisiert werden konnten. Mögliche Werte sind "true" und "false", wobei "false" die Standardeinstellung ist.

**XkbDisable** Aktiviert oder deaktiviert die Xkeyboard-Erweiterung. Diese Option kann durch die Kommandozeilenoption `-kb` überschrieben werden. In der Standardeinstellung ist XKB aktiviert.

## Module

Dieser Abschnitt legt die zusätzlich zu ladenden grafikartenunabhängigen Module fest. Dies können z. B. eine Unterstützung von bestimmten Schriften enthalten und dienen zur direkten Videodarstellung durch den X-Server. Die Direktive `Load` gibt dabei den Namen des Moduls an, dessen Pfad im Abschnitt `Files` festgelegt wurde.

```
Section "Module"
    Load      "dbe"
    Load      "type1"
    Load      "speedo"
    Load      "freetype"
    Load      "extmod"
    Load      "glx"
    Load      "v4l"
EndSection
```

So ist z. B. die passende Datei des Moduls "speedo" im Verzeichnis `/usr/X11R6/lib/modules/fonts` zu finden.

```
ole@enterprise:~> find /usr/X11R6/lib/modules -name "*speedo*"
/usr/X11R6/lib/modules/fonts/libspeedo.a
```

## InputDevice

Der Abschnitt `InputDevice` konfiguriert jeweils ein Eingabegerät. Dies kann z. B. die Maus oder die Tastatur sein. Da es meistens mehrere Eingabegeräte gibt, kommt dieser Abschnitt mehrfach vor. Der Abschnitt enthält drei verschiedene Direktiven. Die Direktive `Identifier` gibt dem Abschnitt einen eindeutigen Namen, mit dem dann aus anderen Abschnitten heraus auf diese Konfiguration Bezug genommen werden kann. Der passende Treiber für das Eingabegerät wird über die Direktive `Driver` angegeben. Mittels der Direktive `Option` können gerätespezifische Werte eingetragen werden. Das folgende Beispiel zeigt zwei `InputDevice`-Abschnitte. Die erste Abschnitt konfiguriert die Tastatur und der zweite die Maus.

```
Section "InputDevice"
    Driver      "Keyboard"
    Identifier   "Keyboard[0]"
    Option      "Protocol" "Standard"
    Option      "XkbLayout" "de"
    Option      "XkbModel" "pc105"
    Option      "XkbRules" "xfree86"
    Option      "XkbVariant" "nodeadkeys"
EndSection
```

```
Section "InputDevice"
    Driver      "mouse"
    Identifier   "Mouse[1]"
    Option      "Device" "/dev/pointer1"
    Option      "Name" "Autodetection"
    Option      "Protocol" "imps/2"
    Option      "Vendor" "Sysp"
EndSection
```

Die `InputDevice`-Abschnitte werden nur dann berücksichtigt, wenn Sie durch einen aktiven `ServerLayout`-Abschnitt referenziert werden.

## Modes

Die eigentliche Einstellung der Grafikkarte erfolgt in diesem Abschnitt. Hier wird die Parameter für den Betrieb der Grafikkarte festgelegt. Die Einträge hier erfordern genaueste Kenntnisse der Hardware und sind der kritischste Punkt der X-Server-Konfiguration. Zum Glück übernehmen die heutigen Konfigurationsprogramme die Ermittlung der gängigen Werte für die Grafikmodi. Es sind mehrere Modes-Abschnitte möglich, daher enthält jeder Abschnitt eine Direktive `Identifier` zur eindeutigen Benennung des Abschnitts. Die Direktive `Modeline` definiert dann die verschiedenen Auflösungen und die dafür notwendigen Grafikkartenparameter.

```
Section "Modes"
    Identifier    "Modes[0]"
    Modeline      "1280x1024" 114.54 1280 1360 1496 1712 1024 1025 1028 1062
    Modeline      "1152x864" 116.71 1152 1240 1368 1584 864 865 868 906
    Modeline      "1024x768" 67.48 1024 1080 1184 1344 768 769 772 797
    Modeline      "800x600" 40.19 800 832 912 1024 600 601 604 623
    Modeline      "640x480" 25.10 640 656 720 800 480 481 484 498
EndSection
```

## Monitor

Auch der Abschnitt `Monitor`, der die Monitoreigenschaften enthält, kann mehrfach vorkommen. Daher ist auch hier die Direktive `Identifier` vorhanden. Mit der Direktive `UseModes` wird der zu verwendende Modes-Abschnitt, dessen Einstellungen auch abhängig vom Monitortyp sind, festgelegt. Den Hersteller und das Modell des eingesetzten Monitors stehen in den Direktiven `VendorName` und `ModelName`. Die Größe des Bildschirms legt die Direktive `DisplaySize` fest, wobei die Größe in Millimetern angegeben wird. Die wichtigsten Direktiven, die Sie sich auch aus dem Monitorhandbuch besorgen können, sind `HorizSync` und `VertRefresh`. In `HorizSync` steht der Bereich der horizontalen Synchronisationsfrequenz, die in kHz angegeben wird. Dagegen wird die Wiederholungsfrequenz der Bilder (`VertRefresh`) in Hz angegeben. Tragen Sie hier bitte nicht zu kleine Werte ein. Zu hohe Frequenzen können Ihrem Monitor schwere Schäden zufügen.

```
Section "Monitor"
    Identifier    "Monitor[0]"
    VendorName    "--> VESA"
    ModelName     "1280X1024@85HZ"
    HorizSync     31-91
    VertRefresh   50-85
    DisplaySize   320 240
    UseModes      "Modes[0]"
    Option        "CalAlgorithm" "CheckDesktopGeometry"
EndSection
```

## Device

Hier wird die verwendete Grafikkarte definiert. Da es in einem Rechner auch mehrere Grafikkarten geben kann, ist auch die Abschnitt möglicherweise mehrfach vorhanden und benötigt deshalb die Direktive `Identifier`. Die Direktive `Driver` legt den zu verwendenden Treiber fest, während die Position im PCI-Bus in der Direktive `BusID` steht. Diese ist vor allem wichtig, wenn zwei gleiche Grafikkarten sich im Rechner befinden. Interessant wird es auch, wenn die Grafikkarte zwei Monitor unterstützt. In diesem Fall kommt dann die Direktive `Screen` zur Geltung, die die Nummer des verwendeten Kartenausgangs enthält. Sollten zwei Monitorausgänge möglich sein, so ist für jeden Ausgang ein Device-Abschnitt einzurichten. Die restlichen Einträge aus dem unteren Beispiel erklären sich von selbst.

```
Section "Device"
    Identifier    "Device[0]"
    VendorName    "NVidia"
    BoardName     "RIVA TNT2"
```

```

BusID      "1:5:0"
Driver     "nv"
Screen     0
Option     "Rotate" "off"
EndSection

```

## Screen

Der Abschnitt **Screen** verknüpft nun Grafikkarten- und Monitoreinstellungen miteinander. Auch dieser Abschnitt kann mehrfach vorkommen. Mit der Direktive **DefaultDepth** wird die Standardfarbtiefe in Bits pro Pixel angegeben. In den Unterabschnitten, die mit dem Schlüsselwort **SubSection** eingeleitet und mit dem Schlüsselwort **EndSubSection** beendet werden, wird der Zusammenhang zwischen Farbtiefe und den dafür möglichen Auflösungen definiert. Zwischen diesen Auflösungen kann durch die Tastenkombinationen **<Strg>+<Alt>+<Nummernfeld-Plus>** und **<Strg>+<Alt>+<Nummernfeld-Minus>** gewechselt werden.

```

Section "Screen"
    Identifier   "Screen[0]"
    Device       "Device[0]"
    Monitor      "Monitor[0]"
    DefaultDepth 16
    SubSection "Display"
        Depth     16
        Modes      "1152x864" "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth     24
        Modes      "1280x1024" "1152x864" "1024x768" "800x600" "640x480"
    EndSubSection
EndSection

```

## ServerLayout

Im letzten Abschnitt **ServerLayout** werden nun die verwendeten Eingabegeräte (**InputDevice**) und Ausgabegeräte (**Screen**) miteinander verknüpft. Beim Einsatz von mehreren Bildschirmen wird hier auch die physikalische Anordnung der Bildschirme festgehalten.

```

Section "ServerLayout"
    Identifier   "Layout[all]"
    Screen       "Screen[0]"
    InputDevice  "Keyboard[0]" "CoreKeyboard"
    InputDevice  "Mouse[1]" "CorePointer"
    Option       "Clone" "off"
    Option       "Xinerama" "off"
EndSection

```

### 19.2.3 Konfigurationsprogramme

Die manuelle Konfiguration eines X-Servers ist mühselig und ziemlich fehleranfällig. Daher wurden schon bald komfortablere Werkzeuge für die Konfiguration der Datei **XF86Config** entwickelt.

**xf86config** Das wohl älteste Programm zur Konfiguration des X-Servers ist **xf86config**. Das rein textbasierende Programm stammt direkt von der XFree86-Gruppe. Für die Bedienung sind gute Kenntnisse der Gerätespezifizierung ausgesprochen nützlich. Es ist kein Tool für Anfänger, aber eigentlich bei jeder Distribution dabei.



**XF86Setup** Wesentlich komfortabler ist das ebenfalls von der XFree86-Gruppe stammende **XF86Setup**. Es besitzt einen graphischen Modus, der eine einfachere Konfiguration ermöglicht. Allerdings ist **XF86Setup** nicht mehr überall vorhanden, da es von den distributionseigenen Konfigurationstools verdrängt wird.

**xvidtune** Dieses Tool ist für die Konfiguration der XFree86-Videomodus-Erweiterung (engl. *XFree86-VidModExtension*) zuständig. Diese Erweiterung des X-Window-Systems ermöglicht es die Lage und Größe des angezeigten Bildes unabhängig von den entsprechenden hardwareseitigen Monitoreinstellungen zu verschieben.

**SaX und SaX2** Das von SuSE entwickelte **SaX** und sein Nachfolger **SaX2** sind sehr komfortable Konfigurationswerkzeuge für X11. Die beiden Programme sind in der Lage die Hardware selbständig zu erkennen und zu konfigurieren. Viele Sachen wie Grafikkarte, Touchscreens, Zeichentablets, 3D-Funktionen, Zugriffsrechte, Tastaturen, Mäuse, Mehrbildschirmssysteme können durch dieses Werkzeug konfiguriert werden.

## 19.2.4 xset

Der Befehl **xset** erlaubt es die Benutzereinstellungen des laufenden X-Servers zu ändern.

**xset** [-display HOST:DISPLAYNR] [OPTIONEN]

Beim Starten des X-Servers können Sie Parameter mitgeben, die Einstellungen in bestimmten Abschnitten der **XF86Config** überschreiben können. Diese Einstellungen können Sie im laufenden Betrieb mit dem Befehl **xset** verändern. Im Gegensatz zu normalen Shellkommandos gibt das führende Minuszeichen an, ob die Option eingeschaltet oder ausgeschaltet wird. Geben Sie den Befehl ohne Schalter ein, so erscheint die Hilfe.

```
usage: xset [-display host:dpv] option ...
To turn bell off:                -b                      b off                b 0
To set bell volume, pitch and duration:  b [vol [pitch [dur]]]    b on
To disable bug compatibility mode:  -bc
To enable bug compatibility mode:    bc
To turn keyclick off:              -c                      c off                c 0
To set keyclick volume:            c [0-100]          c on
To control Energy Star (DPMS) features: -dpms          Energy Star features off
+dpms          Energy Star features on
dpms [standby [suspend [off]]]
force standby
force suspend
force off
force on
(also implicitly enables DPMS features)
a timeout value of zero disables the mode

To control font cache:            fc [hi-mark [low-mark [balance]]]
                                  both mark values specified in KB
                                  balance value specified in percent (10 - 90)

Show font cache statistics:        fc s
To set the font path:              fp= path[,path...]
To restore the default font path:  fp default
To have the server reread font databases: fp rehash
To remove elements from font path: -fp path[,path...]  fp- path[,path...]
To prepend or append elements to font path: +fp path[,path...]  fp+ path[,path...]
To set LED states off or on:       -led [1-32]          led off
                                  led [1-32]          led on

To set mouse acceleration and threshold: m [acc_mult[/acc_div] [thr]]    m default
To set pixel colors:               p pixel_value color_name
To turn auto-repeat off or on:     -r [keycode]          r off
                                  r [keycode]          r on
                                  r rate [delay [rate]]

For screen-saver control:          s [timeout [cycle]]  s default    s on
                                  s blank                s noblank    s off
                                  s expose                s noexpose
                                  s activate              s reset

For status information:            q
```

## Beispiele

Informationen über die aktuellen Einstellungen erhalten Sie über den Schalter `-q`.

```
ole@enterprise:~> xset -q
Keyboard Control:
  auto repeat:  on    key click percent:  0    LED mask:  00000002
  auto repeat delay: 660    repeat rate:  25
  auto repeating keys: 00ffffffdffffbbf
                        fadffdfdfdfef
                        ffffffff
                        ffffffff
  bell percent:  50    bell pitch:  400    bell duration:  100
Pointer Control:
  acceleration:  2/1    threshold:  4
Screen Saver:
  prefer blanking: yes    allow exposures:  yes
  timeout:  0    cycle:  600
Colors:
  default colormap:  0x20    BlackPixel:  0    WhitePixel:  65535
Font Path:
  /usr/X11R6/lib/X11/fonts/misc:unscaled,/usr/X11R6/lib/X11/fonts/75dpi:unscaled,/usr/X11R6/lib/X11/fonts/Type1,/usr/X11R6/lib/
Bug Mode: compatibility mode is disabled
DPMS (Energy Star):
  Standby: 600    Suspend: 1200    Off: 1800
  DPMS is Enabled
  Monitor is On
Font cache:
  hi-mark (KB): 5120    low-mark (KB): 3840    balance (%): 70
File paths:
  Config file:  /etc/X11/XF86Config
  Modules path: /usr/X11R6/lib/modules
  Log file:     /var/log/XFree86.0.log
```

Auch das Powermanagement für Monitore ist über den Befehl `xset` steuerbar. Allerdings wird das Energy Star Powermanagement nicht von jeder Grafikkarten-Monitor-Kombination unterstützt. Der folgende Befehl versetzt den Monitor in den Standbymodus.

```
ole@enterprise:~> xset dpms force standby
```

Die Zeiten, nach denen das automatisch erfolgt, können Sie in drei Stufen (Standby, Suspend und Off) in Sekunden angeben.

```
ole@enterprise:~> xset dpms 300 600 900
```

Die folgenden zwei Befehle schalten bei vielen Tastaturen die LED für Rollen ein und wieder aus.

```
ole@enterprise:~> xset led 3
ole@enterprise:~> xset -led 3
```

## 19.3 Schriften

Eine wichtige Aufgabe des X-Servers ist die Darstellung von Text. Dafür benötigt der Server bestimmte *Schriften* (engl. *Fonts*). Dabei werden Schriftformate wie z. B. die normalen X11-Schriften, Speedo, Type1 (Postscript) und TrueType. Die dazugehörigen Dateien können im lokalen Dateisystem liegen oder über einen Fontserver aus dem Netzwerk bezogen werden.

### 19.3.1 Lokale Schriften

Um Schriften für X11 verwenden zu können, müssen Sie als erstes passende Verzeichnisse anlegen. Normalerweise liegen die Schriftenverzeichnisse im Verzeichnis `/usr/X11R6/lib/X11/fonts`. Die Verzeichnisse müssen Sie dann in der Konfigurationsdatei `XF86Config` im Abschnitt `Files` mit der Direktive `FontPath` eintragen.

Pro Verzeichnis machen Sie normalerweise einen Eintrag. Achten Sie dabei auf die Reihenfolge der Einträge, da die erste passende Schrift ausgewählt wird.

```
Section "Files"
    FontPath      "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
    FontPath      "/usr/X11R6/lib/X11/fonts/Type1"
    FontPath      "/usr/X11R6/lib/X11/fonts/truetype"
    ...
EndSection
```

In diese Verzeichnisse werden dann die Schriftdateien hineinkopiert. Neben den Verzeichnissen in der `XF86Config` können Sie auch temporär dem X-Server Schriftverzeichnisse zuordnen. Sie können dies durch den Befehl `xset` erledigen. Der folgende Befehl meldet ein Verzeichnis beim X-Server temporär an.

```
xset +fp /usr/X11R6/lib/X11/fonts/ttextra
```

Wollen Sie ein Verzeichnis temporär abmelden, dann kommt ein Minus vor die Option `fp`.

```
xset -fp /usr/X11R6/lib/X11/fonts/ttextra
```

### Die Liste: `fonts.dir`

Mit dem Erstellen eines Verzeichnisses, dem Kopieren der Schriftdateien und dem Anmelden des Verzeichnisses beim X-Server ist es noch nicht getan. Jedes Schriftverzeichnis benötigt die Verwaltungsdatei `fonts.dir`. Diese Datei enthält eine Liste aller Schriften im Verzeichnis. In der ersten Zeile wird die Anzahl der zur Verfügung stehenden Schriften aufgeführt. Dabei gelten auch Varianten der Schriften (Neigung, Gewicht, Kodierung) auch jeweils als eine Schrift. Die hier aufgeführte Zahl entspricht der Anzahl der Zeilen minus 1. Daher repräsentiert jede der folgenden Zeilen eine Schrift bzw. Schriftvariante. Als erstes steht in jeder Zeile die zuständige Schriftdatei. Nach einem Leerzeichen folgt dann die sogenannte *X Logical Font Description* (*XLFD*). Der XLFD-String besteht aus mehreren, von Minus-Zeichen getrennten Feldern. Dabei haben die Felder von links nach rechts folgende Bedeutung:

**Hersteller** Auch die Hersteller wie Adobe oder Bitstream möchten erwähnt werden.

**Name** Der Name der Schriftfamilie, wie er auch bei der Fontauswahl in den X-Programmen auftaucht.

**Gewicht** Muß etwa auch eine Schrift auf ihr Gewicht achten? In der Fachsprache bezeichnet das Gewicht die Dicke der Schriftlinien. Hier sind Angaben wie z. B. `thin`, `light`, `medium`, `bold` oder `black` möglich.

**Neigung** Die meisten Schriften stehen normalerweise aufrecht (o). Es gibt auch sogenannte kursive Schriften, die geneigt sind. Dabei wird noch zwischen den nach rechts und links geneigten Italic (i, ri) und Oblique (o, ro) unterschieden. Die meisten Schriftfamilien besitzen allerdings keine extra Schrift für Oblique.

**Breite** Innerhalb einer Schriftfamilie gibt es auch Variationen in der Breite. Hier wird zwischen einer normalen Breite (`normal`), einer engen Schrift (`condensed`) und einer weiten Schrift (`expanded`) unterschieden.

**Stil** Hier ist eine zusätzliche Angabe zum Stil der Schrift wie z. B. `serif`, `sans serif`, `informal` und `decorated` möglich. Allerdings ist in den meisten Fällen dieses Feld leer.

**Pixel** Die Höhe des Zeichens in Pixel. Die Größe ist damit abhängig von der Auflösung des Darstellungsmediums.

**Punkte** Die Höhe des Zeichens in Punkten. Die Größe sollte damit unabhängig von der Auflösung des Darstellungsmediums sein. Idealerweise entspricht die Höhe dann auch der Angabe in Punkten. Dabei entspricht eine Größe von 72 Punkten genau einem Zoll.

**X** Horizontale Auflösung in dpi.

**Y** Vertikale Auflösung in dpi.

**Zeichenbreite** Gibt an, ob es sich um eine Proportional- (p) oder um eine Festbreitenschrift (m) handelt.

**Durchschnittliche Breite** Dieses Feld gibt die zehnfache durchschnittliche Breite des Zeichens in Pixel an.

**Registrierung** Name der Zeichensatzgruppe wie z. B. iso8859, iso10646 oder adobe.

**Kodierung** Der Rest des Zeichensatznamens wie z. B. 1, 9 oder symbol.

Schauen wir doch mal in eine font.dir-Datei hinein.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> head truetype/fonts.dir
2963
dustismo_bold_italic.ttf -adobe-Dustismo-bold-i-normal--0-0-0-0-p-0-iso10646-1
dustismo_bold_italic.ttf -adobe-Dustismo-bold-o-normal--0-0-0-0-p-0-iso10646-1
dustismo_bold.ttf -adobe-Dustismo-bold-r-normal--0-0-0-0-p-0-iso10646-1
dustismo_italic.ttf -adobe-Dustismo-medium-i-normal--0-0-0-0-p-0-iso10646-1
dustismo_italic.ttf -adobe-Dustismo-medium-o-normal--0-0-0-0-p-0-iso10646-1
Dustismo.ttf -adobe-Dustismo-medium-r-normal--0-0-0-0-p-0-iso10646-1
civitype.ttf -alltype-Civitype-thin-r-normal--0-0-0-0-p-0-ibm-cp437
civitype.ttf -alltype-Civitype-thin-r-normal--0-0-0-0-p-0-iso10646-1
civitype.ttf -alltype-Civitype-thin-r-normal--0-0-0-0-p-0-iso8859-1
```

In der ersten Zeile steht die Anzahl der Schriften, die durch dieses Verzeichnis dem X-Server zur Verfügung stehen. Die Anzahl der Einträge in der Datei stimmen jedenfalls mit der Zahl überein.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> wc -l truetype/fonts.dir
2964 truetype/fonts.dir
```

Denken Sie daran eine Zeile abzuziehen, da die erste Zeile ja die Anzahl der Schriften enthält. Wenn wir uns aber mal die Anzahl der Dateien im Verzeichnis anschauen, stoßen wir auf eine leichte Differenz in der Größenordnung des Faktors zwei.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> ls truetype/ | wc -l
1271
```

Schauen wir uns doch mal dazu die Schriftart *Bullpen* an. Für diese Schrift gibt es in der Datei fonts.dir mehrere Einträge.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> grep Bullpen truetype/fonts.dir
bullpen3.ttf -larabiefonts-Bullpen 3D-medium-r-normal--0-0-0-0-p-0-iso10646-1
bullpen3.ttf -larabiefonts-Bullpen 3D-medium-r-normal--0-0-0-0-p-0-iso8859-1
bullpen3.ttf -larabiefonts-Bullpen 3D-medium-r-normal--0-0-0-0-p-0-iso8859-13
bullpen3.ttf -larabiefonts-Bullpen 3D-medium-r-normal--0-0-0-0-p-0-iso8859-2
bullpen3.ttf -larabiefonts-Bullpen 3D-medium-r-normal--0-0-0-0-p-0-iso8859-9
bullpeni.ttf -larabiefonts-Bullpen-medium-i-normal--0-0-0-0-p-0-iso10646-1
bullpeni.ttf -larabiefonts-Bullpen-medium-i-normal--0-0-0-0-p-0-iso8859-1
bullpeni.ttf -larabiefonts-Bullpen-medium-i-normal--0-0-0-0-p-0-iso8859-13
bullpeni.ttf -larabiefonts-Bullpen-medium-i-normal--0-0-0-0-p-0-iso8859-2
bullpeni.ttf -larabiefonts-Bullpen-medium-i-normal--0-0-0-0-p-0-iso8859-9
bullpeni.ttf -larabiefonts-Bullpen-medium-o-normal--0-0-0-0-p-0-iso10646-1
bullpeni.ttf -larabiefonts-Bullpen-medium-o-normal--0-0-0-0-p-0-iso8859-1
bullpeni.ttf -larabiefonts-Bullpen-medium-o-normal--0-0-0-0-p-0-iso8859-13
bullpeni.ttf -larabiefonts-Bullpen-medium-o-normal--0-0-0-0-p-0-iso8859-2
bullpeni.ttf -larabiefonts-Bullpen-medium-o-normal--0-0-0-0-p-0-iso8859-9
bullpen_.ttf -larabiefonts-Bullpen-medium-r-normal--0-0-0-0-p-0-iso10646-1
```

```
bullpen_.ttf -larabiefonts-Bullpen-medium-r-normal--0-0-0-0-p-0-iso8859-1
bullpen_.ttf -larabiefonts-Bullpen-medium-r-normal--0-0-0-0-p-0-iso8859-13
bullpen_.ttf -larabiefonts-Bullpen-medium-r-normal--0-0-0-0-p-0-iso8859-2
bullpen_.ttf -larabiefonts-Bullpen-medium-r-normal--0-0-0-0-p-0-iso8859-9
```

Wenn Sie die Einträge genauer betrachten, werden Sie sehen, daß sich die 20 Einträge in der Datei nur auf 3 Schriftdateien beziehen. Dies liegt zu einem daran, daß die Schriftneigungen Italic (i) und Oblique (o) oft durch gleiche Dateien abgedeckt werden. Auch können manche Schriften mehrere Zeichensätze gleichzeitig abdecken. In unserem Beispiel werden durch die Bullpen-Schriftdateien die Zeichensätze ISO 8859-1, ISO 8859-2, ISO 8859-9, ISO 8859-13 und ISO 10646-1 abgedeckt. Wenn Sie sich dann mal die "Bullpen-Dateien" im Schriftverzeichnis anzeigen lassen, werden Sie neben den TrueType-Schriftdateien (ttf) weitere Dateien entdecken.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> ls -l truetype/bullpen*
-rw-r--r-- 1 root root 15711 2004-05-09 13:34 truetype/bullpen3.afm
-rw-r--r-- 1 root root 118288 2001-03-14 22:45 truetype/bullpen3.ttf
-rw-r--r-- 1 root root 15177 2004-05-09 13:34 truetype/bullpen_.afm
-rw-r--r-- 1 root root 15281 2004-05-09 13:34 truetype/bullpeni.afm
-rw-r--r-- 1 root root 44936 2001-03-14 22:46 truetype/bullpeni.ttf
-rw-r--r-- 1 root root 48176 2001-03-14 22:28 truetype/bullpen_.ttf
```

Die Dateien mit der Endung `afm`, wobei `afm` für Adobe Font Metrics steht, enthalten ausführliche Informationen über Postscript-Schriften. Die Programme OpenOffice und StarOffice benötigen diese Dateien um mit den Schriften umzugehen. Für X11 haben Sie ansonsten keine weitere Bedeutung und dürfen daher nicht in die Zählung der Dateien mit aufgenommen werden.

### mkfontdir

Sie brauchen die Datei `fonts.dir` nicht selbst zu erstellen. Behilflich dabei ist Ihnen der Befehl `mkfontdir`.

