

Dieses Dokument und eventuelle Erweiterungen und Anhänge sind teilweise aus der Linuxfibel von Ole Vanhoefer (Copyright (c) 2000-2003). Dieses Dokument darf nur gemäß der Regeln und Bedingungen wie sie von der Open Publication Licence, Version v0.4, festgelegt werden, verteilt werden (die letzte Version ist gegenwärtig verfügbar unter <http://www.opencontent.org/openpub/>).

1	Grundlagen Unix .....	2
1.1	Was ist ein Kernel und was leistet er? .....	2
1.2	Hardwarevoraussetzungen .....	2
1.3	Das X-Window System .....	2
1.3.1	Windowmanager .....	3
1.3.2	KDE .....	3
1.3.3	KDE Kontrollzentrum .....	3
1.3.4	KDE Systemüberwachung (KSysGuard) .....	3
1.3.5	Der KDE Desktop .....	5
1.4	Die Konsole (Einführung) .....	6
1.4.1	Warum noch eine Konsole, wenn es KDE gibt? .....	6
1.4.2	Starten einer Shell .....	6
1.4.3	Linux-Konsole .....	6
1.4.4	X-Terminal .....	7
1.4.5	Erste Befehle .....	8
1.4.6	Erstellen und Löschen von Verzeichnissen und Dateien .....	13
1.4.7	Kopieren, verschieben und umbenennen von Dateien .....	15
1.4.8	Informationen und Hilfe .....	17
1.5	Die Shell I .....	19
1.5.1	Was ist eine Shell? .....	19
1.5.2	Die Bash .....	20
1.5.3	Arbeit mit Verzeichnissen .....	21
1.5.4	Der Unix Dateibaum .....	24
1.5.5	Arbeit mit Dateien und Verzeichnissen (erweitert) .....	28
1.5.6	Weitere grundlegende Befehle .....	33
1.6	Die Shell II .....	36
1.6.1	Automatische Kommandoergänzung .....	36
1.6.2	Gruppierung von Kommandos .....	36
1.6.3	Bedingte Ausführung .....	37
1.6.4	Substituierung von Kommandos .....	37
1.6.5	Variablen .....	37
1.6.6	Die History .....	41
1.6.7	Umleitungen .....	42
1.7	Hilfe im und zu Unix .....	43
1.7.1	Lokale Hilfequellen .....	43
1.7.2	Befehle für Hilfe .....	46
1.8	Textfilter .....	47
1.8.1	Ausgabe ganzer Dateien .....	47
1.8.2	Textformatierung .....	49
1.8.3	Teilen von Texten .....	50
1.8.4	Textstatistik .....	52
1.8.5	Sortieren .....	52
1.8.6	Zeilenoperationen .....	53
1.8.7	Suchen und Ersetzen .....	54
1.9	Rechte im Unix Dateisystem .....	59
1.9.1	Rechte auf Dateien .....	59
1.9.2	Rechte auf Verzeichnisse .....	59
1.9.3	Spezielle Rechte suid und sgid .....	62
1.9.4	Verwalten und Setzen von Rechten .....	62
1.10	Installation SuSE Linux .....	64
1.10.1	Booten von CDROM .....	64
1.10.2	Konfiguration mit Yast2 .....	65
1.10.3	Benutzer- und Gruppenverwaltung .....	69

# 1 Grundlagen Unix

## 1.1 Was ist ein Kernel und was leistet er?

Der aktuelle Kernel verfügt über folgenden Leistungsmerkmale:

- **POSIX-Konformität:** Der POSIX-Standard (*Portable Operating System Interface*) definiert eine Reihe von Bedingungen für portierbare Betriebssysteme.
- **Multitasking:** Mehrere Prozesse und damit Programme können (scheinbar) gleichzeitig ausgeführt werden.
- **Multibuser:** Mehrere Benutzer können gleichzeitig (z. B. über Telnet-Programme) auf dem Rechner arbeiten.
- **Paging:** Das Paging erlaubt die Auslagerung von Speicherinformationen auf die Festplatte, so daß der Eindruck eines wesentlich größeren Arbeitsspeichers entsteht.
- **Shared Libraries:** Die Shared Libraries sind Programm-Bibliotheken. Bei Bedarf werden sie in den Speicher geladen und mehrere Programme gleichzeitig können auf die enthaltenen Routinen zugreifen. Dies spart Speicherplatz.
- **Shared Memory:** Normalerweise werden beim Multitasking-Betrieb jedem Programm eigene exklusive Speicherbereiche zugewiesen. Die Kommunikation erfolgt über die IPC (*InterProcess Communication*). Um diesen umständlichen Weg zu umgehen, können gemeinsam genutzte Speicherbereiche definiert werden, in denen die Programme ihre Daten austauschen können.
- **Symetric Multi Processing (SMP):** Beim Multitasking bekommen die Prozesse kleine Zeitscheiben zugewiesen, in denen sie ausgeführt werden. Da die Zeitscheiben schnell aufeinanderfolgen sieht es so aus, als ob die Programme gleichzeitig ausgeführt werden. Dies ist aber wirklich nur beim SMP der Fall. Linux ist in der Lage mehrere Prozesse auch auf mehreren Prozessoren gleichzeitig ausführen zu lassen.
- **Protected Mode:** Speicherschutzmechanismen des Prozessors verhindern, daß ein Prozeß auf den Speicherbereich eines anderen Prozesses zugreifen kann. Dadurch erscheint die von Windows noch bekannte, allseits beliebte allgemeine Schutzverletzung bei Linux nicht mehr.

## 1.2 Hardwarevoraussetzungen

Für ein System ohne graphische Oberfläche können Sie sich an folgenden Werten orientieren.

Theoretisches Minimum	386 SX	1 MB	RAM	5MB	Festplatte
Brauchbares Minimum	386 DX/40 mit Coprozessor	8 MB	RAM	150 MB	Festplatte
Brauchbares System	486 DX/66	16 MB	RAM	600MB	Festplatte
Gutes System	Pentium90	32 MB	RAM	1GB	Festplatte

Graphische Systeme machen erst ab einem 486 Prozessor Spaß und für ein KDE-System sollten es minimal 64 MB Arbeitsspeicher sein.

## 1.3 Das X-Window System

- Standardinstallation von SuSE-Linux in der oben genannten Konfiguration enthält das X-Window-System mit dem KDE

- X-Window-System kann durch die Tools sax bzw. sax2 konfiguriert werden. Einstellungen über Graphikkarte, Monitor, Auflösung, Tastatur und Maus erfolgen hier.
- für Bedienung und Arbeit am X-Window-System wird ein Window-Manager benötigt.
- er kümmert sich in erster Linie darum, wie die Programmfenster am Bildschirm dargestellt und bedient werden.
- er stattet die Fenster mit einer Titelleiste und Schaltflächen (Buttons, Menüs etc) aus und hilft bei deren Verwaltung (Verschieben, Größe ändern, Verkleinern zu Icon, Programmwechsel).
- die meisten Windowmanager stellen Menüs für die gängigen Programme bereit.
- diese elementaren Funktionen werden nicht vom X-Window-System zur Verfügung gestellt.
- Der Vorteil dieser Methode ist die große Flexibilität bei der Gestaltung der Oberfläche und damit der Benutzerschnittstelle.
- Nachteil, daß jeder Anwender einen anderen Windowmanager benutzt und damit keine einheitliche Oberfläche existiert.
- Ein Problem, daß vor allem die Anfänger betrifft
- aktuelle SuSE-Distributionen starten automatisch das X-Window-System für die graphische Anmeldung.
- wird das System mit der Textkonsole gestartet, kann von der Textkonsole aus das X-Window-System mit dem Befehl startx gestartet werden.
- Sollte das X-Window-System sich aufhängen, kann es mit der Tastenkombination <STRG> + <ALT> + <BACKSPACE> beendet werden.

### 1.3.1 Windowmanager

### 1.3.2 KDE

- KDE (K Desktop Enviroment) ist eine Sammlung von X-Programmen mit einem Window-Manager für das X-Window-System.
- stellt wie auch die anderen Windowmanager (z. B. Gnome oder fvwm) die Schnittstelle zwischen Benutzer und dem X-Window-System dar.
- KDE umfaßt dabei nicht nur die Konfigurationstools sondern auch eine Vielzahl von Standardprogrammen.

Noch einige Hinweise zu KDE.

- Doppelklicks sind in KDE verpönt und sind nur in Ausnahmefällen erforderlich.
- Mit der rechten Maustaste können Sie in den meisten KDE-Programmen ein Kontextmenü aufrufen.
- Fast alle KDE-Programme kommen mit Drag & Drop zurecht.
- KDE-Programme können in der Regel direkt über einen Menüpunkt konfiguriert werden. Bei einigen Programmen erfolgt die Konfiguration über das KDE-Kontrollzentrum.

### 1.3.3 KDE Kontrollzentrum

- KDE-Kontrollzentrum (KControl) ist das zentrale Konfigurationsprogramm fürs KDE.
- Eigenschaften wie Sprache, Desktopdesign, Sound u. a. können hier eingestellt werden.

### 1.3.4 KDE Systemüberwachung (KSysGuard)

- die KDE-Systemüberwachung zeigt Ihnen Informationen über das System und die darauf laufenden Prozesse an.
- sie können im KDE-Menü über *KDE/System/Systemüberwachung* das Programm starten. Alternativ geben Sie über *KDE/Befehl ausführen* den Befehl ksysguard ein.
- präsentiert sich nach dem Start mit einem fast leeren Feld links und vier Graphen rechts.
- warten Sie einen Moment und Sie sehen langsam von rechts die aktuellen Werte des Systems in die Graphen wandern.

#### 1.3.4.1 CPU Load

- Der Graph zeigt die Auslastung der CPU an.

- Drei Kurven können sie erkennen.
- Blau ist die Auslastung des Prozessors durch Benutzeraktionen, Rot die Auslastung durch Systemprozesse und Hellbraun die Auslastung durch Programme, die mit einer veränderten Priorität gestartet wurden.

#### 1.3.4.2 Load Average

- auf eine Minute gemittelte Auslastung des Systems wird in diesem Graphen dargestellt.
- da die CPU einem Programm immer die volle Leistung zur Verfügung stellt wenn nichts anderes anliegt, kommt es immer wieder zu CPU-Auslastungen von 100%.
- dies können aber auch nur einzelne Peaks sein.
- die über eine Minute gemittelte Auslastung gibt mehr Aufschlüsse über die Dauerbelastung eines Rechners.

#### 1.3.4.3 Physical Memory

- Auslastung des physikalischen Speichers ist ebenso wie die CPU-Leistung ein Kriterium für die Belastung des Rechners.
- Auf den ersten Blick scheint es so, als ob der Speicher, selbst wenn kaum Programme auf dem System laufen, völlig belegt ist.
- Das stimmt auch soweit, da der Kernel den freien Speicher als Cache für Laufwerkszugriffe verwendet.
- Es werden nämlich auch hier drei Graphen angezeigt.
- Blau zeigt den Speicherbedarf der Programme an, dunkles Orange den Bedarf des Puffers, und Hellbraun ist die Farbe für den Festplattencache.

#### 1.3.4.4 Swap Memory

- Neben dem physikalischen Speicher gibt es noch den Swap-Speicher, der als Auslagerungsdatei auf einem Datenträger realisiert ist.
- Beide zusammen ergeben den virtuellen Speicher, der dem System zur Verfügung steht.
- Hier heißt es genau zu beobachten. Denn wenn sich auf diesem Graphen viel tut, sollte man überlegen, ob nicht etwas mehr physikalischer Speicher dem Rechner gut tun würde.

#### 1.3.4.5 Sensor Browser

- Im linken Fenster finden Sie für den lokalen Rechner die möglichen Sensoren, deren Werte Sie in den rechten Graphen anzeigen können.
- Einfach den Sensor anklicken und ihn in den entsprechenden Graphen ziehen.
- Einen Graphen können Sie übrigens löschen, indem Sie mit der rechten Maustaste darauf klicken und *Anzeige entfernen* betätigen.

Über die genaue Bedeutung der einzelnen Anzeige möchte ich jetzt nicht eingehen, da dies den Umfang dieses Kapitels sprengen würde.

#### 1.3.4.6 Prozesstabelle

- Neben der Systemauslastung können Sie sich auch die laufenden Prozesse (Programm bzw. Programmteile) anzeigen lassen.
- In einer Tabelle werden für die Prozesse der Name, die Prozessidentifikationsnummer (PID), die Systemauslastung durch Benutzer und das System, die Priorität des Prozesses, der Speicherbedarf, der ausführende Benutzer und der aufrufende Befehl angezeigt.
- Die Bedeutung der einzelnen Spalten ergibt sich im Laufe dieses Skriptes. Hier an dieser Stelle seien Sie nur erwähnt.

Sie können auch eine Auswahl der anzuzeigenden Prozesse treffen. Sie können sich entweder *Alle Prozesse*, die laufen, sich anzeigen lassen oder sich auf die *Systemprozesse* beschränken. Auch können Sie sich nur mit den Benutzer abgeben und sich die Benutzerprozesse anzeigen. Wenn Sie ganz genügsam sind, begnügen Sie sich mit *Eigene Prozesse*.

Wenn Sie wissen wollen, welcher Prozess durch welchen Prozess gestartet wurde, dann wählen Sie die *Baum-Ansicht*. Den neuesten Stand der Prozessliste können Sie mit dem Button *Neu Aufbauen* erzeugen. Der wichtigste Button ist wohl *Beenden (kill)*, der durch seinen zweiten martialischen Namen schon anzeigt was er macht: Er tötet Prozesse. So können Prozesse, die sich sonst nicht mehr beenden lassen, gestoppt werden.

Wenn Sie ganz schnell an die Prozessliste herankommen wollen, dann betätigen Sie einfach die Tasten <STRG>+<ESC>. Es öffnet sich der KSysGuard mit der Ansicht der Prozeßtafel. Nun können Sie schnell einen Prozess töten oder sich über das Verhalten der Prozesse informieren.

### 1.3.5 Der KDE Desktop

#### 1.3.5.1 Wie lege ich ein Programmicon auf dem Desktop an?

- Am einfachsten können Sie ein Icon anlegen, wenn das Programm schon einen Eintrag im KDE-Menü besitzt.
- Mit der linken Maustaste auf den Punkt klicken und mit gedrückter Maustaste auf den Desktop ziehen.
- Dort dann im nun erscheinenden Menü auf *An diese Stelle kopieren* klicken. Fertig.

Wollen Sie ein Programm auf dem Desktop verewigen, was nicht im Menü erscheint, dann müssen Sie wie folgt vorgehen.

1. Rechte Maustaste auf den Desktop
2. Menüpunkt *Neu erstellen/Verknüpfung mit Programm ...* anklicken
3. Im Reiter *Allgemein* den Namen der Verknüpfung eingeben
4. Durch Klicken auf das Zahnrad-Symbol können Sie ein Symbol für das Programm auswählen
5. Im Register *Ausführen* geben Sie den Namen der Programmdatei an. Der Button *Durchsuchen* ermöglicht Ihnen das Dateisystem nach der Datei zu durchsuchen. Die meisten X-Window Programme finden Sie im Verzeichnis `/usr/X11R6/bin`, `/opt` und `/opt/kde3/bin`.
6. Mit *OK* bestätigen und fertig ist das Icon.

#### 1.3.5.2 KDE Programme

Inzwischen gibt es eine Reihe von Programmen, die speziell für das KDE geschrieben worden sind. Sie basieren auf den KDE-Bibliotheken und benötigen diese zwingend.

##### 1.3.5.2.1 Konqueror

Der Konqueror ist nicht nur der mit dem KDE mitgelieferte Browser sondern auch gleichzeitig der Dateimanager. Da er relativ neu ist, hat er noch längst nicht alle Internet-Standards (besonders JavaScript) völlig implantiert.

##### 1.3.5.2.2 KEdit und KWrite

Auch an Editoren wurde beim KDE gedacht. KEdit und KWrite sind hierfür entwickelt worden. KEdit ist die einfache Version des Editors, während KWrite neben Syntaxhighlighting auch Suchen und Ersetzen mit regulären Ausdrücken beherrscht.

##### 1.3.5.2.3 KMail

Das Programm KMail ist, wie der Name es schon sagt, der Mail-Client vom KDE.

##### 1.3.5.2.4 KSnapshot

Für Dokumentationen ist es oft sinnvoll einen Bildschirmschnappschuß machen zu können. Diese Funktion wird durch das Programm KSnapshot zur Verfügung gestellt.

##### 1.3.5.2.5 KWord

KWord ist die Textverarbeitung.

**Bearbeiten Sie alle Aufgaben unter „Erste Schritte mit KDE“!**

## 1.4 Die Konsole (Einführung)

Das KDE und damit auch das X-Window-System sind nur auf Linux aufgesetzt und dienen als Benutzeroberfläche. Es gibt aber auch eine wesentlich rudimentärere Möglichkeit für den Benutzer mit Linux zu arbeiten: die Shell (*engl. Muschel*). Andere Begriffe für die Shell sind Kommandozeile, Textmodus oder Prompt.

### 1.4.1 Warum noch eine Konsole, wenn es KDE gibt?

KDE ist nur auf sehr wenigen und komfortablen Unix-Systemen verfügbar. Server z.B. werden generell ohne X-Windows bedient. Eine Arbeit auf entfernten Rechnern findet auch nur über die Konsole statt!

### 1.4.2 Starten einer Shell

Die Shell steht auf jedem Linux-System zur Verfügung. Es gibt zwei Möglichkeiten mit der Shell zu arbeiten. Unter dem X-Window-System gibt es sogenannte X-Terminals, die eine solche Shell emulieren. Daneben steht die sogenannte Linux-Konsole zur Verfügung, die im Textmodus der Graphikkarte arbeitet und auch ohne X-Window-System funktioniert.

### 1.4.3 Linux-Konsole

- Die erste Linux-Konsole erreichen Sie aus dem X-Window-System, indem Sie die Tasten <STRG>+<ALT>+<F1> gleichzeitig drücken.
- Unter SuSE ist diese Konsole oftmals sehr bunt und mit einem Hintergrund versehen.
- Dies ist der Bildschirm, den Sie auch beim Booten des Systems gesehen haben. Die letzten Meldungen des Bootvorgangs stehen noch dort.

```
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB (tty1)
```

- Mit dieser Meldung, die Ihnen die Distribution und das eingesetzte Betriebssystem mit Versionsnummer und Kernelversion anzeigt, begrüßt Sie das System.
- Danach steht der Login-Prompt mit Rechnername und wartet darauf, daß Sie sich an dieser Konsole anmelden.

```
archimedes login:
```

- Geben Sie nun Ihren Benutzernamen ein und schließen Sie die Eingabe mit der <Return>-Taste oder <Enter> ab.
- Sie werden nun nach Ihrem Passwort gefragt.
- Dies geben Sie ein und schließen die Eingabe wieder mit <Return> oder <Enter> ab.
- Wundern Sie sich nicht, denn Linux zeigt bei der Eingabe eines Passworts generell nichts an, noch nicht einmal die bei Windows üblichen Sternchen.

```
archimedes login: htor
```

```
Password:
```

```
Have a lot of fun ...
```

```
htor@archimedes:~>
```

- Ein netter Spruch wird ausgegeben und der Prompt erscheint.
- Der Prompt zeigt an, daß die Shell bereit ist von Ihnen ein Kommando zu bekommen.
- Wenn Sie SuSE benutzen, stehen Ihnen sechs von diesen Konsolen zur Verfügung.
- Mit den Tasten <ALT>+<F1> bis <ALT>+<F6> können Sie zwischen den Konsolen wechseln.
- Sie sehen, daß die Konsolen 2 bis 6 einen schwarzen Hintergrund besitzen und die gleiche Begrüßungsmeldung enthalten.
- Dabei weist das tty in der Klammer immer auf die jeweilige Konsole hin.
- Wechseln Sie wieder zur ersten Konsole (<ALT>+<F1>).
- Geben Sie dort am Prompt tty ein und schließen Sie die Eingabe mit <Return> oder <Enter> ab.
- Der Befehl zeigt Ihnen nun an, auf welcher Konsole sie sich befinden.

```
htor@archimedes:~> tty
```

```
/dev/tty1
```

- Um die Shell zu beenden, geben Sie das Kommando logout ein.
- Denken Sie daran, daß jedes Kommando erst dann ausgeführt wird, wenn Sie es mit <Return> oder <Enter> abschließen.

```
htor@archimedes:~> logout
```

```
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB (tty1)
```

```
defiant login:
```

- Nun ist die Konsole wieder zur Anmeldung bereit.
- Jetzt müssen Sie nur noch zum X-Window-System zurückkehren.
- Das X-Window-System ist praktisch die siebente Konsole. Sie können also mit <ALT>+<F7> dorthin gelangen.
- Wenn Sie schon auf der Konsole sind, können Sie alle Konsolen bzw. Terminals durch die Kombination von <ALT> und den Funktionstasten <F1> bis <F6> erreichen.
- Da die <ALT>-Taste unter KDE bzw. X-Window eine andere Bedeutung hat, können Sie hier die Konsolen über eine Kombination von <STRG>+<ALT> und den Funktionstasten <F1> bis <F6> erreichen.

#### 1.4.4 X-Terminal

- Für den Benutzer eines X-Window-Systems gibt es auch eine andere Methode um auf der Shell arbeiten zu können.
- X-Terminals emulieren die Linux-Konsole und bringen ein Textfenster auf den Bildschirm.
- Zum Starten des beim KDE mitgelieferten X-Terminals konsolle brauchen Sie in der Menüleiste nur auf den Monitor mit der Muschel klicken.
- An diesem Terminal brauchen Sie sich nicht anzumelden, da Sie sich durch die graphische Oberfläche schon in einer angemeldeten Sitzung befinden.
- Es erscheint also sofort der Prompt. Testen Sie doch mal, was der Befehl `tty` hier für ein Ergebnis zeigt.

```
htor@archimedes:~> tty
/dev/pts/1
```

Es gibt eine ganze Reihe von X-Terminals mit z. T. sehr unterschiedlichen Fähigkeiten.

##### 1.4.4.1 xterm

- `xterm` ist der Klassiker unter den X-Terminals.
- Er ist sehr einfach.
- Sie können ihn starten, in dem Sie den Menüpunkt *Befehl ausführen ...* aus dem KDE-Menü wählen oder mit den Tasten <ALT>+<F2> das Ausführungsfenster aufrufen.
- Dort können Sie durch Eingabe von `xterm` das Fenster starten.
- Mit dem Befehl `exit` können Sie das `xterm` dann wieder verlassen oder Sie drücken auf den X-Button oben rechts in der Titelleiste.

##### 1.4.4.2 konsole

- Dieses Terminal ist Bestandteil des KDE-Desktops.
- Genau wie das `xterm` kann es durch den Menüpunkt *Befehl ausführen ...* durch Eingabe von `konsole` gestartet werden.
- Im Gegensatz zu `xterm` kann `konsole` einiges mehr.
- So können z. B. mehrere unterschiedliche Konsolen in einem Fenster geöffnet werden.
- Es stehen dann auch spezielle Versionen zur Verfügung, die gleich dem Midnight-Commander, einen textbasierenden Dateimanager, starten oder als Benutzer *root* vorgeben.
- Eine weitere Konsole öffnen Sie über den Button unten links mit der Beschriftung *Neu*.
- Sie können natürlich auch über die Menüleiste über das Menü *Sitzung* gehen.
- Unter dem Menü *Bearbeiten* können Sie u. a. den Verlauf bearbeiten.
- Der Verlauf ist die Sammlung der Zeilen, die auf der Konsole gestanden haben.
- Natürlich werden nicht alle Zeilen gespeichert.
- Die Anzahl der zu speichernden Zeilen finden Sie unter *Einstellungen/Verlaufspeicher*. Voreingestellt sind 1000 Zeilen, was auch erst einmal reichen sollte.

- Sehr nützlich ist es, daß Sie den Verlauf auch nach einem Stichwort durchsuchen können.
- Im Menü *Ansicht* können Sie die laufende Konsole (Sitzung) umbenennen und auf Aktivität oder Nichtaktivität überwachen lassen.
- Sie können Ihre Eingaben auch zu mehreren Sitzungen gleichzeitig senden und die Reihenfolge der Sitzungen in der Sitzungsleiste ändern.
- Das Menü *Einstellungen* erlaubt individuelle Einstellungen für die Konsole.
- Schriftgröße, Signale, Tastatur, Farben und Größe können hier eingestellt werden.
- Unter *Hilfe* finden Sie, wie bei eigentlich jedem KDE-Programm Anleitungen und Informationen über das Programm.

#### 1.4.5 Erste Befehle

- Um die ersten Befehle auszuprobieren, starten Sie jetzt einfach mal die Konsole.
- Der Prompt erscheint und die Shell ist bereit Ihre ersten Kommandos entgegenzunehmen.
- Es gibt eine Reihe von Shells für Unix und Linux-Systeme.
- Die im Linux-Bereich am meisten verwendete Shell ist die Bash.
- Auch bei SuSE ist sie für den Benutzer voreingestellt.
- Der Prompt zeigt uns schon ein paar wichtige Informationen.
- Welcher Benutzer ist auf welchem Rechner eingeloggt und in welchem Verzeichnis befindet er sich gerade.
- Das Verzeichnis `~` ist kein Verzeichnis. Es ist ein Alias oder Synonym für das Heimatverzeichnis des jeweiligen Benutzers.
- Das können wir auch schnell mal vergleichen.
- Geben Sie doch einfach mal hinter dem Prompt den Befehl `pwd` ein.

```
htor@archimedes:~> pwd
/home/htor
htor@archimedes:~>
```

- Bei dem Benutzer *htor* steht das `~` für sein Heimatverzeichnis `/home/htor`.
- Die Heimatverzeichnisse der Benutzer heißen normalerweise wie ihre Besitzer und liegen im Verzeichnis `/home`.
- Es ist aber auch möglich die Heimatverzeichnisse anders zu benennen.

##### 1.4.5.1 Anzeigen des Verzeichnisinhaltes

- Was ist denn in diesem Heimatverzeichnis an Dateien vorhanden?
- Auch hierfür müssen Sie einen Befehl eingeben.
- Das Kommando `ls` zeigt Ihnen den Inhalt des Verzeichnisses an.

```
htor@archimedes:~> ls
Documents public_html
htor@archimedes:~>
```

- Zwei Dateien liegen bei einem frisch eingerichteten Benutzer bei SuSE 8.0 im Heimatverzeichnis.
- Haben Sie auf der graphischen Oberfläche schon gearbeitet, so können hier noch mehr Dateien liegen.

