

Vorbereitung auf LPI 101

Contents

1 Linux installieren	1
1.1 Software aktualisieren, Gasterweiterungen installieren	2
2 Login und kleiner Rundgang	2
3 Grundlegende Konzepte	3
3.1 Benutzerschnittstellen	3
4 Topic 101.1 - Determine and configure hardware settings	4
4.1 Fragen zur Vertiefung	7
5 Topic 101.2 - Boot the system	8
5.1 Typischer Ablauf eines Linux-Bootvorganges	8
5.2 Kernel-Parameter	9
5.3 Initiale RAM-Disk	9
5.4 Startprotokollierung	10
5.5 Init-System, Runlevel	10
6 Topic 101.3 - Change runlevels / boot targets and shutdown or reboot system	12
6.1 Systemstart nach System V (SysVinit)	12
6.2 Runlevel	12
6.3 Rechner herunterfahren / neu starten	12
6.4 Systemstart mit dem neuen Systemd	13
7 Topic 102.1 - Design hard disk layout	13
7.1 Partitionierung	14
7.2 Logical Volume Manager	14
8 Topic 102.2 - Install a boot manager	16
8.1 Unterschiede zwischen GRUB 1 und GRUB 2	16
8.1.1 Der veraltete (legacy) GRUB 1	16
8.1.2 Der neue GRUB 2	17
8.2 Gemeinsamkeiten der beiden GRUB-Versionen:	17
9 Topic 102.3 - Manage shared libraries	18
9.1 Abhängigkeiten ausgeben mit ldd	18
9.2 Den Cache erzeugen mit ldconfig	19
9.3 Konfiguration der Suchpfade für Libraries	19
9.4 Neuerstellen des Caches und Prüfen des Inhaltes	20
9.5 Einfaches Bereitstellen von Libraries via Variable	20

10 Topic 102.4 - Use Debian package management	20
10.1 Das Debian Backend dpkg	20
10.2 Frontends für Debian	22
10.2.1 Konfiguration der apt-Repositories	22
10.2.2 Beispiel Installation u. Rekonfiguration des LDAP-Servers	23
11 Topic 102.5 - Use RPM and YUM package management	23
11.1 Bereitstellen einer VM für Praxisübungen	23
11.2 RedHat Package Manager	23
11.3 Archiv-Format	23
11.4 Nicht installierte Pakete erforschen	24
11.5 Shell-Mitschnitt	24
11.6 Kommandozeilenfrontend yum	25
11.7 yum-Spezialität - Paketgruppen	26
11.8 VBox Gasterweiterungen installieren	26
12 Topic 103.1 - Work on the command line	26
12.1 Konfigurationsdateien der Shell	27
12.1.1 Unterscheidung zwischen alten Shells und der neuen Standardshell: bash	27
12.1.2 Startup-Dateien für User-Profiles	27
12.2 man-Pages	28
12.3 Funktionsweise	28
13 Topic 103.2 - Textströme mit Filtern verarbeiten	29
13.1 Komplettes Ausgeben von Dateien	29
13.2 Zeilen filtern	30
13.3 Zeilen nummerieren	30
13.4 Pager (Textdateien seitenweise lesen)	30
13.5 Tabulatorschritte in Leerzeichen umwandeln	31
14 Topic 103.3 - Perform basic file management	34
14.1 Kommandos für Dateioperationen	34
14.1.1 Kommando "ls"	34
14.1.2 Kommando "cp"	35
14.1.3 Kommando "mv"	36
14.1.4 Kommando "rm"	36
14.2 Kommandos für Verzeichnisoperationen	36
14.2.1 Kommandos <i>mkdir</i> und <i>rmdir</i>	36
14.3 Weitere wichtige Kommandos für Dateiarbeit:	37
14.4 Wildcards für Dateinamen und -pfade (S. 117 f)	37

14.5	Wichtige Kommandos für Archivierung	38
14.5.1	tar - Für Tapes geschaffenes Archivierungsprogramm:	38
14.5.2	cpio (S. Seite 121, über STDIN / STDOUT kopieren): Einsatz beim rpm-Paketformat (rpm2cpio), es kennt drei Modi:	38
14.6	Wichtige Kommandos für Komprimierung	38
14.7	Archivierung mit Komprimierung verbinden	39
15	Topic 103.4 - Use streams, pipes and redirects	40
15.1	Standard-Datenströme der Unix-Shells	40
15.2	Pipelines zum Weiterverarbeiten der Datenströme	41
15.3	Umleitungen der Standardkanäle durchführen	41
15.4	Weitere Verarbeitungskürzel der bash	42
16	Topic 103.5 - Create, monitor and kill processes	42
16.1	Überwachen von Prozessen	43
16.2	Arbeiten mit Signalen	44
16.3	Weitere Tools zur Prozessmanipulation	44
16.4	Jobverwaltung der bash	45
17	Topic 103.6 - Modify process execution priorities	46
17.1	Optionen und Aufrufmöglichkeiten:	48
18	Topic 103.7 - Search text files using regular expressions	49
18.1	Der streaming Editor sed	51
19	Topic 103.8 - Perform basic file editing operations using vi	51
20	Topic 104.1 - Create partitions and filesystems	52
20.1	Ein kleiner Ausflug in die Dateitypen	52
20.2	Partitionierung mit fdisk:	53
20.2.1	Swap-Space erweitern	56
20.2.2	Dateisystem erzeugen und benutzen	57
20.2.3	Neue Partition für /opt verwenden	59
20.2.4	Weitere Aufgaben	61
21	Topic 104.2 - Maintain the integrity of filesystems	64
21.1	Wichtige Tools für den Dateisystem-Check	64
21.2	Tools für das Tuning und für Recovery	65
22	Topic 104.3 - Control mounting and unmounting of filesystems	66

23 Topic 104.4 - Manage disk quotas	67
23.1 Software installieren	67
23.2 Datei /etc/fstab anpassen	67
23.3 Prüfen und einrichten der Quotierung	67
23.4 Quotas aktivieren	68
23.5 Ausflug ins Thema Benutzerverwaltung	68
23.6 Quotas für User/Group editieren	69
23.7 Kontrolle der Einstellungen	70
23.8 Sicherheitshinweis	70
24 Topic 104.5 - Manage file permissions and ownership	70
24.1 Zugriffsrechte im symbolischen Modus	70
24.2 Zugriffsrechte für Sonderfälle	70
24.2.1 Set User ID - mit Besitzerrechten ausführen	71
24.2.2 Set Group ID - mit Gruppenrechten ausführen	71
24.2.3 Sticky Bit - nur der Besitzer/Ersteller darf modifizieren	72
24.3 Standardrechte für neue Files	73
24.4 Praxisübung	74
24.5 Eigentümerschaft von Files ändern	74
24.6 Einen Nutzer ohne Werkzeuge wie <i>useradd</i> und <i>passwd</i> aus dem Standard-Nutzer <i>tux</i> erzeugen	75
24.7 Dateiattribute als weitere Möglichkeit, Berechtigungen zu verwalten	77
25 Topic 104.6 - Create and change hard and symbolic links	77
25.1 Praxisbeispiele zu Hardlinks	78
25.2 Praxisbeispiele zu symbolischen Links	79
26 Topic 104.7 - Find system files and place files in the correct location	80
26.1 Filesystem Hierarchy Standard (FHS)	80
26.2 Suchen nach Dateisystemobjekten	80
26.3 Einen eigenen Index als einfacher Benutzer erzeugen und nutzen	81

1 Linux installieren

Voraussetzung: Downloaden einer ISO-Datei, die im besten Falle für beides erhalten kann: für ein schnelles Arbeiten und Ausprobieren, ohne dass man die Festplatte berühren muss (Live-System) sowie für eine feste Installation. Solche Images findet man (was Debian betrifft) unter:

<http://cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/>

Die Auswahl an ISO-Images ist hier groß, empfehlenswert ist zuerst einmal ein System mit dem schlanken und reich mit Zubehörprogrammen ausgestatteten Desktop LXDE. Debian liegt aktuell (Stand: 28. Juli 2016) in Version 8.5 vor, die entsprechende Datei ist 959M groß. **Bitte hier herunterladen...**

In der bereits installierten Virtualisierungsumgebung *VirtualBox* erstellen wir uns nun einen neuen Gast:

- Schalter "Neu"
- Name: Debian 8 LXDE (Linux 64-bit)
- RAM: 1800 MB
- Festplatte erzeugen (VDI, dynamisch alloziiert, Größe 50 GB)
- Schalter "Start"
- Als Boot-Medium die CD *debian-live-8.5.0-amd64-lxde-desktop.iso* einlegen
- Mit Cursor nach unten ⇒ "Install"
- Sprache wählen: "German", 3x Enter
- Host- und Domainnamen vergeben
- Benutzer einrichten (root, tux)
- Zeitzone: "Berlin"
- Partitionierung: Geführt - vollständige Platte ⇒ sda ⇒ Separate /home, /var- und /tmp-Partitionen
- Partitionierung beenden ⇒ Änderungen schreiben? ⇒ JA
- [Was bei der XFCE-Iso-Installation fehlt: Auswahlmü "tasksel"] (Taskselektor) ein altes Tool: "dselect"
- Kopieren des Systems...
- Netzwerkspiegel (Repository) verwenden? JA, welchen Mirror? ⇒ Deutschland / ftp-stud.hs-esslingen.de
- Proxy? <LEER LASSEN>
- Bootloader installieren: GRUB in den MBR der Festplatte "sda" installieren
(Dies ist das Gerät */dev/sda*, was das Hauptgerät darstellt; hier keine Partitionsnummer angeben.)
- Neustart (CD muss nicht manuell entfernt werden)

Siehe auch das mit vielen Screenshots versehene Tutorial unter <https://www.howtforge.com/tutorial/debian-8-jessie-minimal-server/2/>

1.1 Software aktualisieren, Gasterweiterungen installieren

Sinnvollerweise installiert man sich in den virtuellen Gästen passende Erweiterungen und Treiber, die den Austausch von Daten zwischen Host und Guest ermöglichen und die Handhabung erleichtern.

- System als root (im Terminal: `su -`) aktualisieren: `apt-get update && apt-get -y upgrade`
- Hilfspakete installieren und Module bauen:
 - Tools installieren: `apt-get install build-essential module-assistant`
 - Module bauen: `m-a prepare`
- Gasterweiterungen im laufenden Virtualbox-Gastfenster installieren:
 - Evl. eine eingelegte, andere CD auswerfen: Unter *Geräte / Optische Laufwerke / Medium entfernen* (evl. "Auswurf erzwingen")
 - Unter *Geräte* den Eintrag *Medium mit Gasterweiterungen einlegen...* auswählen
 - CD mounten: `mount /dev/sr0 /mnt`
 - Erweiterungen installieren: `bash /mnt/VBoxLinuxAdditions.run`
 - X-Server (gewaltsam) neu starten: `killall Xorg`
- Gast-Maschine evl. exportieren, um sie später auf anderem Rechner laufen lassen zu können:
 - Gast herunterfahren als root mit `shutdown -h now`
 - Im VirtualBox-Hauptfenster den Export-Assistenten ausführen: *DATEI / APPLIANCE EXPORTIEREN...*

Kontrolle und Nutzung

- A. Austausch von markierten Texten mit der Maus (Voraussetzung: unter *Geräte* wählen wir *Gemeinsame Zwischenablage / bidirektional*)
- B. Vollbildmodus: rechte STRG-Taste + F
(Diese Tastenkombination noch einmal drücken, um diesen Modus wieder zu verlassen)

2 Login und kleiner Rundgang

Es steht für den Login ein grafischer Displaymanager zur Verfügung: Bitte als tux einloggen

Im Hauptmenü unter "Systemwerkzeuge":

- Arbeitsmittel: LXTerminal
- Zum Vergleichen gern den Dateimanager "pcmanfm" öffnen

Im Terminal:

```
=> df -h          # disk free, human readable
```

Vergleich mit Windows:

- Nutzerdaten: `c:\Users` ⇒ `/home`
- Installierte Programme: `c:\Program Files` ⇒ `/usr` (Unix System Ressourcen)
- Betriebssystem: `c:\WINDOWS` ⇒ Rest von `/`

Es gibt keine LW-Buchstaben, die sich verschieben können, einen systemweit einheitlichen Namensraum unter der Wurzel "/"!
Zur Kontrolle der Partitionierung:

- `mount # (cat /proc/self/mounts), gern filtern: mount | grep ^/dev`
- `df -h # disk free`
- `lsblk # list block devices, USB-Hostpluggeräte herausfinden`
- `fdisk -l # Nur als root möglich`
- `blkid # Auflisten der Devices unter Angabe der UUIDs`

3 Grundlegende Konzepte

3.1 Benutzerschnittstellen

a) GUIs (Graphical User Interface)

- X-Window-System (Fenstermanager + X-Server, X-Clients)
- Desktops (LXDE, XFCE, GNOME, KDE), Kennzeichen: Integriert sind Zubehörprogramme, einheitliches Look & Feel

b) Cmdline

- Vorteile: Geringer Ressourcenbedarf, bessere Stabilität, weil kein Programmierballast für Kommandozeilenbackends
- Syntax: `KOMMANDO [OPTIONEN] [PATAMETER]`

BSP: `ncal -w 2016`

Wer kümmert sich um die Interpretation der Kommandozeile?

```
tux@deb8:~$ echo $0
/bin/bash
```

ACHTUNG: Leerzeichen sind Trenner der einzelnen Bestandteile, daher müssen sie ←
evl. maskiert werden:

=> `mkdir Eigene\ Dateien`

Wichtige Mechanismen:

- Dateinamensexpansion (Platzhalter, Wildcard, Joker):

```
rmdir -v Eigen*
touch abc.{1,2,3}                    # Nur zum Erzeugen von Testdateien
rm -v abc.?
touch abc.{1,2,3}{x,y,z}            # Nur zum Erzeugen von Testdateien
rm -v abc.[0-9]*                    # EIN beliebiges Zeichen aus der Menge 0 bis 9 ( ←
Negation: "[!1-5]")
```

- Variablenexpansion (Variableninhalte einfügen)


```
echo $PATH          # Suchpfad für Anwendungen
echo $USER
echo $PWD           # Print Working Directory
```

- Kommandozeilensubstitution (Zuerst eine andere Befehlsfolge ausführen, dann der Rest ringsherum); es gibt zwei Varianten, die gleichberechtigt sind:

```
ls -l /lib/modules/`uname -r`
ls -l /lib/modules/${uname -r}
```

4 Topic 101.1 - Determine and configure hardware settings

Im Buch: S. 21 ff

Ziel: Kernelmodule zur Laufzeit des Rechners laden und kontrollieren können

Der Kernel befindet sich unter: /boot/vmlinuz-<VERSION>

Zur des Laufzeit des Rechners werden Module geladen, aus dem Verzeichnis /lib/modules/`uname -r`

- Ausgabe des Kernel-Releases: `uname -r`
- Ausgabe des Hardware Typs (32 oder 64 bit): `uname -m`
- Wie setzt sich die Release-Nummer zusammen?
"MAJOR.MINOR.PATCH", z.B. 3.16.0
(Weitere Nummern und Kürzel je nach Distribution - die Bezeichnung lässt sich im Makefile mittels "EXTRAVERSION = " festlegen)
- Stabile Kernel: gerade-MINOR-Nummer, Entwicklerkernel: ungerade MINOR-Nummer
- Ab Kernel 3.0.0 ist dieses Unterscheidung aufgegeben worden

BEISPIEL: Suche nach einem Novell-kompatiblen Netzwerkkartentreiber:

```
find /lib/modules -name "*ne2k-pci*"
/lib/modules/3.16.0-4-amd64/kernel/drivers/net/ethernet/8390/ne2k-pci.ko
```

Informationen darüber einholen:

```
root@deb8:~# modinfo ne2k-pci
filename:      /lib/modules/3.16.0-4-amd64/kernel/drivers/net/ethernet/8390/ne2k-pci.ko
license:      GPL
description:   PCI NE2000 clone driver
author:       Donald Becker / Paul Gortmaker
alias:        pci:v00008C4Ad00001980sv*sd*bc*sc*i*
alias:        pci:v000012C3d00005598sv*sd*bc*sc*i*
alias:        pci:v000012C3d00000058sv*sd*bc*sc*i*
alias:        pci:v00001050d00005A5Asv*sd*bc*sc*i*
alias:        pci:v000010BDD00000E34sv*sd*bc*sc*i*
alias:        pci:v00001106d00000926sv*sd*bc*sc*i*
alias:        pci:v00004A14d00005000sv*sd*bc*sc*i*
alias:        pci:v00008E2Ed00003000sv*sd*bc*sc*i*
alias:        pci:v000011F6d00001401sv*sd*bc*sc*i*
alias:        pci:v00001050d00000940sv*sd*bc*sc*i*
alias:        pci:v000010ECd00008029sv*sd*bc*sc*i*
depends:       8390
intree:       Y
vermagic:     3.16.0-4-amd64 SMP mod_unload modversions
```

```
parm:      msg_enable:Debug message level (see linux/netdevice.h for bitmap) (uint)
parm:      options:Bit 5: full duplex (array of int)
parm:      full_duplex:full duplex setting(s) (1) (array of int)
```

Welche Optionen können die Ausgabe einschränken? -a, -d, -l, -p, -n

Zum Beispiel nur den Autor eines Modules ausgeben: Option "-a", z.B.:

```
root@deb8:~# modinfo -a ne2k-pci
Donald Becker / Paul Gortmaker
```

Zum Testen:

```
root@deb8:~# lsmod | grep ne2k
root@deb8:~#
root@deb8:~#
root@deb8:~# insmod ne2k-pci
insmod: ERROR: could not load module ne2k-pci: No such file or directory
root@deb8:~#
root@deb8:~# find /lib/modules -name "ne2k*"
/lib/modules/3.16.0-4-amd64/kernel/drivers/net/ethernet/8390/ne2k-pci.ko
root@deb8:~#
root@deb8:~# insmod /lib/modules/3.16.0-4-amd64/kernel/drivers/net/ethernet/8390/ne2k-pci. ←
ko
insmod: ERROR: could not insert module /lib/modules/3.16.0-4-amd64/kernel/drivers/net/ ←
ethernet/8390/ne2k-pci.ko:
Unknown symbol in module
root@deb8:~#
```

FEHLERURSACHE: Bei insmod werden keine Modulabhängigkeiten aufgelöst! ⇒ modprobe

```
root@deb8:~# modprobe ne2k-pci
root@deb8:~#
root@deb8:~# lsmod | grep ne2k
ne2k_pci          12955  0
8390              17143  1 ne2k_pci
```

Module aus dem RAM entfernen:

- a) `rmmod ne2k-pci`
- b) `modprobe -r ne2k-pci` (BITTE FÜR LPI MERKEN!)

Man sieht hier, dass BEIDE entfernt wurden, da das Modul 8390 nur von ne2k-pci gebraucht wurde (kein anderes Modul benötigt es außerdem auch).

```
root@deb8:~# lsmod | grep ne2k
root@deb8:~#
root@deb8:~# lsmod | grep 8390
root@deb8:~#
```

Kontrolle der Modul-Lade-Aktion: `dmesg` (Debug Messages des Kernels als Ringpuffer)

Abhängigkeiten von Modulen prüfen und in einer Datenbank "modules.dep" zu speichern:

```
depmod
-n (non execute + verbose)  => d.h. KEINE modules.dep erstellen
-v (verbose)
-A (Append = Anhängen, d.h. nur wenn neue Module gefunden wurden, eine neue modules. ←
dep erstellen)
```

Nach erfolgreicher Erstellung der Abhängigkeits-Datenbank:

```
root@deb8:~# grep ne2k-pci /lib/modules/3.16.0-4-amd64/modules.dep
kernel/drivers/net/ethernet/8390/ne2k-pci.ko: kernel/drivers/net/ethernet/8390/8390.ko
```

Wo legt der Kernel Informationen beim Bootvorgang ab? Unter /proc (Processing)

Spezielles Verzeichnis: /proc/sys/kernel

```
root@deb8:~# cat /proc/sys/kernel/ostype
Linux
root@deb8:~#
root@deb8:~# cat /proc/sys/kernel/hostname
deb8
root@deb8:~#
```

AUFGABE FÜR DAS SELBSTSTUDIUM: Welche Dateien unterhalb von /proc/sys/kernel werden ausgelesen, wenn man `uname -a` (all) eingibt?

Weitere Pfade für Hardware- und Kernel-nahe Informationen:

- /sys (Viele Infos aus /proc sind hier hinein gewandert, zB: /sys/bus/usb)
- /dev (Allgemeines Verzeichnis für Gerädateien)
 - IDE-Geräte (max. 4): /dev/hda, /dev/hdb, /dev/hdc, /dev/hdd
 - SCSI-Geräte: /dev/sda, /dev/sdb, ...
 - Partitionsnummern: 1 - 4 (max. 4 primäre Partitionen), die erste log. bekommt immer die 5 zugewiesen (/dev/sda5, max 63 Partitionen)
 - SCSI-CD/DVD/RW-Laufwerke: /dev/sr0, /dev/scd0

AUFGABE: Dateien auf Seite 32 mit cat ausgeben, Namen einprägen!

- /proc/ioports (Gibt nur die Port aus, die gerade benutzt werden.)
- ...

Außerdem

```
=> less /proc/bus/pci/devices
```

Aus diesen Files bezieht `lspci` dann seine Informationen und bereitet sie auf.

Wichtige Optionen für das Kommando `lspci`:

```
-v (Verbose, zeigt auch die Bez. des geladenen Modules an)
-k (Kernel-Module ausgeben)
-n (Numerische Ausgabe von Vendor- und Device-ID)
-t (Ausgabe in Tree-Form)
```

Weitere Aufgaben/Hinweise:

- USB-Geräte einbinden/konfigurieren (S. 34)
- Welcher Host-USB-Controller ist eingebaut? → laut geladener Module `lsmod | grep hci`, u.a. taucht `ohci_pci` und `ehci_pci` auf!
- Seit welchen Kernel-Versionen wird USB überhaupt unterstützt und wie ausgeprägt?
- Unter <http://www.tldp.org/HOWTO/Webcam-HOWTO/hardware.html> liest man: "USB subsystem support has been present in the Linux kernel since the late 2.2 series."

- Bessere USB-Unterstützung gab es erst mit dem Kernel 2.4

USB-Support bei verschiedenen Kernen und Hostcontroller-Treibern:

- Kernel 2.0 = **überhaupt kein USB-Support!**
- Kernel 2.2.7 = USB 1.0 = ohci_pci
- Kernel 2.4.0 = USB 2.0 = ehci_pci
- Kernel 2.6.31 = USB 3.0 = xhci_pci

USB-Geräte im System auffinden:

- a) dmesg
- b) lsusb (Geht die Pfade unter /proc/bus/usb bzw. /sys/bus/usb durch und gibt Infos aus.)

=> apt-get install usbutils

Was findet man unter /dev/bus/usb?

- Zeichengeräte für die Autoerkennung des Userspace Device Managers *udev*
- Siehe auch <http://weininger.net/how-to-write-udev-rules-for-usb-devices.html>

AUFGABE: Erkunden Sie die Device-Manager:

- usbmgr
- hotplug
- udev

Zur Klärung der Aufgabe zuerst einmal Debians Paketmanager heranziehen:

```
apt-cache show udev
...
Description-de: /dev/ und hotplug Hintergrundprozess
udev ist ein Hintergrundprozess der Gerätedateien in /dev/ dynamisch
erzeugt und entfernt, hotplug-Ereignisse verarbeitet und Treiber während
des Systemstarts lädt.
...
```

4.1 Fragen zur Vertiefung

- Welche Verzeichnispfade repräsentieren virtuelle Dateisysteme bzw. Ansprechpunkte für Geräte?
 - /proc (Prozessdateisystem, Hardwareinformationen)
 - /sys (Hardware- und Systeminformationen)
 - /dev (Ansprechpunkte für Geräte)
- Mit welchen Optionen kann man bei *lspci* den Namen des geladenen Kernel-Modules ermitteln?
 - Optionen: *-v* sowie *-k*
- Welche Kommandos können Sie einsetzen, um den soeben angesteckten USB-Stick zu identifizieren?

- dmesg | tail
- lsusb

- Worin besteht ein wesentlicher Unterschied beim Laden von Modulen mit insmod bzw. modprobe?
 - insmod kann nur ein einziges Modul laden (keine abhängigen Module vorher laden), außerdem muss der exakte Modulname samt Pfad angegeben werden
 - modprobe ist intelligent: lädt andere Module in Abhängigkeit mit, es findet das betreffende Modul automatisch unter folgendem Pfad:

```
/lib/modules/`uname -r`
```

- Mit welchen zwei Kommandos und evl. benötigten Optionen kann man Module aus dem Hauptspeicher entfernen?
 - rmmmod
 - modprobe -r
- Worin besteht der Hauptunterschied bei *modinfo* und *depmod -n*?
 - modinfo gibt nur Modulinformationen über ein bestimmtes Modul aus
 - depmod -n (noinstall) testet nur Modulabhängigkeiten und gibt darüber Meldung aus

5 Topic 101.2 - Boot the system

⇒ Siehe Seite 38 ff

5.1 Typischer Ablauf eines Linux-Bootvorganges

1. BIOS startet Power-on-self-Test (POST-Routine)
2. Suche nach einem MBR mit Bootcode über mehrere Datenträger hinweg (laut Bootsequenz)
 MBR sichern: `dd if=/dev/sda of=/root/myMBR.dd count=1 bs=512`
 - if - Input File
 - of - Output File
 - count - Counter (Zähler), wie oft mal der Block kopiert werden soll
 - bs - Block size (kleinste Einheit im Intel-PC = Sektor, = 512 Byte)
3. Bootcode wird geladen und damit der Bootloader, bei aktuellen Linux-Systemen ist das der Grand Unified Boot Loader (GRUB)
4. Bootloader findet Kernel auf Boot-Partition und lädt ihn in den RAM
5. Der sich im RAM breit machende Kernel startet den ersten Prozess init bzw. systemd (Prozess-ID 1, lässt sich nicht killen)
 Zur Kontrolle: `ps tree -pn | less` (Optionen: p = Prozess-IDs zeigen; n = numerische Sortierung)
6. Login via Konsole (tty) oder Displaymanager (grafisch)

5.2 Kernel-Parameter

Im 3. Schritt der obigen Startreihenfolge können dem Kernel verschiedene Parameter mitgegeben werden, z.B.