```
mkfontdir [OPTIONEN] [VERZEICHNIS]
```

Dabei durchsucht `mkfontdir` das aktuelle oder das angegebene Verzeichnis nach Schriftdateien und wertet dieses aus. Aus diesen Informationen erzeugt das Programm dann die Datei `fonts.dir`. Allerdings kann es gerade bei skalierbaren Schriften vorkommen, daß die notwendigen Informationen nicht aus der Datei extrahiert werden können. Für solche Schriften existiert eine Datei `fonts.scale` im gleichen Verzeichnis. Diese Datei ist vom Aufbau identisch mit der Datei `fonts.dir` und wird vom Befehl `mkfontdir` als weitere Informationsquelle benutzt.

### mkfontscale

Auch die Datei `fonts.scale` brauchen Sie nicht selber zu erstellen. Seit XFree86 Version 4.3.0 gibt es den Befehl `mkfontscale`. Dieser ermittelt aus den Dateien skalierbarer Schriften die fürs X11 wichtigen Werte und schreibt sie in die Datei `fonts.scale`.

```
mkfontscale [OPTIONEN] [VERZEICHNIS]
```

### xfontsel

Dieses kleine Programm ermöglicht es Ihnen auf einfache Art die installierten Schriften zu kontrollieren und sie sich anzeigen zu lassen.

```
xfontsel [OPTIONEN]
```

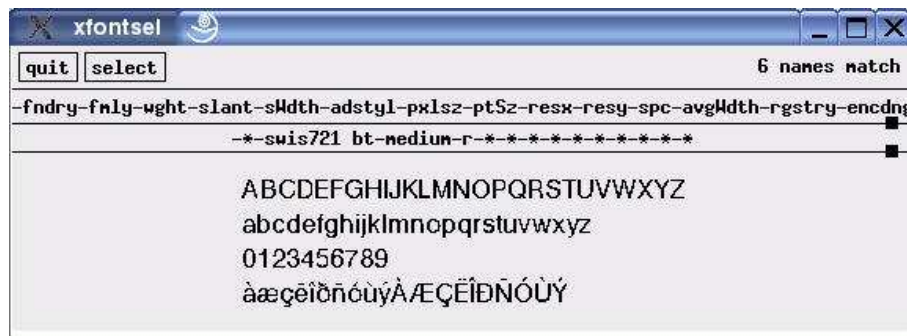


Abbildung 19.2: Die Ansicht von Schriften ermöglicht das Programm xfontsel.

Dabei bedient es sich zur Auswahl und Beschreibung der Schrift der XLFD und zeigt damit nicht nur ein Beispiel, wie die Schrift aussieht. Sondern zeigt auch an, wieviele Einträge für ein XLFD-Muster vorhanden sind.

Ein Screenshot des Programm sehen Sie in Abbildung 19.2 abgebildet.

#### fonts.alias

Es ist möglich bestimmten Schriften eine zweiten Namen bzw. XLFD zu geben. Hierzu muß im Verzeichnis die Datei `fonts.alias` angelegt werden. Die Datei besteht aus zwei Spalten. In der ersten Spalte steht der neue Name und in der zweiten Spalte der passende XLFD-String. Die Vergabe von Alias-Namen für Schriften ermöglicht es z. B. lästige Fehlermeldungen von X-Programmen abzuschalten, die nicht vorhandene Schriften benötigen. Genauso können Sie von Terminals angeforderte Schriften auf ihre Wunschschrift umändern. Eine Alias-Datei könnte wie das folgende Beispiel aussehen.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> cat truetype/fonts.alias
! Meine Alias-Datei für Schriften
"-0les-Liebling-bold-i-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-bold-i-normal--0-0-0-0-p-0-iso8859-1"
"-0les-Liebling-bold-o-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-bold-o-normal--0-0-0-0-p-0-iso8859-1"
"-0les-Liebling-bold-r-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-bold-r-normal--0-0-0-0-p-0-iso8859-1"
"-0les-Liebling-medium-i-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-medium-i-normal--0-0-0-0-p-0-iso8859-1"
"-0les-Liebling-medium-o-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-medium-o-normal--0-0-0-0-p-0-iso8859-1"
"-0les-Liebling-medium-r-normal--0-0-0-0-p-0-iso8859-1" "-misc-Swis721 BT-medium-r-normal--0-0-0-0-p-0-iso8859-1"
```

Ändern Sie die Datei `fonts.alias` im laufenden Betrieb oder führen Sie sonstige Änderungen an den Schriften durch, so ist es angebracht als X-Window-Benutzer das Neuladen der Schriftkonfiguration zu veranlassen. Sie können dafür in einem X-Terminal den folgenden Befehl verwenden.

```
ole@enterprise:/usr/X11R6/lib/X11/fonts> xset fp rehash
```

### 19.3.2 Der Fontserver xfs

Der andere Weg einem X-Window-System Schriften zur Verfügung zu stellen ist der Fontserver `xfs`. Er ermöglicht eine zentrale Verwaltung und Verteilung des X11-Schriften im Netz. Wegen der heutigen Plattengröße wird er nur noch selten benutzt. Allerdings ermöglicht er den Zugriff auf Schriften von einer zentralen Stelle aus und erleichtert damit erheblich die Verwaltung der Schriften. Der Fontserver `xfs` ist ein freistehender Dämon und arbeitet auf dem TCP-Port 7100. Starten können Sie Fontserver direkt über den Programmnamen

```
enterprise:~ # xfs &
```

oder über das RC-Skript.

```
enterprise:~ # rcxfs start
```

## Server-Konfiguration

Zentraler Punkt der Konfiguration ist die Konfigurationsdatei `/etc/X11/fs/config`. Möglich ist auch der Name `xfs.config` im gleichen Verzeichnis. Früher lag war die Konfigurationsdatei `/usr/X11R6/lib/X11/fs/config`, dies ist aber eine veraltete Form der Konfiguration. Der Fontserver greift für seine Tätigkeit auf die lokal installierten Schriften zurück, die natürlich wie im oberen Abschnitt erläutert wurde, ordnungsgemäß installiert sein müssen.

So könnte eine Fontserver-Konfigurationsdatei aussehen.

```
no-listen = tcp
port = 7100
client-limit = 10
clone-self = on
use-syslog = on

catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/Type1,
            /usr/X11R6/lib/X11/fonts/Speedo,
            /usr/X11R6/lib/X11/fonts/CID,
            /usr/X11R6/lib/X11/fonts/kwintv,
            /usr/X11R6/lib/X11/fonts/truetype

# in decipoints
default-point-size = 120
default-resolutions = 75,75,100,100
```

Sollten Sie Änderungen an dieser Konfigurationsdatei durchführen, dann müssen Sie den Fontserver neu initialisieren. Dies können Sie z. B. durch den folgenden Befehl durch das Signal SIGHUP ausführen lassen.

```
enterprise:~ # killall -1 xfs
```

Natürlich können Sie auch das passende RC-Script benutzen.

```
enterprise:~ # rcxfs reload.
```

Was bedeuten nun aber die Direktiven im Einzelnen.

**catalogue** Dies ist wohl wichtigste Direktive der Konfigurationsdatei. Hiermit geben Sie durch Kommata getrennt die Schriftverzeichnisse an, die exportiert werden sollen.

**client-limit** Hiermit können Sie die Zahl der Clients begrenzen, die auf den X-Server zugreifen können.

**clone-self** Sollte den Grenzwert von `client-limit` erreicht werden, so können Sie mit dieser Direktive entscheiden, ob neue xfs-Instanzen gestartet werden sollen.

**default-point-size** Hiermit legen Sie die Standardgröße der Schrift in Punkten fest, wenn die Schrift selber keine Informationen darüber enthält. (Standard 120)

**default-resolutions** Auch die Standardauflösung der Schriften kann der Fontserver angeben, wenn keine Informationen in der Schriftdatei darüber vorhanden sind.

**no-listen** Deaktiviert einen Transporttyp. Wenn Sie hier TCP angeben, dann funktioniert der Server nicht mehr über das Protokoll TCP/IP.

**port** Hier können Sie die Portnummer für den Dienst festlegen.

**error-log** Hiermit können Sie ein Datei angeben, in der die Fehlermeldungen ausgegeben werden sollen.

**use-syslog** Allerdings können Sie auch den normalen syslog-Dämon für diese Aufgabe verwenden.

## Client-Konfiguration

Die Client-Konfiguration ist noch viel einfacher. Sie müssen den Fontserver einfach wie ein Schriftenverzeichnis angeben. Dazu gehen Sie in die Konfigurationsdatei des X-Servers `/etc/X11/XF86Config` und tragen dort im Abschnitt "Files" einen `FontPath`-Eintrag ein. Wenn wir davon ausgehen, daß der Fontserver den Namen "DEFIANT" trägt und ganz normal über den TCP-Port 7100 zu erreichen ist, muß der Eintrag wie folgt lauten.

```
FontPath      "tcp/DEFIANT:7100"
```

Sind Sie allerdings nur an einem Verzeichnis vom Server (z. B. `extrattf`) interessiert, dann lautet der Eintrag

```
FontPath      "tcp/DEFIANT:7100/extrattf"
```

## 19.4 Starten der graphischen Oberfläche ...

Wenn der X-Server gestartet wird, sucht er sich die nächste freie Konsole zur Darstellung der Grafik. Als Konvention hat sich die Konsole `/dev/tty7` durchgesetzt. Beginnt der X-Server mit der Darstellung, so wechselt er automatisch von der aktiven Konsole zu seiner Grafikkonsole. Aus der X-Konsole kann nun mit der Tastenkombination `<Strg>+<Alt>+<FN>` gewechselt werden, wobei *N* die Nummer der Konsole ist.

Jeder graphische Bildschirm bekommt einen eindeutigen *Displaynamen*, der sich aus dem Rechnernamen, der Nummer des X-Servers und der Bildschirmnummer zusammensetzt.

***HOST:SERVERNUMMER.BILDSCHIRM***

Passende Beispiele für den Displaynamen sind also:

**defiant.amov.de:0** Erster Server und erster Bildschirm auf dem Rechner mit dem DNS-Namen `defiant.amov.de`

**217.89.70.60:1** Zweiter Server und erster Bildschirm auf dem Rechner mit der IP-Nummer 217.89.70.60

**localhost:0.1** Erster Server und zweiter Bildschirm auf dem lokalen Rechner

**:0** Erster Server und erster Bildschirm auf dem lokalen Rechner.

Die Bildschirmnummer ist nur notwendig, wenn mehrere Bildschirme im Einsatz sind. Der Einsatz von mehreren Bildschirmen wird z. B. durch die *Xinerama*-Erweiterung von `XFree86` ermöglicht. Auch der Rechnername ist nur notwendig, wenn auf einen entfernten Rechner zugegriffen wird. Bei Ansprache des lokalen Rechners ist die Angabe des Rechnernamens nicht notwendig.

Der Displayname ist deswegen so wichtig, weil die X-Clients, die normale Linux-Programme sind, ja wissen müssen, auf welchem graphischen Bildschirm bzw. X-Server ihre Ausgabe landen soll. Die meisten X-Clients besitzen die Option `-display`, mit der der Displayname angegeben werden kann, wie im folgenden Beispiel zu sehen.

```
ole@enterprise:~> xterm -display :0
```

Es wäre aber etwas mühsam den Wert bei jedem Programmaufruf durch eine Option mitliefern zu lassen. Deshalb orientieren sich die meisten X-Clients an der Umgebungsvariable `DISPLAY`. Im Normalfall ist die Variable richtig gesetzt. Sollten Sie aber zum Beispiel den X-Client mit dem X-Server auf einem entfernten Rechner kommunizieren lassen, dann muß die Variable in der Shell vorher wie im folgenden Beispiel geändert werden.

```
ole@enterprise:~> export DISPLAY=defiant.amov.de:0
```

Aber nun wollen wir uns mal anschauen, wie Sie den X-Server starten können.



### 19.4.1 ... direkt über den X-Server

Der direkte Weg geht natürlich über den X-Server. Der Name des Programms ist denkbar knapp. Es heißt einfach **X**. Bei SuSE 9.0 ist es etwas kompliziert. Das Programm **X** liegt im Verzeichnis `/usr/X11R6/bin/` und ist ein symbolischer Link auf `/var/X11R6/bin/X`. Aber das ist noch nicht das Ziel, denn dies ist ein symbolischer Link auf `/usr/X11R6/bin/XFree86`. Nicht aufregen, nur wundern. ;-)

Jedenfalls können Sie mit dem Befehl **X** den X-Server starten. Sie können ohne Probleme auch zwei oder mehr X-Server parallel fahren lassen. Vorausgesetzt natürlich Sie haben genug Speicher in ihrem Rechner. Schauen Sie sich doch mal folgendes Beispiel an.

```
ole@enterprise:~> X :1 & xterm -display :1
```

Die Befehlskette startet einen X-Server mit der Servernummer 1. Der Prozeß wird dabei in den Hintergrund gelegt, damit noch ein zweiter Befehl ausgeführt werden kann. Ansonsten würden Sie nur einen leeren, grau gemusterten Bildschirm mit einem Kreuz als Mauszeiger sehen. Da der X-Server in den Hintergrund geht, wird der folgende Befehl gleich mit ausgeführt. Ein **xterm** wird gestartet und mit der Option `-display` mit dem zweiten X-Server verbunden.

Noch ist das Bild, was jetzt automatisch auftaucht, etwas mager. Viel kann man mit dem X-Window nicht anfangen. Was fehlt ist der Fenstermanager. Sie können ganz einfach einen starten, wenn Sie in dem **xterm** folgenden Befehl eingeben.

```
ole@enterprise:~> twm &
```

Auch dieser Befehl wird in den Hintergrund geschoben um wieder Platz am Prompt zu schaffen. Aber was machen wir nun? Wie können wir den ganzen Kram beenden? Am einfachsten betätigen Sie jetzt die Tastenkombination `<Strg>+<Alt>+<←>`, wenn Sie nicht in der X-Konfigurationsdatei deaktiviert wurde. Der Server beendet sich nun. Er beendet auch das mit ihm verbundene **xterm** und damit auch den Fenstermanager.

Sie können natürlich auch zur Konsole zurückkehren, wo Sie den Server gestartet haben. Die Tastenkombination `<Strg>+<c>` beendet das **xterm**. Nun können Sie sich mit dem Befehl **jobs** (12.9.4) die Jobnummer des X-Servers ausgeben lassen und ihn dann mit dem Befehl **kill** (12.9.7) beenden.

```
ole@enterprise:~> jobs
[1]+  Running                  X :1 &
ole@enterprise:~> kill %1
[1]+  Done                     X :1
```

Es geht auch noch hübscher ...

```
ole@enterprise:~> X -br :1 & xterm -display :1 & twm -display :1 &
```

Wenn Sie mehr über **X** wissen wollen, konsultieren Sie die Manualpage zu **X(7)** oder rufen den Befehl mit der Option `--help` auf.

### 19.4.2 ... mittels des Duos startx und xinit

Auf jeden Fall komfortabler geht es mit dem Skript **startx**. Das Shellskript ist ein Frontend für das Programm **xinit**. Es erledigt erst einige Initialisierung und startet dann das Programm **xinit**, wobei die an **startx** angehängten Parameter weitergereicht werden.

Das Programm **xinit** benötigt Informationen über den zu startenden X-Server und die zu startenden X-Clients. Daher sucht es zuerst im Heimatverzeichnis des Benutzers nach den Dateien `.xinitrc` und `.xserverrc`. Sollten diese nicht vorhanden sein, dann sucht er nach den globalen Konfigurationsdateien `/etc/X11/xinit/xinitrc` und `/etc/X11/xinit/xserverrc`.

Die Datei `.xserverrc` ist verantwortlich für den Start des X-Servers, falls keiner beim Starten von `xinit` angegeben worden ist. Die Datei `.xinitrc` definiert dagegen die X-Clients, die automatisch gestartet werden sollen. Im Prinzip ist die Datei ein Shellskript. Sie müssen nur darauf achten, daß alle X-Clients im Hintergrund gestartet werden. Der Fenstermanager wird als letzter Dienst im Vordergrund gestartet. Das Programm `init` stellt nämlich noch eine besondere Funktion bereit. Wird der erste im Vordergrund gestartete X-Client beendet, so wird auch der X-Server beendet.

Als letztes noch ein Beispiel, wie ein Benutzer einen zweiten X-Server starten kann. Das zu startende `xterm` und der zu startende Fenstermanager werden in die Datei `~/.xinitrc` eingetragen.

```
walter@enterprise:~> cat .xinitrc
xterm &
twm
walter@enterprise:~> startx -- :1
```

Sie sehen, der Vorgang unter Verwendung der Datei `.xinitrc` ist wesentlich einfach und komfortabler als der Start im vorherigen Abschnitt. Das doppelte Minuszeichen hat hier eine besondere Bedeutung. Alle Parameter links davon beziehen sich auf die X-Clients, alle Parameter rechts davon auf den X-Server.

### 19.4.3 ... über einen Displaymanager

Heute bei den modernen Workstations ist es üblich, daß das ganze System gleich mit dem X-Window-System startet. Für den Start sorgt, wie üblich bei automatisch startenden Servern, ein Init-Skript. Nach dem Standard LSB wird das X-Window-System und ein Displaymanager im Runlevel 5 gestartet. Dies ist auch der einzige Unterschied zwischen den beiden Runleveln. Den Default-Runlevel können Sie in der Datei `/etc/inittab` festlegen. Für Runlevel 3 sieht die entsprechende Zeile wie folgt aus.

```
id:3:initdefault:
```

Nach dem Starten des X11 wird der Displaymanager gestartet. Dieser zeigt ein Fenster zum Einloggen und startet dann die eigentliche Benutzersitzung. Es gibt eine Reihe von Displaymanagern. Hier mal eine kleine Übersicht.

#### **XFree86:** `xdm`

Dies ist der Standarddisplaymanager des XFree86-Projekts. Er bietet nur ein relativ simples graphisches Login-Fenster. Seine Konfigurationsdateien befinden sich im Verzeichnis `/etc/X11/xdm`.

Die Datei `xresources` enthält die Konfiguration des Displaymanagers. Hier können Sie z. B. mit der Direktive `xlogin*greeting` den Begrüßungstext festlegen, mit der Direktive `xlogin*login.greetFont` die dafür passenden Zeichensatz aussuchen oder sogar mit der Direktive `xlogin*logoFileName` ein Logo einblenden.

Dagegen ist die Datei `xsetup` ein Skript, das beim Starten von `xdm` abgearbeitet wird. Mit ihm ist es z. B. möglich ein Hintergrundbild einzubauen.

Welcher X-Server auf welchem Display erscheinen soll, legt die Datei `Xservers` fest. Dies ist vor allem dann notwendig, wenn mehrere Displays zur Verfügung stehen.

Die Datei `xsession` dient zur Initialisierung der Benutzersitzung. Jeder Benutzer kann sich eine persönliche Variante dieser Datei in seinem Heimatverzeichnis unter dem Namen `.xsession` einrichten.

#### **KDE:** `kdm`

Der KDE Displaymanager `kdm` ist eine Erweiterung des `xdm`. Seine Konfiguration entspricht der seines Vorbilds. Sie müssen allerdings nicht in den Konfigurationsdatei selbst herumbasteln, sondern können über das KDE-Kontrollzentrum (`kcontrol`) den Displaymanager konfigurieren. Die dazugehörige Konfigurationsdatei ist `/etc/X11/kdm/kdmrc` oder `/etc/opt/kde3/share/config/kdm/kdmrc` zu finden.

**Gnome: gdm**

Der Displaymanager `gdm` von Gnome ist eine komplette Neuentwicklung. Seine Konfigurationsdateien liegen entweder im Verzeichnis `/etc/X11/gdm` oder unter `/etc/opt/gnome2/gdm`. Auch kann `gdm` komfortabler verwaltet werden, indem Sie das Gnome-Konfigurationsprogramm `gdmconfig` verwenden.

## 19.5 X-Window im Netzwerk

Ein besonderes Merkmal des X-Window ist die Fähigkeit, X-Clients und X-Server auf verschiedenen Rechnern fahren zu können. Der X-Server stellt seine Dienste über den TCP-Port 6000 plus der Servernummer zur Verfügung. Der zweite X-Server (:1) ist also über Port 6001 erreichbar. Damit kann nun von jedem Rechner aus auf den X-Server zugegriffen. Sie könnten z. B. jemanden auf seinem Desktop eine Nachricht zukommen lassen. Das Programm `xmessage` läßt ein Fenster mit einem Text auf dem Desktop erscheinen. Dann müssen Sie nur die Variable `DISPLAY` auf den gewünschten X-Server einstellen und das Fenster landet auf dem entsprechenden Desktop.

```
ole@enterprise:~> export DISPLAY=defiant:0
ole@enterprise:~> xmessage "Hallo"
```

Wenn Sie selber Programme auf einem entfernten Rechner laufen lassen wollen, gehen Sie genauso vor. Im ersten Schritt loggen Sie sich über eine Fernverbindung (`telnet`, `rlogin` oder `ssh`) auf dem entfernten Rechner ein. Dann setzen Sie die Variable `DISPLAY` auf den Displaynamen Ihres Desktops und exportieren Sie. Dann können Sie auf dem entfernten Rechner das X-Programm starten und die Ausgabe und Steuerung erfolgt über den X-Server Ihres Rechners.

Vielleicht ist Ihnen schon das Problem in der Praxis aufgefallen. Ohne besonderen Schutz ist der Zugriff von außen von jedem anderen Rechner aus und von jedem anderen Benutzer möglich. Es dürfte aber nicht in Ihrem Interesse liegen, daß jemand ihre Sitzungen ausspäht oder sogar stört. Deshalb greifen Sicherungsmechanismen, die den Zugriff beschränken. Nur bestimmte Rechner oder Benutzer können Zugriffsrechte erhalten. Für die Konfiguration gibt es zwei Tools, die den Zugriff verwalten: `xhost` für rechnerbezogene und `xauth` für benutzerabhängige Zugriffe.

### 19.5.1 Zugriffsschutz über `xhost`

Der X-Server besitzt einen rechnerbasierenden Zugriffsschutz. Der Zugriffsschutz wird über das Tool `xhost` verwaltet.

```
xhost [[+~]HOST
```

Dieser Schutz ist natürlich nur rudimentär, bietet aber schon einen gewissen Grundschutz, vor allem wenn allen Hosts der Zugriff auf den X-Server verboten wurde. Die Probleme treten immer dann auf, wenn auf den erlaubten Hosts mehrere Benutzer arbeiten können. Denn Sie erlauben mit den Einstellungen jedem Benutzer des jeweiligen Rechners auf Ihren X-Server zuzugreifen. Die Verwendung des Befehls `xhost` wird Ihnen in den folgenden Beispielen erklärt.

#### Beispiele

Eine Übersicht über den Status der Zugriffsschutz und die autorisierten Clients liefert der Befehl `xhost` ohne Parameter.

```
ole@enterprise:~> xhost
access control enabled, only authorized clients can connect
```

So können Sie Rechner hinzufügen. Das Pluszeichen ist dabei optional.

```
ole@enterprise:~> xhost +10.0.0.11
10.0.0.11 being added to access control list
ole@enterprise:~> xhost 10.0.0.4
10.0.0.4 being added to access control list
ole@enterprise:~> xhost
access control enabled, only authorized clients can connect
INET:10.0.0.4
INET:10.0.0.11
```

Bei Verbindungen innerhalb eines Rechners kommt anstatt der TCP-Verbindung ein Unix-Socket zu Einsatz. Um anderen Benutzern auf dem gleichen Rechner den Zugriff auf den X-Server zu erlauben, können Sie entweder den Rechner "localhost" eintragen oder den folgenden Befehl verwenden.

```
ole@enterprise:~> xhost local:
non-network local connections being added to access control list
```

Hosts können Sie mit dem vorangestellten Minuszeichen wieder entfernen.

```
ole@enterprise:~> xhost -10.0.0.11
10.0.0.11 being removed from access control list
ole@enterprise:~> xhost
access control enabled, only authorized clients can connect
INET:10.0.0.4
```

Den Zugriffsschutz können Sie mit einem einzelnen Pluszeichen abschalten, dann kann jeder Host auf den Rechner zugreifen. Das einzelne Minuszeichen schaltet den Zugriffsschutz wieder ein.

```
ole@enterprise:~> xhost +
access control disabled, clients can connect from any host
ole@enterprise:~> xhost -
access control enabled, only authorized clients can connect
```

Der Zugriffsschutz muß nicht nur bei entfernten Rechnern verwendet werden. Auch bei diesem Beispiel ist ein Zugriffsschutz notwendig.

```
ole@enterprise:~> xhost
access control enabled, only authorized clients can connect
ole@enterprise:~> su
Password:
enterprise:/home/ole # nedit
Xlib: connection to ":0.0" refused by server
Xlib: No protocol specified
```

NEdit: Can't open display

```
enterprise:/home/ole # exit
exit
ole@enterprise:~> xhost +localhost
localhost being added to access control list
ole@enterprise:~> su
Password:
enterprise:/home/ole # nedit
```

Der Versuch als Benutzer *root* auf dem X-Server des Benutzers *ole* etwas darzustellen mißlingt. Erst als der Benutzer *ole* für seinen X-Server den allgemeinen Zugriff über den lokalen Rechner (localhost) erlaubt hat, kann *root* seinen X-Client vernünftig starten.