##### 1.4.5.2 Ausführliche Informationen

- Die Informationen, die der Befehl `ls` gibt, sind sehr spärlich.
- Wir können aber dem Kommando `ls` sagen, daß es mehr Informationen ausgeben soll.
- Dies erfolgt über eine *Option*.
- Andere Namen dafür sind *Schalter* oder *Attribut*.
- Die Option für eine ausführliche Ansicht heißt `-l`.
- Das Minus vor dem Buchstaben `l` (wie `long`) zeigt an, daß es sich um eine Option handelt.

```
htor@archimedes:~> ls -l
insgesamt 8
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 Documents
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 public_html
htor@archimedes:~>
```

- Wenn Sie sich nun die Ausgabe anschauen, dann wird Ihnen sicherlich nicht die Bedeutung jeder dieser Spalten klar sein.

- Schauen wir uns dazu die Ausgabe noch einmal genauer an.

#### 1.4.5.2.1 Typ und Rechte

- stehen in der ersten Spalte der Ausgabe
- Die in dieser Spalte stehende Kombination aus Buchstaben und Minus-Zeichen gibt Auskunft über den Dateityp und die Rechte, die für diese Datei gelten.
- Das erste Zeichen steht für den Dateityp.
- Das d in dem Beispiel steht für ein Verzeichnis.
- Würde ein Minuszeichen dort stehen, dann handelt es sich um eine normale Datei.
- Die Tabelle unten zeigt ihnen die anderen möglichen Typen.
- Auf die Bedeutung der einzelnen Typen möchte ich hier nicht eingehen.

Zeichen	Bedeutung
-	gewöhnliche Datei
d	Verzeichnis (directory)
c	zeichenorientierte Gerätedatei (character device)
b	blockorientierte Gerätedatei (block device)
p	FIFO-Pipeline(named pipe)
l	symbolischer Link (symbolic link)
s	Netzwerksocket (socket)

- Die restlichen neun Zeichen geben in Gruppen zu jeweils drei Zeichen die Rechte auf die Datei an.
- Dabei steht das r für „Lesen“ (read), das w für „Schreiben“ (write) und das x für „Ausführen“ (execute).
- die ersten drei Zeichen geben die Rechte für den Besitzer der Datei an.
- die zweiten drei Zeichen geben Auskunft über die Rechte der Gruppe, die dieser Datei zugeordnet wurde
- die letzten drei Zeichen geben dann an, was der Rest der Welt mit dieser Datei darf.

#### 1.4.5.2.2 Links

- Im Gegensatz zu Windows kann eine Datei bei Linux mehrere Namen besitzen.
- Bei Verzeichnissen sind das mindestens zwei.
- Einmal der Name selbst und dann im Verzeichnis die Datei mit dem Namen „.“
- Eine Datei wird erst dann gelöscht, wenn auch der letzte Name, bzw. Hardlink, wie die Dateinamen auch genannt werden, entfernt wurde.

#### 1.4.5.2.3 Besitzer

- Hier wird der Besitzer der Datei angegeben.
- Wenn ein Benutzer eine neue Datei anlegt, wird er automatisch der Besitzer der Datei.

#### 1.4.5.2.4 Gruppe

- Jeder Datei wird eine Gruppe von Benutzern zugeteilt,
- diesen koennen extra Rechte zugeordnet werden.
- Ist ein Benutzer in dieser Gruppe, so gelten die Rechte der Gruppe für ihn, wenn er nicht gerade der Besitzer der Datei ist.

#### 1.4.5.2.5 Größe

- Die Größe der Datei wird in dieser Spalte in Byte angegeben.

#### 1.4.5.2.6 Änderungsdatum

- Normalerweise zeigt der Befehl `ls` an dieser Stelle das Datum der letzten Änderung der Datei an.
- Durch bestimmte Attribute kann der Befehl auch dazu bewegt werden das Datum der Erstellung oder das Datum des letzten Zugriffs anzuzeigen.

#### 1.4.5.2.7 Name

- Hier steht der Name der Datei.
- Die Sortierung der aufgelisteten Dateien erfolgt lexikalisch.
- Beachten Sie dabei, daß die Shell zwischen Groß- und Kleinschreibung unterscheidet und die Großbuchstaben vor den Kleinbuchstaben einsortiert werden!

#### 1.4.5.3 Versteckte Dateien

- Wie Sie oben gesehen haben, enthält das Heimatverzeichnis scheinbar nur zwei Dateien bzw. Verzeichnisse.
- Das dem nicht so ist, zeigt der Einsatz der Option `-a`.

```
htor@archimedes:~> ls -a
.                  .Xresources      .exrc             .urlview          .xserverrc.secure
public_html
..                 .bashrc          .kermrc           .xcoralrc         .xsession
.Xdefaults        .dvipsrc         .muttrc           .xim               .xtalkrc
.Xmodmap          .emacs           .profile          .xinitrc          Documents
htor@archimedes:~>
```

- Anstatt zwei werden nun 22 Dateien angezeigt.
- Diese neu angezeigten Dateien bezeichnet man als *Versteckte Dateien*.
- Sie werden bei normalen Dateioperationen nicht berücksichtigt.
- Diese versteckten Dateien finden Sie auch in anderen Betriebssystemen.
- Dort wird durch ein Attribut der Datei angezeigt, ob eine Datei versteckt ist oder nicht.
- Wenn Sie sich die Liste der Dateien im obigen Beispiel noch einmal genau anschauen, finden Sie recht schnell die Gemeinsamkeit der versteckten Dateien heraus.

**Dateien, deren Name mit einem Punkt beginnt, bezeichnet man als Versteckte Dateien. Sie werden bei normalen Dateioperationen nicht berücksichtigt.**

- Eine besondere Bedeutung haben die Dateien „.“ und „..“.
- Sie sind Bestandteil des Verzeichnisses.
- Der Name „.“ ist ein weiterer Link für das Verzeichnis in dem sich diese Datei befindet.
- Dagegen verweist der Name „..“ auf das Elternverzeichnis des aktuellen Verzeichnisses.
- Wenn Sie das obige Beispiel als Grundlage nehmen, so ist „.“ gleichzusetzen mit `/home/htor` und „..“ mit `/home`.
- Zwei Optionen können auch miteinander kombiniert werden.
- Eine ausführliche Liste aller Dateien bekommen Sie durch die Kombination der Optionen `-l` und `-a`.

```
htor@archimedes:~> ls -a -l
insgesamt 96
drwxr-xr-x  4 htor      users      4096 2003-03-08 17:27 .
drwxr-xr-x  4 root      root       4096 2003-03-08 17:23 ..
-rw-r--r--  1 htor      users      5742 2003-03-08 17:27 .Xdefaults
-rw-r--r--  1 htor      users      1305 2003-03-08 17:27 .Xmodmap
-rw-r--r--  1 htor      users      5742 2003-03-08 17:27 .Xresources
-rw-r--r--  1 htor      users      1286 2003-03-08 17:27 .bashrc
-rw-r--r--  1 htor      users       208 2003-03-08 17:27 .dvipsrc
-rw-r--r--  1 htor      users     1637 2003-03-08 17:27 .emacs
-rw-r--r--  1 htor      users     1124 2003-03-08 17:27 .exrc
-rw-r--r--  1 htor      users      164 2003-03-08 17:27 .kermrc
-rw-r--r--  1 htor      users     2286 2003-03-08 17:27 .muttrc
-rw-r--r--  1 htor      users      934 2003-03-08 17:27 .profile
```

```

-rw-r--r--  1 htor  users      311  2003-03-08 17:27 .urlview
-rw-r--r--  1 htor  users     7913  2003-03-08 17:27 .xcoralrc
-rw-r--r--  1 htor  users     3618  2003-03-08 17:27 .xim
-rwxr-xr-x  1 htor  users     2432  2003-03-08 17:27 .xinitrc
-rw-r--r--  1 htor  users     1101  2003-03-08 17:27
.xserverrc.secure
-rwxr-xr-x  1 htor  users     2794  2003-03-08 17:27 .xsession
-rw-r--r--  1 htor  users      119  2003-03-08 17:27 .xtalkrc
drwxr-xr-x  2 htor  users     4096  2002-12-25 12:17 Documents
drwxr-xr-x  2 htor  users     4096  2002-12-25 12:17 public_html
htor@archimedes:~>

```

- Wer es kürzer mag, kann die Buchstaben auch zusammenziehen.
- Dabei ist es völlig egal, ob Sie -al oder -la schreiben.

```

htor@archimedes:~> ls -la
insgesamt 96
drwxr-xr-x  4 htor  users      4096  2003-03-08 17:27 .
drwxr-xr-x  4 root  root       4096  2003-03-08 17:23 ..
-rw-r--r--  1 htor  users     5742  2003-03-08 17:27 .Xdefaults
-rw-r--r--  1 htor  users     1305  2003-03-08 17:27 .Xmodmap
...

```

#### 1.4.5.4 Parameter und Jokerzeichen

- Der ls Befehl kann nicht nur den Inhalt es aktuellen Verzeichnis anzeigen.
- Wenn Sie z. B. den Inhalt des Verzeichnis Documents sehen wollen, geben Sie folgendes an.

```

htor@archimedes:~> ls -la Documents/
insgesamt 12
drwxr-xr-x  2 htor  users      4096  2002-12-25 12:17 .
drwxr-xr-x  4 htor  users      4096  2003-03-08 17:27 ..
-rw-r--r--  1 htor  users     1106  2002-03-04 18:30 .directory
htor@archimedes:~>

```

- Der Name des Verzeichnis Documents bezeichnet man als Parameter des Befehls ls.
- Die Angabe des Verzeichnisnamens erfolgte hier relativ zum Arbeitsverzeichnis.
- Um das Elternverzeichnis des Arbeitsverzeichnis anzuzeigen können wir den Dateinamen .. verwenden.

```

htor@archimedes:~> ls ..
conny htor monique
htor@archimedes:~>

```

- Sie können die Verzeichnisse auch direkt adressieren, wenn Sie den Verzeichnisnamen mit einem Slash / beginnen lassen.
- Jetzt wird nämlich nicht vom aktuellen Arbeitsverzeichnis aus gesehen, sondern von der Dateiwurzel.

```

htor@archimedes:~> ls /
bin  data1  etc  home  media  perikles  sbin  usr  windrives
boot ddrom  floppy  lib  mnt  proc  srv  var
cdrom dev  gloppy  lost+found  opt  root  tmp  vmware
htor@archimedes:~>

```

```

htor@archimedes:~> ls /home/
conny htor monique
htor@archimedes:~>

```

```

htor@archimedes:~> ls /home/htor
Documents public_html
htor@archimedes:~>

```

- Neben den Verzeichnissen können Sie auch eine Auswahl unter den Dateien treffen, die Sie sich anzeigen lassen wollen.
- Hilfreich sind dabei die sogenannten *Jokerzeichen* oder *Wildcards*.
- Dazu schauen Sie sich doch mal das Verzeichnis `/etc` an, in dem sich die Konfigurationsdateien des Linux-Systems befinden.

```
htor@archimedes:~> ls /etc/
DIR_COLORS      gshadow          lvmtab           qt.fontguess
HOSTNAME        gshadow-         lvmtab.d        raw
SuSE-release    gshadow.YaST2save magic             rc.d
SuSEconfig      gtk              mail.rc          rc.d.README
...
```

- Es werden sehr viele Dateien angezeigt.
- Sie können die Anzahl der angezeigten Dateien verringern, wenn Sie andere Kriterien für die Auswahl stellen.
- So könnten Sie sich nur die Dateien anzeigen lassen, die mit dem Buchstaben „a“ beginnen.

```
htor@archimedes:~> ls /etc/a*
/etc/a2ps-site.cfg /etc/adjtime /etc/aliases.db /etc/asound.state
/etc/a2ps.cfg      /etc/aliases /etc/allegrorc  /etc/at.deny
```

```
/etc/aliases.d:
```

```
/etc/alsa.d:
ad1816a cs4232 dt019x es18xx opl3sa2 sbawe ymfpci
als100 cs4236 emu10k1 fm801 sb16 sonicvibes
als4000 cs4281 es1688 gusextreme sb8 wavefront
```

```
htor@archimedes:~>
```

- Der Asterisk „\*“ steht für eine beliebige Anzahl beliebiger Zeichen und das Fragezeichen „?“ steht für *ein* beliebiges Zeichen.
- Auffällig an dem Beispiel oben ist, daß nicht nur der Name des Verzeichnis angegeben wird, sondern auch der Inhalt.
- Dies ist eine der Eigenschaften von `ls`. Ist ein Verzeichnis ein Ziel, dann wird der Inhalt des Verzeichnis angezeigt.
- Mit der Option `-d` kann dies unterbunden werden.

```
htor@archimedes:~> ls -d /etc/a*
/etc/a2ps-site.cfg /etc/aliases /etc/allegrorc /etc/at.deny
/etc/a2ps.cfg      /etc/aliases.d /etc/alsa.d
/etc/adjtime       /etc/aliases.db /etc/asound.state
```

```
htor@archimedes:~> ls -ld
drwxr-xr-x 4 htor users 4096 2003-03-08 17:27 .
```

```
htor@archimedes:~>
```

- Dies wirkt, wie Sie oben sehen, auch beim aktuellen Arbeitsverzeichnis.
- Es wird nur die Information über das aktuelle Verzeichnis angezeigt und nicht über die enthaltenen Dateien.

#### 1.4.5.5 Inodes

- Wie schon oben erwähnt, kann eine Datei mehrere Namen (Links) besitzen.
- Dies wird dadurch ermöglicht, daß die Benennung der Daten im System über eine sogenannte Inode-Nummer erfolgt.
- Als Inode bezeichnet man den Platz, an dem die Informationen über die Datei gespeichert werden.

- Das Betriebssystem merkt sich also nicht den Namen der Datei sondern die Inode-Nummer. Die Namen sind nur für uns „dumme“ Benutzer, die sich mehr unter dem Namen „amaretti.rezept.txt“ vorstellen können, als unter einer Inode-Nummer „45312“.
- Die Verzeichnisse sind im Prinzip nichts anderes als Tabellen, die den Zusammenhang zwischen Dateinamen und der Inode-Nummer herstellen.
- Die Inode-Nummer können sie sich mit ls unter Verwendung der Option -i anzeigen lassen.

```
htor@archimedes:~> ls -i
164981 Documents      229422 public_html
htor@archimedes:~>
```

#### 1.4.5.6 Hilfe

- Es gibt noch viel mehr Optionen für ls, als die ich Ihnen gerade vorgestellt habe.
- Um eine kurze Übersicht über die Funktion und vorhandenen Optionen eines Befehls zu erhalten, können Sie den Shellbefehl mit der Option --help aufrufen.
- Mit dem doppelten Minuszeichen wird dem Befehl signalisiert, daß es sich nicht um die einzelnen Optionen -h, -e, -l und -p handelt, sondern um eine Option mit dem langen Namen help.

```
htor@archimedes:~> ls --help
Aufruf: ls [OPTION]... [DATEI]...
Auflistung von Informationen der DATEIen (Standardvorgabe ist das momentane Verzeichnis). Alphabetisches Sortieren der Einträge, falls weder -cftuSUX noch --sort angegeben.
```

Erforderliche Argumente für lange Optionen sind auch für kurze erforderlich.

-a, --all	Einträge, die mit . beginnen, nicht verstecken
-A, --almost-all	implizierte . und .. nicht anzeigen
--author	den Urheber jeder Datei ausgeben
-b, --escape	nicht-druckbarer Zeichen oktale ausgeben
...	

- der Hilfetext ist länger als die Konsolenhöhe.
- Auf der Konsole können Sie ohne Probleme scrollen (wenn es nicht mehr als 1000 Zeilen sind).
- Damit die Anzeige nicht durchscrollt, sondern seitenweise anzeigt, können Sie einen sogenannten Pager benutzen.
- Ein solcher Pager ist der Befehl more.
- Sie leiten einfach die Ausgabe des Befehls mit dem Zeichen | an den Pager weiter.

```
htor@archimedes:~> ls --help |more
htor@archimedes:~> ls /etc | more
htor@archimedes:~> ls -l / | more
htor@archimedes:~> ls /dev | more
```

- Mit der Leertaste blättern Sie die Seiten des Pagers more durch.
- Erreicht der Pager das Ende des Textes bzw. der Ausgabe, so beendet er sich automatisch.

### **Bearbeiten Sie alle Aufgaben unter „Erste Schritte auf der Konsole“!**

#### 1.4.6 Erstellen und Löschen von Verzeichnissen und Dateien

- Ein Verzeichnissystem macht nur Sinn, wenn es Dateien und Verzeichnisse enthält.
- Fürs Erstellen und Löschen von Dateien und Verzeichnissen stehen auf der Shell viele Programme zur Verfügung.

##### 1.4.6.1 Erstellen

- Eine einfache Art und Weise eine leere Datei zu erstellen ist der Befehl touch.

```
htor@archimedes:~> touch test
htor@archimedes:~> ls -l
insgesamt 8
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 Documents
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 public_html
-rw-r--r--  1 htor  users    0 2003-03-08 18:49 test
htor@archimedes:~>
```

- Eigentlich soll der Befehl touch die Zeit der letzten Änderung und des letzten Zugriffs auf die aktuelle Zeit stellen.
- Existiert die angegebene Datei nicht, dann wird einfach eine neue Datei erstellt.
- Ein neues Verzeichnis zu erstellen ist ebenso einfach.
- Der Befehl heißt hier mkdir.

```
htor@archimedes:~> mkdir verzeichnis
htor@archimedes:~> ls -l
insgesamt 12
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 Documents
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 public_html
-rw-r--r--  1 htor  users    0 2003-03-08 18:49 test
drwxr-xr-x  2 htor  users  4096 2003-03-08 18:50 verzeichnis
htor@archimedes:~>
```

- Sie müssen sich beim Befehl mkdir, wie auch bei touch, nicht auf einen Parameter beschränken.
- Sie können ruhig mehrere Dateien bzw. Verzeichnisse angeben, die erstellt werden sollen.

```
htor@archimedes:~> touch datei1 datei2 datei3
htor@archimedes:~> ls
Documents datei1 datei2 datei3 public_html test verzeichnis
htor@archimedes:~>
```

- Auch eine Angabe des Verzeichnis mit seinen Elternverzeichnissen ist möglich.
- Allerdings müssen diese vorher existieren.

```
htor@archimedes:~> cd /
htor@archimedes:/> mkdir /home/htor/verzeichnis/verzeichnis2
htor@archimedes:/> ls /home/htor/verzeichnis
verzeichnis2
htor@archimedes:/>
```

### 1.4.6.2 Löschen

- Was man erstellt, muß man auch löschen können.
- Dies gilt für Verzeichnisse wie auch für Dateien.
- Der Befehl rmdir (*remove directory*) sorgt für das Löschen von Verzeichnissen.

```
htor@archimedes:/> cd
htor@archimedes:~> rmdir public_html/
htor@archimedes:~> rmdir verzeichnis/
rmdir: »verzeichnis/«: Das Verzeichnis ist nicht leer
htor@archimedes:~>
```

- Allerdings löscht rmdir nur leere Verzeichnisse.
- Sobald sich nur eine Datei oder ein Verzeichnis darin befindet, wird das Löschen des Verzeichnis verweigert.
- Dateien entledigt man sich dagegen durch den Befehl rm (*remove*).

```
htor@archimedes:~> rm test
```

```
htor@archimedes:~> rm verzeichnis/
rm: Entfernen von »verzeichnis/« nicht möglich: Ist ein Verzeichnis
htor@archimedes:~>
```

- Mit Verzeichnissen tut sich rm dagegen schwer und verweigert das Löschen mit einem entsprechenden Hinweis.
- Dies ist auch die normale Arbeitsweise der Shellkommandos.
- Klappt ein Befehl, so erscheint in der Regel kein Kommentar, sondern der Prompt fordert den Benutzer auf wieder tätig zu werden.
- Nur wenn Fehler aufgetreten sind, melden sich die Kommandos noch einmal beim Benutzer.
- Sie können aber rm überreden auch Verzeichnisse zu löschen.
- Mit der Option `-r` (*recursive*) löscht rm Verzeichnisse *mit* ihrem Inhalt.

```
htor@archimedes:~> rm -r verzeichnis/
htor@archimedes:~>
```

- Manchmal fragt rm auch nach, ob eine Datei gelöscht werden soll. Ist einem das zu lästig, gerade wenn ganze Verzeichnisstrukturen gelöscht werden sollen, dann sorgt der Schalter `-f` für Ruhe.
- Er sorgt dafür, daß die Dateien ohne Nachfrage gelöscht werden.

#### 1.4.7 Kopieren, verschieben und umbenennen von Dateien

Für das Kopieren, Verschieben und Umbenennen von Dateien brauchen wir hauptsächlich zwei Befehle: `cp` (*copy*) und `mv` (*move*).

##### 1.4.7.1 Kopieren

- Um eine Kopie einer Datei zu erstellen benutzen Sie den Befehl `cp`.
- Der Befehl braucht eine Datei, die es zu kopieren gilt, (Quelldatei) und den neuen Namen der Datei (Zieldatei).

```
htor@archimedes:~> ls
Documents datei1 datei2 datei3
htor@archimedes:~> cp datei1 kopie1
htor@archimedes:~> ls
Documents datei1 datei2 datei3 kopie1
htor@archimedes:~>
```

- So kann einfach eine Sicherungskopie einer Datei erstellt werden.
- In den meisten Fällen werden aber eine oder mehrere Dateien in ein anderes Verzeichnis kopiert, wie z. B. in `/floppy` für die Diskette.
- In diesem Fall wird anstatt der Zieldatei ein Zielverzeichnis angegeben.
- Erkennt der Befehl `cp`, daß das Ziel ein existierendes Verzeichnis ist, dann kopiert er die Datei ohne den Namen zu ändern in das Verzeichnis.
- Existiert kein Verzeichnis, so wird die Angabe als Dateiname verstanden und die Datei dementsprechend kopiert.
- Übrigens: `cp` überschreibt in der normalen Einstellung existierende Dateien gnadenlos.

```
htor@archimedes:~> mkdir backup
htor@archimedes:~> ls -l backup/
insgesamt 0
htor@archimedes:~> cp datei1 backup/
htor@archimedes:~> ls -l backup/
insgesamt 0
-rw-r--r--  1 htor  users  0 2003-03-08 19:00 datei1
htor@archimedes:~>
```

- Sie können auch mehrere Dateien zum Kopieren angeben, wenn Ihr Ziel ein Verzeichnis ist.

```
htor@archimedes:~> cp /etc/passwd datei2 /etc/
Display all 180 possibilities? (y or n)
htor@archimedes:~> cp /etc/passwd datei2 /etc/inputrc backup/
htor@archimedes:~> ls -l backup/
insgesamt 12
-rw-r--r--  1 htor  users      0 2003-03-08 19:00 datei1
-rw-r--r--  1 htor  users      0 2003-03-08 19:01 datei2
-rw-r--r--  1 htor  users    4774 2003-03-08 19:01 inputrc
-rw-r--r--  1 htor  users     921 2003-03-08 19:01 passwd
htor@archimedes:~>
```

- Wenn Sie das aktuelle Verzeichnis als Ziel angeben wollen, machen Sie das einfach durch den Punkt „.“.
- Genauso können Sie das Elternverzeichnis mit „..“ als Ziel angeben.
- Um zu sehen, welche Dateien kopiert worden sind, benutzen Sie den Befehl cp einfach mit dem Schalter -v (*verbose*).
- Diese auch als „Blubberschalter“ bekannte Option veranlaßt den Befehl dazu ausführlich über seine Tätigkeit zu berichten.
- Sie werden bei vielen Shell-Befehlen die Option -v wiederfinden.
- Sie können natürlich auch für die Liste der zu kopierenden Dateien die Jokerzeichen wie den Asterisk „\*“ verwenden.

```
htor@archimedes:~> cp -v /etc/f* backup/
»/etc/fam.conf« -> »backup/fam.conf«
»/etc/fb.modes« -> »backup/fb.modes«
»/etc/fdprm« -> »backup/fdprm«
»/etc/filesystems« -> »backup/filesystems«
»/etc/fstab« -> »backup/fstab«
»/etc/ftpusers« -> »backup/ftpusers«
htor@archimedes:~>
```

#### 1.4.7.2 Verschieben und umbenennen

- Der Befehl mv arbeitet im Prinzip genau wie der Befehl cp.
- Allerdings existiert die Quelldatei bzw. Quelldateien danach nicht mehr.
- Im Prinzip ist ein Verschieben und Umbenennen nichts anderes als eine Änderung des Verzeichniseintrags.
- Nur wenn zwischen zwei Partitionen verschoben werden soll, muß die Datei erst kopiert und dann die alte Datei gelöscht werden.
- Dadurch ist Verschieben im Normalfall schneller als Kopieren
- Um also eine Datei umzubennen benutze ich den Befehl mv unter Angabe von altem und neuem Namen.

```
htor@archimedes:~> mv kopie1 KOPIE1
htor@archimedes:~> ls -l
insgesamt 8
drwxr-xr-x  2 htor  users    4096 2002-12-25 12:17 Documents
-rw-r--r--  1 htor  users      0 2003-03-08 18:58 KOPIE1
drwxr-xr-x  2 htor  users    4096 2003-03-08 19:02 backup
-rw-r--r--  1 htor  users      0 2003-03-08 18:51 datei1
-rw-r--r--  1 htor  users      0 2003-03-08 18:51 datei2
-rw-r--r--  1 htor  users      0 2003-03-08 18:51 datei3
htor@archimedes:~>
```

- Um die Syntax eines Shell-Kommandos kurz und knapp darzustellen, werden Syntaxregeln mit einer bestimmten Symbolik verwendet.
- Der mv Befehl kann als Syntaxregel so ausgedrückt werden:
- mv ALTERNAME NEUERNAME
- Die groß geschriebenen Worte stehen für den Wert, der dort eingetragen werden muß.

