Linux-Administration

Zusammengestellt für die FH Kaiserslautern/Zweibrücken im März 2006

Zusammenfassung

Linux ist ein leistungsfähiges, stabiles und äußerst umfangreiches Betriebssystem. Im Rahmen dieses Kurses sollen die grundlegenden Merkmale von Linux als Server-Plattform, insbesondere die Einrichtung und Wartung häufig genutzter Serverdienste erläutert sowie ein Einblick in die wichtigsten Konfigurations- und Administrationsaufgaben des Linux-Administrators gegeben werden.

Ein wenig Geschichte

1970- Erste Unix-Betriebssysteme auf Großrechnern, vor allem an Universitäten.

1988- POSIX 1003.1 als Standard verabschiedet.

1993- Linus Torvalds gibt den Quelltext für Kernel 0.1 frei. Das System wird zusammen mit der GNU-Software der Free Software Foundation von vielen Entwicklern weltweit über das Internet weiterentwickelt.

heute- “Linux World Domination“? (Im Einsatz als kleines und mittleres Server-System auf kostengünstigen PCs im Intranet/Internet-Bereich bereits erreicht.)

Die GNU General Public License

Gibt den Empfängern der Software das Recht, ohne Lizenzgebühren

- den Quelltext zur Software zu erhalten, um diese analysieren und nach eigenen Wünschen modifizieren zu können,
- die Software in beliebiger Anzahl zu kopieren,
- die Software zu modifizieren (s.o.),
- die Software im Original oder in einer modifizierten Version weiterzugeben oder zu verkaufen, auch kommerziell, wobei die Empfänger der Software diese ebenfalls unter den Konditionen der GPL erhalten.

`http://www.gnu.org/`

Andere Lizenzen

Achtung: Nicht alle Software, die für das Linux-Betriebssystem verfügbar ist, unterliegt der GPL!

Verschiedene Hersteller verwenden unterschiedliche Lizenzmodelle für ihre Software, unabhängig von GNU/Linux als Plattform.

Eigenschaften von Unix

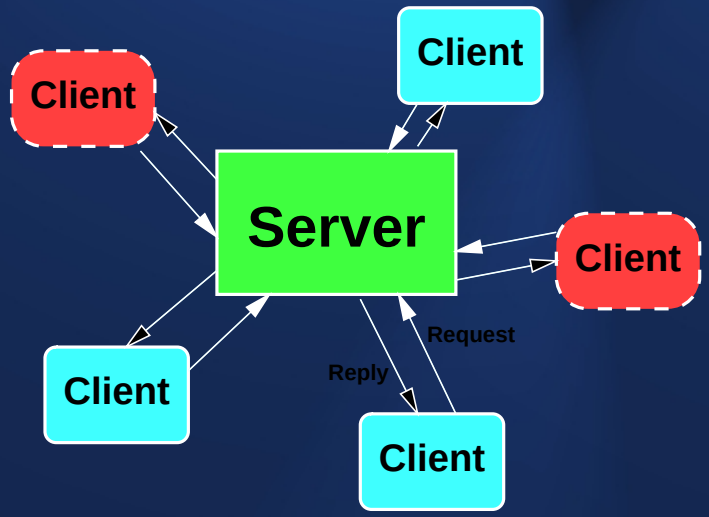
- Mehrere Aufgaben gleichzeitig (Multitasking)
- Mehrbenutzerfähig (Multiuser)
- Auf vielen Hardware-Plattformen lauffähig (portabel)
- Effiziente Ausnutzung der Resourcen (nicht proprietär)
- hierarchisches Dateisystem
- Stabilität durch eigenen Speicherbereich für jedes Programm (Virtual Memory, Speicherschutz)
- strikte Trennung zwischen Betriebssystem („Kernel“) und Anwendersoftware (Desktop-, Server-Suiten)

Verschiedene Unix-Betriebssysteme

Name/TM	**Hersteller**
SunOS/Solaris	SUN Microsystems
HPUX	Hewlett Packard
Aix	IBM
Sinix	Siemens/Nixdorf
Ultrix/DEC Unix(OSF/1)	Digital Equipment
Linux	Community (Entwickler)
FreeBSD/NetBSD	Community (Entwickler)

. . .

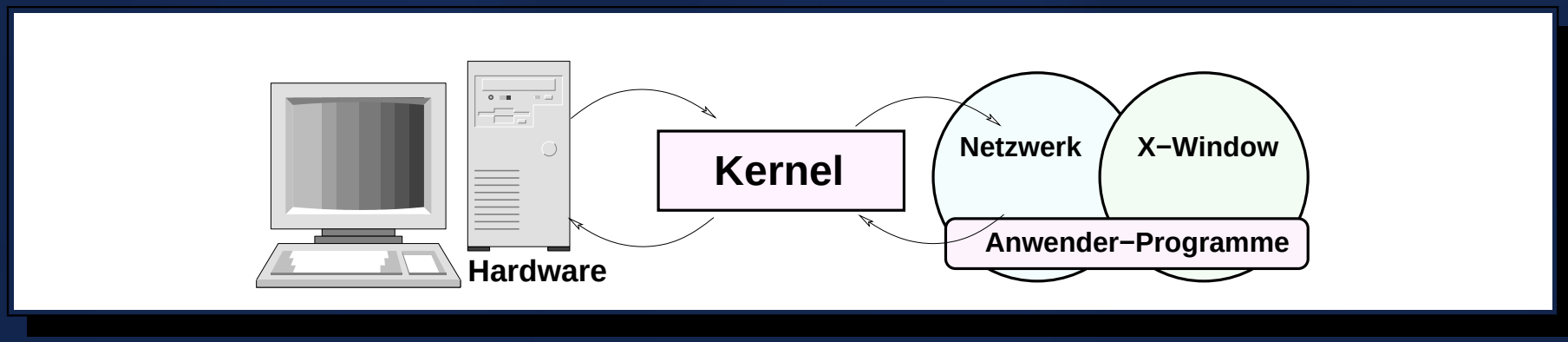
Client/Server-Prinzip



- Client: Dienstanforderer (z.B. Browser)
- Server: Diensterbringer (z.B. WWW-Server)
- Abwicklungsprotokoll, das beide Partner verstehen, z.B. **`http`**.

Linux-Internals

- Der **Kernel** ist die Schnittstelle zwischen Hardware und Anwendersoftware.
- Die Netzwerkfähigkeit ist im Kernel integriert, daher sind prinzipiell **alle** Programme netzwerkfähig.
- Graphische Benutzeroberfläche ist ein Anwender-programm, kein Bestandteil des Betriebssystems.



Kommandos in der Shell

- Viele, kleine Programme für jeweils nur eine Aufgabe,
- extrem kurze, „selbsterklärende“ Kommandonamen,
- leichte Kombinationsmöglichkeit dieser kleinen Programme.

Unterschiede zu DOS/Windows

Einige augenfällige Unterschiede zu DOS:

- Groß-/Kleinschreibung wird beachtet.
- Foreslash „/“ statt Backslash. „\“ als Trenner bei Pfaden.
- Es gibt keine „Laufwerksbuchstaben“.
- Ausgabe-Umleitungen werden nicht über Hilfsdateien „simuliert“.
- Mehrere Programme können gleichzeitig im Hintergrund laufen („&“).

Kommando-Syntax

In der Shell eingegebene Kommandos haben im Allgemeinen das folgende Format:

```
Programmname Optionen Argumente Umleitung
```

Die Umlenkung von Ein- und Ausgabe funktioniert, anders als bei DOS, wo Temporärdateien geschrieben werden, auf direktem Weg auch zwischen Programmen.

> Datei	Umlenkung der Ausgabe in Datei
< Datei	Umlenkung der Eingabe von Datei
| Kommando	Umlenkung der Ausgabe in die Eingabe eines anderen Programms.

Navigieren im Dateisystem mit der Shell

`pwd`	Ausgabe aktuelles Arbeitsverzeichnis
`cd Verzeichnisname`	Wechsel des aktuellen Verzeichnisses
`ls -l [wildcards]`	Ausführliches Auflisten von Dateien*)
`mkdir [-p] Verz.`	Lege [Mit Unterverz.] Verzeichnis an.
`cp [-a] Alt Neu`	Kopiere [Klone Alles] von Alt nach Neu
`mv Alt Neu`	Benenne Alt nach Neu um
`rm [-rf] wildcard`	Lösche unwiderruflich [rekursiv forciert] †)

---

*) Die angezeigten Dateirechte werden noch ausführlich besprochen.

†) Tipp: Überlegen Sie zweimal, bevor Sie mit der Option **-rf** etwas löschen, gerade, wenn Sie momentan Administratorstatus haben!

Linux Benutzer und Administrator

Merkregel: Benutzen Sie den Administratoraccount (User-ID 0) **ausschließlich** zur systemweiten Installation von Programmpaketen und für Konfigurationsarbeiten, **niemals** jedoch zum regulären Arbeiten!
Nur dann kann Unix/Linux Ihnen die vielgepriesene Sicherheit und Stabilität bieten.
Zum regulären Arbeiten in einer komfortablen Umgebung (z.B. graphische Benutzeroberfläche) ist ein normaler Benutzer-Account vollkommen ausreichend, und in vieler Hinsicht im Arbeitskomfort dem Administratoraccount sogar überlegen.

Übung: Benutzer einrichten

Melden Sie sich auf der Textkonsole Ihres Rechners[1)] als **root** an und richten Sie sich mit dem Kommando

```
useradd -c "Mein Name" -m benutzer
```

eine Benutzerkennung ein. Setzen Sie dem neuen Benutzer mit dem Kommando

```
passwd benutzer
```

ein Passwort.[2)]

