

# 2.Übung

## Systemsoftware (SYS)

Christian Baun  
`cray@unix-ag.uni-kl.de`

Hochschule Mannheim – Fakultät für Informatik  
Institut für Robotik

12.10.2007

# Heute

- Einführung für Linux/UNIX-Anwender
  - Zugang zum System
  - Die Linux/UNIX-Verzeichnisstruktur
  - Mit Verzeichnissen und Dateien arbeiten

## Was ist Linux?

- Freies, plattformunabhängiges Mehrbenutzer-Betriebssystem.
- Im September 1991 von Linus Torvalds ins Leben gerufen.
- Eins der erfolgreichsten Open-Source-Produkte überhaupt.
- Eins der am häufigsten portierten Systeme.
- Fast komplett in C geschrieben.
- Besteht aus:
  - Monolithischer Kernel
  - Viel freie Software
- Hervorragend geeignet für Forschung und Lehre (Open-Source, flexibel einsetzbar, plattformunabhängig, kostenlos verfügbar, hohe Verbreitung...)

# Linux läuft auf fast allem, das rechnen kann



## Die Eingabeaufforderung (Prompt)

- Die Eingabeaufforderung ist die Markierung auf der Kommandozeile, an der Kommandozeilenbefehle eingegeben werden können.
- Das Aussehen der Eingabeaufforderung ist systemabhängig.
- Standard unter aktuellen Linux-Distributionen:
  - Für den Systemadministrator:

```
root@rechnername:/verzeichnis#
```
  - Für normale Benutzer:

```
benutzername@rechnername:/verzeichnis$
```
- Das Aussehen kann angepasst werden  
⇒ Shellvariable \$PS1 (Primary Prompt String)

## Kommandos (Aufbau)

Kommandoname [-Optionen] [Argumente...]

- Optionen werden in der Regel ein oder zwei Bindestriche vorangestellt.
- Übersicht über die wichtigsten Optionen eines Kommandos mit den Optionen `--help`, `-help` oder `-h`

<code>ls</code>	<code>-a</code>	<code>/usr</code>
		
Kommando	Option	Argument

- Umfangreiche Dokumentationen zu (fast) jedem Kommando und seinen Optionen finden sich in den Manualseiten.

```
$ man ls
```

## Ein erstes Kommando – cal

- Mit dem Kommando `cal` wird der aktuelle Monat ausgegeben.
- Optionen ändern das Verhalten von `cal`.
- Um mehr über die Optionen und Argumente des Kommandos zu erfahren:

```
$ cal --help
```

```
user@rechner:~$ cal
      September 2007
Mo Di Mi Do Fr Sa So
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

- Um alle Monate eines ganzen Jahres auszugeben, z.B. 2005:

```
$ cal 2005
```

- Um einen bestimmten Monat auszugeben, z.B. März 1979:

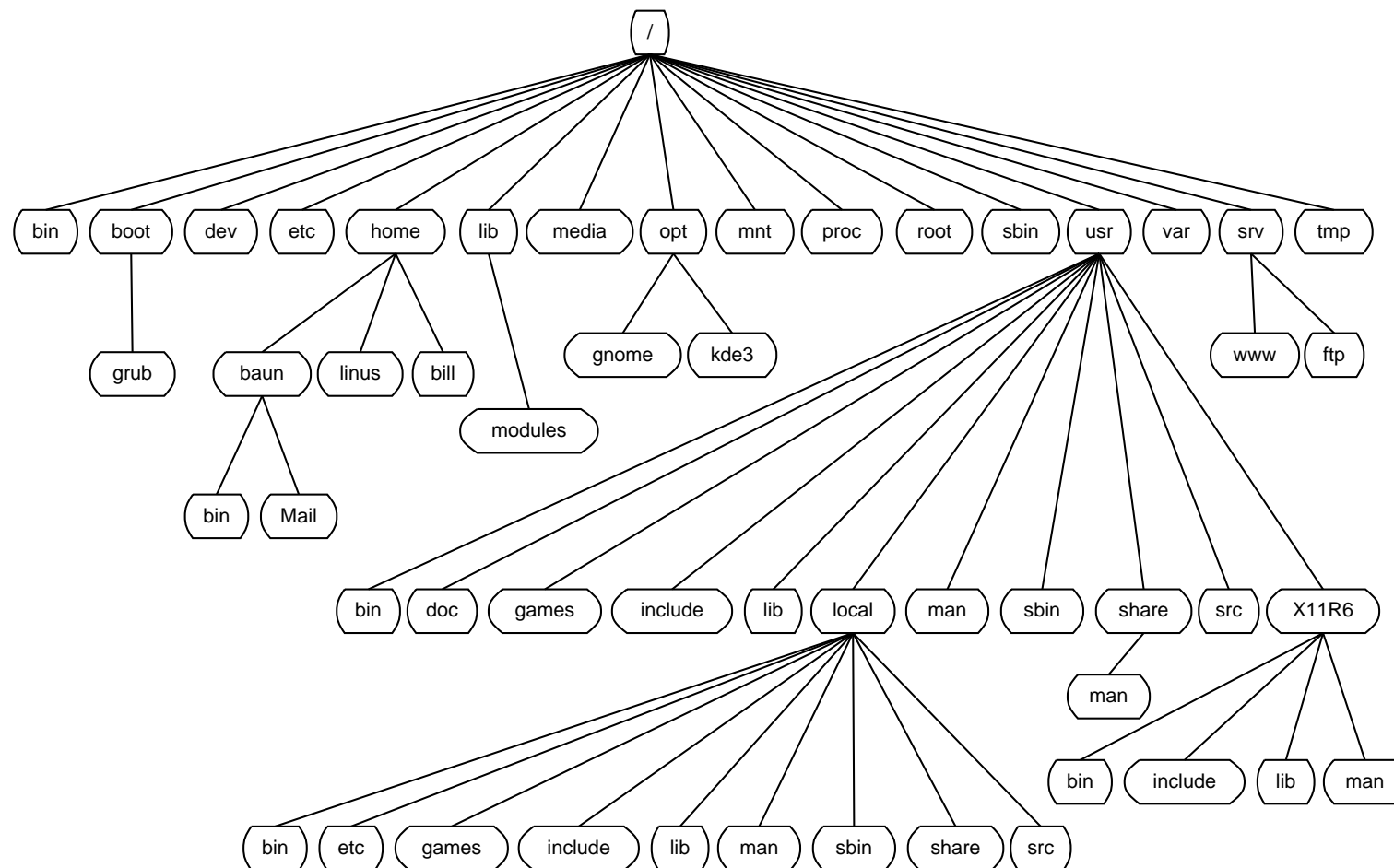
```
$ cal 3 1979
```

## Mit Verzeichnissen arbeiten

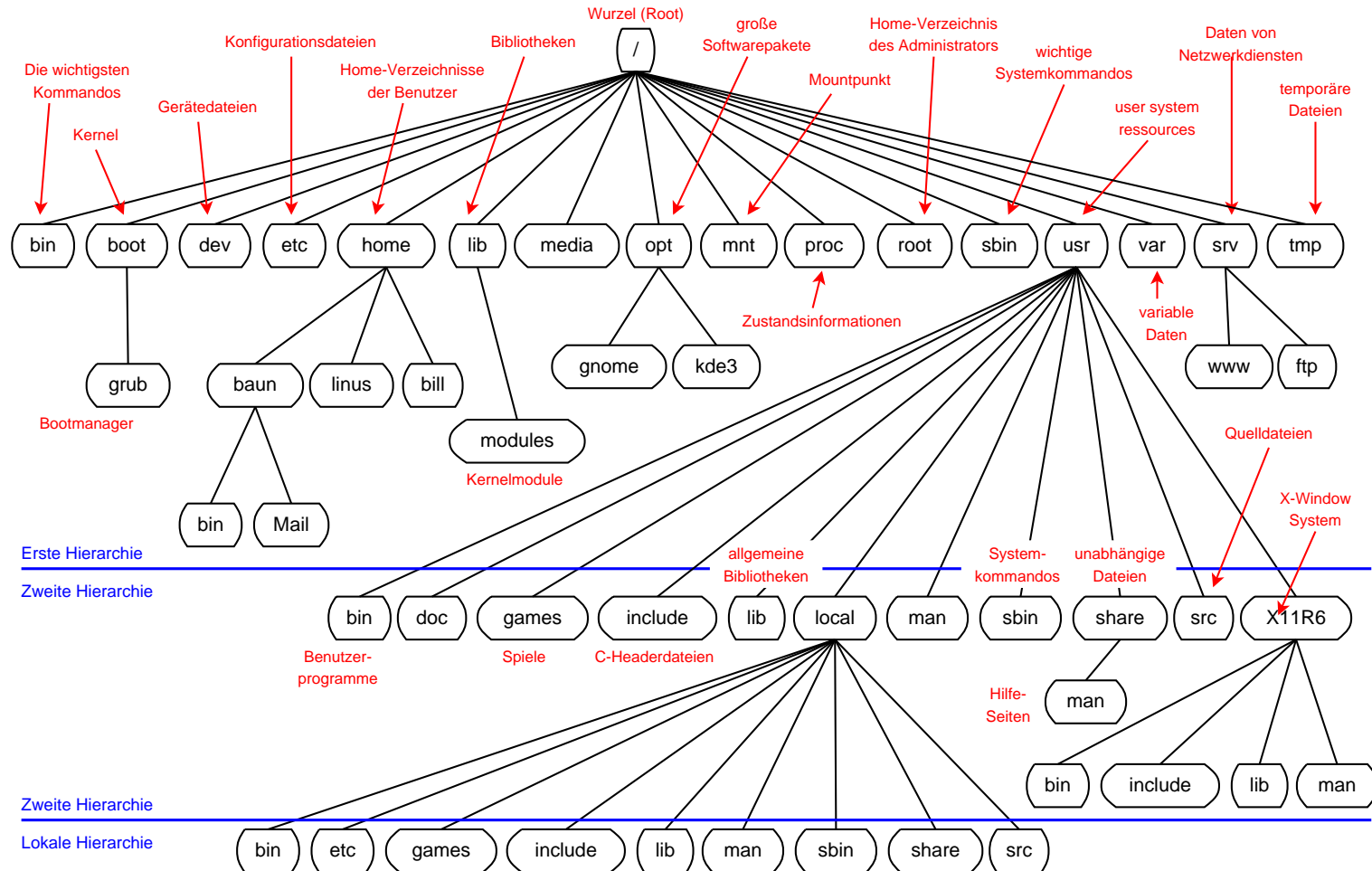
- Durch Verzeichnisse werden Dateien hierarchisch strukturiert.
- Jedes Verzeichnis kann beliebig viele Dateien und (Unter-)Verzeichnisse enthalten.
- Die Struktur der Verzeichnisse ähnelt der eines Baumes.  
⇒ Wurzelverzeichnis /



# Die Linux/UNIX-Verzeichnisstruktur



# Die Linux/UNIX-Verzeichnisstruktur (kommentiert)



## Das aktuelle Verzeichnis ausgeben – pwd

- Abkürzung für **P**resent **W**orking **D**irectory oder **P**rint **W**orking **D**irectory.
- Gibt den kompletten Pfad des aktuellen Verzeichnisses aus.