- `init=/bin/bash` (root-Passwort überschreiben)
- `ide=nodma` (für alte Festplatten: Keinen DMA-Kanal aktivieren)
- `apm=off` (= das alte Verfahren "Advanced Power Management")
- `acpi=off` (= das neue Verfahren "Advanced Configuration Power Interface")

Diese Parameter können im gebooteten System später ausgelesen werden:

```
tux@deb8:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.16.0-4-amd64 root=UUID=d426bdb3-2e30-4599-bc17-810bef991d42 ro ←
quiet
tux@deb8:~$
```

5.3 Initiale RAM-Disk

Vielleicht aus DOS-Zeiten bekannt: RAM-Drive

Ziel: Für den Bootvorgang elementare Treiber frühzeitig bereitstellen (Die root-Partition ist nicht nicht gemountet)

Inhalt der RAM-Disk:

- Minimales Dateisystem ähnlich (/)
- Treiber für Festplatten-Controller
- Dateisystemtreiber für Festplattenpartitionen (root-Partition mounten)

Wo liegt sie und wie groß ist sie?

```
tux@deb8:~$ ls -lh /boot/initrd.img-3.16.0-4-amd64
-rw-r--r-- 1 root root 22M Dez 23 09:34 /boot/initrd.img-3.16.0-4-amd64
tux@deb8:~$
```

Den Inhalt der RAM-Disk erforschen:

```
root@deb8:~# file /boot/initrd.img-3.16.0-4-amd64
/boot/initrd.img-3.16.0-4-amd64: gzip compressed data, last modified: Mon May  9 14:41:46 ←
2016, from Unix
root@deb8:~#
root@deb8:~#
root@deb8:~# gunzip -c /boot/initrd.img-3.16.0-4-amd64 > /tmp/initrd
root@deb8:~#
root@deb8:~# file /tmp/initrd
/tmp/initrd: ASCII cpio archive (SVR4 with no CRC)
root@deb8:~#
root@deb8:~# cd /tmp/
root@deb8:/tmp#
root@deb8:/tmp# cpio -idm < initrd
127156 blocks
root@deb8:/tmp#
root@deb8:/tmp# ls -ltrc | tail -8
drwxr-xr-x 2 root root    1024 Mai 10 16:29/sbin
drwxr-xr-x 2 root root    1024 Mai 10 16:29/run
drwxr-xr-x 2 root root    1024 Mai 10 16:29/lib64
drwxr-xr-x 8 root root    1024 Mai 10 16:29/lib
```

```

-rwxr-xr-x 1 root root      7137 Mai 10 16:29 init
drwxr-xr-x 7 root root      1024 Mai 10 16:29 etc
drwxr-xr-x 3 root root      1024 Mai 10 16:29 conf
drwxr-xr-x 2 root root      5120 Mai 10 16:29 bin
root@deb8:/tmp#
root@deb8:/tmp# ls -ltrc lib | tail -8
drwxr-xr-x 3 root root      1024 Mai 10 16:29 udev
drwxr-xr-x 2 root root      1024 Mai 10 16:29 systemd
drwxr-xr-x 3 root root      1024 Mai 10 16:29 modules
drwxr-xr-x 2 root root      1024 Mai 10 16:29 modprobe.d
-rw-r--r-- 1 root root     47712 Mai 10 16:29 libnss_files.so.2
-rw-r--r-- 1 root root     47712 Mai 10 16:29 libnss_files-2.19.so
-rwxr-xr-x 1 root root     71416 Mai 10 16:29 klibc-IpHGKkbZiB_yZ7GPagmQz2GwVAQ.so
drwxr-xr-x 4 root root      1024 Mai 10 16:29 firmware
root@deb8:/tmp#

```

Tools zur erzeugen der Initrds:

- a) RedHat: `whatis mkinitrd mkinitrd (8)` - is a compat wrapper, which calls dracut to generate an initramfs
- b) Debian: `whatis mkinitramfs mkinitramfs (8)` - low-level tool for generating an initramfs image

5.4 Startprotokollierung

⇒ läuft über zwei Ebenen:

- A. Kernel-Ringbuffer: `dmesg` (Bei RedHat zusätzlich in der Datei `/var/log/boot.log`)
Siehe auch <http://www.ibm.com/developerworks/linux/tutorials/l-1pic1111/>
- B. Nach den Starten des Syslog-Dienstes wird die Datei `/var/log/messages` (oft auch `/var/log/syslog`) geführt.

Am besten mit dem tail-Monitor (als root, Option `-f` = follow Data Stream) betrachten:

```
root@deb8:~# tail -f /var/log/messages
```

Nachrichten als einfacher Nutzer ins Syslog schreiben:

```
=> logger "ALARM: Festplatte voll $(df -h /) "
```

5.5 Init-System, Runlevel

a) Startverfahren nach System V

Konfigurationsdatei: `cat /etc/inittab`, hier wird der Standardrunlevel mit `initdefault` festgelegt

⇒ Diese Datei definiert außerdem Runlevelverzeichnisse, in denen Skripte liegen und deren aufrufende Links ebenso

- Ausführbare Skripte liegen im Verzeichnis: `/etc/init.d`
 - Im Beispiel Slitaz: DHCP-Server starten: `/etc/init.d/udhcpd start`
 - Kontrolle: `netstat -una` ⇒ Port 67 ist offen, Konfigdatei `/etc/udhcpd.conf`
- Runlevelverzeichnisse, in denen die aufrufenden, symbolischen Verknüfungen liegen: `/etc/rc0.d` bis `/etc/rc6.d`
 - Unter DSL-Linux (root werden mit `sudo su`)
 - * Kommando `runlevel` zum Ausgeben des vorherigen (N = kein vorheriger Level) und aktuellen Levels
 - * Änderung der `/etc/inittab` einlesen: `telinit q` (telinit ist nur ein symbolischer Link auf `/sbin/init`)
 - * Runlevelverzeichnis z.B. für Level 5: `/etc/rc5.d`

- * Runlevel wechseln (z.B. in den Level 0 = system halt): `init 0`
- Rechner herunterfahren, wobei alle Nutzer darüber informiert werden: die kann allein nur `shutdown -h 16:00` (Das im Hintergrund für Broadcasts verwendete Kommando lautet `wall`; zur Übung einfach `wall` eingeben, Nachricht schreiben und mit STRG + D abschicken)
- Rechner herunterfahren, wobei beim nächsten Bootvorgang ein Dateisystemcheck durchgeführt werden soll: `shutdown -F -h 16:00` (force file system check)
- Rechner herunterfahren, wobei beim nächsten Bootvorgang ein evl. fälliger Dateisystemcheck verhindert werden soll: `shutdown -f -h 16:00` (no file system check)

b) Ubuntu-Startverfahren

- Name des Systems: `upstart`
- Benutzung mit dem Kommando `service` anhand des ssh-Daemons:
 - Infos ausgeben: `service sshd status`
 - Starten: `service sshd stop`
 - Stoppen: `service sshd start`

Siehe auch Cheat Cheets unter:

- https://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet
- https://raymii.org/s/snippets/Systemd_Rosetta_Reference.html
- <http://terry.im/wiki/terry/systemd.html>

c) Neues Startverfahren mit dem Systemd

Das *d* steht für Disks AND Extensions MONitor (daemon) = Dienst bzw. Service

Es gibt drei Sub-Systeme:

1. `systemd-udev` - für die Geräteerkennung
2. `systemd-journald` - für das Schreiben eines eigenen System-Logs
3. `systemd-logind` - für das Aktivieren der User-Umgebungen beim Login

```
ps x | grep systemd
163 ?      Ss      0:00 /lib/systemd/systemd-udev
164 ?      Ss      0:00 /lib/systemd/systemd-journald
483 ?      Ss      0:00 /lib/systemd/systemd-logind
```

Benutzung als root (ACHTUNG: nicht mit `sshd` ausführen, sondern nur mit `ssh`):

- Den ssh-Dienst stoppen: `systemctl stop ssh`
- Den ssh-Dienst starten: `systemctl start ssh`
- Den Status ausgeben: `systemctl status ssh`
- Den ssh-Dienst nicht mehr automatisch beim Booten starten: `systemctl disable ssh`
- Den ssh-Dienst beim Booten mit starten: `systemctl enable ssh`
- Das Journal des Systemd ausgeben: `journalctl` (Komplettes Logfile mit dem Pager `less` lesen)
- Das Journal des Systemd ausgeben, aber nur die letzten Änderungen mit `follow data stream`: `journalctl -f` (wichtig unter SUSE, da hier kein Syslogd vorinstalliert)

- Das "Fehler"-Journal mit Hinweisen ausgeben: `journalctl -xn`
- Runlevel wechseln (hier Level 1 = SingleUser-Modus, ReparaturModus): `systemctl isolate rescue.target`
- Runlevel wechseln (hier Level 5): `systemctl isolate graphical.target`
- Standard-Runlevel anzeigen: `systemctl get-default`
- Standard-Runlevel auf "nicht-grafisch" ändern: `systemctl set-default multi-user.target`
- Standard-Runlevel wieder auf "grafisch" ändern: `systemctl set-default graphical.target`

Siehe auch: <https://fedoraproject.org/wiki/Systemd>

6 Topic 101.3 - Change runlevels / boot targets and shutdown or reboot system

In diesem Kapitel werden die eben kennengelernten Aspekte über Init-Systeme vertieft und ergänzt.

6.1 Systemstart nach System V (SysVinit)

Wichtige Kennzeichen:

- Dienst `/sbin/init` → Siehe auch `pstree -p | head` ⇒ `init(1)`
- Hauptkonfigurationsdatei `/etc/inittab`
 - Standardrunlevel konfigurieren: `id:3:initdefault:` (Hier bitte NIEMALS 0 (poweroff) oder 6 (reboot) eintragen!!)
 - Affengriff konfigurieren: `ca::ctrlaltdel:/sbin/shutdown -r +4`

6.2 Runlevel

- Runlevel sind bestimmte Systemzustände, in denen unterschiedlich viel Dienste gestartet werden, z.B.
- Im Vergleich: Abgesicherter Modus (Windows) ⇒ Runlevel 1 bzw. Single User
- Herausfinden, in welchem wir uns gerade befinden: `runlevel`
- Wechseln: `init [NR]` bzw. `telinit [NR]`, im Bsp.: `telinit 1`, wieder ins GUI fahren: `telinit 5`
- Kernelparameter für GRUB für Single-User-Modus: `1`
 - Zum Beispiel:

```
kernel /boot/vmlinuz-3.16.0-4-amd64 root=/dev/sda3 ro initrd=/install/initrd.gz 1'
```

6.3 Rechner herunterfahren / neu starten

- Nutzer dabei zu informieren, ist nur mit dem Kommando `shutdown` möglich (im Hintergrund verwendet es `wall`)
- Wichtige Optionen: S. 52 unten, mit der Option `-f` kann ein evl. anstehender `fsck` übergangen werden, ein `-F` erzwingt einen Check.

6.4 Systemstart mit dem neuen Systemd

Wichtige Kennzeichen:

- Dienst `/bin/systemd`
- Geräte automatisch erkennen mit `/lib/systemd/systemd-udev`
- Log-System mit `/lib/systemd/systemd-journald` - Der sich selbst aktualisierende Logmonitor: `journalctl -f`
- Kernelparameter für GRUB, um in den Single-User-Modus zu gelangen: `systemd.unit=rescue.target`
 - Zum Beispiel: `kernel /boot/vmlinuz-3.16.0-4-amd64 root=/dev/sda3 ro initrd=/install/initrd.gz systemd.unit=rescue.target`

BEISPIEL: Den Samba-Hauptdaemon unter Debian stoppen, kontrollieren und deaktivieren:

```
root@deb8:~# systemctl stop smbd
root@deb8:~#

root@deb8:~# systemctl status smbd
● smbd.service - LSB: start Samba SMB/CIFS daemon (smbd)
   Loaded: loaded (/etc/init.d/smbd)
   Active: inactive (dead) since Mi 2016-05-11 16:47:45 CEST; 3s ago
   Process: 18286 ExecStop=/etc/init.d/smbd stop (code=exited, status=0/SUCCESS)
   Process: 17048 ExecReload=/etc/init.d/smbd reload (code=exited, status=0/SUCCESS)
   Process: 16861 ExecStart=/etc/init.d/smbd start (code=exited, status=0/SUCCESS)

Mai 11 16:23:20 deb8 smbd[16861]: Starting SMB/CIFS daemon: smbd.
Mai 11 16:23:28 deb8 smbd[16952]: Reloading /etc/samba/smb.conf: smbd.
Mai 11 16:23:28 deb8 smbd[17006]: Reloading /etc/samba/smb.conf: smbd.
Mai 11 16:23:31 deb8 smbd[17048]: Reloading /etc/samba/smb.conf: smbd.
Mai 11 16:47:44 deb8 smbd[17101]: pam_unix(samba:session): session closed for user nobody
Mai 11 16:47:45 deb8 smbd[18286]: Stopping SMB/CIFS daemon: smbd.
root@deb8:~#

root@deb8:~# systemctl disable smbd
Synchronizing state for smbd.service with SysVinit using update-rc.d...
Executing /usr/sbin/update-rc.d smbd defaults
Executing /usr/sbin/update-rc.d smbd disable
insserv: warning: current start runlevel(s) (empty) of script `smbd' overrides LSB defaults ←
(2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `smbd' overrides LSB ←
defaults (0 1 6).
root@deb8:~#
```

7 Topic 102.1 - Design hard disk layout

⇒ Seite 57

ZIEL: Linux-Server stabil betreiben, ohne dass die Festplatten voll laufen können.

Wir haben bei der Installation bereits Vorsorge getroffen, die veränderlichen Daten liegen auf separaten Partitionen:

```
df -h /home
df -h /var
df -h /tmp
```

FRAGE 1: Welche Problem kann es mit `/boot` geben?

- Durch veraltete BIOS-Versionen konnte der Bootloader u.U. nicht über die Grenze von 1024-Zylindern zugreifen (ca. 8GB), wenn dabei

- LILO als Bootmanager zum Einsatz kam.
- Deshalb musste dann eine separate, kleine /boot-Partition am Anfang der Festplatte angelegt werden.

FRAGE 2: Welche Verzeichnisse müssen auf der Wurzel-Partition verbleiben (Systemrelevant)?

- /dev, /proc/, /sys, /etc, /bin, /sbin, /lib, ... (Siehe auch Screenshot)

Was sollte der Admin auslagern (Nutzer können dort schreiben!):

- /tmp, /var, /home

7.1 Partitionierung

Typischerweise sollte sich die Partitionierung der Festplatte so gestalten:

1. Part.: Größe 300 MB ⇒ Primär ⇒ Anfang ⇒ Dateisystem: ext4 ⇒ Einbindungspunkt: /boot [/dev/sda1]
2. Part.: Größe 30 GB ⇒ Logisch ⇒ Anfang ⇒ Dateisystem: btrfs ⇒ Einbindungspunkt: / ⇒ (Name: root) [/dev/sda5]
3. Part.: Größe 1 GB ⇒ Logisch ⇒ Anfang ⇒ Dateisystem: btrfs, ⇒ Einbindungspunkt: /tmp [/dev/sda6]
4. Part.: Größe 3 GB ⇒ Logisch ⇒ Anfang ⇒ Dateisystem: btrfs, ⇒ Einbindungspunkt: /var [/dev/sda7]
5. Part.: Größe 5 GB ⇒ Logisch ⇒ Anfang ⇒ Dateisystem: xfs, ⇒ Einbindungspunkt: /home [/dev/sda8]
6. Part.: Größe 2 GB (Doppelte RAM-Größe) ⇒ Benutzen als: Swap (kein Einbindungspunkt) [/dev/sda9]

Hier sind die Verzeichnisse, in denen Nutzer Daten ablegen können, konsequent auf getrennte Partitionen verteilt worden. Das bringt Stabilität - auch ohne den Einsatz von Disk-Quotas.

7.2 Logical Volume Manager

Was ist das?

Aus einer LPI-Prüfung:

What can the Logical Volume Manager (LVM) be used for? (Choose THREE correct answers.)

- A. To create RAID 9 arrays.
- B. To dynamically change the size of logical volumes.
- C. To encrypt logical volumes.
- D. To create snapshots.
- E. To dynamically create or delete logical volumes.

Richtige Antworten: B, D, E

Im Experiment:

Eine zweite Festplatte einbauen (VBox: Massenspeicher / Controller SATA markieren, + anklicken...)

```

root@deb8:~# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0   50G  0 disk
├─sda1     8:1    0   8,4G  0 part /
├─sda2     8:2    0    1K  0 part
├─sda5     8:5    0   2,8G  0 part /var
├─sda6     8:6    0   3,4G  0 part [SWAP]
├─sda7     8:7    0  380M  0 part /tmp
└─sda8     8:8    0   35G  0 part /home
sdb        8:16   0    8G  0 disk
sr0       11:0   1  55,5M  0 rom  /media/cdrom0
root@deb8:~#

```

Installation und Einrichtung:

```

apt-get install lvm2

# Physische Kennung für LVM erzeugen
root@deb8:~# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
root@deb8:~#

# Volumengroup erzeugen
root@deb8:~# vgcreate backup /dev/sdb
Volume group "backup" successfully created
root@deb8:~#

# Volumes erzeugen
root@deb8:~# lvcreate -n doc -L 3G backup
Logical volume "doc" created
root@deb8:~#
root@deb8:~# lvcreate -n app -L 2G backup
Logical volume "app" created
root@deb8:~#

# Kontrolle
root@deb8:~# pvs
PV          VG      Fmt  Attr PSize PFree
 /dev/sdb   backup lvm2 a--  8,00g 3,00g
root@deb8:~#
root@deb8:~# vgs
VG      #PV #LV #SN Attr   VSize VFree
 backup  1  2  0 wz--n- 8,00g 3,00g
root@deb8:~#
root@deb8:~# lvs
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
 app    backup -wi-a----- 2,00g
 doc    backup -wi-a----- 3,00g
root@deb8:~#

# Dateisystem aufbringen
root@deb8:~# mke2fs /dev/backup/doc
mke2fs 1.42.12 (29-Aug-2014)
Ein Dateisystems mit 786432 (4k) Blöcken und 196608 Inodes wird erzeugt.
UUID des Dateisystems: 7b7adc78-9537-4197-88b4-efbcd8dacc33
Superblock-Sicherungskopien gespeichert in den Blöcken:
 32768, 98304, 163840, 229376, 294912

```

```
beim Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt
```

```
root@deb8:~#
```

```
# Mounten und kontrollieren
root@deb8:~# mount /dev/backup/doc /mnt
root@deb8:~#
root@deb8:~# ls -l /mnt/
insgesamt 16
drwx----- 2 root root 16384 Mai 12 16:47 lost+found
root@deb8:~#
root@deb8:~# df -h /mnt/
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/backup-doc 3,0G  4,5M  2,8G   1% /mnt
root@deb8:~#
root@deb8:~# mount | tail -1
/dev/mapper/backup-doc on /mnt type ext2 (rw,relatime)
root@deb8:~#
```

Bitte nicht gesamten Platz für Volumes aufbrauchen, da sonst keine LVM-Snapshots angelegt werden können.

Man nutzt u.a. gern Snapshots, um das Wurzeldateisystem überhaupt mit *fsck* prüfen zu können.

8 Topic 102.2 - Install a boot manager

⇒ Seite 60 ff

Bootloader für Linux

- a) LILO (veraltet, 1024-Zylinder-Problem)
- b) GRUB (neu, zwei Versionen)

8.1 Unterschiede zwischen GRUB 1 und GRUB 2

8.1.1 Der veraltete (legacy) GRUB 1

- Konfigurationsdatei: `/boot/grub/menu.lst`
 - Schlüsselwort für die Kernel-Zeile (Übergabe von Bootparametern): *kernel*
z.B.: `kernel /boot/vmlinuz-3.2.4 root=/dev/sda1`
- Interaktive GRUB-Shell vorhanden (S. 62): Kommando: *grub*
- Bezeicher für /boot-Partition z.B. für 1. Festplatte, 1. Partition: `hd0,0`
 - Beispiel: `root (hd0,0)`
 - ACHTUNG: HIER BEGANN MAN BEI DEN PARTITIONEN UNGEWÖHLICHERWEISE MIT NULL ZU ZÄHLEN AN!

8.1.2 Der neue GRUB 2

- Konfigurationsdatei: `/boot/grub/grub.cfg`
 - Schlüsselwort für die Kernel-Zeile (Übergabe von Bootparametern): `linux`
z.B.: `linux /boot/vmlinuz-3.2.4 root=/dev/sda1`
- Interaktive GRUB-Shell NICHT mehr vorhanden
- Bezeichner für /boot-Partition z.B. für 1. Festplatte, 1. Partition: `hd0,1`
 - Beispiel: `root (hd0,1)`

8.2 Gemeinsamkeiten der beiden GRUB-Versionen:

- Zweistufig
 - Stage 1: /boot mounten, Kernel + Initrd in den RAM bringen
 - Stage 2: `sysVinit` bzw. `Systemd` starten, den Rest mounten (siehe `/etc/fstab`)
- Installation von Stage 1 (Bootcode im MBR, 444 Byte von insgesamt 512 Byte)
 - BEISPIEL: `mount /dev/sda1 /mnt grub-install --root-directory=/mnt /dev/sda`
FRAGE DAZU: Wo befindet sich in dem Falle, wenn wir die Wurzelpartition unter /mnt gemountet haben, das Verzeichnis namens `grub`? ANTWORT: Unter /mnt/boot (Gesamter Verzeichnispfad: /mnt/boot/grub)

Siehe auch die LPI-Frage: <http://pemann.de/doc/lpi101/2016-05-17/20160517-153823-014.png>

- Eine weitere Gemeinsamkeit ist die Art, wie eine neue Konfigurationsdatei erzeugt wird:
 - Allgemein mit dem Kommando `grub-mkconfig`

An dieser Stelle leiten wir nur mal in eine temporäre Datei um:

```
root@deb8:~# grub-mkconfig > /tmp/grub.cfg
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Linux-Abbild gefunden: /boot/vmlinuz-3.16.0-4-amd64
initrd-Abbild gefunden: /boot/initrd.img-3.16.0-4-amd64
Found memtest86+ image: /boot/memtest86+.bin
Found memtest86+ multiboot image: /boot/memtest86+_multiboot.bin
erledigt
root@deb8:~#
```

Viele Distributionen haben allerdings ihre eigenen Tools, bei Debian ist das `update-grub`:

```
root@deb8:~# update-grub
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Linux-Abbild gefunden: /boot/vmlinuz-3.16.0-4-amd64
initrd-Abbild gefunden: /boot/initrd.img-3.16.0-4-amd64
Found memtest86+ image: /boot/memtest86+.bin
Found memtest86+ multiboot image: /boot/memtest86+_multiboot.bin
erledigt
root@deb8:~#
root@deb8:~# find /boot/ -cmin 1
/boot/grub
/boot/grub/grub.cfg
root@deb8:~#
```

Weitere Tipps und Dokumentation zu diesem Thema unter <http://pemann.de/doc/lpi1/101/2016-01-08/whlg-ivm-grub.html>

9 Topic 102.3 - Manage shared libraries

Siehe Seite 64 ff

Zum Einstieg:

- Parallele zu Windows? ⇒ dll-Dateien
- *whatis ld* - The GNU linker (= das Hauptwerkzeug)
- Was ist der Vorteil? ⇒ Schlankere Anwendungen (sie greifen auf gemeinsame Funktionsbibliotheken zurück)
- Ist das sicherer als bei statisch gelinkten Programmen? NEIN! Ein Fehler in der Library und alle Apps haben ihn sofort auch.
- Gibt es Programme, die keine shared Libs benötigen? Ja, früher z.B. Opera-Browser, Stichwort: portable Apps)
 - <http://www.srware.net/downloads/iron-linux-64.tar.gz>

9.1 Abhängigkeiten ausgeben mit ldd

Gegeben sei ein Programm: *grep* (Searching with regulary ExPressions)

- Zuerst ermitteln wir, wo es im Dateisystem liegt (über den Suchpfad der Shell hinweg auch gern mittels *which*)
- Danach lassen wir uns mit *ldd* die Abhängigkeiten ausgeben

```
tux@deb8-2:~$ type grep
grep ist /bin/grep
tux@deb8-2:~$
tux@deb8-2:~$ ldd /bin/grep
        linux-vdso.so.1 (0x00007ffe8cf8a000)
        libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007f2b4836e000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f2b4816a000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f2b47dc1000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f2b47ba4000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f2b4880f000)
tux@deb8-2:~$
```

Am Beispiel unserer vermeintlich portablen App *iron*:

```
tux@deb8:~$ tar zxf iron-linux-64.tar.gz
tux@deb8:~$
tux@deb8:~$ cd iron-linux-64/
tux@deb8:~/iron-linux-64$
tux@deb8:~/iron-linux-64$ file iron
iron: ELF 64-bit LSB shared object, x86_64, version 1 (SYSV), dynamically linked, ↔
interpreter
        /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.26, BuildID[sha1]=caa29fe2e, not stripped
tux@deb8:~/iron-linux-64$
tux@deb8:~/iron-linux-64$ du -sh .
219M      .
tux@deb8:~/iron-linux-64$ ldd iron
        linux-vdso.so.1 (0x00007ffc19fda000)
        librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007f39580fc000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f3957ef8000)
        ...
```

Wir sehen hier deutlich, dass nach Shared Libraries auf dem vorliegenden Linux-System gesucht wird.

