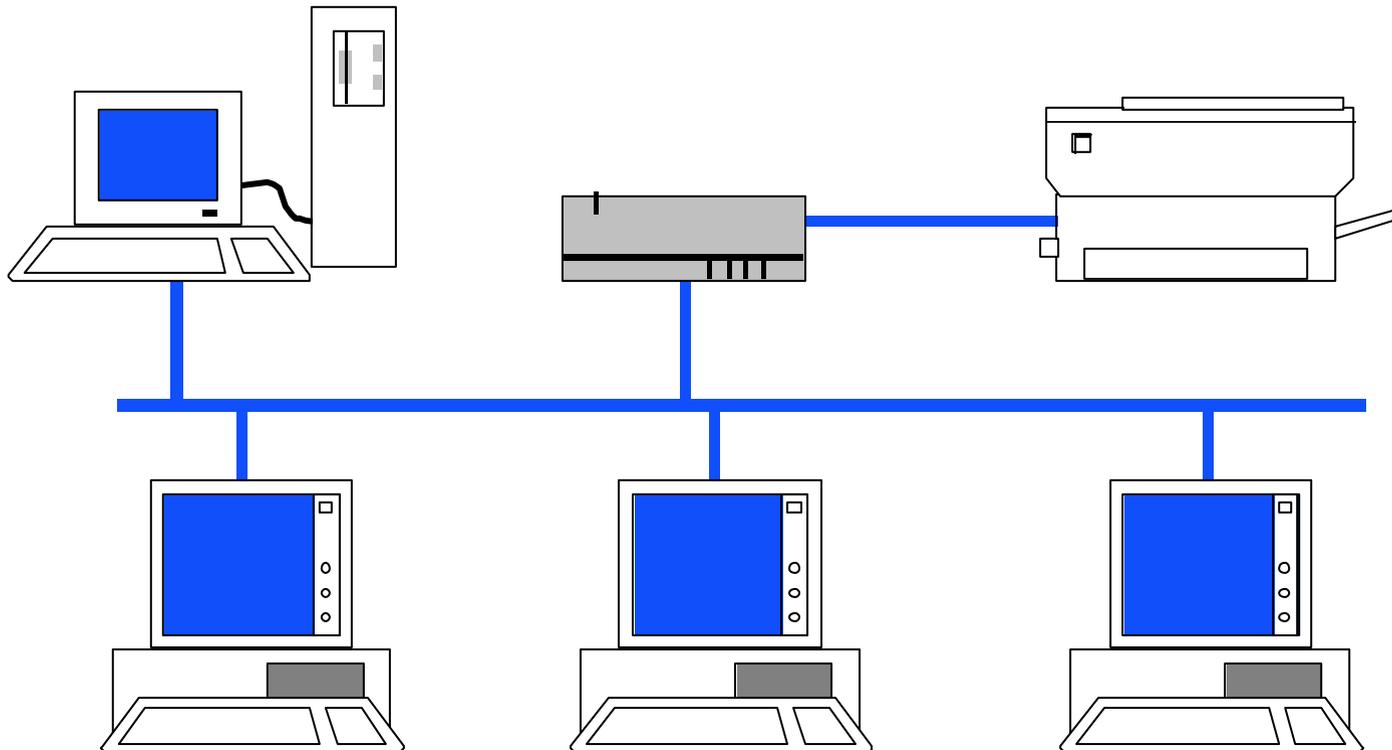


# Linux Einführung

Mag. Dr. Klaus Coufal



# Themenübersicht

- Übersicht über Linux
- „Hardware“-Grundlagen
- Wichtige Basisbefehle
- Arbeiten mit Dateien und Verzeichnissen
- Hilfe

# I. Übersicht über Linux

- Einleitung
- Distributionen
- Einsatzgebiete
- Generelle Unix-Eigenschaften
- Betriebssystemarchitektur
- Kernel
- Dateisysteme
- Shells
- X

# I.1. Einleitung

- 1991 von Linus Torvalds entwickelt (80386)
- GNU-GPL  
(<http://www.gnu.org/copyleft/gpl.html>)
- Viele HW-Plattformen (x86, PPC, 390, ...)
- Wesentliche Unterschiede zu MS-Windows
  - Die Betriebssystemlizenz ist frei
  - Die Konfiguration erfolgt über Textdateien
  - Der Sourcecode ist verfügbar
  - Die graphische Oberfläche ist netzwerkfähig

# I.2. Distributionen

- Red Hat (vor allem in den USA verbreitet)
- S.u.S.E. (in Europa sehr verbreitet)
- Caldera (graph.Installation, Netwaresupport)
- Mandrake
- Debian
- Slackware (die klass.Variante; Linux pur)
- ...

