

# Linux-Training Teil 1: Konzepte

Johannes Franken  
<jfranken@jfranken.de>

# Inhalt

1. Grundlagen
  - a) Hardware-Architekturen
  - b) x86-Hardware
  - c) Betriebssysteme
  - d) Der Kernel
2. Filesysteme
  - a) Das "Virtual Filesystem" (VFS)
  - b) Block-Devices
  - c) Filesystem-Typen
  - d) Mounten
  - e) Filesystem-Elemente
  - f) Meta-Informationen
  - g) Der Filesystem Hierarchy Standard (FHS)
3. Der Bootvorgang
  - a) Grober Ablauf
  - b) Runlevel
  - c) init
  - d) init.d-Scripts
4. Zugriffsrechte
  - a) User
  - b) Gruppen
  - c) Filesystem-Permissions
    - i) Konzept
    - ii) Filesystem-Permissions anzeigen (ls)
    - iii) Nummerische Darstellung
    - iv) Permissions ändern (chmod)
    - v) Eigentümer und Gruppe ändern (chown, chgrp)
5. Konsolen
6. Shells
  - a) Funktion
  - b) verschiedene Shells
7. Logfiles
  - a) syslogd
  - b) syslog-Kategorien
  - c) Kernel-Meldungen
  - d) Eingeloggte User
8. Dokumentation

# Grundlagen

## Hardware-Architekturen

Es gibt verschiedene Hardware-Architekturen, z.B.

- x86 (Standard-PCs)
- x86\_64 (PCs mit "Intel-64"-kompatiblen Prozessoren)
- ia64 (Server Systeme mit Intel Itanium-Prozessor)
- alpha (von HP, Entwicklung inzwischen eingestellt)
- sparc (Sun Microsystems)
- PPC (Macintosh- und Amiga-Modelle mit "PowerPC"-Prozessoren)
- p5 (IBM Serie-P, das sind ziemlich grosse Server)
- s390 (IBM Serie Z, das sind echte Großrechner)

Für jede dieser Hardware-Architekturen gibt es mindestens eine Linux-Distribution.

## x86-Hardware

- Prozessor
- RAM
- BIOS
- PCI-Bus:
  - Netz-Schnittstelle (Ethernet/ISDN)
  - PCI-Grafikkarte
  - SCSI/IDE/SATA-Controller:
    - Festplatten
    - CD/DVD-Laufwerke
    - Bandlaufwerke
  - USB-Bus:
    - Memorysticks
    - Festplatten
    - Drucker
    - Kartenleser
    - Eingabegeräte
      - Maus
      - Tastatur
      - Joystick
    - ...
  - PCMCIA-Bus (Laptops):
    - Modems
    - Netz-Schnittstelle (Wireless/Ethernet/ISDN)
  - ...
- AGP-Bus: AGP-Grafikkarte
- PS/2-Ports: Maus, Tastatur, ...
- Serieller Port: Modem, ...
- Paralleler Port: Drucker, ...
- ...

# Betriebssysteme

Ein Betriebssystem besteht aus

- einem Kernel (auf die Hardware zugeschnitten)
- Systemprogrammen

Im Lieferumfang enthalten sind meist zusätzlich:

- Anwendungsprogramme
- Installationsroutine
- Dokumentation

Verbreitete Betriebssysteme:

- "Unix"-artige:
  - verschiedene Linux-"Distributionen":
    - Debian GNU/Linux
    - RedHat (Fedora und Enterprise-Linux)
    - Novell OpenSuSE und SuSE-Linux-Enterprise-Server
  - verschiedene BSD-Distributionen
  - Sun Solaris
  - IBM AIX
  - Hewlett Packard: UX, Tru64
- Microsoft Windows
- Mac OS
- IBM OS/390 und zOS, HP OpenVMS, ...

## Der Kernel

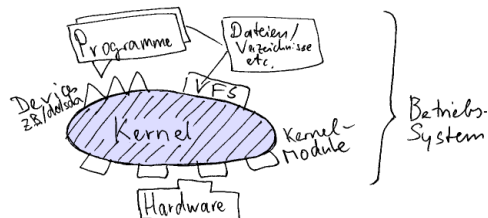


Abbildung: Kernel

Der **Kernel** dient als Abstraktionsschicht für die Hardware. Er besteht aus einer Programm-Datei (meist `vmLinux`) und einer Menge von *Modulen*, welche bei Bedarf hinzugeladen werden können.