Testweise das Programm natürlich auch mal starten: `./iron &`

Eine wichtige Fragestellung, die mit `ldd` beantwortet werden kann ist außerdem: "Kann der SSH-Server mittels dem `tcpd` (`/etc/hosts.allow` bzw. `/etc/hosts.deny`) abgesichert werden?" Ja, wenn er gegen die `libwrap` gelinkt ist. Und das ist hier der Fall:

```
root@deb8:~# ldd `which sshd` | grep --color wrap
      libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007f695be0e000)
root@deb8:~#
```

HINWEIS für die LPI-Prüfung: Es lassen sich auch Bibliotheken abfragen:

```
root@deb8:~# ldd /lib/x86_64-linux-gnu/libresolv-2.19.so
      linux-vdso.so.1 (0x00007ffce804e000)
      libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007feca408e000)
      /lib64/ld-linux-x86-64.so.2 (0x00007feca4650000)
root@deb8:~#
```

9.2 Den Cache erzeugen mit `ldconfig`

- *whatis ldconfig* - configure dynamic linker run-time bindings
 - Erzeugt nach Änderungen in den Configs (i.d.R. `/etc/ld.so.conf`) einen neuen Cache
- Die Datenbank-Cachedatei: `/etc/ld.so.cache`

Im Experiment:

```
root@deb8:~# ldconfig
root@deb8:~#
root@deb8:~# find /etc/ -cmin 1
/etc/
/etc/ld.so.cache
root@deb8:~#

root@deb8:~# file /etc/ld.so.cache
/etc/ld.so.cache: data
root@deb8:~#

root@deb8:~# strings /etc/ld.so.cache | head -7
ld.so-1.7.0
glibc-ld.so.cache1.1
libzipwrap-0.so.13
/usr/lib/x86_64-linux-gnu/libzipwrap-0.so.13
libzipmapped-0.so.13
/usr/lib/x86_64-linux-gnu/libzipmapped-0.so.13
libzipseeko-0.so.13
root@deb8:~#
```

9.3 Konfiguration der Suchpfade für Libraries

Dank des mit `ldconfig` zu erzeugenden Caches ist ein schnelles Auffinden der einzelnen Bibliotheken kein Problem. Es gibt keine Textdateien, in denen alle Libs vermerkt sind, sondern eine zu bildende Datenbankdatei, die `/etc/ld.so.cache` heißt.

Konfiguration: In der Datei `/etc/ld.so.conf` können die einzelnen Pfade direkt hinterlegt werden, in denen dann `ldconfig` nach den `.so`-Dateien sucht (Shared Objects) und den Cache daraus baut.

9.4 Neuerstellen des Caches und Prüfen des Inhaltes

Nach Installation oder Einbindung neuer Bibliotheken muss wie bereits erwähnt `ldconfig` ausgeführt werden. (zur Kontrolle Zeitstempel vergleichen: `ls -ld /etc/ld.so.*`)

Wichtig ist es für die LPI-Prüfung außerdem zu wissen, wie man den Inhalt des Caches ausgeben (= printing) kann: `ldconfig -p`

9.5 Einfaches Bereitstellen von Libraries via Variable

- Ein einfacher Nutzer kann sich seinen Lib-Pfad selbst konfigurieren
- Libs liegen angenommen unter `~/mylibs`
- Variable erstellen und zugleich exportieren: `export LD_LIBRARY_PATH="~/mylibs"`

Fragen - ???

- Kann man Libs anderer Distributionen auf dieses System übertragen und damit einbinden? Ja, aber nur bedingt (gleicher Release-Stand erforderlich)
- Wird das in jedem Falle funktionieren? NEIN!! Siehe <http://linuxundich.de/gnu-linux/linus-torvalds-schimpft-bauen-von-anwendungen-fur-linux/>
- TIPP: Man nutze am besten Distributionen mit breitem Softwareangebot wie Debian (43.000 Packages!): <https://www.debian.org/distrib/packages>

10 Topic 102.4 - Use Debian package management

Seite 66 ff

10.1 Das Debian Backend dpkg

Der "Debian Packager" ist das eigentliche Werkzeug, welches das Entpacken und Ausrollen von Debian Paketen (Datei-Suffix: .deb) durchführt.

Welches Archivformat liegt hierbei vor? **ar** (archiver). Das Paket `dos2unix` im Experiment lediglich herunterladen und entpacken:

```
root@deb8:~# apt-get -d install dos2unix
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden Pakete wurden automatisch installiert und werden nicht mehr benötigt:
  libgssapi3-heimdal libheimntlm0-heimdal
Verwenden Sie »apt-get autoremove«, um sie zu entfernen.
Die folgenden NEUEN Pakete werden installiert:
  dos2unix
0 aktualisiert, 1 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 77,8 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 314 kB Plattenplatz zusätzlich benutzt.
Holen: 1 http://ftp.tu-chemnitz.de/pub/linux/debian/debian/ jessie/main dos2unix amd64 ↵
  6.0.4-1 [77,8 kB]
Es wurden 77,8 kB in 0 s geholt (733 kB/s).
Herunterladen abgeschlossen; Nur-Herunterladen-Modus aktiv
root@deb8:~#
root@deb8:~#
root@deb8:~# mkdir dos2unix
root@deb8:~#
```

```

root@deb8:~# cd dos2unix/
root@deb8:~/dos2unix#
root@deb8:~/dos2unix# ar x /var/cache/apt/archives/dos2unix_6.0.4-1_amd64.deb
root@deb8:~/dos2unix#
root@deb8:~/dos2unix# ls -l
insgesamt 84
-rw-r--r-- 1 root root 1391 Mai 19 16:17 control.tar.gz
-rw-r--r-- 1 root root 76184 Mai 19 16:17 data.tar.xz
-rw-r--r-- 1 root root 4 Mai 19 16:17 debian-binary
root@deb8:~/dos2unix#

```

Der Packager *dpkg* ist ein sehr intelligentes Werkzeug, was alles kann, außer:

- Downloads via Netzwerk durchführen
- Paketabhängigkeiten selbständig auflösen (In begrenztem Maße dank Wildcard *: `dpkg -i *.deb`)

Beispiele:

- Installation eines Paketes: `dpkg -i /var/cache/apt/archives/dos2unix_6.0.4-1_amd64.deb` (Anwendung, z.B. `unix2dos mytextfile.txt`)
- Auflisten aller installierten Pakete: `dpkg -l` oder ausführlich `dpkg --get-selections` (listing, Filtern: `dpkg -l | grep smb`)
- Auflisten aller Dateien, zum installierten Paket gehören: `dpkg -L bash` oder ausführlich `dpkg --get-files bash` (Nach config-files suchen: `dpkg -L bash | grep rc` ⇒ bei der `bash` mit "rc" = run command oder resource control)
- Nach dem Paketnamen suchen, aus dem eine vorgefundene Datei stammt: `-S` bzw. `--search`

```

root@deb8:~# dpkg -S /usr/bin/unix2mac
dos2unix: /usr/bin/unix2mac
root@deb8:~#

```

- Ausgeben von Informationen: `dpkg -s bash` oder alternativ: `dpkg --get-architecture bash` (Für die Prüfung: nicht `--info` merken!!)
- Ausgeben, welche Pakete nur teilweise installiert sind (**and suggest how to get them correctly installed**): `dpkg -C` oder alternativ: `dpkg --audit`
- Selections vornehmen: `echo "fglrx-driver hold" | dpkg --set-selections` Siehe auch <http://debiananwenderhandbuch.de/dpkg.html>
- Entfernen eines einzelnen Paketes (Config-files verbleiben im System): `dpkg -r fortune-mod`
- Komplettes entfernen eines einzelnen Paketes (Config-files werden ebenfalls gelöscht): `dpkg -P fortune` oder mit langer Option `--purge`
- Paket neu konfigurieren (Starten eines Assistenten): `dpkg-reconfigure keyboard-configuration`
 - Installation von *fortune* unter Debian
- Vorbereitung mit Option "download only": `apt-get -d install fortune`, dabei wird automatisch das Paket *fortune-mod* ausgewählt und mitsamt Abhängigkeiten nach `/var/cache/apt/archives` heruntergeladen
- Wegen den Abhängigkeiten erfolgt die Installation dann so:
 - Ins Downloadverzeichnis wechseln: `cd /var/cache/apt/archives`
 - Nachsehen, was soeben heruntergeladen wurde (zeitlich sortiert): `ls -ltrc`
 - Die betreffenden Pakete alle zusammen angeben (Dateijoker: *): `dpkg -i fortune*.deb librecode0_3.6-21_amd64.deb`

10.2 Frontends für Debian

Frontends sorgen dafür, dass die dpkg-Paketdatenbank komfortabel durchsucht und neue Pakete mitsamt Abhängigkeiten via Netzwerk leicht nachinstalliert werden können.

Wichtige Frontends sind:

- dselect (veraltet)
- apt-Tools (apt-get, apt-cache, apt-key, ..., apt)
- aptitude (neu, als pseudografisches GUI und als Cmdline-Tool verwendbar)
 - Verwendet eine andere Datenbank als "apt-get"!
 - Das Feature, in Abhängigkeit mit installierte Pakete automatisch zu entfernen, beherrscht heute *apt-get* (*apt-get autoremove*, s.u.)
- dselect
- synaptic (GUI-Werkzeug für X-Window)

10.2.1 Konfiguration der apt-Repositories

Im Prinzip kann man alle Quellen in einer Datei hinterlegen, der `/etc/apt/sources.list`. Das unten notierte Beispiel beinhaltet die klassischen drei Zeilen für die aktuelle, stabile Distribution *jessie*. Die Komponenten "contrib" und "non-free" (Zeile 1) fehlen bei einer standardmäßigen Installation, können aber jederzeit nachgetragen werden. So werden z.B. oft die Pakete

- *flashplugin-nonfree* (Flash-Player für Firefox)
- *firmware-iwlwifi* (Binary firmware for Intel Wireless cards)

benötigt. Wir lassen uns mit der folgenden Zeile den aktiven Inhalt der Datei ausgeben:

```
=> grep -v ^# /etc/apt/sources.list
```

```
deb http://ftp.tu-chemnitz.de/pub/linux/debian/debian/ jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main
deb http://ftp.tu-chemnitz.de/pub/linux/debian/debian/ jessie-updates main
```

Nach Änderungen in dieser Datei bitte immer zuerst die aktuellen Paketlisten herunterladen:

```
=> apt-get update # Im Vergleich: 'zypper refresh' (SuSE), 'yum check-
update' (RedHat)
```

Weitere Hinweise zur Konfiguration der Quellen siehe auch:

- <https://wiki.debian.org/SourcesList>
- <https://wiki.debian.org/de/DebianEinsteiger>
- <https://www.elektronik-kompodium.de/sites/raspberry-pi/2002201.htm>

Eine darüber hinaus gern genutzte Konfigurationsdatei ist die `/etc/apt/apt.conf`, wenn es darum geht, einen evl. erforderlichen Proxy-Server anzugeben:

```
Acquire::http::Proxy "http://10.0.0.1:3128";
```

Nun geht es los - wir suchen nach Paketen und installieren verschiedene:

```

=> apt-cache search geany      # Suchen nach der Zeichenkette 'geany', = IDE / Editor
=> apt-cache show geany       # Ausführliche Infos ausgeben
=> apt-cache policy nmap      # Versionen ausgeben bei verschiedenen Quellen, Siehe ↔
    auch 'apt-Pinning'
=> apt-get install geany      # In Abhängigkeit wird 'geany-common' mitinstalliert
=> apt-get remove geany      # Entfernt nur das angegebene Paket, wobei die Configs ↔
    auf dem System          # verbleiben siehe 'dpkg -l geany'
=> apt-get autoremove        # Deinstallieren von überflüssigen, in Abhängigkeit ↔
    mit installierten Paketen
=> apt-get purge geany       # Entfernt das Paket komplett

```

10.2.2 Beispiel Installation u. Rekonfiguration des LDAP-Servers

```

=> apt-get install slapd
=> dpkg-reconfigure slapd

```

Weiterführende Literatur: <http://debiananwenderhandbuch.de/apt-get.html>

11 Topic 102.5 - Use RPM and YUM package management

11.1 Bereitstellen einer VM für Praxisübungen

1. Download durchführen

<http://freefr.dl.sourceforge.net/project/virtualboximage/CentOS/6.3/CentOS-6.3-x86.7z>

2. Entpacken...

3. VM in Virtualbox erstellen

```

=> Doppelklick auf die Datei "CentOS 6.3 x86.vbox"
=> VM starten...

```

4. Grafischer Login als Nutzer *centos*

- Passwort: *reverse*
- Tastaturlayout auf Deutsch stellen: *setxkbmap de*
(Bei Problemen mit dem Pipe-Zeichen manchmal ein Ausweg: *setxkbmap us*)

11.2 RedHat Package Manager

Voraussetzung:

```

=> yumdownloader dos2unix

```

11.3 Archiv-Format

Zuerst einmal wollen wir uns das interne Archivierungsformat anschauen. Es handelt sich um ein cpio-Archiv, an das man mittels *rpm2cpio* herankommt:

```

=> rpm2cpio dos2unix-*.rpm > dos2unix.archiv
=> file dos2unix.archiv
=> rpm2cpio dos2unix-*.rpm | cpio -idm

```

11.4 Nicht installierte Pakete erforschen

Wichtige Option in diesem Zusammenhang: `-p` oder `--package`

Erforschen, Testen und Fragen stellen kann man nur im Query-Modus: `-q` oder `--query`

Die Konfigfiles ausgeben: `-c` oder `--configfiles`

11.5 Shell-Mitschnitt

```
[root@localhost ~]# yumdownloader dos2unix
Loaded plugins: fastestmirror, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: ftp.halifax.rwth-aachen.de
 * extras: ftp.halifax.rwth-aachen.de
 * updates: centos.bio.lmu.de
dos2unix-3.1-37.el6.i686.rpm                | 15 kB      00:00
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# file dos2unix-3.1-37.el6.i686.rpm
dos2unix-3.1-37.el6.i686.rpm: RPM v3.0 bin i386/x86_64 dos2unix-3.1-37.el6
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm2cpio dos2unix-3.1-37.el6.i686.rpm > dos2unix.archiv
[root@localhost ~]#
[root@localhost ~]# file dos2unix.archiv
dos2unix.archiv: ASCII cpio archive (SVR4 with no CRC)
[root@localhost ~]#

dos2unix.archiv: ASCII cpio archive (SVR4 with no CRC)
[root@localhost ~]#
[root@localhost ~]# rpm2cpio dos2unix-3.1-37.el6.i686.rpm | cpio -idm
37 blocks
[root@localhost ~]#
[root@localhost ~]# ls -ltr
total 60
-rw-r--r--. 1 root root 15824 Jul  3  2011 dos2unix-3.1-37.el6.i686.rpm
-rw-r--r--. 1 root root 12148 Jul  7  2012 post-install
-rw-r--r--. 1 root root   550 Jul  7  2012 post-install.log
-rw-----. 1 root root   956 Jul 10  2012 anaconda-ks.cfg
-rw-r--r--. 1 root root 18556 May 24 16:31 dos2unix.archiv
drwxr-xr-x. 4 root root  4096 May 24 16:33 usr
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm --package --query dos2unix-3.1-37.el6.i686.rpm
dos2unix-3.1-37.el6.i686
[root@localhost ~]#
[root@localhost ~]# rpm --query dos2unix-3.1-37.el6.i686.rpm
package dos2unix-3.1-37.el6.i686.rpm is not installed
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm --package --query --changelog dos2unix-3.1-37.el6.i686.rpm | head ←
-15
* Wed Mar 03 2010 Tim Waugh <twaugh@redhat.com> 3.1-37
- Added comments for all patches.

* Mon Nov 30 2009 Dennis Gregorovic <dgregor@redhat.com> - 3.1-36.1
- Rebuilt for RHEL 6

* Fri Jul 24 2009 Fedora Release Engineering <rel-eng@lists.fedoraproject.org> - 3.1-36
```

```

- Rebuilt for https://fedoraproject.org/wiki/Fedora_12_Mass_Rebuild

* Tue Feb 24 2009 Fedora Release Engineering <rel-eng@lists.fedoraproject.org> - 3.1-35
- Rebuilt for https://fedoraproject.org/wiki/Fedora_11_Mass_Rebuild

* Wed Sep 24 2008 Tim Waugh <tw Vaughn@redhat.com> 3.1-34
- Moved 'make clean' to prep section and added comment about there
  being no upstream (bug #225706).
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm -q -p -c dos2unix-3.1-37.el6.i686.rpm      # Das Paket besitzt kein ←
  Config-File
[root@localhost ~]# rpm -q -p -l dos2unix-3.1-37.el6.i686.rpm
/usr/bin/dos2unix
/usr/bin/mac2unix
/usr/share/doc/dos2unix-3.1
/usr/share/doc/dos2unix-3.1/COPYRIGHT
/usr/share/man/man1/dos2unix.1.gz
/usr/share/man/man1/mac2unix.1.gz
[root@localhost ~]#
[root@localhost ~]# rpm -qpl dos2unix-3.1-37.el6.i686.rpm
/usr/bin/dos2unix
/usr/bin/mac2unix
/usr/share/doc/dos2unix-3.1
/usr/share/doc/dos2unix-3.1/COPYRIGHT
/usr/share/man/man1/dos2unix.1.gz
/usr/share/man/man1/mac2unix.1.gz
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# man rpm | grep '\-c'
    rpm {-K|--checksig} [--nosignature] [--nodigest]
      [--changelog] [-c,--configfiles] [-d,--docfiles] [--dump]
--changelog
-c, --configfiles
rpm [--checksig] [--nosignature] [--nodigest]
The --checksig option checks all the digests and signatures contained
--checksig is useful to verify all of the digests and signatures asso-
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm -qp --configfiles dos2unix-3.1-37.el6.i686.rpm
[root@localhost ~]#

```

11.6 Kommandozeilenfrontend yum

Am besten, wir vergleichen es mit Debians Frontend *apt*:

Debian	RedHat
/etc/apt/sources.list	Dateien unter /etc/yum.repos.d
apt-get update	yum check-update
apt-get upgrade	yum update
apt-get dist-upgrade	yum upgrade (entspricht <i>yum --obsoletes update</i>)
apt-cache search BEGRIFF	yum search BEGRIFF
apt-get install PACKAGE	yum search PACKAGE
apt-get -d install PACKAGE	yumdownloader PACKAGE
apt-cache show PACKAGE	yum info PACKAGE

11.7 yum-Spezialität - Paketgruppen

Installation von Paketgruppen (Serverkomponenten, Desktop)

```
=> yum grouplist
=> yum groupinstall "Web Server"
=> yum groupremove "Web Server"
```

11.8 VBox Gasterweiterungen installieren

Siehe <https://wiki.centos.org/HowTos/Virtualization/VirtualBox/CentOSguest>

a) Software installieren

- `yum install dkms`
- `yum groupinstall "Development Tools"`
- `yum install kernel-devel`

b) Gast-Erweiterungen installieren

- Im VBox-Gastfenster unter Geräte ⇒ Gasterweiterungen einlegen
- Mounten der CD: `mount /dev/sr0 /mnt`
- Kompilieren und Installieren der Treiber: `/mnt/VBoxLinuxAddition.run`
- Logout/Login oder Neustart...

12 Topic 103.1 - Work on the command line

Siehe Seite 89 ff

Wir unterscheiden zwei Arten von Variablen:

- Umgebungsvariablen: großgeschrieben, werden automatisch vererbt
 - BSP: `echo $PATH`
 - BSP: `echo $PS2`
- Shellvariablen werden manuell deklariert, bei Bedarf können sie exportiert werden
 - BSP: `myname="Axel Pemmman" ; echo $myname`
 - Kindterminal starten: `gnome-terminal &`
 - Wo ist diese Variable denn aber in diesem neuen Terminal? Nicht vorhanden! Sie wurde nicht exportiert.
 - Daher schließen wir das Terminal und starten es erst nach dem Exportieren der Variablen:

```
exit
export myname
gnome-terminal &
```

Zur Kontrolle der Variablen:

- Die umfangreichsten Ausgaben liefert *set* (alle Shell- und Umgebungsvariablen):

```
=> set | grep myname
```

- Nur die Umgebungsvariablen liefert *env*:

```
=> env | grep PATH
```

Wenn nun die selbst erzeugte Shellvariable *myname* auftaucht, dann wurde sie exportiert, was man aber wieder mit

```
=> export -n myname
```

zurücknehmen kann.

Ganz und gar entfernt werden kann diese eigene Variable am Schluss mit:

```
=> unset myname
```

Nicht nur für LPI wichtig: Skripte, die in der aktuellen Shellumgebung eingelesen und aktiv gesetzt werden sollen, müssen mit

```
=> source myscript
```

bzw. alternativ mit einem Punkt, worauf ein Leerzeichen folgt,

```
=> . myscript
```

ausgeführt werden...

12.1 Konfigurationsdateien der Shell

Motivation: Viele Einstellungen lassen sich temporär vornehmen, um sie festzuschreiben, müssen die verschiedensten Konfigurationsdateien editiert werden.

12.1.1 Unterscheidung zwischen alten Shells und der neuen Standardshell: bash

Alte Shells

Beispiele (siehe auch */etc/shells*): */bin/sh*, */bin/ksh*

Seite 90/ 91: Wichtiges Unterscheidungskriterium: Welche Dateinamen beinhalten NICHT den Namen *bash*?

```
=> /etc/profile
```

```
=> ~/.profile
```

12.1.2 Startup-Dateien für User-Profiles

a) Systemweit

- */etc/profile* - Wird als erstes nach dem User-Login abgearbeitet, allgemein für alle Shells, z.B. Variablen *PATH* und *PS1*, evl auch *umask*
- */etc/bashrc* - Wird als nächstes als Konfigurationsdatei speziell für die *bash* abgearbeitet, z.B. *PS1* (für Promptbildung der *bash*)

b) Benutzerdefiniert (Diese Dateien überschreiben die systemweiten Einstellungen)

- *~/.profile* - Wird einmalig nach dem Login nach dem Ausführen der */etc/profile* abgearbeitet, Verwendung wie eben diese Datei, in der Vergangenheit war sie auch für die *bash* zuständig.

- `~/.bash_profile` - Spezielle bash-Startupdatei, die die alte `~/.profile` ersetzt, wird nur einmal abgearbeitet
- `~/.bash_login` - Entspricht der `~/.bash_profile`, wenn sie nicht existiert wird diese nämlich an dessen Stelle ausgeführt
- `~/.bash_logout` - Optionale Datei, die Befehle beim Logout wie `clear` (Bildschirm löschen) ausführt
- `~/.bashrc` - Eine weitere, alte bash-Startupdatei, sie wird jedesmal mit einer neuen bash-Instanz abgearbeitet (neues Terminal öffnen), z.B. Aliase, PS1 (Farbprompt)

12.2 man-Pages

⇒ Seite 96 ff

ZIEL: Sich selber weiterhelfen können... mit dem Online-Manuals (Online = werden mit den Paketen ausgeliefert)

Man-Pages sind Kurzbeschreibungen von Kommandos und Dateien, denen oft Beispiele fehlen. (Von Programmierern für Programmierer), deswegen gibt es die HOWTOs im Internet bzw. als Paket nachinstallieren.

Benutzung

- Aufruf: `man <KOMMADO/LIBRARY/KONFIGDATEI>`
- Navigation (gemäß less-Pager):
 - Cursortasten, Enter, Leertaste
 - F (forward), B (backward)
 - G (go to the end), g (go to the begin)
- Suchen:
 - Vorwärts (nach unten): `/<PATTERN>`, Weitersuchen: `n` (next), Rückwärts nach oben suchen: `N` (Next)
 - Rückwärts (nach oben): `?<PATTERN>`, Weitersuchen: `n` (next), Rückwärts nach unten suchen: `N` (Next)
- Beenden: `q` (quit)

12.3 Funktionsweise

⇒ `man "manpath"`

Eine Konfigdatei, legt die Pfade fest: `/etc/manpath.config`

```
tux@deb8-2:~$ grep MANDB /etc/manpath.config
# MANDB_MAP      global_manpath      [relative_catpath]
MANDB_MAP        /usr/man             /var/cache/man/fsstnd
MANDB_MAP        /usr/share/man       /var/cache/man
MANDB_MAP        /usr/local/man       /var/cache/man/oldlocal
MANDB_MAP        /usr/local/share/man /var/cache/man/local
MANDB_MAP        /usr/X11R6/man       /var/cache/man/X11R6
MANDB_MAP        /opt/man             /var/cache/man/opt
tux@deb8-2:~$
```

Weiterhin wird die Variable `MANPATH` abgearbeitet, falls sie vorhanden ist.

Die man-Pages liegen komprimiert vor, beim Aufruf werden sie unter `/var/cache/man` dekomprimiert abgelegt. Die tatsächliche Pfade ausgeben: `manpath -c`

Der Cache lässt sich mit `catman` bereinigen, den Suchindex für `apropos` kann man mit `mandb` neu aufbauen.

Die man-Pages werden in sogenannten Abschnitten kategorisiert, siehe `man man`, z.B.

- Abschnitt 1: Programme für Benutzer (unter `/bin` oder `/usr/bin` oder `/usr/local/bin`)

- Abschnitt 4: Spezielle Dateien (Geräte-dateien, siehe `man 4 null`)
- Abschnitt 5: Dateiformate und Konventionen (z.B. Aufbau der Datei `/etc/fstab`: `man 5 fstab`)
- Abschnitt 6: Spiele (z.B.: `man 6 fiberlamp`)
- Abschnitt 8: Programme für root (unter `/sbin` oder `/usr/sbin` oder `/usr/local/sbin`)

TIPP für die Praxis: Option `-a` verwenden, um sich alle Abschnitte einer gegebenen `man`-Page anzeigen zu lassen

```
=> man -a mount
```

```
=> man -a crontab      (Hier werden nacheinander die Seiten der Abschnitte 1 und 5 ←  
geöffnet)
```

Eine schnelle Suche im Beschreibungstext des Pages ist mit

- `apropos <PATTERN>`
- `man -k <PATTERN>`

möglich. Hierbei wird aus *NAME* bzw. *BEZEICHNUNG* gesucht. Siehe auch `whatis <KOMMANDO>`.

Außerdem spielt hier auch das Kommando *whereis* eine Rolle, es gibt mehrere Dinge aus:

```
tux@deb8-2:~$ whatis whereis  
whereis (1)          - locate the binary, source, and manual page files for a command  
tux@deb8-2:~$
```

13 Topic 103.2 - Textströme mit Filtern verarbeiten

⇒ Seite 100 ff

ZIEL: Ausfiltern von Zeichenströmen auf bestimmte, interessante Inhalt, dabei ist mehrzeilige Arbeitsweise möglich, wie es z.B.