# I.3. Einsatzgebiete

- Netzwerkserver
  - Fileserver (Mars, Samba, NFS, ...)
  - Applicationserver
  - Internetserver
- Workstation
  - Durch die X-Oberfläche in immer mehr Bereiche

# I.4. Unix-Eigenschaften

- Unix ist ein interaktives Multiuser-/Multitasking Betriebssystem
- Unix ist fileorientiert (z.B.: jedes Gerät ist eine Datei)
- Unix ist netzwerkfähig
- Unix ist ein offenes Betriebssystem und das einzige für das ein Standard geschaffen wurde (POSIX)
- Unix ist flexibel an die Anwenderbedürfnisse anpassbar (verschiedene Shells, ...)
- Unix ist „relativ“ leicht konfigurierbar und kann den Zugriff auf alle Ressourcen sehr fein einstellen (Nicht nur Benutzer/Administrator)

# 1.5. Betriebssystemarchitektur

Shell/Anwendung					Usermode
System-APIs (System Call Interface)					Kernel
Virtual File System (VFS)	character devices	Abstract Network Services	Memory Manager	Process Manager	
FS-Drivers		TCP/IP Driver	VM Driver		
HW-Drivers	HW-Drivers	NIC Driver	Memory Driver	CPU Driver	
HD/FD/CD /...	Konsole/...	NIC	Memory	CPU	Hardware

# 1.6. Kernel

- offizielle Linuxkernel (von Linus Torvalds und dem Kernelteam)
- freier Sourcecode  $\Rightarrow$  maßgeschneiderten Kernel
- Jeder kann einen eigenen Kernel verwenden

# 1.7. Dateisysteme

Unterstützt werden:

- FAT, NTFS, Minix, CDFS, VFAT, HFS(Apple), ...;
- eigene Dateisysteme: extfs2, Reiser, ...;
- Netzwerkdateisysteme (NFS, DFS, ...);
- Kryptographische Dateisysteme (TCFS, ...)

# 1.7. Dateisysteme Beispiel

- / (3. Partition (ext3) auf erster Platte (C:))
- /etc (Unterverzeichnis auf Rootpartition)
- /boot (2. Partition (ext2) auf erster Platte)
- /var (Unterverzeichnis auf Rootpartition)
- /proc (Virtuelles Dateisystem zur Verwaltung)
- /home (Unterverzeichnis auf Rootpartition)
- /floppy (1. Diskettenlaufwerk)
- /mnt (4. Partition (NTFS) auf erster Platte)

# 1.8. Shells

- Bourne-Shell (sh)
- Korn-Shell (ksh)
- C-Shell (csh)
- Bourne Again Shell (bash) - Heute der Quasistandard unter Linux
- ...

# 1.8. Shells – Beispielbefehle

- `ls` Inhaltsverzeichnis (list)
- `cat datei` Anzeige von datei
- `man befehl` Manual für befehl ausgeben
- `cd verz` Wechsle das Verzeichnis auf verz
- `pwd` Anzeige des aktuellen Verzeichnisses  
(print working directory)
- `mkdir verz` Anlegen eines Verzeichnisses
- `rmdir` Löscht ein Verzeichnis
- `mv alt neu` Umbenennen bzw. Verschieben von alt nach neu
- `rm datei` Lösche datei
- `cp quelle ziel` Kopiert quelle nach ziel
- `chmod mode file` Ändern der Berechtigungen

# 1.9. X-Windows

- X-Server (Steuert die Hardware)
- X-Windowmanager (Graphische Präsentation, Aussehen der Fenster)
  - KDE
  - Gnome
- X-Anwendungen (Clientprogramm für die eigentliche Aufgabe)

# II. Hardwaregrundlagen

- Rechneraufbau
- Festplatten
  - Partitionen
  - Partitionstabelle
  - MBR
- Bootvorgang
- Graphiksystem
- Drucksystem

