

# Linux Basiswissen

Wolfgang Karall

[karall@mail.zserv.tuwien.ac.at](mailto:karall@mail.zserv.tuwien.ac.at)

2.-4. April 2001

## **Abstract**

Linux als professionelle Alternative zu herkömmlichen Betriebssystemen kennenlernen und nutzen.

# Linux? Linux!<sup>1</sup>

- Stabilität, Sicherheit, Performance
- Flexibilität, Erweiterbarkeit und Kompatibilität
- freier Sourcecode (GNU<sup>2</sup> General Public License)
- keine Lizenzkosten, frei erhältlich
- ständige Weiterentwicklung
- große Auswahl an freier Software

---

<sup>1</sup>Linux Online, <http://www.linux.org/>

<sup>2</sup>GNU Project, GNU's Not Unix, <http://www.gnu.org/>

# Was ist Linux?

Linux ist ein UNIX<sup>3</sup>-artiges Betriebssystem.

- Multi-User und Multi-Tasking
- 32/64 Bit Architektur, Multi-Plattform
- SMP (Symmetric Multiprocessing)
- POSIX kompatibel
- Netzwerkfähigkeit

---

<sup>3</sup>The Open Group, <http://www.opengroup.org/>

## Linux - Einige historische Fakten

- ursprünglicher Autor: Linus Benedict Torvalds (Finnland)
- erste Veröffentlichung von Linux im Internet im Oktober 1991 (Version 0.02)
- Version 1.0 im Jahre 1994
- aktuelle Version<sup>4</sup>: 2.4.3 (Stand: 31.3.2001)

---

<sup>4</sup>The Linux Kernel Archives, <http://www.kernel.org/>

# Linux Distributionen

- Zusammenstellung von Kernel, Dienstprogrammen, graphischer Benutzeroberfläche und einer Auswahl von Anwendungen
- Eigene Erweiterungen und Konfigurations-Software, Vorkonfiguration des Systems
- Teilweise zusätzliche gedruckte Dokumentation und Support-Dienstleistungen
- Distributionen für unterschiedliche Zwecke und Anwender, kommerzielle und freie Distributionen

## Einige bekannte Distributionen

- SuSE Linux <http://www.suse.de/>
- RedHat Linux <http://www.redhat.com/>
- Debian GNU/Linux <http://www.debian.org/>
- Slackware Linux <http://www.slackware.com/>
- Linux-Mandrake <http://www.linux-mandrake.com/>
- MkLinux <http://www.mklinux.org/>

# Komponenten von Linux bzw. GNU/Linux

Ein Linux System setzt sich zusammen aus:

- Kernel: Geräte-Treiber, Dateisysteme, Netzwerk
- Unix System- und Dienstprogramme: Kommandozeileninterpreter (Shell), Tools für Datei- und System-Operationen, Online-Dokumentation usw. (hauptsächlich GNU-Versionen)
- Graphische Benutzeroberfläche: X Window System (optional)
- Anwendungen: Server-Software, Browser, Office, etc.

# Installation

- Booten von Floppy, Festplatte, CD-ROM oder DVD in ein Basis-Linux-System
- erste Konfiguration des Systems (Sprache, Tastatur, Maus, etc.)
- Paßwort des Benutzers root (Administrator) angeben
- (optional, aber empfehlenswert) normalen Benutzer anlegen
- Partitionieren der Festplatte und Anlegen der Filesysteme
- Installation von Linux auf die Festplatte
- (optional) Installation eines Boot-Managers



## Wichtige Unterschiede Linux - Windows

- Binär-Format (Programme nicht austauschbar)
- Keine Laufwerksbuchstaben sondern ein einziger Verzeichnisbaum
- Pfadtrenner ' / ' im Gegensatz zu ' \ '
- Groß- und Kleinschreibung wird unterschieden, start  $\neq$  Start
- Linux auch ohne graphische Benutzeroberfläche (engl. Graphical User Interface, GUI) nutzbar, strikte Trennung von System und GUI
- Wirkliche Multi-User Funktionalität, mehrere Benutzer können gleichzeitig auf dem Rechner arbeiten