- Ähnlich wie bei cp führt die Angabe eines existierenden Verzeichnis als Ziel dazu, daß die Datei nicht umbenannt, sondern in das Verzeichnis unter Beibehaltung des Namens verschoben wird.

```
htor@archimedes:~> ls -l
insgesamt 8
drwxr-xr-x  2 htor  users  4096 2002-12-25 12:17 Documents
drwxr-xr-x  2 htor  users  4096 2003-03-08 19:05 backup
-rw-r--r--  1 htor  users    0 2003-03-08 18:51 datei1
-rw-r--r--  1 htor  users    0 2003-03-08 18:51 datei2
-rw-r--r--  1 htor  users    0 2003-03-08 18:51 datei3
htor@archimedes:~> ls -l backup/
insgesamt 60
-rw-r--r--  1 htor  users    0 2003-03-08 18:58 KOPIE1
-rw-r--r--  1 htor  users    0 2003-03-08 19:00 datei1
-rw-r--r--  1 htor  users    0 2003-03-08 19:01 datei2
...
```

- Die Kommandosyntax kann durch folgende Regeln ausgedrückt werden:
- `mv DATEILISTE ZIELVERZEICHNIS`
- oder auch als
- `mv DATEI1 [DATEI2 [DATEI3 [...]]] ZIELVERZEICHNIS`
- Die eckigen Klammern zeigen an, daß der entsprechende Teil optional ist und nicht unbedingt notwendig.
- Auch bei mv gibt es den „Blubberschalter“ -v und noch weitere Optionen.
- Die Syntaxregel unter Berücksichtigung der Optionen lautet dann:
- `mv [-v] DATEILISTE ZIELVERZEICHNIS`
- oder auch
- `mv [OPTIONEN] DATEILISTE ZIELVERZEICHNIS`

#### 1.4.8 Informationen und Hilfe

Im Gegensatz zu der graphischen Oberfläche kann auf der Shell nicht einfach durch durchgucken der Menüpunkte der richtige Befehl gefunden werden. Linux stellt Informationen und Hilfen zur Verfügung.

##### 1.4.8.1 Hilfe im Programm

- Fast jeder Befehl und somit auch die Shell, die ja auch nichts anderes als ein Programm ist, stellt eine eigene Kurzhilfe zur Verfügung.
- In der Bash gibt es eingebaute Befehle, die nicht als eigenständige Programme vorliegen.
- Eine Übersicht der Befehle gibt der Befehl `help`.
- Genauere Informationen zu einem Befehl bekommen Sie dann durch Eingabe des Befehls hinter `help`.

```
htor@archimedes:~> help pwd
pwd: pwd [-PL]
Print the current working directory. With the -P option, pwd prints
the physical directory, without any symbolic links; the -L option
makes pwd follow symbolic links.
htor@archimedes:~>
```

- Bei den meisten anderen Befehlen kommen Sie mit dem Schalter `--help` weiter.
- Diese Option veranlaßt den Befehl dazu, eine kurze Erklärung seiner Funktion und eine Übersicht über seine Optionen zu geben.

```
htor@archimedes:~> touch --help
Aufruf: touch [OPTION]... DATEI...
Die Zugriffs- und Modifikationszeiten jeder DATEI auf die
momentane Zeit aktualisieren.
```

Erforderliche Argumente für lange Optionen sind auch für kurze erforderlich.

```
-a                Nur die Zugriffszeit ändern.
-c, --no-create  Keine Dateien erzeugen.
-d, --date=DATUM  DATUM lesen und statt der momentanen Zeit
verwenden.
-f                (ignoriert)
-m                Nur Modifikationszeit ändern.
-r, --reference=DATEI  Die Zeiten dieser Datei anstatt der momentanen
Zeit
                  verwenden.
-t MARKE         [[HH]JJ]MMTTSSmm[.ss] statt momentaner Zeit
verwenden.
  --time=WORT     Die Zeit, die von WORT angegeben wird, setzen:
                  access, atime, (wie -a), mtime, modify (wie -m).
--help          diese Hilfe anzeigen und beenden.
--version       Versionsinformation ausgeben und beenden.
```

Beachten Sie, dass -d und -t verschiedene Zeit-Datum-Formate akzeptieren.

Melden Sie Fehler (auf Englisch, mit LC\_ALL=C) an <bug-fileutils@gnu.org>. htor@archimedes:~>

- Andere Befehle, wie z. B. der Packer zip liefern diese Erklärungen schon, wenn Sie ohne Parameter aufgerufen werden.

```
htor@archimedes:~> zip
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
Zip 2.3 (November 29th 1999). Usage:
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi
list]
The default action is to add or replace zipfile entries from list, which
can include the special name - to compress standard input.
If zipfile and list are omitted, zip compresses stdin to stdout.
-f  freshen: only changed files  -u  update: only changed or new files
-d  delete entries in zipfile    -m  move into zipfile (delete files)
-r  recurse into directories     -j  junk (don't record) directory
names
-0  store only                   -l  convert LF to CR LF (-ll CR LF to
LF)
...
```

#### 1.4.8.2 Manual Pages (Hilfeseiten)

- Für weitergehende Erläuterungen wurden die Manual Pages, oder auch nach Ihrem Verwaltungsprogramm Man-Pages genannt, entwickelt.
- Die Man-Pages sind praktisch die Bedienungsanleitungen für die Shell-Befehle.
- Nach einem man cp wird das Dokument entpackt und dann mit dem passenden Pager angezeigt.
- Mit den Richtungstasten können Sie nun hin- und herscrollen.
- Die Betätigung der Taste <Q> beendet dann das Man-Programm.
- Die Programme whatis und apropos basieren auf man.
- whatis liefert eine Kurzbeschreibung für einen Befehl, während apropos diese Kurzbeschreibungen nach einem Begriff durchsucht.

```
htor@archimedes:~> apropos login
/etc/login.defs (5) [login.defs] - Login configuration
logname (1)          - print user's login name
nologin (8)         - politely refuse a login
sulogin (8)         - Single-user login
```

```

slogin (1) - OpenSSH SSH client (remote login program)
faillog (8) - examine faillog and set login failure limits
ssh (1) - OpenSSH SSH client (remote login program)
chsh (1) - change login shell
faillog (5) - Login failure logging file
rlogin (1) - remote login
login.defs (5) - Login configuration
login (1) - Begin session on the system
htor@archimedes:~>
htor@archimedes:~> whatis telnet
telnet (1) - user interface to the TELNET protocol
htor@archimedes:~>

```

- Die Manual-Pages enthalten nicht nur Informationen zu Befehlen, sondern auch zu Konfigurationsdateien, Kernelfunktionen u.a.

## 1.5 Die Shell I

### 1.5.1 Was ist eine Shell?

- Die Shell ist nur ein Linux-Programm.
- Sie dient als Verbindungsstelle zwischen dem Benutzer und dem Betriebssystem.
- Ihre Aufgabe ist es, die eingegebene Kommandozeile so umzuformen, daß das Betriebssystem sie interpretieren kann.
- Daher gibt es auch nicht nur eine Shell, sondern eine Anzahl von Programmen, die diese Vermittlungsaufgabe übernehmen können.
- Eine Übersicht über die verschiedenen Shells liefert folgende Tabelle

Tabelle: Übersicht über die häufigsten Shells

Shell	Built-In Befehle	Kommandozeilenoptionen
ash	24	10
bash (Bourne Again Shell)	48	12
ksh (Public Domain Korn Shell )	42	20
rcsh (Emulation der csh)	53	18
zsh	84	50

- Im Normalfall, wenn man sich auf einem Linux-Rechner einloggt, benutzt man die bash.
- Die Standard-Shell eines Benutzers wird durch einen Eintrag in der `/etc/passwd` festgelegt.
- Um eine andere Shell zu starten, geben sie einfach das Shellkommando mit seinem Pfad ein.
- Dies startet eine Kindprozess in dem die neue Shell läuft.
- Um die Shell zu beenden geben Sie `exit` ein.
- Im folgenden Beispiel wird durch die Eingabe `csh` zur *C-Shell* gewechselt.
- Dies können Sie an der veränderten Prompt-Einstellung sehen, da die C-Shell die Abkürzung `~` fürs Heimatverzeichnis nicht kennt.

```

htor@archimedes:~> csh
archimedes /home/htor>
archimedes /home/htor> exit
exit
htor@archimedes:~>

```

- Es gibt drei verschiedene Situationen, in denen eine Shell arbeiten kann.
- Als *interaktive Login-Shell*, als *interaktive Shell* und als *nichtinteraktive Shell*.

- Eine interaktive Login-Shell, meistens nur Login-Shell genannt, wird direkt nach dem Einloggen gestartet und Sie können direkt mit ihr Arbeiten.
- Eine interaktive Shell ermöglicht ebenfalls das direkte Arbeiten.
- Allerdings wurde sie nicht durch einen Login-Vorgang gestartet.
- Dies kann z. B. durch die Eingabe des Shellprogramms in einer anderen Shell geschehen, wie Sie im obigen Beispiel sehen konnten, oder z. B. durch das Starten einer Terminalemulation auf der graphischen Oberfläche.
- Mit einer nichtinteraktiven Shell können Sie nicht am Prompt arbeiten.
- Sie wird nur zum Ausführen eines Shell-Skripts gestartet und beendet sich nach dem Ende des Skripts automatisch.
- Damit dient diese Shell nur als eigenständige Umgebung für die Ausführung von vordefinierten Shell-Befehlen ohne Möglichkeiten für den Benutzer direkt einzugreifen.

### 1.5.1.1 Das Kommando chsh

- Das Kommando **chsh** ändert dauerhaft die Login-Shell.

```
chsh [-s LOGINSHELL] [USER]
```

- Der Benutzer kann nur die Shell für sich selbst ändern, während **root** das für jeden Benutzer machen kann.
- Dabei ist aber der Benutzer auf die Shells eingeschränkt, die in der Datei `/etc/shells` aufgelistet werden.
- Wird die Option `-s` nicht angegeben, so erfolgt die Änderung der Daten im interaktiven Modus.
- Vor jeder Änderung wird das Kennwort des Benutzers zur Sicherheit abgefragt.

### Optionen

`-s LOGINSHELL` Die neue Login-Shell

### 1.5.2 Die Bash

- Die erste programmierbare Shell wurde von Steve Bourne entwickelt
- sie war die erste UNIX-Shell.
- Als aufwärtskompatibler Nachfolger wurde später die Bourne-Again-Shell oder kurz Bash entwickelt.
- Sie wurde als Bestandteil des GNU-Betriebssystems entworfen, für das allerdings bisher nur der Compiler, Tools wie z.B. Editoren und die Shell existieren.
- Der wichtigste Teil des Betriebssystems, der Betriebssystemkern, fehlt allerdings noch.
- Da alle Teile freie Software sind, sind sie Bestandteile der freien UNIX-Varianten Linux und Free-BSD, die im Prinzip hauptsächlich aus dem Betriebssystemkern mit Treibern bestehen und die anderen Teile, die für ein funktionierendes System notwendig sind, wie eben Compiler, Shell, Editoren, Tools zum Dateihandling oder ähnliches, aus anderen Quellen beziehen.
- So ist die Bash die Standard-Shell von Linux, wobei auch hier dem Benutzer die Wahlfreiheit gelassen wird, und man jede andere UNIX-Shell verwenden kann.
- Im Prinzip ist die Bash nichts weiter als ein normales Programm, was die Kommandos des Benutzers an das Betriebssystem und andere Programme weiterleitet und deren Ausgaben wieder dem Benutzer zur Verfügung stellt.
- Als allererstes sollten Sie bei der Bash beachten, daß zwischen Groß- und Kleinschreibung peinlich genau unterschieden wird. So sind mit den Namen `testdatei.txt`, `TESTDATEI.TXT` und `TestDatei.TXT` im Gegensatz zu DOS/Windows drei verschiedene Dateien gemeint.
- Da die meisten Befehle auch nur ausführbare Dateien sind, gilt es natürlich auch für diese.
- Der Prompt zeigt nicht nur die Bereitschaft der Shell an ein neues Kommando zu empfangen
- Er kann auch mit Informationen ausgestattet werden.
- Bei SuSE wird voreingestellt der eingeloggte Benutzer, der Rechner und das Verzeichnis angezeigt.

```
htor@archimedes:/var/spool/mail>
```

- Unvollständige Befehlszeilen quittiert die Shell (Bourne und Bash) durch Ausgabe des Hilfsprompts
- In der Regel handelt es sich um ein Größer-Zeichen gefolgt von einem Leerzeichen.
- Häufigste Ursache dafür ist ein fehlendes zweites Anführungszeichen.
- Durch Eingabe dieses Zeichens wird die Befehlszeile vervollständigt und der Befehl kann ausgeführt werden.

```
htor@archimedes:~> echo "test
>
> "
test
```

- Häufig ist es nötig einen Befehl zu wiederholen oder einen Befehl leicht abzuwandeln.
- Unter Linux steht eine Eingabewiederholung ähnlich dem DOSKEY unter Windows zur Verfügung: die History.
- Diese wird im wesentlichen über die Cursortasten Auf und Ab gesteuert.
- Sollten Sie mal mit dem Platz in einer Zeile nicht auskommen, so kann eine Eingabezeile problemlos verlängert werden, indem man anstatt `<RETURN>` zu drücken unmittelbar vor `<RETURN>` einen Backslash `\` eingibt.
- Gerade am Anfang können Sie ungewollt in Programme raten, mit denen Sie sich nicht auskennen.
- Hier ein paar Tips, wie Sie solche Programme beenden können:
  - Die Taste `<Q>` oder `<Q><RETURN>` für „quit“ führen oft zum Erfolg. (z. B. less, more und top)
  - Die Kombination `<STRG>+<D>` steht für Ende-der-Datei. Damit können Sie vor allem Programme beenden, die auf eine Eingabe warten. (z. B. cat, wenn es von der Standardeingabe liest, und at). Sollten Sie sich aber direkt in der Shell befinden, dann werden Sie damit ausgeloggt.
  - Eine etwas härtere Methode ein Programm zu beenden ist die Kombination `<STRG>+<C>`. Hiermit wird das laufende Programm aufgefordert, sich sofort zu beenden.
  - Sollten Sie sich im Editor `vi` verfangen haben, kommen sie durch die Tastenfolge `<ESC><:;><Q><!><RETURN>` heraus.
  - Der Editor `joe` ist da viel leichter zu beenden. Er reagiert auf die Kombination `<STRG>+<C>` als normalen Ende-Befehl.

### 1.5.2.1 Kommandosyntax

- Ein Kommando ist eine Folge von Zeichenketten, die durch ein oder mehrere Leerzeichen oder Tabulatoren getrennt werden.
- Die ersten Zeichenkette legt den Namen des Kommandos fest.
- Die weiteren Zeichenketten werden als Parameter bezeichnet.
- `KOMMANDO [PARAMETER1] [PARAMETER2] ... [PARAMETERn]`
- Dabei gehören die Leerzeichen und Tabulatoren nicht zu den Parametern, sondern werden nur als Trennzeichen gewertet.
- Soll der Parameter hingegen Leerzeichen oder Tabulatoren enthalten, so muß der Ausdruck in einfache (') oder doppelte (") Anführungszeichen gesetzt werden.

Das Kommando legt fest **was** gemacht werden soll. Die Parameter bestimmen **womit** und die Optionen **wie**.

### 1.5.3 Arbeit mit Verzeichnissen

- Nach dem Einloggen befinden Sie sich immer in Ihrem Heimatverzeichnis.
- Dies wird generell abgekürzt durch das Tilde-Zeichen `~`.

- Die folgenden Arbeitsweisen des Tilde-Zeichens werden als Tilde-Ausdehnung (engl. *tilde expansion*) bezeichnet.

~           Heimatverzeichnis des aktuellen Benutzers  
 ~walter   Heimatverzeichnis von dem Benutzer *walter*  
 ~+         Das aktuelle Arbeitsverzeichnis  
 ~-         Das alte Arbeitsverzeichnis

Ihre Funktionsweise können Sie im folgenden Beispiel verfolgen.

```
htor@archimedes:~/backup> echo ~
/home/htor
htor@archimedes:~/backup> echo ~conny
~conny
htor@archimedes:~/backup> echo ~+
/home/htor/backup
htor@archimedes:~/backup> cd ../Documents/
htor@archimedes:~/Documents> echo ~-
/home/htor/backup
htor@archimedes:~/Documents>
```

### 1.5.3.1 Das Kommando pwd

- Der Befehl `pwd` (*Print Working Directory*) zeigt das Verzeichnis an, in dem Sie gerade arbeiten.

### 1.5.3.2 Das Kommando cd

- Um in ein anderes Verzeichnis zu wechseln benutzt man das Kommando `cd` (*Change Directory*).
- `cd [DIRECTORY]`
- Das Zielverzeichnis kann dabei auf absolute oder relative Weise angegeben werden.
- Bei der absoluten Darstellung geht man immer von der Wurzel (*root*) aus.
- Diese Pfadangaben beginnen immer mit einem Slash `/`.
- Bei der relativen Pfadangabe wird dagegen von dem aktuellen Verzeichnis als Startpunkt ausgegangen.
- Jedes Verzeichnis enthält zwei besondere Verzeichnisse.
- Das Verzeichnis `.` steht für das Verzeichnis selber, während `..` für das übergeordnete Verzeichnis (Elternverzeichnis) steht.
- Um wieder zurück ins Heimatverzeichnis zu kommen gibt es mehrere Wege.
- Sie können zum einen den absoluten oder relativen Pfad zum Verzeichnis angeben.
- Kürzer geht es aber mit den Befehlen `cd ~` oder ganz kurz nur mit `cd`.

### 1.5.3.3 Das Kommando ls

- Eine Reihe von Befehlen ermöglicht es, sich den Inhalt eines Verzeichnisses anzusehen.
- Der am häufigsten benutzte Befehl ist `ls` (*LiSt*).
- `ls [OPTIONEN] [DATEINAME]`
- Für `DATEINAME` können Namen von Dateien oder Verzeichnissen verwendet werden.
- Dabei kann, um eine Menge von Dateinamen zu bilden, Metazeichen (Joker) verwendet werden.
- Die bekanntesten Joker sind dabei das Fragezeichen `?`, das für ein einzelnes beliebiges Zeichen steht, und der Asterisk `*`, der für eine beliebige Anzahl beliebiger Zeichen steht.

## Optionen

- a           Anzeige aller Dateien, auch derjenigen, die mit `.` beginnen.
- A           Anzeige aller Dateien außer `.` und `..`
- c           Sortiert zusammen mit `-t` Dateien nach dem Datum der letzten Änderung der Inode (Verwaltungsinformationen). Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit `-l`.  
(`ls -ctl`)
- d           Verzeichnisse in der Liste der Argumente werden wie andere Dateien behandelt. Unterdrückung der Durchsuchung des Inhalts von Verzeichnissen.
- l           Anzeige ausführlicher Dateiinformationen. Viele Schalter haben nur im Zusammenhang mit `-l` eine Bedeutung, da sie nicht automatisch die Information anzeigen sondern nur vorbereiten (z. B. `-e` oder `-k`).
- r           Ausgabe in umgekehrter Sortierreihenfolge
- t           Sortiert Dateien nach dem Datum der letzten Änderung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des Datums erfolgt zusammen mit `-l`.
- x           Spaltenweise Ausgabe der Dateinamen, im Gegensatz zu `-C` jedoch waagrecht geordnet.
- m           Ausgabe der Dateinamen als durch Komma getrennte Liste.
- F           Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen: `/` für Verzeichnisse, `*` für ausführbare Dateien, `@` für Links, `|` für FIFOs und `=` für Sockets.
- R           Rekursive Anzeige, d. h. es werden nicht nur Verzeichnisse nach ihrem Inhalt durchsucht, sondern auch darin enthaltene (Unter-) Verzeichnisse. Nicht zusammen mit `-d` anwendbar.

- Es gibt eine Reihe von alternativen Optionen. Z. B. ist `--tabsize=zahl` eine Alternative zu `-T zahl`.
- Diese Optionen sind selbsterklärend (wenn man Englisch kann und Phantasie hat) und werden stets mit doppeltem Minuszeichen eingeleitet, da sie sonst wegen ihrer Länge mit Zusammenfassungen der herkömmliche Schalter verwechselt werden können.
- Man kann sie sich mit `ls --help` anzeigen lassen.

Über die Ausgabe des `ls`-Befehls sollten sie noch wissen:

- Die standardmäßig vorgegebene Reihenfolge der Ausgabe beruht auf einer lexikalischen Sortierung, wobei erst nach Ziffern, dann nach Großbuchstaben und zuletzt nach Kleinbuchstaben sortiert wird.
- Die Benutzung von `ls` ohne Schalter ergibt eine Liste in vertikal sortierten Spalten.
- Dateien, die mit einem Punkt `.` beginnen, werden als versteckte Dateien bezeichnet. Sie werden nur unter Verwendung der Schalter `-a` und `-A` angezeigt. Im Allgemeinen handelt es sich hier um Konfigurationsdateien.
- Die Farben bei der Ausgabe von `ls` werden in der Datei `/etc/DIR_COLORS` festgelegt. Um die Farben selbst bestimmen zu können, muß die Datei als `.dir_colors` in das Heimatverzeichnis kopiert werden.

#### 1.5.3.4 Das Kommando dir

- Der Befehl dir wird in den einzelnen Distributionen unterschiedlich behandelt.
- Er ist entweder ein Alias für `ls` oder für `ls -l`.

#### 1.5.3.5 Das Kommando mkdir

- Der Befehl mkdir wird dazu benutzt neue Verzeichnisse zu erstellen.
- `mkdir [OPTIONEN] [VERZEICHNISPFADLISTE]`
- Wird die Option `-p` nicht mit angegeben, so müssen die Elternverzeichnisse für das neue Verzeichnis existieren.

##### Optionen

- p Erzeugt auch die nötigen Elternverzeichnisse, wenn diese nicht existieren.
- m RECHTE Erlaubt gleich das Setzen der Rechte für das neue Verzeichnis wie durch `chmod`

#### 1.5.3.6 Das Kommando rmdir

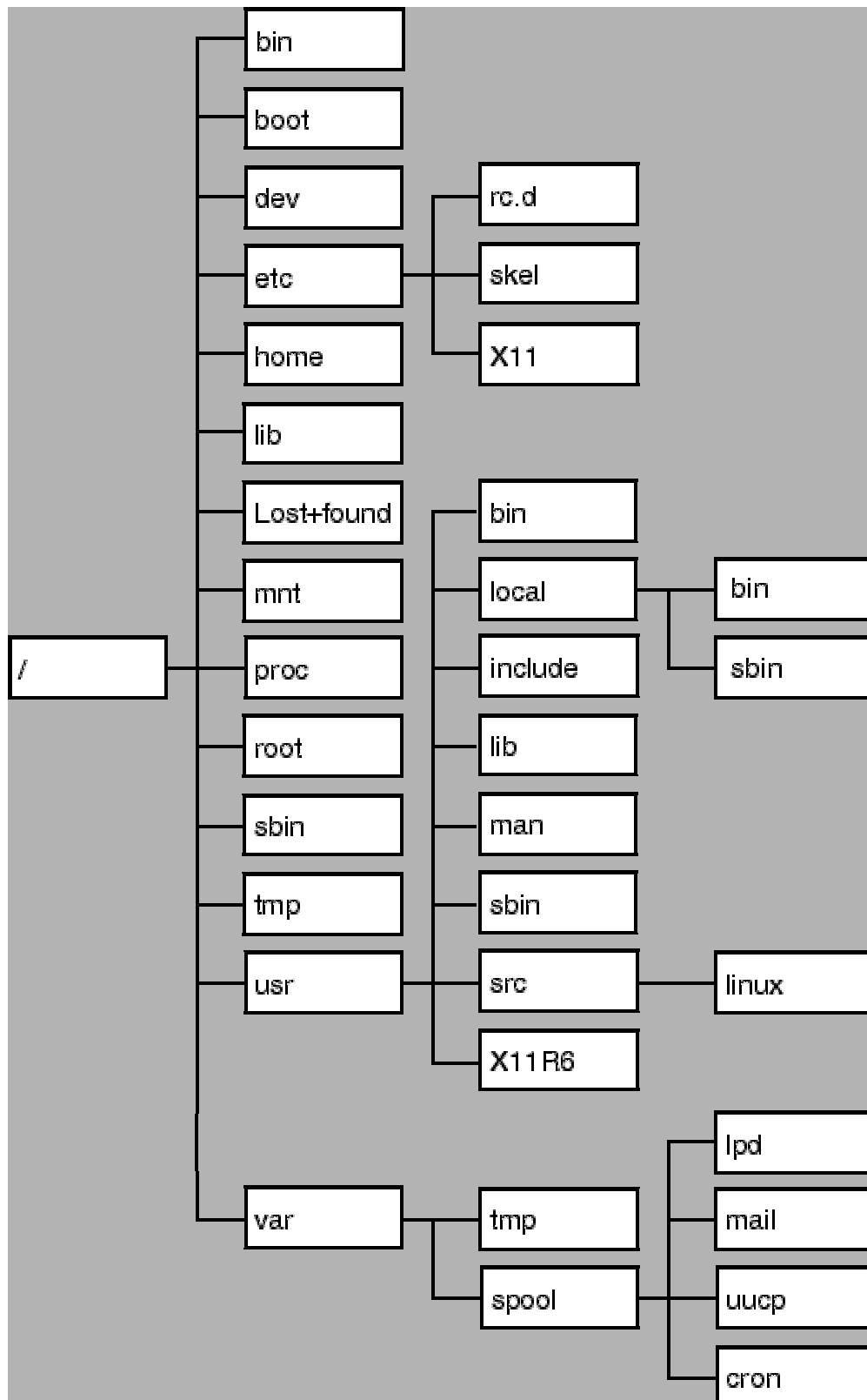
- Um leere Verzeichnisse wieder zu löschen, wird der Befehl rmdir verwendet.
- `rmdir [OPTIONEN] [VERZEICHNISPFADLISTE]`
- Dabei wird immer das letzte Verzeichnis in der angegebenen Verzeichnishierarchie gelöscht.
- Der Schalter `-p` ermöglicht das Löschen ganzer Hierarchien.

##### Optionen

- p Löschen aller leeren Verzeichnisse in der Verzeichnishierarchie.

#### 1.5.4 Der Unix Dateibaum

- Den Linux-Dateibaum gibt es eigentlich nicht.
- Jede Distribution hat einen individuellen Dateibaum für sich entwickelt.
- Allerdings gibt es einige Verzeichnisse, die eigentlich überall vorhanden sein sollten.
- Als Leitlinie für den Baum gilt der sogenannte Dateisystemstandard (*Filesystem-Hierarchy-Standard*) **FHS**.
- Eine Übersicht über die wichtigsten Verzeichnisse liefert folgende Abbildung.