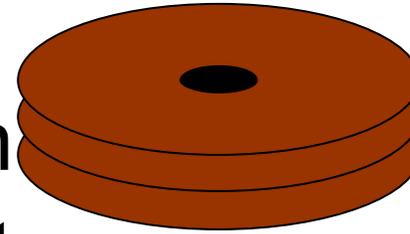
# II.1. Rechneraufbau

- CPU Pentium, ...
- Speicher DDR-RAM, ...
- Bussystem(e) ISA, PCI, AGP, ...
- Netzwerk 10/100 Mbit/s Ethernet, ...
- Festplatte IDE, SCSI, ...
- Drucker Parallel, USB, GDI, ...
- Ausgabe Graphikkarte und Monitor, ...
- Eingabe Maus, Tastatur, Scanner, ...

## II.2. Festplatten

- Werden durch ihre Anschlußtechnik unterschieden (in LINUX andere Namen: hda, hdb, ..., sda, sdb, ...)
- Der physische Aufbau der Festplatte ist heute nebensächlich
- Werden in Partitionen (Verwaltungseinheit) eingeteilt
- Primäre und Erweiterte Partitionen

## II.2. Festplatten 2



- Maximal vier Primäre Partitionen
- Eine dieser vier kann eine erweiterte Partition sein, die Ihrerseits wieder Partitionen aufnehmen kann.
- Ein Masterpartitionstabelle pro Festplatte und je eine weitere Partitionstabelle pro Partition in der erweiterten Partition

## II.2. Festplatten 3

- Auf der ersten Festplatte existiert ein MBR (Master Boot Record), der u.a. die Masterpartitionstabelle enthält und darin ist eine Partition als aktiv markiert.
- Ferner enthält der MBR einen Bootloader, damit das System starten kann.

## II.2. Festplatten 4

- Dadurch ist es möglich mehrere Betriebssysteme auf dem PC zu haben
- Z.B.:
  - 1. Partition: NTFS für Windows 2000
  - 2. Partition: Erweitert
    - 5. Partition swap für Linux
    - 6. Partition ext3 mit / für Linux
    - 7. Partition ext3 mit /home für Linux

# II.3. Bootvorgang

- Einschalten
- POST (Power On Self Test)
- Suchen eines bootbaren Gerätes (BIOS-Einstellungen z.B.: IDE0)
- Laden des MBR und starten des Bootloaders
- Der sucht aktive Partition, lädt von dort den Bootrecord und startet diesen

## II.3. Bootvorgang 2

- Damit immer nur ein Betriebssystem aktiv und daher das Umschalten zwischen zwei Betriebssystemen umständlich
- Bootmanager erleichtert dieses durch eine Auswahlmöglichkeit
  - Lilo, grub
  - Powerquest Bootmagic
  - ...

## II.4. Graphiksystem

- Besteht aus einer Graphikkarte und einem Monitor
- Bei modernen Computern und modernen Distribution i.a. automatisch erkannt (wenn nicht ist Expertenwissen notwendig)
- Shared Memory vermeiden (Notebooks)

# II.5. Drucksystem

- Eine der Schwachstellen von Linux
- Viele PC-Drucker GDI-Drucker
- Diese erfordern viel CPU-Leistung und unter nicht Windows-Systemen eigene Treiber
- Vor Kauf die Unterstützung durch die verwendeten Betriebssystem prüfen

# III. Wichtige Basisbefehle

- Anmelden
- Abmelden
- Informationen über Benutzer
- Passwort ändern
- Datum und Uhrzeit ansehen
- Allgemeine Hinweise

# III.1. Anmelden

- Text:
  - Benutzername
  - Passwort (Alternativen sind möglich)
- X-Windows
  - Benutzername oder Auswahl (Mausklick)
  - Passwort (Alternativen sind möglich)

## III.2. Abmelden

- `logout`
  - Beendet die Loginshell
  - In Subshell wird darauf hingewiesen, daß nicht in der Loginshell gearbeitet wird
- `exit`
  - Beendet die momentane Shell
  - Bei der Loginshell erfolgt dadurch ein Logout

# III.3. Infos über Benutzer

- W
  - Zeigt an wer angemeldet ist und was diese Benutzer machen
  - z.B.:

```
rechner:~ # w
```

```
10:29am up 3:47, 1 user, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE        JCPU        PCPU        WHAT
root      pts/0    192.168.13.2  10:29am    1.00s      0.22s      0.05s      w
```