Übung: Anmelden als Benutzer

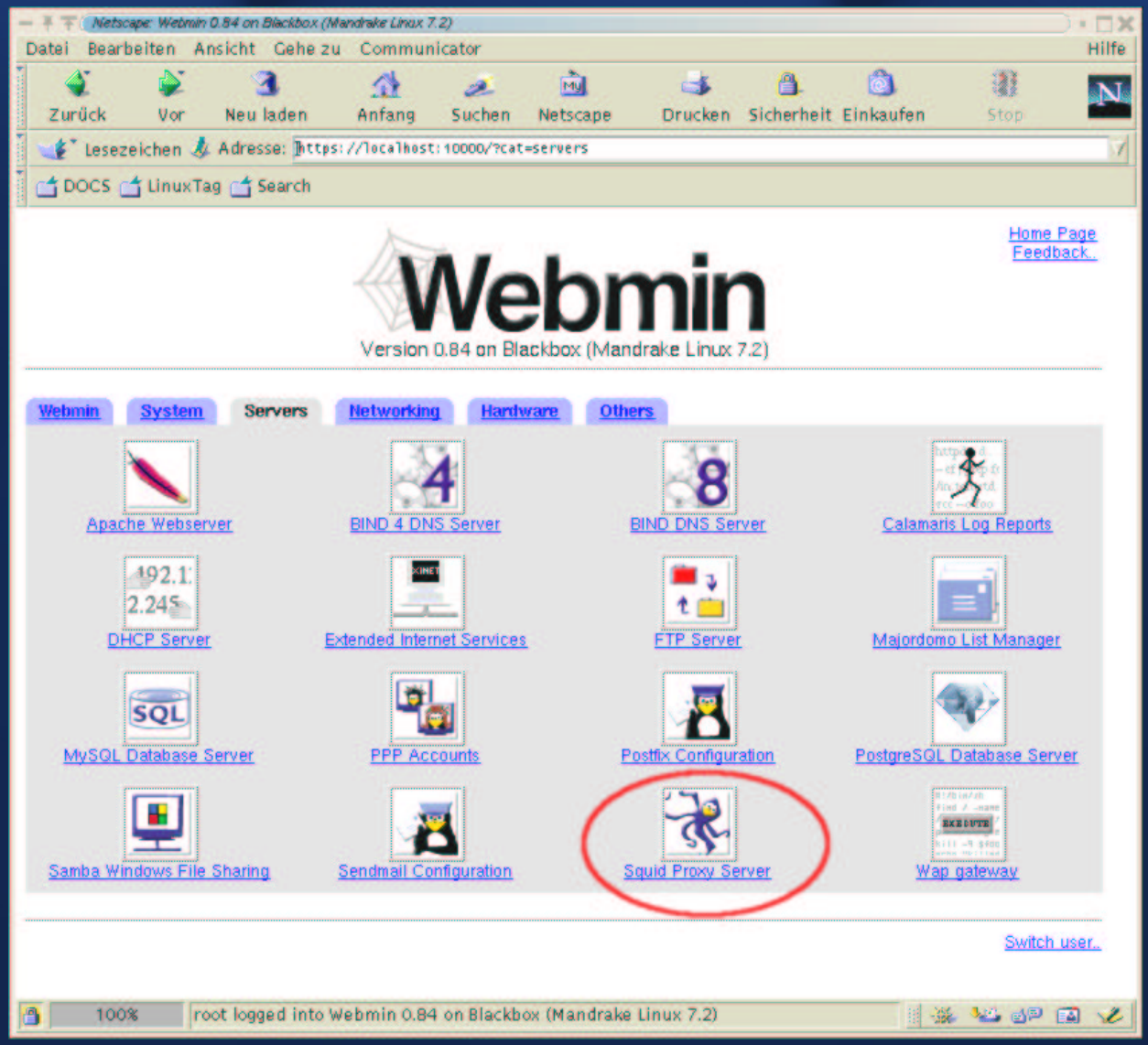
Wechseln Sie wieder zum graphischen Login (`Steuerung-Alt-F7`) oder, falls Ihr Rechner zunächst nur für den Textmodus vorbereitet wurde, geben Sie als **root** das Kommando **xdm** ein.
Melden Sie sich nun mit Ihrer neuen Benutzerkennung an. Starten Sie ein Terminal-Fenster (Menüleiste), denn wir wollen nun mit der Shell weiterarbeiten, ohne auf den Komfort der graphischen Oberfläche verzichten zu müssen.

Wechsel des Benutzerstatus

Mit dem Kommando **su** wechseln Sie in der aktiven Shell zum Status des Systemadministrators. Hierbei werden Sie nach dem Passwort des Administrators gefragt, das Sie (unsichtbar) eingeben müssen.
Bei Erfolg verändert sich Ihr Eingabeprompt, und Sie haben in dieser Shell alle Rechte des Administratoraccounts (welcher auf den meisten Systemen die Benutzerkennung **root** hat).

Vorsicht: ab diesem Zeitpunkt können falsch eingegebene Kommandos, die als normaler Benutzer harmlos sind, in dieser Shell zerstörerische Wirkung auf Ihr System haben!

Remote-Administration mit **Webmin**



Übung: Benutzerattribute mit Webmin

Verbinden Sie sich zum auf Ihrem Rechner laufenden Webmin. Starten Sie hierzu

```
netscape https://localhost:10000/
```

Auch hier müssen Sie sich zunächst als **root** anmelden.
Finden Sie den Konfigurationspunkt „Benutzer und Gruppen“ und lassen Sie sich die Einstellungen für Ihre neu eingerichtete Kennung anzeigen.
Beenden Sie netscape wieder.

vi – Seitenorientierter Texteditor

- Kompakter, schneller Texteditor,
- gehört zum Standard-Equipment auf jedem Unix-System,
- kennt *Kommandomodus* (Befehlseingabe) und *Insert*-Modus (Texteingabe),

Vorsicht: Direkt nach dem Start befindet sich der vi im *Kommandomodus*, d.h. jede Tastatureingabe wird als **Kommando** interpretiert, nicht als Eingabetext!

vi – Kommandomodus

<**Escape**>	Rückkehr vom Insert- in den Kommandomodus
i	Wechsel in den **I**nsert-Modus (Direkteingabe)
o	**O**pen: Neue Zeile anfügen ☞ Insert-Modus
dd	**D**elete: Aktuelle Zeile löschen
p	**P**aste: [Gelöschten] Text einfügen
x	Zeichen, auf dem der Cursor steht, löschen
:r Datei	**R**ead: Datei ab Cursor einfügen
:w	**W**rite: Datei speichern
:q	**Q**uit: `vi` beenden
:wq	**W**rite and **Q**uit: Speichern und Beenden
:q!	Beenden ohne Speichern

:%s/alt/neu/gc Interaktiv `alt` durch `neu` ersetzen

Fast alle Kommandos lassen sich gruppieren oder mit einer vorangestellten Zahl mehrfach ausführen.

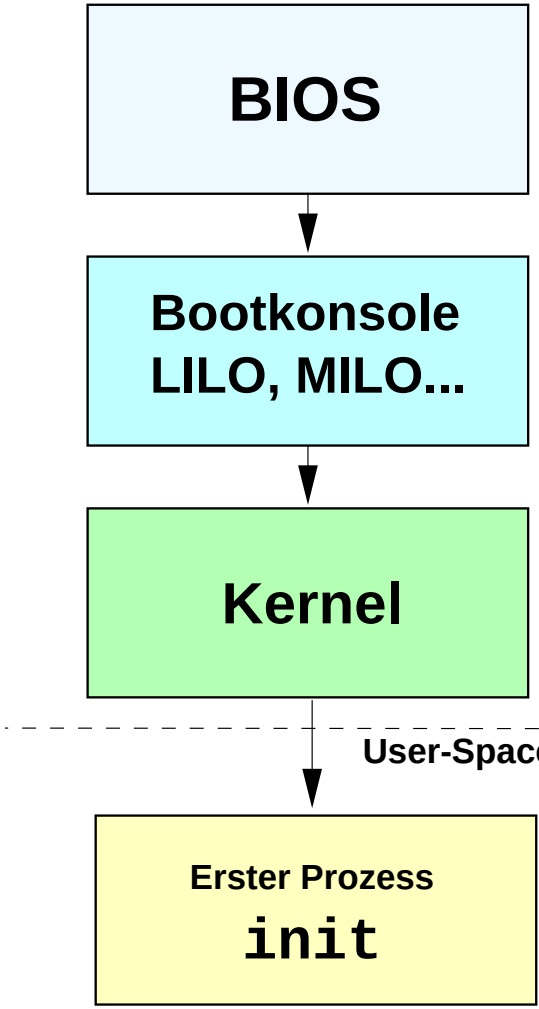
Übung

1. Legen Sie in Ihrem Benutzer-Heimverzeichnis eine neue Datei *uebung1.txt* mit **vi** an und geben Sie einen (beliebigen) Text ein. Speichern Sie die Datei und verlassen Sie vi.

2. Legen Sie mit **`vi`** in Ihrem Benutzer-Heimverzeichnis eine neue Datei *uebung2.txt* an. Laden Sie hier die zuvor erstellte Datei, fügen Sie die Datei */etc/passwd* am Ende an und ersetzen Sie alle „**:**“ durch Leerzeichen. Speichern Sie das Ergebnis unter dem Dateinamen *uebung2b.txt* ab.

3. Verlassen Sie vi und listen Sie mit **ls** Ihr Heimverzeichnis. Fällt Ihnen etwas auf?

Booten von Linux



Chipset und CPU initialisieren
Bootbare Geräte suchen
In Bootroutine verzweigen

(Optional)
Liste von bootbaren Partitionen anzeigen
Interaktive Auswahl / Timeout
Betriebssystem auswählen / laden

Gerätetreiber initialisieren
Hardware (Speicher/Laufwerke) erkennen
Interne Subsysteme initialisieren
Root-Dateisystem mounten

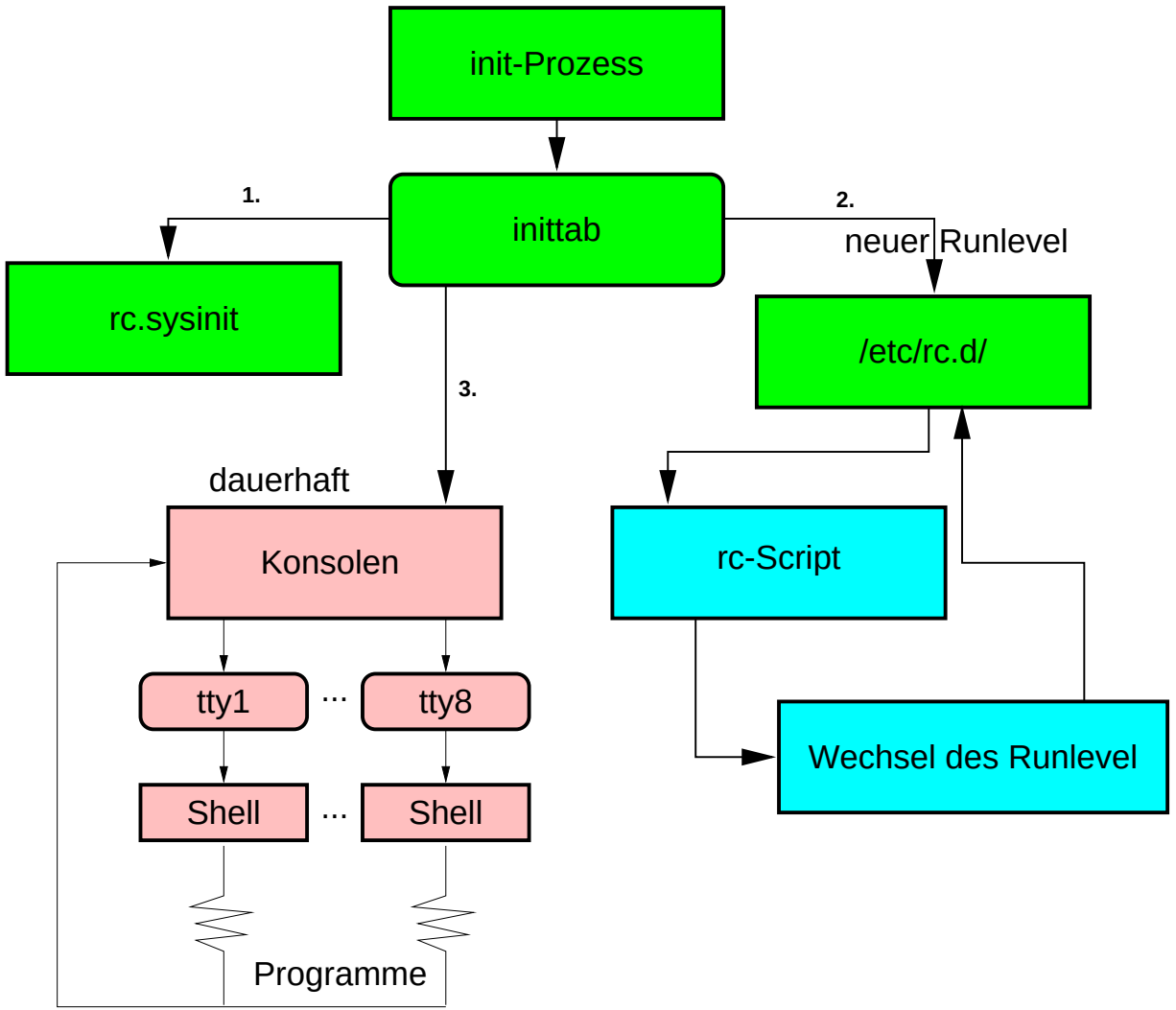
/etc/inittab lesen
Allgemeine Initialisierung starten
(Dateisystem-check, Mounts, Module laden ...)
Runlevel-Subsysteme starten
Permanente Prozesse (getty/login) starten