## 19.5.2 Zugriffsschutz über Magic Cookies und xauth

Einen Zugriffsschutz auf Benutzerebene wird durch die *Magischen Kekse* (engl. *magic cookies*) ermöglicht. Beim Starten des X-Servers wird ein *Magic Cookie* erzeugt. Dieser wird dem X-Server mitgeteilt und gleichzeitig in der Datei `.Xauthority` im Heimatverzeichnis des angemeldeten Benutzers gespeichert. Diese Datei ist natürlich nur für den Benutzer selber lesbar. So könnte eine solche Datei aussehen.

```
ole@enterprise:~> ls -l .Xauthority
-rw----- 1 ole users 375 2004-06-21 20:45 .Xauthority
ole@enterprise:~> od -a .Xauthority
0000000 nul nul nul eot nl nul nul soh nul soh 0 nul dc2 M I T
0000020 - M A G I C - C 0 0 K I E - 1 nul
0000040 dle etx < T 0 E m ' ^ g bs u [ Q stx 8
0000060 ? soh nul nul nl e n t e r p r i s e nul
0000100 soh 0 nul dc2 M I T - M A G I C - C 0
0000120 0 K I E - 1 nul dle etx < T 0 E m ' ^
0000140 g bs u [ Q stx 8 ? soh nul nul dle e n t e
0000160 r p r i s e . l o c a l nul soh 1 nul
0000200 dc2 M I T - M A G I C - C 0 0 K I
0000220 E - 1 nul dle Q etx N enq E dc2 L C N _ &
0000240 Z fs h n del nul nul nul eot nl nul nul soh nul soh 0
0000260 nul dc3 X D M - A U T H 0 R I Z A T
0000300 I 0 N - 1 nul dle , 0 sub f d 8 P I nul
0000320 Z d B Q ~ ack ht soh nul nul nl e n t e r
0000340 p r i s e nul soh 0 nul dc3 X D M - A U
0000360 T H 0 R I Z A T I 0 N - 1 nul dle ,
0000400 0 sub f d 8 P I nul Z d B Q ~ ack ht
0000417
```

### Beispiele

So können Sie vorgehen, um nur *root* den Zugriff auf Ihren X-Server zu gewähren. Erst einmal überprüfen wir, ob nicht der lokale Rechner bereits Zugriffsrechte auf den X-Server besitzt.

```
ole@enterprise:~> xhost
access control enabled, only authorized clients can connect
```

Dann extrahieren wir den passenden Schlüssel für den aktiven X-Serverbildschirm und speichern ihn in die Datei `Xschluessel`.

```
ole@enterprise:~> xauth extract Xschluessel $DISPLAY
ole@enterprise:~> ls -l Xschluessel
-rw----- 1 ole users 111 2004-06-21 23:54 Xschluessel
```

Wir wechseln unsere Identität zu *root*.

```
ole@enterprise:~> su -l
Password:
```

Dann fügen wir den in die Datei `Xschluessel` extrahierten Schlüssel zur `.Xauthority` von *root* hinzu. Da bis jetzt noch keine solche Datei vorhanden war, wird einfach eine neue Datei erstellt.

```
enterprise:~ # xauth merge /home/ole/Xschluessel
xauth: creating new authority file /root/.Xauthority
```

Jetzt nur noch die Umgebungsvariable `DISPLAY` setzen und schon klappt es mit dem X-Client.

```
enterprise:~ # export DISPLAY=:0.0
enterprise:~ # nedit
```

Für den Benutzer *root* geht es sogar noch einfacher. Nach dem Einloggen muß nur die Variable *XAUTHORITY* auf die passende *.Xauthority* gesetzt werden. Und schon klappt es wieder mal mit dem X-Client.

```
ole@enterprise:~> su -l
Password:
enterprise:~ # export XAUTHORITY=/home/ole/.Xauthority
enterprise:~ # nedit
```

Problematischer wird die Übertragung des Magic Cookies bei entfernten Rechnern. Da normalerweise nicht die IP-Nummer des Hosts beim Magic Cookie gespeichert wird sondern sein Hostname, muß auf dem anderen Rechner auch eine passende Namensauflösung laufen.

```
dozent@linux37:~> echo $DISPLAY
:0.0
dozent@linux37:~> xauth list
linux37.local:0 MIT-MAGIC-COOKIE-1 aad2f729bd7ac5cd4c29a91a4867578a
linux37/unix:0 MIT-MAGIC-COOKIE-1 aad2f729bd7ac5cd4c29a91a4867578a
linux37.local:0 XDM-AUTHORIZATION-1 b4396abf05bbad06002c298ce3d8d635
linux37/unix:0 XDM-AUTHORIZATION-1 b4396abf05bbad06002c298ce3d8d635
dozent@linux37:~> xauth extract - linux37.local:0 | \
> ssh -l walter linux36 /usr/X11R6/bin/xauth merge -
Password:
```

Wichtig ist hier nicht einfach die Variable *DISPLAY* zu benutzen. In diesem Fall würde sich der Eintrag nämlich auf *linux37/unix:0* beziehen, der aber den Transport über einen Unix-Socket veranlaßt. Einen solchen Eintrag auf einen anderen Rechner zu übertragen bringt nicht viel.

### 19.5.3 Zugriff über SSH-X11-Weiterleitung

Eine einfachere und sichere Möglichkeit der Bildschirmübertragung ist mittels SSH möglich. Die normale Vorgehensweise zeigt das nächste Beispiel.

```
dozent@linux37:~> xhost +linux36
217.89.70.36 being added to access control list
dozent@linux37:~> ssh -l walter linux36
Password:
Last login: Tue Jun 15 12:50:19 2004
It's a good day to die !
walter@linux36:~> export DISPLAY=linux37:0
walter@linux36:~> nedit
```

Mit SSH geht es nun wesentlich einfacher. Sie verbinden sich einfach wieder über den SSH-Client mit dem entfernten Rechner unter Angabe der Option *-X*. Dann startet der SSH-Server einen virtuellen X-Server (im Beispiel der X-Server *:10*) und setzt die Variable *DISPLAY* entsprechend. Die Daten werden nun vom X-Client an diesen virtuellen X-Server gesendet. Von dort werden die Daten über die SSH-Verbindung weitergeleitet zum lokalen Rechner. Diese Vorgang bezeichnet man auch als *tunneln* und die Verbindung als *SSH-Tunnel*. Der SSH-Client auf dem lokalen Rechner leitet die Daten dann weiter an den lokalen X-Server. Deshalb ist es weder notwendig auf dem entfernten Rechner die Variable *DISPLAY* zu setzen, noch auf dem lokalen Rechner den X-Server für den entfernten Rechner freizugeben.

```
dozent@linux37:~> xhost
access control enabled, only authorized clients can connect
dozent@linux37:~> ssh -X -l walter linux36
```

```

Password:
Last login: Tue Jun 22 11:11:40 2004 from 217.89.70.37
It's a good day to die !
walter@linux36:~> nedit
walter@linux36:~> echo $DISPLAY
localhost:10.0
walter@linux36:~> xauth
Using authority file /home/walter/.Xauthority
xauth> list
linux36/unix:10 MIT-MAGIC-COOKIE-1 6344f27379b67ce38ab6385bc0c760de
xauth> exit

```

Neben der einfachen Konfiguration hat diese Art der Datenübertragung noch einen Vorteil. Da die Päckchen des X-Protokolls nun über die SSH-Verbindung laufen, sind sie verschlüsselt. Ein Außenstehender kann nun nicht mehr den Netzwerkverkehr zwischen X-Client und X-Server belauschen.

### 19.5.4 Zugriff über XDMCP

Ein gute Möglichkeit ältere Rechner noch zu verwerten ist der Einsatz von sogenannten X-Terminals. Diese Geräte stellen im Prinzip nur noch ein rudimentäres Betriebssystem mit einem X-Server zur Verfügung. Wenn das Gerät eingeschaltet und der X-Server gestartet wird, holt sich dieser einen Displaymanager von dem entfernten Rechner. Das Verfahren läuft über das *X Display Manager Control Protocol* kurz *XDMCP* genannt. Für den entfernten Displaymanager werden der UDP-Port 177 und der TCP-Port 6000 benutzt.

#### Entfernter Rechner

Sie müssen dazu auf dem entfernten Rechner das XDMCP für den Displaymanager aktivieren und den Zugriff des X-Terminals auf den Rechner aktivieren. Beim Displaymanager `xdm` müssen Sie den Eintrag `DisplayManager.requestPort` in der Datei `xdm-config` mit einem Ausrufungszeichen auskommentieren.

```

linux36:/etc/X11/xdm # cat xdm-config
!
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
!
!DisplayManager.requestPort:    0

```

Dann muß noch in der Datei `Xaccess` ein Eintrag für alle zugelassenen Rechner gemacht werden. Ein Asterisk dort eingetragen, erlaubt allen Hosts den Zugriff auf den Displaymanager.

Der Displaymanager `kdm` benutzt die Datei `kdmrc` als Konfigurationsdatei. Suchen Sie dort den Abschnitt `[Xdmcp]` auf und ändern Sie den Eintrag `Enable` von `false` auf `true`. Dazu sollte die Portangabe für den UDP-Port auf 177 stehen und eine `Xaccess`-Datei angegeben werden. Da diese Datei von der Syntax identisch mit der Datei für `xdm` ist, können Sie für beide Displaymanager die gleiche Datei verwenden. Der Abschnitt könnte also folgendermaßen lauten.

```

[Xdmcp]
Enable=true
Port=177
Xaccess=/etc/X11/xdm/Xaccess

```

Für den Displaymanager `gdm` müssen Sie in die `gdm.conf` gehen. Dort den Abschnitt `xdmcp` aufsuchen und den Eintrag `Enable=true` hinzufügen. Die Zugriffsbeschränkungen werden auch hier über die Datei `Xaccess` im Konfigurationsverzeichnis geregelt.

## Lokaler Rechner

Auf dem lokalen Rechner müssen Sie nun den X-Server starten. Dabei müssen Sie dem X-Server mitteilen von welchem entfernten Rechner er Konfiguration und X-Clients beziehen soll. Diesen können Sie mit dem Parameter `-query` angeben.

```
linux37:~ # X :0 -query linux36
```

Der lokale Rechner wird damit zum X-Terminal.

## 19.6 X-Clients

Da viele X-Client auf gemeinsamen Bibliotheken (X-Toolkit) basieren, ähnelt sich die Konfiguration der Programme oft. Hier mal ein Blick auf die Konfiguration von X-Clients.

### 19.6.1 Die Standard X-Toolkit Optionen

Die meisten X-Clients unterstützen eine Reihe von Optionen, die aus dem X-Toolkit stammen. Die Befehle betreffen meistens Eigenschaften, die auf alle X-Programme zutreffen können. Hier eine Auswahl von X-Toolkit-Optionen, die von dem Programm `xterm` unterstützt werden.

- bd** *FARBE* Die Option legt die Rahmenfarbe (Border Color) fest. Voreinstellung ist "black".
- bg** *FARBE* Hiermit wird die Hintergrundfarbe (BackGround color) eingestellt. Die Voreinstellung ist "white".
- bw** *NUMMER* Die Breite des Fensters ist auch einstellbar. Die Maßeinheit ist in diesem Fall Pixel. Funktioniert nicht mit jedem Fenstermanager.
- display** *DISPLAY* Diese Option legt den X-Server fest, mit dem sich der X-Client verbinden soll.
- fg** *FARBE* Natürlich ist auch die Vordergrundfarbe einstellbar. Voreinstellung ist "black".
- fn** *SCHRIFT* Den verwendeten Standardfont können Sie hier einstellen. Voreinstellung ist "fixed".
- geometry** *GEOMETRIE* Die Größe und Position eines Fensters kann beim Start mitgegeben werden. Die Angabe *GEOMETRIE* erfolgt in der Form *BREITExHÖHE+XOFF+YOFF*. Die Breite und Höhe des Fensters wird in Abhängigkeit von der Anwendung entweder in Pixel oder Zeichen angegeben. Die Position der linken oberen Ecke dagegen wird immer in Pixel angegeben.
- iconic** Startet das Programm als Icon, wenn es der Fenstermanager erlaubt.
- name** *NAME* Gibt der Anwendung einen anderen Namen als den Dateinamen, damit die Anwendung angesprochen werden kann.
- title** *TEXT* Der Name des Fenster wird hier angegeben, wenn der Fenstermanager dies unterstützt.

## Beispiele

Anhand des Programms `xterm` wollen wir uns mal die Funktion einiger Schalter anschauen.

Sie können die farbliche Gestaltung selbst einstellen. Dabei wird sich der Farbnamen bedient, die in der Farbdatei stehen. Diese Datei wird durch die Direktive `RgbPath` im Abschnitt `Files` der `XF86Config` bestimmt. Sollte kein Eintrag dort stehen ist `/usr/X11R6/lib/X11/rgb.txt` die gesuchte Datei.

```
dozent@linux37:~> grep red /usr/X11R6/lib/X11/rgb.txt
205  92  92      indian red
255  69   0      orange red
```



```

255  0  0          red
219 112 147       pale violet red
199  21 133       medium violet red
208  32 144       violet red
255  0  0          red1
238  0  0          red2
205  0  0          red3
139  0  0          red4
139  0  0          dark red

```

Jetzt können wir unser xterm schön bunt gestalten.

```

dozent@linux37:~> xterm -bg chartreuse -fg goldenrod4 -bd gold1
dozent@linux37:~> xterm -bg black -fg green -bd green
dozent@linux37:~> xterm -bg lavender -fg orchid4 -bd orchid3

```

Auch die Schriftart können Sie vorgeben. Hierbei bedienen Sie sich am besten den Alias-Name von Schriften, ansonsten müßten Sie den kompletten XLFD-String eingeben.

```

dozent@linux37:~> cat /usr/X11R6/lib/X11/fonts/misc/fonts.alias
fixed      -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
variable   -*helvetica-bold-r-normal-**-120-*-*-*-iso8859-1
5x7        -misc-fixed-medium-r-normal--7-70-75-75-c-50-iso8859-1
5x8        -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
6x9        -misc-fixed-medium-r-normal--9-90-75-75-c-60-iso8859-1
6x10       -misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
6x12       -misc-fixed-medium-r-semicondensed--12-110-75-75-c-60-iso8859-1
6x13       -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
6x13bold   -misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso8859-1
7x13       -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
7x13bold   -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
7x13euro   -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-15
7x13eurobold -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-15
7x14       -misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1
7x14bold   -misc-fixed-bold-r-normal--14-130-75-75-c-70-iso8859-1
8x13       -misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
8x13bold   -misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
8x16       -sony-fixed-medium-r-normal--16-120-100-100-c-80-iso8859-1
9x15       -misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
9x15bold   -misc-fixed-bold-r-normal--15-140-75-75-c-90-iso8859-1
10x20      -misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-1
12x24      -sony-fixed-medium-r-normal--24-170-100-100-c-120-iso8859-1

```

Also spielen wir mal mit der Schrift rum. Sollte die angegebene Schrift nicht gefunden werden, wird automatisch *fixed* verwendet.

```

dozent@linux37:~> xterm -fn 5x7
dozent@linux37:~> xterm -fn 12x24
dozent@linux37:~> xterm -fn 8x13bold
dozent@linux37:~> xterm -fn hugo
xterm: unable to open font "hugo", trying "fixed"....

```

Mal eine andere Größe für ein xterm.

```

dozent@linux37:~> xterm -geometry 75x35
dozent@linux37:~> xterm -geometry 75x35+10+10
dozent@linux37:~> xterm -geometry 132x35+50+50
dozent@linux37:~> xterm -geometry 132x35+50+50 -fn 5x7

```

## 19.6.2 Ressourcen

Eine Alternative zu den Kommandozeilenparameter sind die sogenannten Ressourcen. Dies sind Einstellungen, die der Server speichert. Der Client holt sich dann die Informationen vom Server. Die Informationen werden in den Dateien `~/.Xresources` und `/etc/X11/Xresources` abgelegt und beim Start des X-Servers eingelesen und aktiviert. Die Auswertung der Datei und Übertragung der Informationen in den Server erfolgt über das Programm `xrdb`. Dies kann z. B. in einem der `Xsession`-Skripte des Displaymanagers erfolgen, wie es im unteren Beispiel zu sehen ist.

```
dozent@linux37:/etc/X11/xdm> cat Xsession
...
if test -r "$xdefaults" ; then
    if test -r "$XLIBDIR/Xresources"; then
        xrdb -load -retain "$XLIBDIR/Xresources"
        xrdb -I$HOME -merge "$xdefaults"
    else
        xrdb -I$HOME -load -retain "$xdefaults"
    fi
    test -r "$xresources" && xrdb -I$HOME -merge "$xresources"
elif test -r "$xresources" ; then
    if test -r "$XLIBDIR/Xresources"; then
        xrdb -load -retain "$XLIBDIR/Xresources"
        xrdb -I$HOME -merge "$xresources"
    else
        xrdb -I$HOME -load -retain "$xresources"
    fi
elif test -r "$XLIBDIR/Xresources"; then
    xrdb -load -retain "$XLIBDIR/Xresources"
fi
...
```

Natürlich kann man den Befehl auch in der `~/.xinitrc` ausführen lassen.

```
walter@linux37:~> cat .Xresources
# .Xresources
xterm*font:            8x13
xterm*background:      black
xterm*foreground:      green
xterm*geometry:        100x40
walter@linux37:~> cat .xinitrc
# Meine .xinitrc
xrdb .Xresources
xterm &
wm2
```

Im obigen Beispiel können Sie auch typische Ressourcen für `xterm` sehen.

.....  
*Notizen:*

.....

## XXII

Arbeitsblatt  
X-Window

## 22.1

Für diese Aufgaben benötigen Sie zwei PCs. Arbeiten Sie in einem Team von zwei Personen. Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Notieren Sie die Ergebnisse auf einem separaten Zettel.

- 1 Loggen Sie sich auf beiden Rechner (*R1* und *R2*) auf der graphischen Oberfläche als Standardbenutzer ein. Öffnen Sie jeweils drei Terminalemulation, die hier mit *T1*, *T2* und *T3* bezeichnet werden. Ändern Sie in zwei Emulationen dort Ihre Identität einmal zu *walter* (*T2*) und einmal zu *root* (*T3*) indem Sie ein Login simulieren. Beginnen Sie die Aufgaben auf Rechner *R1*.
- 2 In welcher Datei werden die Konfigurationen für den X-Server gespeichert?
- 3 Ermitteln Sie aus dieser Datei den horizontalen und vertikalen Bildschirmfrequenzbereich. In welchen Einheiten wird die Frequenz jeweils angegeben?
- 4 Welche Bildschirmauflösungen stehen Ihnen zur Verfügung? Wo finden Sie die Information in der `XF86Config`.
- 5 Auf welchem PCI-Bus steckt Ihre Graphikkarte laut Konfiguration? Stimmt dies mit dem Informationen des PCI-Systems überein?
- 6 Überprüfen Sie, ob die Windowmanager `twm`, `ctwm`, `ctwm`, `fvwm`, `wm2` und `kdm` installiert sind. Falls dies nicht der Fall sein sollte, holen Sie bitte die Installation nach.
- 7 Wechseln Sie zur Konsole *T2* und starten Sie den X-Server mit dem Befehl `X :1`. Was passiert?
- 8 Beenden Sie diesen X-Server.
- 9 Starten Sie den X-Server erneut und starten Sie noch ein `xterm` für diese X-Server.
- 10 Testen Sie die Windowmanager aus Aufgabe 6 nacheinander aus. Beschreiben Sie kurz für jeden Windowmanager Ihren Eindruck.
- 11 Beenden Sie wieder den X-Server.
- 12 Was passiert, wenn Sie den X-Server mit dem Befehl `startx -- :1` starten?
- 13 Beenden Sie wieder den Server und legen Sie als *walter* in seinem Heimatverzeichnis eine Datei `.xinitrc` an und tragen Sie dort den Windowmanager `twm` ein.
- 14 Führen Sie nun wieder den Befehl `startx -- :1` aus. Was passiert nun. .
- 15 Ändern Sie die Datei `.xinitrc` so ab, daß ein `xterm` und eine X-Message "Guten Tag, Dave" erscheint. Testen Sie dies.
- 16 Welcher RGB-Farbcode versteckt sich im X11 hinter der Farbe "lavender"? Wo finden Sie diese Information?
- 17 Starten Sie ein `xterm` mit der Hintergrundfarbe "MintCream", der Vordergrundfarbe "red4", der Schriftart "6x10", einer Breite von 66 Zeichen und einer Höhe von 33 Zeichen.
- 18 Starten Sie ein `xterm` ohne Optionen?
- 19 Legen Sie als *walter* eine Datei `.Xresources` mit den in Aufgabe 17 genannten Eigenschaften des `xterm` an. Wie muß die Datei lauten?
- 20 Übertragen Sie die Informationen in den X-Server und starten Sie ein `xterm`. Sind die Angaben übernommen worden?

XXII

Arbeitsblatt  
X-Window

22.2

- 21** Beenden Sie den X-Server wieder. Ändern Sie die Datei `.xinitrc` so ab, daß die `xterm`-Einstellungen automatisch beim Starten übernommen werden. Testen Sie dies. Beenden Sie danach den X-Server wieder.
- 22** Ändern Sie die Konfigurationsdatei des X-Servers so ab, daß die Notabschaltung über `<Strg>+<Alt>+<←>` nicht mehr funktioniert.
- 23** Starten Sie wieder mit dem Benutzer `walter` die graphische Oberfläche wie in Aufgabe 14. Testen Sie die Auswirkungen der vorherigen Aufgabe. Beenden Sie wieder die graphische Oberfläche und machen Sie die Änderung rückgängig.
- 24** Legen Sie im Fontverzeichnis (`/usr/X11R6/lib/X11/fonts`) das Verzeichnis `fontasy` an. Starten Sie einen Webbrowser und laden sich von der Webseite <http://www.fontasy.de> fünf Schriften herunter. Speichern Sie die Schriften im neu angelegten Verzeichnis `fontasy`.
- 25** Bereiten Sie das Verzeichnis `fontasy` zum Einsatz als Fontverzeichnis vor. Starten Sie wieder den X-Server wie letztes Mal. Prüfen Sie mit `xfontsel`, ob die Schriften vorhanden sind.
- 26** Binden Sie nun das neue Schriftenverzeichnis temporär ein. Testen Sie den Erfolg mit `xfontsel`.
- 27** Beenden Sie den X-Server wieder und starten ihn erneut. Sind die Schriften noch immer vorhanden?
- 28** Jetzt müssen Sie noch einmal den X-Server beenden. Ändern Sie nun die Konfigurationsdatei so, daß die Schriften immer zur Verfügung stehen. Testen Sie dies wieder in einem neu gestarteten X-Server.
- 29** Lassen Sie sich nun den Displaynamen ausgeben.
- 30** Versuchen Sie nun vom zweiten Rechner (*R2*) aus als Standardbenutzer ein `xterm` auf dem aktuellen Display darzustellen. Was passiert?
- 31** Erlauben Sie X-Clients von *R2* aus Verbindung zu Ihrem X-Server aufzunehmen.
- 32** Wiederholen Sie nun die vorletzte Aufgabe einmal als Standardbenutzer, als `root` und als `walter`. Was passiert?
- 33** Sperren Sie wieder den Zugang für *R2*.
- 34** Lassen Sie sich für `walter` die Magic Cookies anzeigen.
- 35** Übertragen Sie den Magic Cookie für den aktiven X-Server auf *R2* zum Benutzer `walter` und aktivieren Sie ihn.
- 36** Versuchen Sie wieder als Standardbenutzer, als `root` und als `walter` von *R2* aus ein `xterm` auf dem aktuellen X-Server von *R1* anzeigen zu lassen. Was passiert?
- 37** Beenden Sie den aktuellen X-Server auf *R1* und starten ihn dann wieder neu. Wiederholen Sie die letzte Aufgabe. Was passiert?
- 38** Loggen Sie sich nun von *R1* auf *R2* über SSH als `walter` ein. Gehen Sie dabei so vor, daß die X-Verbindung über SSH getunnelt wird. Geben Sie die Variable `DISPLAY` aus und starten Sie ein `xterm`. Was passiert?
- 39** Loggen Sie sich wieder aus.
- 40** Konfigurieren Sie nun *R2* so, daß der Displaymanager auch Verbindungen von anderen Rechnern annimmt.
- 41** Starten Sie wieder als `walter` auf *R1* einen X-Server. Dieser soll nun aber den Displaymanager von *R2* nutzen. Klappt das? Wenn ja, loggen Sie sich ein und erforschen Sie als Abschluß dieser Aufgaben den anderen Rechner.

# Kapitel 20

## Allerlei Wissenswertes

Dieses Kapitel ist ein Sammelsurium von Wissen zu Linux, die ich für mich erst einmal notiert habe und noch keinem Themengebiet zugewiesen habe. Trotzdem finde ich diese Informationen “wissenswert” und möchte Sie Ihnen deshalb nicht vorenthalten.

### 20.1 Dateien und Verzeichnisse

#### 20.1.1 Login Begrüßungstext: `/etc/issue`

Die Datei `/etc/issue` enthält den Begrüßungstext, der vor dem Loginprompt der Konsole steht. Neben reinem Text können aber auch variable Werte durch maskierte Zeichen gesetzt werden.

- `\l` Names des aktuellen Terminals
- `\r` Versionsnummer des Kernels

#### 20.1.2 Nachricht des Tages: `/etc/motd`

Der Inhalt der Datei `/etc/motd` wird nach dem Login durch das Programm `login` dargestellt, bevor die Shell ausgeführt wird. Der Name `motd` steht für “*Message Of The Day*” also “Nachricht des Tages” und wird auch genau dafür verwendet. Dieses Verfahren benötigt wesentlich weniger Ressourcen als eine eMail und informiert doch jeden Benutzer beim Einloggen.

#### 20.1.3 Login-Terminals für `root`: `/etc/securetty`

Die Datei `/etc/securetty` enthält die Terminalbezeichnung für alle Terminals, auf denen sich `root` ins System einloggen darf.

#### 20.1.4 Prozesse sind auch nur Dateien: `/proc`

Schon seit einigen Jahren gibt es ein Pseudo-Dateisystem unter dem Verzeichnis `/proc`. Es wird als Pseudo-Dateisystem bezeichnet, da es sich um ein virtuelles Dateisystem handelt, daß nur im Speicher existiert und kein Äquivalent auf einem Datenträger besitzt. Es enthält Informationen über die im Moment laufender Prozesse und andere Systeminformationen.

Für jeden laufenden Prozess wird ein Verzeichnis mit der PID als Namen angelegt, wie Sie im folgenden Ausschnitt der Dateiliste sehen können.

```
root@defiant:/proc # ls -l
insgesamt 2
dr-xr-xr-x    3 root      root          0 Jan 12 17:52 1
dr-xr-xr-x    3 tapico    users          0 Jan 12 18:00 1001
```

```
dr-xr-xr-x    3 tapico  users          0 Jan 12 18:00 1002
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 11
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 1226
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 13
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 2
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 3
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 324
dr-xr-xr-x    3 root    root            0 Jan 12 18:00 328
...
```

Im Verzeichnis selber befinden sich wiederum Dateien, die nun Informationen zu den Prozeß enthalten. Diese Dateien sind eigentlich nur Verknüpfungen zu Programmen, die diese Informationen zur Verfügung stellen. Schauen wir uns doch mal den init-Prozeß mit der PID 1 an.

```
root@defiant:/proc # cd 1
root@defiant:/proc/1 # ls -l
insgesamt 0
-r--r--r--    1 root    root            0 Jan 12 18:05 cmdline
lrwxrwxrwx    1 root    root            0 Jan 12 18:05 cwd -> /
-r-----    1 root    root            0 Jan 12 18:05 environ
lrwxrwxrwx    1 root    root            0 Jan 12 18:05 exe -> /sbin/init
dr-x-----    2 root    root            0 Jan 12 18:05 fd
-r--r--r--    1 root    root            0 Jan 12 18:05 maps
-rw-----    1 root    root            0 Jan 12 18:05 mem
lrwxrwxrwx    1 root    root            0 Jan 12 18:05 root -> /
-r--r--r--    1 root    root            0 Jan 12 18:05 stat
-r--r--r--    1 root    root            0 Jan 12 18:05 statm
-r--r--r--    1 root    root            0 Jan 12 18:05 status
root@defiant:/proc/1 # head cmdline environ maps stat statm status
==> cmdline <==
init [5]
==> environ <==
HOME=/ TERM=linux BOOT_IMAGE=linux BOOT_FILE=/boot/vmlinuz
==> maps <==
08048000-080ad000 r-xp 00000000 03:04 9011      /sbin/init
080ad000-080b1000 rw-p 00064000 03:04 9011      /sbin/init
080b1000-080b7000 rwxp 00000000 00:00 0
bffff000-c0000000 rwxp 00000000 00:00 0

==> stat <==
1 (init) S 0 0 0 0 -1 4 59 42642 50 71867 0 468 1279 623 9 0 0 0 28 458752 39
4294967295 134512640 134924332 3221225216 3221223412 134581070 0 0 1467013372
680207875 3222540817 0 0 0 0

==> statm <==
42 39 34 35 0 4 5

==> status <==
Name:   init
State:  S (sleeping)
Pid:    1
PPid:   0
TracerPid: 0
Uid:    0      0      0      0
Gid:    0      0      0      0
FDSize: 32
Groups:
VmSize: 448 kB
```

**cmdline** Diese Datei enthält die komplette Kommandozeile für den Prozeß. Es sei denn er wurde ausgelagert oder er ist ein Zombie. In diesem Fall enthält die Datei nur das Null-Zeichen.