# III.3. Infos über Benutzer 2

- who
  - Zeigt wer angemeldet ist
  - z.B.:

```
rechner:~ # who  
root      pts/0      Nov 13 10:29 (192.168.13.2)
```

# III.3. Infos über Benutzer 3

- `whoami`
  - Gibt des effektiven UserID aus
  - z.B.:

```
rechner:~ # whoami  
root
```

# III.3. Infos über Benutzer 4

- `id`
  - Gibt den wirklichen und den effektiven UserId und GroupID aus
  - z.B.:

```
rechner:~ # id
uid=0(root) gid=0(root)
groups=0(root),1(bin),14(uucp),15(shadow),16(dial
out),17(audio),65534(nogroup)
```

# III.4. Passwort ändern

- `passwd`
  - Ändert das Passwort des momentan aktiven Benutzers
  - Erfordert die Eingabe des derzeitigen und die zweimalige Eingabe des neuen Passwortes
  - Selbstverständlich ohne Anzeige

# III.5. Datum und Uhrzeit

- `date`
  - Gibt Datum und Uhrzeit aus
  - z.B.:

```
rechner:~ # date
```

```
Wed Nov 13 10:31:39 MET 2002
```

# III.6. Allgemeine Hinweise 1

-  Tabulator
  - Ergänzen von Dateinamen
-  Space
  - Um eine Seite weiter in vielen Befehlen
-  Enter oder  Return
  - Eingabe oder um eine Zeile weiter
-  Pfeiltasten
  - Kommandoeditor

# III.6. Allgemeine Hinweise 2

- Optionen bei den Befehlen in zwei Formaten
  - -o Einbuchstabile Optionen
  - --option Klartextoptionen
- Trend zu Klartextoptionen

```
rm -d verzeichnis
```

```
rm --directory verzeichnis
```

# III.6. Allgemeine Hinweise 3

## Sonderzeichen innerhalb von Befehlen

- | Verkettung von Befehlen  
befehl1 | befehl2
- > Umleitung der Ausgabe (neue Datei)  
befehl > datei
- >> Umleitung der Ausgabe als Anhang  
befehl >>datei

# III.6. Allgemeine Hinweise 4

- < Umleitung der Eingabe  
befehl < datei
- n>, n>>, n< Umleitung des  
entsprechenden Kanals (n=Nummer)  
befehl 2> datei
- \z Sonderzeichen (\> =>> ...)

# IV. Arbeiten mit Dateien und Verzeichnissen

- pwd, cd, ls
- locate, find
- which, file
- cp, mv
- ln
- mkdir, touch, rm, rmdir
- cat, less, more
- sed, awk,
- lpr
- tar, gzip, bzip2, zip

# pwd

- **print working directory**
- Anzeige des momentanen Arbeitsverzeichnisses (durch cd eingestellt)
- Syntax: `pwd`

```
klaus@rechner:~ > pwd  
/home/klaus
```

# cd

- **change directory**
- Wechsel des Arbeitsverzeichnisses
- **Syntax:** `cd [Zielverzeichnis]`

```
klaus@rechner:~ > cd /etc
```

```
klaus@rechner:/etc > cd
```

```
klaus@rechner:~ >
```

# ls

- **list files**
- Liste Dateien (dir)
- **Syntax:** `ls [optionen] [Muster]`
- **Wichtige Optionen**
  - `-l` Longformat
  - `-a` Alle Dateien (auch versteckte)

# ls – Beispiele

```
klaus@rechner:~ > ls  
down ipchains.txt
```

```
klaus@rechner:~ > ls -l  
insgesamt 20
```

```
-rwxrwxrwx  1 klaus  users          12 Feb 19  2001 down  
-rw-r--r--  1 root   root        13649 Feb  7  2001 ipchains.txt
```

```
klaus@rechner:~ > ls -al .u*
```

```
-rw-r--r--  1 klaus  users      10972 May  5  2000 .uitrc.console  
-rw-r--r--  1 klaus  users       9394 May  5  2000 .uitrc.vt100  
-rw-r--r--  1 klaus  users       9394 May  5  2000 .uitrc.vt102  
-rw-r--r--  1 klaus  users     10687 May  5  2000 .uitrc.xterm  
-rw-r--r--  1 klaus  users       324 May  5  2000 .urlview
```