- `join` (join lines of two files on a common field)
- `paste` (merge lines of files)

möglich machen. Aber der Reihe nach, beginnen wir mit einfacheren Tools.

13.1 Komplettes Ausgeben von Dateien

- `cat` (catenate = verketteten), Abarbeitung von oben nach unten
- `tac` (= reverses cat), Abarbeitung von unten nach oben

Beispieldatei 1: `vi textfile1`

```
1      Birnen  
2      Äpfel  
3      Bananen  
4      Erdbeeren
```

Beispieldatei 2: `vi textfile2`

```
cat textfile2  
hellgrün  
rot  
gelb  
dunkelgrün
```

13.2 Zeilen filtern

Nur eine gewisse Zahl von Zeilen ausgeben:

- head (per Default die ersten 10 Zeilen ausgeben)
 - Wichtige Option: `-n <NUMBERS>` oder `-<NUMBERS>`
- tail (per Default die letzten 10 Zeilen ausgeben)
 - Wichtige Optionen:
 - * Zeilenzahl: `-n <NUMBERS>` oder `-<NUMBERS>`
 - * Folge dem Inputstrom (Monitoring) `-f` pder `--follow`
 - Bsp.: `tail -f /var/log/messages`

Im Praxistest:

```
tux@deb8-2:~$ head -2 textfile1 textfile2
==> textfile1 <==
1      Birnen
2      Äpfel

==> textfile2 <==
hellgrün
rot
tux@deb8-2:~$
```

AUFGABE: Lassen Sie sich nur die Zeilen 2 und 3 der Textdatei "textfile1" ausgeben, die insgesamt aus 4 Zeilen besteht.

- Lösung 1: Nach unten bis einschließlich zur gewünschten Zeile gehen, dann von diesem ausgeschnittenen Bereich nach oben gehend den relevanten Rest stehen lassen:

```
tux@deb8-2:~$ head -3 textfile1 | tail -2
2      Äpfel
3      Bananen
tux@deb8-2:~$
```

- Lösung 2: Von unten bis einschließlich zur gewünschten Zeile gehen, dann von diesem ausgeschnittenen Bereich von oben herab gehend den relevanten Rest stehen lassen

```
tux@deb8-2:~$ tail -3 textfile1 | head -2
2      Äpfel
3      Bananen
tux@deb8-2:~$
```

13.3 Zeilen nummerieren

- nl dateiname (Leerzeilen werden per Default NICHT nummeriert)
- cat -n dateiname

13.4 Pager (Textdateien seitenweise lesen)

- more (Besitzt keine solchen Komfortfunktionen wie less z.B. beim Bildschirmausschnittsblättern)
- less (= der Pager für man-Pages)

Motto: **Less is more.**

13.5 Tabulatorschritte in Leerzeichen umwandeln

Man kann mit `expand/unexpand` Tabulatorschritte in Leerzeichen umwandeln und umgekehrt.

=> AUFGABE: Welches Tool macht was genau?

```
tux@deb8-2:~$ whatis expand
expand (1)          - convert tabs to spaces

tux@deb8-2:~$ whatis unexpand
unexpand (1)       - convert spaces to tabs
```

Weitere interessante Werkzeuge finden wir auf Seite 103 ff:

- `format text: fmt`, Beispiel: Einen Text-Screenshot von virtueller (schwarzer) Konsole 1 formatieren

```
root@deb8-2:~# cat /dev/vcs1 > tty1.out
root@deb8-2:~#
root@deb8-2:~# fmt -w 40 tty1.out | head
5/732160 Bl&#xffffd;cke
[ 12.054488] systemd-fsck[247]:
/dev/sda7: sauber, 27/97536 Dateien,
22590/389          120 Bl&#xffffd;cke
[ 12.115165] systemd-fsck[206]:
/dev/sda6: sauber, 411/223104 Dateien,
36805/8           92416 Bl&#xffffd;cke
[ 12.212506] systemd-fsck[245]:
/dev/sda8: sauber,
42696/2297456 Dateien, 1211
```

- `print (pre stage, = Druckvorstufe z.B. für lp)`:
 - Einen Offset von 7 Zeichen hinzufügen: `pr, z.B. pr -o 7 myfile.txt | lp`
 - Mit `-h` kann man auch einen eigenen Header einfügen: `pr -o 7 -h "Dienste und Portnummern" /etc/services | less`
- `word count: wc`, Beispiel: `wc -l /etc/hosts`, Optionen:
 - Lines: `-l` oder `--lines`
 - Words: `-w` oder `--words`
 - Characters: `-m` oder `--chars`
- `hexadecimal dump: hexdump`, Beispiele:
 - Partitionstabelle ausgeben: `hexdump /dev/sda | head -32 | tail -5`
 - Die Datei `/etc/hostname` im Oktalformat ausgeben: `hexdump -o /etc/hostname`
- `sort characters: sort`, Beispiele:
 - Verzeichnisgrößen berechnen (`-n = numerisch`): `du -s /usr* | sort -n`
 - Top Ten der großen Verzeichnisse (`-r = reverse`): `du -s /usr/share/* | sort -nr | head`
 - Bequemer, weil "human" readeable (`-h`): `du -sh /usr/* | sort -h`
 - Sortierte Ausgabe in selbe Datei wieder zurückschreiben (`-o = output file`): `sort myfile.txt -o myfile.txt`
 - * ACHTUNG, so geht es nicht (DATENVERLUST!): `sort myfile.txt > myfile.txt`
(Zuerst wird wegen der Umleitung ein Filehandle eingerichtet, dann die Datei geleert (mit ">>" würde append durchgeführt) und zum Schluss die leere Datei sortiert.)

- Ignorieren des Case-Sensitivität: Option *-i*
- *uniq* (doppelte Zeilen entfernen): *uniq*, Beispiel: doppelte Dateien finden:
 - Im Experiment mit *find*, *md5sum*, *sort* und *uniq* führt dies schnell zu langen Zeilen, ein erster Versuch sieht so aus:

```
find -type f -exec md5sum {} \; | sort | uniq
```

- Eine praktikable, elegante Lösung ist dann schon komplexer, siehe

<http://www.commandlinefu.com/commands/view/3555/find-duplicate-files-based-on-size-first-then-md5-hash>

```
find -not -empty -type f -printf "%-30s\t\"%h/%f\"\n" | \
  sort -rn -t$'\t' | \
  uniq -w30 -D | \
  cut -f 2 -d $'\t' | \
  xargs md5sum | \
  sort | \
  uniq -w32 --all-repeated=separate
```

- PRÜFUNGSHINWEIS: Das Kommando *uniq* kann u.U. überflüssig werden, wenn *sort -u* (sort-option *uniq*) verwendet wird.
- *splitting files: split*, per Default splittet das Kommando Dateien in **1000 ZEILEN** große Stücke auf, Beispiel:

```
# Top Ten der größten Dateien unter /usr/bin:
ls -lhS /usr/bin/

# Die größte Datei jetzt aufsplitten (14 MB)
split /usr/bin/syncthing

# Zur Kontrolle die Größe der Einzelteile betrachten:
du -sh x*

# Jetzt die Teile wieder unter neuem Namen zusammensetzen:
cat x* > syncthing.merged
```

- *cutter* (Spaltenweises Auftrennen von maschinenlesbaren Dateien): *cut*,
 - Optionen:
 - * Auftrennung nach Feldern (Spalten): *-f 1-3* (nur Felder 1 bis 3 ausgeben)
 - * Angeben des Trenners (delimiter): *-d:* (Hier ist der Trenner ist ein (":"), per Default ist es ein Tabulator-Schritt)
 - * Wegschneiden nach einer bestimmten Anzahl von Characters: *-c*
 - Beispiele:
 - * Ausgeben aller Benutzernamen und deren User-IDs: *cut -d:-f1,3 /etc/passwd*
 - * Herausfiltern des prozentualen Wertes der root-Partitionsauslastung: *df / | tail -1 | cut -c 46-47*
- *paste files* (Dateiinhalte zeilenweise stumpf zu einer neuen, mehrspaltigen Datei zusammensetzen): *paste*, Beispiel:

```
tux@deb8-2:~$ cat textfile1 textfile2
1      Birnen
2      Äpfel
3      Bananen
4      Erdbeeren
hellgrün
rot
```

```

gelb
dunkelgrün
tux@deb8-2:~$

tux@deb8-2:~$ paste textfile1 textfile2
1      Birnen      hellgrün
2      Äpfel      rot
3      Bananen     gelb
4      Erdbeeren   dunkelgrün
tux@deb8-2:~$

```

- joining files (Dateiinhalte zeilenweise unter Verwendung eines gemeinsamen Feldes als Verbindungskriterium zur einer neuen, mehrspaltigen Datei zusammensetzen): *join*, Beispiel:

```

tux@deb8-2:~$ cat obst farbe
1: Birnen
2: Äpfel
3: Bananen
1: hellgrün
2: dunkelgrün
3: gelb
tux@deb8-2:~$

tux@deb8-2:~$ join -t: -j1 obst farbe
1: Birnen: hellgrün
2: Äpfel: dunkelgrün
3: Bananen: gelb
tux@deb8-2:~$

```

- translation of characters, Kommando *tr*, (es handelt sich um eine einfache Form von "sed", das Tool kann immer nur eine gleiche Anzahl von Zeichen übersetzen):

– Beispiele:

- * `echo ABC | tr 'A-Z' 'a-z'`
- * `echo ABC | tr '[:upper:]' '[:lower:]'`
- * Eine DOS-Datei in eine Unix-Datei umwandeln (Hintergrund: <https://de.wikipedia.org/wiki/Zeilenumbruch>):

```

tux@deb8-2:~$ vi dosfile
# Am Zeilenende immer ein STRG + v sowie STRG + m anhängen (Insert-Modus)^M
Zeile 1^M
Zeile 2^M

# Dateityp prüfen
tux@deb8-2:~$ file dosfile
dosfile: ASCII text, with CRLF line terminators
tux@deb8-2:~$

# Zeilenumbruch umwandeln ('tr' kann man nicht Dateinamen übergeben!)
cat dosfile | tr -d '\r' > unixfile
tux@deb8-2:~$
tux@deb8-2:~$ file unixfile
unixfile: ASCII text
tux@deb8-2:~$

```

- stream editor: *sed* (Mehr dazu unter Topic 103.7 - "Reguläre Ausdrücke")

14 Topic 103.3 - Perform basic file management

Literatur: Seite 111 ff, sowie <http://pemmann.de/doc/lpi1/kleine-kommandoreferenz.pdf>

ZIEL: Neben grundlegenden Datei- und Verzeichnisoperationen lernen wir auch Kompressionswerkzeuge und Archivierungstool kennen und verstehen, sie einzusetzen.

Philosophie: Alles ist eine Datei.

14.1 Kommandos für Dateioperationen

14.1.1 Kommando "ls"

Zweck: Auflisten von Verzeichnisinhalten und Dateieigenschaften

Hier gleich mal eine Grafik zur wichtigsten Option `-l` (long), die sehr viele Details liefert:

1 Bit für Dateityp:
 '.' = Reguläre Datei
 'd' = Directory
 'l' = symbol. Link

3*3 Rechtebits
Anzahl der Hardlinks
Zeitstempel
Dateiname (hier dank Tilde mit Speicherort ausgestattet)
Dateigröße [Byte] (= tatsächliche Größe)
User name
Group name

Mit dem Schalter `-l` ist alles zu sehen, außer:
 a) Inode-Nummer (Datei-Knotennummer): Option `'-i'`
 b) Anzahl der Blöcke, die die Datei im Filesystem beansprucht: Option `'-s'` (size)

****Philosophie: 'Alles ist eine Datei.'****
 === Kommando "ls"
 Zweck: Auflisten von Verzeichnisinhalten und Dateieigenschaften
 Wichtige Optionen:
 * `'-l'` - Langes Ausgabeformat

Wichtige Optionen:

- `-l` - Langes Ausgabeformat
- `-i` - Inode-Nummer (Datei-Knotennummer)
- `-s` - Size der belegten Blöcke
- `-a` - all Files: auch die versteckten "hidden" Files anzeigen (ausblenden von Profiles-Dateien, die separat angesprochen werden müssen, sollen sie eine Aktion mit eingeschlossen werden)
- `-h` - Dateigrößen menschengerecht (human readable) in gut lesbare Einheiten umrechnen
- `-S` - Nach der Größe der Datei sortieren
- `-r` - Reverse (umgekehrte Sortierung)

- **-R** - Rekursiv alle Unterverzeichnisse auflisten
- **-d** - Directory-Eigenschaften betrachten, NICHT den Inhalt des Verzeichnisses, Dateijoker "*" entkräften für nichtrekursives Arbeiten:

```
ls -l /usr/share/doc/*ba*
```

- **-t** Time sorted Output: Nach Änderungszeit (modification time) sortiert ausgeben

14.1.2 Kommando "cp"

ZIEL: Kopieren und evl. gleichzeitig es umbenennen:

Es gibt zwei grundsätzliche Modi:

- a) Kopiere eine oder mehrere Dateien in einen Zielordner: `cp /etc/fstab /etc/hostname /etc/hosts /tmp`, ein weiteres Beispiel:

HINWEIS: Wir verwenden hier das Kommando *touch*, das eigentlich ein Zeitstempelwerkzeug ist, um drei leere Dateien anzulegen.

```
touch a b c
mkdir d
cp a b c d
ls -l d/
```

- b) Kopiere eine Datei auf eine andere, wobei sie ohne eine Sicherungskopie anzufertigen überschrieben wird, falls nicht der Schalter **-b** (backup) verwendet wird.

```
echo Hallo... > dateil.txt
cp -b /etc/hostname dateil.txt
cat dateil.txt dateil.txt~
```

AUFGABEN:

1. Kopieren Sie in Ihr Verzeichnis *~/Dokumente* die Dateien */etc/hosts.allow* und */etc/hosts.deny* hinein.
=> `cp /etc/hosts.allow /etc/hosts.deny ~/Dokumente/`
2. Kopieren Sie in Ihr Verzeichnis *~/Dokumente* in das Verzeichnis *'~/Downloads'* hinein.
=> `cp -r ~/Downloads/ ~/Dokumente/`
3. Kopieren Sie die Datei */etc/hosts* unter Umbenennung zu *IP-und-Hostnamen.txt* ins Verzeichnis *~/Dokumente*.
=> `cp /etc/hosts ~/Dokumente/IP-und-Hostnamen.txt`

Bitte weitere Optionen in *man cp* studieren, welche Optionen vereint der der Schalter **-a**?

Siehe <https://wiki.ubuntuusers.de/cp/> <https://wiki.ubuntuusers.de/cp/>

- **-R** - Rekursiv arbeiten
- **--preserve=all** - Preserve all: Dateirechte, Zeitstempel usw. erhalten
- **-d** - no Dereference SymLinks: belasse die Links beim Kopieren so wie sie sind, nicht die dahinter stehenden Dateien kopieren

Beispiel (als root auszuführen):

```
root@deb8-2:~# cp -a /var/lib/nfs /root/var_lib_nfs
root@deb8-2:~# ls -ld /root/var_lib_nfs
drwxr-xr-x 1 statd nogroup 134 Jan 19 16:29 /root/var_lib_nfs
```


14.1.3 Kommando "mv"

Das Kommando bedeutet *move* und macht den Unterschied zu *cp* aus:

- Es benennt eine Datei oder ein Verzeichnis um, wenn die Operation im selben Pfad geschieht.
- Es verschiebt Objekte, wenn als Zielpfad ein anderes Verzeichnis angegeben wird.

ACHTUNG: Anders als *cp* kann *mv* immer rekursiv arbeiten (um die Umlagerung im Dateibaum zu ermöglichen), außerdem bleiben dabei naturgemäß Dateirechte, Zeitstempel usw. erhalten.

14.1.4 Kommando "rm"

Mit *remove* werden zuerst einmal nur Dateien gelöscht, will man Verzeichnisse löschen, muss rekursiv gearbeitet werden: Option *-r*

Aus Sicherheitsgründen sollte ein Alias eingerichtet werden, der versehentliches Löschen durch interaktives Rückfragen vermeiden hilft:

```
root@deb8-2:~# alias rm='/bin/rm -i'
root@deb8-2:~# cp -a /etc/ /root/etc_$(date +%F)
root@deb8-2:~#
root@deb8-2:~#
root@deb8-2:~# du -sh /etc/ /root/etc_2016-01-20/
12M      /etc/
12M      /root/etc_2016-01-20/
root@deb8-2:~#
root@deb8-2:~# rm -r /root/etc_2016-01-20/
/bin/rm: in Verzeichnis „/root/etc_2016-01-20/“ absteigen? j
/bin/rm: reguläre Datei „/root/etc_2016-01-20/fstab“ entfernen? j
/bin/rm: in Verzeichnis „/root/etc_2016-01-20/.java“ absteigen? j^C
root@deb8-2:~#
```

Wie kann dann aber ein schnelles Löschen erzwungen werden?

1. Alias für einmalige Ausführung umgehen: Backslash voranstellen: `\rm -r /root/etc_2016-01-20/`
2. Alias aus der Umgebung entfernen: `unalias rm ;rm -r /root/etc_2016-01-20/`
3. Mit Hilfe der *rm*-Option *-f* (force): `rm-rf /etc/*host*`

14.2 Kommandos für Verzeichnisoperationen

14.2.1 Kommandos *mkdir* und *rmdir*

Mit *mkdir* können neue Verzeichnisse angelegt werden, komplette Strukturen nur mit der Option *-p* (parents)

```
mkdir -p ~/Briefe/Privat/Einkaufen/Lebensmittel/Gesunde
```

Sicheres Löschen von Verzeichnisstrukturen am besten mit *rmdir* durchführen, das nur leere Ordner löschen kann; hier gibt es auch wieder *-p* (parents):

```
rmdir -p ~/Briefe/Privat/Einkaufen/Lebensmittel/Gesunde
```

Metapher: Von rechts solange zurückrollen, bis man auf einen Widerstand stößt: das kann eine versteckte Datei z.B. *.keep* oder auch ein leerer Ordner sein.

VORSICHT: Diese Art Aufräumaktionen NICHT mit der Systemwurzel durchführen!!

```
tux@deb8-2:~$ touch ~/Briefe/Privat/.keep
tux@deb8-2:~$ rmdir -p ~/Briefe/Privat/Einkaufen/Lebensmittel/Gesunde
rmdir: konnte Verzeichnis „/home/tux/Briefe/Privat“ nicht entfernen: Das Verzeichnis ist ↩
nicht leer
tux@deb8-2:~$
```

14.3 Weitere wichtige Kommandos für Dateiarbeit:

- **touch** - Files "berühren", um den Zeitstempel zu ändern, wenn das File nicht existiert, wird es einfach erzeugt
 - FRAGE: Welchen der Zeitstempel kann *touch* nicht ändern?
 - ANTWORT: Die *Change Inode Time* (chown/chmod; find-Kürzel: -cmin/-ctime) zu ändern ist nicht direkt bzw. unabhängig möglich!
 - * Modification Time (vi / nano, libreoffice; find-Kürzel: -mmin/-mtime): Aus der man-Page: *-a change only the access time*
 - * Access Time (cat, more, less; find-Kürzel: -amin/-atime): Aus der man-Page: *'-m change only the modification time'*
- **dd** - disk dump: Files bestimmter Größe erzeugen, z.B. 100kB: `dd if=/dev/zero of=myLittleFile.img bs=100k count=1`
- **find** - finden von Files aller Art, grundlegende Syntax `find [PFADE] [PFADTIEFE] [DATEISYSTEM] [BEDINGUNGen...] [AKTIONen...]`
 - Wichtige Optionen:
 - * Case-Sensitiv: `-name <PATTERN>` (es arbeitet immer rekursiv, auch versteckte Dateien werden mit gefunden)
 - * Ignore Case-Sensitivity: `-iname <PATTERN>`
 - * Dateityp-Suche: `-type <fdlsp>` (Typen: file, directory, symlink, socket, pipe)
 - * Max. Verzeichnistiefe: `-maxdepth <ZAHL>`, Beispiel: `find /-maxdepth 3 -name resolv.conf`
 - * Einschränkung auf das lokale Dateisystem: `-xdev`, Beispiel: `find /-maxdepth 5 -xdev -name "*bashrc*"`
 - * Zeitstempel: siehe oben bei *touch*
 - * Besitzer von Files: `'-user <USERNAME/ID>, '-uid <UID>`
 - * Leerzeichen durch Null-Zeichen für nachfolgende Ausführung via `xargs` ersetzen: `-print0`

```
tux@deb8-2:~$ find Dokumente/2016-01-21/* -print0 | xargs -0 du -h
4,0K      Dokumente/2016-01-21/dateimanagement-redirects-pipes.adoc
0         Dokumente/2016-01-21/dateiname mit spaces.txt
tux@deb8-2:~$
```

```
### Weitere Beispiele:
# Alle Dateien, die in eben dieser Minute durch 'make install' nach /usr kopiert ↩
wurden,
# ausgeben:
find /usr -cmin 1

# Ein Verzeichnis als Vergleichswert für Zeitstempelsuche heranziehen
find -cnewer /home/tux/TimeStamp/ -type f -mmin -60
```

14.4 Wildcards für Dateinamen und -pfade (S. 117 f)

- Stern (*): Bringt Übereinstimmung für eine beliebig lange Zeichenkette beliebiger Zeichen, auch kein Zeichen
- Fragezeichen (?): Bringt Übereinstimmung für ein beliebiges Zeichen

- In eckigen Klammern (*[abcxyz]*): Bringt Übereinstimmung für ein beliebiges Zeichen aus der hier angegebenen Menge bzw. des Bereiches
Negation: Mit einem nach der öffnenden Klammer eingefügten Ausrufezeichen (!) wird Umkehrung erreicht (ACHTUNG: Bei regex ist dies ähnlich, aber mit einem anderen Zeichen verbunden: (^) Zum Beispiel nicht die Dateien ausgeben, die entweder auf eine 1 oder eine 2 enden: `ls -l textfile[!12]`

14.5 Wichtige Kommandos für Archivierung

14.5.1 tar - Für Tapes geschaffenes Archivierungsprogramm:

- Ohne Komprimierung: `tar cf Dokumente.tar Dokumente/`
- Mit GnuTar zugleich komprimieren (Dies können alte Unixe nicht!): `tar czf Dokumente.tar.gz Dokumente/`
 - Hierbei bedeuten die Optionen: `c` = create, `z` = Packen mit `gzip`, `f` = filename of the archive
 - Weitere Optionen:
 - * `-v` = verbose
 - * `-C` = change directory (`mkdir TMP ;tar xzvf Dokumente.tar.gz -C TMP/`)
 - * `-x` = eXtract
 - * `-j` = Packen mit `bzip2`
 - * `-J` = Packen mit `xz`
 - * `-t` = table of content ausgeben, nicht entpacken

14.5.2 cpio (S. Seite 121, über STDIN / STDOUT kopieren): Einsatz beim rpm-Paketformat (rpm2cpio), es kennt drei Modi:

A. Copy-IN: Schalter `-i` oder `--extract` (ins Dateisystem extrahieren)

- Beispiel (im Verzeichnis /tmp): `cpio -i < ~/textfiles.cpio`

B. Copy-OUT: Schalter `-o` oder `--create` (aus dem Dateisystem ins Archiv kopieren; = Erzeugung des Archivs)

- Beispiel: `ls *.txt | cpio -o > textfiles.cpio`
- Dateityp feststellen: `file textfiles.cpio` (textfiles.cpio: cpio archive)

C. Copy-TROUGH: Schalter `-d` oder `--pass-through` (Via Standardinput Daten aufnehmen und ins Dateisystem kopieren)

- Beispiel "Benutzer manuell erzeugen" (auszugsweise, als root durchzuführen):

```
# Heimatverzeichnis erzeugen
mkdir /home/lisa

# Vorlagen einkopieren (cp würde dot-files mit kopieren!)
cd /etc/skel
find | cpio -pdm /home/lisa
```

14.6 Wichtige Kommandos für Komprimierung

Anders als bei den Werkzeugen `tar` und `cpio` werden die Daten hier lediglich komprimiert, nicht in ein Archiv überführt.

- `gzip` - Entpacken: `gunzip` oder `gzip -d`
- `bzip2` - Entpacken: `bunzip2` oder `bzip -d`

- xz - Entpacken: `unxz` oder `xz -d`

Die typische Unix-Komprimierung läuft so, dass direkt mit der Originaldatei gearbeitet wird, die danach ein anderes Dateisuffix erhält und somit umbenannt wurde!

Am Beispiel von `gzip`:

```
tux@deb8-2:~$ cp /etc/services .
tux@deb8-2:~$
tux@deb8-2:~$ ls -lh services
-rw-r--r-- 1 tux tux 20K Jan 22 10:40 services
tux@deb8-2:~$
tux@deb8-2:~$ gzip -9 services
tux@deb8-2:~$
tux@deb8-2:~$ ls -lh services
ls: Zugriff auf services nicht möglich: Datei oder Verzeichnis nicht gefunden
tux@deb8-2:~$
tux@deb8-2:~$ ls -lh services*
-rw-r--r-- 1 tux tux 7,4K Jan 22 10:40 services.gz
tux@deb8-2:~$
tux@deb8-2:~$ file services.gz
services.gz: gzip compressed data, was "services", last modified: Fri Jan 22 10:40:29 2016, ←
    max compression,
    from Unix
tux@deb8-2:~$
tux@deb8-2:~$ ### Wollen wir den Inhalt der komprimierten Datei sehen, ohne sie zu ←
    entpacken, müssen wir
    mit '-c' über STDOUT gehen:
tux@deb8-2:~$ gunzip -c services.gz | less
```

AUFGABE: Führen Sie die selben Operationen mit `bzip2` und `xz` durch.