Die **Programme** können z.B. über *Devices* auf Funktionen des Kernels zugreifen. Zum Beispiel ist das Device `/dev/sda` der ersten SCSI-Festplatte zugeordnet.

# Filesysteme

## Das "Virtual Filesystem" (VFS)

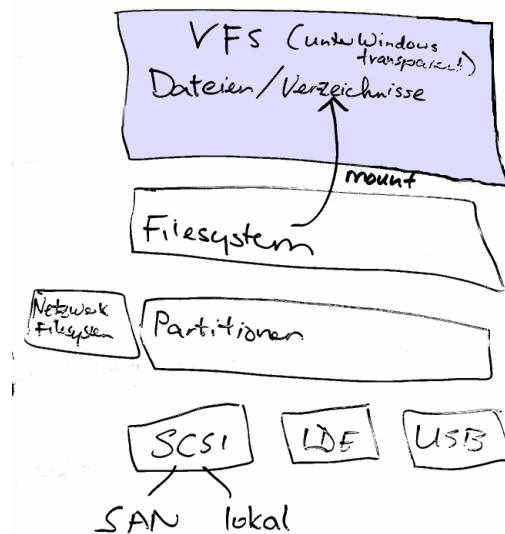


Abbildung: Virtual Filesystem (VFS)

Das VFS ist hierarchisch angeordnet, d.h. es gibt genau ein **Root-Verzeichnis (/)**. Im Gegensatz zu Windows kommen *keine Laufwerksbuchstaben* zum Einsatz, sondern die verschiedenen Filesysteme werden an verschiedenen Stellen in das VFS "hinein-gemounted".

Die einzelnen **Filesysteme** können auf verschiedenen Festplatten-Partitionen, Wechselmedien oder auf Fileservern im Netz liegen.

## Block-Devices

Festplatten sind Block-Devices. Sie werden über Controller angesprochen.

Man unterscheidet folgende Controller:

- IDE (auch AT-Bus, ATA, ATAPI oder PATA genannt)
- SATA
- SCSI:
  - über Hostadapter: lokale Festplatten
  - über Hostbus (FibreChannel)-Adapter: SAN
  - über USB: Memorysticks/Wechselplatten

# Filesystem-Typen

Filesysteme auf Blockdevices:

- ext3 (Linux native)
- fat32/vfat (Windows95 native)
- msdos (MS DOS native)
- iso9660 (CD)

Netzwerkfilesystems (von Fileservern):

- ntfs (Windows native)
- smbfs (Windows-Fileserver)
- ncfs (Novell Netware)
- ...

Besondere Filesysteme:

- swapfs
- proc filesystem

## Mounten

**Befehle:**

- **mount** *Block-Device Mountpoint -o Optionen*
- **umount** *Block-Device oder Mountpoint oder beides*
- **mkfs -t** *Filesystem-Typ Optionen Blockdevice*
- **chkfs** *Block-Device*

In Datei **/etc/fstab** steht, welche Block-Devices beim Bootvorgang gemounted werden sollen:

```
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/hda1      /          ext3    noatime,errors=remount-ro 0    1
/dev/hda3      /tmp       ext3    noatime,nosuid             0    2
/dev/hda5      /var       ext3    noatime                     0    2
/dev/hda6      /export    ext3    noatime,nosuid,nodev      0    0

# technical devices:
/dev/hda2      none       swap    sw,pri=2                   0    0
proc           /proc     proc    defaults                    0    0

# removable media:
/dev/cdrom     /mnt/cdrom auto    noauto,user,nodev,nosuid  0    0
/dev/fd0       /mnt/floppy auto    noauto,user,nodev,nosuid  0    0
```

*Listing: /etc/fstab*

# Filesystem-Elemente

Folgende Elemente können in einem Verzeichnis enthalten sein:

- Dateien
- Verzeichnisse
- Hardlinks\*: weiterer Dateiname für einen Dateiinhalt (nicht über Filesystemgrenzen hinweg möglich). Kommt ausserdem bei den Verzeichnissen "." und ".." zum Einsatz.
- Symlinks\*: Verweis auf andere Datei (auch Filesystem-übergreifend)
- Block-devices\*: z.B. Festplatten-Partition: `/dev/sda1`
- Character-devices\*: z.B. `/dev/mouse`
- Fifos\*: zur Kommunikation zwischen Programmen über stdin/stdout
- Sockets\*: zur Kommunikation zwischen Programmen über Socket-Library

\*: nicht bei allen Filesystemtypen

# Meta-Informationen

Zu jedem Element werden folgende Meta-Informationen gespeichert:

- Name des Elements
- Größe
- Inoden-Nummer (Adresse innerhalb des Filesystems)
- Link-Count (Zahl der Hardlinks auf diesen Dateiinhalt)
- Datum der letzten Änderung
- Zugriffsrechte (Benutzer, Gruppe, Permissions)\*
- Datum der Erstellung\*
- Datum des letzten Lesezugriffs\*
- erweiterte Zugriffsrechte (ACL)\*
- Kommentar\*

\*: nicht bei allen Filesystemtypen

# Der Filesystem Hierarchy Standard (FHS)

Der FHS beschreibt die Organisation innerhalb des VFS:

Unterhalb des Rootverzeichnisses liegen folgende Unterverzeichnisse:

- **/dev**: Devices
- **/etc**: Konfigurationsdateien
- **/home**: Homeverzeichnisse der Benutzer ("Eigene Dateien")
- **/var**: dynamische Dateien für diesen Computer (z.B. Logfiles, Druckjobs)
- **/usr**: Unix System Resources (Programme). Darunter: **/usr/bin**, **/usr/sbin**
- **/sbin**: Wichtige Programme für den Bootvorgang (nur für Admins interessant)
- **/bin**: Wichtige Programme für den Bootvorgang (auch für User interessant)

Kriterien zur Einteilung in die Verzeichnisse:

- *static* (selten ändernd) und *dynamic* (laufend ändernd)
- *shareable* (auf mehreren Computer identisch) und *unshareable* (Computer-spezifisch)

Beispiele:

	shareable	unshareable
static	<b>/usr</b>	<b>/etc</b>
dynamic	<b>/home</b>	<b>/var</b>

# Der Bootvorgang

## Grober Ablauf

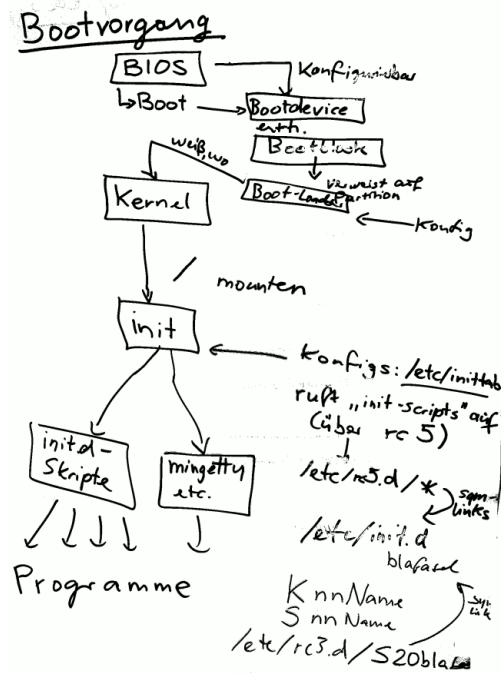


Abbildung: Linux-Bootvorgang

Der Bootvorgang erfolgt in fünf Phasen:

- BIOS
- Bootloader (lilo/grub/syslinux, liegt im MBR)
- Kernel (/boot/vmlinuz)
- /sbin/init
- weitere Programme (abhängig vom gewünschtem Betriebsmodus (**Runlevel**), u.a. die Konsolen (mingetty, X11))

# Runlevel

`init` bringt das System in einen der folgenden Runlevel:

	Debian	SuSE	RedHat	Solaris
Runlevel 0	Herunterfahren			
Runlevel 1	Wartung (ohne Hintergrundprozesse)			
Runlevel 2	Normalzustand	Textmodus (ohne X)		ohne Serverdienste
Runlevel 3	-		Normalzustand	
Runlevel 4	-			
Runlevel 5	-	Normalzustand	Hardwareerkennung	Herunterfahren +Ausschalten
Runlevel 6	Herunterfahren +Neu starten			