- `/` Enthält alle anderen Verzeichnisse. Es ist die Wurzel des Dateisystems und wird auch Wurzel- (root-) verzeichnis genannt.
- `/bin` Enthält die wichtigsten, für alle Benutzer verfügbaren LINUX-Befehle.
- `/boot` Enthält den Kernel und für das Booten wichtige Dateien.

- `/dev` Das Verzeichnis für die Gerätedateien.
- `/etc` Konfigurations- und Informationsdateien, darunter `magic`, `DIR_COLORS`, `input_rc` u. v. m.
- `/etc/rc.d` Enthält die für den Bootvorgang wichtigen Skripte und die rc-Verzeichnisse.
- `/etc/skel` Aus diesem Verzeichnis werden die Konfigurationsdateien (*skeleton user files*) für einen neuen Benutzer in dessen Heimatverzeichnis kopiert.
- `/etc/X11` Die Konfigurationsdateien des X-Window-Systems liegen hier.
- `/home` beherbergt die Heimatverzeichnisse, in denen normalerweise keine Rechte die Arbeit des Benutzers einschränken. Es ist somit für den normalen Benutzer von existentieller Bedeutung. In größeren LINUX-Systemen sollte für dieses Verzeichnis entsprechender Platzbedarf einkalkuliert werden, da hier die Benutzerdaten abgelegt werden. Häufig wird es dann auf einen Extra-Datenträger ausgelagert, der beim Systemstart separat gemountet wird.
- `/lib` Hier findet man die Bibliothek des C-Compilers, sowie sogenannte Shared Libraries, Programmbibliotheken (vereinfacht gesagt Programmteile), die von verschiedenen Anwendungen benutzt werden können. Diese Programmteile werden erst zur Laufzeit in den Arbeitsspeicher geladen, wenn das betreffende Programm sie benötigt. Dadurch und durch die Tatsache, daß verschiedene Programme darauf zugreifen können (shared) wird der Arbeitsspeicher wesentlich entlastet. Sind Sie Windows-Benutzer kennen Sie sicherlich .DLL-Dateien. Diese werden unter Windows entsprechend genutzt.
- `/lost+found` Dieses Verzeichnis wird für wiederhergestellte Dateien verwendet.
- `/mnt` Hilfsverzeichnis zur Aufnahme von Dateisystemen, die in den LINUX-Verzeichnisbaum gemountet werden sollen.
- `/proc` ist ein virtuelles Dateisystem für Informationen über die laufenden Prozesse. Der Befehl `ps` (process status) liefert seine Angaben auf der Basis des Inhalts dieses Verzeichnisses. Jeder Prozeß erhält ein eigenes Unterverzeichnis in dem die Informationen über ihn gespeichert werden.
- `/root` ist das Heimatverzeichnis der Superusers (**root**).
- `/sbin` Hier liegen die Befehle zur Systemverwaltung wie `adduser`, mit dessen Hilfe neue Benutzer definiert werden, `fsck` zur Überprüfung von Dateisystemen oder `shutdown`, mit dem das System heruntergefahren werden kann. Programme für Systemstart wie `init`, mit dem diverse Systemdienste (z. B. Drucker- und Zeitdienste) initiiert werden, darunter auch `getty` (ebenfalls in `/sbin` zu finden). Über `getty` erfolgt das Anmelden an der Konsole oder an Terminals, die via serieller Leitung mit dem LINUX-Rechner verbunden sind.
- `/tmp` Verzeichnis zum Zwischenlagern von Dateien. Hier darf jeder schreiben und lesen.
- `/usr` Enthält alle wichtigen Programme und Daten wie etwa das Online-Manual, die nicht zum Booten des Systems benötigt werden. Es kann ein Verzeichnis sein, das sich physikalisch auf einem anderen Datenträger befindet als das Bootmedium, auf einer CD-ROM etwa oder an einem entfernten NFS-Server. In dieser Situation müßte dieser Teil des Dateisystems nach dem Booten erst verfügbar gemacht werden. Für den Fall, daß es dabei zu Problemen kommen sollte, stehen die allernötigsten Befehle in verschiedenen anderen Verzeichnissen, z. B. `/bin`, zur Verfügung.
- `/usr/bin` enthält die Befehle, die allen Teilnehmern zur Verfügung stehen sollten. Ein Teil dieser Benutzer-Befehle befindet sich aus den oben genannten Gründen (vgl. Eintrag bei `/usr`) in `/bin`.

- `/usr/local` wird für Programme verwendet, die nicht Bestandteil des Betriebssystems sind, wie z. B. das Dokumentenverzeichnis für den Webserver.
- `/usr/local/bin` Hier werden die Binärdateien von diesen Programmen gespeichert. Dieses Verzeichnis sollte im Suchpfad des Benutzers enthalten sein.
- `/usr/local/sbin` ist für lokal installierte Administrationstools zuständig.
- `/usr/include` enthält die Standard-C/C++-Header-Dateien.
- `/usr/lib` Die statischen Programmbibliotheken sowie die Unterverzeichnisse für Bibliotheken verschiedener Programmiersprachen sind hier zu finden. Enthält auch Links zu X-Window-Dateien.
- `/usr/man` ist eins der Verzeichnisse für das Online-Manual. Eine unerschöpfliche Informationsquelle für den Benutzer.
- `/usr/sbin` Analog zu `/usr/bin` gibt es dieses Verzeichnis für den Superuser (**root**).
- `/usr/src` enthält die Quellcodes für die Systemprogramme.
- `/usr/src/linux` In diesem Unterverzeichnis ist der Quellcode des Systemkerns zu finden.
- `/usr/X11R6` Enthält Dateien für das X-Window-System. Oft auch nur ein Link auf ein anderes Verzeichnis.
- `/var` Informationsdateien für das System sind hier abgelegt. Dies sind Tabellen, die von LINUX laufend verändert werden, z. B. die Dateien `utmp` und `wtmp`, die Informationen über die eingeloggten Benutzer enthalten. (Standard-UNIX: Die Tabellen befinden sich zum größten Teil in `/etc`.)
- `/var/tmp` ist das Ziel der temporären Dateien, die von Anwendungen aus verschiedenen Gründen während eines Programmlaufs angelegt und verändert, zum Programmende aber wieder gelöscht werden. Editoren oder Textverarbeitungen speichern unter Umständen den vom Anwender geschriebenen Text zur Sicherheit in einer temporären Datei. Temporäre Dateien sind kurzlebig und ständiger Änderung unterworfen. Sie sind ein sprechendes Beispiel für die allgemeine Bedeutung des Verzeichnisses `/var`.
- `/var/spool` ist der Ort der Warteschlangen (*Queues*) diverser Spoolprogramme. Dies sind Programme, die im Hintergrund arbeitend Aufträge der verschiedenen Benutzer entgegennehmen und in einer vernünftigen Reihenfolge abwickeln. Da mehrere Benutzer (beinahe) gleichzeitig z. B. einen Druckauftrag an den Druckspooler geben können, sprich drucken wollen, muß dieser dafür sorgen, daß die einzelnen Aufträge sich nicht vermischen und etwa am Drucker zwei Texte zu einem - bunt gemischten - werden. als Hilfsmittel dienen ihnen Warteschlangen, in die die einzelnen Aufträge nacheinander eingereiht und (nach dem Wirkungsprinzip einer FIFO) nach und nach abgearbeitet werden.
- `/var/spool/lpd` Warteschlangen des Druckerspoolers
- `/var/spool/mail` Warteschlange für die elektronische Post
- `/var/spool/uucp` Warteschlange für den Kommunikationsdienst uucp (unix-to-unix-copy)
- `/var/spool/cron` ist für die Abwicklung von Zeitdiensten über den cron-Dämonenprozeß wichtig, der in regelmäßigen Abständen Tabellen in `/var/spool/crontabs` daraufhin überprüft, ob sie Anweisungen enthalten, die zum gegenwärtigen Zeitpunkt auszuführen sind.

## 1.5.5 Arbeit mit Dateien und Verzeichnissen (erweitert)

### 1.5.5.1 Das Kommando touch

- Das Kommando `touch` ändert die Zeit des letzten Zugriffs und der letzten Änderung auf die aktuelle Zeit.
- Existiert die Datei nicht, so wird eine neue Datei erstellt.
- `touch [OPTIONEN] [DATEILISTE]`

### 1.5.5.2 Das Kommando cat

- Das Kommando `cat` (concatenate file) kann dazu benutzt werden neue Dateien zu erstellen.
- Es wird aber hauptsächlich dazu verwendet Dateien zusammenzufügen und sie auf dem Bildschirm oder einem anderen Gerät auszugeben.
- `cat [OPTIONEN] [DATEILISTE]`
- `cat` wird vor allem als Tool zum Anzeigen und Zusammenfügen von Dateien verwendet und als schneller Mini-Editor.

#### Optionen

-b	Nummeriert alle nichtleeren Zeilen durch
-e	Gleich mit -vE
-n	Nummeriert alle Zeilen durch (-b hat Vorrang)
-E	Fügt am Ende jeder Zeile ein \$ ein
-s	Faßt aufeinanderfolgende Leerzeilen zu einer Leerzeile zusammen
-v	Zeigt alle nichtdruckbaren Zeichen durch Metazeichen an
-T	Zeigt Tabulatoren als Í an
-A	Zeigt alle nichtdruckbaren Zeichen, Tabulatoren als Í und das Zeilenende als \$ an (vgl. -vET)

- **Beispiele:**

Fügt die Dateien `t1` und `t2` zusammen und schreibt sie in die Datei `t3`.

```
cat t1 t2 > t3
```

Ein einfacher Editor kann mit `cat` realisiert werden.

```
cat > text.txt
```

Dieser Befehl liest die Daten nicht aus einer Datei sondern von der Standardeingabe und schreibt sie dann in die Datei `text.txt`. Die Eingabe wird mit <CTRL>+<D> (EOF-Zeichen) beendet.

### 1.5.5.3 Das Kommando cp

- Um Kopien von einer Datei oder einem Verzeichnis zu erstellen wird das Kommando `cp` verwendet.
- `cp [OPTIONEN] QUELLDATEI ZIELDATEI`
- `cp [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS`
- Wird ein Dateiname für die Quelle angegeben, so kopiert `cp` die Datei in eine zweite Datei (`ZIELDATEI`).
- Ist der letzte Parameter ein Verzeichnis, dann wird die Datei in das Verzeichnis kopiert.
- Listen von Quelldateien können nur in ein Verzeichnis kopiert werden.
- Allerdings kann `cp` in der Normaleinstellung keine Verzeichnisse kopieren.

### Optionen

- a Kopiert die Dateien unter Beibehaltung von Struktur und Attributen
- b Erzeugt von jeder überschriebenen Datei eine Sicherheitskopie
- f Überschreiben von vorhandenen Zielf Dateien
- i Nachfragen vorm Überschreiben von vorhandenen Zielf Dateien
- l Legt Hardlinks anstatt Kopien der Dateien an
- s Legt Softlinks anstatt Kopien der Dateien an
- p Überträgt Besitzer, Gruppe, Rechte und Zeitmarken an die neue Datei
- P Kopieren der Dateien inklusiver ihrer Verzeichnisstruktur
- r Kopiert rekursiv Dateien aus Verzeichnissen
- R Wie -r, aber anstatt die Inhalte der Dateien zu kopieren, werden die Datei wie sie sind kopiert
- u Kopiert nur Dateien, die jünger sind als die Zielf Dateien
- v Zeigt die Namen der kopierten Dateien an

- **Beispiel:**

Die folgende Kommandosequenz kopiert alle Dateien des aktuellen Verzeichnis unter Beibehaltung ihrer Strukturen.

```
cp -dpRP * backup
```

#### 1.5.5.4 Das Kommando dd

- Das Programm *dd* (*Device to Device copy*) ist ein spezielles Kopierprogramm.
- Es wird in aller erster Linie dazu genutzt um Dateien von einem Gerät zu einem anderen zu kopieren.
- `dd [OPTIONEN]`

### Optionen

- `if=DATEI` Die Eingabedatei
- `of=DATEI` Die Ausgabedatei
- `bs=BLOCKGRÖSSE` Anzahl der Bytes, die auf einmal gelesen bzw. geschrieben werden.

#### 1.5.5.5 Das Kommando mv

- Um Dateien zu verschieben oder umzubenennen wird das Kommando *mv* verwendet.
- `mv [OPTIONEN] ALTERDATEINAME NEUERDATEINAME`
- `mv [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS`
- Werden zwei Dateinamen als Parameter vergeben, dann wird die Datei umbenannt.
- Ist der letzte Parameter ein Verzeichnis, dann werden die angegebenen Dateien in dieses Verzeichnis verschoben.

### Optionen

- b Erzeugt von jeder überschriebene Datei eine Sicherheitskopie
- f Überschreiben von vorhandenen Zieldateien
- i Nachfragen vorm Überschreiben von vorhandenen Zieldateien
- u Verschiebt nur Dateien, die jünger sind als die Zieldateien

- **Beispiele:**

Die folgende Kommandosequenz benennt die Datei `megabox.txt` in die Datei `MEGABOX.txt` um.  
`mv megabox.txt MEGABOX.txt`

Mit dem folgenden Kommando werden alle Dateien aus dem aktuellen Verzeichnis in das Verzeichnis `/home/harald` verschoben und eventuell vorhandene Zieldateien überschrieben.

```
mv -f * /home/harald
```

#### 1.5.5.6 Das Kommando rm

- Der Befehl `rm` löscht die angegebenen Dateien. In seiner Standardeinstellung löscht er keine Verzeichnisse.
- `rm [OPTIONEN] DATEILISTE`
- Um eine Datei zu löschen, benötigt man das Schreibrecht (`w`) auf das Verzeichnis, aber nicht auf die Datei.
- Sollte ein Datei nicht das Schreibrecht besitzen, so erbittet `rm` nur eine Bestätigung des Löschbefehls, wenn kein `-f` oder wenn ein `-i` gesetzt ist.

### Optionen

- d Löscht ein Verzeichnis durch Entfernen des Hardlinks. Volle Verzeichnisse werden auch gelöscht. Da die enthaltenen Dateien nicht mehr referenziert werden, ist es ratsam ein `fsck` danach auszuführen. (Nur Superuser)
- f Löscht alle Dateien ohne explizites Nachfragen. Überlagert die Option `-i`
- i Fragt vor jedem Löschen einer Datei um Bestätigung
- r Löscht Verzeichnisse und deren Inhalt rekursiv

- **Beispiele:**

Der folgende Befehl löscht alle Dateien, die mit `.temp` enden.

```
rm *.temp
```

#### 1.5.5.7 Dateibezeichnung mit Jokerzeichen

- Bei der Arbeit mit der Shell kommen Sie oft in Situationen, in denen Sie nicht nur einzelne Dateien oder alle Dateien in einem Verzeichnis ansprechen wollen, sondern nur eine Gruppe von Dateien (z. B. alle HTML-Dateien) bearbeiten möchten.
- Für diesen Zweck besitzt die Shell die sogenannten Jokerzeichen (*Wildcards*).
- Wenn diese Jokerzeichen eingesetzt werden, gibt die Shell nicht einen einzelnen Dateinamen zurück, sondern eine Liste von Dateinamen, auf die das Muster zutrifft.
- So erhalten Sie mit dem Muster `*.jpg` eine Liste aller Dateinamen, die mit `.jpg` enden.

```
htor@archimedes:~> ls *.jpg
bild1.jpg bild3.jpg bild5.jpg bild7.jpg bild9.jpg
bild2.jpg bild4.jpg bild6.jpg bild8.jpg
```

```
htor@archimedes:~> mv *.jpg bilder/
htor@archimedes:~> ls *.jpg
ls: *.jpg: Datei oder Verzeichnis nicht gefunden
htor@archimedes:~>
```

- Denken Sie daran.
- Die Shell und nicht der Befehl interpretiert das Kommando.
- Deshalb müssen die Jokerzeichen in manchen Fällen maskiert werden, damit sie unbeschadet an den Befehl übergeben werden können.
- um Jokerzeichen selbst zu übergeben muss ein „\“ vorangestellt werden
- Probieren Sie den unteren Befehl mal ohne Backslash aus.

```
htor@archimedes:~> echo * Hallo *
backup bilder datei1 datei2 datei3 Documents Hallo backup bilder datei1
datei2 datei3 Documents
htor@archimedes:~> echo \* Hallo \*
* Hallo *
htor@archimedes:~>
```

Eine Reihe von Jokerzeichen steht Ihnen zur Verfügung.

#### 1.5.5.8 Asterisk \*

- Der Asterisk bedeutet im Prinzip alles oder nichts.
- Er steht für eine beliebige Anzahl beliebiger Zeichen. Wobei beliebige Anzahl auch kein Zeichen bedeuten kann. Z. B. stimmt das Muster `lk*` mit folgenden Dateinamen überein.

```
htor@archimedes:~> ls dd*
dd_bild.pdf dd_dokument.pdf dd_klausur.tex
dd_bild.toc dd_hintergrund.tex dd_namen.tex
htor@archimedes:~>
```

#### 1.5.5.9 Fragezeichen ?

- Das Fragezeichen steht für *genau ein* beliebiges Zeichen.

```
htor@archimedes:~> ls lk*
lk.pdf lk.ps lk.tex lk.toc lk_neu.pdf lk_test.pdf
htor@archimedes:~> ls lk.???
lk.pdf lk.tex lk.toc
htor@archimedes:~> ls lk.??
lk.ps
htor@archimedes:~>
```

#### 1.5.5.10 Menge [ZEICHEN]

- Mit den eckigen Klammern ist es möglich eine Menge von Zeichen vorzugeben, die an dieser Stelle stehen können.
- So steht `[Aa]` für ein Zeichen, daß entweder ein großes oder ein kleines `A` sein kann.
- Ein Ausrufezeichen in der eckigen Klammer negiert die Bedeutung der Menge.
- So steht `[!aeiou]` für ein Zeichen, daß kein Vokal ist.
- Innerhalb der Klammern können nicht nur einzelne Zeichen angegeben werden, sondern auch Bereiche. So steht z. B. `[a-z]` für alle Kleinbuchstaben und `[0-9]` für alle Zahlen.

```
htor@archimedes:~> ls lk.???
lk.pdf lk.tex lk.toc
htor@archimedes:~> ls lk.[!t]??
lk.pdf
htor@archimedes:~> ls lk.?[eo]*
lk.tex lk.toc
htor@archimedes:~>
```

### 1.5.5.11 Klammerexpansion {WORT1,WORT2,WORT3,...}

- Die Klammerexpansion (*brace expansion*) ist im eigentlichen Sinne kein Jokerzeichen.
- Sie funktioniert nämlich auch ohne existierende Dateien.
- Die Funktion ist einfach.
- Der Ausdruck wird jeweils mit dem in den geschweiften Klammern angegebenen und durch Kommata getrennten Zeichenketten ausgegeben.

```
htor@archimedes:~> echo "ich bin "{gut,besser,super}""
ich bin gut. ich bin besser. ich bin super.
htor@archimedes:~> echo {1,2,3}{1,2,3}
11 12 13 21 22 23 31 32 33
htor@archimedes:~>
```

Trotzdem können Sie die Klammerexpansion natürlich auch auf Dateioperationen anwenden.

```
htor@archimedes:~> ls lk*
lk.pdf lk.ps lk.tex lk.toc lk_neu.pdf lk_test.pdf
htor@archimedes:~> ls lk.{toc,tex}
lk.tex lk.toc
htor@archimedes:~>
```

### 1.5.5.12 Das Kommando more

- Der Befehl `more` gehört zur Gruppe der Befehle, mit denen man sich den Inhalt von Dateien anschauen kann (*pager*).
- Im Gegensatz zu `cat` zeigt er die Daten aber seitenweise an.
- `more [OPTIONEN] DATEILISTE`

#### **Befehle**

<LEERTASTE>	Eine Seite nach unten scrollen
<f>	Eine Seite nach unten scrollen
<b>	Eine Seite nach oben scrollen
<RETURN>	Eine Zeile nach unten scrollen
<s>	Eine Zeile nach unten scrollen
</>MUSTER	Durchsuchen des Textes nach dem regulären Ausdruck MUSTER
<n>	Weitersuchen nach unten
<q>	Beenden

### 1.5.5.13 Das Kommando less

- Der Befehl `less` ist die Weiterentwicklung von `more`.
- `less [OPTIONEN] DATEILISTE`
- Neben den Möglichkeiten von `more` bietet `less` die Möglichkeit mit den Richtungstasten zu scrollen und mit Lesezeichen, Zeilennummern und prozentualen Textpositionen zu arbeiten. Auch wird `less` nicht am Ende der Datei beendet.

#### **Zusätzliche Befehle**

<j>	Eine Zeile nach unten scrollen
<k>	Eine Zeile nach oben scrollen
<g>	An den Anfang der Datei scrollen
<G>	Ans Ende der Datei scrollen

- </>MUSTER    Durchsuchen des Textes nach dem regulären Ausdruck  
MUSTER nach unten
- <?>MUSTER    Durchsuchen des Textes nach dem regulären Ausdruck  
MUSTER nach oben
- <n>                Weitersuchen nach unten
- <N>                Weitersuchen nach oben

#### 1.5.5.14 Das Kommando lesskey

- Das Programm lesskey erlaubt eine individuelle Konfiguration der Steuerbefehle von less.

#### 1.5.5.15 Das Kommando file

- Der Befehl file analysiert Dateien und gibt ihren Typ aus.
- `file DATEILISTE`
- `file` prüft zunächst anhand der Inode-Informationen, um welchen Datentyp es sich handelt, prüft aber zusätzlich den Anfang der Datei anhand der Informationen, die in der Datei `/etc/magic` enthalten sind.

##### Optionen

- b                Ausgabe ohne Angabe der Dateinamen
- f                Die Dateiliste wird aus der Datei DATEINAME entnommen  
DATEINAME    (Ein Dateiname pro Zeile).
- z                Bearbeitet auch den Inhalt von gepackten Dateien

```
htor@archimedes:~> ls
Documents backup bilder datei1 datei2 datei3 fifo
htor@archimedes:~> file *
backup:      directory
bilder:      directory
datei1:      ASCII text
datei2:      Bourne-Again shell script text
datei3:      ASCII text
Documents:  directory
fifo:        fifo (named pipe)
htor@archimedes:~>
```

### 1.5.6 Weitere grundlegende Befehle

#### 1.5.6.1 Das Kommando clear

- Dieser Befehl löscht den Bildschirm.

#### 1.5.6.2 Das Kommando cal

- Das Kommando cal zeigt einen Kalender an.
- `cal [[MONAT] JAHR]`
- Für JAHR sind Werte zwischen 1 und 9999 erlaubt und für MONAT zwischen 1 und 12.

##### Optionen

- m                Zeigt Montag als ersten Tag der Woche an
- j                Benutzt das Julianische Datum

-y Zeigt einen Kalender für das aktuelle Jahr

### 1.5.6.3 Das Kommando date

- Dieser Befehl zeigt und setzt das aktuelle Datum und die Uhrzeit. Allerdings kann nur *root* die Zeit setzen.

#### 1.5.6.3.1 Zeit anzeigen

- Die Anzeige der aktuellen Zeit erfolgt über
- `date [+FORMATSTRING]`
- `FORMATSTRING` ist ein beliebiger Text, der, falls er Sonderzeichen enthält, in Anführungszeichen zu setzen ist. Sinn macht dieser Text allerdings erst, wenn er Platzhalter enthält, die Datum oder Zeit anzeigen.
- Einige Beispiele für solche Platzhalter sind:

%m	Monat	%H	Uhrzeit
%d	Tag	%M	Minuten
%y	Jahr (zweistellig)	%S	Sekunden
%Y	Jahr (vierstellig)	%T	Zeit (hh:mm:ss)

#### 1.5.6.3.2 Zeit stellen

- Der Systemverwalter kann auch Zeit und Datum über diesen Befehl ändern.
- `date MMTTSSmm[ [HH]JJ] [.ss]`
- Wobei `MM` für Monat, `TT` für Tag des Monats, `SS` für Stunde, `mm` für Minute, `HH` fürs Jahrhundert, `JJ` fürs Jahr und `ss` für die Sekunden stehen.
- Die Änderung der Systemzeit hat keine Auswirkungen auf die Hardware-Uhr.
- Nach dem Neustart des Systems sind die Einstellungen wieder verloren.
- Für weitere Informationen zu Schaltern und Platzhaltern geben Sie `man date` ein.

#### 1.5.6.3.3 Beispiele

```
htor@archimedes:~> date +"Es ist der %d.%m.%y"
Es ist der 09.03.03
htor@archimedes:~> date +"Es ist %T Uhr"
Es ist 10:52:49 Uhr
htor@archimedes:~>
```

### 1.5.6.4 Das Kommando echo

- Der Befehl `echo` wirkt wie ein Echo, indem er einfach den angegebenen Text wieder auf den Bildschirm ausgibt.
- Seine Haupteinsatz findet er daher in den Shellskripten, der Ausgabe von Texten in Dateien und dem Anzeigen der Inhalte von Variablen.
- `echo [OPTIONEN] [TEXT]`

#### Optionen

- e erlaubt die Benutzung von Sonderzeichen im Text, z. B:
  - \n Zeilenumbruch
  - \a Piepton (ASCII-Zeichen 7)
  - \b Backspace
  - \f Seitenvorschub

\ooo beliebiges ASCII-Zeichen in oktaler Notation

-n unterdrückt Zeilenumbruch nach Ausgabe des Textes

- Texte mit Sonderzeichen (z. B. Leerzeichen) sollten in Anführungszeichen stehen.
- `echo` ohne Angabe eines Textes bewirkt die Ausgabe einer Leerzeile.

### 1.5.6.5 Farben im Terminal

- Um Terminals zu konfigurieren werden Codesequenzen benutzt.
- Sie werden als Escape-Sequenzen bezeichnet, da sie mit dem Escape-Zeichen (0x1B) beginnen.
- So kann die Farbe der Ausgabe festgelegt werden.
- Mit der Sequenz `<ESC>[attr];{fg};{bg}m` kann nun die Farbe vorgegeben werden.
- So bewirkt der Befehl
- `echo "^[[0;31;40mIch sehe rot"`
- daß der Text `Ich sehe rot' in roter Farbe ausgegeben wird.
- Das erste Zeichen ist das Escape-Zeichen.
- Es wird als `^[]` dargestellt. Um es zu schreiben drücken Sie `<STRG>+<V>` und dann `<ESC>`. Um wieder den Normalzustand herzustellen, müssen Sie für die Konsole den Befehl
- `echo "^[[0;37;40m"` eingeben.
- Eine Liste der möglichen Farben und Attribute liefert folgende Tabelle.
- Diese Werte werden auch in den Dateien `/etc/DIR_COLORS` und `.dir_colors` für die Farbcodierung verwendet.