# locate

- Sucht in einer Datenbank nach Dateien, die einem Muster entsprechen (Datenbankupdate mittels `updatedb`)
- Syntax: `locate [--help] pattern`

```
klaus@rechner:~ > locate resolv.conf  
/etc/resolv.conf  
/usr/share/man/man5/resolv.conf.5.gz  
/var/adm/SuSEconfig/md5/etc/resolv.conf
```

# find

- Sucht nach Dateien in der Verzeichnishierarchie
- **Syntax:** `find [path...] [expression]`

```
klaus@rechner:/home > find . -name down  
./klaus/down
```

```
find: ./shutdown: Keine Berechtigung
```

# which

- Zeigt den Ort eines Befehls an
- Syntax: `which progname`

```
klaus@rechner: /home > which ls
```

```
klaus@rechner: /home > which find  
/usr/bin/find
```

# file

- Versucht den Typ einer Datei zu eruieren
- Syntax: `file datei`

```
klaus@rechner:~ > file /usr/bin/file
/usr/bin/file: ELF 32-bit LSB executable, Intel
 80386, version 1, dynamically linked (uses
  shared libs), stripped
klaus@rechner:~ > file down
down: ASCII text
klaus@ rechner:~ > file .nc_keys
.nc_keys: English text
```

# cp

- **copy** files or directories
- kopiert eine oder mehrere Dateien
- **Syntax:** `cp [optionen] Quelle Ziel`

```
klaus@rechner:~ > cp down new
```

```
klaus@rechner:~ > cp /etc/resolv.conf .
```

# mv

- **move** files or directories
- verschiebt eine Datei oder benennt sie um
- **Syntax:** `mv [optionen] Quelle Ziel`

```
klaus@rechner:~ > mv new neue_datei (rename)
```

```
klaus@rechner:~ > mv ipchains.txt sub (move)
```

# ln

- **link file or directory**
- erzeugt einen Verzeichniseintrag einer existierenden Datei unter anderem Namen
- **Syntax:** `ln [optionen] Quelle [Ziel]`
- **Hardlinks und Symbolische Links**

# In – Beispiele

```
klaus@rechner:~/sub > dir
insgesamt 16
-rw-r--r--  1 root      root          13649 Feb  7  2001 ip.txt
klaus@rechner:~/sub > ln ip.txt test1
klaus@rechner:~/sub > ln -s ip.txt test2
klaus@rechner:~/sub > dir
insgesamt 32
-rw-r--r--  2 root      root          13649 Feb  7  2001 ip.txt
-rw-r--r--  2 root      root          13649 Feb  7  2001 test1
lrwxrwxrwx  1 klaus    users          12 Nov 13 12:23 test2 -> ip.txt
```

# mkdir

- **make directory**
- erzeugt ein leeres Verzeichnis
- **Syntax:** `mkdir Verzeichnis`

```
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus   users           12 Feb 19   2001 down
-rw-r--r--   1 klaus   users           313 Nov 13 12:10 resolv.conf
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 sub
klaus@rechner:~ > mkdir neu
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus   users           12 Feb 19   2001 down
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 neu
-rw-r--r--   1 klaus   users           313 Nov 13 12:10 resolv.conf
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 sub
```

# touch

- ändert die Zeitmarkierung einer Datei bzw. legt eine Datei an
- **Syntax:** touch [optionen] Datei

```
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus    users           12 Feb 19  2001 down
-rw-r--r--   1 root     root          13649 Feb  7  2001 ipchains.txt
klaus@rechner:~ > touch testdatei
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus    users           12 Feb 19  2001 down
-rw-r--r--   1 root     root          13649 Feb  7  2001 ipchains.txt
-rw-r--r--   1 klaus    users              0 Nov 13 13:28 testdatei
```

# rm

- **remove files**
- Entfernt Dateien
- **Syntax:** `rm [optionen] Pfad`
- **Wichtige Optionen**
  - `-d` Um Verzeichnisse zu löschen
  - `-r` Rekursiv (d.h. Unterverzeichnisse und Dateien darin)

```
klaus@rechner:~ > rm testdatei
```

# rmdir

- **remove directory**
- löscht leere(!) Verzeichnisse
- **Syntax:** `rmdir [-p] Verzeichnis`

```
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus   users           12 Feb 19  2001 down
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 neu
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 sub
klaus@rechner:~ > rmdir neu
klaus@rechner:~ > dir
-rwxrwxrwx   1 klaus   users           12 Feb 19  2001 down
drwxr-xr-x   2 klaus   users          4096 Nov 13 13:22 sub
```