Beispiele:

- Bei Aufruf von `init 6` rebootet das System. Dieser Befehl kann auch durch Drücken von `strg+Alt+Entf` ausgelöst werden.
- Mit `init 1` versetzt man das System in den Wartungsmodus

# init

Konfigurationsdatei: `/etc/inittab`

```
# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change of runlevel
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# /sbin/getty invocations for the runlevels.
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

# Fax/Dialin/Anrufbeantworter (Modem und ISDN)
I0:2345:respawn:/usr/sbin/vboxgetty -d /dev/ttyI0
I1:2345:respawn:/sbin/mgetty ttyI0
I2:2345:respawn:/sbin/mgetty ttyS0
```

*Listing: /etc/inittab*

Der "Default"-Runlevel, welcher beim Hochfahren automatisch angesteuert wird, ist in `/etc/inittab` in der Zeile `id:2:initdefault:` angegeben.

## init.d-Scripts

- `/etc/init.d/*`: Aufruf mit Parameter `start` oder `stop`
- `/etc/rc[0-6].d/[KS][0-9][0-9]*`: Symlinks auf `/etc/init.d/...`
- `/etc/rc.boot/*`

(Abweichende Pfade bei einigen Distributionen)

Beispiel: `/etc/init.d/apache2 restart`

# Zugriffsrechte

## User

Verschiedene User

- Persönliche User
- Technische User (Systemuser)
- root

können sich gleichzeitig am System anmelden und haben eigene

- Prozesse
- Dateirechte

Userliste: `/etc/passwd` . Pflege mit `useradd`, `userdel`, `usermod`

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
[...]
jfranken:x:1000:100:Johannes Franken,,,:/export/home/jfranken:/bin/bash
bfranken:x:1001:100:Beate Franken,1,-12,:/export/home/bfranken:/bin/bash
[...]
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
```

*Listing: /etc/passwd*

Passwörter werden nicht aus Sicherheitsgründen nicht gespeichert. Allerdings werden die Hashwerte (z.B. md5-Checksumme) der Passwörter in `/etc/shadow` hinterlegt:

```
root:$1$4zU1GhCL$8yk51.RNza60JWJm4GA0U.:13363:0:99999:7:::
daemon:*:10810:0:99999:7:::
bin:*:10810:0:99999:7:::
[...]
jfranken:t73jv3leib7Bw:11331:0:99999:7:::
[...]
```

*Listing: /etc/shadow*

Linux überprüft das beim Login eingegebene Passwort, indem es dessen Hashwert mit dem hinterlegten Hashwert vergleicht. Eine "Entschlüsselung" der Hashwerte ist derzeit nur über Brute-Force-Methoden möglich, z.B. mit den Tools `qcrack` oder `john`.

Die hinterlegten Hashwerte können mit dem `passwd`-Kommando geändert werden.

# Gruppen

Jeder User gehört einer oder mehreren Gruppen an.

Gruppenliste: `/etc/group`