## Online-Dokumentation

- Unix Manuals, sortiert nach Sektionen ( `man(1)` bedeutet zum Beispiel Befehl `man` in Sektion 1)
  - anzeigen mit `man [<Sektion>] <Befehl>`
  - suchen in den Manual-Beschreibungen mit `apropos <Suchbegriff>` bzw. `man -k <Suchbegriff>`
- Info Dokumentation, anzeigen mit `info <Befehl>`
- `<Befehl> --help` oder `<Befehl> -h`
- Dokumentation in Textdateien, anzeigen mit `less <Textdatei>`
  - HOWTO-Dateien (bei SuSE in `/usr/share/doc/howto/` )
  - Weitere Dokumentation zu einzelnen Paketen (bei SuSE in `/usr/share/doc/packages/<Paketname>/`)

# Benutzer und Gruppen

- Benutzer eindeutig identifizierbar durch Login-Namen, eindeutig authentifizierbar durch Login-Namen und Passwort
- Benutzer können in Gruppen zusammengefaßt werden
- Benutzer können gleichzeitig in mehreren Gruppen sein
- Zugriff auf Ressourcen je nach Benutzer/Gruppe steuerbar
- Programme laufen mit Benutzer- bzw. Gruppen-Berechtigung

# Virtuelle Konsolen und Remote-Login

- Virtuelle Konsolen (engl. Virtual Terminals, VTs)
  - ermöglichen mehrmaliges lokales Einloggen am Rechner
  - Umschalten mit `ALT-<F1-F6>` bzw. `STRG-ALT-<F1-F6>` aus dem Graphikmodus, mit `ALT-F7` zurück
- Remote-Login
  - einloggen von einem entfernten Rechner
  - Nutzung sowohl textbasierter als auch graphischer Programme

## Graphische Benutzeroberfläche: X

- portables, netzwerkfähiges Window System, auf UNIX-Rechnern ein Standard
- XFree86<sup>5</sup> Version 4 ist eine freie Implementation des X Window System<sup>6</sup> (oft auch nur X oder X11 genannt)
- darauf aufsetzend arbeitet ein Window Manager und (optional) ein Desktop Environment (z.B. CDE, KDE, Gnome)
- mehrere, komplett unterschiedliche GUIs gleichzeitig nutzbar
- Konfiguration in `/etc/X11`

---

<sup>5</sup>The XFree86 Project, Inc., <http://www.xfree86.org/>

<sup>6</sup>The X Consortium, <http://www.x.org/>

# Die Shell - Der Kommandozeilen-Interpreter

- Schnittstelle zum Betriebssystem
- führt binäre Programme und Skripte aus (Befehlsinterpreter)
- die gebräuchlichsten Linux-Shells:
  - `bash` (GNU Bourne-Again SHell)
  - `tcsh` (TENEX C SHell)
  - `ksh` (Korn SHell)
- Shell  $\neq$  Konsole, z.B. in einem `xterm` oder einem anderen Terminalemulator im X Window System

## Wichtige Tastenkombinationen

- in der Shell:
  - STRG-C : Abbrechen des laufenden Programms
  - STRG-Z : Programm wird in den Hintergrund geschickt und dort angehalten
  - STRG-L : Neuzeichnen des Bildschirms
  - STRG-D : Senden des Steuerzeichens EOF (End Of File)
- im X Window System:
  - STRG-ALT-BACKSPACE : Abbrechen des X Window System
- in der Konsole:
  - STRG-ALT-ENTF : Neustart des Systems

# Befehle (1)

Befehl [<Optionen>] [<Argumente>]