Aufgaben von `init`

- System-Initialisierungen (`rc.sysinit` bzw. Skripte in `rcS.d`),
- dauerhaft zu kontrollierende Prozesse (Logins, USV-Kontrolle),
- Subsysteme in bestimmten Systemzuständen (Netzwerk, Serverdienste, Remote-Dateisystem, ...)

Initialiserungen und *dauerhafte Einstellungen* ändern sich selten, *Subsysteme* werden jedoch häufiger, auch im laufenden Betrieb, umkonfiguriert ☞ Runlevel.

Ablauf von `init`



Beispiele für Runlevel

1 Eingeschränktes System nur für den Systemverwalter zu Administrationszwecken (Single-User-Mode)

2 Komplettes System, fertig für Benutzerbetrieb, ohne Netzwerk Filesystem

3 Komplettes System, fertig für Benutzerbetrieb, mit Netzwerk Filesystem

5 Komplettes System, fertig für Benutzerbetrieb, mit Netzwerk Filesystem und X-Window Login

Beispiele für Runlevel

Spezielle Runlevel:

0 Halt/Powerdown (nur in diesem Zustand kann der Rechner ohne Gefahr von Datenverlust ausgeschaltet werden)

6 Reboot

Runlevel-Skripte

Installation: Beim Installieren eines Programmpaketes, das einen Dienst zur Verfügung stellt, wird häufig ein Init-Skript automatisch nach **`/etc/init.d/dienstname`** installiert.
Mit distributionsabhängigen Tools werden dann, automatisch oder manuell, die entsprechenden Start- und Stopp-Links in die Runlevel-Verzeichnisse eingetragen (was sich natürlich genausogut manuell erledigen lässt).

Runlevel-Skripte

- RPM-basierte Systeme: **`chkconfig`**
- DEB-basierte Systeme: **`update-rc.d`**

Übung

- Sehen Sie in **`/etc/inittab`** nach, in welchem Runevel Ihr System normalerweise startet. Falls nicht per Default, eingetragen, ändern Sie den Default-Runlevel zu **`5`** (Multi-userbetrieb incl. graphischem Login) und teilen Sie **`init`** die Änderung mit, indem Sie ein **`HUP`**-Signal an den Prozess mit der Prozess-ID **`1`** schicken (**`init`** hat immer die erste Prozess-ID im System).
- Sorgen Sie dafür, dass in allen Runlevels automatisch der Apache-Webserver gestartet wird. Beobachten Sie, was sich durch **`update-rc.d`** im Verzeichnis **`/etc/rc5.d`** und **`/etc/rc0.d`** geändert hat!

Was ist ein „Block-Device“?

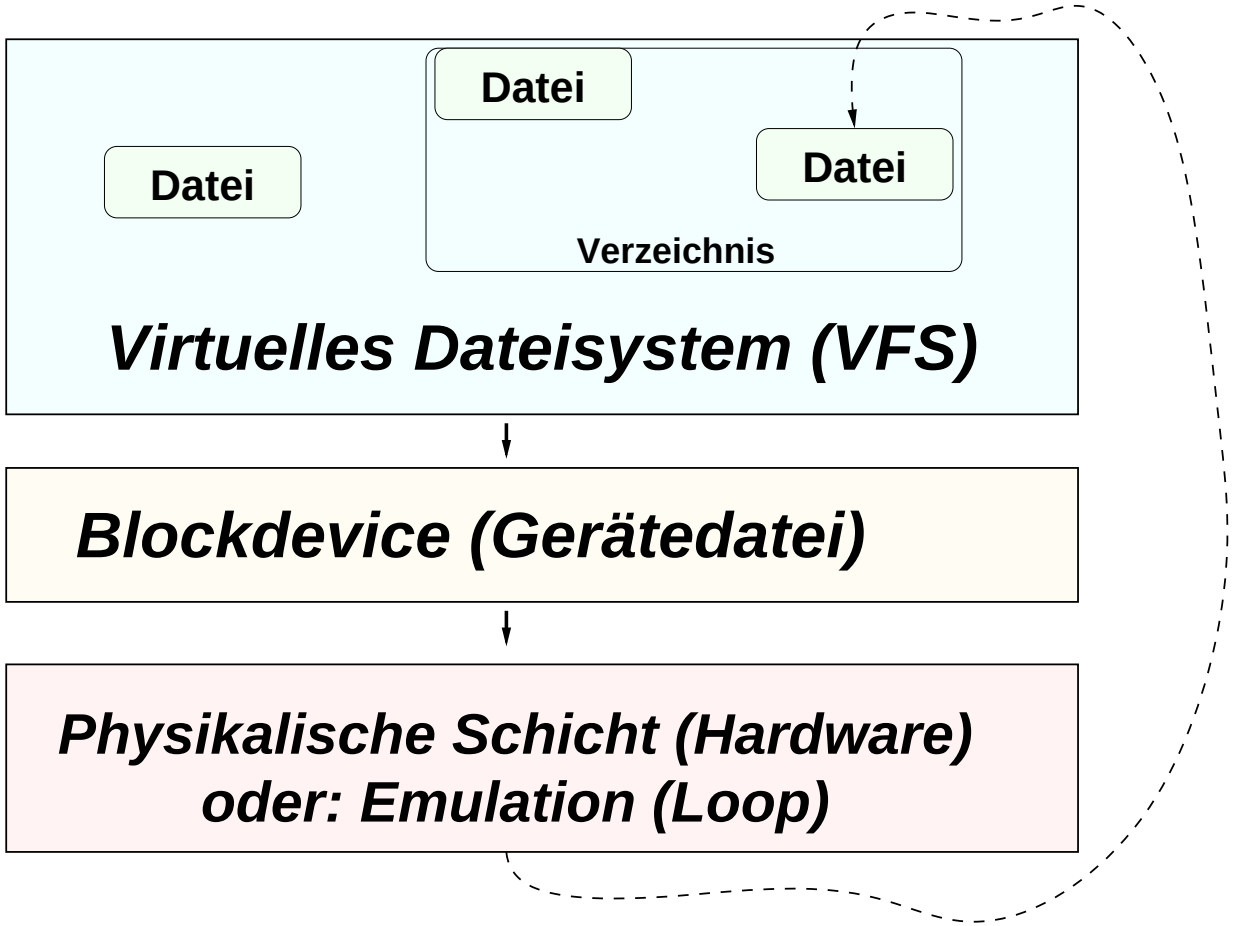
- Prinzipiell ein durchgängiger Bereich, auf dem Daten untergebracht sind,
- keinerlei sichtbare „Struktur“, abgesehen von der Unterteilung in „Blöcken“ konstanter Größe,
- kann eine Datei, eine Partition, oder ein kompletter Datenträger sein.

Block Device-Dateien

```
knopper@Koffer:~$ ls -l /dev/hda /dev/hda1

brw-rw---- 1 root disk 3, 0 2006-03-25 14:31 /dev/hda
brw-rw---- 1 root disk 3, 1 2006-03-25 14:31 /dev/hda1
```

Dateisystem und Blockdevice-Layer



mount

Syntax:
```
mount -t dateisystemtyp \
      -o optionen,... \
      blockdevice \
      zielverzeichnis
```

Aufgabe: Abbilden der „unstrukturierten“ Daten eines Block Device in eine Verzeichnisstruktur.
Hierbei sind die Optionen und die Wirkung hochgradig Dateisystemtyp-spezifisch!

Unterstützte Dateisysteme (1)

Native (blockdevice-basiert)

ext2	Standard
ext3	ext2 mit Journal-File
reiserfs	Journaling, b-tree
jfs	Journaling, b-tree (IBM)
xfs	Journaling, b-tree (SGI)
iso9660	ISO/Rockridge/Joliet/zISO
minix	heute eher akademisches FS

Unterstützte Dateisysteme (2)

RAM/Flash/Packet

jffs2	Journaling Flash Filesystem
romfs	ROM-FS
cramfs	komprimiertes ROM-FS
ramfs	experimentelles Ramdisk-FS
tmpfs	skalierendes Ram-FS
udf	DVD/CDRW-Packetmode FS (u.U. RW)

Unterstützte Dateisysteme (3)

Compatibility	
adfs	Acorn (Archimedes) Disc Filing System
affs	Commodore Amiga Fast File System
hfs	“Hierarchical File System“ (MAC)
bfs	“Boot File System“ (SCO Unix)
efs	Altes IRIX (SGI) Dateisystem
vxfs	VERITAS VxFS (SCO UnixWare)
ntfs	NTFS/WinFS (vorwiegend ro)
hpfs	“High Performance FS“ (OS/2)
qnx	QNX FS (vorwiegend RO)
sysv	XENIX/SCO/Coherent+PDP11
ufs	BSD/SunOS/NeXTstep FS
vfat	Altes und neues MSDOS Format

Unterstützte Dateisysteme (4)