Tabelle: Farbcodes fürs Terminal

{attr}		{fg}		{bg}	
0	Normaler Modus	30	Schwarz	40	Schwarz
1	Hell oder Fett	31	Rot	41	Rot
2	Dunkel	32	Grün	42	Grün
3	Unterstrichen	33	Gelb	43	Gelb
5	Blinkend	34	Blau	44	Blau
7	Invertiert	35	Margenta	45	Margenta
8	Versteckt	36	Türkis	46	Türkis
		37	Weiß	47	Grau
				48	Weiß

### 1.5.6.6 Das Kommando logout oder exit

- Dieses bash-Kommando beendet die Sitzung eines Benutzers.

### 1.5.6.7 Das Kommando tty

- Die Ausgabe eines Befehls landet immer auf der Konsole oder dem Terminal, auf dem es gestartet wurde.
- Aber auf welcher Konsole bin ich denn überhaupt?
- Da hilft der Befehl `tty` weiter.

**Bearbeiten Sie alle Aufgaben unter „Einführung in die Shell“!**

## 1.6 Die Shell II

### 1.6.1 Automatische Kommandoerganzung

- Die Shell ist konzipiert worden, um dem Benutzer eine komfortable Umgebung zum Arbeiten zu schaffen.
- Ein Tool dafur ist die automatische Kommandoerganzung.
- Oft reicht es aus, nur die ersten Zeichen eines Befehls oder Dateinamens einzugeben um diesen eindeutig zu identifizieren.
- Durch Betatigen der Tabulator-Taste (<TAB>) versucht die Shell diesen Namen zu vervollstandigen.
- So erweitert die Shell nach Betatigung der Tabulator-Taste die Buchstaben ec zu dem Befehl echo.
- Ist der Befehl der Dateiname nicht eindeutig uber die ersten Zeichen definiert, so gibt die Shell ein Signalton (Piep) zuruck.
- Wird die Tabulator-Taste noch einmal betatigt, zeigt die Shell eine Liste von Kommandos, die mit den entsprechenden Zeichen beginnen.
- So erhalten Sie beispielsweise nach Eingabe von mk und zweimaligem Betatigen der Tabulator-Taste folgende Liste.

```
htor@archimedes:~> mk
mkcfm      mkfontdesc  mkhybrid    mkmanifest  mktexlsrc
mkdir      mkfontdir   mkindex     mknod       mktexmf
mkdirhier  mkfontscale mkinfodir   mkpasswd    mktexpk
mkfifo     mkhtmlindex mkisofs     mktemp      mktextfm
```

- Auf diese Art und Weise konnen Sie sich auch eine Liste aller zur Verfugung stehenden Befehle ausgeben lassen.
- Drucken Sie einfach am leeren Prompt zweimal die Tabulator-Taste.
- Da es nicht gerade wenig Befehle sind, werden Sie dann erst einmal von der Bash gefragt, ob Sie auch wirklich alle Treffer anzeigen lassen mochten.

```
htor@archimedes:~><TAB><TAB>
Display all 1854 possibilities? (y or n)
```

- Genauso kann man sich auch schnell durch das Dateisystem bewegen.
- Durch die Eingabe von
- `cd /h<TAB>t<TAB>h<TAB>`
- erhalt man z. B. die Kommandosequenz `cd /home/htor`.
- Auch wenn Sie den Pfad nicht kennen, hilft die Tabulatortaste weiter.
- Einfach zweimal die Tabulatortaste drucken und alle in Frage kommenden Verzeichnisse werden angezeigt.

```
htor@archimedes:~> cd /var/
adm  games  lock  mail  opt  spool  X11R6
cache  lib  log  named  run  tmp
htor@archimedes:~>
```

Und was fur Verzeichnisse gilt, gilt auch fur alle anderen Dateien. Dann naturlich nicht mit `cd`, sondern einem dateibezogenen Befehl wie z. B. `less`.

```
htor@archimedes:~> less /etc/cron<TAB><TAB>
cron.d      cron.hourly  crontab      cron.weekly
cron.daily  cron.monthly crontab.old
htor@archimedes:~> less /etc/cron
```

### 1.6.2 Gruppierung von Kommandos

- Unter Linux ist es moglich, mehrere Kommandos in einer Zeile einzugeben.
- Dabei werden die einzelnen Kommandos durch das Zeichen „;“ getrennt.

```
htor@archimedes:~> date; echo -n "Heimatverzeichnis: "; echo ~
Son Mär  9 11:22:11 CET 2003
Heimatverzeichnis: /home/htor
htor@archimedes:~>
```

Damit die Ausgabe aller Befehle in eine Datei umgeleitet werden kann, müssen die Befehle gruppiert werden. Denn die Befehlssequenz

```
date; echo -n "Heimatverzeichnis: "; echo ~ > datei.txt
```

würde nur die Ausgabe des letzten echo Befehls in die Datei umlenken. Richtig muß die Kommandosequenz lauten:

```
(date; echo -n "Heimatverzeichnis: "; echo ~) > datei.txt
```

Die Klammern gruppieren die Kommandos zu einem Befehl. Deshalb wird dies als Kommandogruppierung bezeichnet.

### 1.6.3 Bedingte Ausführung

- Die Zeichen && und || verknüpfen zwei Befehle miteinander, wobei der zweite Befehl in Abhängigkeit vom Erfolg des ersten Befehls ausgeführt wird.
- `less index.htm || less index.html`
- bewirkt, daß nur wenn der erste Befehl keinen Erfolg hat, weil z. B. die Datei nicht existiert, der zweite Befehl ausgeführt wird.
- `cat index.html && cp index.html /backup`
- bewirkt, daß nur wenn der erste Befehle erfolgreich war, der zweite Befehl ausgeführt wird.

### 1.6.4 Substituierung von Kommandos

- Oft ist es sinnvoll die Ausgabe eines Kommandos in einem anderen Kommando zu verwenden.
- Das Einschließen eines Kommandos in `$(...)` oder in ``...`` bewirkt, daß die Ausgabe des Kommandos verwendet wird.

```
htor@archimedes:~> echo $(pwd)
/home/htor
htor@archimedes:~> echo `pwd`
/home/htor
htor@archimedes:~>
```

- Die obere Methode ist die neue Syntax der bash. Die zweite zeigt die Syntax der Bourne Shell.
- Mindestens eine der beiden Methode wird in der bash funktionieren.

- **Beispiel:**

Sie wollen den Kalender des aktuellen Monats in eine gleichnamigen Datei speichern.

```
htor@archimedes:~> cal -m > kalender-$(date +"%Y-%m").txt
htor@archimedes:~> ls -l kalender*
-rw-r--r--  1 htor  users      146 2003-03-09 11:25 kalender-2003-03.txt
htor@archimedes:~>
```

### 1.6.5 Variablen

- Um bestimmte, immer wiederkehrende Zeichenfolge zu speichern, werden in der Shell sogenannte Variablen verwendet.
- Jede Variable besitzt einen Namen, über den auf diese Zeichenfolge zugegriffen werden kann.
- Einer *Benutzervariable* kann ein Name und ein Wert zugewiesen werden, während die Namen der *Umgebungsvariablen* festgelegt sind.
- Eine besondere Form der Benutzervariablen sind die *Aliase*.

#### 1.6.5.1 Benutzervariablen

- Benutzervariablen werden meistens in Skripten verwendet.

- Die Variablennamen dürfen aus Buchstaben, Ziffern und dem Unterstrich bestehen.
- Dabei darf das erste Zeichen keine Ziffer sein.
- Den Variablen wird ein Wert durch folgende Sequenz zugewiesen.
- `VARIABLENNAME=WERT`
- Um eine Variable zu löschen, reicht es ihr einen Null-Wert zu geben.
- `VARIABLENNAME=`
- Die Variablen können nur von dem erzeugenden Prozess und seinen Kinderprozessen verwendet werden. Es sind lokale Variablen.

### 1.6.5.2 Das Kommando export

- Um eine Variable überall zugänglich zu machen, wird das bash-Kommando `export` verwendet.
- `export [OPTIONEN] [NAME [=WERT]]`
- `export` ohne die Angabe von Optionen zeigt eine Liste aller exportierten Variablen.

#### Optionen

- f Die Variable wird als Funktion interpretiert
- n Löschen einer Variablen aus der Liste
- p Liste aller exportierten Variablen (standard)

Folgende Kommandos erstellen eine Variable und machen sie global zugänglich.

```
ILove=Linux; export ILove
```

Tabelle 5.2: Auszug aus den Umgebungsvariablen

Name	Bedeutung
?	Fehlercode des letzten ausgeführten Kommandos (0 für ``OK"")
CDPATH	Suchpfad für das cd-Kommando
EDITOR	Name des Editors, den man verwenden möchte
EUID	Effektive Benutzerkennung (ro)
FCEDIT	Editor für den Befehl fc
HISTFILE	Name und Pfad für die History-Datei
HISTFILESIZE	Max. Größe der History-Datei
HISTSIZE	Max. Anzahl von Befehlen zur Speicherung in der History-Datei
HOME	Heimatverzeichnis des Benutzers
HOSTNAME	Name des Rechners
INPUTRC	Konfigurationsdatei für Kommandozeileneditor
PATH	Suchpfad für Kommandos
PS1	Aktueller Prompt
PS2	Prompt für die Eingabe fehlender Parameter
PWD	Name des aktuellen Verzeichnisses
RANDOM	Zufallszahl
SECONDS	Sekunden nach Starten der Shell
SHELL	Name der benutzten Login-Shell

TERM	Art des Linux-Terminals
UID	Benutzerkennung (ro)
USER	Login-Name des Users

### 1.6.5.3 Umgebungsvariablen

- Aussehen und Funktionsweise der Shell werden durch die Umgebungsvariablen festgelegt.
- Die Festlegung der Umgebungsvariablen erfolgt in verschiedenen Dateien, die in `/etc` oder im Heimatverzeichnis `~` liegen. So findet man z. B. die Umgebungsvariablen der `bash` in der Datei `/etc/profile` wieder.
- Ausführlichere Informationen über die Umgebungsvariablen liefert das Kommando `man bash`.

Schauen wir uns als Beispiel die Variable `PWD` an, die den Namen des aktuellen Verzeichnisses enthält.

```
htor@archimedes:~> echo $PWD
/home/htor
```

Der Variablen `PWD` kann ein neuer Wert zugewiesen werden.

```
htor@archimedes:~> PWD=/home/htor/Documents/
htor@archimedes:~/Documents/>
htor@archimedes:~/Documents/> pwd
/home/htor
```

```
htor@archimedes:~/Documents/> echo $PS1
\u@\h:\w>
htor@archimedes:~/Documents/>
```

Nach der Änderung der Umgebungsvariable `PWD` ändert sich auch das Aussehen des Prompts, da dieser die Umgebungsvariable nutzt. Wie aber der Befehl `pwd` zeigt, hat sich das aktuelle Verzeichnis nicht geändert.

### 1.6.5.4 Das Kommando printenv

- Um sich die angelegten Umgebungsvariablen ansehen zu können, wird der Befehl `printenv` verwendet.
- `printenv [VARIABLE]`
- Wird keine spezielle Variable angegeben, so zeigt `printenv` eine Liste aller Variablen an.

### 1.6.5.5 Das Kommando env

- `env` startet ein Kommando mit anderen Umgebungsvariablen.
- `env [OPTIONEN] [VARIABLE=WERT] [BEFEHL [ARG1] ...]`
- Wird kein `BEFEHL` angegeben, so listet `env` die resultierende Umgebung auf. Es arbeitet dann wie `printenv`.

### 1.6.5.6 Der Suchpfad für Programme

\* Die Umgebungsvariable `PATH` enthält eine Liste von Verzeichnissen, die nach einem Kommando durchsucht werden. Die einzelnen Verzeichnisnamen werden durch einen Doppelpunkt `:` getrennt. Im Gegensatz zu DOS wird das aktuelle Verzeichnis `.` nicht immer durchsucht, sondern muß auch in der `PATH`-Variable angegeben werden.

```
htor@archimedes:~> echo $PATH
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnome/bin:/opt/kde3/bin:/opt/kde2/bin:/usr/lib/java/bin:/opt/gnome/bin
htor@archimedes:~>
```

- Im Normalfall ist das aktuelle Verzeichnis sogar das letzte in der Suchreihenfolge.
- Das kann zu Problemen führen.
- So wird beim Ausprobieren von Skripten z. B. sehr gerne der Name `test` als Dateiname verwendet.
- `test` ist aber ein Programm im Verzeichnis `/usr/bin`.
- Daher wird beim Aufruf von `test` nicht das Skript im aktuellen Verzeichnis, sondern das Programm in `/usr/bin` gestartet, weil es dort zuerst gefunden wurde.
- Die Variable `PATH` wird in der Datei `/etc/profile` definiert.
- Um die Datei für sich selber zu ändern, sollte die Änderung in den Datei `~/.bash_profile`, `~/.profile` oder `~/.bashrc` erfolgen.
- Welche Datei als Beste dafür geeignet ist, ist von Distribution zu Distribution unterschiedlich.  
Bsp.: `PATH=$PATH:$HOME/bin`

### 1.6.5.7 Das Kommandoprompt

- Der Prompt steht immer am Anfang einer Kommandozeile.
- Er zeigt an, daß die Shell auf eine Eingabe wartet.
- Das Aussehen des Prompts wird durch die Umgebungsvariable `PS1` festgelegt.

```
htor@archimedes:~> echo $PS1
\u@\h:\w>
htor@archimedes:~>
```

- `\u`, `\h` und `\w` sind besondere Zeichen, die ausführliche Informationen enthalten. Für eine Liste der speziellen Zeichen siehe folgende Tabelle.
- Wie der Name `PS1` schon vermuten läßt, existiert noch eine Umgebungsvariable `PS2`.
- Diese steuert das Aussehen des zweiten Prompts.
- Dieser Prompt taucht immer dann auf, wenn Sie einen Befehl noch nicht vollständig eingegeben und doch die Eingabetaste gedrückt haben.
- Er zeigt an, daß noch etwas fehlt.

Im folgenden Beispiel fehlt das schließende Anführungszeichen.

```
htor@archimedes:~> echo "hallo
>"
hallo
```

```
htor@archimedes:~> PS2="ergaenzen >"
htor@archimedes:~> echo "hallo
ergaenzen >"
hallo
```

Tabelle: Spezielle Zeichen für den Prompt. Die kursiven Worte sind Eselsbrücken zum Merken

Zeichen		Bedeutung
<code>\a</code>	<i>alert</i>	Piepton
<code>\d</code>	<i>date</i>	Datum Format: Wochentag Monat Tag
<code>\e</code>	<i>escape</i>	ESC-Zeichen
<code>\h</code>	<i>host</i>	Hostname bis zum ersten .
<code>\H</code>	<i>host</i>	Der ganze Hostname
<code>\n</code>	<i>newline</i>	Neue Zeile
<code>\r</code>	<i>return</i>	Wagenrücklauf
<code>\s</code>	<i>shell</i>	Name der Shell

\t	<i>time</i>	Zeit Format: 24 Stunden HH:MM:SS
\T	<i>time</i>	Zeit Format: 12 Stunden HH:MM:SS
\@		Zeit Format: 12 Stunden a.m./p.m.
\u	<i>user</i>	Username des aktuellen Benutzers
\v	<i>version</i>	Version der bash
\V	<i>version</i>	Ausgabennummer der bash (Version + Patchstufe)
\w	<i>working</i>	Aktuelle Arbeitsverzeichnis
\W		Der Basisname des aktuellen Verzeichnisses
!\		Die Nummer in der History für den aktuellen Befehl
\#		Die Nummer des Kommandos für den aktuellen Befehl
\\$		Wenn die effective UID 0 ist, dann #, sonst \$
\nnn		Zeichen mit dem oktalen ASCII-Code nnn
\\		Backslash
\[		Beginn einer Folge von nichtdruckbaren Zeichen
\]		Ende einer Folge von nichtdruckbaren Zeichen

### 1.6.5.8 Der Befehl alias

- Für häufig wiederkehrende Befehle kann ein Alias angelegt werden.
- Dieses Alias ist vergleichbar mit einem kleinen Skript, wird aber wesentlich schneller ausgeführt.
- Um z. B. für den Befehl `cd ..` den Alias `..` zu setzen, wird folgendes Kommando eingegeben.
- `alias ..="cd .."`
- Die allgemeine Syntax für den Befehl alias lautet:
- `alias [NAME[=KOMMANDO]]`
- Dabei gibt das Kommando alias unter Angabe des Aliasnamen das dazugehörige Kommando aus.
- Der Aufruf ohne Parameter gibt eine Liste aller definierten Aliase mit ihren Kommandosequenzen aus.
- Der Befehl `alias` sowie der Gegenbefehl `unalias` sind fester Bestandteil der Bash und direkt eingebaut.

### 1.6.5.9 Der Befehl unalias

- Um die ganze Alias-Geschichte rückgängig zu machen, wird der Befehl `unalias` verwendet.
- Damit kann der normale Benutzer Aliase wieder entfernen, die der Systemverwalter für alle Benutzer durch Eintragen in die `/etc/.profile` angelegt hat.
- `unalias [-a] [NAMENSLISTE]`
- Wenn Sie den Schalter `-a` verwenden, arbeitet `unalias` sehr gründlich: Alle Alias-Definitionen werden gelöscht.

### 1.6.6 Die History

- Die History-Liste umfaßt die letzten eingegeben Kommandos (*events*).

- Diese Liste kann dazu benutzt werden um alte Befehle wieder zu benutzen, sie zu bearbeiten oder sie zu analysieren.
- Dabei wird jedes Element von der Shell mit einer Nummer versehen.
- Beim Starten der Shell wird die History-Liste aus der Datei `.bash_history` im Heimatverzeichnis erstellt.
- Die Umgebungsvariable `HISTFILE` enthält den Namen dieser Datei.
- Die Größe der Datei in Zeilen wird durch `HISTFILESIZE` bestimmt.
- Beim Ausloggen wird eine Anzahl von Zeilen, die in der Variablen `HISTSIZE` definiert worden ist, an die Datei `.bash_history` angehängt oder die komplette Datei wird durch diese ersetzt, je nachdem ob das `histappend`-Attribut der Shell gesetzt ist oder nicht.
- Mit den Shellvariablen `HISTCONTROL` und `HISTIGNORE` kann in Abhängigkeit von im Befehl enthaltenen Zeichen bestimmt werden, welcher Befehl in der History gespeichert wird und welcher nicht.

#### 1.6.6.1 Das Kommando history

- Das `bash`-Kommando `history` zeigt den Inhalt der History-Liste an.
- `history [OPTIONEN] [ZAHL]`
- Ohne Angabe von Parametern wird die gesamte History-Liste angezeigt. Durch Angabe einer Zahl werden die letzten `ZAHL` Zeilen der History-Liste angezeigt.

##### Optionen

- c Löschen der History-Liste

#### 1.6.7 Umleitungen

- Standardmäßig kommen die Eingaben von der Tastatur und die Ausgabe des Programms und seiner Fehlermeldungen landen auf dem Bildschirm.
- Diese Ausgaben können aber in Dateien oder Geräte umgeleitet werden.

##### 1.6.7.1 Umleitung der Standardeingabe

- Mit dem Zeichen `<` liest das Programm nicht von der Standardeingabe sondern holt seine Daten aus der angegebenen Datei.
- `cat < meinText`

##### 1.6.7.2 Umleitung der Standardausgabe

- Mit dem Zeichen `>` oder `1>` wird die Ausgabe des Programm in die angegebene Datei umgelenkt.
- `cat * > alleDateien`
- Dabei wird jedesmal eine neue Datei erstellt und der Inhalt der alten Datei gelöscht.
- Soll die Ausgabe aber an eine Datei angehängt werden, so verwendet man `>>`.
- `echo $UID $USER >> logdatei`

##### 1.6.7.3 Umleitung der Fehlerausgabe

- Mit `2>` wird die Ausgabe der Fehlermeldung umgeleitet.
- Oft stören diese, wenn sie nur melden, daß keine Zugriffsrechte auf ein Verzeichnis bestehen.
- `ls -Rl 2> /dev/null`

##### 1.6.7.4 Umleitung beider Ausgaben

- Um beide Ausgaben (Fehler und Programm) gemeinsam in eine Datei zu leiten, wird `&>` verwendet.
- `cat *.log &> allogs.txt`
- oder `cat *.log >& allogs.txt`
- Es geht aber auch

- `cat *.log > allogs.txt 2>&1`

### 1.6.7.5 Pipelines

- Pipelines ermöglichen es, die Ausgabe eines Kommandos direkt an die Eingabe eines anderen Kommandos zu senden.
- Realisiert wird dies durch ein zwischengeschaltetes Pipe-Zeichen `|`.
- `ls -Al | less`

### 1.6.7.6 Das Kommando tee

- Das Kommando tee sendet gleichzeitig die Ausgabe an eine Datei und an die Standardausgabe, die der Bildschirm oder eine weitere Pipeline sein kann.
- `tee [OPTIONEN] DATEILISTE`

#### Optionen

`-a` Die Daten werden an eine vorhandene Datei angehängt

- Diese Kommandosequenz sendet den Inhalt des Verzeichnisses in die Datei `myDir` und auf den Bildschirm.
- `ls -al | tee myDir`
- Hier wird der Inhalt der Datei `amAngelhaken` an die Datei `angeln.txt` angefügt und gleichzeitig mit `less` angezeigt.
- `cat amAngelhaken | tee -a angeln.txt | less`

## 1.7 Hilfe im und zu Unix

### 1.7.1 Lokale Hilfequellen

- Es gibt eine Reihe von lokalen Dokumentationen in einem Linux-System.
- Folgende Tabelle zeigt die Typen und ihre Position im Dateisystem.

Tabelle: Die lokalen Dokumentationen und ihre Position

Dokumentation	Position	SuSE
Manual-Pages	<code>/usr/man</code>	<code>/usr/share/man</code>
info	<code>/usr/info</code>	<code>/usr/share/info</code>
HOWTOs	<code>/usr/doc/HOWTO</code>	<code>/usr/share/doc/howto</code>
Frequently Ask Questions (FAQ)	<code>/usr/doc/FAQ</code>	nicht mehr vorhanden
Programm Dokumentationen	<code>/usr/doc/PROGRAMMNAME</code>	<code>/usr/share/doc/packages/PAKETNAME</code>

#### 1.7.1.1 Manual Pages

- Die Manual-Pages, auch oft Man-Pages genannt, sind eine extensive Online-Hilfe zu fast jedem Tool, Systemaufruf oder Kommando des Systems.
- Sie sind nicht gerade benutzerfreundlich, da sie mehr als Referenz für die Syntax gedacht sind und nicht als Online-Tutorial.
- Für die Verwendung der Manual-Pages wird das Tool `man` verwendet. Dieses Tool sucht den passenden Hilfetext heraus und stellt ihn durch einen Textbetrachter dar.
- In den meisten Fällen ist `less` der Pager.
- Die Hilfetexte sind in mehrere Teile aufgeteilt, wie in Tabelle zu sehen.

- Es muß aber nicht unbedingt jeder Abschnitt in dem Text vorkommen.

Tabelle 6.2: Die Abschnitte in den Manual-Pages

Abschnitt	Beschreibung
NAME	Name und kurze Beschreibung des Kommandos
SYNOPSIS	Anwendung und Syntax des Kommandos
DESCRIPTION	Beschreibung des Kommandos und seiner Optionen
OPTIONS	Liste aller Optionen mit Kurzbeschreibung
SEE ALSO	Manual-Pages zu verwandten Inhalten
DIAGNOSTICS	Beschreibung von Fehlermeldungen
FILES	Liste der Dateien, die von dem Kommando verwendet werden
BUGS	Bekannte Probleme
HISTORY	Meilensteine der Programmentwicklung
AUTHOR	Autor des Programms und Mitarbeiter

- Die Dateien für die Manual-Pages können in verschiedenen Pfaden liegen.
- Generell ist dies der Pfad `/usr/man`.
- Bei der SuSE-Distribution liegen die Hilfetexte im Verzeichnis `/usr/share/man`.
- Die Umgebungsvariable `MANPATH` enthält die Pfade zu den Hilfetexten.
- In diesem Verzeichnis existieren mehrere Unterverzeichnisse für die neun „Kapitel“ der Manual-Pages.
- Eine Übersicht über die Kapitel liefert die folgende Tabelle.
- Wenn nun nach einem Thema gesucht wird, wird die erste Seite angezeigt, die gefunden wird.
- Dabei werden die Kapitel in folgender Reihenfolge durchsucht:
  - 1, 8, 2, 3, 4, 5, 6, 7, 9

Tabelle: Die neun „Kapitel“ der Manual-Pages

Verzeichnis	Beschreibung
man1	Ausführbare Programm oder Shell-Kommandos
man2	Systemaufrufe (Kernelfunktionen)
man3	Bibliotheksaufrufe (Funktionen mit Systembibliotheken)
man4	Spezielle Dateien (meistens in <code>/dev</code> )
man5	Dateiformate und Konventionen
man6	Spiele
man7	Makropakete und Konventionen
man8	Systemadministrationskommandos
man9	Kernelroutinen (kein Standard)

#### 1.7.1.1.1 Das Kommando man

- Ruft die Online-Dokumentation (Manual-Pages) zu einem Befehl auf.

- `man [OPTIONEN] [SEKTION] [THEMA]`
- Dabei wird die erste passende Seite zu dem Thema angezeigt.

### Optionen

-a	Zeigt alle passenden Seiten hintereinander an
-f	Zeigt eine Kurzbeschreibung zu den passenden Seiten an.
-k	Stichwortsuche in der Hilfe nach den Kurzbeschreibungen
-K	Volltextsuche in der Hilfe (Achtung: Nicht in jedem man-Tool implantiert)

- die Konfigurationsdatei für man ist `/etc/man.config` bzw. `/etc/manpath.config`.
- Die Umgebungsvariable `PAGER` enthält den Namen des Textbetrachters, den man verwendet.

## 1.7.1.2 TexInfo

- TexInfo, die GNU Hypertext Dokumentation, ist einfacher zu benutzen als die Manual-Pages und oft auch ausführlicher.
- Für die Darstellung der Seiten wird das Kommando `info` verwendet.