Beispielhafte Lösungen:

```
cp /etc/services .
bzip2 services
cp /etc/services .
xz services
cp /etc/services .
xz -9 -c services > services-level9.xz
xz -1 -c services > services-level11.xz

# Zum besseren Vergleichen nur die tatsächliche Dateigröße in Byte und den Dateinamen ←
    ausgeben
ls -l services* | awk '{print$5, $9}'
19605 services
7229 services.bz2
7548 services.gz
7484 services-level11.xz
7156 services-level9.xz
7156 services.xz
```

14.7 Archivierung mit Komprimierung verbinden

Angenommen, es liegt nicht das moderne GNU-tar vor, sondern ein ein altes Unix-System, können die beiden Sparten trotzdem miteinander arbeiten:

a) Erzeugen eines komprimierten Archivs:

```
# Der Bindestrich bedeutet in diesem Kontext: STDOUT für das Erzeugen des Archiv-files ←
    benutzen
tar cvf - foodir | gzip > foo.tar.gz
```

b) Entpacken des komprimierten Archivs:

```
# Der Bindestrich bedeutet in diesem Kontext: STDIN für das Entpacken des Archiv-files ↔
  benutzen
gunzip -c foo.tar.gz | tar xvf -
# oder
gunzip < foo.tar.gz | tar xvf -
```

Bitte den Schalter `-c` bei den Komprimierern `gzip`, `bzip2` und `xz` wahrnehmen, durch ihn lassen sich erst die von `tar` stammenden Daten weiterverarbeiten.

15 Topic 103.4 - Use streams, pipes and redirects**15.1 Standard-Datenströme der Unix-Shells**

Siehe S. 125 oben

- **Kanal STDIN:** Dateideskriptor = 0, Umleitungs-Kürzel: "<", anhängende Arbeitsweise als Here-Dokument: "<<MARKE"
- **Kanal STDOUT:** Dateideskriptor = 1, Umleitungs-Kürzel: ">", anhängende Arbeitsweise ">>"
- **Kanal STDERR:** Dateideskriptor = 2, Umleitungs-Kürzel: "2>", anhängende Arbeitsweise "2>>"

ANMERKUNG: Den Dateideskriptor kann man weglassen, wenn das Kürzel eine eindeutige Interpretation durch die "Pfeilrichtung" zulässt, ansonsten muss die Ziffer angegeben werden. Bei STDERR muss sie immer mit notiert werden.

a) Zu Kanal STDIN:

Motto: Ein Kommando liest von der Tastatur:

```
tux@deb8-2:~$ wc -l

Hallo?
ABC...
Fertig?
5
tux@deb8-2:~$
tux@deb8-2:~$ ##### Ja, durch Drücken von 'STRG + D' (End of File, EOF)
```

b) Zu Kanal STDOUT:

```
tux@deb8-2:~$ df -h /
Dateisystem    Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda5      28G   6,2G   21G   24% /
tux@deb8-2:~$
```

c) Zu Kanal STDERR (wird genauso wie STDOUT ins Terminal gesendet):

```
tux@deb8-2:~$ df -h /SERVER
df: „/SERVER“: Datei oder Verzeichnis nicht gefunden
tux@deb8-2:~$
```

Siehe <http://www.netzmafia.de/skripten/unix/unix2.html>

15.2 Pipelines zum Weiterverarbeiten der Datenströme

Das Kürzel für Pipes ist lediglich ein einzelnes Bar-Zeichen (|), dabei wird die Ausgabe eines Kommandos immer zur Eingabe des nächsten (STDOUT wird zu STDIN):

```
=> cmd1 | cmd2 | cmd3
```

Beispiel:

```
tux@deb8-2:~$ df -h / | tail -1
/dev/sda5      28G   6,2G   21G   24% /
tux@deb8-2:~$
```

Beim Durchreichen der Daten tritt jetzt allerdings ein Problem auf: **Zwischenergebnisse gehen verloren!** Im Beispiel von oben ist uns ja der Header, also die folgende Zeile verlorengegangen:

```
Dateisystem    Größe Benutzt Verf. Verw% Eingehängt auf
```

Wie kann man nun darauf zurückgreifen: Es muss ein T-Stück (engl: *tee*) eingebaut werden:

```
tux@deb8-2:~$ df -h / | tee header+content.txt | tail -1
/dev/sda5      28G   6,2G   21G   24% /
tux@deb8-2:~$
tux@deb8-2:~$ cat header+content.txt
Dateisystem    Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda5      28G   6,2G   21G   24% /
tux@deb8-2:~$
```

Siehe: https://de.wikipedia.org/wiki/Tee_%28Unix%29

Dort liest man auch über die für uns wichtige Option *-a* (append): "Mit *-a* wird an die angegebene Datei angehängt (anstatt eine neue leere Datei zu eröffnen)"

15.3 Umleitungen der Standardkanäle durchführen

Siehe oben bei der Erklärung der drei Standard-Ströme.

Es lassen sich Kanäle verbinden, dazu wird das Kaufmanns-Und eingesetzt:

```
Aus Prüfung B)
In bash, inserting 2>&1 after a command redirects
A. standard error to standard input.
B. standard input to standard error.
C. standard output to standard error.
D. standard error to standard output.
E. standard output to standard input.
Richtige Antwort: D
```

Hintergrund: Für typische Aufgaben wie das Kompilieren von Anwendungen soll jeglicher Output, auch Fehlermeldungen gespeichert werden:

```
make > my-important-file.log 2>&1
```

Weitere Verbindungsmöglichkeiten: https://www.thomas-krenn.com/de/wiki/Bash_stdout_und_stderr_umleiten

```
### Bash redirection

# stdout -> Datei umleiten
programm > Datei.txt

# stderr -> Datei umleiten
```

```
programm 2> Datei.txt

# stdout UND stderr -> Datei umleiten
programm &> Datei.txt

# stdout -> stderr
programm 1>&2

#stderr -> stdout
programm 2>&1
```

Ein Beispiel mit einer Datei, in der Klein- und Großschreibung wild gemischt sind. Es soll alles in Kleinbuchstaben umgewandelt werden:

```
tux@deb8-2:~$ tr '[:upper:]' '[:lower:]' < kontakte | nl
 1      müller
 2      meier
 3      schulze
tux@deb8-2:~$
```

15.4 Weitere Verarbeitungskürzel der bash

- Kommandozeilen OHNE Prüfung des Exitcodes aneinanderreihen mit Hilfe dem Semikolon (;)
- Kurztests auf Erfolg durchführen
 - Nur bei Erfolg weiterarbeiten (zwei Ampersand-Zeichen): &&
 - Nur bei Misserfolg weiterarbeiten (zwei Bar-Zeichen): ||

Beispiele:

```
tux@deb8-2:~$ cat /etc/hostname && echo Alles OK.
deb8-2
Alles OK.
tux@deb8-2:~$
tux@deb8-2:~$ cat /etc/HOSTNAME && echo Alles OK.
cat: /etc/HOSTNAME: Datei oder Verzeichnis nicht gefunden
tux@deb8-2:~$
```

```
tux@deb8-2:~$ cat /etc/hostname || echo Wir gehen zu Plan über....
deb8-2
tux@deb8-2:~$
tux@deb8-2:~$ cat /etc/HOSTNAME || echo Wir gehen zu Plan über....
cat: /etc/HOSTNAME: Datei oder Verzeichnis nicht gefunden
Wir gehen zu Plan über....
tux@deb8-2:~$
```

16 Topic 103.5 - Create, monitor and kill processes

Siehe Seite 128 ff

16.1 Überwachen von Prozessen

Wichtige Werkzeuge sind unter anderem:

- **ps** - Process Snapshots (ohne Option: Nur die Prozesse der aktuellen Shellumgebung werden angezeigt)
 - Optionen (Achtung: Bindestriche zu verwenden, bedeutet die Syntax gemäß POSIX/Unix (Stichwörter: AT&T, Solaris) anzuwenden, keine Bindestriche dagegen heißt BSD-Syntax zu wollen):
 - * **a** - Alle Prozesse ausgeben, auch die von anderen Benutzern
 - * **u** - User defined Format (ausführlich)
 - * **x** - Nicht mit einer Shell verbundenen Daemons ebenfalls ausgeben (Kennzeichen in Spalte *TTY*: (?))
 - * **l** - Long Listing, auch Prioritäts- und Nice-werte auflisten
 - * **-C <DIENST>** - Die Prozesse, die ein bestimmter Dienst verursacht, auflisten, Bsp.: `ps -C sshd`
 - * **-U <USER>** - Die Prozesse, die ein bestimmter Nutzer verursacht, auflisten
- **pstree** - Prozess in Baumform ausgeben
 - Optionen:
 - * **-p** - Prozess-IDs ausgeben
 - * **-n** - Prozess-ID numerisch sortieren
- **top** - Prozesse in einem sich selbst aktualisierenden Monitor präsentieren (top = die größten Ressourcenverbraucher zuerst)
 - Wichtige Ausgaben im Programmfenster:
 - * Erste Zeile: Statistiken gemäß Kommando *uptime*
 - * Zweite und dritte Zeile: Statistiken über Prozesse und CPU-Auslastung
 - * Vierte und fünfte Zeile: RAM-Auslastung gemäß Kommando *free* anzeigen
 - * Die weiteren Zeilen: Live-Monitor
 - Interaktive Bedienung des Programmes:
 - * **q** - quit
 - * **1** - alle Prozessoren anzeigen
 - * **u <USERNAME>** - Nach einem Nutzer filtern
 - * **r** - Renice von Prozessen durchführen (Nice-Wert nachträglich ändern)
 - * **k** - Killen von Prozessen (PID und Signalnummer angeben)
 - Wichtige Optionen:
 - * **-i** - Nur aktive Prozesse anzeigen
 - * **-d <TIME>** - Delay-Zeit [s] angeben (Standard = 1s)
 - * **-q** - Ausgaben in Echtzeit liefern
 - Praxisbeispiel: Ein Programm frisst so viel Ressourcen, dass das System langsam wird (Out of Memory-Killer ⇒ OOM-Killer)
 - * Firefox mit laufenden Flash-Plugins
 - * Den "Taskmanager" top starten: top
 - * Die vorgeschlagene PID übernehmen oder anpassen, Taste *k* (kill) drücken
 - * Die vorgeschlagene Signalnummer (15) übernehmen oder anpassen, eine 9 macht ein Clean up des Programmes unmöglich (SIGKILL)!
 - * Beobachten, ob der Systemzustand wieder in Ordnung ist
 - * Beenden des Programmes *q*

16.2 Arbeiten mit Signalen

Recherche-Aufgaben: Suchen Sie nach Bildern, die im Zusammenhang mit Unix/Linux und dem Prozesszyklus stehen (Lebenszyklus, fork(), suspend, running)

⇒ http://www2.htw-dresden.de/~fritzschn/BS/bs_script.html

Seite 133: Bitte die folgenden Signale lernen:

- SIGHUP = 1, No Hangup, das bedeutet heute: Daemons sollen ihre Konfiguration neu einlesen
- SIGINT = 2 (Via Tastatur: STRG + c), die Abarbeitung von Programmen abbrechen
- SIGKILL = 9, die Abarbeitung eines Programmen gewaltsam beenden
- SIGTERM = 15, die Abarbeitung von Programmen auf normalen Wege terminieren
- SIGCONT = 18, (Via Tastatur: STRG + q) die Abarbeitung eines Programmes nach einem Stop weiterarbeiten lassen
- SIGSTOP = 19, (Via Tastatur: STRG + s) die Abarbeitung eines Programmes pausieren lassen (einen Stop einlegen)

AUFGABE: Bitte studieren Sie die Einführung von *man 5 signal* sowie die Tabellen der Signalnummern. (5 min) * Default-Verhalten?

Prozesse können Signale abfangen, weil sie der Meinung sind, bestimmte Aufgaben zu Ende bringen zu müssen (Signal 15 ermöglicht: Clean up).

Ein Signal allerdings kann nicht abgefangen werden: SIGKILL (Signal Nr. 9).

Übung: Starten Sie zum Testen ein ziemlich nutzloses Programm im Hintergrund: `sleep 60 &` (gern auch mehrmals)

```
tux@deb8-2:~$ sleep 60 &
[1] 4061
tux@deb8-2:~$
tux@deb8-2:~$ sleep 60 &
[2] 4062
tux@deb8-2:~$ sleep 60 &
[3] 4063
tux@deb8-2:~$
tux@deb8-2:~$ kill 4061
tux@deb8-2:~$
[1] Beendet          sleep 60
tux@deb8-2:~$
tux@deb8-2:~$ kill -9 4062
tux@deb8-2:~$
[2]- Getötet        sleep 60
tux@deb8-2:~$
```

16.3 Weitere Tools zur Prozessmanipulation

- **killall** - Via exakten Namen alle gleichnamigen Prozesse manipulieren (max. 64 Prozesse)

```
tux@deb8-2:~$ sleep 60 &
[1] 4519
tux@deb8-2:~$ sleep 60 &
[2] 4520
tux@deb8-2:~$ sleep 60 &
[3] 4521
tux@deb8-2:~$ sleep 60 &
[4] 4522
tux@deb8-2:~$
tux@deb8-2:~$
```

```
tux@deb8-2:~$
tux@deb8-2:~$ killall sleep
[1] Beendet sleep 60
[2] Beendet sleep 60
[3]- Beendet sleep 60
[4]+ Beendet sleep 60
tux@deb8-2:~$
```

- **pgrep** - Eine Prozesssuche gestaltet sich viel schneller als mit `ps aux | grep <PATTERN>`.)
- **pkill** - Vorsicht hier wird mit einem Pattern gearbeitet, es kann u.U. zu viel treffen!
- **xkill** - Grafische Anwendungen mit einem einfachen Linksklick gewaltsam beenden (Aus der man-Page: "This program is very dangerous, but is useful for aborting programs that have displayed undesired windows on a user's screen.")
- Weitere Monitoring-Tools:
 - **htop** (Debian: "Htop ist ein ncurses-basierter Prozessbetrachter ähnlich wie top, jedoch ermöglicht er Ihnen die Liste vertikal und horizontal zu durchlaufen")
 - **dstat** (Debian: "Vielseitiges Ressourcenstatistikwerkzeug. Dstat ist ein vielseitiger Ersatz für vmstat, iostat und ifstat.)
 - **monitorix** (Webbasiertes, grafischen Auswertungstool (CGI-basiert), siehe <http://www.monitorix.org/screenshots.html>)

Tipp: Beenden verschiedenster Apps:

- STRG + c (Signal 2)
- STRG + d (done, End of File)
- ESC (alsamixer)
- q
- quit
- exit

16.4 Jobverwaltung der bash

Motivation: aus einer Shell-Sitzung lassen sich viele Programme gleichzeitig starten und verwalten (Komfort).

Wichtige Kürzel für den bash-Mechanismus:

- **&** Ein einzelnes Ampersand, was man dem Kommandoaufruf mitgibt, startet das Programm im Hintergrund und führt es aus (*ding* ist ein Englisch/Deutsch-Dictionary, es muss nachinstalliert werden), ⇒ synchrone Abarbeitung, der Shellprompt wird frei gemacht

```
tux@deb8-2:~$ ding &
[1] 4663
tux@deb8-2:~$
```

- **fg** - foreground: Ein im Hintergrund laufendes Programm nach vorn holen, so dass es mit Tastenkombinationen manipuliert werden kann
- **bg** - background: Ein mit `STRG + z` suspendiertes Programm im Hintergrund fortfahren lassen (so als ob man ein `%` angehängt hätte)

```
tux@deb8-2:~$ ding
^Z
[1]+  Angehalten                ding
tux@deb8-2:~$
tux@deb8-2:~$ jobs
[1]+  Angehalten                ding
tux@deb8-2:~$
tux@deb8-2:~$ bg
[1]+  ding &
tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ jobs
[1]+  Läuft                    ding &
tux@deb8-2:~$
```

- **jobs** - Jobs auflisten
- **kill <%PID>** - Prozesse über ihre Job-Nummer manipulieren

Beispiel: Ein Programm (bc = basic calculator) manuell in den Hintergrund suspendieren und aus dem Vordergrund heraus normal beenden:

```
tux@deb8-2:~$ bc -l
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.

3/1.5
2.00000000000000000000000000000000

^Z
[1]+  Angehalten                bc -l
tux@deb8-2:~$ killall bc
tux@deb8-2:~$ killall bc
tux@deb8-2:~$ killall bc          ### Kein Erfolg: Der Prozess müsste hier mit '-9' ←
    gekillt werden!
tux@deb8-2:~$
tux@deb8-2:~$ jobs
[1]+  Angehalten                bc -l
tux@deb8-2:~$
tux@deb8-2:~$ kill -15 %1        ### So gelingt ein sanftes Beenden

[1]+  Angehalten                bc -l
tux@deb8-2:~$
[1]+  Beendet                  bc -l
tux@deb8-2:~$
```

17 Topic 103.6 - Modify process execution priorities

Siehe Seite 140

Hier spielen für LPI zwei Kommandos eine Rolle:

- **nice** (voranzustellendes Kommando) - Einen anderen Freundlichkeitswert beim Starten eines Kommandos setzen, per Default wird der positive Wert 10 benutzt)
- **renice** (im nachhinein auszuführen, PID erforderlich) - Einen anderen Freundlichkeitswert im nachhinein, per Default wird der positive Wert 10 benutzt)

Darüber hinaus gibt es speziell für Datenträgeroperationen (kopieren, Backups anfertigen) das Kommando: *ionice*

Wie funktioniert das Ganze?

- nice-Werte sind nur Empfehlungen, der Kernel kann sie überstimmen (Bitte Spalte PRI bei *ps l* beachten: Dies legt allein der Kernel fest!)
- nice-Werte verhalten sich zu Prioritäten umgekehrt proportional: Hoher nice-Wert = geringe Priorität und umgedreht.
- Bereich für nice: -20 bis +19
 - Benutzer *root*: Er darf wie immer alles, also auch in den negativen Bereich gehen
 - Normale Benutzer: Nur positive Werte sind erlaubt.

Ein paar Beispiele:

```
tux@deb8-2:~$ nice sleep 120 &
[1] 7346
tux@deb8-2:~$
tux@deb8-2:~$ ps l
F  UID  PID  PPID PRI  NI   VSZ  RSS WCHAN  STAT TTY          TIME COMMAND
0  1000 1423 1418 20   0  24276  676 -        Ss+ pts/0        0:00 bash
0  1000 2633 1418 20   0  23920  416 -        Ss  pts/2        0:00 bash
0  1000 3312 1418 20   0  23960 4388 -        Ss  pts/1        0:00 bash
0  1000 4053 1418 20   0  23876  220 -        Ss+ pts/4        0:00 bash
0  1000 4338 1418 20   0  23936  352 -        Ss+ pts/5        0:00 bash
0  1000 6751 6748 20   0  23840 5736 -        Ss+ pts/7        0:00 -bash
1  1000 6801   1  20   0  24372 2336 -        S   pts/7        0:00 /usr/bin/dbus-launch ←
    --autolaunch ...
0  1000 7346 3312 30  10   5808  632 -        SN  pts/1        0:00 sleep 120
0  1000 7347 3312 20   0  10688 2092 -        R+  pts/1        0:00 ps l
tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ renice -n 15 7346
7346 (Prozesskennung) alte Priorität 10, neue Priorität 15
tux@deb8-2:~$
tux@deb8-2:~$ ps l
F  UID  PID  PPID PRI  NI   VSZ  RSS WCHAN  STAT TTY          TIME COMMAND
0  1000 1423 1418 20   0  24276  676 -        Ss+ pts/0        0:00 bash
0  1000 2633 1418 20   0  23920  416 -        Ss  pts/2        0:00 bash
0  1000 3312 1418 20   0  23960 4388 -        Ss  pts/1        0:00 bash
0  1000 4053 1418 20   0  23876  220 -        Ss+ pts/4        0:00 bash
0  1000 4338 1418 20   0  23936  352 -        Ss+ pts/5        0:00 bash
0  1000 6751 6748 20   0  23840 5736 -        Ss+ pts/7        0:00 -bash
1  1000 6801   1  20   0  24372 2336 -        S   pts/7        0:00 /usr/bin/dbus-launch ←
    --autolaunch ...
0  1000 7346 3312 35  15   5808  632 -        SN  pts/1        0:00 sleep 120
0  1000 7349 3312 20   0  10688 2096 -        R+  pts/1        0:00 ps l
tux@deb8-2:~$
tux@deb8-2:~$ nice --20 sleep 120 &
[1] 7356
tux@deb8-2:~$ nice: die Priorität kann nicht gesetzt werden: Keine Berechtigung

tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ nice --19 sleep 120 &
[2] 7357
tux@deb8-2:~$
tux@deb8-2:~$ ps l
F  UID  PID  PPID PRI  NI   VSZ  RSS WCHAN  STAT TTY          TIME COMMAND
0  1000 1423 1418 20   0  24276  676 -        Ss+ pts/0        0:00 bash
0  1000 2633 1418 20   0  23920  416 -        Ss  pts/2        0:00 bash
```

```

0  1000  3312  1418  20   0  23960  4388  -    Ss  pts/1    0:00  bash
0  1000  4053  1418  20   0  23876   220  -    Ss+ pts/4    0:00  bash
0  1000  4338  1418  20   0  23936   352  -    Ss+ pts/5    0:00  bash
0  1000  6751  6748  20   0  23840  5736  -    Ss+ pts/7    0:00  -bash
1  1000  6801   1    20   0  24372  2336  -    S   pts/7    0:00  /usr/bin/dbus-launch ←
    --autolaunch ...
0  1000  7356  3312  20   0   5808   688  -    S   pts/1    0:00  sleep 120
0  1000  7357  3312  39  19   5808   672  -    SN  pts/1    0:00  sleep 120
0  1000  7358  3312  20   0  10688  2088  -    R+  pts/1    0:00  ps 1
tux@deb8-2:~$
[1]-  Fertig                nice --20 sleep 120
[2]+  Fertig                nice +-19 sleep 120
tux@deb8-2:~$

```

Praktische Anwendungsfälle:

a) root möchte ressourcenschonend den Suchindex von *locate* erneuern:

```
=> nice -19 updatedb &
```

b) tux möchte ressourcenschonend eine find-Suche für eigenen Bedarf durchführen:

```
=> nice -n +19 find 2> /dev/null > allMyFiles.txt
```

c) root will beim Erstellen von GnuGP-Schlüsseln mehr Dampf geben (für duplicity-Backups):

```
=> nice --19 gpg --gen-key
```

17.1 Optionen und Aufrufmöglichkeiten:

Dies alles ist als Benutzer möglich:

```

tux@deb8-2:~$ nice ++13 sleep 2
tux@deb8-2:~$
tux@deb8-2:~$ nice -n+13 sleep 2
tux@deb8-2:~$ nice -n13 sleep 2
tux@deb8-2:~$ nice -n 13 sleep 2
tux@deb8-2:~$ nice -n +13 sleep 2
tux@deb8-2:~$

```

Das Folgende ergibt allerdings NEGATIVE Werte:

```

tux@deb8-2:~$ nice -n-13 sleep 2
nice: die Priorität kann nicht gesetzt werden: Keine Berechtigung
tux@deb8-2:~$
tux@deb8-2:~$ nice --13 sleep 2
nice: die Priorität kann nicht gesetzt werden: Keine Berechtigung
tux@deb8-2:~$

```

Bei *renice* gibt es nicht ganz so viele Möglichkeiten, die Kürzel mit Minus/Plus (-+) oder Minus/Minus (--), funktionieren nicht: wir brauchen also die Option *-n*, außerdem muss ein Leerzeichen zwischen *-n* und dem Wert stehen.

```

tux@deb8-2:~$ renice 19 7464
7464 (Prozesskennung) alte Priorität 0, neue Priorität 19
tux@deb8-2:~$
tux@deb8-2:~$ renice -19 7464
renice: Priorität für 7464 konnte nicht gesetzt werden (Prozesskennung): Keine Berechtigung
tux@deb8-2:~$
tux@deb8-2:~$ renice -n +19 7546
7546 (Prozesskennung) alte Priorität 0, neue Priorität 19
tux@deb8-2:~$

```

18 Topic 103.7 - Search text files using regular expressions

Siehe Seite 143 ff

Es handelt sich hierbei um eine in vielen Kommandos nutzbare Filtersprache (u.a. in Perl).

Welche Tools können damit originär umgehen?

- **grep** (kein logisch ODER)
- **egrep** (oder `grep -E`, = extended grep, beherrscht u.a. logisch ODER)
- **sed** (stream editor)
- **vim**

ACHTUNG: Das Tool *fgrep* (oder `grep -F`) kann NICHT mit regulären Ausdrücken (das "f" bedeutet *fixed* Strings), z.B. wenn man vermeiden möchte, dass der Punkt als "ein beliebiges Zeichen" interpretiert wird:

Beispiel: Der Punkt ist hier ein normales Zeichen:

```
root@deb8-2:~# find /boot/ | fgrep .cfg
/boot/grub/grub.cfg
root@deb8-2:~#
```

Gegenprobe, ob *fgrep* wirklich nichts mit diesen regulären Ausdrücken anfangen kann:

```
### grep kann es:
root@deb8-2:~# grep -v ^# /etc/hosts
127.0.0.1      localhost
127.0.1.1      deb8-2.myDomain.site      deb8-2

::1          localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
10.22.15.119 c7
root@deb8-2:~#

### fgrep kann den Kommentar nicht ausblenden
root@deb8-2:~# fgrep -v ^# /etc/hosts
127.0.0.1      localhost
127.0.1.1      deb8-2.myDomain.site      deb8-2

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
10.22.15.119 c7
root@deb8-2:~#
```

Wollen wir alternativ z.B. Car oder Auto in einer Datei finden, muss für diese logisch-ODER-Verknüpfung das *Extended grep* herangezogen werden:

```
root@deb8-2:~# cat > textdatei.txt <<EOF
Ein Car ist ein Mobil,
das wir auch Auto nennen.
Fahrradfahren ist freilich umweltfreundlicher.
EOF
root@deb8-2:~# grep -E -i 'car|auto' textdatei2.txt
Ein Car ist ein Mobil,
das wir auch Auto nennen.
root@deb8-2:~#
```

Gibt es auch logisch-UND? Ja, hierfür brauchen wir zwei *grep*-Zeilen via Pipe: Zuerst filtern wir z.B. alle Gerätedateien aus der */etc/fstab* heraus, danach folgt ein weiterer Filter, der nur ext-Dateisysteme berücksichtigt:

```
root@deb8-2:~# grep '^/dev' /etc/fstab | fgrep ext
/dev/store01/www      /srv/www      ext3          defaults      0            2
root@deb8-2:~#
```

Aufgabenstellung:

- Schneiden Sie alle Kommentar- und Leerzeilen aus der Datei */etc/hosts* weg, so dass nur der wichtige Rest stehenbleibt:

```
root@deb8-2:~# grep -v '^#' /etc/hosts | grep -v '^$'
127.0.0.1          localhost
127.0.1.1          deb8-2.myDomain.site      deb8-2
::1               localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.22.15.119 c7
root@deb8-2:~#
root@deb8-2:~#
```

Die Option *-v* kann entfallen, wenn die Negation mit einem Metazeichen durchgeführt wird:

```
=> grep ^[^\#] /etc/hosts | grep -v ^$
```

- Ein bestimmtes Wort finden: `grep -i '\<car\>' textdatei.txt`

Weitere reguläre Ausdrücke sind (Auszug von Seite 144 f):

- Ein einzelner Punkt: (.) - ersetzt EIN beliebiges Zeichen.
- Ein Stern (*) - Wiederholungsfaktor: Das davor stehende Zeichen beliebig oft

Somit lässt sich der gewöhnliche Dateijoker (*) auf diese Art als regulärer Ausdruck realisieren: (.*)

Wichtige Optionen:

- **-i** - Ignoriere Groß- und Kleinschreibung
- **-v** - inVertiere die Ergebnisse (negieren)
- **-r** - Rekursiv arbeiten (Verzeichnisse samt Inhalt durchsuchen)
- **-E** - Extended Arbeitsweise aktivieren (= *egrep*)
- **-F** - Fixed-Strings-Arbeitsweise aktivieren (= *fgrep*)
- **-c** - Counting: Zählt nur die Übereinstimmungen (gibt nur eine Zahl aus)
- **-n** - Number of Line: Die betreffende Zeilennummer mit ausgeben
- **-l** - Liste nur die Dateinamen auf, nicht den gefundenen Inhalt
- **-H** - Liste die Dateinamen und den gefundenen Inhalt auf, ideal für eine kleine Suchmaschine:

```
find /etc -type f -exec grep -iH --color linus {} \;
```

18.1 Der streaming Editor sed

Sed kann tatsächlich Dateien editieren, obwohl er sie nicht direkt wie *vim* bearbeiten kann. Sein Name bedeutet: "Stream Editor":

- Typisches Verhalten: Arbeiten im Datenstrom (Pipes)
- Direkt eine Datei editieren: Option *-i* bzw. *--in-place*

Seine Stärke liegt im Suchen und Ersetzen im Datenstrom.