Virtuelle

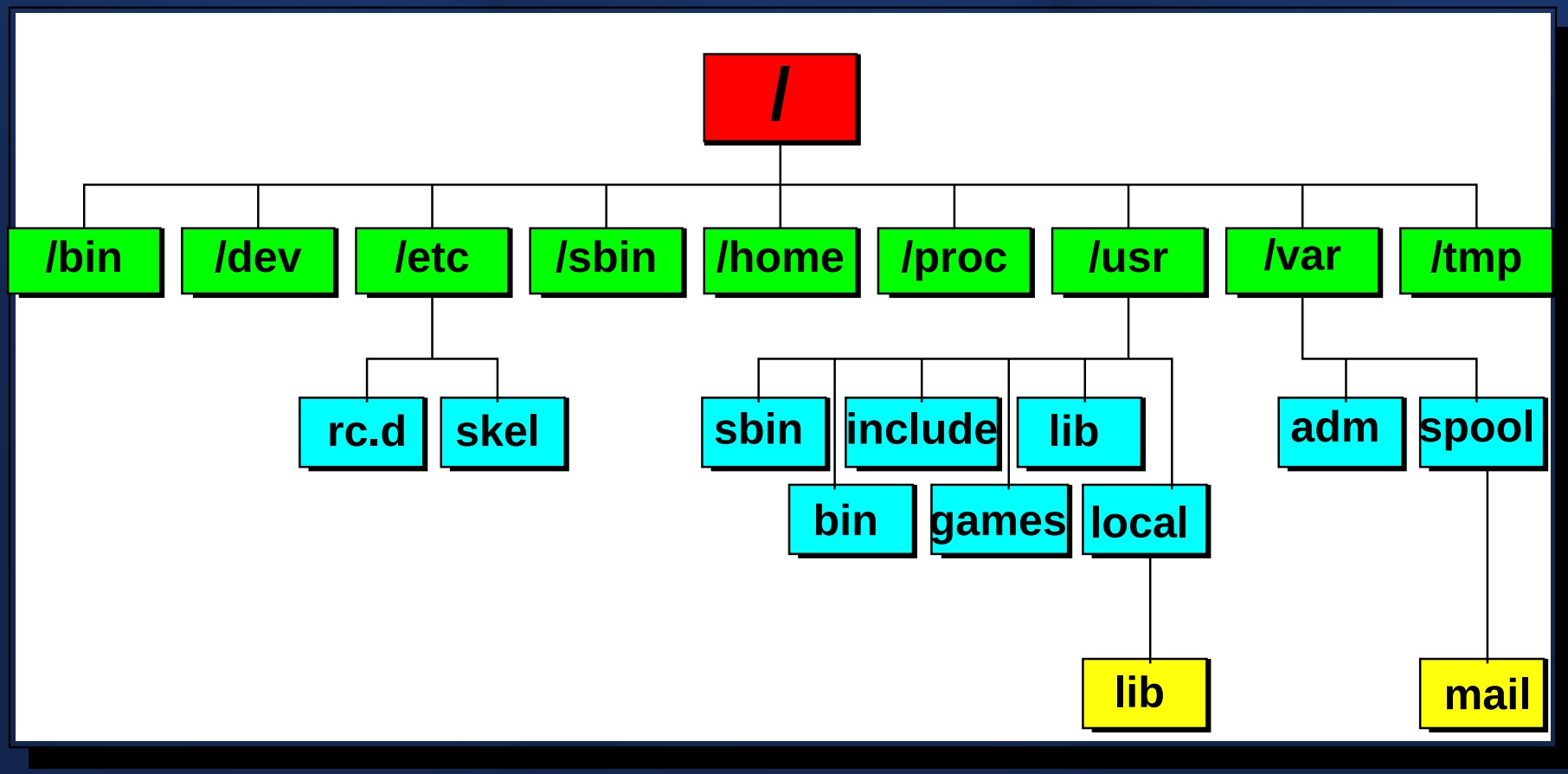
proc	virt. Kernel-Filesystem (wichtig!)
devfs	automatische Device-Generierung (RIP†)
devpts	virt. Terminals
usbfs	USB-spezifisch
capifs	ISDN-Karten

Unterstützte Dateisysteme (5)

Network

nfs	Network Filesystem v2/3 (Client+Server)
coda	NFS-ähnlich mit Replikation+Disconnect (C.)
intermezzo	NFS-ähnlich mit Disconnect (Client)
smbfs	SAMBA/LANMANAGER (Client)
ncpfs	Novell Netware (Client)

Eine Reise durch das Unix-Dateisystem



„Everything is a file“

- Gewöhnliche Dateien (Daten, Texte, Konfigurations-dateien...)
- Verweise ☞ „Links“ (hard, soft)
- Spezialdateien: **Geräte** (meist in **/dev**), named Pipes, Unix-Domainsockets, ...

Beispiel:

```
$ cat /dev/mouse
#@$#^%$&^&UYUDFGHJK$^%&*&$%^RT&Y*U^&*|
```

`mount` – Einbinden von Dateisystemen

```
$ mount -t ext2 /dev/hda2 /usr

$ mount -t iso9660 -o ro /dev/hdc /mnt/cdrom

$ mount -t vfat /dev/fd0 /mnt/floppy

$ mount /mnt/floppy

$ mount pizza:/mnt/cdrom /mnt/pizza/cdrom
```

Quota

Auf Mehrbenutzerrechnern wird den Benutzern oft nur ein bestimmter Plattenplatzverbrauch zugestanden. Dieser kann mit dem Befehl `quota` überprüft und vom Administrator mit `edquota benutzer` für jeden Benutzer festgelegt werden.

```
$ quota -v


Disk quotas for knopper (uid 26001):
 Filesystem   usage    quota      limit    timeleft    files    quota    limit    time
 /home        46451    50000    2997944         ---     3068    50000    90000      --
```

Anlegen einer ext3-Partition – Schritte

```
1    $ fdisk /dev/hdb
2    $ mke2fs -N 10000000 -b 1024 /dev/hdb1
3    $ e2fsck /dev/hdb1
4    $ tune2fs -m 0 -j /dev/hdb1
5    $ mount -t ext3 /dev/hdb1 /tmpbig
```

/etc/fstab

```
# <device>    <mountpoint>   <type>  <options> <dump> <fsck>

/dev/hda2     /              ext2    defaults  1      1
/dev/hda3     /usr           ext2    defaults  1      2
/dev/hda5     /home          ext2    defaults  1      3

pizza:/www    /mnt/pizza/www nfs     rw,hard,intr,bg 0 0

/dev/hda3     none           swap    sw
none          /proc          proc    defaults

/dev/fd0      /mnt/floppy    vfat    noauto    0      0
/dev/hdc4     /mnt/cdrom     iso9660 noauto,ro,user  0 0
```

Übung

(Überlegen oder testen Sie, für welche dieser Aktionen Sie Administratorrechte benötigen!)

1. Formatieren Sie eine 1.44MB-Diskette mit **`fdformat /dev/fd0H1440`** und legen Sie ein Dateisystem auf der Diskette an mit **`mkdosfs /dev/fd0H1440`**
2. Melden Sie die Diskette an mit **`mount -t vfat /dev/fd0H1440 /mnt/floppy`**
3. Kopieren Sie ein paar der in den vorigen Übungen erzeugten Textdateien aus Ihrem Heimverzeichnis auf die Diskette.
4. Melden Sie die Diskette wieder ab und überprüfen Sie mit Hilfe von **`mdir`**, ob das Kopieren geklappt hat.

Loopback-Device (Disk)

Direktes Einbinden von Partitions-Images als Dateisystem:

```
losetup /dev/loop2 cdrom.iso
mount -r /dev/loop2 /media/cdrom
```

oder:

```
mount -o loop,ro cdrom.iso /media/cdrom
```

Loopback-Device (Swap)

Swappen auf Dateien wird auch direkt von **mkswap** und **swapon** unterstützt, geht aber auch mit

```
dd if=/dev/zero of=/media/hda1/datei.swp \
   bs=1000k count=100
mkswap /media/hda1/datei.swp
losetup /dev/loop3 /media/hda1/datei.swp
```

Loopback-Device (Crypto)

Voraussetzungen:

1. Kernel mit crypto-lopback Support,
2. crypto-loopback-fähiges **`losetup`** und **`mount`**-Kommando.

Loopback-Device (Crypto)

Einrichten:

```
dd if=/dev/zero of=geheim.img bs=1000k \
   count=100
losetup -e AES256 /dev/loop4 geheim.img
mke2fs /dev/loop4
```

Anschließend kann **`/dev/loop4`** direkt gemountet werden.

Loopback-Device (Crypto)

Mounten (abgekürzt):

```
mount -o loop,encryption=AES256 \
  geheim.img geheim
```

Übung

1. Erzeugen Sie sich eine „virtuelle Partition“ als Image-Datei, die ein verschlüsseltes ext2-Dateisystem enthalten soll, und mounten Sie diese in ein Unterverzeichnis Ihres Heimverzeichnisses. Sorgen Sie (ggf. als Administrator) dafür, dass der Benutzer (einer) dies auch ohne Zuhilfenahme von root tun kann.

2. Werden Sie paranoid: Verschlüsseln Sie Ihre Swap-Partition (oder legen Sie eine verschlüsselte Swap-Datei an und binden Sie diese ins System als zusätzlichen Speicher ein).

Dateitypen unter Unix/Linux

In diesem Beispiel wurde ein Verzeichnis namens **`test`** angelegt, in dem sich verschiedene Dateiarten befinden.

```
knopper@Koffer:~$ ls -l test/
insgesamt 8
brw-rw---- 1 root    floppy 2,  0 2006-04-04 12:06 blockdevice
crw-rw---- 1 root    root  10,  1 2006-04-04 12:06 chardevice
-rw-r--r-- 2 knopper users      5 2006-04-04 13:14 datei.txt
drwxr-xr-x 2 knopper users     48 2006-04-04 13:17 directory
prw-r--r-- 1 knopper users      0 2006-04-04 13:16 fifo
-rw-r--r-- 2 knopper users      5 2006-04-04 13:14 hardlink
srwxrwxrwx 1 knopper users      0 2006-04-04 10:06 socket
lrwxrwxrwx 1 knopper users      9 2006-04-04 13:14 symlink
                                                   -> datei.txt
```

Der erste von **`ls -l`** angezeigte Buchstabe in den Datei-rechten kennzeichnet die Art der Datei.

Einfache Dateien

Einfache Dateien können Dokumente, Programme, Bibliotheken oder Daten jedweder Art sein.

```
-rw-r--r-- 2 knopper users     5 2006-04-04 13:14 datei.txt
```

Verzeichnisse

Verzeichnisse und Unterverzeichnisse sind ein Ordnungsmittel, um Dateien zu kategorisieren und leichter wiederauffindbar zu machen.

```
drwxr-xr-x 2 knopper users    48 2006-04-04 13:17 directory
```

Symlinks

Symbolische Links, also „Namensverknüpfungen“, sind Zeiger auf Datei- oder Verzeichnisnamen, die den Zugriff auf die entsprechenden Daten unter einem anderen Namen ermöglichen. Sie werden mit dem Kommando
`ln -s datei verknüpfung`
angelegt. Der symbolische Link und sein Ziel ist bei **`ls -l`** deutlich identifizierbar.

```
-rw-r--r-- 2 knopper users 5 2006-04-04 13:14 datei.txt
lrwxrwxrwx 1 knopper users 9 2006-04-04 13:14 symlink -> datei.txt
```

Achtung: Wird die Originaldatei gelöscht, so ist der Inhalt auch nicht mehr über den symbolischen Link verfügbar, obwohl dieser vorhanden bleibt.