<code>ls</code>	listet Dateien und Verzeichnisse auf
<code>cd</code>	wechselt in ein Verzeichnis
<code>cp</code>	Kopieren einer Datei
<code>mv</code>	Verschieben einer Datei
<code>rm</code>	Löschen einer Datei
<code>mkdir</code>	Anlegen eines Verzeichnisses
<code>rmdir</code>	Löschen eines Verzeichnisses
<code>pwd</code>	Anzeigen des aktuellen Arbeitsverzeichnisses
<code>less</code>	Anzeigen einer Datei
<code>grep</code>	Durchsuchen von Dateien
<code>find</code>	Finden von Dateien



## Befehle (2)

- Befehl `>` Datei (Standard-Ausgabe `stdout`)
- Befehl `2>` Datei (Standard-Error-Ausgabe `stderr`)
- Befehl `&>` Datei (sowohl `stdout` als auch `stderr`)
- Befehl `>>` Datei (Anhängen von `stdout` an die Datei)
- Befehl `<` Datei (Lesen der Datei über die Standard-Eingabe `stdin`)
- Befehl1 `|` Befehl2 (`stdout` von Befehl1 wird auf `stdin` von Befehl2 umgeleitet)

# vi (1)

- der klassische UNIX Editor
- 3 Modi
  - Eingabemodus
  - Befehlsmodus
  - "Last-Line"-Modus
- Beenden von vi
  - ESC ⇒ Befehlsmodus
  - ':' ⇒ "Last-Line"-Modus
  - ':q!' ⇒ Beenden von vi , Änderungen werden verworfen

## vi (2)

i	in den Eingabemodus
a	wie i, Cursor nach der aktuellen Position
A	wie i, Cursor am Ende der Zeile
R	wie i, Zeichen werden überschrieben
x	Löscht das Zeichen unter dem Cursor
dd	Löscht die aktuelle Zeile
J	die folgende Zeile wird an die aktuelle angehängt
u	nimmt letzten Befehl / letzte Eingabe zurück
.	wiederholt den letzten Befehl
:w <Datei>	schreibt den Text nach <Datei>
:e <Datei>	lädt die <Datei> zum Editieren
:x	speichert den Text und beendet vi

# Der Boot-Vorgang (1)

- BIOS (Basic Input Output System)
  - Initialisierung der Hardware
  - Auswahl des Boot-Mediums (Festplatte, CD-ROM, Floppy)
- Laden des Kernels, eventuell mit Übergabe von Parametern
  - Direkt von Floppy (ohne Parameter)
  - Mittels Boot-Loader, z.B. LILO
  - Mittels `loadlin` aus DOS heraus

## LILO - Der Linux Loader

- Installation in Master Boot Record (MBR) oder Bootsektor einer Partition
- Booten von verschiedenen Kernelversionen und anderen Betriebssystemen
  - Laden und starten eines Kernels
  - Laden und starten des Bootsektors einer Partition
- Übergabe von Kernel-Parametern
- (optional) Passwort-Schutz
- Konfiguration in `/etc/lilo.conf`
- nach jeder Änderung in der Konfiguration `/sbin/lilo` aufrufen!

## Ein Beispiel einer /etc/lilo.conf

```
boot=/dev/hda          # LILO Installationsziel
prompt                # Eingabeaufforderung zeigen
timeout=50           # 5sec Auszeit fuer den Prompt
message=/boot/welcome # Textdatei vor dem LILO Prompt
append="mem=192M"    # Kernelparameter
vga=normal           # Textmodus 80x25 Zeichen

image=/vmlinuz        # Linux Kernel-Image
    label=Linux       # Label im Bootmenu
    root=/dev/hdb1   # Root-Partition

other=/dev/hda1      # "C: Platte"
    label=Windows
```

## Booten mit `loadlin`

- Starten von Linux aus DOS oder Windows heraus
- Kernel (und optional `initrd`, "initial ramdisk") muß auf eine Partition kopiert werden auf die DOS bzw. Windows Zugriff haben

- Aufruf von `loadlin` zum Beispiel durch:

```
loadlin C:\linux\kernel\vmlinuz-2.2.18 root=/dev/hdb1 ro
```

- Einbindung in ein DOS-Bootmenü oder Aufruf über ein Icon vom Windows-Desktop aus möglich