```
user@rechner:~$ pwd  
/home/user
```

## Verzeichnis anlegen und löschen – mkdir, rmdir

`mkdir [Verzeichnis] ...`

- Das Kommando `mkdir` (Make Directory) erzeugt ein Verzeichnis mit dem als Argument übergebenen Verzeichnisnamen.
- Es können mehrere Verzeichnisse mit einem Aufruf erzeugt werden.

`rmdir [Verzeichnis] ...`

- Das Kommando `rmdir` (Remove Directory) löscht ein **leeres** Verzeichnis.
- nicht-leere Verzeichnisse löschen  $\implies$  `rm -r Verzeichnisname`

## Einfache Beispiele zu mkdir und rmdir

```
user@rechner:~/SYS1$ mkdir testVerzeichnis
user@rechner:~/SYS1$ ls -a
.  ..  testVerzeichnis

user@rechner:~/SYS1$ rmdir testVerzeichnis
user@rechner:~/SYS1$ ls -a
.  ..

user@rechner:~/SYS1$ rmdir testVerzeichnis
rmdir: testVerzeichnis: Datei oder Verzeichnis nicht gefunden
```

## Das Verzeichnis wechseln – cd

`cd [Verzeichnis]`

- Wenn das Kommando `cd` (**C**hange **D**irectory) ohne Verzeichnis als Argument aufgerufen wird, wechselt `cd` in das Home-Verzeichnis des aktuellen Benutzers.
- Die Tilde `~` steht für das Homeverzeichnis und kann in jedem Befehl als Synonym für das eigene Home-Verzeichnis eingesetzt werden.