```
root:x:0:
[...]
fax:x:21:jfranken,bfranken
cvs:x:103:wml,jfranken,bfranken
[...]
nogroup:x:65534:
```

*Listing: /etc/group*

Gruppen hinzufügen, umbenennen und löschen mit `groupadd`, `groupdel`, `groupmod`

Primären Gruppe eines Users zuordnen:

- `usermod -g Gruppenname Username` oder
- `vi /etc/passwd` (oder besser: `vipw`)

Weitere Gruppen eines Users zuordnen:

- `usermod -G Gruppe1,Gruppe2,... Username` oder
- `vi /etc/group` (oder besser: `vigr`)

## **Achtung:**

`usermod -G` ersetzt *alle* vorhandenen weiteren Gruppenzuordnungen. Daher unbedingt die *vollständige* Gruppenliste angeben.

Änderungen an der Gruppenzuordnung wirken erst beim nächsten Login.

# Filesystem-Permissions

## Konzept

Unix unterscheidet folgende Berechtigungen:

- Bzgl. Dateien/Devices/Fifos/Sockets:
  - **read**: Lesen des Inhaltes
  - **write**: Ändern des Inhaltes
  - **execute**: Ausführen
- Bzgl. Verzeichnissen:
  - **read**: Auflisten eines Verzeichnisses
  - **execute**: Betreten eines Verzeichnisses
  - **write**: Anlegen/Löschen/Umnennen eines Elements in dem Verzeichnis

(Symlinks erben die Berechtigungen ihres Zieles.)

Für jedes Filesystem-Element ist festgelegt, welche dieser Berechtigungen

- **user**: der Eigentümer
- **group**: eine genau benannte Gruppe
- **others**: alle Übrigen

auf ihm haben.

Zusätzlich gibt es noch die "Sonderrechte" (special bits), die aber nur selten erforderlich sind:

- **setuid** (u+s, bei ausführbaren Dateien): Das Binary läuft unter der Kennung des Eigentümers (nicht des Aufrufers).
- **setgid** (g+s, bei ausführbaren Dateien): Das Binary läuft unter der Gruppe des Eigentümers (nicht des Aufrufers).
- **setgid** (g+s, bei Verzeichnissen): Alle neu angelegten Elemente übernehmen die Gruppe des Verzeichnisses (nicht des anlegenden Users)
- **sticky** (o+t, bei Verzeichnissen): Nur die Eigentümer dürfen die Elemente in diesem Verzeichnis löschen oder umnennen (nicht alle auf das Verzeichnis schreibberechtigten User)

# Filesystem-Permissions anzeigen (ls)

Beispiel:

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root staff 3020 2006-12-15 17:06 /etc/passwd
```

Bedeutung der Spalten:

1. **Typ** (hier: -):
  - -: Datei (auch Hardlink)
  - d: Verzeichnis
  - l: Symlink
  - c: Character-device
  - b: Block-device
  - s: Socket
  - p: Fifo
2. **Rechte:**

User			Group			Others ("world")		
read(r)	write(w)	execute(x)/ setuid(s)	read(r)	write(w)	execute(x)/ setgid(s)	read(r)	write(w)	execute(x)/ sticky(t)
r	w	-	r	-	-	r	-	-

3. **Link-Count** (hier: 1): wieviele Verzeichniseinträge auf diesen Dateinbalt zeigen.
4. **User** (hier: root), dem die Datei gehört (Eigentümer)
5. **Group** (hier: staff), auf die sich die o.g. Gruppenrechte beziehen
6. **Dateigröße** (hier: 3020)
7. **Modification-Time** (hier: 2006-12-15 17:06)
8. **Dateiname** (hier: /etc/passwd)

Ergebnis: Jeder kann die Datei /etc/passwd lesen, aber nur der Eigentümer (User=root) kann sie ändern.

## Numerische Darstellung

Man kann die Dateirechte numerisch darstellen:

Special bits			User			Group			Others ("world")		
setuid (u+s)	setgid (g+s)	sticky (o+t)	read(r)	write(w)	execute(x)	read(r)	write(w)	execute(x)	read(r)	write(w)	execute(x)
4	2	1	4	2	1	4	2	1	4	2	1
addieren (0 kann entfallen)			addieren			addieren			addieren		

Beispiel:

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root staff 3020 2006-12-15 17:06 /etc/passwd
```

## Permissions ändern (chmod)

chmod ändert die Dateirechte.

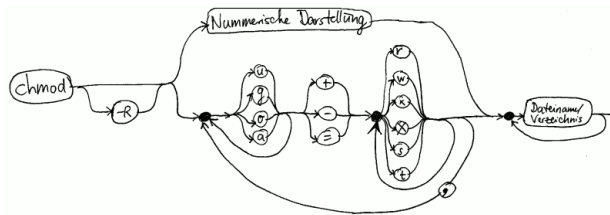


Abbildung: chmod-Aufruf

Beispiele:

- `chmod g+w /home/jfranken/incoming/`
- `chmod -R 750 /home/jfranken/`

## Eigentümer und Gruppe ändern (chown, chgrp)

chown und chgrp ändern Eigentümer und Gruppe.

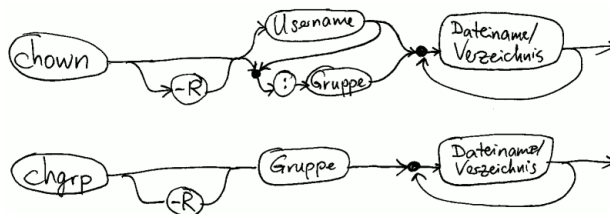


Abbildung: chown und chgrp-Aufruf

Beispiele:

- `chown jfranken:www-data /var/www/index.html`
- `chown -R jfranken /export/home/jfranken`

## Konsolen

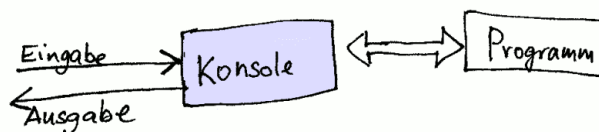


Abbildung: Konsole

- Textkonsole
- Serieller Port
- Pseudo-Terminals:
  - X-Clients:
    - xterm (Shift-PgUp/Dn, Ctrl-Maustaste)
    - konsole (Menüleiste)
    - ...
  - sshd
  - ...

# Shells

## Funktion

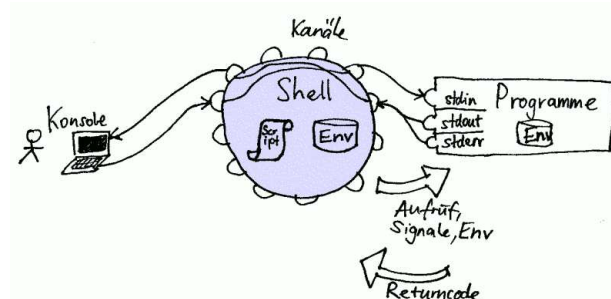


Abbildung: Aufgaben der Shell

- Ein/Ausgabe-Kanäle "miteinander verdrahten"
- Programme aufrufen
- Shellscripts abarbeiten
- Environment bereitstellen
- Signale an Programme senden
- Returncodes abfangen

## verschiedene Shells

Standard-Shell unter Linux:

- **bash**: Erweiterte sh (komfortablere Nutzerschnittstelle)

Darüber hinaus gibt es noch einige weitere Shells, z.B.

- **sh**: auf allen Unixsystemen verfügbar (insb. AIX/Solaris)
- **ksh**: Erweiterte sh (für Shellscript-Programmierung), Standard unter AIX/Solaris.
- **zsh**: Erweiterte sh (Komfort + Programmierung)

sowie

- **csh**: Komplette andere Syntax (nicht sh-kompatibel). Angeblich für C-Programmierer geeignet.
- **tcsh**: Erweiterte csh (Komfort)

### Achtung:

Shells sind nicht zwangsläufig kompatibel zwischen verschiedenen Releases und Betriebssystemen. Daher kann beim Umzug auf einen anderen Rechner eine Anpassung Ihrer Shellscripts erforderlich sein.

# Logfiles

## syslogd

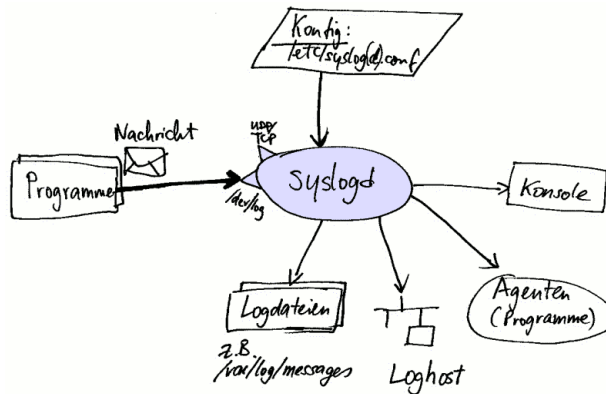


Abbildung: syslogd

```
# First some standard logfiles.  Log by facility.
auth,authpriv.*          -/var/log/auth.log
*.*;auth,authpriv.none  -/var/log/syslog
cron.*                   -/var/log/cron.log
daemon.*                 -/var/log/daemon.log
kern.*                   -/var/log/kern.log
lpr.*                    -/var/log/lpr.log
local2.*                 -/var/log/ppp.log

# Logging for the mail system.
mail.info                -/var/log/mail.info
mail.warn                -/var/log/mail.warn
mail.err                 /var/log/mail.err