Allgemeine Syntax: `sed 'BEFEHL' datei`

Allgemeine Syntax für das Suchen und Ersetzen: `sed 's/ALT/NEU/[g]' datei`

Dabei bitte beachten:

- Das kleine *s* steht für "substitute", damit wird der Vorgang eingeleitet
- Die Zeichenkette *ALT* soll on the fly durch *NEU* ersetzt werden
- Das optionale kleine *g* steht für "global auf der Zeile agieren", was bedeutet, dass ein Mehrfachvorkommen auf der Zeile beachtet wird

Beispiele:

- `echo "Ich heiÙe Bob, mein Bobschlitten ist Bobbig." | sed 's/Bob/bob/'`
⇒ ERGEBNIS: *Ich heiÙe bob, mein Bobschlitten ist Bobbig.*
- `echo "Ich heiÙe Bob, mein Bobschlitten ist Bobbig." | sed 's/Bob/bob/g'`
⇒ ERGEBNIS: *Ich heiÙe bob, mein bobschlitten ist bobbig.*
- `echo BOB > bob.txt ; sed -i 's/BOB/bob/g' bob.txt`
- Ein sehr experimentelles Beispiel:

```
tux@deb8-2:~$ echo ALT, ALT, ALT | sed 's/ALT/NEU/'
NEU, ALT, ALT
tux@deb8-2:~$
tux@deb8-2:~$ echo ALT, ALT, ALT | sed 's/ALT/NEU/g'
NEU, NEU, NEU
tux@deb8-2:~$
```

19 Topic 103.8 - Perform basic file editing operations using vi

Siehe Seite 152 ff

Es gibt drei Modi:

- COMMAND (ESC)
- INSERT (i, o, a, A)
- EXECUTE (:wq / :w / ZZ / :q! / :wq! / :e datei)

Navigation:

- Ersatz der Cursortasten: h, j, k, l
- Einen Bildschirmausschnitt nach unten (forward) blättern: STRG + f

- Einen Bildschirmausschnitt nach unten (backward) blättern: STRG + b

Ganze Zeilen bearbeiten

- **dd** - (delete) - eine ganze Zeile in den Löschpuffer (= Ausschneiden)
- **yy** - (yank) - eine ganze Zeile in den Kopierpuffer legen
- **p** - (paste) - Pufferinhalt eine Zeile darunter einfügen
- **HINWEIS:** Dabei können den Befehlen ein Zahl für den Wiederholungsfaktor vorangestellt werden

Siehe auch auf <http://pemann.de/doc/lpi1/kleine-kommandoreferenz.pdf>

20 Topic 104.1 - Create partitions and filesystems

20.1 Ein kleiner Ausflug in die Dateitypen

Die sieben Linux-Dateitypen anhand von Beispielen kurz erklärt:

```
tux@deb8-2:~$ ### Reguläre Files
tux@deb8-2:~$ ls -l /etc/passwd /usr/bin/passwd
-rw-r--r-- 1 root root 2966 Feb  2 09:02 /etc/passwd
-rwsr-xr-x 1 root root 54192 Nov 18 09:11 /usr/bin/passwd

tux@deb8-2:~$ ### Directories
tux@deb8-2:~$ ls -ld /etc /tmp
drwxr-xr-x 1 root root 4196 Feb  2 09:02 /etc
drwxrwxrwt 1 root root 462 Feb  2 11:36 /tmp

tux@deb8-2:~$ ### Symbolische Links
tux@deb8-2:~$ ls -l /var/run /sbin/telinit
lrwxrwxrwx 1 root root 14 Nov 21 20:00 /sbin/telinit -> /bin/systemctl
lrwxrwxrwx 1 root root  4 Jan  7 12:24 /var/run -> /run

tux@deb8-2:~$ ### Gerätedateien (Character / Block)
tux@deb8-2:~$ ls -l /dev/ttyS0 /dev/sda
brw-rw---- 1 root disk  8,  0 Feb  2 07:48 /dev/sda
crw-rw---- 1 root dialout 4, 64 Feb  2 07:48 /dev/ttyS0

tux@deb8-2:~$ ### Prozesse via benannter Pipe und Unix-Domain-Socket
tux@deb8-2:~$ ls -l /dev/xconsole /tmp/.X11-unix/X0
prw-r----- 1 root adm  0 Feb  2 11:20 /dev/xconsole
srwxrwxrwx 1 root root 0 Feb  2 07:48 /tmp/.X11-unix/X0
```

Situation: Auf / wird der Plattenplatz knapp, Grund ist das installierte Dokumentenmanagementsystem *Agorum* (/opt beinhaltet eine große DB)

AUFGABE: Gastsystem bitte herunterfahren, Festplatte einbauen... (8 GB)

Kontrollieren wir, ob die neue Festplatte da ist:

- fdisk -l
- dmesg
- blkid
- lsblk
- parted (Automatische Auswahl von /dev/sda, Partitionen auflisten: print all, quit)
- gdisk (zur Kontrolle: gdisk -l /dev/sdb)

20.2 Partitionierung mit `fdisk`:

Die Vorgehensweise im Überblick:

1. **m** - Ausgeben der integrierten `fdisk`-Befehle
2. **p** - Ausgeben der Partitionstabelle, die zur Zeit noch leer ist
3. **n** - (new) Neue Partition anlegen
4. **e** - Typ festlegen: erweiterte Partition, Größe: vorgeschlagene Start- und Endwerte mit ENTER übernehmen (ganze Platte)
5. **n** - eine weitere neue Partition anlegen
6. **l** - Typ festlegen: logische Partition, Größe: vorgeschlagenen Startwert mit ENTER übernehmen, Endwert: `+1G`
7. **n** - eine weitere neue Partition anlegen
8. **l** - Typ festlegen: logische Partition, Größe: vorgeschlagenen Startwert mit ENTER übernehmen, Endwert: `+2G`
9. **p** - ein Print zur Endkontrolle
10. **w** - (write) Änderungen speichern und Programm beenden

Hier der Vorgang im einzelnen:

```
root@deb8-2:~# fdisk /dev/sdb
Willkommen bei fdisk (util-linux 2.25.2).
Änderungen werden vorerst nur im Speicher vorgenommen, bis Sie sich
entscheiden, sie zu schreiben.
Seien Sie vorsichtig, bevor Sie den Schreibbefehl anwenden.

Gerät enthält keine erkennbare Partitionstabelle.
Created a new DOS disklabel with disk identifier 0x23695331.

Befehl (m für Hilfe): m

Hilfe:

DOS (MBR)
 a  (De)Aktivieren der bootfähig-Markierung
 b  verschachtelte BSD-Festplattenbezeichnung bearbeiten
 c  (De)Aktivieren der DOS-Kompatibilitätsmarkierung

Generisch
 d  Eine Partition löschen
 l  Die bekannten Dateisystemtypen anzeigen
 n  Eine neue Partition anlegen
 p  Die Partitionstabelle ausgeben
 t  Einen Partitionstyp ändern
 v  Die Partitionstabelle überprüfen

Sonstiges
 m  Dieses Menü anzeigen
 u  Die Einheit für die Anzeige/Eingabe ändern
 x  Zusätzliche Funktionen (nur für Experten)

Speichern und Beenden
 w  Die Tabelle auf die Festplatte schreiben und das Programm beenden
 q  Beenden ohne Speichern der Änderungen

Eine neue Bezeichnung erstellen
 g  Eine neue leere GPT-Partitionstabelle erstellen
```

```

G   Eine neue leere SGI (IRIX)-Partitionstabelle erstellen
o   Eine neue leere DOS-Partitionstabelle erstellen
s   Eine neue leere Sun-Partitionstabelle erstellen

```

```

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331

```

```

Befehl (m für Hilfe): n
Partitionstyp
  p   Primär (0 primär, 0 erweitert, 4 frei)
  e   Erweitert (Container für logische Partitionen)
Wählen (Vorgabe p): e
Partitionsnummer (1-4, Vorgabe 1):
Erster Sektor (2048-16777215, Vorgabe 2048):
Letzter Sektor, +Sektoren oder +Größe{K,M,G,T,P} (2048-16777215, Vorgabe 16777215):

```

Eine neue Partition 1 des Typs »Extended« und der Größe 8 GiB wurde erstellt.

```

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331

```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	5	Extended

```

Befehl (m für Hilfe): n
Partitionstyp
  p   Primär (0 primär, 1 erweitert, 3 frei)
  l   Logisch (nummeriert ab 5)
Wählen (Vorgabe p): l

```

```

Logische Partition 5 wird hinzugefügt
Erster Sektor (4096-16777215, Vorgabe 4096):
Letzter Sektor, +Sektoren oder +Größe{K,M,G,T,P} (4096-16777215, Vorgabe 16777215): +1M

```

Eine neue Partition 5 des Typs »Linux« und der Größe 1 MiB wurde erstellt.

```

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331

```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	5	Extended
/dev/sdb5		4096	6143	2048	1M	83	Linux

```

Befehl (m für Hilfe): n
Partitionstyp
  p Primär (0 primär, 1 erweitert, 3 frei)
  l Logisch (nummeriert ab 5)
Wählen (Vorgabe p): l

Logische Partition 6 wird hinzugefügt
Erster Sektor (8192-16777215, Vorgabe 8192):
Letzter Sektor, +Sektoren oder +Größe{K,M,G,T,P} (8192-16777215, Vorgabe 16777215): +2G

Eine neue Partition 6 des Typs »Linux« und der Größe 2 GiB wurde erstellt.

```

```

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331

```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	5	Extended
/dev/sdb5		4096	6143	2048	1M	83	Linux
/dev/sdb6		8192	4202495	4194304	2G	83	Linux

```

Befehl (m für Hilfe): w

```

Die Partitionstabelle wurde verändert.
 ioctl() wird aufgerufen, um die Partitionstabelle neu einzulesen.
 Festplatten werden synchronisiert.

```

root@deb8-2:~#

```

Nun Kontrollieren wir die erfolgreiche Partitionierung:

```

root@deb8-2:~# ls -l /dev/sdb*
brw-rw---- 1 root disk 8, 16 Jan 27 12:58 /dev/sdb
brw-rw---- 1 root disk 8, 17 Jan 27 12:58 /dev/sdb1
brw-rw---- 1 root disk 8, 21 Jan 27 12:58 /dev/sdb5
brw-rw---- 1 root disk 8, 22 Jan 27 12:58 /dev/sdb6
root@deb8-2:~#

```

Es muss hierbei im Hintergrund so etwas wie *partprobe* gelaufen sein, um udev über die Änderungen zu informieren und anzuweisen, neue Gerätedateien zu erzeugen.

Als nächstes können auch schon Dateisysteme aufgebracht werden:

```

root@deb8-2:~# mkfs /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
In /dev/sdb1 wurde eine dos-Partitionstabelle gefunden
Trotzdem fortfahren? (j,n)

root@deb8-2:~# mkfs /dev/sdb5
mke2fs 1.42.12 (29-Aug-2014)
Ein Dateisystems mit 262144 (4k) Blöcken und 65536 Inodes wird erzeugt.
UUID des Dateisystems: 39aead11-c212-4372-8933-fa7ea1070152
Superblock-Sicherungskopien gespeichert in den Blöcken:
    32768, 98304, 163840, 229376

beim Anfordern von Speicher für die Gruppentabellen: erledigt

```

```
Inode-Tabellen werden geschrieben: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt
```

Diese neuen Dateisysteme wollen wir natürlich gleich einbinden:

```
root@deb8-2:~# mount /dev/sdb5 /mnt/
root@deb8-2:~#
root@deb8-2:~# mount | tail -3
tmpfs on /run/user/118 type tmpfs (rw,nosuid,nodev,relatime,size=102444k,mode=700,uid=118, ↵
gid=126)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=102444k,mode=700,uid ↵
=1000,gid=1000)
/dev/sdb5 on /mnt type ext2 (rw,relatime)
root@deb8-2:~#
root@deb8-2:~#
root@deb8-2:~# mke2fs -j /dev/sdb6
mke2fs 1.42.12 (29-Aug-2014)
Ein Dateisystems mit 524288 (4k) Blöcken und 131072 Inodes wird erzeugt.
UUID des Dateisystems: 31c9d432-5648-4441-8067-e441a0f35f94
Superblock-Sicherungskopien gespeichert in den Blöcken:
    32768, 98304, 163840, 229376, 294912

beim Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Das Journal (16384 Blöcke) wird angelegt: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt

root@deb8-2:~#
```

Wir lassen uns den aktuellen Stand der Arbeiten ausgeben:

```
root@deb8-2:~# fdisk -l /dev/sdb | grep ^/dev
/dev/sdb1          2048 16777215 16775168    8G  5 Extended
/dev/sdb5          4096  2101247  2097152    1G  83 Linux
/dev/sdb6          2103296 6297599 4194304    2G  83 Linux
root@deb8-2:~#
```

20.2.1 Swap-Space erweitern

Die zweite Partition /dev/sdb6 soll für swap-Nutzung umgewidmet werden:

- Dateisystemtyp ändern: Von 83 (Linux: ext*, xfs, reiserfs, btrfs, jfs) auf 82 (Swap)
 - `fdisk /dev/sdb` ⇒ Taste "t" drücken, eine "6" eingeben, dann mit "L" alle Typen auflisten
- Vor der Erweiterung den aktuellen Zustand feststellen:
 - `free -h`
 - `swapon -s`
- Partition für swap vorbereiten:

```
root@deb8-2:~# mkswap /dev/sdb6
mkswap: /dev/sdb6: Warnung: Alte ext3-Signatur wird gelöscht.
Auslagerungsbereich Version 1 wird angelegt, Größe = 2097148 KiB
keine Bezeichnung, UUID=71e5903e-41f2-41f2-8a20-70c649a042be
root@deb8-2:~#
```

- Die swap-Partition zuschalten: swapon

```
root@deb8-2:~# swapon -v /dev/sdb6
swapon /dev/sdb6
swapon: /dev/sdb6: Swap-Signatur gefunden: Version 1d, Seitengröße 4, Byte-Reihenfolge ←
gleich
swapon: /dev/sdb6: Seitengröße=4096, Swapgröße=2147483648, Gerätegröße=2147483648
root@deb8-2:~#
```

- Vor der Erweiterung den aktuellen Zustand feststellen: Wie oben free und swapon -s
- Wieder freigeben der swap-Partition:

```
root@deb8-2:~# swapoff -v /dev/sdb6
swapoff /dev/sdb6
root@deb8-2:~#
```

****NICHT VERGESSEN:** Dies ist nur temporär, am Ende muss die Datei */etc/fstab* bearbeitet werden!

20.2.2 Dateisystem erzeugen und benutzen

ZIEL: Auf einer weiteren Partition sollen 4 GB für spätere Nutzung für /opt bereitgestellt werden. Als Dateisystem zuerst bitte ext2 verwenden, später soll ein Journal hinzugefügt werden (ext3).

Die Schritte im Überblick:

- Die zweite Festplatte partitionieren (ID = 83)
- Dateisystem aufbringen
- Temporär nach /mnt mounten (selbständig Erfolg prüfen mit zwei verschiedenen Tools)

Ausführung dieser 3 Schritte:

```
root@deb8-2:~# fdisk /dev/sdb

Willkommen bei fdisk (util-linux 2.25.2).
Änderungen werden vorerst nur im Speicher vorgenommen, bis Sie sich
entscheiden, sie zu schreiben.
Seien Sie vorsichtig, bevor Sie den Schreibbefehl anwenden.

Befehl (m für Hilfe): n
Partitionstyp
  p  Primär (0 primär, 1 erweitert, 3 frei)
  l  Logisch (nummeriert ab 5)
Wählen (Vorgabe p): l

Logische Partition 7 wird hinzugefügt
Erster Sektor (6299648-16777215, Vorgabe 6299648):
Letzter Sektor, +Sektoren oder +Größe{K,M,G,T,P} (6299648-16777215, Vorgabe 16777215): +4G

Eine neue Partition 7 des Typs »Linux« und der Größe 4 GiB wurde erstellt.

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
```

```
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	5	Extended
/dev/sdb5		4096	2101247	2097152	1G	83	Linux
/dev/sdb6		2103296	6297599	4194304	2G	82	Linux swap / Solaris
/dev/sdb7		6299648	14688255	8388608	4G	83	Linux

```
Befehl (m für Hilfe): w
Die Partitionstabelle wurde verändert.
ioctl() wird aufgerufen, um die Partitionstabelle neu einzulesen.
Das erneute Einlesen der Partitionstabelle ist fehlgeschlagen.: Das Gerät oder die
Ressource ist belegt
```

```
Der Kernel benutzt noch die alte Tabelle. Die neue Tabelle wird beim nächsten Neustart
verwendet werden oder
nachdem Sie partprobe(8) oder kpartx(8) ausgeführt haben.
```

```
root@deb8-2:~#
```

```
root@deb8-2:~# ls -l /dev/sdb*
brw-rw---- 1 root disk 8, 16 Jan 28 10:18 /dev/sdb
brw-rw---- 1 root disk 8, 17 Jan 28 10:18 /dev/sdb1
brw-rw---- 1 root disk 8, 21 Jan 28 10:18 /dev/sdb5
brw-rw---- 1 root disk 8, 22 Jan 28 10:18 /dev/sdb6
```

```
root@deb8-2:~#
```

```
root@deb8-2:~# partprobe
```

```
Warning: Unable to open /dev/sr0 read-write (Das Dateisystem ist nur lesbar). /dev/sr0 has
been opened read-only.
```

```
Warning: Unable to open /dev/sr0 read-write (Das Dateisystem ist nur lesbar). /dev/sr0 has
been opened read-only.
```

```
root@deb8-2:~#
```

```
root@deb8-2:~#
```

```
root@deb8-2:~# ls -l /dev/sdb*
brw-rw---- 1 root disk 8, 16 Jan 28 10:19 /dev/sdb
brw-rw---- 1 root disk 8, 17 Jan 28 10:19 /dev/sdb1
brw-rw---- 1 root disk 8, 21 Jan 28 10:19 /dev/sdb5
brw-rw---- 1 root disk 8, 22 Jan 28 10:19 /dev/sdb6
brw-rw---- 1 root disk 8, 23 Jan 28 10:19 /dev/sdb7
```

```
<===== Dank 'partprobe' wurde
```

```
das Gerät erzeugt
```

```
root@deb8-2:~#
```

```
root@deb8-2:~# ### Jetzt erzeugen wir ext2 (ohne Journal) auf dreierlei Art:
```

```
root@deb8-2:~# mkfs /dev/sdb7
```

```
mke2fs 1.42.12 (29-Aug-2014)
```

```
Ein Dateisystems mit 1048576 (4k) Blöcken und 262144 Inodes wird erzeugt.
```

```
UUID des Dateisystems: 68fdb24b-e5f9-4edf-bcd2-8a0c4ef34edc
```

```
Superblock-Sicherungskopien gespeichert in den Blöcken:
```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736
```

```
beim Anfordern von Speicher für die Gruppentabellen: erledigt
```

```
Inode-Tabellen werden geschrieben: erledigt
```

```
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt
```

```
root@deb8-2:~#
```

```
root@deb8-2:~# mkfs -t ext2 /dev/sdb7
```

```
mke2fs 1.42.12 (29-Aug-2014)
```

```
/dev/sdb7 hat ein ext2-Dateisystem
```

```

    auf Thu Jan 28 10:21:23 2016
    erzeugtTrotzdem fortfahren? (j,n) j
    Ein Dateisystems mit 1048576 (4k) Blöcken und 262144 Inodes wird erzeugt.
    UUID des Dateisystems: 71254867-c4b3-403c-aebc-1340e15cbd23
    Superblock-Sicherungskopien gespeichert in den Blöcken:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

beim Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt

root@deb8-2:~#
root@deb8-2:~# mke2fs /dev/sdb7
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb7 hat ein ext2-Dateisystem
    auf Thu Jan 28 10:21:45 2016
    erzeugtTrotzdem fortfahren? (j,n) j
    Ein Dateisystems mit 1048576 (4k) Blöcken und 262144 Inodes wird erzeugt.
    UUID des Dateisystems: a13f6800-8a04-4b60-95b7-0a32f0b58ad7
    Superblock-Sicherungskopien gespeichert in den Blöcken:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

beim Anfordern von Speicher für die Gruppentabellen: erledigt
Inode-Tabellen werden geschrieben: erledigt
Die Superblöcke und die Informationen über die Dateisystemnutzung werden
geschrieben: erledigt

root@deb8-2:~#
root@deb8-2:~# mount /dev/sdb7 /mnt/
root@deb8-2:~#
root@deb8-2:~# mount | tail -1
/dev/sdb7 on /mnt type ext2 (rw,relatime)
root@deb8-2:~#
root@deb8-2:~# df -h /mnt
Dateisystem    Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sdb7      4,0G    8,0M  3,8G    1% /mnt
root@deb8-2:~#
root@deb8-2:~# ls -l /mnt/
insgesamt 16
drwx----- 2 root root 16384 Jan 28 10:22 lost+found
root@deb8-2:~#

```

20.2.3 Neue Partition für /opt verwenden

Problem: /opt lässt sich nicht umbenennen, um die neue Partition, die jetzt unter /mnt gemountet ist, anstelle dessen für /opt zu verwenden! Grund: Sie wird gerade benutzt, siehe das Listing offener Files:

- `lsof /| grep opt`
- `fuser -mv /opt`
- `ls -la /opt`

⇒ Am besten sich eine Auszeit nehmen und in den sicheren Runlevel 1 wechseln, hier die Möglichkeiten bei Debian 8:

- `init 1`
- `telinit 1`

- `systemctl isolate runlevel1.target`
- **reboot**, dann den GRUB-Bootprompt editieren und dem Kernel eine "1" mitgeben (`linux /boot/vmlinuz-VERSION ...root=.....1`)

```
# In den abgesicherten Modus fahren
systemctl isolate runlevel1.target

# Das Originalverzeichnis umbenenn
mv /opt /opt.old
mkdir /opt

# Falls das Gerät noch von vorherigen Tests gemountet ist
umount /mnt

# Für robuste Gerätebezeichnungen Universally Unique Identifier benutzen
blkid /dev/sdb7 >> /etc/fstab

# File System Table editieren
vi /etc/fstab
```

Hier bitte Anpassen:

- Einen Hash-Tag vor die lange Zeile setzen
- Nach `/dev/sdb7` einen Zeilenumbruch einfügen
- Die Double Ticks (Gänsefüßchen) rings um die UUID entfernen
- Die Zeile vervollständigen:

```
# /dev/sdb7
UUID=HEXWERT-OHNE-TICKMARKS /opt ext2 defaults 0 2
```

Die nächsten Schritte sind, das Ganze noch einmal zu kontrollieren, das Gerät zu mounten und die Daten auf dem neuen Volume bereitzustellen:

```
# Kontrolle
grep /opt /etc/fstab

# Jetzt mounten (alle Einträge der fstab, die nicht die Option 'noauto' aufweisen)
mount -a
df -h /opt
ls -la /opt

# Die originalen Daten herüberkopieren
cp -a /opt.old/* /opt

# In den grafischen Runlevel fahren
systemctl isolate runlevel5.target
```

Nach dem Wiedereinloggen auf der grafischen Oberfläche, den alten Ordner bitte nach ausgiebigen Tests noch mit `\rm -r /opt.old` löschen (der Backslash deaktiviert für den Moment der Ausführung einen möglicherweise vorhanden alias).

Damit ist die Aufgabenstellung der Datenmigration abgeschlossen.