```
user@rechner:/tmp$ cd ~  
user@rechner:~$ pwd  
/home/user
```

- Jedes Verzeichnis hat die Einträge `.` und `..`.
  - ⇒ `.` verweist auf das aktuelle Verzeichnis.
  - ⇒ `..` verweist auf das übergeordnete Verzeichnis (Vaterverzeichnis).

## Einige einfache Beispiele zu Verzeichniswechseln

```
user@rechner:~/SYS1$ cd .
```

```
user@rechner:~/SYS1$ cd ..
```

```
user@rechner:~$ pwd  
/home/user
```

```
user@rechner:~/SYS1$ cd /
```

```
user@rechner:/$ pwd  
/
```

```
user@rechner:/$ cd /usr/local/
```

```
user@rechner:/usr/local$ pwd  
/usr/local
```

## Den Inhalt eines Verzeichnis ausgeben – `ls`

`ls [Option] ... [Datei] ...`

- Mit `ls` (List) kann der Inhalt von Verzeichnissen ausgegeben werden.
- Das Kommando kennt sehr viele Optionen. Einige ausgewählte sind:
  - a    *Alle* Einträge im Verzeichnis. Auch die Geheimen.
  - l    Eine *Liste* mit Benutzer- und Gruppenrechten, Dateigrößen und Datum ausgeben.
  - R    Auch den Inhalt der Unterverzeichnisse (*rekursiv*) ausgeben.
  - h    Größenangaben in menschenlesbarem (*human readable*) Format. z.B. 23M
  - B    Einträge, die mit ~ enden (*backups*) ignorieren.
  - d    Nur Verzeichnisse (*directories*) ausgeben. Keine Dateien.
  - t    Sortiert nach dem Zeitpunkt (*time*) der letzten Änderung.
  - S    Sortiert nach der Dateigröße (*size*).



## Weitere Optionen von `ls`

- s    Ausgabe der Dateigrößen in Blöcken.
  - m    alle Einträge durch Kommata getrennt in einer Zeile ausgeben.
  - r    Umgekehrte Reihenfolge (*reverse*) beim Sortieren.
  - g    Wie -l, aber den Besitzer nicht auflisten.
  - G    Wie -l, aber die *Gruppe* nicht auflisten.
- 
- Um die Lesbarkeit zu erhöhen, können unterschiedliche Arten von Dateien und Verzeichnisse mit verschiedenen Farben markiert werden.
- 
- color=always    Verwendet immer Farben.
  - color=none      Verwendet nie Farben.
- 
- Eine kurze Hilfe ausgeben:
- 
- help

## Einige einfache Beispiele zu Verzeichnissen

```
user@rechner:~$ pwd
/home/user
user@rechner:~$ mkdir SYS1
user@rechner:~$ cd SYS1/
user@rechner:~/SYS1$ pwd
/home/user/SYS1

user@rechner:~/SYS1$ ls -la
insgesamt 20
drwxr-xr-x  2 user user    48 2006-10-16 10:42 .
drwxr-xr-x 253 user user 20400 2006-10-16 10:42 ..

user@rechner:~/SYS1$ mkdir test
user@rechner:~/SYS1$ ls -a
.  ..  test
user@rechner:~/SYS1$ rmdir test
```

## Das Hilfesystem von Linux: Die Manualseiten (1)

- Alle Linux/UNIX-Systeme verfügen über ein ausgereiftes Hilfesystem: Die Manualseiten. Zu fast jedem Kommando gibt es in den Manualseiten eine ausführliche Beschreibung.
- Aufruf von Manualseiten  $\implies$  `man`
- Die Manpage zu `<Stichwort>` aufrufen:

```
$ man <Stichwort>
```

- Alle Manualseiten zu `<Stichwort>` hintereinander aufrufen:

```
$ man -a <Stichwort>
```

- Den kompletten Pfad der Manualseite zu `<Stichwort>` anzeigen:

```
$ man -w <Stichwort>
```

## Das Hilfesystem von Linux: Die Manualseiten (2)

- In den Manualseiten nach <Stichwort> suchen und die gefundenen Manpages auflisten:

```
$ man -f <Stichwort>
```

⇒ Identisches Ergebnis mit dem Kommando: `whatis`

- In allen Kapiteln der Manualseiten nach <Stichwort> suchen und die gefundenen Manualseiten auflisten:

```
$ man -k <Stichwort>
```

⇒ Identisches Ergebnis mit dem Kommando: `apropos`

- Die Manualseite zu <Stichwort> aus <Kapitel> aufrufen. z.B. `man 1 man` ruft die Manualseite zum Befehl `man` aus dem Kapitel 1 auf:

```
man <Kapitel> <Stichwort>
```

## Kapitel der Manualseiten

- 1** Benutzerkommandos
- 2** Systemaufrufe. Funktionen, die vom Kernel bereitgestellt werden
- 3** C-Bibliotheksfunktionen. Funktionen innerhalb von Systembibliotheken
- 4** Gerätedateien (Devices). Üblicherweise im Verzeichnis /dev zu finden
- 5** Dateiformate, Protokolle und Konventionen. z.B. /etc/passwd
- 6** Spiele
- 7** Makropakete und Konventionen
- 8** Kommandos für die Systemadministration
- 9** Kernelroutinen (kein Standard!)

## Aufbau der Manualseiten

- Manualseiten besitzen üblicherweise folgende Abschnitte. Es gibt aber keinen festen Standard für Inhalt und Reihenfolge der Abschnitte, an den sich alle halten:
  - **NAME**: Kommandoname bzw. Dateiname und eine Kurzbeschreibung
  - **SYNOPSIS** / **SYNTAX**: Aufrufsyntax. Schema der Argumente, Optionen und Parameter
  - **DESCRIPTION**: Beschreibung
  - **FILES**: Dateien, die benötigt, erzeugt oder verändert werden
  - **SEE ALSO**: Querverweise zu anderen Dokumenten und verwandte Kommandos
  - **EXAMPLE**: Beispiele zur Verwendung
  - **AUTHOR**: Angaben zum Autor
  - **BUGS**: Bekanntes Fehlverhalten und Einschränkungen

## Unterschiedliche Dateiarnten unter Linux/UNIX (1)

- **Reguläre Dateien** (normale/einfache Dateien) mit einer Bitfolge als Inhalt.
- **Symbolische Links** sind Verweise auf eine bestehende Datei. Ein Symbolischer Link ist eine Datei, die nur den Dateinamen einer anderen Datei enthält.
- **Verzeichnisse** sind Dateien, die eine Liste mit Dateinamen enthalten.
- **Warteschlangen**, auch **Pipes** oder **FIFO-Dateien** genannt, sind feststehende Verbindungsleitungen zwischen verschiedenen Programmen bzw. Prozessen.

## Unterschiedliche Dateiarten unter Linux/UNIX (2)

- **Gerätedateien** ermöglichen den Zugriff auf Hardwaregeräte oder andere Systemkomponenten und befinden sich im Verzeichnis /dev
  - **Blockorientierte Geräte** sind Repräsentanten von Hardware-Geräten, die nicht einzelne Zeichen verarbeiten, sondern ganze Blocks  
⇒ Festplatten, Disketten, usw.
  - **Zeichenorientierte Geräte** sind Repräsentanten von Hardware-Geräten, die nicht blockweise angesteuert werden sondern durch einzelne Bytes  
⇒ Serielle oder Parallele Schnittstellen, Soundkarten, usw.
  - **Socketorientierte Geräte** machen im Prinzip Netzwerkverbindungen im Dateisystem sichtbar (und nutzbar).  
⇒ Druckerwarteschlange, Syslogdaemon, usw.
  - **Virtuelle Gerätedateien** steuern kein reales Gerät an.  
⇒ /dev/null, /dev/zero, /dev/random, usw.



## Eine leere Datei anlegen – touch

- Mit dem Kommando touch wird eine leere Datei angelegt.