# cat

- **concatenate** files
- Gibt Dateien aus oder verkettet Dateien
- **Syntax:** `cat [optionen] [Datei...]`

```
klaus@rechner:~ > cat down  
su -c halt
```

# less

- Zeigt Dateien seitenweise an
- Möglichkeit des Blätterns und Suchens
- Syntax: `less [optionen] datei`
- Blättern z.B. mit den Pfeiltasten
- Suchen mit „/suchtext“
- Ende mit 

```
klaus@rechner:~ > less ipchains.txt
```

# more

- Zeigt Dateien seitenweise an
- Environmentvariable MORE enthält die Anzahl der Zeilen
- Syntax: `more [optionen] datei`
- Ende mit 

```
klaus@rechner:~ > more ipchains.txt
```

# sed

- **streamedit** files
- Editieren von Dateien in nicht interaktiver Form
- **Syntax:** `sed [optionen] [Datei...]`

```
klaus@rechner:~ > cat sedtest
```

```
Das ist eine Testdatei fuer SED
```

```
klaus@rechner:~ > sed -e s/e/o/g sedtest
```

```
Das ist oino Tostdatoi fuor SED
```

```
klaus@rechner:~ > sed -e s/e/o/ sedtest
```

```
Das ist oine Testdatei fuer SED
```

# awk

- Musterscanning und Verarbeitung
- von **A**ho, **K**ernighan, and **W**einberger entwickelt
- Syntax:  

```
awk [opt] -f program-file file
```
- Vollständige Programmiersprache

# lpr

- **off line print**
- Drucken über einen Spooler
- Syntax: `lpr [optionen] [datei]`

# tar

- **tape archiver**
- **verwaltet Dateiarhive**
- **Syntax: tar [optionen] Dateien**

```
klaus@rechner:~ > tar -c -f testtar *
```

```
klaus@rechner:~/test > tar -x -f testtar
```

# gzip

- Dateikomprimierung nach LZ77
- Ersetzt Datei durch komprimierte Version
- Syntax: `gzip [optionen] datei`

```
klaus@rechner:~ > gzip testtar
```

```
klaus@rechner:~ > gzip -d testtar
```

# bzip2, zip

- Andere Filekompressionsprogramme
- Syntax:  
bzip2 [optionen] Dateien  
bunzip2 [optionen] Dateien  
zip [optionen] zipfile dateien
- Kann mehrere Dateien in einem Durchgang komprimieren und ersetzt die Datei nicht

# V. Hilfe

- man
- info
- --help
- -h
- whatis
- apropos
- HOWTOs

# man

- **manual pages**
- Ansehen der Manuals
- **Syntax:** `man [optionen] befehl`  
`man [seite] befehl`
- Ende mit 

```
klaus@rechner:~ > man bzip2
```

```
klaus@rechner:~ > man ls
```

# info

- Lese die info-Dokumente
- Syntax: `info [optionen] thema`
- Ende mit 

```
klaus@rechner:~ > info man
```

```
klaus@rechner:~ > info login
```

# --help, -h

- Bei vielen Befehlen erhält man Hilfe mit der Optionen `--help` oder `-h`
- Syntax: `befehl -h`

```
klaus@rechner:~ > tar --help
```

```
klaus@rechner:~ > man -h
```

# whatis

- durchsucht die Indexdatenbank nach Kurzbeschreibungen
- **Syntax:** `whatis [optionen] suchwort`

```
klaus@rechner:~ > whatis bash
```

```
bash (1) - GNU Bourne-Again SHell
```

# apropos

- sucht die Manualkurzbeschreibung in der Indexdatenbank
- **Syntax:** `apropos [optionen] suchwort`

```
klaus@firewall:~ > apropos bash
bash (1) - GNU Bourne-Again SHell
bashbuiltins (1) - bash built-in commands, see bash(1)
rbash (1) - restricted bash, see bash(1)
bashbug (1) - report a bug in bash
```

# HOWTOs

- Textdateien (heute oft auch HTML-Dateien) mit der Beschreibung eines bestimmten Vorganges
- Autor mit e-Mail-Adresse angegeben
- Oft auch verschiedene Sprachversionen
- English am umfangreichsten