Hardlinks

Ähnlich wie beim Symbolischen Link auf den NAMEN einer Datei verwiesen wird, wird beim Hardlinks auf den Datei-INHALT verwiesen. D.h. es wird mit

`ln datei hardlink`

eine neue Datei angelegt, deren Inhalt aber mit dem der ersten Datei immer identisch ist. Aus dem Verzeichnislisting ist die Tatsache, dass es sich um einen Hardlink handelt, allerdings nicht ohne weiteres erkennbar.

```
-rw-r--r-- 2 knopper users     5 2006-04-04 13:14 datei.txt
-rw-r--r-- 2 knopper users     5 2006-04-04 13:14 hardlink
```

Wird eine der beiden Dateien gelöscht, so ist der Inhalt nach wie vor unter dem anderen Dateinamen verfügbar. Originaldatei und Hardlink sind also “gleichberechtigt“.

Character und Block Devices

Character Devices erlauben sequentielles, zeichenweises Schreiben und Lesen von Daten auf die damit verbundenen Geräte (vergl. voriges Kapitel über Dateisysteme und Blockdevices).

Block Devices erlauben wahlfreien Zugriff auf beliebige „Blöcke“ eines Gerätes, ohne dass ein „Vor-“ oder „Zurückspulen“ nötig ist.

```
brw-rw---- 1 root    floppy 2, 0 2006-04-04 12:06 blockdevice
crw-rw---- 1 root    root  10, 1 2006-04-04 12:06 chardevice
```

Block Devices bilden üblicherweise Festplatten ab, während Char Devices Mäuse oder Bandlaufwerke und Modems abbilden.

Fifos und Sockets

Während Sockets mit Netzwerkoperationen (z.B. Kommunikation mit dem X-Server) verknüpft sind, sind Fifos ein Mittel, um die Ein- und Ausgabe von Programmen miteinander zu verknüpfen, ähnlich wie mit dem Pipe-Symbol |.

```
prw-r--r-- 1 knopper users     0 2006-04-04 13:16 fifo
srwxrwxrwx 1 knopper users     0 2006-04-04 10:06 socket
```

chown – Setzen des Dateibesitzers

```
chown [Optionen] Benutzer Datei(en)...
```

chown ändert das Besitzer-Attribut von Dateien und Verzeichnissen. Der chown-Befehl kann auf POSIX-konformen Unix-Systemen nur vom Systemadministrator ausgeführt werden.
Der ursprüngliche Besitzer der Datei verliert mit sofortiger Wirkung die Besitzer-Rechte an dieser Datei und kann nur noch aufgrund gesetzter Gruppen- oder globaler Rechte auf die Datei oder das Verzeichnis zugreifen.

```
chown -R demo /home/demo
```

Mit der Option `-R` kann rekursiv das Besitzerattribut ganzer Verzeichnisbäume geändert werden.

`chgrp` – Ändern der Gruppenzugehörigkeit

`chgrp [Optionen] Gruppe Dateien...`

`chgrp` ändert die Unix-Gruppe von Dateien und Verzeichnissen. Der Befehl kann vom Besitzer einer Datei ausgeführt werden, wenn er selbst Mitglied der angegebenen Unix-Gruppe ist (POSIX).

```
$ ls -l helloworld.c
-rw-r--r-- 1 knopper users     29 Aug 5 22:39 helloworld.c
$ groups
users developer
$ chgrp developer helloworld.c
$ ls -l helloworld.c
-rw-r--r-- 1 knopper developer 29 Aug 5 22:39 helloworld.c
```

chmod – Ändern von Rechten

`chmod [Optionen] Änderungen Dateien`

`chmod` ändert die Zugriffsrechte von Dateien und Verzeichnissen. Man kann die **Rechte**

r = read lesen	w = write schreiben	x = execute ausführen	s = suid set ID

an bestimmte **Personenkreise** vergeben

u = user Besitzer	g = group Gruppe	o = others Andere

Mit der Option `-R` werden die Änderungen auch für Unterverzeichnisse durchgeführt.

Beispiele zu **chmod**

```
$ ls -l
total 11
-rw-r--r--  1 knopper  users  7185 Nov 20 23:17 auswertung.sh
-rw-------  1 knopper  users   938 Nov 20 23:17 juli.dat
-rw-------  1 knopper  users   469 Nov 20 23:17 juni.dat
-rw-------  1 knopper  users    54 Nov 20 23:17 mai.dat
$ chmod u+x auswertung.sh
```

Das Script `auswertung.sh` wird zum Ausführen freigegeben.

Beispiele zu **chmod**

```
$ chmod og+r *.dat
$ ls -l
total 11
-rwxr--r--  1 knopper  users  7185 Nov 20 23:17 auswertung.sh
-rw-r--r--  1 knopper  users   938 Nov 20 23:17 juli.dat
-rw-r--r--  1 knopper  users   469 Nov 20 23:17 juni.dat
-rw-r--r--  1 knopper  users    54 Nov 20 23:17 mai.dat
```

Alle dürfen ab jetzt die „`.dat`“-Dateien lesen.

Spezielle Dateiattribute

Neben den Standard-Rechten Lesen, Schreiben und Ausführen existieren noch weitere Dateiattribute, die vom Besitzer einer Datei oder vom Systemadministrator gesetzt werden können.

```
$ chmod u+s /usr/bin/cdrecord
$ ls -l /usr/bin/cdrecord
-rwsr-xr-x 1 root root 13956 May 10 17:31 /usr/bin/cdrecord
```

Durch das Setzen des s-Attributes (“s-Bit“) für den Besitzer bzw. die Gruppe einer Datei wird beim Ausführen der Datei der Besitzer bzw. die Gruppe des neuen Prozesses auf den Besitzer bzw. die Gruppe der Datei gesetzt.

Übung

1. Kontrollieren Sie mit **`ls -l`** die gesetzten Dateiattribute der Disketten-Gerätedateien **`/dev/fd0*`**.

2. Geben Sie die Disketten-Gerätedateien für alle Benutzer zum Lesen und Schreiben frei.

3. Sorgen Sie dafür, dass Ihr selbst eingerichteter Normalbenutzer CDs brennen kann, indem Sie ihn in die Benutzergruppe aufnehmen, welche mit dem Programm **`cdrecord`** auf das CD-Brenner-Gerät schreiben kann.

/etc/modules.conf konfigurieren

```
### /etc/modules.conf
alias eth0 wd
options wd io=0x360 irq=5
```

Mit **modprobe eth0** oder beim ersten Zugriff auf die erste Netzwerkkarte wird das passende Modul mit den angegebenen Optionen geladen.

WLAN-Karten (Hardware)

PCI-Karten: Wie Netzwerkkarten (LAN)

PCMCIA-Adapter: Über `cardmgr` (**`/etc/pcmcia/wireless.conf`**) oder `hotplug/udev` automatisch, sofern Module und Firmware vorhanden. Notfalls mit **`ndiswrapper`** den Windows-Treiber laden.

USB-Adapter: Über `hotplug/udev` automatisch, sofern Module und Firmware vorhanden. Notfalls mit **`ndiswrapper`** den Windows-Treiber laden.

Ethernet-Parameter einstellen

Bringt viele Administratoren MAC-basierter Firewalls zur Verzweiflung:

```
ifconfig eth0 down
ifconfig eth0 hw ether 00:04:23:44:22:11
```

Hiermit wird die „Hardware-Adresse“ von Netzwerkkarten eingestellt.

Ethernet-Parameter (WLAN) einstellen

```
iwconfig interface [essid {NN|on|off}]
                   [mode {managed|ad-hoc|...}
                   [channel N]
                   [ap {N|off|auto}]
                   [key {NNNN-NNNN|off}]
```

Z.B.: `iwconfig eth1 essid "fhzw" key 123456789ABCDEF01234567890`

Ethernet-Parameter (WLAN) einstellen

Einige Karten (v.a. Prism2) werden nur von „wlan-ng“ (WLAN Next Generation) unterstützt, das eine wesentlich kompliziertere Syntax hat.

Wer WPA statt WEP zur Authentifizierung/Verschlüsselung einsetzen will oder muss, kann auf den **`wpasupplicant`** zurückgreifen.

ifconfig – IP-Adresse und Netzmaske

```
ifconfig eth0 192.168.0.1
       netmask 255.255.255.0
       broadcast 192.168.0.255

ifconfig eth0
```

```
eth0  Link encap:10Mbps Ethernet  HWaddr 00:00:C0:68:FB:29
      inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0
      TX packets:0 errors:0 dropped:0 overruns:0
      Interrupt:5 Base address:0x310 Memory:ca000-cc000
```

route – Netzwerkrouten und Gateway(s)

route add -net 192.168.1.0
netmask 255.255.255.0 dev eth0

Setzt eine Route zum Netzwerk 192.168.1.0 auf die gleiche Netzwerkkarte wie vorher 192.168.0.0. Es muß allerdings ggf. vorher eine zweite lokale IP-Adresse auf dem Interface **eth0:1** gesetzt werden, die diesem Netz entspricht.

route add default gw 192.168.0.254

Setzt das „Tor zur Welt“ über den Rechner mit der IP-Adresse 192.168.0.254.

/etc/resolv.conf – Nameserver

In der Datei **/etc/resolv.conf** werden mit dem vorangestellten Schlüsselwort **nameserver** die IP-Adressen der Nameserver angegeben, die befragt werden sollen, wenn Ihr Rechner versucht, einen DNS-Namen aufzulösen (DNS = „Domain Name System“ oder „Service“). Fehlt dieser Eintrag, so kann lediglich über die numerische Adresse, nicht aber über Rechnernamen, auf andere Rechner im Internet zugegriffen werden.

Dynamische Dienste wie **DHCP** (**pump**, **dhclient**, **dhcpcd**) oder **PPP** setzen bei erfolgreichem Verbindungsaufbau automatisch einen gültigen Nameserver in **/etc/resolv.conf** ein. Ein Programm, mit der Sie überprüfen können, ob Ihr Nameserver korrekt arbeitet, ist **nslookup** (interaktiv) oder **host rechnername** (nicht-interaktiv).

Übung

1. Setzen Sie mit **`ifconfig`** eine zweite IP-Adresse auf das virtuelle Netzwerkkarten-Interface **`eth0:1`**, welche eine um 1 erhöhte Netzwerkadresse (nicht Host-Adresse!) besitzen soll.
2. Sehen Sie mit **`route`** nach, ob der Kernel automatisch eine Netzwerkroute für dieses Interface angelegt hat.
3. Versuchen Sie mit **`ping`**, das zweite, virtuelle Interface Ihres Tischnachbarn “anzupingen“.
4. Sehen Sie mit dem Kommando **`traceroute IP-Adresse`** nach, welchen Weg Ihre IP-Pakete nehmen, wenn Sie einen bestimmten Rechner außerhalb des lokalen Netzes zu erreichen versuchen.