## Der Boot-Vorgang (2)

- Konsole für die Ausgabe wird geöffnet, entweder Bildschirm (default) oder serielle Schnittstelle (über Kernelparameter)
- (optional) `initrd` wird geladen
- Kernel lädt das Root-Filesystem /
- Kernel startet den Prozeß `init`



## Der Prozeß `init` und die Runlevel 0 bis 6

- `init` started alle anderen Prozesse  $\Rightarrow$  `init` ist damit Parent-Prozeß aller Prozesse
- Konfiguration von `init` in `/etc/inittab`
- Verschiedene Runlevel für verschiedene Betriebszustände
  - mit/ohne Netzwerk
  - mit/ohne graphischem Login
  - Multi-User bzw. Single-User (1)
  - Shutdown (0) und Reboot (6)
- Wechseln des Runlevels mit `telinit <neuer Runlevel>`

## Die Datei `/etc/inittab`

Wichtige Einträge:

- `id:5:initdefault:`  
Runlevel, der per default geladen wird
- `15:5:wait:/etc/init.d/rc 5`  
`init` startet im Runlevel 5 diese Kommandozeile und wartet bis zum Ende dieses Prozesses
- `ca::ctrlaltdel:/sbin/shutdown -r -t 4 now`  
Kommando welches bei Drücken von STRG-ALT-ENTF ausgeführt wird.
- `4:2345:respawn:/sbin/mingetty tty4`  
In Runlevel 2, 3, 4 oder 5 wird `/sbin/mingetty` für die Konsole `tty4` immer wieder neu gestartet. *Wichtig:* Die Zahl am Anfang und Ende muß hier gleich sein!

## Der Boot-Vorgang (3)

- `/etc/init.d/rc #` für Runlevel # wird gestartet  $\Rightarrow$ 
  - Ausführen der `/etc/rc.d/rc#.d/K*` Skripte, Parameter `stop`
  - Ausführen der `/etc/rc.d/rc#.d/S*` Skripte, Parameter `start`(eventuell auch `/etc/rc#.d/K*` bzw. `/etc/rc#.d/S*` )
- Login über `xdm` im X Window System bzw. `getty` auf einer der virtuellen Konsolen

# Wiederholung

- Einführung in Linux
  - Linux Distributionen
  - Eigenschaften und Komponenten eines Linux Systems
  - Unterschiede Linux - Windows
  - Online-Dokumentation
- Installation von SuSE 7.1 Professional
- Erste Befehle und `vi`
- Boot-Vorgang
  - LILO und `loadlin`
  - `init` und die Runlevel ( `/etc/inittab` , `/etc/rc.d/rc#.d/` )

## Reguläre Ausdrücke mit egrep

.	1 Zeichen
^	Zeilenanfang
\$	Zeilenende
[abc]	eines der Zeichen a, b oder c
[^abc]	jedes Zeichen außer a, b oder c

?	Wiederholung 0 oder 1 mal
*	Wiederholung 0 oder mehrmals
+	Wiederholung 1 oder mehrmals
{x}	Wiederholung genau x mal
{x,}	Wiederholung x oder mehrmals
{x,y}	Wiederholung min. x, max y mal
(A B)	Ausdruck A oder B

# Das Filesystem<sup>7</sup> von Linux

- nur ein Verzeichnisbaum
- andere Filesysteme können in den Verzeichnisbaum ein- und ausgehängt werden
  - Festplatten-Partitionen
  - Disketten, CD-ROM, DVD, etc.
  - Netzwerk-Filesysteme
- Zugriffsrechte

---

<sup>7</sup>Filesystem Hierarchy Standard, <http://www.pathname.com/fhs/>

## Spezielle Verzeichniseinträge und -bezeichnungen

- ' . ': das aktuelle Verzeichnis (bei Linux im Allg. nicht im Pfad)
- ' .. ': das Verzeichnis über dem aktuellen Verzeichnis
- ' ~ ': das Benutzerverzeichnis des jeweiligen Benutzers
- ' / ': das Root-Verzeichnis
- versteckte Dateien beginnen mit einem Punkt '.', z.B. .profile