20.2.4 Weitere Aufgaben

- Betreffend /dev/sdb7: An das vorhandene, in Benutzung befindliche ext2-Dateisystem ein Journal anfügen
- Eine weitere Partition anlegen, die mit xfs zu formatieren ist
- Auf die erste Partition der zweiten Festplatte das "Virtual File Allocation Table" Dateisystem aufbringen

Hier die drei Schritte relativ unkommentiert via Shell-Mitschnitt:

```
root@deb8-2:~# fdisk /dev/sdb

Willkommen bei fdisk (util-linux 2.25.2).
Änderungen werden vorerst nur im Speicher vorgenommen, bis Sie sich
entscheiden, sie zu schreiben.
Seien Sie vorsichtig, bevor Sie den Schreibbefehl anwenden.

Befehl (m für Hilfe): n
Partitionstyp
  p   Primär (0 primär, 1 erweitert, 3 frei)
  l   Logisch (nummeriert ab 5)
Wählen (Vorgabe p): l

Logische Partition 8 wird hinzugefügt
Erster Sektor (14690304-16777215, Vorgabe 14690304):
Letzter Sektor, +Sektoren oder +Größe{K,M,G,T,P} (14690304-16777215, Vorgabe 16777215):

Eine neue Partition 8 des Typs »Linux« und der Größe 1019 MiB wurde erstellt.

Befehl (m für Hilfe): p
Festplatte /dev/sdb: 8 GiB, 8589934592 Bytes, 16777216 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0x23695331

Device      Boot      Start          End      Sectors  Size Id Type
/dev/sdb1   boot        2048 16777215 16775168    8G  5 Extended
/dev/sdb5                   4096 2101247 2097152    1G 83 Linux
/dev/sdb6                   2103296 6297599 4194304    2G 82 Linux swap / Solaris
/dev/sdb7                   6299648 14688255 8388608    4G 83 Linux
/dev/sdb8                   14690304 16777215 2086912 1019M 83 Linux

Befehl (m für Hilfe): w
Die Partitionstabelle wurde verändert.
ioctl() wird aufgerufen, um die Partitionstabelle neu einzulesen.
Das erneute Einlesen der Partitionstabelle ist fehlgeschlagen.: Das Gerät oder die ↵
Ressource ist belegt

Der Kernel benutzt noch die alte Tabelle. Die neue Tabelle wird beim nächsten Neustart ↵
verwendet werden oder
nachdem Sie partprobe(8) oder kpartx(8) ausgeführt haben.

root@deb8-2:~#

root@deb8-2:~# ls -l /dev/sdb8
ls: Zugriff auf /dev/sdb8 nicht möglich: Datei oder Verzeichnis nicht gefunden
```

```

root@deb8-2:~#
root@deb8-2:~# partprobe
Warning: Unable to open /dev/sr0 read-write (Das Dateisystem ist nur lesbar). /dev/sr0 has ←
been opened read-only.
Warning: Unable to open /dev/sr0 read-write (Das Dateisystem ist nur lesbar). /dev/sr0 has ←
been opened read-only.
root@deb8-2:~#
root@deb8-2:~# ls -l /dev/sdb8
brw-rw---- 1 root disk 8, 24 Jan 28 11:50 /dev/sdb8
root@deb8-2:~#
root@deb8-2:~#

### Wie man sieht fehlen einfach nur die XFS-Werkzeuge (Der Kernel muss es freilich ←
unterstützen!)
root@deb8-2:~# mkfs.
mkfs.bfs      mkfs.cramfs   mkfs.ext3     mkfs.ext4dev  mkfs.minix    mkfs.ntfs
mkfs.btrfs    mkfs.ext2     mkfs.ext4     mkfs.fat      mkfs.msdos    mkfs.vfat
root@deb8-2:~# mkfs.

### Also die Programme einfach installieren:
apt-get install xfsprogs

root@deb8-2:~# mkfs.xfs /dev/sdb8
meta-data=/dev/sdb8          isize=256      agcount=4, agsize=65216 blks
=                               sectsz=512    attr=2, projid32bit=1
=                               crc=0         finobt=0
data      =                  bsize=4096   blocks=260864, imaxpct=25
=                               sunit=0      swidth=0 blks
naming    =version 2        bsize=4096   ascii-ci=0 ftype=0
log       =Internes Protokoll bsize=4096   blocks=853, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =keine           extsz=4096   blocks=0, rtextents=0
root@deb8-2:~#

root@deb8-2:~# mkfs.vfat -F32 /dev/sdb5
mkfs.fat 3.0.27 (2014-11-12)
root@deb8-2:~#

```

Benutzung der xfs-Partition als HOME für neue User (weil es natives Quota beherrscht)

- Mountpunkt: /Users
- fstab-Eintrag erzeugen und Gerät mounten
- User mit \$HOME auf dieser Partition erzeugen
 - `useradd -m -d /Users/susi -s /bin/bash susi && passwd susi`

Hier ist die Lösung:

```

root@deb8-2:~# mkdir /Users
root@deb8-2:~#
root@deb8-2:~# blkid /dev/sdb8 >> /etc/fstab
root@deb8-2:~#
root@deb8-2:~# tilde /etc/fstab
root@deb8-2:~#
root@deb8-2:~# tail -3 /etc/fstab

#/dev/sdb8:
UUID=2cc89fa8-1601-47eb-b356-ed0b804dad51 /Users xfs defaults 0 2

```

```

root@deb8-2:~#
root@deb8-2:~# mount -a
root@deb8-2:~#
root@deb8-2:~# mount | tail -1
/dev/sdb8 on /Users type xfs (rw,relatime,attr2,inode64,noquota)
root@deb8-2:~#

root@deb8-2:~# useradd -m -d /Users/susi -s /bin/bash scgsusi && passwd susi
Geben Sie ein neues UNIX-Passwort ein:
Geben Sie das neue UNIX-Passwort erneut ein:
passwd: Passwort erfolgreich geändert
root@deb8-2:~#
root@deb8-2:~# ls -la /Users/susi/
insgesamt 12
drwxr-xr-x 5 susi susi   92 Jan 28 12:19 .
drwxr-xr-x 3 root root   17 Jan 28 12:19 ..
-rw-r--r-- 1 susi susi  220 Nov 13  2014 .bash_logout
-rw-r--r-- 1 susi susi 3515 Nov 13  2014 .bashrc
drwxr-xr-x 2 susi susi    6 Jan 22 10:23 Desktop
drwxr-xr-x 2 susi susi    6 Jan 22 10:23 Musik
-rw-r--r-- 1 susi susi  675 Nov 13  2014 .profile
drwxr-xr-x 2 susi susi    6 Jan 22 10:23 Video
root@deb8-2:~#

### Testweise mal schnell root werden
root@deb8-2:~# su - susi
susi@deb8-2:~$
susi@deb8-2:~$ pwd
/Users/susi
susi@deb8-2:~$
susi@deb8-2:~$ id
uid=1004(susi) gid=1004(susi) Gruppen=1004(susi)
susi@deb8-2:~$
susi@deb8-2:~$ groups
susi
susi@deb8-2:~$ exit
Abgemeldet
root@deb8-2:~#

```

Benutzung der vfat-Partition als Austausch-Ordner für Windows-User (weil wir nicht den Tuxera-Treiber für rw-NTFS einsetzen wollen und der Kernel selbst es nicht unterstützt).

- Mountpunkt: /media/austausch
- fstab-Eintrag erzeugen und Gerät nicht automatisch mounten lassen: Der User möge es selber tun.) ⇒ defaults, noauto, user
- Weiteren Handlungsbedarf gibt es beim Zeichensatzmapping für UTF-8 - ISO8559-1 (siehe <http://www.heise.de/ct/hotline/Linux-vermatscht-Umlaute-326142.html>)

Und hier ist die Lösung:

```

root@deb8-2:~# mkdir /media/austausch
root@deb8-2:~#
root@deb8-2:~# blkid /dev/sdb5 >> /etc/fstab
root@deb8-2:~#
root@deb8-2:~# tilde /etc/fstab
root@deb8-2:~#
root@deb8-2:~# tail -2 /etc/fstab
#/dev/sdb5:
UUID=490D-7B8A /media/austausch vfat defaults,noauto,user,icharset=utf8 0 0
root@deb8-2:~#

```

```
root@deb8-2:~# ### Jetzt hilft uns 'mount -a' nicht mehr (wegen "noauto"), deshalb:
root@deb8-2:~#
root@deb8-2:~# su - susi
susi@deb8-2:~$
susi@deb8-2:~$ df -h /media/austausch/
Dateisystem Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda5 28G 6,5G 20G 25% /
susi@deb8-2:~$
susi@deb8-2:~$ mount /media/austausch/
susi@deb8-2:~$
susi@deb8-2:~$ df -h /media/austausch/
Dateisystem Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sdb5 1022M 4,0K 1022M 1% /media/austausch
susi@deb8-2:~$
susi@deb8-2:~$
susi@deb8-2:~$
susi@deb8-2:~$ ##### Verwirrend, dass 'disk free' uns den Superblock via Angabe irgendeines
Unterverzeichnis ansprechen lässt:
susi@deb8-2:~$ df -h /usr/share/doc/bash
Dateisystem Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda5 28G 6,5G 20G 25% /
susi@deb8-2:~$
susi@deb8-2:~$
susi@deb8-2:~$ exit
Abgemeldet
root@deb8-2:~#
```

Abschließende Aufgaben:

- Installation von tree-Size Alternativen für Linux (GUI): baobab, k4dirstat (gerne auch *dolphin*)
<http://www.makeuseof.com/tag/how-to-analyze-your-disk-usage-pattern-in-linux/>

21 Topic 104.2 - Maintain the integrity of filesystems

Siehe Seite 170 ff

21.1 Wichtige Tools für den Dateisystem-Check

- **fsck** - file system check (= Frontend für verschiedene Dateisysteme, die einzelnen Tool sind z.B.
 - e2fsck
 - fsck.ext3
 - fsck.ext4
 - fsck.xfs
- **shutdown** - Das Kommando ist in der Lage, mit der Option *-f* einen evl. anstehenden Check zu unterbinden, mit *-F* wird dagegen beim nächsten Startvorgang eine Prüfung erzwungen (Force).

Wichtige Optionen:

- **-f** - force (erzwingen), wird erforderlich, wenn die per Default versuchte, automatische Reparatur fehlgeschlagen ist, z.B. könnte das so aussehen:

```

root@deb8-2:~# fsck -f -n /dev/sdb7
fsck von util-linux 2.25.2
e2fsck 1.42.12 (29-Aug-2014)
Warnung! /dev/sdb7 ist eingehängt.
Durchgang 1: Inodes, Blöcke und Größen werden geprüft
Durchgang 2: Verzeichnisstruktur wird geprüft
Durchgang 3: Verzeichnisverknüpfungen werden geprüft
Durchgang 4: Referenzzähler werden überprüft
Durchgang 5: Zusammengefasste Gruppeninformation wird geprüft
Die Anzahl freier Blöcke ist falsch (1030066, gezählt=989364).
Reparieren? nein

Die Anzahl freier Inodes ist falsch (262133, gezählt=259988).
Reparieren? nein

/dev/sdb7: 11/262144 Dateien (1254.5% nicht zusammenhängend), 18510/1048576 Blöcke
root@deb8-2:~#

```

- **-n** - no repair, dry run (für Tests gut, keine Garantie dass die Aussagen stimmen!), besser also: Dateisystem aushängen
- **-A** - alle Dateisystem, die in der `/etc/fstab` beschrieben sind prüfen
- **-y** - Alle Fragen nach "fixed?" automatisch mit `yes` beantworten

21.2 Tools für das Tuning und für Recovery

- **tune2fs** - Tuningmöglichkeiten (S. 172 f)
 - Journal einem ext2-Dateisystem beifügen, Option: `-j`
 - Anzahl der max. Mountvorgänge: `-c (count)`
 - Intervall zwischen den Prüfungen [Tage]: `-i` (ohne Zeiteingabe wird die regelmäßige Überprüfung ausgeschaltet)
 - Den Zeitpunkt für den zuletzt stattgefunden Check manuell festlegen: `-T <TIME>`
- **dumpe2fs** - einen Dateisystemdump erstellen, mit `-h` nur den Header ausgeben: `dumpe2fs -h /dev/sdb7 | grep -i check`
- **debugfs** - ein interaktives Debugging und Recoverywerkzeug (Tipp: anstelle `extundelete` verwenden)
- **xfs_growfs** - ein XFS-Dateisystem vergrößern
- **xfs_quota** - Diskquotas für XFS einrichten
- **df -i** - Anzeigen der freien Inodes (Dateiknotennummer), ACHTUNG: dummerweise wird eine feste Anzahl bei den ext-Dateisystemen reserviert:

```

root@deb8-2:/opt# df -i .
Dateisystem  Inodes IBenutzt  IFrei IUse% Eingehängt auf
/dev/sdb7    262144    2156 259988    1% /opt
root@deb8-2:/opt#
root@deb8-2:/opt# df -h .
Dateisystem  Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sdb7    4,0G   167M  3,6G    5% /opt
root@deb8-2:/opt#
root@deb8-2:/opt# while true; do echo > datei-'date +%s'.txt; done
root@deb8-2:/opt#

```

Parallel kann man in einem anderen Terminal Monitoring betreiben: `watch -n1 -d df -i /opt/`

22 Topic 104.3 - Control mounting and unmounting of filesystems

Wichtiges Grundwissen: Aufbau der Datei /etc/fstab:

```
#<Ressource>      <Mountpunkt>      <Dateisystemtyp>      <Optionen>      <Sichern mit dump?>  ↔
    <Prüfen mit fsck?>
# Wurzel:
UUID=9f9e3961-38c4-4dc8-b809-88b063ea905a /          btrfs  defaults          0          1

# Optionale Daten:
UUID=a13f6800-8a04-4b60-95b7-0a32f0b58ad7 /opt  ext2  defaults 0 2

# Network File System - Freigabe:
10.0.0.1:/srv/backup      /media/backup      nfs      soft,bg      0  0

# Windows-Freigabe (SMB, samba-Server):
//10.0.0.1/backup      /media/samba      cifs      guest,noexec,nosuid,umask=777      0  0
```

Die Optionen lassen sich natürlich auch beim manuellen Mounten übergeben:

```
=> mount -o guest //10.0.0.1/backup /media/samba
=> mount -o remount,rw /          # Siehe "root-Passwort vergessen": GRUB/Kernel- ↔
    Parameter: `init=/bin/sh` ...
```

Aufgabe: Studieren Sie die mount-Optionen, siehe Seite 182 f:

- auto
- noauto
- usrquota
- grpquota
- suid (Dateien mit erhöhten Rechten akzeptieren)
- nosuid (Dateien mit erhöhten Rechten ignorieren)
- exec
- noexec
- ro
- rw
- user (Ein beliebiger Nutzer darf diese Ressource mounten, wobei nur derjenige, welcher sie gemountet hatte, auch wieder aushängen kann.)
- nouser
- users (Ein beliebiger Nutzer darf diese Ressource mounten; hier dürfen auch x-beliebige sie wieder aushängen)
- defaults (= rw,suid,dev, exec,auto,nouser,async)
- sync (Schreiboperationen sofort synchronisieren, keinen Cache nutzen)
- async (asynchron arbeiten, bei gelegener Zeit Cache auf Platte schreiben)

23 Topic 104.4 - Manage disk quotas

ZIEL: Speicherplatzkontingentierung für viele Nutzer auf einer /home-Partition

Siehe Seite 184 ff

Praktische Vorgehensweise:

23.1 Software installieren

Unter Debian ist dies schnell erledigt:

```
apt-get install quota
```

23.2 Datei /etc/fstab anpassen

Die zu reglementierende Partition muss in Feld 4 als weitere Option entweder *usrquota*, *grpquota* oder beides erhalten.

Danach muss die Partition neu eingehängt werden, um die Einstellung aktiv zu setzen. Hier lässt sich oft sehr gut mit der mount-Option *remount* arbeiten:

```
mount -o remount /opt
```

Im Beispiel hier mal die betreffenden Zeilen:

```
root@deb8-2:~# grep quota /etc/fstab
UUID=a13f6800-8a04-4b60-95b7-0a32f0b58ad7 /opt ext2 defaults,grpquota 0 2
root@deb8-2:~#
root@deb8-2:~#
root@deb8-2:~#
root@deb8-2:~# mount | grep quota
/dev/sda8 on /home type xfs (rw,relatime,attr2,inode64,noquota)
root@deb8-2:~#
root@deb8-2:~# mount -o remount /opt
root@deb8-2:~#
root@deb8-2:~# mount | grep quota
/dev/sda8 on /home type xfs (rw,relatime,attr2,inode64,noquota)
/dev/sdb7 on /opt type ext2 (rw,relatime,grpquota) <<<<<----- Jetzt taucht hier ←
'grpquota' auf!
root@deb8-2:~#
```

23.3 Prüfen und einrichten der Quotierung

Dies geschieht mit

```
quotacheck -avum [-g] # => Datei /aquota.user [/aquota.group] wird dabei erstellt
```

In manchen Fällen ist es erforderlich, den Schalter *-m (--no-remount)* mitzunehmen, da ansonsten die Aktion fehlschlägt. Beim Check versucht nämlich dieses Kommando, die Partition read-only zu mounten.

```
root@deb8-2:~# quotacheck -agv
quotacheck: Prüfe /dev/sdb7 [/opt] fertig
quotacheck: Old user file name could not been determined. Usage will not be subtracted.
quotacheck: Cannot stat old group quota file /opt/aquota.group: Datei oder Verzeichnis ←
nicht gefunden. Usage
will not be subtracted.
quotacheck: 546 Verzeichnisse und 1602 Dateien geprüft
quotacheck: Alte Datei nicht gefunden.
```



```
root@deb8-2:~#  
root@deb8-2:~# ls -l /opt/  
insgesamt 32  
-rw----- 1 root root 6144 Jan 29 12:33 aquota.group  
drwxr-xr-x 7 root root 4096 Jan 18 10:03 brackets  
drwx----- 2 root root 16384 Jan 28 10:22 lost+found  
drwxr-xr-x 9 root root 4096 Jan 7 12:57 VBoxGuestAdditions-4.3.34  
root@deb8-2:~#
```

23.4 Quotas aktivieren

Je nachdem welche Quotaform man möchte, muss sie nun aktiviert werden:

=> `quotaon -avu [-g]`

In unserem Beispiel sieht das so aus:

```
root@deb8-2:~# quotaon -agv  
/dev/sdb7 [/opt]: group Quotas angeschalten  
root@deb8-2:~#
```

23.5 Ausflug ins Thema Benutzerverwaltung

```
root@deb8-2:~# groupadd technik  
root@deb8-2:~#  
root@deb8-2:~# useradd -m -g technik -G audio,video,lpadmin,plugdev -d /opt/heide heide  
root@deb8-2:~# passwd heide  
Geben Sie ein neues UNIX-Passwort ein:  
Geben Sie das neue UNIX-Passwort erneut ein:  
passwd: Passwort erfolgreich geändert  
root@deb8-2:~#  
root@deb8-2:~# ls -l /opt/  
insgesamt 36  
-rw----- 1 root root 7168 Jan 29 12:33 aquota.group  
drwxr-xr-x 7 root root 4096 Jan 18 10:03 brackets  
drwxr-xr-x 5 heide technik 4096 Jan 29 12:40 heide  
drwx----- 2 root root 16384 Jan 28 10:22 lost+found  
drwxr-xr-x 9 root root 4096 Jan 7 12:57 VBoxGuestAdditions-4.3.34  
root@deb8-2:~#  
root@deb8-2:~#  
root@deb8-2:~# su - heide  
$  
$ bash  
heide@deb8-2:~$  
heide@deb8-2:~$ pwd  
/opt/heide  
heide@deb8-2:~$  
heide@deb8-2:~$ quota  
heide@deb8-2:~$  
heide@deb8-2:~$ quota -g  
Dateisystemquotas für group audio (gid 29): kein  
Dateisystemquotas für group video (gid 44): kein  
Dateisystemquotas für group plugdev (gid 46): kein  
Dateisystemquotas für group lpadmin (gid 110): kein  
Dateisystemquotas für group technik (gid 1005): kein  
heide@deb8-2:~$  
heide@deb8-2:~$ exit  
$ exit  
root@deb8-2:~#
```

23.6 Quotas für User/Group editieren

Allgemeine Syntax:

```
=> edquota -u max
=> [Für Gruppen-Quota: edquota -g users]
```

Hinweise zur Berechnung:

- Als Einheit werden Blöcke in der Größe von 1kB angegeben.
- Gehen wir von der Standard-Dateisystemblockgröße von 4096 Byte aus (siehe `dumpe2fs -h /dev/sdb7 | grep Block\size`), verbrauchen wir zwangsläufig 4 Blöcke (zu je 1 kB), um eine winzige Datei von einem Byte speichern zu können: \Rightarrow `echo > 1-byte-datei`
- 1MB beansprucht dann: $1\text{kB} * 1024 = 1024\text{kB} = 1024$ Blöcke
- Einfache Umrechnungshilfe: `Angezeigte Blockgröße / 1024 = Größe in MB`

Editieren wir nun als root die Gruppen-Quota und setzen testhalber einen Minimalwert, indem wir zum aktuellen Verbrauch (Spalte 2: Blöcke, im folgenden Beispiel 20) vier Blöcke hinzuaddieren. Am besten tun wir dies beim Soft- und Hardlimit gleichermaßen (`weich = 24`, `hart = 24`). Um einen kleineren Wert (z.B. 3 Blöcke) zu erhöhen, macht keinen Sinn - das Dateisystem gibt das 4-kByte-Raster vor!

Setzen wir dies nun praktisch um, sieht das so aus:

```
edquota -g technik
```

Wir befinden uns nun im Editiermodus, wobei der vorkonfigurierte Standardeditor aufgerufen wird (Ein Ausweg wäre `EDITOR=vi edquota -g technik`):

```
Datenträgerquotas für group technik (gid 1005):
Dateisystem      Blöcke    weich    hart    Inodes    weich    hart
/dev/sdb7        20        24       24       5         0        0
```

Mit dieser Einstellung testen wir nun die Angelegenheit:

```
root@deb8-2:~# su - heide
$
$ bash
heide@deb8-2:~$ pwd
/opt/heide
heide@deb8-2:~$
heide@deb8-2:~$ ls -la
insgesamt 64
drwxr-xr-x 2 heide technik 4096 Jan 29 13:55 .
drwxr-xr-x 6 root  root  45056 Jan 29 12:40 ..
-rw----- 1 heide technik  303 Jan 29 13:33 .bash_history
-rw-r--r-- 1 heide technik  220 Nov 13  2014 .bash_logout
-rw-r--r-- 1 heide technik 3515 Nov 13  2014 .bashrc
-rw-r--r-- 1 heide technik  675 Nov 13  2014 .profile
heide@deb8-2:~$
heide@deb8-2:~$ echo > 1-byte-datei
heide@deb8-2:~$
heide@deb8-2:~$ echo > 1-byte-datei_No2
sdb7: write failed, group block limit reached.
bash: echo: Schreibfehler: Der zugewiesene Plattenplatz (Quota) ist überschritten.
heide@deb8-2:~$
```

23.7 Kontrolle der Einstellungen

- Als User kann man sein Kontingent so prüfen: `quota [-g]`
- root kann sich einen kompletten Report so ausgeben: `repquota -a [-g]`

23.8 Sicherheitshinweis

Wird mit Gruppen-Quotas gearbeitet, kann ein Benutzer die Restriktionen u.U. folgendermaßen überwinden:

- Er schaut mit `id` nach, in welchen (sekundären) Gruppen er außerdem Mitglied ist und betritt sie mit `newgrp <GRUPPE>`.
z.B.: `'newgrp audio'` (per Default ohne Passwortabfrage möglich)
- Mit dieser neuen primären Gruppe, für die es keine Beschränkung gibt, kann er nun wieder Daten ablegen!

Abhilfe: Entfernen des SUID-Bits von `/usr/bin/newgrp` (Damit schaltet man aber diesen Mechanismus komplett ab.)

24 Topic 104.5 - Manage file permissions and ownership

24.1 Zugriffsrechte im symbolischen Modus

Neben dem schon bekannten oktalen Modus, gibt es auch den symbolischen, der leichter bei einfachen Änderungen zu handhaben ist. In der Regel arbeitet er relativ zu dem bereits bestehenden Recht:

Z.B. ein Skript ausführbar machen:

```
echo "echo Hallo Welt..." > myworld.sh
chmod +x myworld.sh          # => -rwxr-xr-x    (ist die Kurzform von 'chmod a+ ←
    x myworld.sh')
```

Möchte man, dass nur der Besitzer das Skript ausführen kann, müssen jetzt das x-Recht bei Gruppe und Others entziehen:

```
chmod go-x myworld.sh      # -rwxr--r--
```

Soll eine Datei keiner nutzen dürfen, kann das auf zweierlei Weise geschehen:

```
chmod ugo=--- myworld.sh
```

Das folgende ist nur die Kurzform (leerer Wert bedeutet "auf Null setzen"):

```
chmod ugo= myworld.sh
```

24.2 Zugriffsrechte für Sonderfälle

Siehe Seite 193

Motivation: Gewisse Unix-Mechanismen würde ohne Sonder-Bits nicht funktionieren, z.B. könnte ein Benutzer sein Passwort nicht ändern.

24.2.1 Set User ID - mit Besitzerrechten ausführen

Beginnen wir mit dem ersten Sonderbit, das für erhöhte Rechte beim Ausführen eines Programmes sorgt: Der aufrufende, einfache Nutzer soll mit den Rechten arbeiten können, die sonst nur dem eingetragenen Besitzer zustehen würden.

```
ls -l /usr/bin/passwd      # Man beachte das 's' bei User anstelle des 'x':      ↔
-rwsr-xr-x
```

Weil das *s* bei *User* steht, bezeichnet man dies als Set User ID Bit (SUID).

Im praktischen Falle bedeutet dies, dass ein Nutzer die */etc/shadow* nur dank dem SUID-Sonderbit ändern kann.

Wie setzt man das SIUD-Bit?