Eigenschaften eines TCP/IP-Paketes

- SOURCE (Herkunfts-) **Adresse**,
- DESTINATION (Ziel-) **Adresse**,
- SOURCE (Herkunfts-) **Port**,
- DESTINATION (Ziel-) **Port**,
- Protokolltyp (**TCP** oder **UDP**).

Der SOURCE-Port auf einem Server kennzeichnet i.d.R. den angesprochenen **Dienst** (s.a. **`/etc/services`**).

Netzdienste starten

- Einen Server-Dienst starten, der sich (gemäß seiner Einstellungen) auf einen bestimmten **Port** bindet, oder
- Mit Hilfe des Internet-Metadämons **`inetd`** einen Dienst oder ein Programm mit einem wählbaren **Port** verbinden.

/etc/inetd.conf

```
# inetd.conf
#       This file describes the services that will be available
#       through the INETD TCP/IP super server.  To re-configure
#       the running INETD process, edit this file, then send the
#       INETD process a SIGHUP signal.
#
# <service_name> <sock_type> <proto> <flags> <user> <ppath> <args>
#
finger  stream  tcp    nowait  root  /usr/sbin/in.fingerd fingerd
ftp     stream  tcp    nowait  root  /usr/sbin/tcpd  in.ftpd -l -a
telnet  stream  tcp    nowait  root  /usr/sbin/tcpd  in.telnetd
shell   stream  tcp    nowait  root  /usr/sbin/tcpd  in.rshd
talk    dgram   udp    wait    root  /usr/sbin/tcpd  in.talkd
```

Übung

1. Aktivieren Sie den **pop3**- und den **ftp**-Server in **/etc/inetd.conf**, indem Sie die ggf. vor den entsprechenden Zeilen stehenden Kommentarzeichen # entfernen, **/etc/inetd.conf** abspeichern und dem **inetd**-Prozess ein **HUP**-Signal senden (z.B. mit **killall -HUP inetd**, wenn Sie nicht die Prozess-ID des inetd-Servers nachsehen wollen).

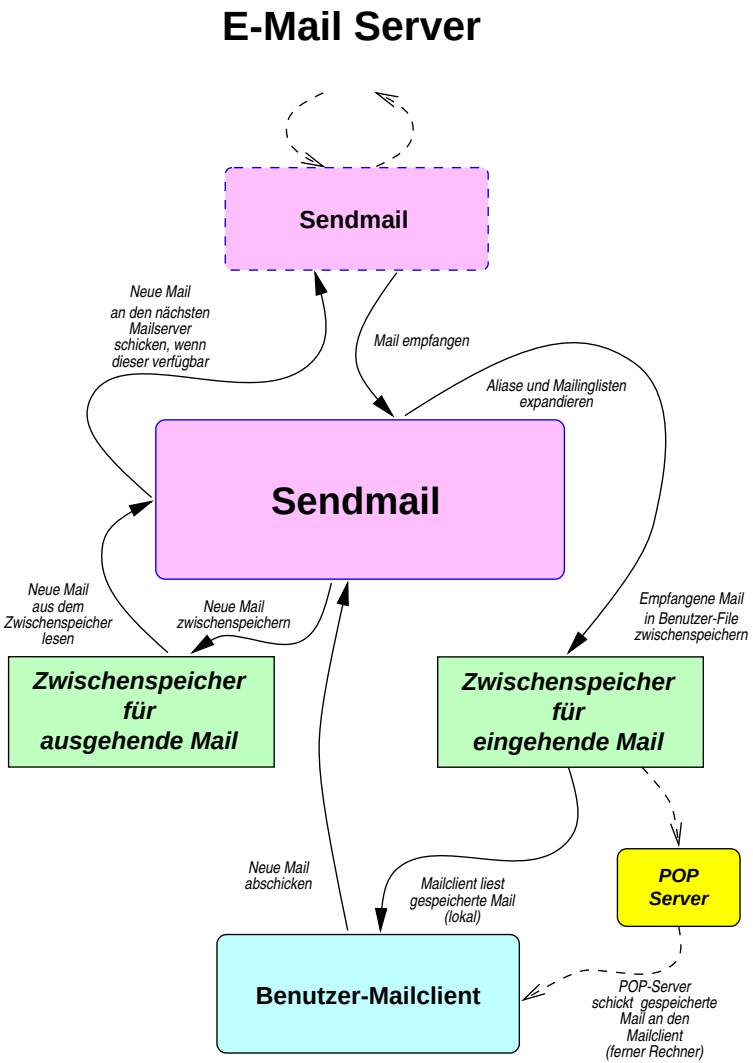
2. Testen Sie das Vorhandensein der nun freigeschalteten Dienste, indem Sie mit dem Universalclient **telnet localhost portnummer** eine Verbindung zu Ihrem eigenen Rechner aufbauen.

Übung: sendmail aktivieren

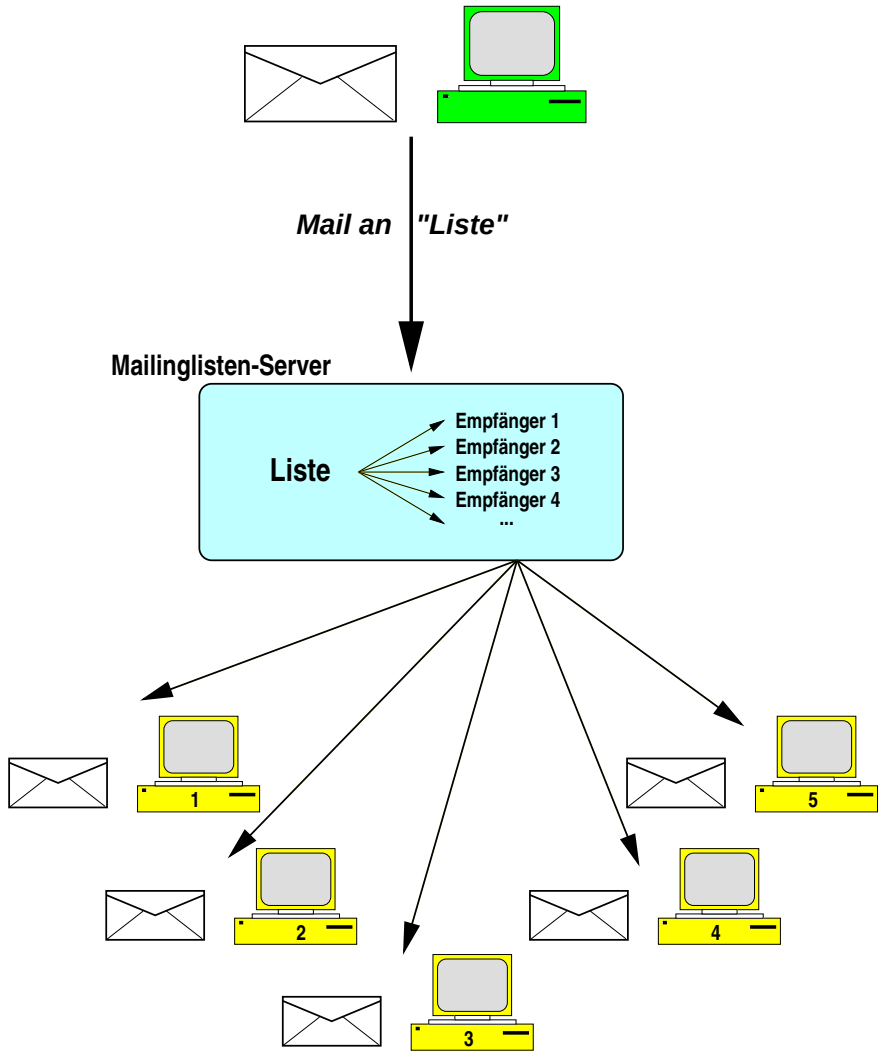
1. Konfigurieren Sie mit **yast** oder **webmin** den Mail-Versanddienst **sendmail** für die direkte Zustellung im lokaen Netzwerk, und starten Sie den Dienst, indem Sie das Skript **/etc/init.d/sendmail** mit dem Parameter **start** aufrufen.

2. Testen Sie sendmail, indem Sie **telnet localhost smtp** starten, und bei erfolgreicher Verbindung, das SMTP-**HELP** und **QUIT**-Kommando eingeben.

Sendmail (Schema)



Mailinglisten mit Majordomo



Übung: Majordomo

Wenn Majordomo erfolgreich auf Ihrem System eingerichtet wurde, finden Sie die (Test-)Mailinglisten im Verzeichnis **`/var/lib/majordomo/lists`**. Dies sind gewöhnliche Textdateien, die Sie z.B. mit **`vi`**, oder mit **webmin** verändern können.

In **`/etc/mail/aliases`** finden Sie die Verweise auf Majordomo-Listen. Diese Datei ist Bestandteil des **`sendmail`**-Systems, und ermöglicht eine Zuordnung zwischen Mailadressen und Benutzern (oder, wie im Fall von Majordomo, von Programmen).

NFS – Serverdienste

Um einen NFS-Server zu betreiben, ist neben dem RPC-Portmapper **`portmap`** mindestens der Start der folgenden Server notwendig:

`rpc.mountd` Kontrolliert und autorisiert **`mount`**-Zugriffe auf die freigegebenen Verzeichnisse aufgrund der Einstellungen in **`/etc/exports`**.

`rpc.nfsd` Ist in einer oder mehreren Instanzen für die tatsächliche Datenübertragung verantwortlich.

Debian: `/etc/init.d/portmap start ; /etc/init.d/nfs-kernel-server start`

Auf der Client-Seite ist nur ein NFS-fähiger Kernel und das **`mount`**-Kommando erforderlich!

`/etc/exports` – Dateisysteme exportieren

Die einzige Konfigurationsdatei auf NFS-Server-Seite ist **`/etc/exports`**, welche die *Namen* der exportierten Verzeichnisse sowie die erlaubten *IP-* oder *Netzwerkadressen* der Clients enthält.

```
/mnt/cdrom pizza(ro,nosuid,nodev)
/usr       lasagne(ro)          sushi(ro)
/tmp       lasagne(rw,no_root_squash)
```

mounten von NFS-Dateisystemen

Das **mount**-Kommando kennt einige Erweiterungsoptionen für NFS-Dateisysteme, die gerade beim automatischen Mounten von NFS-Verzeichnissen beim Systemstart nützlich sein können.

```
mount -o bg,hard,intr lasagne:/tmp /mnt/tempnfs
```

bg „Background“, wenn das Remote-System momentan nicht erreichbar ist, legt sich das **mount**-Kommando nach kurzem Timeout in den Hintergrund und versucht weiter, das NFS-Verzeichnis einzubinden.

hard Netzausfälle führen nicht zu Schreibfehlern, sondern es wird so lange gewartet, bis das NFS-Dateisystem wieder zur Verfügung steht. Notfalls unendlich lange.

intr Schreib-/Lesezugriffe auf NFS dürfen durch Steuerung-C oder Signale unterbrochen werden.