## Das root-Filesystem

<code>/bin</code>	Wichtige Programme für alle Benutzer
<code>/boot</code>	Statische Dateien für den Systemstart
<code>/dev</code>	Geräte-dateien
<code>/etc</code>	Konfigurationsdateien
<code>/home</code>	Benutzerverzeichnisse
<code>/lib</code>	Wichtige System-Bibliotheken und Kernel-Module
<code>/mnt</code>	für temporäre Filesysteme
<code>/opt</code>	Optionale Software-Pakete
<code>/proc</code>	Dynamisches Pseudo-Filesystem des Linux Kernels
<code>/root</code>	Verzeichnis des Benutzers root
<code>/sbin</code>	Wichtige System-Programme
<code>/tmp</code>	Temporäre Dateien
<code>/usr</code>	Programme und Bibliotheken
<code>/var</code>	Variable Systemdaten



# Arbeiten mit Filesystemen

- `mount` und `umount` zum Ein- respektive Aushängen
- `/etc/fstab` zur Konfiguration
  - Device
  - Mount-Point
  - Filesystem-Typ
  - Optionen
- Anzeige der aktuell eingehängten Filesysteme in `/etc/mtab` bzw. mittels des Befehls `mount` ohne Argumente
- Vorsicht: Disketten erst nach einem `umount` auswerfen

# Filesystem-Typen

## Filesysteme auf Datenträgern

ext2	Standard von Linux
reiserfs	neues Journaling Filesystem für Linux
msdos	MS-DOS, keine langen Dateinamen
vfat	Windows9x, lange Dateinamen
ntfs	WindowsNT und Windows2000
iso9660	CD-ROMs
udf	DVDs

## Netzwerk-Filesysteme

nfs	Network File System
smbfs	Samba File System

## Die /usr Hierarchie

- kann mit anderen Rechnern geteilt verwendet werden
- Inhalt ähnlich dem von /
  - X11R6 - X Window System
  - games - Spiele, Edutainment
  - include - Header-Dateien
  - local - lokal installierte Software
  - share - Architektur-unabhängige Dateien
  - src - Sourcecode

## Die /var Hierarchie

- variable Daten
- kann nur teilweise mit anderen Rechnern geteilt verwendet werden
- wichtige Verzeichnisse:
  - lock - Lock-Dateien
  - log - Log-Dateien
  - mail - Briefkästen der Benutzer
  - run - Daten von Prozessen
  - spool - Warteschlangen

# Die Zugriffsrechte (1)

- Dateien und Verzeichnisse gehören einem Benutzer und einer Gruppe
- Rechte für den Benutzer, die Gruppe und die "Anderen" einstellbar
  - lesen
  - schreiben
  - ausführen

<code>ls -l</code>	Anzeigen der Rechte
<code>chown</code>	Benutzer ändern (Change Owner)
<code>chgrp</code>	Gruppe ändern (Change Group)
<code>chmod</code>	Rechte ändern (Change Mode)

## Die Zugriffsrechte (2)

```

-rw-r--r--  1 root    root    35036 Mar 23 07:21 file
  |          |      |
  Rechte    Benutzer Gruppe      Datei

```

Typ	Benutzer			Gruppe			"Andere"		
-	r	w	-	r	-	-	r	-	-

4	2	1	4	2	1	4	2	1
---	---	---	---	---	---	---	---	---

- Typ: '-' für Dateien, 'd' für Verzeichnisse
- rwx: Rechte fürs Lesen ('r'), Schreiben ('w') und Ausführen ('x'), '-' bedeutet das Fehlen dieses Rechtes