### 1.7.1.2.1 Das Kommando info

- Ruft die Online-Dokumentation (TexInfo-Dateien) eines Befehls auf.
- `info [KOMMANDO]`
- Bei Linux befinden sich die TexInfo-Dateien in den Verzeichnissen `/usr/local/info`, `/usr/share/info` und `/usr/info`.

### Steuerung

<q>	beendet das Programm
<STRG>+<h>	ruft die Hilfe auf
<n>	zum nächsten Knoten (Menüpunkt)
<p>	zum vorherigen Knoten (Menüpunkt)
<u>	zum aufwärtigen Knoten (Menüpunkt)
<l>	zum letzten dargestellten Knoten (auch die Hilfe ist ein Knoten)
<m>	Menüpunkt mit Namen aufrufen
<1> ... <9>	Menüpunkt mit dieser Nummer folgen

### 1.7.1.3 Das Kommando help

- Zeigt hilfreiche Informationen über in die bash eingebaute Befehle an.
- `help [KOMMANDO]`

### 1.7.1.4 HOWTO's

- Die Linux HOWTOs sind ausführliche Dokumente, die die Bearbeitung bestimmter Aufgaben in Linux beschreiben.
- Die HOWTO-Dokumente behandeln meist komplexere Aufgaben und sind daher länger.
- Für einfachere Aufgaben wie LILO oder Drucken gibt es kürzere Texte, die *mini-HOWTOs*.
- Die HOWTO-Texte befinden sich meistens in komprimierter Form im Verzeichnis `/usr/doc/HOWTO`, `/usr/doc/howto` bzw. `/usr/share/doc/howto`.
- Bezugsquelle für die HOWTOs sind

- <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>
- <http://sunsite.unc.edu/LDP/HOWTO/>
- <http://www.linuxdoc.org/HOWTO/>
- bzw. einer Ihrer Spiegelserver.

### 1.7.1.5 FAQ

- Die FAQs (Frequently Asked Questions) sind eine Sammlung von Dokumenten im Zusammenhang mit den HOWTOs.
- Sie sind in der Form Frage-Antwort aufgebaut, wobei die Fragen gesammelt werden, die die Benutzer am häufigsten stellen.
- Die FAQs sind im HTML-, PostScript- und Text-Format vorhanden und befinden sich im Verzeichnis `/usr/doc/FAQ`.
- Die Online-Quelle für die neuesten FAQs ist
  - <http://www.linuxdoc.org/FAQ/>
- Seit SuSE 8.0 sind die FAQs kein Bestandteil mehr der Distribution.
- Das ist aber zu verschmerzen, da ihr Informationswert nicht sehr groß ist.

### 1.7.1.6 Programmdokumentation

- Dokumentationen über spezielle Programme finden Sie im Verzeichnis `/usr/doc` in einem Unterverzeichnis mit dem Namen des Programms.
- Bei der SuSE-Distribution finden Sie weitere Informationstexte zu den Programmen der Softwarepakete im Verzeichnis `/usr/share/doc/packages` und dessen Unterverzeichnissen.
- Die meisten Texte können mit `less` oder `more` gelesen werden.
- Andere Dokumentationen sind als L<sup>A</sup>TEX-DVI, PostScript, HTML oder PDF zugänglich
- Manche Programme legen aber auch bei Ihrer Installation eine Manual-Page an.

## 1.7.2 Befehle für Hilfe

### 1.7.2.1 Das Kommando whereis

- Das Kommando `whereis` lokalisiert die Programmdatei, Quellcodedatei und die Online-Hilfe (Manual-Pages) für ein Kommando.
- `whereis [OPTIONEN] KOMMANDO`
- Die Suche ist dabei auf eine limitierte Anzahl von bekannten Verzeichnissen beschränkt.
- Wenn die gesuchte Datei nicht gefunden wurde, wird nur der gesuchte Name zurückgegeben.

#### **Optionen**

<code>-b</code>	Suche nach Programmdatei (binary)
<code>-s</code>	Suche nach Quellcodedatei (source file)
<code>-m</code>	Suche nach Online-Hilfe (manpages)
<code>-u</code>	Suche nach anderen Dateien (unusal entries)

- Voreinstellung ist `-bmsu` zur Suche nach allen möglichen Dateien.

### 1.7.2.2 Das Kommando which

- Das Kommando `which` zeigt an, welches Programm mit welchem Pfad bei Eingabe eines Kommandos gestartet wird.
- `which KOMMANDO`
- `which` durchsucht die Verzeichnisse, die in der Umgebungsvariablen `PATH` angegeben sind, und gibt den Pfad des ersten Fundorts des gesuchten Programms an.

### 1.7.2.3 Das Kommando whatis

- Zeigt die Kurzbeschreibung der Online-Hilfe (Manual-Pages) an.
- `whatis [OPTIONEN] THEMA`
- Innerhalb jeder Manualseite ist eine Kurzbeschreibung vorhanden. `whatis` sucht Schlüsselwort in den Kurzbeschreibungen der Indexdatenbank.
- Falls es eine solche nicht im Manualpfad gibt, durchsucht es die `whatis`-Datenbank, die sich in `/usr/man/man1` befindet, nach dem Schlüsselwort.
- `whatis` verhält sich wie `man -f`.

### 1.7.2.4 Das Kommando apropos

- Der Befehl `apropos` ermöglicht eine Stichwortsuche in der Kurzbeschreibung der Hilfe.
- `apropos [OPTIONEN] STICHWORT`
- Die Suche mit `apropos` ist identisch zu `man -k`.

### 1.7.2.5 Das Kommando type

- Das Kommando `type` zeigt an, wie ein Begriff interpretiert wird, wenn er als Kommando benutzt wird.
- `type [OPTIONEN] BEGRIFFE`

#### Optionen

- t Ausgabe eines einzelnen Wortes ('alias', 'keyword', 'function', 'builtin', 'file' oder '')
- p Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn -t 'file' liefert.)
- a Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn -t 'file', 'alias' oder 'function' liefert.)

Die Schalter -t, -p und -a können nicht zusammen verwendet werden.

## 1.8 Textfilter

### 1.8.1 Ausgabe ganzer Dateien

- Um ganze Dateien auszugeben haben wir schon die Pager `less` und `more` sowie den Verbindungsbefehl `cat` kennengelernt.
- Aber es gibt noch weitere Befehle, die sich mit der Ausgabe der ganzen Datei beschäftigen.

#### 1.8.1.1 Das Kommando tac

- Das Kommando `tac` dreht die Reihenfolge der Datensätze (Voreinstellung: Zeilen) in Dateien um.
- Diese werden dann zusammengefügt und ausgegeben.
- `tac [OPTIONEN] [DATEILISTE]`
- `tac` dreht die Reihenfolge der Datensätze in einer Datei um.
- Werden keine speziellen Datensatztrenner (Separatoren) angegeben, wird die Zeile als Datensatz interpretiert.

#### Optionen

- r Separator wird als regulärer Ausdruck interpretiert
- s SEP Neuen Separator eingeben (Voreinstellung: NeueZeile)

Gibt die Datei `t1` in umgekehrter Reihenfolge in die Datei `t2` aus.

```
tac t1 > t2
```

### 1.8.1.2 Das Kommando nl

- Mit diesem Kommando werden der Ausgabe einer Textdatei Zeilennummern hinzugefügt.
- `nl [OPTIONEN] [DATEILISTE]`
- `nl` fügt für jede Zeile einer Textdatei eine Zeilennummer hinzu.
- Die Nummerierung beginnt bei jeder neuen Seite wieder neu.
- Eine Seite gliedert sich in drei Abschnitte: header, body und footer (Kopf, Rumpf und Fuß).

#### Optionen

<code>-b STYLE</code>	Wählt die Nummerierungsart für den Rumpf aus. Wird eine Zeile nicht nummeriert, so wird der Zeilenzähler nicht erhöht. Das Trennzeichen wird aber eingefügt. Als Stilarten stehen zu Verfügung
<code>-a</code>	alle Zeilen nummerieren
<code>-n</code>	keine Zeilen nummerieren (header, footer)
<code>-t</code>	alle Zeilen mit Inhalt nummerieren (body)
<code>-pREGEXP</code>	nur Zeilen, die den Ausdruck REGEXP enthalten, nummerieren
<code>-h</code>	Wie <code>-b</code> , aber für den Kopf.
<code>-f</code>	Wie <code>-b</code> , aber für den Fuß.
<code>-d CD</code>	Setzt die Abschnittsmarkierung neu
<code>-i N</code>	Schrittweite der Nummerierung. Für jede Zeile wird N hinzuaddiert. (Voreinstellung: 1)
<code>-l N</code>	Zählt N aufeinanderfolgende leere Zeilen als eine (Voreinstellung: 1)
<code>-n FORMAT</code>	Setzt das Format für die Zeilennummerierung fest
<code>ln</code>	links ausgerichtet ohne führende Nullen
<code>rn</code>	rechts ausgerichtet ohne führende Nullen
<code>rz</code>	rechts ausgerichtet mit führenden Nullen
<code>-p</code>	Startet die Nummerierung am Anfang einer logischen Seite nicht neu
<code>-s STRING</code>	STRING ist die Trennzeichenkette zwischen Nummerierung und Zeile (Voreinstellung: TAB-Zeichen)
<code>-v N</code>	Startet die Nummerierung mit N (Voreinstellung: 1)
<code>-w N</code>	Anzahl der Zeichen für die Nummerierung (Voreinstellung: 6)

### 1.8.1.3 Das Kommando od

- Das Kommando `od` gibt den Inhalt einer Datei in Oktal-, Dezimal-, Hexadezimal- oder ASCII-Darstellung auf der Standardausgabe aus.
- `od [OPTIONEN] [DATEILISTE]`
- Dieser Befehl wird meistens zur Ausgabe von Binärdateien eingesetzt.

#### Optionen

<code>-b</code>	Byteweise Ausgabe in oktaler Form
<code>-c</code>	Byteweise Ausgabe in ASCII-Format
<code>-d</code>	Dezimale Ausgabe von je zwei Byte

-x Hexadezimale Ausgabe von je zwei Byte

## 1.8.2 Textformatierung

- Die Befehle `fmt`, `pr` und `fold` formatieren den Dateitext um ihn z. B. für Ausgaben zu optimieren.

### 1.8.2.1 Das Kommando `fmt`

- Das Kommando `fmt` erzeugt durch Trennen und Zusammenfügen Ausgabezeilen mit vorgegebener Zeilenlänge.
- Dabei liest `fmt` aus Dateien oder von der Standardeingabe und gibt das Ergebnis auf der Standardausgabe aus.
- `fmt [OPTIONEN] [DATEILISTE]`

#### Optionen

- s Erlaubt nur das Zerlegen von Zeilen und nicht das Zusammenfügen von zu kurzen Zeilen
- u Reduziert die Anzahl der Leerzeichen zwischen Wörtern auf ein und zwischen Sätzen auf zwei Leerzeichen
- WIDTH Gibt Zeilen mit einer Breite von WIDTH Zeichen aus.  
-w WIDTH (Voreinstellung: 75)
- p PREFIX Es werden nur Zeilen verarbeitet, die mit der Zeichenfolge PREFIX beginnen. Führende Leerzeichen werden ignoriert. Führende Leerzeichen und PREFIX werden entfernt, die Zeilen werden umgebrochen und dann werden vor jeder Zeile die Leerzeichen und der PREFIX wieder angefügt. Einsatzgebiet: Umbrechen von Kommentaren in Programmen, ohne daß der Programmcode verändert wird.

Diese Kommandosequenz entfernt aus einer HTML-Datei alle überzähligen Leerzeichen.

```
fmt -u index.html > index.neu.html
```

Hier werden die Kommentare in einem Perl-Skript auf eine Zeilenlänge von 70 Zeichen umgebrochen.

```
fmt -p \# -w 70 sort.pl > sort.new.pl
```

### 1.8.2.2 Das Kommando `pr`

- Das Kommando `pr` formatiert eine Textdatei entsprechend festgelegter Optionen.
- `pr [OPTIONEN] [DATEILISTE]`
- Wenn keine Optionen ausgewählt werden, werden die Dateien seitenweise formatiert.
- Jede Seite besteht aus 66 Textzeilen: einem fünfzeiligen Kopf, einem Textbereich und einem fünfzeiligen Fuß.
- Die Kopfzeile enthält Seitenzahl, Dateiname, Datum und Uhrzeit.
- Alternativ kann der Text auch in Spalten ausgegeben werden.

#### Optionen

- +ANFANG[:ENDE] Ausgabe von Seite ANFANG bis Seite ENDE
- SP Gibt den Text in SP Spalten aus (Voreinstellung: 1)
- c Ausgabe von nichtdruckbaren Zeichen (show-control-char)
- d Fügt Leerzeilen ein (double-space)
- f Verwendet Zeilenvorschub anstatt NeueZeile

-F	Kopf- und Fußbereich werden auf drei Zeilen reduziert (form-feed)
-h HEADER	Ersetzt den Dateinamen im Kopf durch die Zeichenkette HEADER
-l LENGTH	Setzt die Seitenlänge (Voreinstellung: 66 Zeilen) (page-length)
-m	Fügt Dateien spaltenweise zusammen ( <b>merge</b> )
-n[SEP[ZAHL]]	Nummeriert die Zeilen durch SEP Trennzeichen ZAHL Anzahl der Ziffern
-N ANFANG	Beginnt Nummerierung der Zeilen mit ANFANG (nur im Zusammenhang mit -n verwendet)
-s SEP	Trennzeichen für Spalten
-S SEP	Trennzeichenkette für Spalten
-t	Führt keine Seitenformatierung durch
-w WIDTH	Einstellung Textbreite auf WIDTH Zeichen

Gibt die Datei zweispaltig von Seite 4 bis Seite 10 aus.

```
pr -2 +4:10 linux.tex
```

Gibt die Datei mit Zeilennummerierung (Trennzeichen `:', 4 Ziffern) auf dem Drucker aus.

```
pr -n:4 hallo.pl > lpr
```

### 1.8.2.3 Das Kommando fold

- Das Kommando `fold` gibt den Inhalt einer Datei aus, wobei die Zeilen auf eine definierte Länge umgebrochen werden.
- `fold [OPTIONEN] [DATEILISTE]`
- Im Normalfall ermittelt `fold` die Bildschirmspalten und bricht die Zeilen auf diese Länge um.
- Ein Tabulatorzeichen kann mehrere Spalten umfassen.

#### Optionen

-b	Zählung nach Bytes und nicht nach Spalten ( <b>bytes</b> )
-s	Umbruch erfolgt wenn möglich nur an Leerzeichen ( <b>spaces</b> )
-w WIDTH	Stellt Textbreite auf WIDTH Spalten/Bytes ein ( <b>width</b> )

### 1.8.3 Teilen von Texten

- Diese Befehlsgruppen zeigt nur Teile des Textes (head und tail) oder zerlegt ihn in mehrere Teile (split).

#### 1.8.3.1 Das Kommando head

- Das Kommando `head` gibt den Anfang einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe auf.
- `head [OPTIONEN] [DATEILISTE]`
- Werden mehrere Dateien angegeben, so fügt `head` den Dateinamen in der Form `==> DATEINAME <==`
- als Kopf vor dem Ausgabebetext ein. Der Befehl `tail` arbeitet ähnlich.

### Optionen

- c B           Gibt anstatt der ersten 10 Zeilen, die ersten B Zeichen aus.
- n N           Gibt anstatt der ersten 10 Zeilen, die ersten N Zeilen aus.
- N
- q            Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
- v            Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe

Diese Kommandosequenz gibt die ersten 5 Zeilen der Datei mit dem Dateinamen als Kopf aus.  
`head -n 5 -v links.html`

### 1.8.3.2 Das Kommando tail

- Das Kommando `tail` gibt das Ende einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe aus
- `tail [OPTIONEN] [DATEIILISTE]`
- Werden mehrere Dateien angegeben, so fügt `tail` den Dateinamen in der Form
- `==> DATEINAME <==`
- als Kopf vor dem Ausgabebetext ein. Der Befehl `head` arbeitet ähnlich.

### Optionen

- c B           Gibt anstatt der letzten 10 Zeilen, die letzten B Zeichen aus.
- n N           Gibt anstatt der letzten 10 Zeilen, die letzten N Zeilen aus.
- N
- q            Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
- v            Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe
- f            Mit dieser Option überwacht `tail` kontinuierlich das Ende einer oder mehrerer Dateien. Wird an die Datei etwas angehängt, so werden die Änderungen ausgegeben. Die Überwachung wird mit STRG+C abgebrochen.

Diese Kommandosequenz gibt die letzten fünf Zeilen der Dateien ohne den Dateinamen als Kopf aus.

```
tail -n 5 -q seite1.html seite2.html seite3.html
```

Mit der folgenden Sequenz werden mehrere Dateien auf Veränderung überwacht.

```
tail -fn 15 /etc/passwd /etc/group /tmp/mylogfile
```

### 1.8.3.3 Das Kommando split

- Das Kommando `split` zerlegt Dateien in mehrere kleinere Dateien.
- `split [OPTIONEN] [DATEI [PREFIX]]`
- Wird keine Datei angegeben bzw. „-“ als Dateiname, dann wird von der Standardeingabe gelesen.
- `split` zerlegt eine große Datei in kleinere Dateien vordefinierter Größe (Voreinstellung: 1000 Zeilen).
- Die dabei entstehenden Dateien beginnen mit einem Prefix (Voreinstellung: x) und werden zweistellig mit Buchstaben durchnummeriert. (xaa, xab, xac, ...).

### Optionen

- LINES       Größe der Ergebnisdateien in Zeilen
- l LINES

- b BYTES      Größe der Ergebnisdateien in Bytes
- b      angehängt bedeutet BYTES x 512
- k      angehängt bedeutet BYTES x 1024
- m      angehängt bedeutet BYTES x 1048576

## 1.8.4 Textstatistik

- Die folgenden Befehle analysieren den Text

### 1.8.4.1 Das Kommando wc

- Das Kommando `wc` zählt die Zeichen, die durch Leerzeichen getrennten Worte und die Zeilen einer Eingabe.
- `wc [OPTIONEN] [DATEILISTE]`
- `wc` gibt pro Datei die Anzahl der Zeilen, die Anzahl der Worte, die Anzahl der Zeichen und den Namen der Datei.
- Werden mehrere Dateien angegeben, so wird am Schluß eine Zusammenfassung aller Dateien angezeigt.

#### Optionen

- c            Anzahl der Zeichen ausgeben
- w            Anzahl der Worte ausgeben
- l            Anzahl der Zeilen ausgeben

Gibt die Anzahl der Worte und Zeilen in den drei Dateien an.

```
wc -lw artikel1.txt artikel2.txt artikel3.txt
```

Bestimmt die Anzahl der Dateien im Verzeichnis `/etc`.

```
ls -Al /etc | wc -l
```

## 1.8.5 Sortieren

- Daten werden meistens erst dann richtig lesbar, wenn Sie sortiert werden.
- Die Tools `sort`, `comm` und `uniq` kümmern sich um diesen Bereich.

### 1.8.5.1 Das Kommando sort

- Der Befehl `sort` durchsucht, verbindet oder vergleicht alle Zeilen einer Datei.
- `sort [OPTIONEN] [DATEILISTE]`
- `sort` kennt drei Arten der Funktion. Sortieren, Verbinden und Prüfen auf Sortierung.

#### Optionen

- c            Prüft, ob die Datei sortiert ist (check sort)
- m            Verbindet mehrere Dateien (die nicht sortiert werden, sondern als Gruppe behandelt werden) miteinander
- b            Führende Leerzeichen eines Schlüsselfeldes werden ignoriert
- d            Sortieren als Telefonverzeichnis: Nur Buchstaben, Ziffern und Leerzeichen werden berücksichtigt
- f            Unterscheidet nicht zwischen Groß- und Kleinbuchstaben
- g            Sortiert nach Zahlen anstatt lexikalisch
- n            Die Optionen unterscheiden sich in der Konvertierung der Zahlen zum Vergleich

-o DATEI	Ausgabe in die Datei DATEI umlenken
-r	Umkehrung der Sortierung
-t SEP	Trennzeichen für die Felder definieren
-u	Unterdrückung von gleichen Zeilen
+POS1[-POS2]	Als Schlüsselfelder werden die Felder ab POS1 bis POS2 (oder bis zum Zeilenende) verwendet.

- Die Zählung der Spalten beginnt bei 0!

Daher sortiert dieser Befehl die Datei mai.log nach der sechsten Spalte. Trennzeichen ist das Leerzeichen.

```
sort +5 -t " " mai.log > hits.log
```

Dieser Kommandosequenz sortiert eine Datei nach Telefonbuchmethode und unter Vernachlässigung der Groß- und Kleinschreibung.

```
sort -fd telefon.buch | less
```

### 1.8.5.2 Das Kommando uniq

- Das Kommando uniq entfernt aus einer sortierten Datei die doppelten Zeilen.
- `uniq [OPTIONEN] [INPUT [OUTPUT]]`

#### Optionen

-c	Gibt an, wie oft die Zeile vorkommt
-i	Ignoriert Groß- und Kleinschreibung
-d	Gibt nur die doppelten Zeilen aus
-u	Gibt nur die nicht doppelten Zeilen aus

- Dieser Befehl extrahiert die siebente Spalte aus der Datei mai.log, sortiert sie, entfernt die doppelten Zeilen und gibt an, wie oft die doppelten Zeilen vorkamen.

```
cut -f 7 -d " " mai.log | sort | uniq -c | less
```

### 1.8.6 Zeilenoperationen

- Um einen Text Zeile für Zeile zu analysieren und zu bearbeiten sind die folgenden Befehle programmiert worden.

#### 1.8.6.1 Das Programm cut

- Das Kommando cut gibt ausgewählte Felder (Spalten) aus jeder Zeile einer Datei auf der Standardausgabe aus.

```
cut [OPTIONEN] [DATEILISTE]}
```

#### Optionen

-b BYTES	Zeige die angegebenen BYTES an.
-f NR	Zeige das Feld (die Spalte) Nummer NR (field)
-d ZEICHEN	ZEICHEN ist der Spaltentrenner (delimiter)
-s	Zeilen ohne Trennzeichen (delimiter) werden nicht ausgegeben

- Das Tabulatorzeichen ist als Trennzeichen voreingestellt.
- Das Trennzeichen kann ein Leerzeichen oder ein anderes Sonderzeichen sein.

- Dann sollte es in Anführungsstriche gesetzt (oder allgemeiner maskiert) werden.

Diese Kommandosequenz gibt die Benutzernamen aus der Datei /etc/passwd aus.

```
cut -f 1 -d : /etc/passwd
```

Gibt die ersten 10 Zeichen jeder Zeile aus.

```
ls -l | cut -b -10
```

### 1.8.7 Suchen und Ersetzen

- Um bestimmte Stellen aus Texten zu extrahieren oder zu ändern werden die Tools grep, tr, expand, sed, diff und patch verwendet.

#### 1.8.7.1 Das Kommando grep

- Der Befehl grep (*Global Regular Expression Print*) durchsucht eine Textdatei nach bestimmten Mustern und gibt die Zeilen, in denen das Muster vorkommt, auf der Standardausgabe aus.
- `grep [OPTIONS] MUSTER [DATEILISTE]`

#### Optionen

-G	Interpretiert das MUSTER als regulären Ausdruck; Standardeinstellung. Nicht zusammen mit -F und -E verwenden
-E	Interpretiert das MUSTER als erweiterten regulären Ausdruck. Nicht zusammen mit -F und -G verwenden
-F	Interpretiert das MUSTER als einfache Zeichenkette. Nicht zusammen mit -F und -E verwenden
-c	Zeigt nur die Zeilennummern der gefundenen Zeilen an
-n	Zeigt zusätzlich zur Zeile auch die Zeilennummer an
-v	Zeigt die Zeilen an, die nicht dem MUSTER entsprechen
-f DATEINAME	Die Liste der zu bearbeitenden Dateien
-h	Unterdrückt die Ausgabe des Dateinamens bei Verwendung einer Dateiliste.
-i	Unterscheidet bei der Suche nicht nach Groß- und Kleinschreibung
-w	MUSTER wird als ganzes Wort und nicht als Teil des Wortes betrachtet
-l	Zeigt den Namen der Datei an, wenn die Zeile darin gefunden wurde
-s	Fehlermeldungen unterdrücken
-r	Durchsucht auch die Unterverzeichnisse

Durchsucht die Datei kuno.txt nach Zeilen mit dem Wort Bruno.

```
grep Bruno kuno.txt
```

Durchsucht alle Dateien im Verzeichnis nach der Zeichenkette `midnight' ohne Berücksichtigung der Groß- und Kleinschreibung und gibt die Namen der Dateien aus, die die Zeichenkette enthalten.

```
grep -il midnight *
```

### 1.8.7.2 Das Kommando egrep

- Das Kommando `egrep` entspricht dem Befehl `grep -E`. Es verwendet erweiterte reguläre Ausdrücke.

### 1.8.7.3 Das Kommando tr

- Das Kommando `tr` sendet die Standardeingabe zur Standardausgabe, wobei es mehrere Operationen auf die Daten ausführt.
- Mögliche Operationen sind:
  - Verändern von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.
  - Löschen von Zeichen, die wiederholt werden.
  - Löschen von Zeichen.
  - Löschen von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.
- `tr [OPTIONEN] ZEICHENKETTE1 [ZEICHENKETTE2]`

#### Optionen

- d Löscht die angegebenen Zeichen
- s Löscht doppelt vorkommende Zeichen

So wandelt `tr` mit dem Befehl

```
cat stundenplan.txt | tr m M > stundenplan.neu
```

jedes Vorkommen des Buchstabens ``m`` in den Buchstaben ``M`` um.

Das folgende Kommando löscht die Zeichen ``m`` und ``y``.

```
cat etwas.txt | tr -d my
```

Doppelt vorkommende Zeichen werden mit dem Schalter `-s` gelöscht.

```
tr -s le < harry.txt
```

Dabei wird dann aus der ``Allee`` ganz schnell ``Ale``.

Um den einfachen ROT13 Verschlüsselungsalgorithmus zu verwenden, reicht die folgende Sequenz.

```
cat harry.txt | tr ' [A-M] [N-Z] [a-m] [n-z] ' ' [N-Z] [A-M] [n-z] [a-m] '
```

oder

```
cat harry.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
```

Dabei wird der erste Buchstabe mit dem 13. Buchstaben, der zweite mit dem 14. Buchstaben, u. s. w. getauscht. Wendet man ROT13 wieder auf verschlüsselten Text an, so erhält man den Originaltext.