**environ** Diese Datei enthält die Umgebungsvariablen für diesen Prozeß. Die Einträge werden durch das Null-Zeichen getrennt.

**exe** Diese Link ist ein Wegweiser zu der für diesen Befehl ausgeführten Binärdatei.

**fd** Dieses Verzeichnis enthält Links zu den Datei, die durch diesen Prozeß geöffnet worden sind.

**map** Diese Datei enthält die für diesen Prozeß genutzten Speicherbereiche und die dafür gültigen Zugriffsrechte. Die möglichen Zugriffsrechte sind

r = lesen  
w = schreiben  
x = ausführen  
s = gemeinsam genutzt (*shared*)  
p = privat

**stat** Diese Datei enthält die Statusinformationen des Prozesses. Der Befehl **ps** (12.8.1) benutzt diese Informationen. Die Informationen sind die PID, Dateiname der Programmdatei in Klammern, Status (z. B. R für Running), PPID, Prozeßgruppen-ID, Session ID, Terminal, Prozeßgruppen-ID des Prozesses, der gerade im Terminal läuft, Flags, 9 Zähler für Speicheroperationen, nice-Wert plus 15, Zeit zum nächsten Timeout, Zeit zum nächsten SIGALRM, Startzeit des Prozesses, Größe des virtuellen Speichers, Anzahl der Speicherseiten im Arbeitsspeicher u.s.w.

Weitere Informationen zu den Prozeßinformationen erhalten Sie über die Manualpages (**man 5 proc**).

Neben den Prozesse enthält das Verzeichnis **/proc** auch Dateien mit allgemeineren Informationen über das System. Die folgende gekürzte Liste zeigt eine Auswahl dieser Dateien und Verzeichnisse.

```
root@defiant:/proc # ls -ld [a-z]*
-r--r--r-- 1 root root 0 Jan 12 18:46 cmdline
-r--r--r-- 1 root root 0 Jan 12 18:46 cpuinfo
-r--r--r-- 1 root root 0 Jan 12 18:46 devices
-r--r--r-- 1 root root 0 Jan 12 18:46 dma
-r--r--r-- 1 root root 0 Jan 12 18:46 filesystems
-r--r--r-- 1 root root 0 Jan 12 16:49 interrupts
-r--r--r-- 1 root root 0 Jan 12 18:46 ioports
-r----- 1 root root 67047424 Jan 12 18:46 kcore
-r----- 1 root root 0 Jan 12 16:47 kmsg
-r--r--r-- 1 root root 0 Jan 12 18:46 ksyms
-r--r--r-- 1 root root 0 Jan 12 18:46 loadavg
-r--r--r-- 1 root root 0 Jan 12 18:46 meminfo
-r--r--r-- 1 root root 0 Jan 12 18:46 modules
-r--r--r-- 1 root root 0 Jan 12 18:46 mounts
dr-xr-xr-x 4 root root 0 Jan 12 18:46 net
-r--r--r-- 1 root root 0 Jan 12 18:46 partitions
-r--r--r-- 1 root root 0 Jan 12 18:46 pci
dr-xr-xr-x 2 root root 0 Jan 12 18:46 scsi
-r--r--r-- 1 root root 0 Jan 12 18:46 stat
-r--r--r-- 1 root root 0 Jan 12 18:46 swaps
-r--r--r-- 1 root root 0 Jan 12 18:46 uptime
-r--r--r-- 1 root root 0 Jan 12 18:46 version
```

**cpuinfo** Informationen über die CPU und ihre Leistung

**devices** Liste der wichtigsten Gerätetreiber

**dma** Liste der verwendeten DMA-Kanäle

**filesystems** Liste der bekannten Dateisysteme

**interrupts** Liste der Interrupts und wie häufig sie aufgerufen worden sind.

**ioports** Lister der bekannten und definierten IO-Schnittstellen

**kcore** Gerätedatei für den Zugang zum physikalischen Hauptspeicher

**kmsg** Hier können Systemfehlermeldungen ausgelesen werden

**ksyms** Liste der Kernelsymbole und ihrer Speicheradressen

**loadavg** Durchschnittliche Belastung des Systems in den letzten Minuten

**meminfo** Enthält Informationen über die aktuelle Speicherauslastung

**modules** Liste der geladenen und aktuelle genutzten Module

**mounts** Liste der eingehängten Dateisysteme und deren Mount-Points

**net** Verzeichnis mit Informationen zur Netzwerkauslastung

**partitions** Liste der vorhandenen Partitionen

**pci** Liste der gefundenen PCI-Geräte

**scsi** Verzeichnis mit Informationen über SCSI-Geräte

**stat** Allgemeine Informationen über das System

**swaps** Liste der Swap-Partitionen

**uptime** Zeit in Sekunden seit Systemstart und die Leerlaufzeit seitdem

**version** Aktuelle Version des Linux-Kernels

## 20.2 YaST

Das “Yet another Setup Tool” gibt es in zwei Versionen. Die Urversion ist rein textbasiert. Sie ist einfach zu bedienen und vor allem schnell. Yast 2 hingegen ist graphikorientiert (kommt daher den Mausschubsern sehr entgegen) und bietet auch einige verbesserte und zusätzliche Funktionen an. Yast 2 kann auch in einem Textmodus betrieben werden, allerdings ist die Handhabung dort nicht sehr effektiv. Seit SuSE 8.0 gibt es kein YaST mehr sondern nur noch Yast 2.

### 20.2.1 Benutzerverwaltung unter YaST

Starten Sie zunächst eine Konsole im X-Window-System oder wechseln Sie zu einer Textkonsole (STRG + ALT + F1). Starten Sie dort YaST indem Sie **yast** eintippen und mit der Return-Taste bestätigen. In dem daraufhin erscheinenden Programm wählen Sie durch die Cursor-Tasten den Menüpunkt: *Administration des Systems*. Dann wählen Sie den Menüpunkt *Benutzerverwaltung*.

Geben Sie im ersten Feld den Namen *walter* ein. Die folgenden Felder werden von YAST automatisch ausgefüllt. Bestätigen Sie einfach die Angaben mit der Return-Taste. Als Passwort geben Sie **hallo** ein und wiederholen das Passwort im Wiederholungsfeld. Mit der Taste F4 legen Sie dann den Benutzer an. Mit der Taste F10 können Sie die Maske wieder verlassen.

### 20.2.2 Installation von weiteren Paketen mit YaST

Da nicht alle Pakete bei einer Standardinstallation installiert werden, ist es oft notwendig weitere Pakete von den SuSE-CDs zu installieren. Auch diese Aufgabe übernimmt **yast**. Starten Sie wieder als *root yast*, wie Sie es auch bei der Benutzereinrichtung gemacht haben.



## Auswahl der Installationsquelle

Unter anderem zeigt das Menüfenster das aktuelle Installationsmedium an (Quellmedium). Steht dort nichts oder der Hinweis [NOT OK], dann müssen Sie das Installationsmedium noch einmal angeben. Wählen Sie dazu den Menüpunkt *Einstellungen zur Installation* und dort den Punkt *Installationsquelle auswählen*. Wenn Sie von CD oder DVD installieren wollen, wählen Sie die passende Rubrik. Dann müssen Sie noch den passenden Gerätenamen angeben. Drücken Sie einmal die Taste <ESC> und schon sind Sie wieder im Startmenü.

## Installation

Wählen Sie diesmal den Menüpunkt *Paketverwaltung*. Über den Menüpunkt *Konfiguration ändern/erstellen* gelangen Sie nun zur Paketauswahl. Das kann etwas länger dauern, da alle Paketbeschreibungen eingelesen werden müssen (Über 2000 Stück bei SuSE 7.3 Professional). Leider bietet *yast* in diesem Menü keine Suche in den Paketen an. Die Pakete sind in Serien thematisch sortiert. Um jetzt z. B. die Korn-Shell nachzuinstallieren, schauen Sie ins Paket *ap*. Da es sich hier um die *Public Domain Korn Shell* handelt, müssen Sie das Paket *pdksh* markieren. Dies erfolgt mit der Leertaste. Die Buchstaben in den eckigen Klammern vorm Paketnamen bedeuten:

X	Zur Installation vorgesehen
i	Ist bereits installiert
D	Zur Deinstallation vorgesehen
R	Zum Updaten vorgesehen

Mit <F10> verlassen Sie das jeweilige Menü wieder. Mit der Taste <F4> können Sie die Sortierung von den Serien zu einer alphabetischen Aufzählung aller Pakete ändern. Dies ist insofern nützlich, da Sie nicht immer wissen, in welcher Serie sich ein Paket befindet.

Nachdem Sie nun die zu installierenden Pakete ausgewählt haben und mit <F10> die Auswahl verlassen haben, können Sie die Installation nun mit dem Menüpunkt *Installation starten* beginnen. Nach der Installation der Pakete startet nun *SuSEConfig* um die Eintragungen in die Konfigurationsdateien für die installierten Programme zu erledigen. Dies kann etwas länger dauern.

## 20.3 Grafik und Bilder

### 20.3.1 convert

Das Kommandozeilentool *convert* konvertiert Bilddateien von einem Bildformat zu einem anderen. Dabei können Filter auf die Bilder angewendet werden.

```
convert [OPTIONEN] QUELLBILDDATEI ZIELBILDDATEI
```

Das Format für die Zieldatei ergibt sich aus der Endung. Neben den gängigen Formaten wie JPEG, PNG, BMP und TIFF werden eine Vielzahl weiterer Formate unterstützt. Eine genaue Liste können Sie der Manpage entnehmen.

Aus der Vielzahl von Optionen möchte ich Ihnen nur ein paar vorstellen.

#### Optionen

<b>-colorspace RAUM</b>	Erzeugt einen anderen Farbraum für das neue Bild. Mögliche Werte sind GRAY, OHTA, RGB, Transparent, XYZ, YCbCr, YIQ, YPbPr, YUV, oder CMYK.
<b>-monochrome</b>	Wandelt das Bild in ein 1-Bit S/W-Bild um.
<b>-verbose</b>	Liefert Informationen über das Bild.

#### Beispiele

Um ein PNG-Bild in ein Postscript-Bild zu konvertieren, wenden Sie folgenden Befehl an:

```
convert screenshot.png screenshot.ps
```

Um das Bild dann noch in ein Schwarz-Weiß-Bild umzuwandeln verwenden Sie  
`convert -monochrome screenshot.png screenshot.ps`

Um das Bild dann noch in ein Schwarz-Weiß-Bild umzuwandeln verwenden Sie  
`convert -colorspace GRAY screenshot.png screenshot.ps`

## 20.4 Internet

### 20.4.1 wget

Jeder kennt sicherlich das Problem eine große Datei herunterzuladen. Gerade wenn man bei 99,99% angekommen ist, bricht der Download ab und man muß wieder von vorne beginnen. Das Kommandozeilentool `wget` ermöglicht einen abgebrochenen Download wieder aufzunehmen. Und nicht nur das. Auch komplette Seiten mit Bildern und sogar ganze Homepages können zum Offline-Lesen heruntergeladen werden. Da es über die Shell gestartet wird, steht einer zeitgesteuerten Nutzung mit den Dämonen `atd` (13.1) und `crond` (13.2.1) auch nichts im Wege. So kann `wget` als Download-Manager wie auch als Offline-Reader eingesetzt werden.

`wget [OPTIONEN] [URL]`

GNU Wget unterstützt die Protokolle HTTP, HTTPS, und FTP sowie die Arbeit mit HTTP Proxies.

#### Optionen

<code>-V</code>	Version ( <code>--version</code> )
<code>-b</code>	Arbeitet im Hintergrund ( <code>--background</code> )
<code>-c</code>	Abgebrochenen Download wieder aufnehmen ( <code>--continue</code> )
<code>-r</code>	Folgt den Links ( <code>--recursive</code> )
<code>-k</code>	Konvertiert Links um fürs Offline-Lesen ( <code>--convert-links</code> )
<code>-p</code>	Alle Elemente (z. B. Bilder) einer Webseite werden ebenfalls heruntergeladen ( <code>--page-requisites</code> )
<code>-l TIEFE</code>	Gibt die Tiefe der Verfolgung an ( <code>--level=TIEFE</code> )
<code>-m</code>	Erstellt exakte Kopie der Seite ( <code>--mirror</code> )
<code>-H</code>	Folgt auch Links zu anderen Hosts ( <code>--span-hosts</code> )
<code>-np</code>	Folgt nicht den Links ins Elternverzeichnis ( <code>--no-parent</code> )
<code>-t VERSUCHE</code>	Anzahl der Downloadversuche ( <code>--tries=Versuche</code> )
<code>-A MUSTER</code>	Lädt nur Dateien, die das MUSTER enthalten ( <code>--accept MUSTER</code> )

#### Beispiele

**Download abgebrochen** Sie wollen die große Datei `http://www.fibel.org/download/lfo-0.4.pdf` (1,5 MB) herunterladen.

`wget http://www.fibel.org/download/lfo-0.4.pdf`

Bricht der Download ab, können Sie durch Eingabe des gleichen Befehle den Download fortsetzen.

`wget http://www.fibel.org/download/lfo-0.4.pdf`

Explizit können Sie `wget` mit der Option `-c` dazu auffordern, den Rest einer Datei zu laden.

`wget -c http://www.fibel.org/download/lfo-0.4.pdf`

Dabei schaut `wget` nach, ob eine Datei `lfo-0.4.pdf` existiert und beginnt dann den restlichen Teil vom Server zu laden. Dies funktioniert natürlich nur bei Servern, die dies auch unterstützen.

**Download über schlechte Leitung** Bei einer schlechten Leitung erfordert es oft mehrere Versuche um eine Datei komplett herunterzuladen. Daher kann `wget` angewiesen werden, den Download öfter zu versuchen.

`wget -t 45 http://www.fibel.org/download/lfo-0.4.pfd.org`

**Download einer Seite inklusive Bilder** Wollen Sie nicht nur die reine HTML-Seite herunterladen, sondern auch alle anderen Elemente, die zur Darstellung dazugehören, dann kommt der Schalter `-p` ins

Spiel.

```
wget -p http://www.fibel.org/index.html
```

**Offline-Version einer Homepage erstellen** Sie haben im Netz eine interessante Webpräsenz entdeckt und wollen diese mit ihrer gesamten Struktur auf Ihrer Platte speichern. Das erreichen Sie durch folgenden Befehl:

```
wget -r -k -l 5 http://www.oleswelt.de/rezepte/
```

Das `-r` sorgt dafür, daß auch die verlinkten Seiten heruntergeladen werden. Über wie viele Webseiten den Links gefolgt werden soll, kann mit dem Schalter `-l` und der Angabe der Tiefe geregelt werden. Der Schalter `-k` sorgt dafür, daß absolute Links in relative Links umgewandelt werden. Wurde die Seite nicht mit heruntergeladen, so wird ein absoluter Link mit mit Protokoll und Domain-Name (z. B. aus `apfelmus.html` wird `http://www.oleswelt.de/rezepte/apfelmus.html`) erzeugt.

**Spiegelung einer Homepage erstellen** Eine Spiegelung (Mirror) ist eine exakte Kopie einer Homepage um sie zusätzlich auf einem oder mehreren Servern anzubieten. Dies erfolgt in der Regel um die Last bei stark frequentierten Webpräsenzen zu verteilen.

```
wget -m 5 http://www.oleswelt.de/rezepte/
```

Entspricht den Schaltern `-r -N -l inf -nr`.

**Nur bestimmte Dateien herunterladen** Um z. B. nur die JPEG-Bilder aus einem FTP-Server-Verzeichnis herunterzuladen, kommt der Schalter `-A` ins Spiel.

```
wget -r -l 1 -np -A .jpg,.jpeg ftp://ftp.heidibilder.de/pic/
```

## 20.5 Die Tastatur

Wenn Sie eine Taste auf Ihrer Tastatur drücken, sendet der Tastatur-Controller einen sogenannten *Scancode* an den Tastatur-Treiber des Kernels. Wie es mit dem Tastenklick weitergeht, hängt von dem verwendeten Programm ab. So arbeiten die meisten X-Server im *Scancode-Modus*. In diesem Fall sendet der Tastatur-Treiber die empfangenen Code-Sequenzen einfach an das Anwendungsprogramm weiter.

Im Fall des *Keycode-Modus* wandelt der Tastatur-Treiber die ankommenden Scancodes in *Keycodes*<sup>1</sup> um, bevor er die Signale an die Anwendung schickt. Dabei kann ein Tastendruck bis zu sechs Scancodes senden.

In allen anderen Fällen wird in einer sogenannten *Keymap* nachgeschaut, welches Zeichen oder welche Zeichenkette an das Anwendungsprogramm gesendet wird, bzw. welche Aktion ausgeführt werden soll.

Schauen wir uns dazu einfach mal ein paar Beispiele mit dem Programm `showkey` an.

### 20.5.1 showkey

Dieser Befehl untersucht die Tastaturcodes auf der Konsole. Dabei gibt er auf der Standardausgabe entweder den Scancode, den Keycode oder den ASCII-Code jeder gedrückten Taste aus.

```
showkey [OPTIONS]
```

#### Optionen

```
-a          (--ascii)
-k          (--keycodes)
-s          (--scancodes)
-h          (--help)
```

Wird keine Option angegeben, dann befindet sich das Programm nach dem Start im Keycode-Modus.

---

<sup>1</sup>Keycode kann als Tastencode übersetzt werden. Wegen der Symmetrie zu Scancode wird hier der Begriff Keycode verwendet.

Im Keycode- und Scancode-Modus beendet sich das Programm, wenn 10 Sekunden nach dem letzten Tastendruck kein weiteres Signal das Programm erreicht. Den ASCII-Modus können Sie durch die Tastenkombination <STRG>+<D> beenden.

### Beispiel

Die Tasten <A>, <B>, <C>, <F1>, <F2>, <ENTF>, <ENDE> und <POS1> in den drei verschiedenen Modi.

```
root@enterprise:~ # showkey -a
Press any keys - Ctrl-D will terminate this program
```

```
a  97 0141 0x61
b  98 0142 0x62
c  99 0143 0x63
^[[A  27 0033 0x1b
    91 0133 0x5b
    91 0133 0x5b
    65 0101 0x41
^[[B  27 0033 0x1b
    91 0133 0x5b
    91 0133 0x5b
    66 0102 0x42
^[[3~  27 0033 0x1b
    91 0133 0x5b
    51 0063 0x33
    126 0176 0x7e
^[[4~  27 0033 0x1b
    91 0133 0x5b
    52 0064 0x34
    126 0176 0x7e
^[[1~  27 0033 0x1b
    91 0133 0x5b
    49 0061 0x31
    126 0176 0x7e
    4  0004 0x04
```

```
root@enterprise:~ # showkey -s
kb mode was XLATE
```

```
press any key (program terminates 10s after last keypress)...
```

```
0x9c
0x1e 0x9e
0x30 0xb0
0x2e 0xae
0x3b 0xbb
0x3c 0xbc
0xe0 0x2a 0xe0 0x53 0xe0 0xd3 0xe0 0xaa
0xe0 0x2a 0xe0 0x4f 0xe0 0xcf 0xe0 0xaa
0xe0 0x2a 0xe0 0x47 0xe0 0xc7 0xe0 0xaa
```

```
root@enterprise:~ # showkey -k
kb mode was XLATE
```

```
press any key (program terminates 10s after last keypress)...
```

```
keycode 28 release
keycode 30 press
keycode 30 release
```

```
keycode 48 press
keycode 48 release
keycode 46 press
keycode 46 release
keycode 59 press
keycode 59 release
keycode 60 press
keycode 60 release
keycode 111 press
keycode 111 release
keycode 107 press
keycode 107 release
keycode 102 press
keycode 102 release
```



# Anhang A

## Listen

### A.1 Die Shell-Befehle

#### A

**alias** Legt eine Kurzbezeichnung für einen Befehl an (5.2.8)  
**apropos** Durchsucht die Whatis-Datenbank nach einem Stichwort (6.3.5)  
**at** Startet einen Job zu einem bestimmten Zeitpunkt (13.1.1)  
**atq** Zeigt die aktuellen Jobs an (13.1.2)  
**atrm** Löscht wartende **at**-Jobs (13.1.3)

#### B

**badblocks** Testet die Oberfläche des Datenträgers auf Fehler (11.2.3)  
**batch** Legt einen Hintergrundprozeß mit sehr geringer Priorität an (13.1.5)  
**bg** Startet einen im Hintergrund gestoppten Job (12.9.6)  
**bunzip2** Entpackt komprimierte Dateien mit der Endung **.bz2** und **.bz** (13.6.2)  
**bzcat** Gibt komprimierte Dateien mit der Endung **.bz2** und **.bz** dekomprimiert auf der Standardausgabe aus (13.6.2)  
**bzip2** Komprimiert einzelne Dateien unter Verwendung der Endung **.bz2** (13.6.2)  
**bzip2recover** Repariert komprimierte Dateien mit der Endung **.bz2** und **.bz** (13.6.2)

#### C

**cal** Zeigt einen Kalender an (4.6.2)  
**case** Führt bestimmte Kommandos in Abhängigkeit von einem Schlüsselwort aus. (15.3.4)  
**cat** Zeigt den Inhalt (mehrerer) Dateien an (4.5.2)  
**cd** Wechselt das Arbeitsverzeichnis (4.3.2)  
**cdparanoia** Liest Audio-CDs aus und schreibt die Tracks auf Festplatte ( )  
**cdrdao** Schreibt Audio-CDs im Disc-at-Once-Modus ( )  
**cdrecord** Brennt Audio- und Daten-CDs von einem Master ( )  
**chgrp** Ändert die Gruppe einer Datei (9.2.2)  
**chmod** Ändert die Rechte für eine Datei (9.2.3)  
**chown** Ändert Besitzer und Gruppe einer Datei (9.2.1)  
**chpasswd** Ändert die Paßwörter mehrerer Benutzer anhand einer Liste (8.2.5)  
**chsh** Ändert die Login-Shell eines Benutzers (4.1.1)  
**cksum** Erzeugt eine CRC-Checksumme für eine Datei (7.4.3)  
**clear** Löscht den Bildschirm (4.6.1)  
**comm** Vergleicht zwei sortierte Dateien miteinander (7.5.2)  
**compress** Komprimiert einzelne Dateien; Endung **.Z** (13.6.3)

**configure** Bei vielen Quellpaketen mitgeliefertes Skript zur Vorbereitung der Installation (14.1.3)

**convert** Kommandozeilentool zur Manipulation und Konvertierung von Bildern (20.3.1)

**cp** Legt Kopien von Dateien an (4.5.3)

**cpio** Kopiert Dateien in und aus einem Archiv (13.5.3)

**crontab** Konfigurationsprogramm für **crond** (13.2.2)

**ctwm** Von Claude Lecommandeur entwickelter Fenstermanager (19.1.2)

**cut** Gibt Spalten aus einer Ausgabe oder Datei aus (7.6.1)

**D**

**date** Zeigt die aktuelle Uhrzeit/Datum an und setzt sie auch (4.6.3)

**dd** Kopiert den Inhalt kompletter Geräte in eine Datei und umgekehrt (4.5.4)

**depmod** Erstellt die Datei **modules.dep** mit den Modulabhängigkeiten (14.5.8)

**df** Zeigt die Nutzung der Datenträger an (10.8.2)

**dialog** Ermöglicht Fenstertechnik auf der Textkonsole (15.3.5)

**diff** Vergleicht zwei Textdateien und gibt die Unterschiede in einem für **patch** lesbaren Format aus (7.7.7)

**dir** Zeigt den ausführlichen Inhalt eines Verzeichnis an, meist ein Alias (4.3.4)

**dmesg** Zeigt den Meldungsbuffer des Kernels an (12.1.3)

**dnsdomainname** Zeigt den DNS-Domainnamen an (18.2.4)

**do** Einleitung des Anweisungsblock für **for** und **while** (15.3.8)

**domainname** Setzt und zeigt den NIS-Domainnamen (18.2.4)

**done** Ende des Anweisungsblock für **for** und **while** (15.3.8)

**du** Zeigt den Platz an, den Verzeichnisse und ihre Dateien einnehmen (10.8.1)

**dumpe2fs** Gibt den Superblock und die Blockgruppeninformationen eines Gerätes aus (11.1.4)

**E**

**e2fsck** Wartungstool für das ext2-Dateisystem (11.2.2)

**echo** Ausgabe von Text auf dem Bildschirm (4.6.5)

**edquota** Konfiguriert mit Hilfe des Standardeditors die Diskquota (11.4.4)

**egrep** Wie **grep** nur mit erweiterten regulären Ausdrücken (7.7.2)

**env** Startet ein Kommando mit anderen Umgebungsvariablen (5.2.5)

**elif** Einleitung eines zusätzlichen Entscheidungsblocks (15.3.3)

**else** Einleitung des Default-Entscheidungsblock (15.3.3)

**expand** Wandelt Tabzeichen in Leerzeichen um (7.7.5)

**export** Macht Variablen überall zugänglich (5.2.2)

**F**

**false** Shell zum Verhindern des Einloggens (8.2.3)

**fc** Bearbeitung der History-Liste (5.3.2)

**fdformat** Führt eine Low-Level-Formatierung für Disketten aus (10.4.2)

**fdisk** Ändert die Partitionseinstellung (10.1.1)

**fg** Holt einen im Hintergrund laufenden Job in den Vordergrund (12.9.5)

**fgrep** Wie **grep**, aber das Suchmuster wird als Zeichenkette interpretiert (7.7.3)

**fi** Ende der **if**-Anweisung (15.3.3)

**file** Gibt den Typ anhand von Inode und **/etc/magic** aus (4.5.11)

**find** Durchsucht das Dateisystem nach Dateien (11.3.1)

**finger** Ausführliche Informationen über eingeloggte Benutzer (8.7.5)

**fips** DOS-Programm zum Verkleinern von Partitionen (1.3.5)

**fmt** Bricht durch Trennen und Zusammenfügen die Ausgabezeilen auf eine vorbestimmte Länge um (7.2.1)

**fold** Bricht durch Trennen die Ausgabezeilen auf eine vorbestimmte Länge um (7.2.3)

**for** Schleife zur Abarbeitung einer Liste (15.3.8)



**free** Zeigt die Speicherauslastung von Arbeitsspeicher und Swap an (12.8.5)  
**fsck** Frontend für die Wartung des Dateisystems (11.2.1)  
**function** Definiert einen neuen Befehl mit Parametern (15.1.3)  
**fvwm** Fenstermanager, gibt es auch im Look-and-Feel von Windows 95 (19.1.2)