Übung

1. Exportieren Sie Ihr **`/tmp`**-Verzeichnis read-only an ALLE.
2. Teilen Sie diese Änderung den NFS-Diensten durch Aufruf von **`exportfs -a`** mit. Falls diese noch nicht gestartet wurden, starten Sie sie manuell durch Aufruf von **`/etc/init.d/nfs-server start`**.
3. Überprüfen Sie die exportierten Verzeicnisse mit **`showmount -e`**.
4. Versuchen Sie, die von Ihrem Tischnachbarn freigegebenen Verzeichnisse auf Ihren eigenen Rechner unter dem Pfad **`/mnt/nfs`** zu mounten (Sie müssen dieses Verzeichnis vorher anlegen).

Übung – Windows-Heimverzeichnis mounten

1. Versuchen Sie, Ihr Windows-Heimverzeichnis auf Ihren Linux-Client einzubinden! Im folgenden Beispiel ist **`benutzername`** Ihr Windows-Login-Name. Nach dem zugehörigen Passwort werden Sie beim Mounten gefragt.

```
cd
mkdir fh-home
su
mount -t cifs -o user=benutzername,domain=DS \
  //zwo222-fs1.ds.fh-kl.de/benutzername\$ fh-home
```

2. Bauen Sie eine „Abkürzung“ in /etc/fstab ein, so dass der Benutzer in Zukunft nur noch **`mount fh-home`** in seinem Heimverzeichnis tippen muss, ohne root zu werden.

Übung – SAMBA-Freigaben

1. Überprüfen Sie, ob die Benutzer-Heimverzeichnisse (Samba-Sharename **`[homes]`**) mit oder ohne Anmeldung freigegeben werden in der **`/etc/samba/smb.conf`**.
2. Geben Sie Ihr CDROM-Verzeichnis für ALLE, OHNE PASSWORT unter dem Sharenamen **`[public]`** frei.
3. Überprüfen Sie die SAMBA-Konfiguration mit **`testparm`**.
4. Starten Sie den SAMBA-Server über sein Init-Skript, und überprüfen Sie mit Hilfe von konqueror (**`smb:/`**), ob Ihr Samba-Server im Netz sichtbar ist. Falls er nicht öffentlich freigegeben ist (**`browsable`**-Direktive), versuchen Sie die Adresse **`smb://localhost`**.

SSH

- kann eine Verbindung zwischen zwei Rechnern verschlüsseln,
- kann über **Public Key Verfahren** (**??**) authentifizieren/einloggen,
- kann beliebige Ports und Dienste über die verschlüsselte Verbindung tunneln (**-L** und **-R** Optionen),
- kann auch graphische Programme (X11) über eine authentifizierte, verschlüsselte Verbindung tunneln,
- kann einige Dienste (FTP, rsync, rsh) in einer verschlüsselten Variante zur Verfügung stellen.

Passwortlose Authentifizierung per SSH

1. Keypaar (einmalig!) erzeugen:

```
ssh-keygen -t dsa
```

2. Public Key (**`.ssh/id_dsa.pub`**) auf Remote-System in **`.ssh/authorized_keys`** eintragen.

3. Host-Keys in Hin- und Rückrichtung austauschen (ssh-connect).

Graphische Programme remote starten

```
ssh -X user@remotehost gimp
```

Man beachte auch den durch **`-X`** auf dem **`remotehost`** geöffneten Tunnel-Port, auf den die Variable **`DISPLAY`** verweist.

vanced: Verschlüsselen Proxy-Zugang einrich

```
ssh -C -L 3128:localhost:3128 user@proxy-host
```

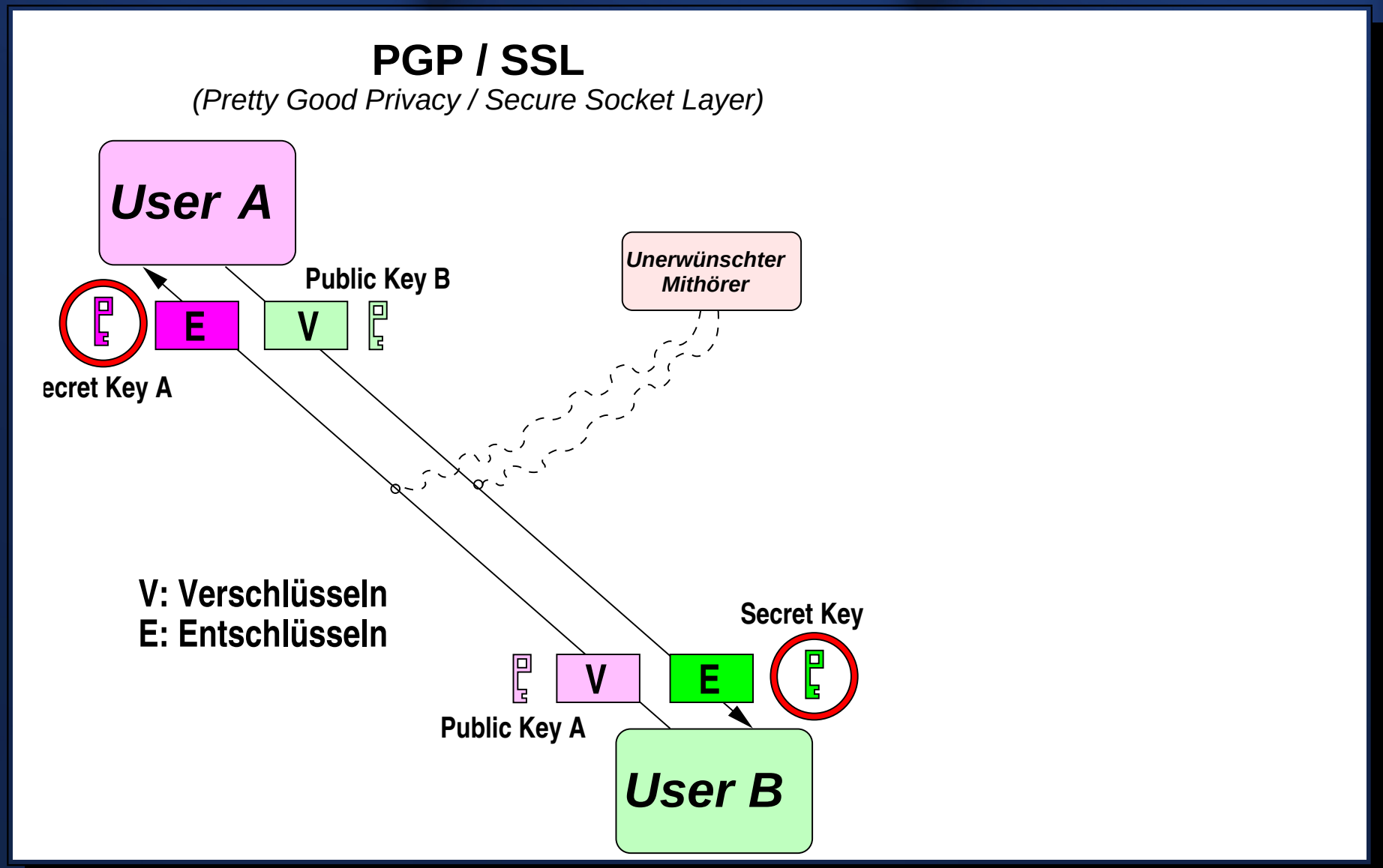
`-L localport:remotehostip:remoteport` öffnet einen lokalen Port auf dem Client, der über eine verschlüsselte SSH-Session mit dem **`remoteport`** auf der **`remotehostip`**-Adresse verbunden wird. Die Datenüberragung ist wegen **`-C`** zusätzlich komprimiert.

Eine Verbindung auf dem SSH-Clientrechner zu dessen Port **`3128`** nach dem o.a. Kommando würde also mit dem Port **`3128`** auf dem entfernten Rechner Kontakt aufnehmen, wodurch man z.B. einen Proxy im Außennetz ansprechen kann, wenn man im lokalen Browser **`localhost:3128`** als Proxy-Adresse einträgt.

Übung zur SSH

1. Erzeugen Sie sich ein Public/Secret-Key Paar für SSH, und konfigurieren Sie den passwortlosen Remote-Login damit (testen können Sie dies in der nächsten Aufgabe).
2. Starten Sie als root **`sshd`** und loggen Sie sich (als Normalbenutzer) auf dem eigenen Rechner per SSH ein.
3. Wiederholen Sie die letzte Aufgabe mit einer neuen Shell, tunneln Sie diesmal alle X11-Verbindungen, sowie Port 25 des „Remote-Rechners“ auf Port 2500 Ihres „Client-Rechners“. Starten Sie den sendmail-Daemon und testen Sie, ob Sie ein connect zum Remote-Sendmail über Port 2500 bekommen (**`telnet localhost 25`**).

SSL - Secure Socket Layer



Übung

Mit

```
openssl req -new -x509 -nodes \
        -out cert.crt -keyout cert.key
```

erzeugen Sie ein SSL-Zertifikat für Apache. Wichtig ist hierbei der **Common Name**, der den nach außen gültigen DNS-Namen (oder die IP-Adresse) des WWW-Servers enthalten muss. Kopieren Sie anschließend das neue Zertifikat an die Stelle, die in den Direktiven **`SSLCertificateFile`** und **`SSLCertificateKeyFile`** in der Apache-Konfiguration angegeben ist. Starten Sie Apache neu und testen Sie, ob Sie über **`https:`** nun einen neuen Public-Key in Netscape von Ihrem Apache-Server erhalten.

`ps` – Prozessinformationen anzeigen

`ps` `[Optionen]`

`ps` zeigt die Liste der laufenden Prozesse (=Programme) an. Das Kommando ist insbesondere im Zusammenspiel mit `kill` sehr praktisch, um die Prozess-ID „amoklaufender“ Programme zu erfahren und diese „gewaltsam“ zu beenden.

```
$ ps aux
USER       PID %CPU %MEM   SIZE   RSS TTY STAT START   TIME COMMAND
root         1  0.0  0.0    796   128  ?  S    Jun 12   0:58 init
root         2  0.0  0.0      0     0  ?  SW   Jun 12   3:45 (kflushd)
root         3  0.0  0.0      0     0  ?  SW   Jun 12   0:00 (kpiod)
root         4  0.0  0.0      0     0  ?  SW   Jun 12   6:53 (kswapd)
root       296  0.2  0.0    820   188  ?  S    Jun 12 101:09 syslogd
knopper   4832  0.1  0.8   3044  2148  pf S     23:09   0:30 gv unixkurs.ps
knopper   4936  0.0  0.3   1436   996  q3 S     23:15   0:01 vi unixkurs.tex
```

`kill` – Signal an Prozess schicken

`kill` `[Signal] Prozeßnummer`

`kill` versendet Signale an einen laufenden Prozess. Wenn `kill` ohne die Signal-Option ausgeführt wird, wird der angeführte Prozeß mit dem Signal `TERM` beendet. Die Signal-Option `-HUP` (Hang Up) gibt dem Programm die Möglichkeit, sich „sauber“ zu beenden, und dient bei einigen Systemprozessen (daemons) dazu, Konfigurationsdateien neu einzulesen. Bei hartnäckigeren Fällen hilft das Signal `-KILL`, gegen das sich kein Prozess wehren kann.

```
$ kill 1234
$ kill –KILL 1233
```