# Links

- Symbolische Links mit `ln -s`
  - verweisen auf eine andere Datei oder ein anderes Verzeichnis
  - Ziel muß nicht existieren
  - Ziel kann auf einem anderen Filesystem liegen
  - beim Löschen wird nur der Link, nicht die Datei gelöscht
- Hardlinks mit `ln`
  - Ziel kann kein Verzeichnis sein
  - nur innerhalb eines Filesystems
  - sind verschiebbar innerhalb dieser Grenzen
  - Datei bleibt solange bestehen, solange ein Hardlink darauf existiert
- Link und Ziel haben die gleichen Rechte

## Gerätedateien (engl. Devices)

- Linux spricht Geräte über spezielle Dateien im Filesystem an
- Gerätedateien befinden sich in `/dev`
- Block-Devices (Festplatten, Floppy, Ram-Disk, ...)
- Character-Devices (Maus, Drucker, ...)



## Die wichtigsten Devices<sup>8</sup>

<code>/dev/hd[a-d]</code>	Festplatten und CD-ROM am IDE-Bus
<code>/dev/hda?</code>	Partitionen der 1. Platte am 1. IDE-Kanal
<code>/dev/sd?</code>	SCSI Festplatten
<code>/dev/sr?</code>	SCSI CD-ROM
<code>/dev/fd?</code>	Disketten-Laufwerke
<code>/dev/lp?</code>	Parallele Schnittstellen (Drucker)
<code>/dev/ttyS?</code>	Serielle Schnittstellen (Maus, Modem)
<code>/dev/psaux</code>	PS/2 Maus
<code>/dev/dsp</code>	Digital Signal Processor der Soundkarte
<code>/dev/mixer</code>	Mixer der Soundkarte
<code>/dev/null</code>	Digitaler Mistkübel

---

<sup>8</sup>siehe auch `Documentation/devices.txt` im Sourcecode des Kernels

## Die Partitionierung der Festplatte(n)

- nur / wirklich notwendig, Swap-Partition und /boot aber empfehlenswert
- mehrere Swap-Partitionen auf mehreren Festplatten erhöhen die Geschwindigkeit
- Partitionierung trotzdem zu empfehlen
  - /boot auf eine kleine Partition am Beginn der Platte
  - /usr und /opt auf eigene Partitionen oder auch Mounten via nfs
  - /tmp aus Sicherheitsgründen (kein Auffüllen der ganzen / -Partition durch einen böswilligen Benutzer oder Prozeß)

# Die Benutzerverwaltung (1)

- Benutzer
- Gruppen
- Shells
- Benutzer-Verzeichnis
- numerische Benutzerkennung (User ID, UID)
- numerische Gruppenkennung (Group ID, GID)
- Ablaufdatum

## Der Administrator-Account `root`

- Zugriff auf *alle* Dateien und Ressourcen des Systems
  - Anlegen und Entfernen von Benutzern
  - Installation von Software ("Mounten" von CD-ROMs, Disketten und anderen Filesystemen)
  - Konfiguration des Systems
  - Anhalten und Neustart des Systems
- temporärer Wechsel zum `root`-Account mit dem Befehl `su`

Nur für obige Tätigkeiten verwenden, nicht zum täglichen Arbeiten!

## /etc/passwd

Ein Zeile entspricht einem Account, Felder mit : getrennt

```
spiney:x:1000:1000:Wolfgang Karall:/home/spiney:/bin/bash
```

- Login Name
- verschlüsseltes Passwort, meist aber in /etc/shadow
- UID
- GID
- Kommentarfeld
- Benutzerverzeichnis
- Shell

## `/etc/shadow` **und** `/etc/group`

- `/etc/shadow` : nur von root lesbar und schreibbar
  - Login Name
  - verschlüsseltes Passwort
  - 6 Felder mit Zeitinformation
- `/etc/group` :
  - Name der Gruppe
  - Passwort
  - numerische Gruppenkennung (GID)
  - Aufzählung der Mitglieder der Gruppe

## Die Benutzerverwaltung (2)

- Konfigurationstool der Distribution (empfohlen)
- manuell `/etc/passwd` , `/etc/shadow` und `/etc/group` editieren (sehr riskant)
- `useradd` , `userdel` und `usermod` (praktisch bei Anlegen vieler Benutzer, z.B. mit Hilfe eines Skripts)