## G

**gpsswd** Mitglieder- und Paßwortverwaltung der Gruppen (8.4.8)  
**grep** Filtert Ausgaben nach bestimmten Suchmustern (7.7.1)  
**groupadd** Legt ein neues Gruppenkonto an (8.4.7)  
**groupdel** Löscht ein Gruppenkonto (8.4.11)  
**groupmod** Ändert die Eigenschaften einer Gruppe (8.4.9)  
**groups** Zeigt die Gruppen an, in denen der Benutzer Mitglied ist (8.4.4)  
**grub-install** Installationsskript für den GRUB-Bootloader (12.2.3)  
**grpconv** Aktiviert das Shadow-System und konvertiert die Gruppenpaßwörter (8.5.5)  
**grpunconv** Deaktiviert das Shadow-System und konvertiert die Gruppenpaßwörter (8.5.6)  
**gunzip** Entpackt mit **gzip** gepackte Dateien (13.6.1)  
**gzip** Komprimiert einzelne Dateien (13.6.1)

## H

**halt** Hält den Rechner an ohne Benutzer zu informieren (12.5.2)  
**head** Gibt den Anfang einer Datei aus (7.3.1)  
**help** Zeigt die Hilfe für die in die Shell integrierten Befehle an (6.1.5)  
**history** Zeigt den Inhalt der History-Liste an (5.3.1)  
**hostname** Setzt und zeigt den Rechnernamen an (18.2.4)  
**hwclock** Synchronisiert System- und Hardwareuhr (4.6.4)

## I

**id** Zeigt UID und GID eines Benutzers an (8.4.3)  
**if** Führt Anweisungen in Abhängigkeit Aussagen aus (15.3.3)  
**ifconfig** Konfiguriert Netzwerkschnittstellen (18.1.1)  
**in** Leitet die Menge beim for-Befehl ein (15.3.8)  
**info** Betrachter für die TexInfo-Seiten (6.1.4)  
**init** Wechselt den Runlevel (12.4.4)  
**insmod** Bindet Module in den Kernel ein ohne Abhängigkeitsprüfung (14.5.3)  
**isapnp** Konfiguriert ISA PnP-Geräte ( )  
**isserial** Testet ob ein Terminal eine serielle Schnittstelle ist (16.3.3)

## J

**jobs** Zeigt laufende Hintergrundjobs an (12.9.4)  
**join** Gibt die Zeilen aus zwei Dateien aus, die identische Vergleichsfelder besitzen (7.5.3)

## K

**kdesu** Ausführung von X-Programmen unter anderer Identität (2.3.3)  
**kdm** KDE Desktop Manager ( )  
**kill** Sendet Signale an Prozesse unter Angabe der PID (12.9.7)  
**killall** Sendet Signale an Prozesse unter Angabe des Programmnamens (12.9.8)  
**konsole** Termialemulation des KDE ( )

## L

**last** Zeigt an, wer sich zuletzt eingeloggt hat (8.7.6)  
**lastlog** Zeigt die letzten Einlogzeitpunkte der Benutzer an (8.7.7)  
**ldconfig** Aktualisiert die Liste der genutzten Bibliotheken für **ld.so** (14.2.3)  
**ldd** Zeigt eine Liste der für ein Programm benötigten Bibliotheken an (14.2.2)

**less** Zeigt Dateien seitenweise an, komfortabler als **more** (4.5.10)  
**lesskey** Konfiguriert die Steuerbefehle von **less** (4.5.10)  
**lilo** Einrichten des Bootloaders Lilo (12.3.1)  
**ln** Legt einen harten oder symbolischen Link an (10.7.3)  
**locate** Durchsucht das Dateisystem nach Dateien anhand einer Datenbank (11.3.2)  
**logname** Name, unter dem eingeloggt wurde (8.7.3)  
**logout** Beendet die aktuelle Sitzung (4.6.7)  
**logrotate** Konfiguriert **syslogd**, Verwaltung der alten Logs (13.3.4)  
**ls** Zeigt den Inhalt von Verzeichnissen an (4.3.3)  
**lsmod** Zeigt alle geladenen Kernelmodule an (14.5.2)  
**lspci** Zeigt die PCI-Geräte an (16.2.2)  
**lsusb** Zeigt die USB-Geräte an (16.6.3)

## M

**make** Tool für das Kompilieren von Programmen aus den Quellen (14.1.4)  
**man** Betrachter für die Manual-Pages (6.1.2)  
**mkdir** Legt Verzeichnisse an (4.3.6)  
**mkfifo** Erzeugt eine Named Pipe (10.6.1)  
**mkfontdir** Erzeugt die Schriftübersichtsdatei **font.dir** (19.3.1)  
**mkfontscale** Erzeugt die Übersichtsdatei für skalierbare Schriften (19.3.1)  
**mkfs** Formatiert eine Partition mit einem Dateisystem (10.4.1)  
**mkswap** Erstellt das Swap-Dateisystem (10.5.2)  
**modinfo** Informationen über eingebundene Module (14.5.5)  
**modprobe** Einbinden von Modulen in den Kernel mit erweiterten Funktionen (14.5.6)  
**more** Zeigt Dateien seitenweise an (4.5.9)  
**mount** Bindet ein Dateisystem in den aktuellen Verzeichnisbaum ein (10.3.1)  
**mt** Kontrollprogramm für Magnetbandlaufwerke ( )  
**mv** Ändert Pfad und Namen von Dateien (Verschieben, Umbenennen) (4.5.6)

## N

**newgrp** Wechselt die aktuelle Hauptgruppe eines Benutzers (8.4.5)  
**nice** Legt die Prozeßpriorität eines Kommandos beim Starten fest (12.9.1)  
**nl** Nummeriert die Zeilen vor der Ausgabe durch (7.1.2)  
**nohup** Schützt Programme vor dem Signal **SIGHUP** (12.9.9)  
**nologin** Diese Shell verhindert das Einloggen mit Meldung ( )

## O

**od** Gibt eine Binärdatei in verschiedenen Notationen aus (7.1.3)

## P

**passwd** Ändert das Paßwort eines Benutzers (8.2.4)  
**paste** Fügt Dateien spaltenweise zusammen (7.6.3)  
**patch** Wandelt eine Datei mit einer Patchdatei zu einer neuen Version um (7.7.8)  
**ping** Überprüft ob ein Rechner im Netz erreichbar ist (18.3.1)  
**pnpdump** Liefert ISA-Plug-And-Play-Geräte-Informationen ( )  
**pr** Formatiert eine Textdatei für die Druckausgabe (7.2.2)  
**printenv** Zeigt die Umgebungsvariablen an (5.2.4)  
**procinfo** Zeigt ausführliche Informationen über das Prozeßsystem an (12.8.6)  
**ps** Zeigt die aktuell laufenden Prozesse an (12.8.1)  
**pstree** Zeigt die laufenden Prozesse als Baumstruktur an (12.8.2)  
**pwconv** Aktiviert das Shadow-System und konvertiert die Paßwörter (8.5.3)

**pwd** Zeigt den Namen und Pfad des aktuellen Arbeitsverzeichnis an (4.3.1)  
**pwunconv** Deaktiviert das Shadow-System und konvertiert die Paßwörter (8.5.4)

## Q

**quota** Zeigt die Quoten (Plattenplatz) für die Benutzer an (11.4.1)  
**quotacheck** Aktualisiert die Quoteninformationen (11.4.6)  
**quotaoff** Schaltet die Quotenüberwachung aus (11.4.3)  
**quotaon** Schaltet die Quotenüberwachung ein (11.4.2)  
**quotastats** Zeigt Statistiken über die Quoten an (11.4.7)

## R

**read** Liest eine Zeile von der Shell (15.3.9)  
**reboot** Startet den Rechner neu ohne die Benutzer zu informieren (12.5.2)  
**remount** Funktion zum erneuten Einbinden eines Dateisystems (10.3.2)  
**renice** Verändert die Prozeßpriorität laufender Prozesse (12.9.2)  
**repquota** Überprüfung der Nutzung und der Quoten einer Partition (11.4.5)  
**rm** Löscht den Hardlink zu den angegebenen Dateien (4.5.7)  
**rmdir** Löscht leere Verzeichnisse (4.3.7)  
**rmmod** Entfernt Module aus dem Kernel (14.5.4)  
**route** Konfiguriert die Routingtabelle (18.1.2)  
**rpm** Tool für die Arbeit mit RPM-Paketen (14.3)  
**runlevel** Zeigt alten und neuen Runlevel an (12.4.3)

## S

**script** Mitloggen der Bildschirmausgabe in einer Datei (4.6.8)  
**scsiinfo** Informationen über angeschlossene SCSI-Geräte (16.5.2)  
**sed** Streaming Editor, konvertiert eingegebene Daten (7.7.6)  
**setserial** Konfiguration der seriellen Schnittstelle (16.3.2)  
**sh** Programmname für die Bourne-Shell (8.6.3)  
**shopt** Setzen und Anzeigen von Optionen der Shell (15.4.1)  
**shutdown** Stoppen und Neustarten des Rechners (12.5.1)  
**sort** Sortiert die Zeilen einer Ausgabe oder Datei (7.5.1)  
**source** Ausführen von Bash-Programmen (15.4.2)  
**split** Zerlegt eine Datei in mehrere Dateien (7.3.3)  
**startx** Frontend für **xinit** (19.4.2)  
**stat** Zeigt die Informationen der Inode für eine Datei an (10.6.4)  
**su** Wechselt die Benutzeridentität (8.1.2)  
**sudo** Ausführen von Kommandos unter einer anderen Identität (9.3.1)  
**swapoff** Schaltet ein Swap-Gerät aus (10.5.3)  
**swapon** Schaltet ein Swap-Gerät an (10.5.3)  
**sum** Ermittelt eine Checksumme für eine Datei (7.3.3)  
**sync** Synchronisiert die Daten im Platten-cache mit denen des Datenträgers (10.3.4)  
**syslogd** Der Systemlog-Daemon (13.3.1)

## T

**tac** Gibt die Zeilen einer oder mehrerer Dateien in umgekehrter Reihenfolge aus (7.1.1)  
**tail** Gibt das Ende einer Datei aus (7.3.2)  
**tar** Legt Archive aus mehreren Dateien an (13.5.1)  
**tee** Pipeline mit einer Abzweigung in eine Datei (5.4.3)  
**telinit** Wechselt den Runlevel (12.4.5)  
**telnet** Terminalemulation für Verbindungen mit entfernten Rechnern (18.3.4)

**test** Überprüft Dateien und vergleicht Werte (15.3.2)  
**then** Leitet Anweisungsblock einer Entscheidung ein (15.3.3)  
**top** Echtzeitüberblick über laufende Prozesse (12.8.3)  
**touch** Ändert die Zeiten für den letzten Zugriff und die letzte Änderung auf die aktuelle Zeit (4.5.1)  
**tr** Verändert einzelne Zeichen der Ausgabe (7.7.4)  
**tty** Informationen über die Konsole (4.6.10)  
**tune2fs** Ändert Einstellungen des ext2/ext3-Dateisystems (11.1.5)  
**twm** Der erste weitverbreitete Windowmanager (19.1.2)  
**type** Zeigt Typ und Ort eines Kommandos an (6.3.6)

## U

**umask** Setzt die Standardmaske für die Rechte neu erstellter Dateien (9.2.4)  
**umount** Entfernt ein Dateisystem aus dem aktuellen Verzeichnisbaum (10.3.3)  
**uname** Liefert Informationen über den installierten Kernel (14.4.1)  
**unalias** Löscht ein Alias (5.2.9)  
**uncompress** Enpackt mit **compress** gepackte Dateien (13.6.3)  
**unexpand** Wandelt Leerzeichen in Tabulatoren um (7.7.5)  
**uniq** Entfernt doppelte Zeilen aus sortierten Ausgaben oder Dateien (7.5.3)  
**until** Schleife, die solange läuft bis eine Bedingung wahr wird (15.3.7)  
**updatedb** Aktualisiert die Datenbank `/var/lib/locatedb` für den Befehl **locate** (11.3.2)  
**uptime** Information über Laufzeit und Durchschnittslast des Systems (12.8.4)  
**usbmodules** Ermittelt passende Kernelmodule für USB-Geräte (16.6.4)  
**useradd** Einrichten eines Benutzerkontos (8.3.2)  
**usermod** Einstellungen eines Benutzerkontos ändern (8.3.3)  
**userdel** Löschen eines Benutzerkontos (8.3.5)

## V

**vdir** Zeigt den Inhalt eines Verzeichnis an (4.3.5)  
**vi** Der Standard-Editor unter Linux ( )  
**visudo** Editor für die Datei **sudoers** (9.3.3)

## W

**w** Zeigt die eingeloggten Benutzer an. Ausführlicher als **who** (8.7.1)  
**wc** Zählt Zeichen, Worte und Zeilen (7.4.1)  
**whatis** Kurzbeschreibung des Kommandos (**man -f**) (6.3.3)  
**whereis** Lokalisiert die Programmdatei, Quellcodedatei und die Manual-Page für ein Kommando (6.3.1)  
**which** Zeigt den Ort des Programms an, das durch den Aufruf in der Shell gestartet wird (6.3.2)  
**while** Schleife, die solange läuft solange eine Bedingung wahr ist (15.3.6)  
**who** Liste der eingeloggten Benutzer (8.7.1)  
**whoami** Wer bin ich? (8.7.2)  
**wm2** Sehr kleiner Fenstermanager (19.1.2)

## X

**X** Der X-Window-Server (19.4.1)  
**xargs** Wandelt die Ausgabe eines Befehls in die Parameter eines anderen Befehls um (5.4.4)  
**xauth** Programm zum Erstellen und Verteilen von Magic Cookies (19.5.2)  
**xdm** X-Window Displaymanager (19.4.3)  
**xf86config** Textorientiertes Konfigurationsprogramm für X11 (19.2.3)  
**XF86Setup** Graphisches Konfigurationsprogramm für X11 der XFree86-Gruppe (19.2.3)  
**xfontsel** X-Programm zum Anzeigen der Schriften (19.3.1)  
**xfst** Der X-Fontserver (19.3.2)

**xhost** Tool zur rechnerbasierenden Zugriffsbeschränkung des X-Server (19.5.1)  
**xinit** Komfortables Startprogramm für den X-Window-Server (19.4.2)  
**xrdb** Programm zum Übertragen von X-Client-Ressourcen in den X-Server (19.6.2)  
**xset** Tool zum Verändern von Einstellungen des laufenden X-Servers (19.2.4)  
**xterm** Einfache Terminalemulation des XFree86-Projekts ( )  
**xvidtune** Programm zur Justierung der X-Server-Anzeige für den Montior (19.2.3)  
**Z**  
**zcat** Zeigt den Inhalt von mit **gzip** oder **compress** gepackten Dateien an (13.6.1)

## A.2 Verzeichnisse

**/** Das oberste Verzeichnis (4.4)  
**/bin** Die wichtigsten Programme (4.4)  
**/boot** Kernel und Boot-Dateien (4.4)  
**/dev** Gerätedateien (4.4)  
**/etc** Konfigurations- und Informationsdateien (4.4)  
**/etc/rc.d** Skripte und rc-Verzeichnisse für den Bootvorgang (4.4)  
**/etc/skel** Musterdateien für den Benutzer (4.4)  
**/etc/X11** Konfigurationsdateien X-Window-System (4.4)  
**/home** Die Heimatverzeichnisse der Benutzer (4.4)  
**/lib** Bibliotheken des C-Compilers (Shared Libraries) (4.4)  
**/lib/modules** Elternverzeichnis für die Modulverzeichnisse (14.5.1)  
**/Lost+found** Verzeichnis für wiederhergestellte Dateien (4.4)  
**/mnt** Hilfsverzeichnis zum Mounten (4.4)  
**/proc** Virtuelles Dateisystem zur Verwaltung der Systemprozesse (4.4)  
**/root** Heimatverzeichnis des Superusers (4.4)  
**/sbin** Die wichtigsten Programme für den Superuser (4.4)  
**/tmp** Verzeichnis für temporäre Dateien (4.4)  
**/usr** Wichtige Programme, die nicht zum Booten nötig sind, sowie Dokumentationen (4.4)  
**/usr/bin** Befehle für alle Benutzer (4.4)  
**/usr/doc/FAQ** Frequently Asked Questions (FAQ) ( 6.1)  
**/usr/doc/HOWTO** Die HOWTO-Dokumentationen ( 6.1)  
**/usr/doc/<Programmname>** Dokumentationen für Programme ( 6.1)  
**/usr/include** Standard C/C++-Header-Dateien (4.4)  
**/usr/info** Dateien für TexInfo (info) ( 6.1)  
**/usr/lib** Statische Programmbibliotheken verschiedener Programmiersprachen (4.4)  
**/usr/local** Programme, die nicht Bestandteil des Systems sind (4.4)  
**/usr/local/bin** Binärdateien für die Programme, die nicht Bestandteil des Systems sind (4.4)  
**/usr/local/sbin** Lokal installierte Administrator-Tools (4.4)  
**/usr/man** Dateien für Man-Pages ( 6.1)  
**/usr/sbin** Analog zu **/usr/bin** für den Superuser (4.4)  
**/usr/share/doc/FAQ** Frequently Asked Questions (FAQ) (SuSE) ( 6.1)  
**/usr/share/doc/howto** Die HOWTO-Dokumentationen (SuSE) ( 6.1)  
**/usr/share/doc/packages** Dokumentationen für Programmpakete (SuSE) ( 6.1)  
**/usr/share/info** Dateien für TexInfo (info) (SuSE) ( 6.1)  
**/usr/share/man** Dateien für Man-Pages (SuSE) ( 6.1)  
**/usr/src** Quellcodes der Programme (4.4)

`/usr/src/linux` Quellcode des Linux-Kernels (14.6.1)  
`/usr/X11R6` Dateien für das X-Window-System (4.4)  
`/var` Informationsdateien für das System (4.4)  
`/var/tmp` Verzeichnis für temporäre Dateien (4.4)  
`/var/spool` Warteschlangen (4.4)  
`/var/spool/lp` Druckaufträge (4.4)  
`/var/spool/mail` Mails (4.4)  
`/var/spool/uucp` Kommunikationsdienst uucp (4.4)  
`/var/spool/cron` Aufträge des Daemons `crond` (4.4)

## A.3 Wichtige Dateien

`/etc/at.allow` Liste der erlaubten Benutzer für `at` (13.1.4)  
`/etc/at.deny` Liste der verbotenen Benutzer für `at` (13.1.4)  
`/etc/conf.modules` Konfigurationsdatei für die Module (siehe auch `modules.conf`) (14.5.9)  
`/etc/cron.allow` Liste der erlaubten Benutzer für `crond` (13.2.3)  
`/etc/cron.deny` Liste der verbotenen Benutzer für `crond` (13.2.3)  
`/etc/bashrc` Konfigurationsdatei für interaktive Shells, die nicht die Login-Shell sind (8.6.2)  
`/etc/DIR_COLORS` Konfigurationsdatei für die Farbausgabe von `ls` (4.3.3)  
`/etc/fstab` Konfigurationsdatei für das Einbinden von Dateisystemen in den Verzeichnisbaum (10.3.5)  
`/etc/group` Liste der Gruppenkonten (8.4.1)  
`/etc/gshadow` Liste der Gruppenpaßwörter für das Shadow-System (8.5.2)  
`/etc/host.conf` Resolverkonfiguration für alte `libc` (18.2.1)  
`/etc/inittab` Konfigurationsdatei für den Init-Daemon (12.4.1)  
`/etc/inputrc` Konfigurationsdatei für die Eingabezeile (5.1.2)  
`/etc/issue` Enthält die Identifikationsmeldung, die vor dem Login-Prompt erscheint (20.1.1)  
`/etc/issue.net` Enthält die Identifikationsmeldung für Telnet-Sitzungen ( )  
`/etc/ld.so.cache` Liste der Bibliotheken für den Runtime Linker `ld.so` (14.2.1)  
`/etc/ld.so.conf` Konfigurationsdatei für `ldconfig` (14.2.3)  
`/etc/lilo.conf` Konfigurationsdatei für den Bootmanagerersteller `lilo` (12.3.2)  
`/etc/login.defs` Konfigurationsdatei für den Login-Vorgang (8.2.6)  
`/etc/logrotate.conf` Konfigurationsdatei für das Programm `logrotate` (13.3.4)  
`/etc/magic` Signaturen für Dateitypen (4.5.12)  
`/etc/man.config` Konfigurationsdatei für `man` (6.1.2)  
`/etc/manpath.config` Konfigurationsdatei für `man` (6.1.2)  
`/etc/modules.conf` Konfigurationsdatei für die Module (siehe auch `conf.modules`) (14.5.9)  
`/etc/motd` Enthält die Meldung, die nach dem Einloggen erscheint (20.1.2)  
`/etc/mtab` Liste der zur Zeit gemounteten Dateisysteme (10.3.6)  
`/etc/nsswitch.conf` Resolverkonfiguration für neue `libc6` (18.2.1)  
`/etc/passwd` Liste der Benutzerkonten (8.2.1)  
`/etc/profile` Konfigurationsdatei für die Shell (8.6.1)  
`/etc/securetty` Liste der Terminals für `root`-Anmeldung (20.1.3)  
`/etc/shadow` Liste der Paßwörter für das Shadow-System (8.5.1)  
`/etc/shells` Liste der erlaubten Shells (4.1.1)  
`/etc/shutdown.allow` Liste der Benutzer, die das System herunterfahren dürfen (12.5.1)  
`/etc/syslog.conf` Konfigurationsdatei für den den Daemon `syslogd` (13.3.2)  
`/usr/share/misc/magic` siehe `/etc/magic` (4.5.12)

`/var/lib/locatedb` Datenbank für den Befehl `locate` (11.3.2)  
`/var/log/lastlog` Speichert den letzten Zeitpunkt des Einloggens des Benutzers (13.3.3)  
`/var/log/messages` Meldungen über das Hochfahren des Rechners (13.3.3)  
`/var/log/wtmp` Speichert Logzeiten und Systemstarts (13.3.3)  
`/var/run/utmp` Speichert die aktuell eingeloggtten Benutzer (13.3.3)  
`.bash_history` Speicher für die zuletzt ausgeführten Befehle (5.3)  
`.bash_profile` Individuelle Konfigurationsdatei für interaktive Shells Shell (5.2.6)  
`.bashrc` Individuelle Konfigurationsdatei für interaktive Shells Shell (8.6.2)  
`.dir_colors` Individuelle Konfigurationsdatei für die Farbausgabe von `ls` (4.3.3)  
`.inputrc` Individuelle Konfigurationsdatei für die Eingabezeile (5.1.2)  
`.profile` Individuelle Konfigurationsdatei für die Shell (8.6.1)  
`Makefile` Installationsanleitung für das Tool `make` (14.1.4)  
`modules.dep` Liste der Modulabhängigkeiten (14.5.7)  
`quota.group` Konfigurationsdatei für die Gruppenquoten einer Partition (11.4)  
`quota.h` Konfigurationsdatei für die Standardeinstellung der Quoten einer Partition (11.4.4)  
`quota.user` Konfigurationsdatei für die Userquoten einer Partition (11.4)

## A.4 Umgebungsvariablen

`LD_LIBRARY_PATH` Pfad zu zusätzlichen Programmbibliotheken (14.2.1)  
`LANG` Verwendete Sprache und Währung (6.1.2)  
`MANPATH` Suchpfad für Manualpages (6.1.1)  
`MODPATH` Suchpfad für Kernelmodule (14.5.3)  
`PAGER` Programm zum Anzeigen von Informationen (6.1.2)

## A.5 Glossar

**context switch** Englischer Begriff für *Kontextwechsel*.

**dynamic executable** Programm, das zur Ausführung externen Programmbibliotheken benötigt. Diese werden dynamisch beim Starten des Programms eingebunden. Das Gegenteil sind *static executables*.

**Flaschenhals** Die schwächste Ressource im System, die letztendlich die Verarbeitungsgeschwindigkeit bestimmt. Aufgabe der Systemoptimierung ist es, solche Ressourcen aufzuspüren und den Flaschenhals zu beseitigen.

**Header-Dateien** Dateien, die die Schnittstellenbeschreibungen von Funktionen (meistens C oder C++) enthält, aber nicht den dazugehörigen Code. Wenn ein Programm gegen externe Bibliotheken gelinkt werden, werden die entsprechenden Header-Dateien unbedingt benötigt.

**Kontextwechsel** Das Speichern und Wiederherstellen des CPU-Status (Kontext), damit mehrere Prozesse eine CPU-Ressource gemeinsam nutzen können. Der *Kontextwechsel* ist eine notwendige Bedingung eines Multitasking-Betriebssystems. Da die Kontextwechsel sehr rechenaufwändig sind, wird sehr viel Entwicklungsarbeit in die Optimierung dieses Vorgangs gesteckt.