# Emergencies are sent to everybody logged in.
*.emerg                  *

# Ausgewaehlte Nachrichten auf Konsole 8 ausgeben (Alt-F8)
daemon,mail.*;*.=debug;*.=info;*.=notice;*.=warn  /dev/tty8
```

Listing: /etc/syslog.conf

Mit dem **logger**-Programm kann Nachrichten an den syslogd senden (z.B. in Shellscripts)

# syslog-Kategorien

## Facility:

- auth (=security)
- authpriv
- cron
- daemon
- ftp
- kern
- lpr
- mail
- mark
- news
- syslog
- user
- uucp
- local[0-7]

## Priority: (aufsteigende Wichtigkeit)

1. debug
2. info
3. notice
4. warning (früher: warn)
5. err (früher: error)
6. crit
7. alert
8. emerg (früher: panic)

# Kernel-Meldungen

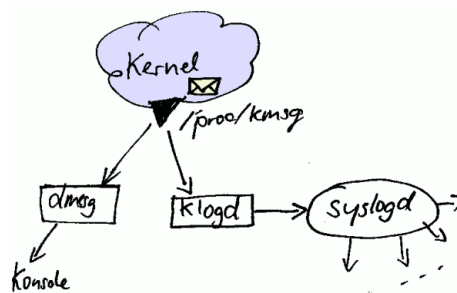


Abbildung: Kernel-Meldungen

Der Kernel speichert Kernelmeldungen in einem Ringpuffer, den man

- mit **dmesg** auslesen oder
- mit dem **klogd** in den **syslogd** einspeisen kann.

```
$ dmesg
usb-storage 1-1:1.0: usb_probe_interface - got id
scsi5 : SCSI emulation for USB Mass Storage devices
  Vendor: Maxtor    Model: OneTouch III    Rev: 0344
  Type:   Direct-Access    ANSI SCSI revision: 02
SCSI device sda: 1465149168 512-byte hdwr sectors (750156 MB)
sda: assuming drive cache: write through
sda: sdal
Attached scsi disk sda at scsi5, channel 0, id 0, lun 0
atkbd.c: Keyboard on isa0060/serio0 reports too many keys pressed.
IN=ppp0 OUT= MAC= SRC=192.168.100.253 DST=84.178.163.142 LEN=61 TOS=0x00 PREC=0x00 TTL=120 ID=49363 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=11833
Adding 1959920k swap on /dev/hda2.  Priority:2 extents:1
eth2: link down
eth2: link up, 10Mbps, half-duplex, lpa 0x0000
e100: eth0: e100_watchdog: link up, 100Mbps, full-duplex
```

Listing: dmesg

# Eingeloggte User

werden in `/var/log/wtmp` (Binärdatei) protokolliert.  
Auslesen mit **last**:

```
$ last
jfranken pts/11   gericom.jfranken Sat Mar 10 12:52   still logged in
bfranken pts/2   motte.jfranken.d Sat Mar 10 10:22   gone - no logout
jfranken ttyS0   mgetty - /dev/tt Fri Mar 9 10:54 - 13:39   (02:45)
bfranken pts/11   motte.jfranken.d Thu Mar 8 22:04 - 07:26   (09:21)
hrother pts/2   p54b07827.dip.t- Thu Mar 8 21:54 - 06:44   (08:49)
fax_inc ttyS0   +49 641 9750230 Thu Mar 8 07:54 - 07:54   (00:00)
bfranken pts/1   hummel.jfranken. Tue Mar 6 20:34 - 21:17   (00:42)
fax_inc ttyS0   +49 6654 8354 Tue Mar 6 11:26 - 11:27   (00:00)
bfranken pts/2   hummel.jfranken. Tue Mar 6 07:21 - 21:15   (13:54)
jfranken pts/11   gericom.jfranken Mon Mar 5 20:39 - 10:34   (1+13:54)
[...]
```

*Listing: last*

## Dokumentation

- Man-Pages (zu jedem Programm und zu jeder Konfigurationsdatei)
  - **man ls** (Ausgabe im **less**. q=Beenden, /=suchen, n=weetersuchen)
  - **xman** (Grafischer Manpage-Browser unter X)
  - **man -t ps ls | lpr** (Ausgabe auf Drucker)
- `/usr/share/doc` (Handbücher)
- Bücher (z.B. im O'Reilly- oder MITP-Verlag)
- Internet (z.B. <http://www.linuxwiki.org/>, <http://www.tldp.org/>)