# Wiederholung

- Reguläre Ausdrücke mit `egrep`
- Filesysteme
  - Wichtige Verzeichnis-Hierarchien
  - Arbeiten mit Filesystemen, Filesystem-Typen, `/etc/fstab`
  - Zugriffsrechte, `chmod`, `chown` und `chgrp`
  - Links mit `ln`
- Gerätedateien in `/dev`
- Benutzerverwaltung, `/etc/passwd`, `/etc/shadow` und `/etc/group`



# Prozesse

Die wichtigsten Attribute eines Prozesses

- Prozeßnummer (PID)
- Eltern-Prozeß (PPID)
- Benutzer- (UID) und Gruppenkennung (GID)
- Aufrufparameter und Environment

# Prozeßhierarchie

Anzeigen mit `ps tree`

```
startx---xinit-+-XFree86
              '-sh-+-enlightenment-+-E-Clock.epplet
                |                       |-2*[E-Cpu.epplet]
                |                       |-E-MemWatch.eppl
                |                       |-E-Mountbox.eppl
                |                       |-2*[E-NetGraph.eppl]
                |                       '-wmmixer
                |-panel
                |-rt
                |-ssh-agent
                '-xscreensaver
```

## Prozeßstatus und -termination

Anzeigen des Status mit `ps` , Statuscodes:

R Running  
S Sleeping  
T Stopped  
Z Zombie

Terminieren von Prozessen mit:

`kill [<Signal>] <PID>`

SIGHUP	1	Hangup
SIGINT	2	Interrupt
SIGKILL	9	Kill
SIGTERM	15	Terminate

## Vorder- und Hintergrund-Prozesse

- Prozeß in den Hintergrund mit:
  - `<Befehl> ⇒ STRG-Z`
  - `<Befehl> &`
- `jobs` - Anzeigen der laufenden Prozesse
- `fg` - Prozeß in den Vordergrund
- `bg` - Prozeß in den Hintergrund
- `nohup <Befehl>` - Prozeß wird durch `SIGHUP` nicht mehr beendet

# Netzwerke mit Linux

Grundkonfiguration durch:

- Rechner- und Domain-Name
- IP-Adresse und Netzmaske
- Gateway
- Netzwerk und Broadcast
- Nameserver

# Manuelle Konfiguration des Netzwerks

- `/sbin/ifconfig`
- `/sbin/route`
- `/etc/hosts` und `/etc/networks`
- `/etc/nsswitch.conf` und `/etc/host.conf`
- `/etc/resolv.conf`

# Der Linux Kernel

- Sourcecode normalerweise in `/usr/src/linux`
- Konfiguration `make config` bzw. `make menuconfig`
- Neu-Übersetzen durch `make dep && make clean && make zImage`  
(eventuell `bzImage` )
- Kopieren von `/usr/src/linux/arch/i386/boot/zImage` (bzw.  
`/usr/src/linux/arch/i386/boot/bzImage` ) nach `/boot` , eventuell umbenennen
- `/etc/lilo.conf` anpassen und LILO durch Aufruf von `/sbin/lilo` neu installieren

# Die Kernel Module

- Neu-Übersetzen:
  - mit `make modules` übersetzen
  - mit `make modules_install` nach `/lib/modules/<Kernel-Version>/` installieren
- Verwenden:
  - `lsmod` - Anzeigen geladener Module
  - `modprobe` - Laden von Modulen
  - `rmmod` - Entfernen von Modulen aus dem Speicher
  - `/etc/modules.conf`



# Software installieren

- Distributions-abhängige Packages verwenden
- Sourcecode übersetzen:
  - `tar xvfz software.tar.gz` bzw. `tar xvfj software.tar.bz2`
  - `cd software`
  - README oder INSTALL lesen
  - `./configure`
  - `make`
  - `make install` als root