Ein Kontextwechsel erfolgt normalerweise in folgenden Schritten:

- Stoppen des Prozesses. Die aktuelle Stelle im Programm wird im Programmzeiger festgehalten.
- Speichern der Kontextinformationen des ersten Prozesses im Arbeitsspeicher. Die Kontextinformationen umfassen u.a. die Werte der Register für den Programmzeiger, die Prozessorkontrolle und weitere übliche Register.
- Laden der Kontextinformationen des zweiten Prozesses aus dem Arbeitsspeicher.
- Der Programmzeiger zeigt auf die Stelle im Programm, wo die Bearbeitung abgebrochen wurde. Die Abarbeitung des neuen Prozesses an dieser Stelle fortführen.

Um diesen zeitaufwändigen Vorgang zu verhindern, besitzen einige CPUs die Möglichkeit mehrere Kontexte gleichzeitig in der CPU zu halten.

**Patch** (engl. *Flicken*) Kleine Software-Pakete, mit denen die Entwickler Fehler in Produkten beseitigen. Da sie nur die Korrektur der fehlerhaften Programmabschnitte erhalten, sind sie viel kleiner. Es werden lediglich einige Daten und Funktionen ausgetauscht.

**Prefix** Die vor einem Namen (i.a. Dateinamen) vorangestellte typisierende Zeichenkette. So kennzeichnet z. B. der Prefix `tn_` ein kleines Voransichtbild eines größeren Bildes. Das Gegenteil vom *Prefix* ist der **Suffix**.

**Suffix** Die an einem Namen (i.a. Dateinamen) angehängte typisierende Zeichenkette. So kennzeichnet z. B. der Suffix `.jpg` eine Bilddatei im JPEG-Format. Das Gegenteil vom *Suffix* ist der **Prefix**.

**static executable** Programm, das zur Ausführung keine externen Programmbibliotheken benötigt. Die benötigten Bibliotheksfunktionen werden statisch beim Erstellen des Programms eingebunden. Das Gegenteil sind *dynamic executables*.



## Anhang B

# Der Unterricht

Dieses Skript ist als Begleitmaterial zu meinem Unterricht gedacht. Deshalb möchte ich an dieser Stelle Hinweise und Ratschläge für den Unterricht unterbringen.

### B.1 Exportieren des X11-Desktops

Gerade bei der Arbeit mit dem X-Window-System ist es mühsam den Kursteilnehmern die Arbeitsweise der GUI-Programme zu erklären. Besser ist es, wenn alle Teilnehmer den Desktop beobachten können, während der Dozent die Funktionen vormacht. Meistens wird in solchen Fällen ein sogenannter Beamer eingesetzt, der das Monitorbild auf eine Leinwand wirft. Diese Geräte sind natürlich nicht gerade billig und geraten bei ungünstigen Raum- und Lichtverhältnissen schnell an ihre Grenzen.

Unter Windows ermöglicht das Programm Netmeeting nicht nur das telefonieren übers Netz, sondern auch der Desktop kann auf andere Rechner übertragen werden.

Unter Linux bietet sich natürlich der Einsatz eines X11-Servers an. Das Projekt Virtual Network Computing (VNC) der University of Cambridge, des Department of Engineering und AT&T arbeitet an einem virtuelle X11-Server, der auf anderen Rechnern bedient werden kann.

Dieser X-Server und seine Viewer sind für verschiedene Betriebssysteme wie u.a. Linux und Windows erhältlich. Das erlaubt z. B. die Arbeit auf einem Linux-System, während der Benutzer auf einem Windows-Rechner arbeitet.

Das Paket `vnc` ist auf der SuSE 8.0 enthalten. Nach der Installation steht der X-Server (`Xvnc`), das Startprogramm für den Server (`vncserver`), der Client `vncviewer`) und das Passwort-Programm (`vncpasswd`) zur Verfügung.

Die aktuelle Software und weitergehende Informationen erhalten Sie über die Webseite des Projekts <http://www.uk.research.att.com/vnc/> im Internet.

#### B.1.1 Xvnc

`Xvnc` ist der Unix VNC-Server, der auf einem normalen X-Server basiert. Die Programme können diesen Server genau wie einen normalen X-Server für die Darstellung ihrer Fenster verwenden. Allerdings erscheinen die Fenster auf jedem VNC-Betrachter (`vncviewer`), der mit dem Server verbunden ist. Auch auf dem lokalen Rechner, auf dem Server und Programm laufen, wird ein Betrachter benötigt.

Damit ist `Xvnc` in Wirklichkeit zwei Server in einem. Für Programme ist er ein X-Server und für den VNC-Betrachter ein VNC-Server. Die Display-Bezeichnung beim VNC-Server ist an den normalen X-Server angelehnt. So bezeichnet `defiant:3` das dritte Display auf dem Rechner mit dem Namen *defiant*. Natürlich können auch IP-Nummern wie in `217.89.70.60:13` verwendet werden um ein Display auf einen entfernten Rechner anzusprechen.

Der VNC-Server wird normalerweise nicht direkt über das Programm `Xvnc` gestartet. Das mitgelieferte Perl-Skript `vncserver` vereinfacht den Start des Servers wesentlich.

**Xvnc** [OPTIONEN] [:DISPLAYNUMMER]

Die Optionen für den Xvnc-Befehl sind teilweise speziell, teilweise sind sie aber mit denen des normalen X-Servers identisch. Hier eine Auswahl der wichtigsten Optionen.

#### Optionen

<b>-h</b>	Zeigt die Hilfe an
<b>-name</b>	Name des Servers bzw. exportierten Desktops, erscheint im Viewer
<b>-geometry BIPxHIP</b>	Größe des zu exportierenden Desktops in Pixeln (Standard 1024x768)
<b>-depth TIEFE</b>	Farbtiefe der Anzeige in Bit (Standard 8 bit)
<b>-inetd</b>	Ermöglicht den Start des Servers über inetd
<b>-alwaysshared</b>	Es können mehrere Clients zur gleichen Zeit am gleichen Server arbeiten
<b>-nevershared</b>	Es kann nur ein Client zur Zeit am Server arbeiten
<b>-dontdisconnect</b>	Bei einer eingehenden "not-shared" Verbindung wird der neue Client zurückgewiesen; ohne diese Option wird der alte Client vom Server getrennt
<b>-localhost</b>	Nur Betrachter vom eigenen Rechner dürfen den Server kontaktieren
<b>-rfbauth DATEI</b>	Verwendet die DATEI als Passwort-Datei

### B.1.2 vncserver

Dieses Perl-Skript erlaubt einen wesentlich einfacheren Start des VNC-Servers. Es benutzt die gleichen Optionen wie Xvnc.

**vncserver** [OPTIONEN] [:DISPLAYNUMMER]  
**vncserver -kill** :DISPLAYNUMMER

Die Dateien für VNC werden im Heimatverzeichnis in dem versteckten Verzeichnis `.vnc` abgelegt. Für jeden gestarteten Server wird dort eine Log-Datei und eine PID-Datei abgelegt. Mit Hilfe dieser PID-Datei kann das Skript über die Option `-kill` einen VNC-Server beenden.

Dort befinden sich auch die Passwort-Dateien, denn nur nach Eingabe eines gültigen Passworts kann sich der Betrachter mit dem Server verbinden.

Sehr wichtig ist dort die Datei `xstartup`, denn diese wird nach dem Starten des X-Servers ausgeführt und sorgt für den Start des Windowmanagers und bestimmter Programme.

#### Listing B.1 *Standard xstartup*

```
1: #!/bin/sh
2:
3: xrdp $HOME/.Xresources
4: xsetroot -solid grey
5: xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
6: twm &
```

Im obigen Beispiel wird das X-Window-System mit `xrdb` konfiguriert unter Benutzung der normalen X-Window-Ressourcen-Datei des Benutzers. Der Hintergrund wird mit `xsetroot` auf grau gesetzt. Danach werden ein `xterm`-Terminal und der Windowmanager `twm` gestartet. Die Variable `$VNCDESKTOP` enthält den Namen des laufenden Servers bzw. Desktops.

### B.1.3 vncpasswd

Beim ersten Starten eines VNC-Servers fragt das System nach einem Passwort. Mit diesem Passwort wird sichergestellt, daß nur autorisierte Personen Zugang zum X11-Server erlangen. Das Passwort wird in der Datei `~/.vnc/passwd` gespeichert. Der Befehl `vncpasswd` ändert dieses Passwort bzw. legt weitere Passwortdateien an.

**vncpasswd** [PASSWORTDATEI]

## Beispiel

Sie wollen das Standard-Passwort für die VNC-Server ändern. Dann geben Sie einfach folgendes ein.

```
ole@enterprise:~> vncpasswd
Password:
Verify:
```

Sie werden noch zweimal nach dem neuen Passwort gefragt und schon ist die Datei angelegt. Bei der Angabe eines anderen Dateinamens für die Passwortdatei muß der Name mit Pfad angegeben werden. So legt

```
ole@enterprise:~> vncpasswd public
Password:
Verify:
```

zwar eine Passwortdatei namens `public` an. Diese befindet sich aber im Heimatverzeichnis und nicht im Unterverzeichnis `.vnc`. Richtig ist folgender Befehl.

```
ole@enterprise:~> vncpasswd .vnc/public
Password:
Verify:
```

### B.1.4 vncviewer

Der Client für den VNC-Server ist der `vncviewer`. Er wird in der Kommandozeile gestartet und stellt den Desktop mit Hilfe des X-Servers der Client-Maschine dar.

```
vncviewer [HOST] [:NUMMER] [OPTIONEN]
```

Der `vncviewer` kann ohne Parameter aufgerufen werden. In diesem Fall fragt das Programm interaktiv nach dem Server und der Desktop-Nummer.

#### Optionen

<code>-shared</code>	Normalerweise kann sich nur ein Client mit dem VNC-Server verbinden; diese Option erlaubt mehrfache Verbindungen
<code>-display NUMMER</code>	Gibt den Desktop an, auf dem der Viewer sein Fenster darstellen soll
<code>-passwd DATEI</code>	Verwendet die angegebene DATEI als Passwortdatei anstatt den Benutzer nach dem Passwort zu fragen
<code>-viewonly</code>	Erlaubt keine Maus- oder Tastaturaktivitäten vom Client aus
<code>-fullscreen</code>	Startet den Viewer im Vollbildmodus
<code>-geometry GEO</code>	Hier kann die Geometrie und Position des Viewers angegeben werden.

## Beispiele

Dieser Befehl ruft den VNC-Server mit der Nummer 10 auf dem lokalen Rechner auf.

```
vncviewer :10
```

Hier entgegen wird der Desktop 0 des Rechners *defiant* angesprochen. Dies ist identisch mit der Angabe `defiant:0`.

```
vncviewer defiant
```

Dieser Browser wird als "shared" im Betrachtungsmodus gestartet. Die Passwordeingabe erfolgt über eine Passwortdatei.

```
vncviewer defiant:11 -shared -viewonly -passwd ~/.vnc/public
```

### B.1.5 Beispiel: Einrichtung als Server für zwei Personen

Walter und Willi, die sich an verschiedenen Orten befinden, wollen gemeinsam an einem Programmierproblem arbeiten. Deswegen wollen beide an der Datei gleichzeitig arbeiten können.

Hierfür wird auf der Servermaschine von *walter* der VNC-Server gestartet und das Passwort aus der Datei `~/vnc/work` verwendet.

```
walter@enterprise:~> vncserver :10 -alwaysshared -rfbauth ~/vnc/work
```

```
New 'X' desktop is enterprise:10
```

```
Starting applications specified in /home/walter/.vnc/xstartup  
Log file is /home/walter/.vnc/enterprise:10.log
```

Walter startet den Viewer auf seinem Rechner. Er braucht den Server nicht anzugeben, da er sich auf dem gleichen Rechner befindet.

```
walter@enterprise:~> vncviewer :10  
VNC server supports protocol version 3.3 (viewer 3.3)  
Password:  
VNC authentication succeeded  
Desktop name "walter's X desktop (enterprise:10)"  
Connected to VNC server, using protocol version 3.3
```

Willi startet den Viewer auch auf seinem Rechner. Er muß den Server mit anzugeben.

```
willi@defiant:~> vncviewer enterprise:10  
VNC server supports protocol version 3.3 (viewer 3.3)  
Password:  
VNC authentication succeeded  
Desktop name "willi's X desktop (enterprise:10)"  
Connected to VNC server, using protocol version 3.3
```

Nun können beide zur gleichen Zeit auf dem Server arbeiten und gemeinsam an ihrem Programm arbeiten.

### B.1.6 Beispiel: Einrichtung als Klassenraumserver

Eine der interessantesten Aufgaben des VNC-Servers ist der Export eines X11-Desktops für den Unterricht. Hier tritt das Problem auf, daß ein Betrachter, nämlich der des Dozenten, auf dem X-Server arbeiten muß, während die Kursteilnehmer nicht auf dem Server arbeiten dürfen. Leider gibt es keine Option, die eine solche Einstellung erlaubt. Das Setzen des Viewonly-Flags erfolgt nur über den Viewer.

Der Trick an der Sache ist es zwei X-Server aufzusetzen. Einen "Master"-Server auf dem gearbeitet werden darf und einen "Slave"-Server der nur einen Viewer auf den "Master"-Server enthält. Dieser Viewer ist auf Viewonly gestellt.

Mit diesem Start-Skript werden die Server gestartet.

#### Listing B.2 *vnc*

```
1: #!/bin/sh  
2: # Startet zwei VNC-Server (Master und Slave)  
3:  
4: # Auflösung festlegen  
5: GEOM=950x680  
6: CDEPTH=16  
7:  
8: # Variable auswerten
```

```

 9: case $1 in
10:   start)
11:       # Starten der VNC-Server
12:       # Master starten
13:       # Passworddatei ~/.vnc/privat
14:       vncserver :1 \
15:           -geometry $GEOM \
16:           -depth $CDEPTH \
17:           -alwaysshared \
18:           -name master \
19:           -rfbauth $HOME/.vnc/privat
20:
21:       # Slave starten
22:       # Passworddatei ~/.vnc/public
23:       vncserver :2 \
24:           -geometry $GEOM \
25:           -depth $CDEPTH \
26:           -alwaysshared \
27:           -name slave \
28:           -rfbauth $HOME/.vnc/public
29:
30:       # Viewer zur Kontrolle des Master-Servers starten
31:       # Falls in einem X-Terminal als anderer Benutzer gestartet zeigt
32:       # die Display-Variable auf das aktuelle X-Window
33:       # Der Server muß aber vorher mit
34:       # xhost localhost
35:       # freigegeben werden
36:       DISPLAY=:0.0
37:       export DISPLAY
38:       # Viewer starten
39:       vncviewer :1
40:       ;;
41:
42:   stop)
43:       # Beenden des VNC-Servers
44:       vncserver -kill :1
45:       vncserver -kill :2
46:       ;;
47:   *) # Falsches Kommando
48:       echo "Syntax: vnc start|stop"
49:       ;;
50: esac
51:
52: # Ende

```

Mit dem Parameter start werden die zwei Server und der Master-Viewer gestartet.

```
ole@enterprise:~> vnc start
```

```
New 'master' desktop is enterprise:1
```

```
Starting applications specified in /home/ole/.vnc/xstartup
Log file is /home/ole/.vnc/enterprise:1.log
```

```
New 'slave' desktop is enterprise:2
```

Starting applications specified in /home/ole/.vnc/xstartup  
 Log file is /home/ole/.vnc/enterprise:2.log

VNC server supports protocol version 3.3 (viewer 3.3)

Password:

VNC authentication succeeded

Desktop name "ole's master desktop (enterprise:1)"

Connected to VNC server, using protocol version 3.3

Der Parameter stop sorgt dafür, daß die beiden Server beendet werden.

```
ole@enterprise:~> vnc stop
Killing Xvnc process ID 1994
Killing Xvnc process ID 2014
```

Vorm Einsatz des Skripts müssen mit

```
ole@enterprise:~> vncpasswd ~/.vnc/privat
Password:
Verify:
ole@enterprise:~> vncpasswd ~/.vnc/public
Password:
Verify:
```

zwei Passwortdateien angelegt werden. Die Kursteilnehmer erhalten natürlich nur das Passwort aus der Datei public.

Damit die Server unterschiedlich in ihrer Ausführung sind, muß die Datei xstartup die Server unterscheiden.

### Listing B.3 xstartup

```
1: #!/bin/sh
2:
3: # Start ist abhängig vom Servernamen
4: case $VNCDESKTOP in
5:   slave)
6:       # Slave-Server nur mit Viewer
7:       echo "Slave Modus: $VNCDESKTOP"
8:       vncviewer -viewonly -share -fullscreen \
9:         -passwd $HOME/.vnc/privat :1
10:      ;;
11:
12:   master)
13:       # Master-Server mit KDE
14:       echo "Master Modus: $VNCDESKTOP"
15:       xrdb $HOME/.Xresources
16:       xsetroot -solid grey
17:       #xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
18:       kde &
19:       ;;
20:   *)
21:       # Normaler Server zum Arbeiten
22:       echo "Normaler Server"
23:       xrdb $HOME/.Xresources
24:       xsetroot -solid grey
25:       xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
26:       twm &
27:       ;;
28: esac
```

## Anhang C

# Zertifizierung: Das LPIC-Programm

Das LPIC-Programm (*Linux Professional Institut Certification*) wurde geplant, um die Kompetenz in der Linux-Systemadministration und in der Benutzung der entsprechenden Werkzeugen, zu zertifizieren. Es wurde Distributionsneutral geplant, den Empfehlungen der Linux Standard Base und anderen bedeutenden Standards und Konventionen folgend.

Das Programm wurde dreistufig geplant, welche der Aufgaben zu welcher Stufe passen, wurde in einer Aufgabenbeschreibungsanalyse (*Job Task Analysis (JTA)*) untersucht. Wie alle Prüfungsentwicklungen des LPI wurde auch die JTA unter Anwendung psychometrischer Prozesse durchgeführt, um die erforderliche Qualität sicherzustellen.

### C.1 Junior Level Administration (LPIC1)

Dieses Skript unternimmt den Versuch Ihnen den Stoff des ersten LPIC-Levels näher zu bringen. Die Prüfungen des ersten Levels stehen seit dem 11. Januar 2000 zur Verfügung. Die neueste Fassung stammt vom 28. März 2003. Um die Prüfungen des ersten Levels ablegen zu können gibt es keine Vorbedingungen. Sie müssen lediglich die Prüfungen 101 und 102 bestehen.

Um die Prüfungen des ersten Levels bestehen zu können, müssen Sie:

- auf der UNIX-Kommandozeile arbeiten können,
- leichte Wartungsaufgaben wie Hilfe für Benutzer, Hinzufügen von Benutzern, Backup, Restore sowie Shutdown und Reboot ausführen können,
- die Installation und Konfiguration einer Workstation (einschließlich X) und Anbindung dieser an das LAN, oder Anschluß eines Einzelplatz-PC an das Internet, mittels Modem, durchführen können.

Ausgehend von der Webseite <http://www.lpi.org/de/lpic.html> können Sie sich die “Detaillierten Lernziele” der Prüfungen anzeigen lassen. Die folgenden Texte sind eine leicht gekürzte Fassung der Informationen von dieser Webseite und ihren Unterseiten. Jedem Lernziel wird eine Wichtung zugeordnet, die von 1 bis 10 reicht. Sie zeigt die relative Wichtigkeit eines Lernziels. Lernziele mit höheren Wichtungen werden in der Prüfung mit mehr Fragen berücksichtigt.

#### C.1.1 Prüfung 101

##### Thema 101: Hardware und Systemarchitektur

###### 1.101.1 Konfiguration Grundlegender BIOS Einstellungen Wichtung: 1

Kandidaten sollten in der Lage sein, grundlegende Hardware durch korrekte BIOS Einstellungen zu konfigurieren. Dieser Punkt schliesst ein gutes Verständnis der verschiedenen BIOS Einstellungen, wie die Benutzung von LBA bei IDE Festplatten mit mehr als 1024 Zylindern ein, sowie das Anschalten oder Abschalten von

internen Peripheriegeräten und die Konfiguration ohne externe Geräte, wie Tastaturen. Ebenfalls eingeschlossen sind die korrekte Verwendung der Einstellungen für IRQ, DMA und I/O Adressen für alle, im BIOS einzustellenden Ports und die Fehlerbehandlung.

#### **1.101.3 Konfiguration von Modems und Soundkarten**    Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein sicherzustellen, dass die Geräte Kompatibilitätskriterien erfüllen (speziell, dass es sich bei einem Modem NICHT um ein Win-Modem handelt). Ebenfalls enthalten ist die Überprüfung, dass sowohl Modem und Soundkarte eigene und die richtigen Interrupts, I/O- und DMA-Adressen verwenden, die Installation und Ausführung von sndconfig und isapnp bei PnP-Soundkarten, die Konfiguration des Modems für DFÜ und PPP/SLIP/CSLIP-Verbindungen und das Setzen des seriellen Ports auf 115.2 kbps.

#### **1.101.4 Einrichten von SCSI-Geräten**    Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, SCSI-Geräte unter Verwendung des SCSI-BIOS und der notwendigen Linux-Werkzeuge zu konfigurieren. Sie sollten ebenso in der Lage sein, zwischen den verschiedenen SCSI-Typen zu unterscheiden. Dieses Lernziel beinhaltet die Handhabung des SCSI-BIOS zum Auffinden von verwendeten und freien SCSI-IDs und zum Setzen der korrekten ID-Nummer für verschiedene Geräte, im speziellen für das Boot-Device. Ebenso enthalten ist das Verwalten der Einstellungen im System-BIOS zur Bestimmung der gewünschten Bootreihenfolge, wenn sowohl SCSI- als auch IDE-Laufwerke verwendet werden.

#### **1.101.5 Einrichtung verschiedener PC-Erweiterungskarten**    Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, verschiedene Karten für die unterschiedlichen Erweiterungssteckplätze zu konfigurieren. Sie sollten die Unterschiede zwischen ISA- und PCI-Karten in Hinsicht auf Konfigurationsfragen kennen. Dieses Lernziel beinhaltet das korrekte Setzen von Interrupts, DMAs und I/O-Ports der Karten, speziell um Konflikte zwischen Geräten zu vermeiden. Ebenfalls enthalten ist die Verwendung von isapnp, wenn es sich um eine ISA PnP-Karte handelt.

#### **1.101.6 Konfiguration von Kommunikationsgeräten**    Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, verschiedene interne und externe Kommunikationsgeräte wie Modems, ISDN-Adapter und DSL-Switches zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet die Prüfung von Kompatibilitätskriterien (besonders wichtig, wenn es sich um ein Winmodem handelt), notwendige Hardwareeinstellungen für interne Geräte (Interrupts, DMAs, I/O-Adressen) und das Laden und Konfigurieren von passenden Gerätetreibern. Ebenfalls enthalten sind Anforderungen an die Konfiguration von Kommunikationsgeräten und -schnittstellen, wie z.B. der korrekte serielle Port für 115.2 kbps und korrekte Modemeinstellungen für ausgehende PPP-Verbindungen.

#### **1.101.7 Konfiguration von USB-Geräten**    Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, die USB-Unterstützung zu aktivieren und verschiedene USB-Geräte zu verwenden und zu konfigurieren. Dieses Lernziel beinhaltet die korrekte Auswahl des USB-Chipsatzes und des dazugehörigen Moduls. Ebenfalls enthalten ist das Wissen über die allgemeine Architektur des USB-Schichtenmodells und die verschiedenen Module, die in den einzelnen Schichten verwendet werden.

### **Thema 102: Linux Installation und Package Management**

#### **1.102.1 Entwurf einer Festplattenaufteilung**    Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, ein Partitionsschema für ein Linux-System zu entwerfen. Dieses Lernziel beinhaltet das Erzeugen von Dateisystemen und Swap-Bereichen auf separaten Partitionen oder Festplatten und das Maßschneiden des Systems für dessen geplanter Verwendung. Ebenfalls enthalten ist das Platzieren von /boot auf einer Partition, die den BIOS-Anforderungen für den Systemstart genügt.



**1.102.2 Installation eines Bootmanagers** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, einen Bootmanager auszuwählen, zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet das Bereitstellen alternativer und Sicherungsbootmöglichkeiten (z.B. mittels Bootdiskette).

**1.102.3 Erstellen und Installieren von im Sourcecode vorliegenden Programmen** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, ein ausführbares Programm aus dem Quellcode zu erstellen und zu installieren. Dieses Lernziel beinhaltet die Fähigkeit, ein Source-Paket aus einem Archiv zu extrahieren. Kandidaten sollten in der Lage sein, einfache Anpassungen im Makefile vorzunehmen, wie z.B. Pfade zu ändern oder zusätzliche Verzeichnisse einzubinden.

**1.102.4 Verwaltung von Shared Libraries** Wichtung: 2

Beschreibung: Die Prüfungskandidaten sollten in der Lage sein, Shared Libraries, die von ausführbaren Programmen benötigt werden, zu bestimmen und nötigenfalls zu installieren. Sie sollten ebenfalls in der Lage sein, anzugeben wo sich die Systembibliotheken befinden.

**1.102.5 Verwendung des Debian Paketmanagements** Wichtung: 8

Die Prüfungskandidaten sollten in der Lage sein, mit dem Debian Paketmanagement umzugehen. Dieses Lernziel beinhaltet das Benutzen von Kommandozeilen- und interaktiver Werkzeuge zum Installieren, Updaten oder Deinstallieren von Paketen sowie das Auffinden von Paketen, die spezifische Dateien oder Software enthalten (installierte bzw. nicht installierte Pakete). Ebenfalls enthalten ist das Abfragen von Informationen wie Version, Inhalt, Abhängigkeiten, Paketintegrität und Installationsstatus (ob installiert oder nicht) von Paketen.

**1.102.6 Verwendung des Red Hat Package Managers (RPM)** Wichtung: 8

Die Prüfungskandidaten sollten in der Lage sein, Paketmanagement auf Linuxdistributionen, die RPM-Pakete für die Paketverwaltung verwenden, durchzuführen. Dieses Lernziel beinhaltet die (Neu-)Installation, das Updaten und Entfernen von Paketen sowie das Abfragen von Status- und Versionsinformationen. Ebenfalls enthalten ist das Abfragen von Paketinformationen wie Abhängigkeiten, Integrität und Signaturen. Die Kandidaten sollten auch in der Lage sein, zu bestimmen, welche Dateien von einem Paket zur Verfügung gestellt werden, und das Paket zu finden, dem eine bestimmte Datei entstammt.

**Thema 103: GNU- und Unix-Befehle****1.103.1 Arbeiten auf der Kommandozeile** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, mit der Shell und Kommandos auf der Kommandozeile umzugehen. Das beinhaltet das Schreiben gültiger Kommandos und Kommandoabfolgen, das Definieren, Referenzieren und Exportieren von Umgebungsvariablen, die Benutzung der Kommando-History und Eingabemöglichkeiten, das Aufrufen von Kommandos im und außerhalb des Suchpfades, das Benutzen von Kommandosubstitution, das rekursive Anwenden von Kommandos über einen Verzeichnisbaum und das Verwenden von man, um Informationen über Kommandos zu erhalten.