### 1.8.7.4 Das Kommando expand

- Das Kommando `expand` liest aus Dateien oder von der Standardeingabe, wandelt die Tabzeichen in Leerzeichen und gibt das Ergebnis auf der Standardausgabe aus.
- Wird nichts anderes angegeben, dann wird ein Tabulator durch acht Leerzeichen ersetzt.
- `expand [OPTIONEN] [DATEILISTE]`

#### Optionen

- TABx ist die Anzahl der Leerzeichen, durch die das TAB1[,TAB2[,...]] jeweilige Tabulatorzeichen ersetzt werden soll.
- t TAB1[,TAB2[,...]]

-i                    Tabulatoren ohne vorhergehende Leerzeichen werden nicht konvertiert.

- Wird nur für das erste Tabulatorzeichen ein Wert angegeben, so gilt dieser Wert für alle Tabulatoren.
- Werden zwei oder mehr Werte eingegeben, so gelten Sie für die jeweiligen Tabulatoren.
- Tabulatoren, für die kein Wert angegeben wurde, werden durch einfache Leerzeichen ersetzt.

Diese Kommandosequenz ersetzt den ersten Tabulator jeder Zeile der Datei helloworld.java durch 6 Leerzeichen, den zweiten durch 8 Leerzeichen und alle folgenden Tabulatoren durch ein Leerzeichen. Das Ergebnis wird in die Datei helloworld.txt geschrieben.

```
expand -6,8 helloworld.java > helloworld.txt
```

### 1.8.7.5 Das Kommando sed

- `sed` ist die Abkürzung für **Streaming Editor**.
- Eigentlich ist das Kommando kein Editor sondern ein Textfilter, der bestimmte Zeichenkombinationen sucht und ersetzt.
- Die Verarbeitung eines Streams oder Datenstroms bedeutet, daß von der Standardeingabe (stdin) Daten angenommen werden und auf die Standardausgabe (stdout) wieder ausgegeben werden.
- Als Datenquelle kann neben der Standardeingabe auch eine Datei fungieren.
- `sed [OPTIONEN] [KOMMANDO] [DATEILISTE]`
- `sed [OPTIONEN] [-f SCRIPTFILE] [DATEILISTE]`
- Es werden grundsätzlich zwei Möglichkeiten genutzt um `sed` aufzurufen.
- Im ersten Fall wird das `KOMMANDO` auf der Befehlszeile eingegeben und auf die angegebenen Dateien angewendet.
- Im zweiten Fall stehen die Anweisungen in einer externen Skriptdatei.
- Diese wird abgearbeitet und die darin enthaltenen Kommandos auf den Inhalt der Dateien angewendet.
- Sollte keine Datei angegeben sein, so liest `sed` von der Standardeingabe.
- Dabei wird jede gelesene Zeile mit den `sed`-Kommandos bearbeitet und in einen Puffer geschrieben.
- Dessen Inhalt wird dann zum Schluß auf der Standardausgabe ausgegeben.

#### Optionen

-V	Versionsnummer
-h	Hilfe
-e SCRIPT	Zusätzliche Skriptanweisung zur Bearbeitung
-f SCRIPTFILE	Datei, die die Skriptbefehle enthält
-n	Ausgabe erfolgt nur bei Benutzung des `p' Kommandos

- Jeder Befehl kann durch einen Bereich, für den er gültig ist, eingeschränkt werden. Die Bereichsangabe erfolgt im Format
- `VON`  
oder  
`VON, BIS`
- Dabei können die Werte für `VON` und `BIS`
  - die Zeilennummern,
  - das Zeichen \$ für das Ende der Datei oder

- ein Suchmuster, begrenzt durch das Zeichen /, z. B. /unix/, sein.

#### 1.8.7.5.1 Ersetzen

- Die folgenden Sequenzen weisen sed an, eine Zeichenkette durch eine andere zu ersetzen.

```
sed s/SUCHEN/ERSETZEN/
sed s/SUCHEN/ERSETZEN/g
sed s/SUCHEN/ERSETZEN/p
sed -n s/SUCHEN/ERSETZEN/gp
sed s3/SUCHEN/ERSETZEN/
```

- Die Zeichenkette zwischen dem ersten und dem zweiten Schrägstrich gibt das Suchmuster an, während die Zeichenkette zwischen dem zweiten und dem dritten Schrägstrich den Ersetzungstext enthält.
- Um das Suchmuster effektiver zu gestalten, können reguläre Ausdrücke verwendet werden. Diese bestehen normalerweise aus zwei Komponenten: Die Angabe, nach welchem Zeichen gesucht wird, und die Angabe, wie oft das Zeichen auftreten darf.
- Die hinten angestellten Optionen verändern die Arbeitsweise von sed.
- Dabei bewirkt g, daß alle Zeichenketten in der Zeile durch die neue Zeichenkette ersetzt werden (Sonst nur die erste Zeichenkette).
- Das p führt dazu, daß die Zeilen ausgegeben werden, in denen eine Ersetzung erfolgte.
- Sind beide Optionen gesetzt, so wird bei einer mehrfachen Ersetzung in einer Zeile die Zeile auch mehrfach angezeigt.
- Verhindert wird dies durch den Schalter -n.
- Das s3 sorgt dafür, daß das dritte Vorkommen von SUCHEN in der Zeile ersetzt wird.

#### 1.8.7.5.2 Suchen

- Die Sequenz

```
/SUCHEN/p
```

- im Zusammenhang mit dem Schalter -n bewirkt, daß nur die Zeilen, die das Suchmuster enthalten, ausgegeben werden. (vgl. grep)

#### 1.8.7.5.3 Zeilen löschen

- Wenn der Schalter -n nicht gesetzt ist, werden alle Zeilen nach der Bearbeitung ausgegeben.
- Um zu verhindern, daß bestimmten Zeilen ausgegeben werden, wird die Sequenz

```
/SUCHEN/d
```

- verwendet.

#### 1.8.7.5.4 Dateien einfügen

- Durch die Sequenz

```
/SUCHEN/r DATEINAME
```

- werden alle Zeilen, die das Suchmuster enthalten, durch den Inhalt der Datei DATEINAME ersetzt.

#### 1.8.7.5.5 Auswahl in Datei schreiben

- Durch die Sequenz

```
/SUCHEN/w DATEINAME
```

- werden alle Zeilen, die das Suchmuster enthalten, in die Datei DATEINAME geschrieben.

### 1.8.7.5.6 Beispiele

Ersetzt jedes Wort „UNIX“ durch das Wort „Linux“ in der Datei einleitung.tex.

```
sed 's/UNIX/Linux/g' einleitung.tex
```

Ersetzt jedes Wort „paragraph“ durch das Wort „subsection“ in der Datei linux.tex und schreibt das Ergebnis in die Datei linuxneu.tex.

```
sed 's/paragraph/subsubsection/g' linux.tex > linuxneu.tex
```

Einen interessanten Effekt hat das Zeichen '&' in der Ersetzungszeichenkette. Die gesuchte Zeichenkette wird nicht ersetzt, sondern die Ersetzungszeichenkette wird hinten angefügt.

```
sed 's/</&lt;/g' index.html
```

Damit die schließende spitze Klammer ersetzt wird, muß das kaufmännische Und auskommentiert werden.

```
sed 's/</\&lt;/g' index.html
```

Ersetzt alle Zeichenfolgen „mann“ durch „frau“ in den Zeilen 1 bis 3 in der Datei einleitung.tex. Dabei werden nur die Zeilen ausgegeben, in denen die Änderung erfolgte.

```
sed -n '1,3s/mann/frau/p' einleitung.tex
```

Gibt alle Zeilen aus, die entweder „man“ oder „frau“ enthalten.

```
sed -n -e '/man/p' -e '/frau/p' einleitung.tex
```

Das folgende Kommando löscht alle Zeilen, die die Zahl „0“ enthalten.

```
sed '/0/d' einleitung.tex
```

Alle Zeilen, die die Zeichenkette „include“ enthalten, werden durch den Inhalt der Datei include.txt ersetzt.

```
sed '/include/r include.txt' einleitung.txt
```

Alle Zeilen, die die Zeichenkette „section“ enthalten, werden in die Datei inhalt.txt geschrieben.

```
sed '/section/w inhalt.txt' einleitung.tex
```

### 1.8.7.6 Das Kommando diff

- Das Kommando diff vergleicht zwei Textdateien miteinander.
- Als Ergebnis wird eine Liste aller Zeilen ausgegeben, die voneinander abweichen.
- Dabei erkennt das Programm auch eingefügte Zeilen und arbeitet danach reibungslos weiter.
- Im Gegensatz zu comm kann diff auch auf unsortierte Dateien angewendet werden.
- Der Befehl wird hauptsächlich dazu verwendet die Abweichungen zwischen zwei Versionen eines Programmlistings rasch zu dokumentieren.
- `diff [OPTIONEN] DATEI1 DATEI2`

#### Optionen

-b	Betrachtet mehrfache Leerzeichen und Leerzeilen als einfache Leerzeichen bzw. Leerzeilen
-c	Ausgabe als 'Context Diffs'
-e	Ausgabe als Skript für den Befehl ed
-r	Vergleicht den Inhalt zweier Verzeichnisbäume
-u	Ausgabe als 'Unified Diffs'
-w	Ignoriert Leerzeichen und Leerzeilen ganz

- Die Ausgabe von diff besteht aus sogenannten 'hunks', was man frei als 'Stücke' übersetzen kann.
- Ein Hunk besteht aus bis zu vier Teilen: einer Informationszeile, dem alten Text, einer Trennzeile und dem neuen Text.
- Beim Einfügen und Löschen von Zeilen fehlt die alte oder neue Textinformation und die dann überflüssige Trennzeile fällt auch weg.
- In der Informationszeile steht, an welcher Stelle in der jeweiligen Datei die Änderung stattgefunden hat und welche Operationen nötig sind um die erste Eingabedatei DATEI1 in die zweite Eingabedatei DATEI2 zu verwandeln.

- Dabei stehen die Buchstaben 'a', 'c' und 'd' für 'add' (Zeilen hinzufügen), 'change' (Zeilen verändern) und 'delete' (Zeilen löschen).
- Die Zahlen bzw. Zahlenbereiche links und rechts von den Buchstaben geben die betroffenen Zeilen an
- Alle Hunks zusammengenommen bilden den Patch, der die erste Datei in die zweite Datei verwandelt.
- Es geht auch umgekehrt, da Patches dieser Art symmetrisch sind und rückwärts angewendet werden können ('reverse patching');

Um zwei Perlscripte miteinander zu vergleichen, ohne mehrfache Leerzeichen zu berücksichtigen, gibt man das folgende Kommando ein.

```
diff -b div.pl.old div.pl
```

Soll die Ausgabe für den Editor ed verarbeitbar sein, dann muß das Kommando lauten:

```
diff -be div.pl.old div.pl
```

## 1.9 Rechte im Unix Dateisystem

- Basierend auf der Art, wie auf Dateien zugegriffen werden kann, unterscheidet Linux drei Rechte.
- Sie werden abkürzend mit r, w oder x bezeichnet.
- Die einzelnen Rechte sind an die jeweilige Datei gebunden und werden in der Inode der Datei gespeichert.
- Dabei ist besonders wichtig in welcher Stellung Sie sich zu der Datei befinden.
- Sind Sie der Besitzer, gehören Sie zur Gruppe, die Zugriff auf die Datei hat, oder gehören Sie zum Rest der Welt.
- Wie vom Befehl ls -l bekannt, besteht die Rechttabelle aus 9 Angaben, z. B. rwxr--xr--.
- Die ersten drei angegebenen Rechte gelten dabei für den Dateibesitzer, die folgenden drei für die Besitzergruppe und die letzten für den Rest der Welt.

### 1.9.1 Rechte auf Dateien

- Die Wirkung der Rechte ist bei Dateien und Verzeichnissen unterschiedlich.
- Bei Dateien arbeiten Sie wie folgt:

**r (read = lesen)** Der Benutzer kann den Inhalt der Datei einsehen, d. h. unter anderem, er kann sie am Bildschirm anzeigen lassen, sie drucken oder kopieren.

**w (write = schreiben)** Der Benutzer kann die Datei verändert unter dem bisherigen Namen speichern.

**x (execute = ausführen)** Die Datei kann als Programm gestartet werden. Dies setzt natürlich voraus, daß die Datei ein Programm ist.

Angenommen also, Sie haben an der Datei tagebuch kein r-Recht, wollen Sie aber trotzdem auf dem Bildschirm sehen. Die Eingabe von

```
cat tagebuch
```

führt dann zu einer Fehlermeldung der Art:

```
tagebuch: permission denied
```

- Viele Anwender, die aus einer Einplatzumgebung zu Linux stoßen, überschätzen allerdings die Bedeutung der Rechte, die man an einer (gewöhnlichen) Datei hat.
- Dies mag daran liegen, das MS-DOS und viele seiner Verwandten den Zugriff auf Dateien durch Attribute zu regeln versuchen. Die vielzitierte „schreibgeschützte Datei“ gibt es unter Linux nicht.
- Der Dateischutz wird ausschließlich über die Rechte gewährleistet.
- Hierbei gilt eine glasklare Logik, die sich im wesentlichen aus dem Dateisystem ergibt:
- Wird in einer Befehlszeile ein Pfad angegeben, auf den der Befehl zugreifen soll, werden die einzelnen Bestandteile des Pfades daraufhin überprüft, ob hinreichende Rechte bestehen.
- Ist dies nicht der Fall, wird der Zugriff verweigert.

### 1.9.2 Rechte auf Verzeichnisse

- Verzeichnisse sind für Linux nur eine spezielle Art von Dateien.

- Dies macht sich hier daran bemerkbar, daß die Rechte `r` und `w` für Verzeichnisse genauso gehandhabt werden, wie für Dateien.
- Daraus ergeben sich weitreichende Konsequenzen für die darin enthaltenen Dateien!
- Das `r`-Recht an einem Verzeichnis zu besitzen heißt, man kann den Inhalt des Verzeichnisses lesen, mit anderen Worten, man kann den Befehl `ls` für dieses Verzeichnis einsetzen.
- Was aber ist nun der Inhalt eines Verzeichnisses?
- Ein Verzeichnis ist nichts anderes als eine kleine Tabelle. Jede Zeile dieser Tabelle hat zwei Einträge:
  - Den Dateinamen und die zugehörige Inode-Nummer.
- Wie Sie bereits wissen, ist der Dateiname für Linux nicht das entscheidende Kriterium für die Bearbeitung einer Datei.
- Linux orientiert sich an der Inode.
- Was bedeutet in diesem Zusammenhang das `r`-Recht für ein Verzeichnis?
- Ganz einfach: Mit dem `r`-Recht am Verzeichnis können darin Dateien gefunden werden (der Befehl `ls` liefert die Namen der im Verzeichnis enthaltenen Dateien), zusätzlich kann man die Inode-Nummern dieser Dateien abfragen (`ls -li`).
- Da in der Inode selbst die Verwaltungsinformationen (darunter die Rechtetabelle für die Datei) gespeichert sind, ist ein Zugriff auf eine Datei also erst dann möglich, wenn man ihre Inode kennt!
- Das heißt also, wenn Ihnen das `r`-Recht an einem Verzeichnis fehlt, haben Sie keinen Zugang zu den darin enthaltenen Dateien!
- Insbesondere sind auch Unterverzeichnisse dem Zugriff des Anwenders entzogen!
- Das für das Verständnis der Linux-Rechte wichtigste Recht ist wohl das `w`-Recht für Verzeichnisse.
- Dieses Recht beinhaltet die Möglichkeit ein Verzeichnis verändert abzuspeichern.
- Anders ausgedrückt: Ein Benutzer, der an einem Verzeichnis das `w`-Recht hat, kann dieses verändern.
- Nun ist wie gesagt ein Verzeichnis eine Liste von Dateinamen (plus zugehöriger Inode-Nummer).
- Besitzen Sie das `w`-Recht an einem Verzeichnis, können Sie einen Verzeichniseintrag löschen, das Verzeichnis wird verändert gespeichert, der Dateiname und die Inode-Nummer existieren nicht mehr.
- Einen Verzeichniseintrag löschen heißt also den Zugang (Link) zu einer Datei löschen! Dies geschieht allerdings nicht mit einem Editor sondern mit Hilfe des Befehls `rm`.
- Verzeichnisse sind insofern „schreibgeschützt“, als Linux aus Konsistenzgründen schreibende Zugriffe auf Verzeichnisse nur mit dafür geeigneten Befehlen erlaubt.
- Das Bearbeiten mit einem Editor würde viele Benutzer überfordern, denn es muß Rücksicht auf die Struktur des Verzeichnisses genommen werden.
- Wird diese aber zerstört, sind die Daten im Verzeichnis nicht mehr verfügbar!
- Nehmen wir nun einen extremen Fall an.
- Eine Datei, an der Sie keinerlei Rechte besitzen, befindet sich in einem Verzeichnis, in dem Sie das `w`-Recht haben.
- Diese Datei können Sie, auch ohne Rechte an ihr selbst, löschen, denn Sie können das Verzeichnis manipulieren!
- Sie entfernen mit dem Befehl `rm` den Eintrag der Datei im Verzeichnis und somit löschen Sie den Zugang zu den Daten
- Gleiches gilt für das Umbenennen (ändert den Eintrag in einem Verzeichnis), Verschieben (löscht den Eintrag in einem Verzeichnis) oder Erstellen (erzeugt einen neuen Eintrag in einem Verzeichnis) einer Datei.
- Auch hierzu ist das `w`-Recht am Verzeichnis von entscheidender Bedeutung.
- Wie Sie sehen hängen sehr viele Dateioperationen von Verzeichnisrechten ab.
- Vereinfachend gesagt, Verwaltungsoperationen mit Dateien, wie z. B. Löschen, Verschieben oder Umbenennen sind Eingriffe in die Verzeichnisstruktur und werden deshalb über Rechte an Verzeichnissen geregelt.
- Die inhaltliche Manipulation von Dateien ist dagegen nur möglich, wenn Sie Rechte direkt an der Datei haben.
- Es kann sein, daß Sie eine Datei löschen, umbenennen oder verschieben können, ohne ihren Inhalt auch nur sehen zu können.

- Zur Manipulation der Dateiinhalte benötigt Sie zumindest das r-Recht an ihr, alle anderen genannten Operationen erfordern das w-Recht am Verzeichnis.
- Alles in allem kann man feststellen, daß Verzeichnisse bezüglich der Rechte r und w wie gewöhnliche Dateien behandelt werden.
- Da man aber mit einem Verzeichnis eine Liste von Dateinamen bearbeitet, bedeutet das Ändern von Verzeichnissen sehr häufig ein Löschen, Umbenennen oder Verschieben von Dateien.
- Bis jetzt haben sich Verzeichnisse wie Dateien verhalten.
- Beim x-Recht gibt es aber einen signifikanten Unterschied. x steht für das Recht ein Programm ausführen zu können. In diesem Sinne kann also ein Verzeichnis nicht ausführbar sein.
- Das x-Recht an einem Verzeichnis bedeutet nun, daß Sie
  - mit cd in das Verzeichnis wechseln dürfen und
  - Zugang zu den Inodes aller darin enthaltenen Dateien erhalten.
- Rechte an einem Verzeichnis betreffen die darin enthaltenen Dateien.
- Wollen Sie sich den Inhalt der Datei `/home/htor/.profile` mit `cat /home/htor/.profile` ansehen, so setzt dies folgende Rechte voraus:
  - Das r-Recht an der Datei selbst.
  - Das r-Recht am Verzeichnis `/home/htor`
  - Das x-Recht an diesem Verzeichnis.
- cat muß zunächst den Inhalt von `/home/htor` lesen, um zu prüfen, ob die Datei überhaupt existiert.
- Dies erlaubt das r-Recht am Verzeichnis.
- Weiter benötigt cat die Inode-Nummer der Datei, um über diese die Inode selbst finden, Zugang dazu erhält cat über das x-Recht an `/home/htor` In der Inode sind nun aber alle relevanten Informationen, die cat braucht, um den Dateiinhalt endlich lesen zu können.
- Hier ist die Rechtetabelle, die über das r-Recht die Erlaubnis zum Lesen gewährt und schließlich sind hier die Informationen abgelegt, mit deren Hilfe die Daten auf der Platte gefunden werden können.
- In der Praxis erhält man mit dem r-Recht an einem Verzeichnis stets auch das x-Recht. Natürlich kann man das x-Recht wieder nehmen (sofern man dazu die Erlaubnis hat), dann schafft man aber eine skurile Situation:
  - Das Leserecht am Verzeichnis erlaubt es dann zum Beispiel ls, Dateien zu finden, weitere Informationen werden aber verweigert. Man erhält in jedem Fall eine Fehlermeldung wie die folgende
  - `/bin/ls: /home/htor/.profile: Permission denied`
  - ls meldet, daß man auf die Inode von `.profile` im Verzeichnis `/home/htor` nicht zugreifen darf: `Permission denied.`
- An den meisten Verzeichnissen im Dateisystem hat der gewöhnliche Benutzer die Rechte r und x, am Heimatverzeichnis aber alle Rechte.
- Neben den Abkürzungen r, w und x gibt es auch noch eine oktale Notation der Rechte.
- Dies ist eine dreistellige Zahl, bei der die erste Ziffer für den Besitzer gilt, die zweite für die Gruppe, die dritte für alle anderen.

Ziffer	Bedeutung
4	r-Recht
2	w-Recht
1	x-Recht
0	kein Recht

- Die Rechtekombinationen ergeben sich durch Addition! So bedeutet 754 dasselbe wie
  - rwxr-xr--
- Anstelle der Kennzeichnung x treten hin und wieder ein s oder ein t auf.

### 1.9.3 Spezielle Rechte suid und sgid

- SUID (*Set User ID*) und SGID (*Set Group ID*) werden durch ein s anstelle eines x bei den Eigentümer- und bei den Gruppenrechten ausgedrückt.
- Das s-Recht an einer ausführbaren Datei bedeutet, daß der Benutzer, der es startet, während des Programmlaufes die UID des Dateibesitzers bzw. die GID der Besitzergruppe erhält.
- Das hängt davon ab, ob das s beim Besitzer oder der Besitzergruppe steht.
- Im Falle des Programms `/bin/passwd` wird man z. B. während des Programmlaufs zum Superuser!

```
htor@archimedes:~> ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root 921 2003-03-08 17:23 /etc/passwd
-rw-r----- 1 root shadow 631 2003-03-09 11:40 /etc/shadow
htor@archimedes:~> ls -l /usr/bin/passwd
-rwsr-xr-x 1 root shadow 68680 2002-10-14 22:40 /usr/bin/passwd
htor@archimedes:~>
```

- Wie Sie sehen, hat nur der Superuser Schreibrechte auf die Dateien `/etc/passwd` und `/etc/shadow`.
- Das s-Recht auf den Befehl `/usr/bin/passwd` bewirkt, daß der Befehl immer unter der UID des Superuser ausgeführt wird.
- So kann auch ein normaler Benutzer sein Kennwort ändern.
- Wird das SGID-Recht auf ein Verzeichnis gegeben, so hat es eine andere Funktion.
- In diesem Fall erhalten alle Dateien, die in diesem Verzeichnis erstellt werden, automatisch die gleiche Gruppe, die auch dem Verzeichnis zugeordnet ist.

### 1.9.4 Verwalten und Setzen von Rechten

- Das Verwalten und Setzen von Rechten übernehmen Programme wie `chown`, `chmod`, `chgrp` und `umask`.

#### 1.9.4.1 Das Kommando chown

- Mit dem Befehl `chown` ist es möglich der Datei einen neuen Besitzer und eine neue Gruppe zu geben.
- `chown [OPTIONEN] [BENUTZER] [:GRUPPE] DATEILISTE`
- Allerdings ist es nur dem Superuser gestattet, den Eigentümer einer Datei zu ändern.

#### Optionen

-c	Informationen über alle geänderten Dateien werden angezeigt
-v	Informationen über alle Aktionen werden ausgegeben
-f	Fehlermeldungen werden nicht ausgegeben
-R	Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

#### 1.9.4.2 Das Kommando chgrp

- Ändert für eine Datei die zugehörige Gruppe.
- `chgrp [OPTIONEN] GRUPPE DATEILISTE`
- Die Änderung der Gruppe ist für den Besitzer und für den Superuser erlaubt.
- Der Besitzer darf allerdings seiner Datei nur eine Gruppe zuordnen, in der er selber Mitglied ist.

### Optionen

- c Informationen über alle geänderten Dateien werden angezeigt
- v Informationen über alle Aktionen werden ausgegeben
- f Fehlermeldungen werden nicht ausgegeben
- R Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

```
chgrp users pampelmuse
```

ändert die Gruppe der Datei pampelmuse auf **users**.

#### 1.9.4.3 Das Kommando chmod

- Der Befehl chmod erlaubt das Ändern der Rechte für eine Datei.
- `chmod [OPTIONEN] RECHTE DATEILISTE`
- Die Rechte können dabei in der Form *WerWieWas* oder *Maske* angegeben werden.
- *Wer* ist dann u (user/Besitzer), g (group/Gruppe), o (others/Sonstige), eine Kombination daraus oder a (all user/alle Benutzer).
- *Was* ist eines der Rechte r, w, x, s, t oder eine Kombination daraus.
- Wobei sich x, s und t sich gegenseitig ausschließen.
- *Wie* ist = (Rechte werden exakt so gesetzt wie in *Was* angegeben), + (Rechte aus *Was* werden zusätzlich vergeben) oder - (Rechte aus *was* werden entzogen)
- *WerWieWas* kann mehrfach, dann durch Komma getrennt angegeben werden.
- In der ganzen Liste der *WerWieWas* darf kein Leerzeichen sein, da alles nach diesem Leerzeichen als Dateiname interpretiert wird
- *Maske* ist eine dreistellige (Superuser auch vierstellige) oktale Zahl

### Optionen

- c Informationen über alle geänderten Dateien werden angezeigt
- v Informationen über alle Aktionen werden ausgegeben
- f Fehlermeldungen werden nicht ausgegeben
- R Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

```
chmod u+x memo*.txt
```

fügt für den Besitzer das x-Recht hinzu.

```
chmod 755 makeitso
```

setzt die Rechte auf rwxr-xr-x.

```
chmod a-rwx,u=rwx meinsein
```

entzieht allen alle Rechte und gibt dem Besitzer alle Rechte.