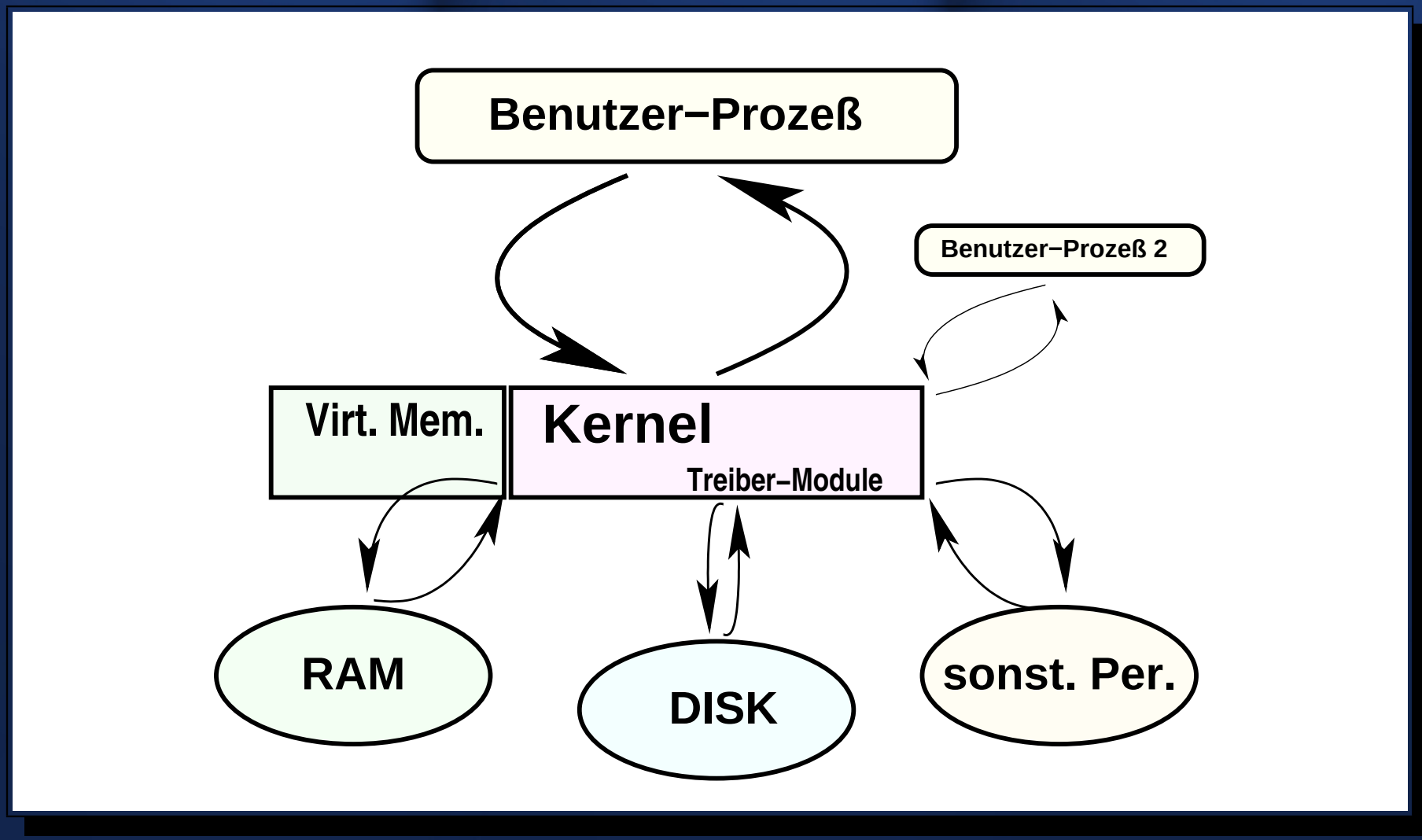
Übung

1. Starten Sie das Programm **`xclock`** in der Shell als *Hintergrundprozess*.
2. Kontrollieren Sie mit dem Shell-internen Kommando **`jobs`** Ihre Hintergrundprozesse.
3. Finden Sie die Prozess-ID des laufenden `xclock` mit Hilfe von **`ps aux | grep xclock`**.
4. Halten Sie den Prozess kurzfristig an, indem Sie ihm mit **`kill -STOP Prozess-ID`** ein STOP-Signal schicken. Beobachten Sie die Reaktion des Programms.
5. Lassen Sie den Prozess weiterlaufen, indem Sie ihm das Signal **`CONT`** schicken.
6. Terminieren Sie den Prozess mit dem Signal **`TERM`**.

Der Linux-Kernel

- kontrolliert und steuert alle Zugriffe von Anwender-software auf Hardware und Peripheriegeräte,
- enthält den Prozess-Scheduler (multitasking),
- verwaltet alle Datenträger und bildet deren Inhalt auf Verzeichnisse ab,
- verwaltet „echtes“ und virtuelles RAM,
- regelt die Kommunikation zwischen Prozessen,
- enthält den Netzwerk-Stack und beherrscht die zu-grundeliegenden Netzwerk-Protokolle,
- bestimmt die Stabilität des Gesamtsystems.

Kernel-Schema



Übung

1. Ändern Sie (als Administrator) rekursiv (d.h. inclusive Unterverzeichnissen) die Besitzrechte des Verzeichnisses **`/usr/src/linux/.`** auf die Benutzerkennung Ihres normalen Benutzeraccounts.

2. Wechseln Sie (als Normalbenutzer) in das Verzeichnis **`/usr/src/linux/.`** und führen Sie **`make xconfig`** aus.

3. Sehen Sie sich ein wenig in den Kerneloptionen um. Sie können auch Änderungen durchführen, brauchen diese jedoch nicht zu speichern.

Netzwerkkarten-Modul aktivieren

1. Starten Sie (als Normalbenutzer) die Kernel-Konfiguration im Textmodus mit **`cd /usr/src/linux`** und **`make menuconfig`**.

2. Suchen Sie nach dem Modul für eine ältere **WD800x**-Netzwerkkarte. Dieses verbirgt sich unter dem normalerweise inaktiven Punkt „other ISA cards“ im Netzwerkkarten-Konfigurationsmenü.

3. Aktivieren Sie die Unterstützung für diese Karte als **Modul**, speichern Sie die Einstellung und starten Sie die Kernel-Übersetzung mit **`make bzImage modules`**.

Bootfähige Diskette mit Kernel-Image

```
cd /usr/src/linux
dd if=arch/i386/boot/bzImage of=/dev/fd0
```

erzeugt eine bootfähige Diskette für i386-basierte PCs. Mit

```
rdev /dev/fd0 /dev/hda1
```

können Sie nachträglich einstellen, dass das Wurzelverzeichnis Ihres Linux-Systems sich auf der ersten Partition der ersten Festplatte am ersten IDE-Controller befindet.

Der Logger-Dämon **`syslogd`**

- nimmt Statusmeldungen von verschiedenen Systemdiensten sowie Kernel-Meldungen entgegen,
- schreibt diese in Logdateien unter **`/var/log`**, per Mail an Benutzer, auf lokale Konsolen oder schickt sie übers das Netz an weitere *syslogd*-Prozesse,
- kann über die Einstellungen in **`/etc/syslog.conf`** entscheiden, **welche** (subsystem, loglevel) Meldungen **wohin** geschickt werden.

Übung

Stellen Sie mit dem Kommando

```
grep pop /var/log/* | less
```

fest, in welcher Logdatei pop-3 Verbindungen (also Versuche, Mail von Ihrem Rechner abzurufen) mitprotokolliert wurden, und wann zuletzt eine solche pop-3 Verbindung stattgefunden hat.

Übung

Sehen Sie sich die Datei **`/var/log/messages`** mit dem Kommando **`less`** etwas genauer an. Können Sie feststellen, wann der Rechner das letzte Mal neu gebootet wurde, und was dabei im Detail ablief?
Vergleichen Sie die entsprechenden Meldungen mit der Ausgabe des Kernel-Logbuffers, die Sie mit dem Kommando **`dmesg`** erhalten.

Übung

Stellen Sie in **`/etc/syslog.conf`** ein, dass zusätzlich ALLE, auch die „unwichtigen“ Systemmeldungen (**`*.*`**) auf die Textkonsole 12 (auf diese können Sie, wie Sie sich vielleicht erinnern, mit Steuerung-Alt-F12 umschalten, zurück zu X11 finden Sie mit Steuerung-F7) ausgegeben werden. Auch der **`syslogd`** verwendet das Signal **`HUP`**, um von der Änderung informiert zu werden und diese zu beachten.

crontab

richtet eine **cron**-**Tab**elle ein, in der die Zeiten und Kommandos angegeben sind, die von **crond** automatisch abgearbeitet werden sollen. Das am Ende jeder Zeile angegebene Kommando wird mit der Benutzer-ID ausgeführt, unter der **crontab** aufgerufen wurde.

Vorsicht: Wird **crontab** ohne Argumente aufgerufen, so überschreibt es eine eventuell existierende Einstellung mit den Eingaben von Tastatur, auf die es beharrlich bis zur Eingabe von Steuerung-C (Abbruch) oder Steuerung-D wartet!

Der komfortabelste Aufruf ist **crontab -e**, wodurch die aktuellen Einstellungen, sofern existent, in den **vi** geladen und bearbeitet werden können.

Übung

Richten Sie sich mit **`crontab`** eine Zeittabelle ein, in der zu jeder vollen Stunde einmal die Logdatei **`/var/log/auth.log`** auf fehlgeschlagene Login-Versuche hin untersucht wird. Da diese Logdatei i.A. nur vom Administrator lesbar ist, müssen Sie dies ausnahmsweise als **`root`** tun. Tragen Sie mit dem Kommando **`crontab -e`** folgende Tabelle ein:

```
0 * * * * grep fail /var/log/auth.log
```

und speichern Sie sie mit dem vi wie gewohnt mit **`:wq`** .
Sie (bzw. **`root`**) sollten nun zu jeder vollen Stunde eine E-Mail mit einer entsprechenden Ausgabe (oder Fehlermeldung) erhalten.

Übung

In der vorangegangenen Aufgabe hat die eingetragene Zeile den Nachteil, dass auch bei unverändertem Stand der Dinge immer eine Meldung erzeugt wird. Das folgende Shellskript, das Sie unter dem Namen **/usr/local/bin/checklogins** speichern und in der *crontab* anstelle von *grep* aufrufen können, schafft hier Abhilfe. Vergessen Sie nicht, das Skript nach dem Anlegen mit **chmod +x** ausführbar zu machen!

```
#!/bin/sh
grep fail /var/log/auth.log > /var/log/failed.new
touch /var/log/failed.old
if test -s /var/log/failed.new; then
 diff -N /var/log/failed.old /var/log/failed.new
fi
mv /var/log/failed.new /var/log/failed.old
exit 0
```