**1.103.2 Texte mittels Filterprogrammen bearbeiten** Wichtung: 6

Die Prüfungskandidaten sollten in der Lage sein, Filter auf Textströme anzuwenden. Dieses Lernziel beinhaltet das Senden von Textdateien und -ausgaben durch Textfilterprogramme, um die Ausgabe zu modifizieren, und die Verwendung von Standard-Unix-Kommandos, die im GNU textutils Paket enthalten sind.

**1.103.3 Grundlagen des Datei-Managements** Wichtung: 2

Die Prüfungskandidaten sollten in der Lage sein, die grundlegenden Unix Kommandos zum Kopieren und Verschieben von Dateien und Verzeichnissen zu benutzen. Dieses Lernziel beinhaltet das Durchführen fortgeschrittener Dateiverwaltungs-Operationen wie dem rekursiven Kopieren mehrerer Dateien, das rekursive

Löschen von Verzeichnissen und das Verschieben von Dateien, deren Namen einem bestimmten Muster entsprechen. Dies beinhaltet das Benutzen einfacher und komplexerer Wildcard-Muster für das Bestimmen von Dateien, sowie die Verwendung von `find`, um Dateien aufgrund von Typ, Größe oder Zeitstempel ausfindig zu machen und Operationen an ihnen durchzuführen.

#### **1.103.4 Benutzen von Unix Streams, Pipes und Umleitungen** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, Ein- und Ausgabeströme umzuleiten und sie zu verknüpfen, um Textdaten effizient zu verarbeiten. Dieses Lernziel beinhaltet das Umleiten von Standardeingabe, Standardausgabe und Standardfehlerausgabe, das Umleiten der Ausgabe eines Kommandos in die Eingabe eines anderen Kommandos, das Verwenden der Ausgabe eines Kommandos als Argument für ein anderes Kommando und das gleichzeitige Senden einer Ausgabe sowohl an die Standardausgabe als auch in eine Datei.

#### **1.103.5 Erzeugung, Überwachung und Terminierung von Prozessen** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, Prozesse zu verwalten. Dieses Lernziel beinhaltet das Ausführen von Prozessen in Vorder- und Hintergrund, das Bringen eines Jobs vom Hintergrund in den Vordergrund und umgekehrt, das Starten eines Prozesses, der ohne Verbindung zu einem Terminal laufen soll, und die Mitteilung an ein Programm, daß es nach Abmelden weiterlaufen soll. Ebenfalls enthalten ist die Überwachung aktiver Prozesse, die Auswahl und das Sortieren von Prozessen für die Ausgabe, das Senden von Signalen an Prozesse, das Terminieren von Prozessen sowie das Erkennen und Terminieren von X-Anwendungen, die nach dem Schließen der X-Sitzung nicht ordnungsgemäß beendet wurden.

#### **1.103.6 Modifizieren von Prozeßprioritäten** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, Prozeßprioritäten zu verwalten. Dieses Lernziel beinhaltet das Ausführen von Prozessen mit höherer oder niedrigerer Priorität, das Bestimmen der Priorität eines Prozesses und das Ändern der Priorität eines laufenden Prozesses.

#### **1.103.7 Durchsuchen von Textdateien mittels regulärer Ausdrücke** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, Dateien und Textdaten mittels regulärer Ausdrücke zu bearbeiten. Dieses Lernziel beinhaltet das Erzeugen einfacherer regulärer Ausdrücke mit verschiedenen Elementen. Ebenfalls enthalten ist die Verwendung von Regex-Werkzeugen zum Durchführen von Suchen über ein Dateisystem oder Dateiinhalte.

#### **1.103.8 Grundlagen der Dateibearbeitung mit vi** Wichtung:1

Die Prüfungskandidaten müssen in der Lage sein, Textdateien mit `vi` zu bearbeiten. Dieses Lernziel beinhaltet Navigation im `vi`, die grundlegenden `vi` Modi, Einfügen, Bearbeiten, Löschen, Kopieren und Auffinden von Text.

### **Thema 104: Devices, Linux-Dateisysteme, Filesystem Hierarchy Standard**

#### **1.104.1 Erzeugen von Partitionen und Dateisystemen** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, Plattenpartitionen zu konfigurieren und Dateisysteme auf Medien wie Festplatten zu erzeugen. Dieses Lernziel beinhaltet verschiedene `mkfs` Kommandos zur Erzeugung von verschiedenen Dateisystemen auf Partitionen, einschließlich `ext2`, `ext3`, `reiserfs`, `vfat` und `xfs`.

#### **1.104.2 Erhaltung der Dateisystemintegrität** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, die Integrität von Dateisystemen zu prüfen, freien Speicherplatz und Inodes zu überwachen und einfache Dateisystemprobleme zu beheben. Dieses Lernziel beinhaltet die Kommandos, die für die Verwaltung eines Standard-Dateisystems notwendig sind sowie die zusätzlichen Notwendigkeiten eines Journaling Dateisystems.

**1.104.3 Kontrolle des Ein- und Aushängen von Dateisystemen** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, das Einhängen (Mounten) eines Dateisystems zu konfigurieren. Dieses Lernziel beinhaltet die Fähigkeit, Dateisysteme manuell ein- und auszuhängen, die Konfiguration des Mountens von Dateisystemen bei Systemstart und das Konfigurieren von wechselbaren Datenträgern, die von Benutzern gemountet werden können, wie z.B. Bänder, Disketten und CDs.

**1.104.4 Verwalten von Diskquotas** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, Diskquotas für Benutzer zu verwalten. Dieses Lernziel beinhaltet das Einrichten von Diskquotas für ein Dateisystem, das Bearbeiten, Prüfen und Erstellen von Berichten über Userquotas.

**1.104.5 Zugriffskontrolle auf Dateien mittels Zugriffsrechten** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, den Dateizugriff mittels Zugriffsberechtigungen zu steuern. Dieses Lernziel beinhaltet Zugriffsrechte auf reguläre und spezielle Dateien sowie Verzeichnisse. Ebenfalls enthalten sind Zugriffsmodi wie `suid`, `sgid` und `sticky bit`, die Verwendung des Gruppenfeldes für die Vergabe von Zugriffsrechten an Arbeitsgruppen, das `immutable flag` und der voreingestellte Dateierstellungsmodus.

**1.104.6 Verwaltung von Dateieigentümerrechte** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, die Benutzer- und Gruppeneigentümerrechte von Dateien zu steuern. Dieses Lernziel beinhaltet das Ändern des Besitzers und der Gruppenzugehörigkeit einer Datei sowie des Standardeigentümers von neuen Dateien.

**1.104.7 Erzeugen und Ändern von echten und symbolischen Links** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, echte und symbolische Links auf eine Datei zu verwalten. Dieses Lernziel beinhaltet das Erzeugen und Bestimmen von Links, das Kopieren von Dateien über Links und das Verwenden von verknüpften Dateien zur Unterstützung von Tätigkeiten der Systemadministration.

**1.104.8 Auffinden von Systemdateien und Platzieren von Dateien an den korrekten Ort** Wichtung: 5

Die Prüfungskandidaten sollten ausreichend mit dem Filesystem Hierarchy Standard, einschließlich typischer Speicherort von Dateien und der Einteilung der Verzeichnisse, vertraut sein. Dieses Lernziel beinhaltet das Auffinden von Dateien und Kommandos auf einem Linux-System.

**Thema 110: Das X-Window-System****1.110.1 Installation und Konfiguration von XFree86** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, X und einen X-Fontserver zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet die Überprüfung, ob Grafikkarte und Monitor von einem X-Server unterstützt werden, und das Anpassen und Trimmen von X für Grafikkarte und Monitor. Ebenfalls enthalten ist die Installation eines X-Fontservers, die Installation von Schriftarten und das Konfigurieren von X zur Benutzung des Fontservers (möglicherweise durch manuelles Bearbeiten des Abschnitts "Files" in `/etc/X11/XF86Config`).

**1.110.2 Einrichten eines Display Managers** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, einen Display Manager einzurichten und anzupassen. Dieses Lernziel beinhaltet das Aktivieren und Deaktivieren des Display Managers und das Ändern der Willkommensmeldung. Ebenfalls enthalten ist das Ändern der voreingestellten Bitplanes des Display Managers. Weiters enthalten ist die Konfiguration des Display Managers für die Verwendung auf X-Stationen. Das Lernziel deckt die Display Manager XDM (X Display Manager), GDM (Gnome Display Manager) und KDM (KDE Display Manager) ab.

**1.110.4 Installation und Anpassung eines Windowmanagers** Wichtung: 5

Die Prüfungskandidaten sollten in der Lage sein, eine systemweite Desktopumgebung und/oder einen Window Manager einzurichten und ein Verständnis der Anpassungsprozedur für Menüs der Fenstermanager und oder der Desktop-Panels demonstrieren. Dieses Lernziel beinhaltet die Auswahl und Konfiguration des gewünschten X-Terminals (xterm, rxvt, atterm etc.), das Prüfen und Auflösen von Bibliotheksabhängigkeiten von X-Anwendungen und das Exportieren der Ausgabe von X-Anwendungen auf einen Client.

**C.1.2 Prüfung 102****Thema 105: Kernel****1.105.1 Verwalten/Abfragen von Kernel und Kernelmodulen zur Laufzeit** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, Kernel und ladbare Kernelmodule zu verwalten und/oder abzufragen. Dieses Lernziel beinhaltet die Verwendung von Kommandozeilen-Anwendungen, um Informationen über den gerade laufenden Kernel und die Kernelmodule zu erhalten. Ebenfalls enthalten ist das manuelle Laden und Entladen von Modulen nach Bedarf. Dies beinhaltet auch bestimmen zu können, wann Module entladen werden können und welche Parameter von Modulen akzeptiert werden. Prüfungskandidaten sollten in der Lage sein, das System so zu konfigurieren, daß Module auch durch andere Namen als ihre Dateinamen geladen werden können.

**1.105.2 Konfiguration, Erstellung und Installation eines angepassten Kernels und angepasster Kernel-Module** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, einen Kernel und ladbare Kernelmodule aus Quellcode anzupassen, zu erzeugen und zu installieren. Dieses Lernziel beinhaltet das Anpassen der aktuellen Kernelkonfiguration, das Erzeugen eines neuen Kernels und das Erzeugen von Kernelmodulen nach Bedarf. Ebenfalls enthalten ist die Installation des neuen Kernels sowie jedweder Module und das Sicherstellen, daß der Bootmanager den neuen Kernel und die dazugehörigen Dateien findet (generell in /boot, siehe Lernziel 1.102.2 für mehr Details über Bootmanager und deren Konfiguration).

**Thema 106: Booten, Initialisierung, Shutdown, Runlevels****1.106.1 Booten des Systems** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, das System durch den Bootprozeß zu führen. Dieses Lernziel beinhaltet das Übergeben von Kommandos an den Bootlader und die Übergabe von Optionen an den Kernel zum Bootzeitpunkt sowie das Überprüfen von Ereignissen in den Logdateien.

**1.106.2 Ändern des Runlevels und Niederfahren oder Neustart des Systems** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, den Runlevel des Systems zu verwalten. Dieses Lernziel beinhaltet den Wechsel in den Single-User Modus und das Niederfahren oder Rebooten des Systems. Kandidaten sollten auch in der Lage sein, Benutzer vor dem Wechsel des Runlevels zu benachrichtigen und Prozesse sauber zu beenden. Dieses Lernziel beinhaltet auch das Setzen des Standard-Runlevels.

**Thema 107: Drucken****1.107.2 Verwaltung von Druckern und Druckerwarteschlangen** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, Druckerwarteschlangen und Druckjobs von Benutzern zu verwalten. Dieses Lernziel beinhaltet die Kontrolle von Druckservern und Druckerwarteschlangen und das Lösen allgemeiner Druckprobleme.

**1.107.3 Druck von Dateien** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, Druckerwarteschlangen zu verwalten und Druckjobs zu bearbeiten. Dieses Lernziel beinhaltet das Hinzufügen und Entfernen von Jobs an konfigurieren Druckerwarteschlangen und das Konvertieren von Textdateien in PostScript für den Ausdruck.

#### **1.107.4 Installation und Konfiguration von lokalen und Netzwerkdruckern** Wichtung: 1

Prüfungskandidaten sollten in der Lage sein, einen Printerdämon zu installieren und einen Druckerfilter (z.B. `apsfilter` oder `magicfilter`) zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet das Konfigurieren von lokalen und Netzwerkdruckern für ein Linux-System, inklusive PostScript-, Non-PostScript- und Samba-Druckern.

### **Thema 108: Dokumentation**

#### **1.108.1 Benutzung und Verwaltung lokaler Systemdokumentation** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, `man` und die Materialien in `/usr/share/doc/` zu benutzen und zu verwalten. Dieses Lernziel beinhaltet das Auffinden relevanter `man` Pages, das Durchsuchen von `man` Page-Abschnitten, das Auffinden von Kommandos und dazugehöriger `man` Pages und die Konfiguration des Zugangs zu `man` Sourcen und dem `man` System. Ebenfalls enthalten ist die Verwendung der Systemdokumentation in `/usr/share/doc/` und das Bestimmen, welche Dokumentation in `/usr/share/doc/` zu behalten ist.

#### **1.108.2 Finden von Linux-Dokumentation im Internet** Wichtung: 3

Beschreibung: Die Prüfungskandidaten sollten in der Lage sein, Linux-Dokumentation zu finden und zu verwenden. Dieses Lernziel beinhaltet die Verwendung von Linux-Dokumentation bei Quellen wie dem Linux Documentation Project (LDP), auf den Webseiten von Distributoren und Drittanbietern, Newsgruppen, Newsgruppen-Archiven und Mailing-Listen.

#### **1.108.5 Benachrichtigen von Benutzern über systemrelevante Ereignisse** Wichtung: 1

Beschreibung: Die Prüfungskandidaten sollten in der Lage sein, Benutzer über aktuelle Themen bezüglich des Systems zu benachrichtigen. Dieses Lernziel beinhaltet die Automatisierung des Kommunikationsprozesses, z.B. durch Login-Meldungen.

### **Thema 109: Shell, Scripting, Programmieren und Kompilieren**

#### **1.109.1 Anpassung und Verwendung der Shell-Umgebung** Wichtung: 3

Beschreibung: Die Prüfungskandidaten sollten in der Lage sein, Shell-Umgebungen auf die Bedürfnisse der Benutzer hin anzupassen. Dieses Lernziel beinhaltet das Setzen von Umgebungsvariablen (z.B. `PATH`) beim Login oder beim Aufruf einer neuen Shell. Ebenfalls enthalten ist das Schreiben von Bash-Funktionen für oft benutzte Kommandoabfolgen.

#### **1.109.2 Anpassen und Schreiben einfacher Scripts** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, existierende Scripts anzupassen und einfache neue (ba)sh Scripts zu schreiben. Dieses Lernziel beinhaltet die Verwendung der Standard `sh` Syntax (Schleifen, Tests), das Verwenden von Kommandosubstitution, das Prüfen von Kommando-Rückgabewerten, Testen des Status einer Datei und Schicken von Mails an den Superuser unter bestimmten Bedingungen. Ebenfalls enthalten ist die Vorsorge, daß der korrekte Interpreter auf der ersten Zeile (`#!`) von Scripts aufgerufen wird. Weiters enthalten ist die Verwaltung von Speicherort, Eigentum und Ausführungs- und `suid`-Rechten von Scripts.

### **Thema 111: Administrative Tätigkeiten**

#### **1.111.1 Verwalten von Benutzer- und Gruppenkonten und verwandte Systemdateien** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, Benutzerkonten hinzuzufügen, zu löschen, zu deaktivieren und zu ändern. Enthaltene Tätigkeiten beinhalten das Hinzufügen und Löschen von Gruppen und das Ändern der Benutzer- und Gruppeninformation in den passwd/group Datenbanken. Ebenfalls enthalten ist das Erstellen von speziellen und eingeschränkten Konten.

#### **1.111.2 Einstellen von Benutzer- und Systemumgebungsvariablen** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, globale und Benutzerprofile zu modifizieren. Dieses Lernziel beinhaltet das Setzen von Umgebungsvariablen, das Verwalten von skel Verzeichnissen für neue Benutzerkonten und das Setzen des Suchpfades auf die richtigen Verzeichnisse.

#### **1.111.3 Konfigurieren und Nutzen der Systemlogs im Rahmen der administrativen und Sicherheitsanforderungen** Wichtung: 3

Beschreibung: Die Prüfungskandidaten sollten in der Lage sein, die Systemaufzeichnungen zu konfigurieren. Dieses Lernziel beinhaltet das Einstellen der Art und Menge der aufgezeichneten Informationen, das manuelle Prüfen von Logdateien auf wichtige Aktivitäten, das Überwachen der Logdateien, das Einrichten automatischer Rotation und Archivierung der Logs und das Verfolgen von Problemen, die in den Logdateien aufgezeichnet wurden.

#### **1.111.4 Automatisieren von Systemadministrationsaufgaben durch Planen von zukünftig laufenden Jobs** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, cron oder anacron zu verwenden, um Prozesse in regelmäßigen Intervallen ausführen zu lassen, und at zu benutzen, um Jobs zu einer bestimmten Zeit auszuführen. Dies beinhaltet das Verwalten von cron- und at-Jobs und die Konfiguration von Benutzerzugang zu cron- und at-Diensten.

#### **1.111.5 Planung einer effektiven Datensicherungsstrategie** Wichtung: 3

Die Prüfungskandidaten sollten in der Lage sein, eine Sicherungsstrategie zu planen und Dateisysteme automatisch auf verschiedene Medien zu sichern. Die Tätigkeiten beinhalten das Sichern einer rohen Partition in eine Datei oder umgekehrt, das Durchführen teilweiser und manueller Backups, das Überprüfen der Integrität von Backupdateien und teilweises oder vollständiges Wiederherstellen von Backups.

#### **1.111.6 Verwalten der Systemzeit** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, die Systemzeit richtig zu verwalten und die Uhr über NTP zu synchronisieren. Dieses Lernziel beinhaltet das Setzen von Systemdatum und -zeit, das Setzen der BIOS-Uhr auf die korrekte Zeit in UTC, die Konfiguration der korrekten Zeitzone des Systems und die Konfiguration der automatischen Korrektur der Zeit auf die NTP-Uhr.

### **Thema 112: Netzwerkgrundlagen**

#### **1.112.1 Grundlagen von TCP/IP** Wichtung: 4

Die Prüfungskandidaten sollten ein richtiges Verständnis der Netzwerk-Grundlagen demonstrieren können. Dieses Lernziel beinhaltet das Verständnis von IP-Adressen, Netzwerkmasken und ihrer Bedeutung (d.h. Bestimmung einer Netzwerk- und Broadcast-Adresse für einen Host auf Grundlage seiner Subnetzmaske in "dotted quad" oder abgekürzter Notation oder die Bestimmung der Netzwerk-Adresse, Broadcast-Adresse und Netzwerkmaske bei gegebener IP-Adresse und Anzahl von Bits). Dies deckt auch das Verstehen der Netzwerkklassen und klassenloser Subnetze (CIDR) und der für private Netzwerke reservierten Adressen ab. Ebenfalls enthalten ist das Verstehen der Funktion und Anwendung der Default Route. Weiters enthalten ist das Verstehen der grundlegenden Internet-Protokolle (IP, ICMP, TCP, UDP) und der gebräuchlicheren TCP- und UDP-Ports (20, 21, 23, 25, 53, 80, 110, 119, 139, 143, 161).

#### **1.112.3 TCP/IP Konfiguration und Problemlösung** Wichtung: 7

Die Prüfungskandidaten sollten in der Lage sein, Konfigurationseinstellungen und den Status verschiedener Netzwerkschnittstellen einzusehen, zu ändern und zu überprüfen. Dieses Lernziel beinhaltet die manuelle und automatische Konfiguration von Schnittstellen und Routing-Tabellen. Dies bedeutet im speziellen das Hinzufügen, Starten, Stoppen, Neustarten, Löschen und Rekonfigurieren von Netzwerkschnittstellen. Dies beinhaltet auch das Ändern, Listen und Konfigurieren der Routing-Tabelle und die manuelle Korrektur einer falsch gesetzten Default-Route. Kandidaten sollten auch in der Lage sein, Linux als DHCP-Client und TCP/IP-Host zu konfigurieren und Probleme im Zusammenhang mit der Netzwerkkonfiguration zu lösen.

#### **1.112.4 Konfiguration von Linux als PPP-Client** Wichtung: 3

Die Prüfungskandidaten sollten die Grundlagen des PPP-Protokolls verstehen und in der Lage sein, PPP für ausgehende Verbindungen zu konfigurieren und zu verwenden. Dieses Lernziel beinhaltet die Beschreibung der Sequenz des Verbindungsaufbaus (bei gegebenem Login-Beispiel) und das Einrichten von automatisch bei Verbindungsaufbau auszuführenden Kommandos. Ebenfalls enthalten ist die Initialisierung und die Beendigung einer PPP-Verbindung mittels Modem, ISDN oder ADSL und die Einstellung der automatischen Neuverbindung bei Verbindungsabbruch.

### **Thema 113: Netzwerkdienste**

#### **1.113.1 Konfiguration und Verwaltung von inetd, xinetd und verwandten Diensten** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, die über inetd verfügbaren Dienste zu konfigurieren, tcp-wrappers für das Erlauben oder Verweigern von Diensten auf Host-Ebene zu verwenden, Internetdienste manuell zu starten, stoppen und neu zu starten und grundlegende Netzwerkdienste einschließlich Telnet und FTP zu konfigurieren. Ebenfalls enthalten ist das Ausführen von Diensten unter einer anderen als der voreingestellten Benutzerkennung in inetd.conf.

#### **1.113.2 Verwendung und allgemeine Konfiguration von Sendmail** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, einfache Einstellungen in den Sendmail Konfigurationsdateien vorzunehmen (einschließlich des "Smart Host" Parameters, wenn notwendig), Mail-Aliases anzulegen, die Mail-Queue zu verwalten, Sendmail zu starten und zu stoppen, Mail-Weiterleitung zu konfigurieren und allgemeine Sendmail-Probleme zu lösen.

#### **1.113.3 Verwendung und allgemeine Konfiguration von Apache** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, einfache Einstellungen in den Apache Konfigurationsdateien vorzunehmen, httpd zu starten, anzuhalten und neu zu starten und das automatische starten von httpd beim Systemstart einzurichten. Dies beinhaltet nicht die fortgeschrittene Konfiguration von Apache.

#### **1.113.4 Richtiges Verwalten von NFS, smb und nmb Dämonen** Wichtung: 4

Die Prüfungskandidaten sollten wissen, wie man im Netzwerk freigegebene Dateisysteme mittels NFS einbindet, wie man NFS für den Export lokaler Dateisysteme konfiguriert und wie man den NFS-Server startet, anhält und neustartet. Ebenfalls enthalten ist die Installation und Konfiguration von Samba unter Verwendung des mitgelieferten GUI-Tools oder durch direkte Bearbeitung von /etc/smb.conf (Achtung: Bewußt ausgeschlossen sind fortgeschrittene Themen der NT-Domänen; einfaches Freigeben von Verzeichnissen und Druckern sowie das korrekte Einrichten von nmbd als WINS-Client sind jedoch enthalten).

#### **1.113.5 Einrichtung und Konfiguration grundlegender DNS-Dienste** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, Namensauflösung zu konfigurieren und Probleme mit lokalen caching-only Nameservern zu lösen. Dies benötigt ein Verständnis der Prozesse der Domainregistrierung und der DNS-Auflösung. Ebenfalls erforderlich ist ein Verständnis der wichtigsten Unterschiede der Konfigurationsdateien von BIND und BIND 8.

#### **1.113.7 Einrichten von Secure Shell (OpenSSH)** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, OpenSSH zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet grundlegende Installation und Problemlösung von OpenSSH sowie die Konfiguration von sshd für den automatischen Start beim Booten.

## **Thema 114: Sicherheit**

### **1.114.1 Durchführung von sicherheitsadministrativen Tätigkeiten** Wichtung: 4

Die Prüfungskandidaten sollten in der Lage sein, die Systemkonfiguration zu überprüfen, um die Sicherheit des Hosts in Übereinstimmung mit lokalen Sicherheitsrichtlinien sicherzustellen. Dieses Lernziel beinhaltet die Konfiguration von tcpwrappers, das Finden von Dateien mit gesetztem SUID/SGID-Bit, das Überprüfen von Softwarepaketen, das Setzen oder Ändern von Benutzerkennwörtern und des Ablaufs von Kennwörtern, das Aktualisieren von Programmdateien nach Empfehlung von CERT, Bugtraq und/oder Sicherheitswarnungen des Distributors. Ebenfalls enthalten ist ein grundsätzliches Wissen über ipchains und iptables.

### **1.114.2 Grundabsicherung von Rechnern** Wichtung: 3

Die Prüfungskandidaten sollten über Grundlagen zur Absicherung von Rechnern Bescheid wissen. Diese Tätigkeiten enthalten die Konfiguration von syslog, Shadow-Paßwörter, das Einrichten eines Mail-Alias für die Mails von root und das Stilllegen von nicht verwendeten Netzwerkdiensten.

### **1.114.3 Absicherung von Rechnern auf Benutzerebene** Wichtung: 1

Die Prüfungskandidaten sollten in der Lage sein, Sicherheit auf Benutzerebene zu implementieren. Die Tätigkeiten beinhalten das Verwalten von Beschränkungen auf Benutzerlogins, Prozesse und Speichernutzung.



# Anhang D

## Open Publication License

### D.1 Englische Version

#### 1. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

Copyright (c) 2000-2004 by Ole Vanhoefer.  
This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section 6).

Commercial redistribution of Open Publication-licensed material is permitted.

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

#### 2. COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

#### 3. SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

**SEVERABILITY.** If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

**NO WARRANTY.** Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

#### 4. REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

- (a) The modified version must be labeled as such.

- (b) The person making the modifications must be identified and the modifications dated.
- (c) Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
- (d) The location of the original unmodified document must be identified.
- (e) The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

#### 5. GOOD-PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

- (a) If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
- (b) All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
- (c) Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

#### 6. LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

- (a) To prohibit distribution of substantively modified versions without the explicit permission of the author(s). Substantive modifications defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.  
To accomplish this, add the phrase 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.
- (b) To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.

## D.2 Deutsche Version

Inoffizielle deutsche Übersetzung des englischen Originals (von Stefan Meretz).