```
$ touch <Dateiname>
```

- Wird mit dem Kommando touch auf eine Datei zugegriffen, die bereits besteht, werden die Zugriffs- und Modifikationszeiten der Datei auf die aktuelle Zeit gesetzt.
- Mit der Option -t [[HH] JJ]MMTTSSmm] können Dateien auch beliebige Zugriffs- und Modifikationszeiten erhalten.

## Einfache Beispiele zu touch

```
user@rechner:~/SYS1$ touch test.txt
```

```
user@rechner:~/SYS1$ ls -la
```

```
insgesamt 20
```

```
drwxr-xr-x  2 user user    72 2006-10-17 08:42 .
```

```
drwxr-xr-x 253 user user 20400 2006-10-16 21:34 ..
```

```
-rw-r--r--  1 user user      0 2006-10-17 08:42 test.txt
```

```
user@rechner:~/SYS1$ touch -m -t 197905231215 test.txt
```

```
user@rechner:~/SYS1$ ls -la
```

```
insgesamt 20
```

```
drwxr-xr-x  2 user user    72 2006-10-17 08:42 .
```

```
drwxr-xr-x 253 user user 20400 2006-10-16 21:34 ..
```

```
-rw-r--r--  1 user user      0 1979-05-23 12:15 test.txt
```

## Dateien ausgeben und verknüpfen – cat

`cat [Option] ... [Datei] ...`

- Mit dem Kommando `cat` (*concatenate*) ist es möglich, den Inhalt von Dateien auszugeben und Dateien miteinander zu verknüpfen.
- Wenn `cat` beim Aufruf ohne Option eine Datei übergeben wird, wird der Inhalt der Datei in der Standardausgabe ausgegeben.
- Mit der Option `-n` (*number*) wird vor jeder von `cat` vor jeder Zeile die Zeilennummer ausgegeben.

<code>cat Datei</code>	Gibt den Inhalt von Datei aus.
<code>cat Datei1 Datei2 &gt; Alles</code>	Fügt mehrere Dateien zur Datei Alles zusammen.
<code>cat Datei1 &gt;&gt; Datei2</code>	Hängt Datei1 an Datei2 an.
<code>cat &gt; Datei</code>	Erzeugt eine neue Datei. Eingabe beenden: Strg-C
<code>cat &gt; Datei &lt;&lt; ENDE</code>	Erzeugt eine neue Datei. Eingabe beenden: ENDE

## Dateien rückwärts ausgeben und verknüpfen – tac

`tac [Option] ... [Datei] ...`

- Mit dem Kommando `tac` ist es möglich, den Inhalt von Dateien rückwärts auszugeben und Dateien miteinander zu verknüpfen.
- Wenn `tac` beim Aufruf ohne Option eine Datei übergeben wird, wird der Inhalt der Datei in der Standardausgabe rückwärts ausgegeben.

```
$ cat testdatei.txt  
Zeile 1  
Zeile 2  
Zeile 3  
Zeile 4  
Zeile 5
```

```
$ tac testdatei.txt  
Zeile 5  
Zeile 4  
Zeile 3  
Zeile 2  
Zeile 1
```

## Den Inhalt einer Datei anzeigen – more und less

`more [Option] ... [Datei] ...`

`less [Option] ... [Datei] ...`

- more und less sind **Pager**, mit denen man durch Dateien blättert.
- less hat eine höhere Funktionalität als more.

SPACE	eine Bildschirmseite weiter blättern.
RETURN	eine Zeile weiter blättern.
b	eine Bildschirmseite zurück blättern.
q	den Pager beenden.
=	gibt die aktuelle Zeilennummer aus (nur more).
:n	nächste Datei in der Dateiliste öffnen (nur less).
.p	vorherige Datei in der Dateiliste öffnen (nur less).
:x	erste Datei in der Dateiliste öffnen (nur less).

## Das Ende einer Datei anzeigen – tail

`tail [Option] ... [Datei]`

- Das Kommando `tail` gibt die **letzten** 10 Zeilen einer Datei auf der Standardausgabe aus.
- Einen anderen Wert als 10 Zeilen, kann man mit der Option `-n <Zeilen>` festlegen.

```
$ tail -n 5 dateiname
```

- Das Kommando eignet sich gut zum Verfolgen von Log-Dateien.
- Durch die Option `-f` (*follow*) werden neue Einträge ausgegeben. z.B.:

```
$ tail -f /var/log/messages
```

## Den Anfang einer Datei anzeigen – head

`head [Option] ... [Datei] ...`

- Das Kommando `head` gibt die **ersten** 10 Zeilen einer oder mehrerer Dateien auf der Standardausgabe aus.
- Einen anderen Wert als 10 Zeilen, kann man mit der Option `-n <Zeilen>` festlegen.