- a) `chmod 4755 /usr/bin/passwd`
- b) `chmod u+s /usr/bin/passwd`

Hier die im Zusammenhang stehenden Dateien und ihre Rechte:

```
root@deb8-2:~# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 54192 Nov 18 09:11 /usr/bin/passwd
root@deb8-2:~

root@deb8-2:~# echo UNIX User DBMS: ; ls -l /etc/passwd /etc/shadow /etc/group
UNIX User DBMS:
-rw-r--r-- 1 root root 1339 Feb 2 09:01 /etc/group
-rw-r--r-- 1 root root 2966 Feb 2 09:02 /etc/passwd
-rw-r----- 1 root shadow 2172 Feb 2 08:56 /etc/shadow
root@deb8-2:~#
```

24.2.2 Set Group ID - mit Gruppenrechten ausführen

Es wird benötigt, um erhöhte Aufgaben mit den Rechten einer bestimmten Gruppe auszuführen, beispielsweise Nachrichten auf Tele Types senden:

Kein anderer Nutzer könnte Mitteilungen auf mein Terminal senden bei dieser Rechtesituation (Lesen und Schreiben ist nur für mich selbst zulässig!)

```
tux@deb8-2:~$
tux@deb8-2:~$ tty
/dev/pts/0
tux@deb8-2:~$
tux@deb8-2:~$ ls -l /dev/pts/0
crw----- 1 tux tty 136, 0 Feb 2 10:57 /dev/pts/0
tux@deb8-2:~$
```

Der Schlüssel für das Problem liegt in den Sonderrechten des ausführbaren Programmes, mittels dessen wir dies nur bewerkstelligen werden:

```
tux@deb8-2:~$ ls -lL /usr/bin/wall /usr/bin/write
-rwxr-sr-x 1 root tty 27232 Mär 30 2015 /usr/bin/wall
-rwxr-sr-x 1 root tty 14592 Okt 17 2014 /usr/bin/write
tux@deb8-2:~$
```

Bitte beachten: Entscheidend ist die Gruppenzuordnung der Applications zu *tty* bei gleichzeitigem Setzen des *s*-Bits beim Gruppenzugriffsrecht.

Und natürlich können das SUID- und das SGID-Bit gleichzeitig vorkommen:

```
tux@deb8-2:~$ find /usr/bin -perm 6755
/usr/bin/X
/usr/bin/at
/usr/bin/procmail
tux@deb8-2:~$
```

Der Sinn dahinter ist z.B. beim Zeitplaner *at*, Spool-Jobs unter */var/spool/cron/* verwalten zu können. Die Rechte sehen bei Debian so aus:

```
drwxrwx--T 1 daemon daemon 36 Feb  2 10:48 atjobs
```

Wir sehen hier ein *T* bei *Others*, was uns zum nächsten Sonderbit, dem Sticky-Bit führt.

24.2.3 Sticky Bit - nur der Besitzer/Ersteller darf modifizieren

Bedeutung: "klebrig" — es wurde früher Applications mitgegeben, die nicht aus dem Arbeitsspeicher verdrängt werden, sondern dort kleben bleiben sollten.

Heute wird es für eine Art Austausch-Verzeichnis eingesetzt, das erst einmal Schreibrechte für alle ermöglicht, wobei es dann einschränkend wirkt: Nur der Besitzer/Ersteller darf seine Files modifizieren (auch wenn er den Fehler begehen würde, Others Schreibrechte zu geben) Es ignoriert im Prinzip die gesetzten Rechte bei Group und Others.

Im Praxisbeispiel:

```
tux@deb8-2:~$ ls -ld /tmp/ /var/tmp/
drwxrwxrwt 1 root root 462 Feb  2 11:45 /tmp/
drwxrwxrwt 1 root root 160 Feb  2 07:51 /var/tmp/
tux@deb8-2:~$
```

Wollen wir die Auswirkungen testen, dann legen wir mit anderer Nutzerkennung Files an, und versuchen sie als anderer User zu bearbeiten:

- Als Standardnutzer *tux*: `echo Zeile 1 > /tmp/vonTux`
- Als *linus*: `echo Zeile 2 >> /tmp/vonTux` (FEHLER: Keine Schreibrechte!)
- Als *root*: `chmod o-t /tmp` (Die Sicherheitsbarriere wird entfernt)
- Als *linus* den Versuch von b) wiederholen... (Funktioniert jetzt)
- Am Schluss bitte wieder als *root* das sticky-Bit hinzufügen mit `chmod o+t /tmp`, alternativ: `chmod 1777 /tmp`

In der Umsetzung sieht das so aus:

```
tux@deb8-2:~$ echo Zeile 1 > /tmp/vonTux
tux@deb8-2:~$ chmod 646 /tmp/vonTux
tux@deb8-2:~$ ls -l /tmp/vonTux
-rw-r--rw- 1 tux tux 8 Feb  2 12:27 /tmp/vonTux
tux@deb8-2:~$
tux@deb8-2:~$ su - linus
Passwort:
$ echo Zeile 2 >> /tmp/vonTux
$
$ rm /tmp/vonTux
rm: das Entfernen von „/tmp/vonTux“ ist nicht möglich: Die Operation ist nicht erlaubt
$
$ cat /tmp/vonTux
Zeile 1
Zeile 2
$
$ exit
```

```
tux@deb8-2:~$ /bin/su -
Passwort:
root@deb8-2:~# ls -ld /tmp
drwxrwxrwt 1 root root 474 Feb  2 12:24 /tmp
root@deb8-2:~# chmod o-t /tmp
root@deb8-2:~# ls -ld /tmp
drwxrwxrwx 1 root root 474 Feb  2 12:24 /tmp
root@deb8-2:~# exit
Abgemeldet
tux@deb8-2:~$ su - linus
Passwort:
$ rm /tmp/vonTux
$
$ echo 'Ätsch: ' ; ls -l /tmp/vonTux
Ätsch:
ls: Zugriff auf /tmp/vonTux nicht möglich: Datei oder Verzeichnis nicht gefunden
$ exit
tux@deb8-2:~$ /bin/su -
Passwort:
root@deb8-2:~# chmod o+t /tmp
root@deb8-2:~# ls -ld /tmp
drwxrwxrwt 1 root root 462 Feb  2 12:31 /tmp
root@deb8-2:~# exit
Abgemeldet
tux@deb8-2:~$
```

Ein gutes Dokument, was u.a. dieses Sticky-Bit gut erklärt, ist hier zu finden: <http://ruban.de/wp-content/uploads/2012/02/Unix-s-bit-and-t-bit.pdf>

24.3 Standardrechte für neue Files

Motivation: Wieso haben neue Dateien das Recht 644, neue Verzeichnisse aber 755?

Das Kommando *umask* stellt Standardrechte zur Verfügung, die eine Negativmaske zu den eigentlichen Rechtebits darstellt. Mathematisch werden die Zahlen negiert und mit UND verknüpft.

Einfacher **Ausgangspunkt**: Das Executable-Recht muss für Dateien immer entfernt werden, es könnten ja die englischen Wörter in der Shell eine Sonderbedeutung als Kommando haben!

a) Maximalrecht einer Datei: 666

b) Maximalrecht eines Verzeichnisses: 777

Davon wird nun der umask-Wert abgezogen, was herauskommt ist der tatsächliche Wert der neuen Dateisystemobjekte.

Beispiel Standard-umask-Wert 022:

a) Datei (Minus eins, tiefer als 0 ist nicht zulässig, ist das x-Recht schon abgezogen, gibt es kein -1)

```
  666
-  022
-----
  644
=====
```

b) Verzeichnis (Maximalwert: 7)

```
  777
-  022
-----
  755
=====
```

Je nach Distribution wird `umask` an anderen Stellen gesetzt, ursprünglich war die `/etc/profile` dafür gedacht, heute findet sich die Einstellung meist in der `/etc/login.defs` (`UMASK 022`). Eigene Einstellungen können diesen Wert natürlich überschreiben, eine Möglichkeit ist die Datei `~/.profile`.

24.4 Praxisübung

Temporär soll die `Umask` so gesetzt werden, dass KEINE Rechte für künftige neue Dateisystemobjekte gesetzt sind.

```
tux@deb8-2:~$ umask
0022
tux@deb8-2:~$ umask 027
tux@deb8-2:~$
tux@deb8-2:~$ touch datei-01
tux@deb8-2:~$ mkdir verzeichnis-01
tux@deb8-2:~$
tux@deb8-2:~$ ls -ld datei-01 verzeichnis-01/
-rw-r----- 1 tux tux 0 Feb  3 10:40 datei-01
drwxr-x--- 2 tux tux 6 Feb  3 10:40 verzeichnis-01/
tux@deb8-2:~$
```

Setzen Sie einen passenden `umask`-Wert, der dem Besitzer nur Lesen und Ausführen gibt, Gruppenmitgliedern Lesen, Schreiben und Ausführen, der Rest erhält kein Recht.

=> `umask 207`

Erweitern Sie die Aufgabenstellung, so dass Dateien explizit Ausführungsrechte erhalten.

=> `umask 107`

ACHTUNG: Das bringt das Schreibrecht für den Besitzer, was nicht gewollt war. (7-1=6)! Diese Aufgabe lässt sich nicht umsetzen, da bei Dateien automatisch -1 gerechnet wird.

Gängige `umask`-Werte (Quelle: <http://www.webune.com/forums/umask-calculator.html>)

Umask	Created Files	Created Directories
000	666 (rw-rw-rw-)	777 (rwxrwxrwx)
002	664 (rw-rw-r--)	775 (rwxrwxr-x)
022	644 (rw-r--r--)	755 (rwxr-xr-x)
027	640 (rw-r-----)	750 (rwxr-x---)</td
077	600 (rw-----)	700 (rwx-----)
277	400 (r-----)	500 (r-x-----)

24.5 Eigentümerschaft von Files ändern

⇒ Siehe Seite 195 f

Das Kommando `chgrp` zum Ändern der Gruppenzugehörigkeit wird eher selten eingesetzt, weil man sowohl den Eigentümer wie auch den Gruppenbesitz mit dem Kommando `chown` ändern kann:

```
root@deb8-2:~# chgrp audio /home/tux/Austausch
root@deb8-2:~#
root@deb8-2:~# ls -ld /home/tux/Austausch
drwxr-xr-x 2 root audio 6 Feb  3 11:39 /home/tux/Austausch
root@deb8-2:~#
root@deb8-2:~# chown .video /home/tux/Austausch
root@deb8-2:~#
root@deb8-2:~# ls -ld /home/tux/Austausch
drwxr-xr-x 2 root video 6 Feb  3 11:39 /home/tux/Austausch
```

```

root@deb8-2:~#
root@deb8-2:~# chown :plugdev /home/tux/Austausch
root@deb8-2:~#
root@deb8-2:~# ls -ld /home/tux/Austausch
drwxr-xr-x 2 root plugdev 6 Feb  3 11:39 /home/tux/Austausch
root@deb8-2:~#

```

Gibt es Inhalte, die auch mit bearbeitet werden sollen, muss die Option **-R** (Großes R!) Verwendung finden:

```

root@deb8-2:~# cp -v /etc/host* /home/tux/Austausch
„/etc/host.conf“ -> „/home/tux/Austausch/host.conf“
„/etc/hostname“ -> „/home/tux/Austausch/hostname“
„/etc/hosts“ -> „/home/tux/Austausch/hosts“
„/etc/hosts.allow“ -> „/home/tux/Austausch/hosts.allow“
„/etc/hosts.deny“ -> „/home/tux/Austausch/hosts.deny“
root@deb8-2:~#
root@deb8-2:~# ls -l /home/tux/Austausch
insgesamt 20
-rw-r--r-- 1 root root  9 Feb  3 11:42 host.conf
-rw-r--r-- 1 root root  7 Feb  3 11:42 hostname
-rw-r--r-- 1 root root 247 Feb  3 11:42 hosts
-rw-r--r-- 1 root root 411 Feb  3 11:42 hosts.allow
-rw-r--r-- 1 root root 711 Feb  3 11:42 hosts.deny
root@deb8-2:~#
root@deb8-2:~# chown -R tux /home/tux/Austausch/
root@deb8-2:~#
root@deb8-2:~# ls -l /home/tux/Austausch
insgesamt 20
-rw-r--r-- 1 tux root  9 Feb  3 11:42 host.conf
-rw-r--r-- 1 tux root  7 Feb  3 11:42 hostname
-rw-r--r-- 1 tux root 247 Feb  3 11:42 hosts
-rw-r--r-- 1 tux root 411 Feb  3 11:42 hosts.allow
-rw-r--r-- 1 tux root 711 Feb  3 11:42 hosts.deny
root@deb8-2:~#

# => Wird nach dem Punkt oder Doppelpunkt keine Gruppe explizit benannt,
#      setzt das Kommando 'chown' automatisch die primäre Gruppe ein:
root@deb8-2:~# chown -R tux: /home/tux/Austausch/
root@deb8-2:~#
root@deb8-2:~# ls -l /home/tux/Austausch
insgesamt 20
-rw-r--r-- 1 tux tux  9 Feb  3 11:42 host.conf
-rw-r--r-- 1 tux tux  7 Feb  3 11:42 hostname
-rw-r--r-- 1 tux tux 247 Feb  3 11:42 hosts
-rw-r--r-- 1 tux tux 411 Feb  3 11:42 hosts.allow
-rw-r--r-- 1 tux tux 711 Feb  3 11:42 hosts.deny
root@deb8-2:~#

```

24.6 Einen Nutzer ohne Werkzeuge wie *useradd* und *passwd* aus dem Standard-Nutzer *tux* erzeugen

Schritt 1: Es beginnt mit dem Bearbeiten der Benutzerdatenbankdateien, wobei wir das auf sichere Art tun wollen, nämlich über einen im Hintergrund laufenden Locking-Mechanismus.

Kommando: `vipw`

Für Einsteiger sollen an dieser Stelle die Kommandos für das Bearbeiten im *vi* mitgegeben werden:

```

/^tux

yy

```



```

p

Mit 'l' auf Zeile 4 wandern

i      (INSERT), eine 2 an tux anfügen:    tux2

ESC

Mit 'l' weiterwandern: Ziel: User-ID auf 1200 ändern

ESC

Mit 'l' wieder weiterwandern: Ziel: Heimatverzeichnis auf /home/tux2 ändern

ESC

:wq!

```

Nun muss entsprechend auch die dazugehörige Datei */etc/shadow* sicher bearbeitet werden:

Kommando: `vipw -s`

⇒ Nun das Ganze synonym zu oben anpassen, dh.

- Wieder die Zeile von *tux* kopieren
- Den Nutzernamen von *tux* auf *tux2* ändern.

Schritt 2: Heimatverzeichnis erzeugen und mit Standardinhalt befüllen

```

mkdir /home/tux2
cd /etc/skel
find | cpio -pdm /home/tux2

```

Schritt 3: Berechtigungen rekursiv anpassen:

```

root@deb8-2:~# chown -R -v tux2: /home/tux2
der Eigentümer von „/home/tux2/.bash_logout“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2/.bashrc“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2/.profile“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2/Desktop“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2/Musik“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2/Video“ wurde von root:root in tux2:tux geändert
der Eigentümer von „/home/tux2“ wurde von root:root in tux2:tux geändert
root@deb8-2:~#

```

Schritt 4: Jetzt bitte noch Funktionstests durchführen:

1. Klappt der Login mit *tux2* und dem Passwort von *tux*?
Kommando: `su -tux2`
2. Kann *tux2* in */home/tux2* Daten ablegen? Kommando: `pwd && echo 456 >> datei && echo OK.`
Alternativ: `pwd && echo 456 >> datei || echo Es ist etwas schiefgegangen`

Troubleshooting

Sind die Datenbankeinträge OK?

```

root@deb8-2:~# grep tux2 /etc/passwd /etc/shadow
/etc/passwd:tux2:x:1200:1000:tux,,,:/home/tux2:/bin/bash
/etc/shadow:tux2:$6$SqfMZVZl$1mlVxeConR8hautrtW9eADNMergNUnikPIu0/6kKtQhEdS...m6e ←
/:16834:0:99999:7:::
root@deb8-2:~#

```

Gibt es das Heimatverzeichnis mit den passenden Rechten?

Was macht die folgende Kommandozeile im Detail? Hinweis: Zwischen den Back Tick Marks ist eine Kommandozeilensubstitution eingebettet.

```
root@deb8-2:~# ls -ld `cut -d: -f6 /etc/passwd | grep tux2`
drwxr-xr-x 5 tux2 tux 104 Feb  3 12:16 /home/tux2
root@deb8-2:~#
```

Außerdem gibt es noch das Kommando *pwck*, um die Datenbankdateien auf korrekte Syntax zu prüfen.

24.7 Dateiattribute als weitere Möglichkeit, Berechtigungen zu verwalten

Einige Dateisysteme, u.a natürlich btrfs, gestatten die Nutzung von Attributen, wie

Immutable

Eine Änderung ist nicht mal von root möglich, Rechte setzen mit der Option *+i*

Append only

Nur Anhängen ist möglich, Rechte setzen mit der Option *+a*

Directory updates sync

Verzeichnisänderungen synchron auf HDD schreiben, Option: *+D*

Benutzung im Beispiel:

```
tar czf /root/etc-backup-`date +%F_%H:%M`.tar.gz /etc
chattr +i *backup*
lsattr *backup*
```

Dieses Immutable-Recht kann mit *-i* wieder entfernt werden, geht es um eine ganze Verzeichnisstruktur, wo in einem unbekanntem Verzeichnis eine solch immunisierte Datei liegt, dann kann zusätzlich *-R* verwendet werden, was aber einem groben *überbügel*n entspricht.

Frage: Kann *find* evl. schon diese Attribute finden: `man find | grep -i -B3 -A3 attr`

Wir erhalten keine Ausgabe; das bedeutet also nein: Wie es scheint, lässt sich so einfach nicht danach suchen.

25 Topic 104.6 - Create and change hard and symbolic links

Siehe Seite 199 ff

Es gibt zwei Arten von Links, die Hinweisschilder auf die eigentlichen Dateien bzw. Verzeichnisse darstellen.

Hard-Links

- Vom Prinzip her handelt es sich nur um weitere Namen für ein und dieselbe Datei, dh. es wird kein zusätzlicher Speicherplatz auf dem Datenträger benötigt. Daraus ergibt sich, dass sie nicht partitionsübergreifend angelegt werden können.
- Sie sind Grundlage für die Konstruktion von Verzeichnisstrukturen, so sieht man immer bei Verzeichnissen die Anzahl von Hardlinks als Unterverzeichniszähler.
- Eine Verwendung von Hardlinks anstelle von Symlinks scheint nicht immer plausibel zu sein, betrachtet man aber das Dateisystem als Datenbank, in der es keine unnötigen Redundanzen geben soll, hilft das Hardlinken dabei, ein konsistentes System zu erhalten. Siehe auch unten bei "speicherschonende Snapshots" (*cp -al ...*).
- Hardlinks können nicht auf Verzeichnisse gesetzt werden.

Symbolische Links

- Hier handelt es sich um die bestens bekannten Verküpfungen, die als Zeiger auf ein Verknüpfungsziel zeigen. Man kann sie sich auch als Knauf für einen bequemen Zugriff vorstellen.
- Anders als bei Hardlinks ist hier partitions- und netzwerkübergreifende Arbeitsweise möglich.
- Symlinks lassen sich unter bestimmten Umständen verschieben (mittels relativer Pfade angelegt).
- Sie können sowohl auf Dateien wie auch auf Verzeichnisse gesetzt werden.

25.1 Praxisbeispiele zu Hardlinks

Grundsätzlich benötigen Dateisystemstrukturen (Stichwort: Filesystem Hierarchy Standard "HFS") Hardlinks, um Unterverzeichnisse zu referenzieren, damit schließlich auch ein `cd ..` funktioniert:

```
tux@deb8-2:~$ mkdir Container
tux@deb8-2:~$
tux@deb8-2:~$ ls -ld Container/
drwxrwxr-x 2 tux tux 6 Feb  4 10:23 Container/
tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ ls -la Container/
insgesamt 8
drwxrwxr-x  2 tux tux    6 Feb  4 10:23 .
drwxr-xr-x 58 tux tux 4096 Feb  4 10:23 ..
tux@deb8-2:~$
tux@deb8-2:~$ mkdir Container/Kiste
tux@deb8-2:~$
tux@deb8-2:~$ ls -ld Container/
drwxrwxr-x 3 tux tux 18 Feb  4 10:23 Container/
tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ cd Container/Kiste/
tux@deb8-2:~/Container/Kiste$
tux@deb8-2:~/Container/Kiste$ cd ..
tux@deb8-2:~/Container$
tux@deb8-2:~/Container$ cd ..
tux@deb8-2:~$
tux@deb8-2:~$ cd ..
tux@deb8-2:/home$
tux@deb8-2:/home$ cd ..
tux@deb8-2:/$
```

Erzeugen von Hardlinks am Beispiel (aus einem alten Windows NT Lehrbuch):



Figure 1: Ein Henkeltöpfchen mit zwei Anfassern

```
tux@deb8-2:~$ echo Spieglein, Spieglein... > spieglein
tux@deb8-2:~$ ln spieglein spiegel
tux@deb8-2:~$ ls -li spieg*          # WICHTIG: Beide Dateien haben die selbe Inode-Nummer
3760 -rw-rw-r-- 2 tux tux 24 Feb  4 10:28 spiegel
3760 -rw-rw-r-- 2 tux tux 24 Feb  4 10:28 spieglein
tux@deb8-2:~$
```

Eine weitere Einsatzmöglichkeit sind speicherschonende Snapshots (Kein Backupersatz!!). Dies kann vor größeren Auf- und Umräumaktionen sinnvoll sein, um vor ungewollten Datenverlust gewappnet zu sein.

```
cp -al Dokumente Dokumente.hardlinked
```

25.2 Praxisbeispiele zu symbolischen Links

Erzeugen von symbolischen Links (Fortsetzung des obigen Beispiels aus dem alten Windows NT Lehrbuch):

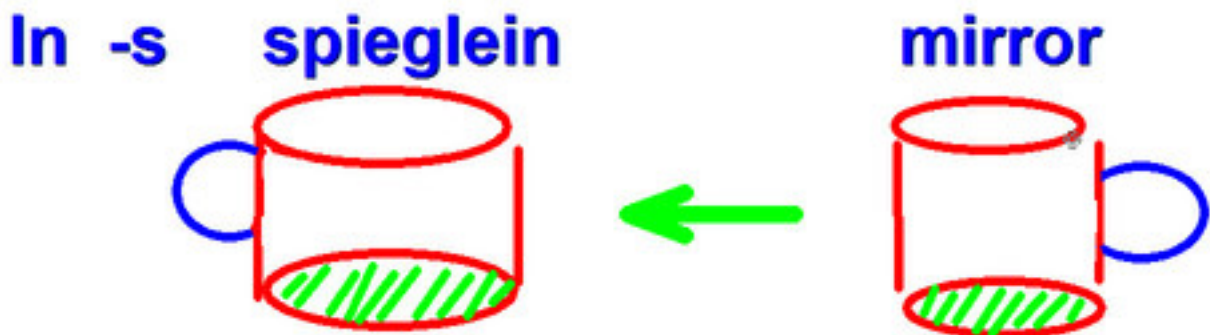


Figure 2: Ein Henkeltöpfchen mit seinem Wegweiser

```
tux@deb8-2:~$ ln -s spieglein mirror
tux@deb8-2:~$
tux@deb8-2:~$
tux@deb8-2:~$ ls -li spieg* mirror
2648738 lrwxrwxrwx 1 tux tux  9 Feb  4 10:53 mirror -> spieglein      # 'mirror' ist eine ↔
      separate Datei
      3760 -rw-rw-r-- 2 tux tux 24 Feb  4 10:28 spiegel          #      mit eigener Inode
      3760 -rw-rw-r-- 2 tux tux 24 Feb  4 10:28 spieglein
tux@deb8-2:~$
```

In der Praxis werden bestimmte Gerätedateien gerne über allgemeine Symlinks angesprochen:

```
# CDROM/DVD-Laufwerk (Brenner)
ln -s /dev/sr0 /cdv/cdrom

# Modem an COM 1
ln -s /dev/ttyS0 /dev/modem
```

26 Topic 104.7 - Find system files and place files in the correct location

Seite 204 ff

Ziel: Lage und Bezeichnung wichtiger Verzeichnispfade kennen und mit Tools herausfinden

26.1 Filesystem Hierarchy Standard (FHS)

Grundsätzlich unterscheidet man 3 Ebenen:

- Wurzel (/): /bin, /sbin, /lib, /etc
- Unix System Resources (statisch): /usr/bin, /usr/sbin, /usr/lib (/usr kann read-only gemountet werden)
- Unix System Resources, eigene Software (statisch): /usr/local/bin, /usr/local/sbin, /usr/local/lib, /usr/local/etc

Wichtige Verzeichnisse, die gesichert werden sollten:

- Alle Nutzerdatenverzeichnisse: /home, /var/mail, /var/spool/mail, /tmp
- Alle Serverfreigabeverzeichnisse: /srv
- Konfigurationen: /etc, /usr/local/etc, /opt
- Betriebssystemdaten: /bin, /sbin, /lib*, /boot, /etc, /usr, /root, /var

Sinnvollerweise brauchen NICHT gesichert werden: /proc, /sys, /run, /tmp

26.2 Suchen nach Dateisystemobjekten

a) Langsame Just-in-Time-Suche mit *find* (siehe man-Page)

b) Schnelle Datenbank-basierte Suche: locate / updatedb

- Paketname: *slocate* = Secure locate, damit einfache Nutzer nicht die Daten anderer angezeigt bekommen
- Aktualisieren des Indizes: updatedb (*nice +19 updatedb &*)

- Speicherort des Datenbank-Indexes: `/var/lib/locatedb` (Debian: `/var/lib/mlocate/mlocate.db`)
- Konfiguration von nicht mit zu indizierenden Verzeichnissen: `/etc/updatedb.conf`
 - EINSTELLUNG: `PRUNEPATHS="/tmp /var/spool /media"`
 - Ist es möglich, mit Pattern wie z.B. `*BACKUP*` zu arbeiten? `man 5 updatedb.conf`
Leider nein: *"Each path name must be exactly in the form in which the directory would be reported by locate(1)."*
 - Dann eben einzeln: `PRUNEPATHS="/tmp /var/spool /media /BACKUP /boot/BACKUP /var/BACKUP"`

26.3 Einen eigenen Index als einfacher Benutzer erzeugen und nutzen

⇒ Siehe *man 8 updatedb*, dieses Fallbeispiel ist dort im Prinzip beschrieben:

- `updatedb --require-visibility 0 --database-root /home/tux --output /home/tux/.my_locatedb`
- `locate --database /home/tux/.my_locatedb .bashrc`
⇒ dies findet `/home/tux/.bashrc`