### 1. ERFORDERNISSE FÜR UNMODIFIZIERTE UND MODIFIZIERTE VERSIONEN

Open-Publication-Arbeiten dürfen als Ganzes oder in Teilen reproduziert und verteilt werden, in beliebigen Medien, physisch oder elektronisch, vorausgesetzt, die Bedingungen dieser Lizenz gehören dazu, und diese Lizenz oder ein Verweis auf diese Lizenz (mit jeder Option, die von dem Autor / den Autoren und/oder dem Herausgeber gewählt wurde) wird in der Reproduktion angezeigt.

Eine geeignete Form einer Aufnahme durch Verweis lautet wie folgt:

Copyright (c) 2000-2004 by Ole Vanhofer. Dieses Material darf nur gemäß der Regeln und Bedingungen wie sie von der Open Publication Licence, Version v0.4, festgelegt werden, verteilt werden (die letzte Version ist gegenwärtig verfügbar unter <http://www.opencontent.org/openpub/>).

Die kommerzielle Weiterverbreitung von Open Publication lizenziertem Material ist zu den aufgeführten Bedingungen ausdrücklich gestattet.

Jegliche Publikation im Standard- (Papier-) Buch-Format erfordert die Zitierung der Original-Herausgeber und Autoren. Die Namen von Herausgeber und Autor/en sollen auf allen äußeren Deckflächen des Buchs erscheinen. Auf allen äußeren Deckflächen des Buchs soll der Name des Original-Herausgebers genauso groß sein wie der Titel der Arbeit und so einnehmend genannt werden im Hinblick auf den Titel.

### 2. COPYRIGHT

Das Copyright jeder Open Publication gehört dem Autor / den Autoren oder Zeichnungsberechtigten.

### 3. GÜLTIGKEITSBEREICH DER LIZENZ

Die nachfolgenden Lizenzregeln werden auf alle Open-Publication-Arbeiten angewendet, sofern nicht explizit anders lautend im Dokument erwähnt.

Die bloße Zusammenfassung von Open-Publication-Arbeiten oder eines Teils einer Open-Publication-Arbeit mit anderen Arbeiten oder Programmen auf dem selben Medium bewirkt nicht, dass die Lizenz auch auf diese anderen Arbeiten angewendet wird. Die zusammengefasste Arbeit soll einen Hinweis enthalten, die die Aufnahme von Open-Publication-Material und eine geeignete Copyright-Notiz angibt.

**ABTRENNBARKEIT.** Wenn irgendein Teil dieser Lizenz durch irgendeine Rechtsprechung außer Kraft gesetzt werden, bleiben die verbleibenden Teile der Lizenz in Kraft.

**KEINE GEWÄHRLEISTUNG.** Open-Publication-Arbeiten werden lizenziert und verbreitet "wie sie sind" ohne Gewährleistung jeglicher Art, explizit oder implizit, einschließlich, aber nicht begrenzt auf, der impliziten Gewährleistung des Vertriebs und der Geignetheit für einen besonderen Zweck oder eine Gewährleistung einer non-infringement.

### 4. ERFORDERNISSE FÜR MODIFIZIERTE ARBEITEN

Alle modifizierten Versionen, die durch diese Lizenz abgedeckt werden, einschließlich von Übersetzungen, Anthologien, Zusammenstellungen und Teildokumenten, müssen die folgenden Erfordernisse erfüllen:

- (a) Die modifizierte Version muss als solche gekennzeichnet werden.
- (b) Die Person, die die Modifikationen vornimmt, muss genannt und die Modifikationen müssen datiert werden.
- (c) Danksagungen der Original-Autoren und -Herausgebers - sofern vorhanden - müssen in Übereinstimmung mit der normalen akademischen Zitierungspraxis erhalten bleiben.

- (d) Der Ort des originalen unmodifizierten Dokuments muss benannt werden.
- (e) Die Namen der Original-Autoren dürfen nicht benutzt werden ohne die Erlaubnis des Original-Autors / der Original-Autoren.

### 5. EMPFEHLUNGEN EINER GUTEN PRAXIS

In Ergänzung zu den Erfordernissen dieser Lizenz, wird von den Weiterverteilenden erwartet und ihnen stark empfohlen:

- (a) Wenn Sie Open-Publication-Arbeiten als Hardcopy oder auf CD-ROM verteilen, schicken Sie eine E-Mail-Ankündigung Ihrer Absicht der Weiterverteilung mindestens dreißig Tage bevor Ihr Manuskript oder das Medium endgültig festgelegt ist, um den Autoren Zeit zu geben aktualisierte Dokumente anzubieten. Die Ankündigung sollte die Änderungen beschreiben, die gegebenenfalls am Dokument vorgenommen wurden.
- (b) Alle substantiellen Modifikationen (einschließlich Löschungen) sind entweder im Dokument klar zu kennzeichnen oder sonst in einem Anhang zu beschreiben.
- (c) Schließlich, obwohl nicht erforderlich unter dieser Lizenz, ist es, eine vorgeschlagene gute Form einer kostenlosen Kopie jedes Hardcopy- und CD-ROM-Ursprungs einer unter Open Publication lizenzierten Arbeit dem Autor / den Autoren anzubieten.

### 6. LIZENZ-OPTIONEN

Der/die Autor/en und/oder der Herausgeber eines unter Open Publication lizenzierten Dokuments darf bestimmte Optionen durch Anhängen von Regelungen an den Lizenz-Verweis oder die Lizenz-Kopie wählen. Diese Optionen sind empfohlener Teil der Lizenzbestimmungen und müssen in abgeleiteten Arbeiten in die Lizenz eingefügt werden.

- (a) Verhindern der Verteilung von substantiell modifizierten Versionen ohne explizite Erlaubnis des Autors / der Autoren. "Substantielle Modifizierung" ist definiert als eine Änderung des semantischen Inhalts des Dokuments und schließt bloße Format-Änderungen oder typographische Korrekturen aus.

Zur Anwendung fügen Sie den Satz 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder' (Verbreitung von substantiell modifizierten Versionen dieses Dokuments ist ohne die explizite Erlaubnis des Copyright-Inhabers untersagt) dem Lizenz-Verweis oder der Lizenz-Kopie hinzu.

- (b) Verhindern jeglicher Veröffentlichung dieser Arbeit oder abgeleiteter Arbeiten im Ganzen oder in Teilen in Standard- (Papier-) Buchform für kommerzielle Zwecke ohne vorherige Erlaubnis durch den Copyright-Inhaber.

Zur Anwendung fügen Sie den Satz 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder' (Verbreitung dieser Arbeit oder abgeleiteter Arbeiten in Teilen in Standard- (Papier-) Buchform für kommerzielle Zwecke ohne vorherige Erlaubnis durch den Copyright-Inhaber ist untersagt) dem Lizenz-Verweis oder der Lizenz-Kopie hinzu.

# Index

.bash\_history, 99  
.dir\_colors, 73  
.inputrc, 91  
.plan, 162  
.profile, 160  
.project, 162  
.xinitrc, 441  
.xserverrrc, 441  
/bin/false, 147  
/bin/true, 147  
/boot/grub/menu.lst, 233  
/boot/vmlinuz, 230  
/dev/zero, 194  
/etc/DIR\_COLORS, 73  
/etc/X11/XF86Config, 428  
/etc/X11/fs/config, 439  
/etc/X11/fs/xf86.config, 439  
/etc/X11/xinit/xinitrc, 441  
/etc/X11/xinit/xserverrrc, 441  
/etc/at.allow, 267  
/etc/at.deny, 267  
/etc/bashrc, 160  
/etc/conf.modules, 323  
/etc/cron.allow, 270  
/etc/cron.daily, 268  
/etc/cron.deny, 270  
/etc/cron.monthly, 268  
/etc/cron.weekly, 268  
/etc/crontab, 268  
/etc/fstab, 189  
/etc/group, 154  
/etc/gshadow, 158  
/etc/host.conf, 412  
/etc/hosts, 413  
/etc/init.d, 245  
/etc/init.d/setserial, 365  
/etc/inittab, 240  
/etc/inputrc, 91  
/etc/issue, 453  
/etc/ld.so.cache, 295  
/etc/ld.so.conf, 296  
/etc/lilo.conf, 238  
/etc/login.defs, 148  
/etc/magic, 83  
/etc/manpath.config, 112  
/etc/modules.conf, 243, 323  
/etc/motd, 453  
/etc/mtab, 190  
/etc/networks, 413  
/etc/nologin.txt, 147  
/etc/nsswitch.conf, 412  
/etc/passwd, 144  
/etc/profile, 159  
/etc/resolv.conf, 414  
/etc/securetty, 453  
/etc/shadow, 157  
/etc/shells, 68  
/etc/sudoers, 174  
/etc/syslog.conf, 270  
/proc, 453  
/proc/bus/pci, 358  
/proc/bus/pci/device, 358  
/proc/dma, 361  
/proc/interrupts, 360  
/proc/ioports, 360  
/proc/loadavg, 249  
/proc/modules, 319  
/proc/pci, 359  
/proc/scsi, 370  
/sbin/nologin, 147  
/usr/share/pci.ids, 358  
/var/cache/man/index.db, 118  
/var/lib/locatedb, 218  
/var/log/boot.msg, 273  
/var/log/lastlog, 273  
/var/log/messages, 230, 272  
/var/log/wtmp, 272  
/var/run/syslogd.pid, 270  
/var/run/utmp, 272  
/var/spool/cron, 268  
/var/spool/cron/tabs, 268  
?, 339  
\$?, 339  
0.0.0.0, 406  
  
Adressierung, 386  
Advanced Linux Sound Architecture, 367  
afm, 437  
Alias, 98  
alias, 98  
AllowMouseOpenFail, 429  
ALSA, 367  
apropos, 118  
ARPA, 382  
ARPANET, 383  
Asterisk, 80

- at, 266
- at-Dämon, 265
- atd, 265
- atq, 267
- atrm, 267
- Ausfall-Server, 275
- Ausgabe, 100
- Backup
  - Lagerung, 278
  - Planung, 275
  - Strategien, 274
  - Typen, 277
- Backup-Server, 275
- badblocks, 216
- Bash, 68
  - Hilfe, 113
- batch, 267
- Benutzer, 143
  - Anlegen, 34, 151
  - Bearbeiten, 35, 152
  - Einrichten, 151
  - Entfernen, 35, 152
- Benutzerkonto, 144
- Benutzervariablen, 96
- Benutzerverwaltung, 34
- bg, 259
- Bibliothek, 294
- Binärsystem, 384
- BIOS, 355
- Blockgruppe, 211
- BoardName, 431
- Boleyn, Erich, 232
- Bootblock, 211
- Bootdiskette, 28, 79
- Bootflag, 185
- Bootloader, 230
- Bootmanager, 230
- BOOTP, 411
- Bootstrap Protocol, 411
- Bootvorgang, 229
- Bourne-Again-Shell, 68
- Bourne-Shell, 67
- brace expansion, 81
- Broadcast
  - Direkter, 388
  - Limitierter, 388
  - Subnetz, 393
- bunzip2, 286
- bus mastering, 360
- BusID, 431
- bzcat, 286
- bzip2, 285
- bzip2recover, 286
- C, 292
- C++, 292
- C-Programme, 292
- C-Shell, 67
- cal, 84
- case, 346
- cat, 78
- cd, 71
- cdparanoia, 369
- cdrdao, 369
- cdrecord, 369
- Chain-Loading, 235
- chgrp, 171
- chmod, 172
- chown, 171
- chpasswd, 148
- chsh, 68
- cksum, 128
- clear, 84
- Cluster, 275
- comm, 129
- compress, 287
- conf.modules, 323
- config.in, 324
- configure, 292
- context switch, 471
- convert, 457
- core dump, 204
- cp, 59, 78
  - Links, 202
- cpio, 282
- cron, 268
- crontab, 268
- ctwm, 427
- cut, 129
- cylinder, 183
- DARPA, 383
- date, 84
- Datei
  - ausführbare, 337
  - gewöhnliche, 197
  - Rechte, 167
- Datenfernübertragung, 362
  - Analog, 362
  - Breitband, 367
  - Digital, 366
- dd, 79
- default, 406
- default route, 406
- DefaultDepth, 432
- depmod, 322
- Deskriptor, 374
- DESTINATION-UNREACHABLE, 417
- Device, 431
- Dezimalsystem, 384
- df, 203
- DHCP, 411
- Diagnosewerkzeuge, 417
- dialog, 348

- diff, 134
- dir, 73
- Disk Quotas, 219
- Display, 432
- Displaymanager, 427
- Displayname, 440
- DisplaySize, 431
- DMA, 360
- dmesg, 231
- dnsdomainname, 416
- do, 351
- DoD, 383
- domainname, 416
- done, 351
- DontVTSwitch, 429
- DontZap, 429
- DontZoom, 429
- Download, 458
- Driver, 430
- DSL, 367
- du, 203
- dumpe2fs, 212
- dynamic executable, 471
- Dynamic Host Configuration Protocol, 411
- e2fsck, 215
- echo, 87
- ECHO-REPLY, 417
- ECHO-REQUEST, 417
- Editor
  - vi, 58
- edquota, 221
- egrep, 131
- ehci-hcd.o, 374
- Eingabe, 101
- elif, 344
- else, 344
- EndSection, 428
- EndSubSection, 432
- env, 97
- expand, 132
- Export
  - X11-Desktop, 473
- export, 96
- false, 147
- FAQ, 114
- Farbcode, 87
- fc, 100
- fdformat, 192
- fdisk, 183
- Fehlercode, 339
- Fenstermanager, 426
- Festplatte, 25
- fg, 259
- fgrep, 131
- FHS, 74
- fi, 344
- FIFO, 198
- file, 83
- File System Standard, 74
- Files, 428
- Filesystem Hierarchy Standard, 74
- find, 216
- finger, 162
- fips, 28
- Flaschenhals, 249, 471
- fmt, 124
- fold, 126
- Font, 434
- FontPath, 428
- fonts.alias, 438
- fonts.dir, 435
- fonts.scale, 437
- Fontserver, 438
- for, 352
- Formatieren
  - fdformat, 192
  - mkfs, 191
  - Skript, 170
- FQDN, 411
- free, 254
- fsck, 215
- FSSTND, 74
- Full Qualified Domain Name, 411
- function, 335
- Funktion, 335
- fvwm, 427
- fvwm2, 427
- fvwm95, 427
- gdm, 443
- Gerätedatei, 198
- GID, 146
- gpasswd, 156
- grep, 130
- groupadd, 156
- groupdel, 157
- groupmode, 156
- groups, 155
- grpconv, 158
- grpquota, 220
- grpunconv, 158
- GRUB, 232
  - Bootdiskette, 234
- grub-install, 234
- Gruppe
  - Anlegen, 35, 156
  - Bearbeiten, 36, 156
  - Löschen, 36, 157
- Gruppenverwaltung, 35
- gunzip, 285
- gzip, 283
- halt, 248
- Harddisk, 25

- Hardware, 355
- head, 126
- Headcrash, 25
- Header-Dateien, 471
- help, 61, 113
- Hexadezimalsystem, 385
- Hilfe, 55, 61
- history, 99
- History-Liste, 99
- HorizSync, 431
- host.conf, 412
- hostname, 415
- hosts, 413
- Hotplug, 373
- HOWTO, 114
- Hunk, 134
- hwclock, 86
- I/O-Ports, 360
- id, 155
- Identifier, 430
- if, 344
- ifconfig, 403
- in, 346
- info, 112
- init, 244
- Init-Daemon, 240
- Inode, 55, 214
- InputDevice, 430
- InputDevices, 429
- INPUTRC, 91
- inputrc, 91
- insmod, 319
- insservinsserv, 246
- Installation
  - Pakete, 32
- Interface, 403
- Interrupt, 230, 360
- Interrupt-Vektor-Tabelle, 230
- IP, 381
- IP-Forwarding, 410
- IRQ, 360
- isapnp, 361
- ISDN, 366
- isserial, 366
- Job, 258, 265
- jobs, 258
- Jobverwaltung, 267
- join, 130
- Jokerzeichen, 80
- KAB, 45
- KBear, 45
- KCalc, 45
- KControl, 42
- KDE, 41
  - Kontrollzentrum, 42
  - Systemüberwachung, 42
- kdesu, 44
- KDE su, 44
- kdm, 442
- KEdit, 44
- Kernel
  - Kompilieren, 324
  - Konfiguration, 325
  - Quellen, 325
- Kernel-Routing-Tabelle, 406
- Kernelmodule, 317
  - Dateien, 317
- Keycode, 459
- Keycode-Modus, 459
- Keymap, 459
- kill, 260
- killall, 260
- KJots, 45
- Klammerexpansion, 81
- KMail, 44
- KNode, 45, 116
- KNotes, 45
- KOffice, 45
- Kommandogruppierung, 95
- Kompression, 283
- Konquerer, 44
- konsole, 51
- Kontextwechsel, 471
- KOrganizer, 45
- KSnapshot, 44
- KSpread, 45
- KSysGuard, 42
- KWord, 45
- KWrite, 44
- last, 163
- lastlog, 163
- ld-linux.so, 295
- ld.so, 295
- LD\_LIBRARY\_PATH, 295
- ldconfig, 296
- ldd, 295
- less, 82
- lesskey, 83
- libc, 411
- LILO, 235
- lilo, 236
- Links, 198, 200
  - harte, 200
  - symbolische, 201
- Linux-Konsole, 49
- ln, 201
- Load, 430
- Load Average, 249
- localhost, 389
- locate, 218
- locatedb, 218

- Logdateien, 270
- logname, 162
- logout, 87
- logrotate, 273
- Loop-Back, 389
- Loopback-Interface, 406
- LPI, 479
- LPIC, 479
- LPIC1, 479
- ls, 51, 71
- lsmod, 318
- lspci, 358
- lsusb, 376
- LUN, 370
- Magic Cookie, 445
- Magische Nummer, 186
- make, 293
  - bzImage, 328
  - clean, 327
  - cloneconfig, 326
  - config, 326
  - dep, 327
  - menuconfig, 326
  - modules, 328
  - modules\_install, 328
  - oldconfig, 326
  - xconfig, 326
  - zdisk, 327
  - zImage, 327
- Makefile, 293
- man, 62, 111
- Man-Pages, 109
- mandb, 118
- MANPATH, 110
- manpath, 112
- Manual-Pages, 62, 109
- Master Boot Record, 185
- MBR, 185
- Memory-Pages, 192
- mini-HOWTO, 114
- Mini-Startprogramm, 185
- MIT, 425
- mkdir, 57, 73
- mkfifo, 198
- mkfontdir, 437
- mkfontscale, 437
- mkfs, 191
- mkswap, 196
- Modeline, 431
- ModelName, 431
- Modem, 362
- Modes, 431
- modinfo, 320
- MODPATH, 319
- modprobe, 321
- Module, 430
- ModulePath, 429
- modules, 319
- modules.conf, 323
- modules.dep, 321
- modules.usbmap, 377
- Monitor, 431
- more, 82
- mount, 187
- Mount Point, 187
- Mounten, 187
- mt, 369
- mv, 60, 79
- Name Service Switch, 412
- Named Pipe, 198
- Namensauflösung, 411
- netstat, 420
- networks, 413
- Netze
  - Private, 389
- Netzwerk, 403, schnittstellen403
- Netzwerk-Sniffer, 422
- Netzwerkadresse, 388
- Netzwerkklassen, 386
- newgrp, 155
- Newsgroups, 116
- Newsserver, 116
- nice, 257
- nl, 123
- nohup, 260
- nologin, 147
- NSS, 412
- nsswitch.conf, 412
- od, 124
- Offline-Reader, 458
- Offsetregister, 229
- OHCI, 373
- Oktalsystem, 385
- Online-Update, 33
- Open Publication License, 489
- Open Sound System, 367
- Option, 429
- OSS, 367
- Paging, 192
- Paket-Manager, 298
- Pakete
  - RPM, 297
- Partition, 25
  - Erweiterte, 25
  - Primäre, 25
  - Swap, 28, 192
- Partitionstabelle, 185
- passwd
  - Befehl, 148
  - Datei, 144
- paste, 130

- Patch, 472
- patch, 135
- Patchen, 135
- PATH, 97
- pci.ids, 358
- ping, 417
- Plug-and-Play, 361
- pnpdump, 362
- POST, 229
- power good, 229
- Power On Self Test, 229
- pr, 125
- Prefix, 472
- printenv, 97
- procinfo, 254
- Programmdokumentation, 115
- Promiscuous Modus, 422
- Prompt, 50, 98
- Prozeß
  - Überwachung, 249
  - Administration, 257
  - Tools, 250
  - Verwaltung, 248
- ps, 250
- PS1, 98
- pstree, 252
- pwconv, 158
- pwd, 51, 70
- pwunconv, 158
- Quellcode, 291
- quota, 220
- quota.group, 220
- quota.h, 222
- quota.user, 220
- quotacheck, 222
- quotaoff, 221
- quotaon, 220
- quotastats, 222
- Rückgabewert, 339
- rc.serial, 365
- read, 352
- Readline Library, 91
- real mode, 229
- Realer Modus, 229
- reboot, 248
- Rechte, 167
- remount, 189
- renice, 257
- repquota, 222
- Request for Comments, 381
- resolv.conf, 414
- Resolver, 411
- RFC, 381
- RFC-Editor, 382
- RgbPath, 429
- rm, 57, 80
- rmdir, 57, 74
- rmmmod, 319
- root, 74, 143
- Route, 406
- route, 406
- Routencache, 407
- Router, 409
- RPM, 297
- rpm, 298
  - Überprüfung, 314
  - Deinstallation, 305
  - Enfernen von Paketen, 305
  - Informationen, 306
  - Installation, 299
  - Upgrade, 299
- RPM Package Manager, 297
- RPM-Manager, 298
- RPM-Pakete, 297
- RS-232, 362
- Runlevel, 240, 242, 248
- runlevel, 244
- Runnable, 248
- Running, 248
- Runtime Linker, 295
- SANE, 369
- SASI, 367
- Save Text Bit, 170
- sax, 433
- sax2, 433
- Scancode, 459
- Scancode-Modus, 459
- Schrift, 434
- Screen, 431, 432
- script, 88
- Scripting, 333
- SCSI, 367
- SCSI-Adressierung, 369
- SCSI-Subsystem, 368
- scsi\_info, 371
- scsiinfo, 371
- SD, 369
- sd\_mod.o, 369
- Section, 428
- sed, 132
- Segmentregister, 229
- ServerFlags, 429
- ServerLayout, 432
- setserial, 364
- SG, 369
- sh, 161
- Shared Libraries, 295
- Shell, 49
  - Login-, 68
- Shell-Scripting, 333
- shopt, 353
- showkey, 459



- Shugart, 367
- shutdown, 247
- SIGHUP, 260
- Sockets, 198
- sort, 128
- source, 354
- split, 127
- Spur, 183
- SR, 369
- sr\_mod.o, 369
- SSH-X11-Weiterleitung, 446
- ST, 369
- st.o, 369
- Standardroute, 406
- Standby-Server, 275
- startx, 441
- stat, 200
- static executable, 472
- Sticky Bit, 170
- stty, 92
- su, 143
- subnetmask, 391
- Subnetting, 389
- Subnetzmaske, 391
- SubSection, 432
- Suchpfad, 97
- sudo, 173
- Suffix, 472
- sum, 128
- Superblock, 212
- Superuser, 143
- Swap-Partition, 192
- Swap-Space, 193
- swapoff, 197
- swapon, 196
- sync, 189
- syslogd, 270
- Systemlastüberwachung, 249
- 
- tac, 123
- tail, 126
- tar, 279
- tar-Archiv, 279
  - Anlegen, 279
  - Entpacken, 281
- Tarball, 281, 291
- TCP, 381
- TCP/IP, 381
- tcpdump, 422
- tee, 102
- telinit, 245
- telnet, 88, 421
- test, 342
- TexInfo, 112
- then, 344
- tilde expansion, 70
- Tilde-Ausdehnung, 70
- 
- top, 252
- touch, 56, 77
- tr, 131
- traceroutetraceroute, 419
- track, 183
- Transaction Translator, 373
- true, 147
- ttf, 437
- tty, 88
- tune2fs, 213
- twm, 426
- type, 118
- 
- UHCI, 373
- uhci.o, 374
- UID, 34, 143, 145
- umask, 172
- Umgebungsvariablen, 97
- Umleitung, 100
- umount, 189
- unalias, 99
- uname, 317
- uncompress, 288
- unexpand, 132
- uniq, 129
- Universal Serial Bus, 372
- until, 352
- Update
  - Online, 33
- updatedb, 219
- Upgrade, 299
- uptime, 254
- USB, 372
- usb-ohci.o, 374
- usb-uhci.o, 374
- usbcore.o, 374
- usbdevfs, 374
- usbfs, 374
- usbmodules, 377
- UseModes, 431
- Usenet, 116
- useradd, 151
- userdel, 153
- usermod, 152
- usrquota, 220
- 
- Variablen, 95
- vdir, 73
- VendorName, 431
- VertRefresh, 431
- Verzeichnis, 197
  - Rechte, 168
  - SGID, 170
- Verzeichnisbaum, 74
- vi, 58
- vim, 58
- vimtutor, 58
- visudo, 175

vmlinux, 230  
vncpasswd, 474  
vncserver, 474  
vncviewer, 475

w, 162  
Waiting, 248  
wc, 127  
wget, 458  
whatis, 118  
whereis, 117  
which, 118  
while, 351  
who, 161  
whoami, 161  
Wildcards, 80  
Window Maker, 427  
Window Manager, 426  
WinModem, 363  
wm2, 427  
Wurzel, 74

X, 425  
X, 441  
X-Client, 425  
X-Protokoll, 426  
X-Server, 425  
X-Terminal, 50  
X-Window, 425  
X-Window-System, 41, 425  
X11, 425  
xargs, 103  
xauth, 445  
XAUTHORITY, 446  
xdm, 442  
XDMCP, 447  
XF86Config, 428  
xf86config, 432  
XF86Setup, 433  
xfontsel, 437  
XFree86, 425  
XFree86-VidModExtension, 433  
xfs, 438  
xhost, 443  
xinit, 441  
xinit, 441  
XkbDisable, 429  
XLFD, 435  
xrdb, 450  
Xresources, 442  
Xservers, 442  
Xsession, 442  
xset, 433  
Xsetup, 442  
xterm, 50  
xvidtune, 433  
Xvnc, 473

YaST, 30  
    Online-Update, 33  
    Paket-Installation, 32  
YOU, 33

Zahlensysteme, 384  
zcat, 285  
Zombie, 248  
Zylinder, 183  
Zylinder-1024-Problem, 356