#### 1.9.4.4 Das Kommando umask

- Für die Erstellung neuer Dateien gibt es für die Rechte eine Maske, die bestimmt welche Rechte vergeben werden.
- Für diese Maske ist der Befehl umask (*User's creation MASK*) zuständig.
- `umask [MASKE]`
- Die Eingabe von umask ohne Parameter gibt die aktuell eingestellte Maske wieder.
- Die Maske selber ist eine dreistellige oktale Zahl.
- Die Bedeutung der Zahlen ist identisch mit derer für die Rechte
- Allerdings gibt umask nicht an, welche Rechte gegeben werden, sondern welche Rechte entzogen werden
- Allerdings verhält sich die Rechtevergabe doch anders als erwartet.

- Manche Systeme liefern auch eine vierte Ziffer (an erster Stelle stehend) zurück.
- Diese beschreibt das besondere Verhalten der Rechtevergabe für neue Dateien
- Dabei verhält sich die bash wie durch den Wert 0 gegeben.
- Die für die Benutzergruppen angegebenen Rechte werden von dieser Grundeinstellung abgezogen.

Tabelle: Besondere Berechtigungen für umask

Wert	Bedeutung
0	rw für Dateien, rwx für Verzeichnisse
1	rw für Dateien und Verzeichnisse
2	r für Dateien, rx für Verzeichnisse
3	r für Dateien und Verzeichnisse
4	w für Dateien, wx für Verzeichnisse
5	w für Dateien und Verzeichnisse
6	x für Dateien und Verzeichnisse
7	Keine Rechte für Dateien und Verzeichnisse

- Beim Kopieren von Dateien hat umask auch seine Finger im Spiel. Als Vorgabe gelten die Rechte, die die Quelldatei besitzt. Diese werden dann mit der Einstellung von umask gefiltert.
- Der durch umask eingestellte neue Wert gilt nur für die Dauer einer Sitzung.
- Eine permanente Änderung erreichen Sie, wenn Sie den Befehl umask in die Konfigurationsskripte `.profile` oder `.bashrc` eintragen.

```
umask 022
```

setzt die Defaultmaske auf `rw-r-xr-x`.

## 1.10 Installation SuSE Linux

- Damit wir mit Linux arbeiten können ist es notwendig das Betriebssystem zu installieren.
- Die Installation ist stark abhängig von der verwendeten Distribution.
- Bei der SuSE-Distribution kann eine Installation von CD, Festplatte oder über NFS oder FTP erfolgen.
- Dabei können die Installations- und Administrationstools YaST 1 und Yast 2 verwendet werden.
- Seit SuSE 8.0 befindet sich das für ein Textterminal optimierte Yast 1 nicht mehr auf der Distribution.
- Das für graphische Oberflächen entwickelte Yast 2 besitzt mehr Möglichkeiten.
- Es kann auch unter einer nichtgraphischen Oberfläche eingesetzt werden, ist dort aber nicht sehr handlich.
- Da auch andere Benutzer das gleiche Problem hatten, wurde in SuSE 8.1 die Bedienung von Yast im Textmodus deutlich verbessert.

### 1.10.1 Booten von CDROM

Hierzu muss das CDROM-Laufwerk des Rechners als Boot-Laufwerk im BIOS eingestellt werden!

1. Starten Sie den Rechner und legen Sie die CD 1 oder die DVD während der POST-Phase ein.
2. Beim jetzt erscheinenden Auswahlmenü wählen Sie Installation. Sollte die Installation nicht klappen, da Treiber z. B. für die SCSI-Festplatte fehlen, muß die "Manuelle Installation" ausgewählt werden. Die wird wie oben beschrieben durchgeführt.
3. Nun wird der Kernel geladen und das System gestartet.

4. Danach erscheint das eigentliche Installationsmenü.

### 1.10.2 Konfiguration mit Yast2

- Im Gegensatz zu Windows, wo die Einstellungen des Systems in der nicht gerade übersichtlichen Registry gespeichert werden, existieren unter Linux viele Textdateien, die die Konfigurationswerte enthalten.
- So ist es möglich nur mit einem Texteditor das komplette System zu administrieren.
- Obwohl diese Konfigurationsdateien normalerweise gut dokumentiert sind, gehört doch etwas Wissen und Selbstvertrauen dazu, die Einstellung per Hand vorzunehmen.
- SuSE hat für die Administration das Tool YaST2 für die graphische Oberfläche entwickelt.
- Es erlaubt eine wesentlich komfortablere aber nicht unbedingt schnellere Administration des Systems.
- Auch normale Benutzer können die YaST2 Kontrollzentrum oder die einzelnen Module aufrufen.
- Vor der Ausführung wird aber nach dem Passwort für *root* geführt.
- Daher müssen Sie sich nicht extra als *root* anmelden um Ihr System administrieren zu können.
- Im YaST-Kontrollzentrum finden Sie die Module in verschiedenen Gruppen zusammengefaßt.
- Software
  - Dieser Abschnitt beschäftigt sich mit der Installation, Deinstallation und Update der Software auf dem System.
- Installationsquelle wechseln
  - YaST merkt sich die letzte Installationsquelle. Erfolgt nun die Installation von einem anderen Medium, so kann dies hier eingestellt werden.
- Patch CD-Update
  - Für bestimmte Kunden stellt SuSE in regelmäßigen Abständen CDs mit aktuellen Patches zur Verfügung. Hier können Sie eingespielt werden.
- Update des Systems
  - Über dieses Modul können Sie das bestehende System auf einen neuen Distributionsstand bringen, z. B. von SuSE 8.0 auf SuSE 8.1.
- Online-Update
  - Fehlerbehebung ist ein wichtiger Faktor in der EDV. Mit dem Online-Update können Sie kleine Programme zur Fehlerkorrektur (*Patches*) Online herunterladen und Ihr System auf den neuesten Stand bringen.
- Software installieren/löschen
  - Dieses Modul ermöglicht es Ihnen Pakete von der SuSE-Distribution einzuspielen, upzudaten oder wieder vom System zu entfernen.
- Hardware
  - Wie der Namen schon erahnen läßt finden Sie hier die Möglichkeit die Hardware zu konfigurieren oder Informationen über Sie einzuholen. Spezielle für Drucker, Grafikkarte und Monitor, Festplatte, Scanner, Maus, Soundkarte und TV-Karte gibt es hier Module.
- Automatische Druckererkennung
  - für die Installation lokaler Drucker.
- Drucker bearbeiten
  - Mit diesem Modul können existierende Drucker bearbeitet werden und neue lokale und Netzwerkdrucker eingerichtet werden.
- Grafikkarte und Monitor
  - Hiermit können Sie die Einstellung für die Grafikkarte ändern und die Ausgabe an den Monitor anpassen.
- Hardware-Info
  - Liefert Ihnen eine Übersicht über die eingebaute Hardware.
- IDE DMA-Modus
  - Hiermit können Sie den DMA-Modus (Direct Memory Access) Ihrer Laufwerke aktivieren. Dies ist z. B. für das Betrachten von DVDs wichtig.
- Mausmodell wählen
  - Was soll ich da noch viel schreiben ...

- Partitionieren
  - Hier gelangen Sie zu dem Tool, bei dem Sie schon bei der Installation die Festplatten Ihres Rechners in einzelne Abschnitte aufteilen können.
- Scanner
  - Auch Scanner wollen konfiguriert werden ...
- Sound
  - Bei der Soundkonfiguration sollten Sie die passende Lautstärke beachten und richtig einstellen. Aus Sicherheitsgründen ist die Grundeinstellung meistens sehr leise. Dies und ein sehr leise gestellter Lautsprecher haben schon oft zur falschen Annahme geführt, daß die Soundkarte unter Linux nicht funktioniert.
- TV-Karten
  - Seit SuSE 8.0 hat sich die Unterstützung von TV-Karten sehr verbessert. Hier werden Sie automatisch erkannt und konfiguriert.
- Netzwerk/Basis
  - Die Konfiguration von Netzwerkkomponenten (Netzwerkkarten, Modems, ISDN-Karten, etc.) oder Diensten (E-Mail, DSL, Serverdienste) wird durch die Module dieser Gruppe erledigt.
  - E-Mail
  - Konfiguration der Netzwerkkarte
  - Konfiguration des Modems
  - Konfiguration von DSL
  - Konfiguration von ISDN
  - Start oder Stopp von Systemdiensten
- Netzwerk/Erweitert
  - Die speziellen Netzwerkfunktionen für DNS, NFS, NIS und Routing werden hier konfiguriert.
  - HostName und DNS
  - LDAP-Client
  - NFS-Client
  - NFS-Server
  - NIS+-Client
  - NIS-Client
  - NIS-Server
  - Routing
- Sicherheit & Benutzer
  - Benutzerverwaltung und die Richtlinien zur Sicherheit (Firewall) werden hier abgehandelt.
  - Benutzer bearbeiten und anlegen
  - Einstellungen zur Sicherheit
  - Firewall
  - Gruppen bearbeiten und anlegen
  - Neue Gruppe anlegen
  - Neuen Benutzer anlegen
- System
  - Diese Module kümmern sich um die systemnahen Einstellungen, wie Backup, Bootdisketten, Starteinstellungen, Automatisches Starten von Diensten und Grundkonfigurationen.
  - Backup des Systems erstellen
  - Boot-, Rettungs- oder Moduldiskette erzeugen
  - Konfiguration des Bootloaders
  - LVM
  - Runlevel-Editor
  - Sprache wählen
  - Sysconfig-Editor
  - Tastaturlayout auswählen
  - Zeitzone auswählen
- Sonstiges
  - Hier finden Sie den Rest der YaST-Module, die SuSE nicht in eine der anderen Gruppen einordnen konnte.
- Drucker für CUPS

- Konfiguriert Drucker für das Drucksystem CUPS
- Drucker für LPD
  - Konfiguriert Drucker für das Drucksystem LPD
- Startprotokoll anzeigen
  - Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/boot.msg` gestartet wird.
- Support-Anfrage stellen
  - Hier können Sie eine Anfragen an den Online-Support stellen.
- Systemprotokoll anzeigen
  - Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/messages` gestartet wird.
- Treiber-Cd des Herstellers laden
  - Ermöglicht die Installation von Treibern, die vom Hersteller der Hardware kommen.

Der folgende Abschnitt zeigt Ihnen die Administration einiger Funktionen. Diese ist im Prinzip über YaST sehr einfach. In den meisten Fällen muß man nur wissen was man will und man muß lesen können.

### 1.10.2.1 Wie installiere ich weitere Softwarepakete

- Das es um Software geht, wählen Sie im YaST-Kontrollzentrum die Gruppe *Software*.
- Ein Blick auf die Liste der Module zeigt uns, daß Software installieren/löschen wohl eine gute Wahl ist.
- Falls Sie die Installationsquelle gewechselt haben, müssen Sie diese vorher mit dem Modul *Installationsquelle wechseln* neu einstellen.
- Nach dem Starten des Moduls sucht YaST nach der Paketliste.
- Dies kann einen Moment dauern.
- Findet YaST kein Installationsmedium, so behilft er sich mit der letzten gespeicherten Liste.
- **Achtung!** Installation über NFS: Sollte sich YaST bei dieser Phase aufhängen und nicht weitermachen, dann liegt es in den meisten Fällen daran, daß der NFS-Server nicht erreichbar ist. Da es sich um eine harte Verbindung handelt, dauert es sehr lange bis es zu einem Timeout kommt und YaST den Fehler bemerkt. Um diesen Fehler zu umgehen, wählen Sie vorher ein anderes Installationsmedium aus.
- Nach dem Laden der Paketlisten erscheint das Auswahlfenster für die Pakete.
- Sie sehen dort drei Tabellen.
- Links oben zeigt Ihnen die Themengruppen oder Paketserien an, in die die Pakete einsortiert worden sind.
- Unten links sehen Sie, wieviel Platz noch auf der Festplatte vorhanden ist.
- Die rechte Tabelle zeigt die Pakete in der Gruppe an.
- Durch Doppelklicken oder Betätigen des Buttons *Aus-/Abwählen* können Sie den Zustand des Pakets ändern.
- Dieser Zustand wird durch ein Zeichen vor dem Paketnamen angezeigt.
- Dabei steht ein **X** für ein Paket, das zur Installation vorgesehen ist.
- Das kleine **i** steht für ein bereits installiertes Paket und das kleine **d** für ein installiertes Paket, daß deinstalliert werden soll.
- Auch ein kleines **a** kann als Markierung vorkommen.
- Diese Pakete wurden automatisch ausgewählt zur Installation, da diese von anderen Paketen benötigt werden.
- Keine Markierung bedeutet natürlich, daß dieses Paket nicht installiert und auch nichts für dieses Paket geplant ist.
- Zu den meisten Paketen existiert neben der Kurzbeschreibung in der Tabelle noch eine etwas ausführlichere Beschreibung des Pakets.
- Diese erreichen Sie über den Button *Beschreibung*, was Sie natürlich nie erraten hätten. ;-)
- In vielen Fällen kennt man den Namen des Programms, aber nicht in welchem Paket oder in welcher Themengruppe/Paketserie das Paket liegt.
- In den meisten Fällen heißt das Programmpaket genau wie das Programm.
- Über eine Suchfunktion (Button *Suchen*) können die Paketnamen und sogar die Beschreibungen nach einem Stichwort durchsucht werden.
- Unter *Extras* finden Sie weitere Funktionen um Pakete auszuwählen oder abzuwählen.

- Besonders interessant ist die Möglichkeit, die aktuelle Paketauswahl abzuspeichern oder eine gespeicherte Paketauswahl einzuladen.
- Dadurch ist es relativ schnell möglich Rechner mit der gleichen Softwareausstattung zu installieren.
- Nach der Paketauswahl, die Sie mit dem Button *OK* abschließen, werden die Pakete installiert.
- Bei der CD-Version werden Sie dann und wann aufgefordert die entsprechenden CDs einzulegen.
- Mit einem Fortschrittsbalken werden Sie darüber informiert, wie weit Ihre Installation ist.
- Nach Abschluß der Installation startet YaST das Programm SuSEConfig, daß die Konfigurationsdateien des Systems für die neuen Programmen konfiguriert.

Und schon sind die Programme installiert.

### 1.10.2.2 Online Update

- *Keine Software ist fehlerfrei!* Diesem gerne von einer Firma aus Redmond als Entschuldigung herangezogenen Satz kann ich nur zustimmen.
- Wer selber mal ein längeres Programm geschrieben hat, weiß, wieviel Fehler sich einschleichen können.
- Und wir reden hier von Programmen, deren Sourcecode ausgedruckt mehrere Aktenordner füllen würden.
- Programme *sind* fehlerhaft. Finden Sie sich damit ab.
- Zum Glück gibt es so Leute, die sich nicht damit abfinden und die Fehler in Programmen bereinigen.
- Bei OpenSource-Software kann dies theoretisch jeder.
- Wohlgemerkt: theoretisch.
- Um sich in einem großen Programm zurechtzufinden braucht es schon etwas längere Einarbeitungszeit und gute Kenntnisse in den Programmiersprachen und natürlich der Programmierung selber.
- Jedenfalls werden viele Fehler bzw. Sicherheitslücken im Laufe der Zeit in Programmen entdeckt und oft auch behoben.
- Um nicht das ganze Programm neu einspielen zu müssen, benutzt man sogenannte Patches (Flicken), die den Binärkode des ``alten" Programms auf den neuesten Stand bringen.
- Die Fehlerkorrekturen - manche Leute reden auch von Fehlerveränderungen – können Sie manuell einspielen.
- SuSE bietet Ihnen den Service eines Online-Updates.
- Das YOU (*YaST Online Update*) stellt Ihnen die neuesten Patches zu den SuSE-Paketen zur Verfügung.
- Es stellt fest, was an neuen Patches vorhanden ist.
- Dann lädt es die Patches für die installierten Programme runter und installiert Sie.
- Das YOU finden Sie auch im Kontrollzentrum unter *Software*.
- Nach dem Starten des Moduls sucht das System nach verfügbaren Update-Servern.
- Sie müssen natürlich auch eine Verbindung zum Internet besitzen.
- Sie können nun zwischen manuellem und automatischem Update wählen.
- Entscheiden Sie sich für das manuelle Update, denn dort können Sie besser sehen, was installiert wird.
- Sie können auch den Server wählen, von dem Sie die Patches beziehen möchten. Dies sollten Sie tun, falls der erste Server überlastet ist.
- Nun lädt YOU die Liste der zur Verfügung stehenden neuen Updates herunter.
- Nachdem die Verbindung beendet wurde, geht es mit *Weiter* weiter.
- Sie sehen nun die Liste der zur Verfügung stehen Patches mit Erläuterungen.
- Die Patches, die Sie benötigen sind schon mit einem X gekennzeichnet zum Download vorgesehen.
- Sie können nun einfach mit *Weiter* zum nächsten Schritt, dem Herunterladen der Pakete, gehen.
- Nachdem dieses auch abgeschlossen ist und Sie mit *Weiter* zum nächsten Bildschirm gewechselt sind, kann es nun endlich mit dem Einspielen der Patches losgehen.

- Danach wird noch einmal SuSEconfig gestartet um die Konfigurationsdateien anzupassen und schon ist Ihr System auf dem neuesten Stand.

### 1.10.3 Benutzer- und Gruppenverwaltung

- Linux ist ein Multiuser-Betriebssystem.
- Dadurch ist es mehreren Personen nacheinander oder sogar gleichzeitig möglich auf einem Rechner arbeiten.
- Damit es zu keinem Durcheinander der Daten kommt, muss sich jeder Benutzer eindeutig identifizieren, wenn er mit Linux arbeiten möchte.
- Außerdem gehört jeder Benutzer mindestens einer Gruppe an.
- Jeder Benutzer bekommt einen eindeutigen Namen und damit er sich auch authentifizieren kann ein Passwort.
- Für die Benutzerverwaltung finden wir in der Gruppe *Sicherheit & Benutzer* die Module *Benutzer bearbeiten und anlegen*, *Neuen Benutzer anlegen*, *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen*.
- Diese Module sind Frontends zu der eigentlichen Benutzerverwaltung von Linux.
- Ein Frontend ist ein Programm, das eigentlich nichts selber macht, sondern sich als Bedienungselement zwischen den Benutzer und dem eigentlichem Programm setzt.
- Das Ziel eines Frontends ist es, die Bedienung von Programmen und Konfigurationsdateien zu erleichtern und damit komfortabler und fehlerfreier zu machen.
- Die Konfigurationsdateien für die Benutzerverwaltung befinden sich wie die meisten Konfigurationsdateien im Verzeichnis `/etc`.
- Die Liste der Benutzer ist in der Datei `/etc/passwd` abgelegt.
- Die Passwörter befinden sich in der Datei `/etc/shadow`.

#### 1.10.3.1 Wie erstelle, bearbeite und lösche ich neue Nutzer?

- Die Module *Benutzer bearbeiten und anlegen* und *Neuen Benutzer anlegen* sind für die Benutzerverwaltung verantwortlich.

##### 1.10.3.1.1 Neuen Nutzer anlegen

- Um einen neuen Benutzer anzulegen, starten Sie das Modul *Neuen Benutzer anlegen*.
- Im nun erscheinenden Fenster können Sie den Benutzernamen und das Passwort des neuen Benutzers eintragen.
- Bei der Eingabe eines Passworts müssen Sie zwischen Groß- und Kleinschreibung unterscheiden.
- Ein Passwort sollte mindestens fünf Zeichen lang sein und darf keine Sonderzeichen (z. B. Akzente) enthalten.
- Es ist ausreichend, ein Passwort mit der Länge von maximal acht alphanumerischen Zeichen zu wählen.
- Alle darüber hinausgehenden Zeichen werden bei der normalerweise eingestellten Crypt-Verschlüsselungen ignoriert.
- Das Passwort müssen Sie zweimal eintragen, da es aus Sicherheitsgründen nicht angezeigt wird.
- So können die meisten Tippfehler verhindert werden.
- Für die Benutzernamen und Passwörter gibt es Regeln.
- So darf der Benutzername nur aus Kleinbuchstaben, Ziffern, „-“ und „\_“ bestehen und muß mit einem Buchstaben oder „\_“ beginnen.
- Dies ist aber nur eine Vorgabe von YaST.
- Linux selber erlaubt auch Großbuchstaben und andere Zeichen für den Benutzernamen.
- Die Felder Vorname und Nachname sind eigentlich Spaß.
- Die entsprechenden Daten werden nur in das Kommentarfeld des Benutzers in der Datei `/etc/passwd` eingetragen.
- Hinter dem Button *Details* verbergen sich weitere Einstellungsmöglichkeiten wie Home-Verzeichnis oder die Benutzerkennung (*UID*).
- Jedem Benutzer bekommt eine eindeutige Benutzerkennung (*UID*) zugewiesen.
- Für normale Benutzer sollte eine UID größer als 499 verwendet werden, da die kürzeren UIDs vom System für spezielle Zwecke und Pseudo-Benutzernamen verwendet werden.

- Benutzer können durch eine Mitgliedschaft in einer Gruppe weitere Rechte bekommen.
- Zu jedem Benutzer gehört eine Standardgruppe in der er Mitglied ist.
- In dem Feld *Standardgruppe* können Sie eine Gruppe aus der Liste aller in Ihrem System vorhandenen Gruppen wählen.
- Zusätzliche Gruppenzugehörigkeiten können sie über das entsprechende Feld dem Benutzer zuweise.
- Hier wird angezeigt, zu welchen Gruppen der Benutzer noch gehört.
- Der Benutzer kann auch ein Heimatverzeichnis (*Home Directory*) besitzen um seine Daten zu dort zu speichern. Normalerweise ist dies `/home/BENUTZERNAME`.
- Für den Betrieb auf der Konsole (Textbetrieb) wird eine Login-Shell (der sog. Kommandozeileninterpreter) benötigt.
- Wählen Sie im entsprechenden Feld eine Shell aus der Liste der auf Ihrem System installierten Shells aus.

#### 1.10.3.1.2 Nutzer bearbeiten

- Eine Übersicht der angelegten Benutzer liefert das Modul *Benutzer bearbeiten und anlegen*.
- Beim Start werden nur die normalen Benutzer angezeigt.
- Sie können sich aber auch die für interne Zwecke (z. B. Serverdienste) angelegten Systembenutzer anzeigen lassen.
- Durch anklicken der Spaltenüberschriften können Sie die Sortierung der Spalten ändern.
- In diesem Fenster stehen Ihnen drei Funktionen zur Verfügung:
  - Einen neuen Benutzer anlegen (*Hinzufügen*),
  - den markierten Benutzer bearbeiten (*Bearbeiten*) und
  - den markierten Benutzer löschen (*Löschen*).
- Wenn Sie den Button *Hinzufügen* betätigen landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*.
- Auch nachdem Sie den Button *Bearbeiten* landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*.
- Nur daß die Felder nicht leer sondern mit den Daten des zu bearbeitenden Benutzers gefüllt sind.
- Wenn Sie den markierten Benutzer löschen wollen, dann betätigen Sie einfach den Button *Löschen*.
- Sie werden dann noch gefragt, ob das Heimatverzeichnis des Benutzers auch mit gelöscht werden soll.
- Wenn Sie sich dann fürs Löschen entscheiden ist es vorbei mit dem Benutzer.
- Vorbei? Nicht ganz.
- Die Änderungen werden erst dann ins System übertragen, wenn Sie mit dem Button *Beenden* das Modul verlassen.
- Das Verlassen über den Button *Abbrechen* läßt das System unberührt.

#### 1.10.3.2 Wie erstelle, bearbeite und lösche ich Gruppen?

- Um bestimmten Mengen von Benutzern mehr Rechte zu geben, wurden die Gruppen entwickelt. Benutzer können Mitglied in einer Gruppe sein und haben dann die Rechte, die der Gruppe zugewiesen wurde, zusätzlich zu Ihren eigenen Rechten.
- Die Module *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen* sind für die Verwaltung der Gruppen zuständig.

##### 1.10.3.2.1 Gruppen anlegen

- Nach dem Start des Moduls *Neue Gruppen anlegen* erscheint ein Fenster mit Einträgen für den Gruppennamen, die Gruppenkennung, dem Passwort und den Mitgliedern für die Gruppe.
- Der Name der Gruppe sollte nicht zu lang sein.
- Normalerweise werden 1-10 Zeichen verwendet.
- Ansonsten gelten die gleichen Regeln wie für die Benutzernamen.
- Wie auch die Benutzer haben die Gruppe eine interne Kennung.

- Diese GID liegt irgendwo zwischen 0 und 60000. Einige der IDs werden bereits während der Installation vergeben.
- YaST2 gibt eine Warnmeldung aus, wenn Sie aus Versehen eine bereits vergebene verwenden.
- Auch Gruppen haben ein Passwort.
- Falls die zugehörigen Benutzer der Gruppe sich beim Wechseln in diese neue Gruppe identifizieren sollen, können Sie dieser Gruppe ein Passwort zuteilen.
- Aus Sicherheitsgründen wird dieses Passwort hier nicht angezeigt.
- Dieses Eingabefeld muss nicht ausgefüllt werden. Um Tipfehler zu vermeiden, müssen Sie das Passwort zweimal eingeben.
- Das Fenster *Mitglieder dieser Gruppe* zeigt eine Liste aller angelegten Benutzer an.
- Hier können Sie einstellen, wer Mitglieder dieser Gruppe sein soll.

#### 1.10.3.2.2 Gruppen bearbeiten

- Das Modul *Gruppen bearbeiten und anlegen* zeigt eine große Ähnlichkeit mit dem Modul *Benutzer bearbeiten und anlegen*.
- Kein Wunder, es ist dasselbe Programm.
- Ein Klick auf *Benutzerverwaltung* und schon sind Sie in dem bereits bekannten Fenster. Dementsprechend gibt es hier auch drei Funktionen.
  - Eine neue Gruppe anlegen (*Hinzufügen*),
  - die markierte Gruppe bearbeiten (*Bearbeiten*) und
  - die markierte Gruppe löschen (*Löschen*).
- Beim Anlegen und Bearbeiten einer Gruppe landen Sie im gleichen Fenster wie im Modul *Neue Gruppen anlegen*.
- Das Löschen von Gruppen gestaltet sich etwas schwieriger, da erst alle Benutzer aus der Gruppe entfernt werden müssen, bevor sich die Gruppe löschen lässt.
- Auch hier gilt, alle Änderungen werden erst nach dem Verlassen des Moduls über den Button *Beenden* in das System übertragen.