```
$ head -n 5 dateiname
```

- Werden dem Kommando mehrere Dateien übergeben, wird zur besseren Orientierung vor der Ausgabe jeder Datei eine Kopfzeile ausgegeben. Diese Kopfzeile kann mit der Option `-q` (*quiet*) unterdrückt werden.
- Mit der Option `-v` (*verbose*) wird auch bei nur einer Datei eine Kopfzeile ausgegeben.

## Dateien kopieren – cp

```
cp [Option] ... [Datei1] [Datei2]  
cp [Option] ... [Datei] ... [Verzeichnis]
```

- Das Kommando cp (*copy*) kopiert eine oder mehrere Dateien und Verzeichnisse.
- Kopiert ~/SYS1/Datei1.txt nach /tmp/Datei2.txt

```
$ cp ~/SYS1/Datei1.txt /tmp/Datei2.txt
```

- f Überschreibt existierende Zielformateien ohne Warnung (*force*).
- b Erzeugt eine Sicherheitskopie von Dateien, die sonst überschrieben würden (*backup*).
- i Verlangt vor dem überschreiben einer existierende Zielformateien eine Bestätigung.
- r Kopiert Dateien und Unterverzeichnisse samt Inhalt rekursiv.
- v Gibt den Namen jeder Datei aus, die kopiert wird (*verbose*).



## Dateien verschieben/umbenennen – mv

`mv [Option] ... [Quelle] ... [Ziel]`

- Das Kommando `mv` (*move*) verschiebt eine oder mehrere Dateien und Verzeichnisse bzw. benennt sie um
- Verschiebt `~/SYS1/Datei1.txt` nach `/tmp/Datei2.txt`

```
$ mv ~/SYS1/Datei1.txt /tmp/Datei2.txt
```

- f Überschreibt existierende Zielformateien ohne Warnung (*force*).
- b Erzeugt eine Sicherheitskopie von Dateien, die sonst überschrieben würden (*backup*).
- i Verlangt vor dem überschreiben einer existierende Zielformateien eine Bestätigung.
- v Gibt den Namen jeder Datei aus, die kopiert wird (*verbose*).
- u Verschiebt Dateien nur, wenn Sie neuer sind als die Gleichnamigen (*update*).

## Dateien löschen – rm

`rm [Option] ... [Datei] ...`

- Das Kommando `rm` (*remove*) löscht eine oder mehrere Dateien und Verzeichnisse.

- löscht `Datei1` und `Datei2`

```
$ rm Datei1 datei2
```

- löscht das Verzeichnis `TestVerzeichnis` und seinen Inhalt rekursiv

```
$ rm -rf TestVerzeichnis
```

- `-f` Löscht schreibgeschützte Dateien ohne ohne Warnung (*force*).
- `-r` Löscht Dateien und Unterverzeichnisse samt Inhalt rekursiv.
- `-i` Verlangt vor jeder Löschung eine Bestätigung (*interactive*).
- `-v` Gibt den Namen jeder Datei aus, die gelöscht wird (*verbose*).

## Einfache Beispiele zu rm

```
user@rechner:~/SYS1$ ls -la
insgesamt 20
drwxr-xr-x  2 user user    48 2007-03-22 12:24 .
drwxr-xr-x 259 user user 20120 2007-03-22 00:23 ..

user@rechner:~/SYS1$ mkdir testVerzeichnis

user@rechner:~/SYS1$ touch testVerzeichnis/testDatei

user@rechner:~/SYS1$ rm testVerzeichnis
rm: Entfernen von »testVerzeichnis« nicht möglich: Ist ein Verzeichnis

user@rechner:~/SYS1$ rmdir testVerzeichnis
rmdir: testVerzeichnis: Das Verzeichnis ist nicht leer

user@rechner:~/SYS1$ rm -rf testVerzeichnis

user@rechner:~/SYS1$ ls -la
insgesamt 20
drwxr-xr-x  2 user user    48 2007-03-22 12:24 .
drwxr-xr-x 259 user user 20120 2007-03-22 00:23 ..
```

Nächste Übung:  
**19.10.2007**