1.11 Die Shell III

1.11.1 Starten von Programmen in den Hintergrund

- Die Shell ist Kommandozeileninterpreter und Programmiersprache zugleich.
- Die Kommandos können einzeln (synchron) oder mehrere parallel nebeneinander (asynchron) ausgeführt werden.
- Wenn ein synchrones Kommando ausgeführt wird, wartet die Shell bis der Befehl abgearbeitet worden ist, bevor sie weitere Eingaben akzeptiert.
- Dies bezeichnet man auch als einen Prozess im *Vordergrund* laufen zu lassen.
- Asynchrone Kommandos laufen ab, während die Shell weitere Kommandos ausführt.
- Dieses Kommando läuft dann im *Hintergrund*.
- Also immer wenn Sie einen Befehl eingeben, läuft dieser Befehl im Vordergrund.
- Die Shell wartet so lange, bis der Befehl abgearbeitet worden ist.
- Das kann z. B. beim `find`-Befehl etwas länger dauern.
- Um das Terminal dabei nicht zu blockieren, kann der Prozeß in den Hintergrund gelegt werden.
- Dabei sollte natürlich die Ausgabe des Befehls abgefangen und in eine Datei umgeleitet werden.
- Ein an die Zeile angefügtes kaufmännisches Und (`&`) sorgt für das Verschieben in den Hintergrund.
- `find / -user tapico -exec rm -f {}; 2> /dev/null &`
- Ein laufendes Programm können Sie unter der `bash` mit der Tastenkombination `<Strg>+<z>` stoppen und in den Hintergrund verschieben.
- Ein Job ist eine Folge von einem oder mehreren Kommandos.
- Also immer wenn Sie Linux einen einzelnen oder mehrere miteinander verknüpfte Befehle geben, erstellen Sie einen Job.
- Die Shell ist in der Lage diese Jobs zu kontrollieren und weißt diesen Jobs daher Identifikationsnummern zu.

1.11.2 Das Kommando jobs

- Um sich die Jobs anzeigen zu lassen, die sich im Moment im Hintergrund befinden, können Sie den Befehl `jobs` verwenden.
- `jobs [OPTIONEN] [JOB]`

Optionen

`-r`	Zeigt laufende Jobs an
`-s`	Zeigt gestoppte Jobs an

1.11.3 Das Kommando fg

- Um einen im Hintergrund laufenden Prozeß wieder in den Vordergrund zu holen, kann der Befehl `fg` verwendet werden.
- `fg [JOB]`
- Außerdem geht auch `%` oder `fg %`. Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden

1.11.4 Das Kommando bg

- Wenn ein Prozeß mit `<STRG>+<Z>` gestoppt und in den Hintergrund geschoben worden ist, kann er mit dem Befehl `bg` im Hintergrund wieder gestartet werden.
- `bg [JOB]`
- Außerdem geht auch `bg %N` oder `fg %NAME`.
- Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden.

1.12 Der Editor vi

- ist ein weit verbreiteter Unix-Editor und somit auf den meisten Unix Systemen verfügbar.
- Mittlerweile gibt es eine extrem verbesserte Version namens vim, die jedoch nur auf wenigen neueren Systemen zu finden ist.
- Syntax: `vi [datei]`
- im `vi` gibt es hauptsächlich zwei Modi, den Befehlsmodus und den Einfügemodus.
- Im Befehlsmodus kann man sich im Text bewegen, löschen, und andere Formatierungen vornehmen.
- Im Einfügemodus kann man Texte eingeben.
- Die genaue Bedienung des `vi` ist dem Anhang „vi Kurzreferenz“ zu entnehmen!

1.13 Grundlagen der Systemverwaltung

1.13.1 Dateinamen

- MS-DOS beschränkt die Größe des Namens einer Datei auf acht Zeichen plus einem mit einem Punkt abgetrennten Dateikürzel mit drei Zeichen.
- das erste UNIX-Dateisystem erlaubte 14 Zeichen.
- das LINUX-Dateisystem ext2 (*extended filesystem, version 2*) erlaubt bis zu 255 Zeichen.
- der Punkt ist nur eines von vielen Zeichen und kann beliebig oft angewendet werden.
- im Gegensatz zu MS-DOS unterscheidet LINUX bei Dateinamen zwischen Groß- und Kleinschreibung
- `TEST`, `Test` und `test` sind drei verschiedene Dateinamen.
- Aus praktischen Gründen werden die Dateinamen meistens klein geschrieben.
- Bei Pfadangaben werden die einzelnen Verzeichnisse durch einen Schrägstrich `/` (*slash*) getrennt.
- Daher ist dieses Zeichen für Dateinamen verboten.
- Auch das Minuszeichen kann am Anfang einer Datei Probleme bereiten.
- So stellt sich bei `ls -fault` die Frage, ob nun die Datei `-fault` oder die Schalter `f`, `a`, `u`, `l` und `t` gemeint sind.
- Probleme bereiten auch die nichtdruckbaren (ASCII-)Zeichen 0 bis 32 sowie `< > | * ? [ ] { } ( ) $ ` ' " ! ; \ # ^`
- Diese Zeichen sind nicht verboten, können aber große Probleme bereiten und sollten deshalb auch vermieden werden!

1.13.2 Dateitypen

Dateityp	Beschreibung	Erzeugung	Zeichen im `ls -l` / `ls -F`
Verzeichnis vgl. Ordner oder Schubfach	enthält weitere Dateien oder Verzeichnisse	`mkdir`	`d` / `/`
normale Dateien Dateien im Dateisystem	die beliebige Daten beinhalten	`touch`	`-` / [nichts]
ausführbare Dateien	sind normale Dateien, deren Rechte auf ausführbar gesetzt sind - z.B. Programme, Scripte ...	`chmod`	`-` / `*`
symbolische Links (symlinks)	sind Verknüpfungen auf andere Dateien	`ln -s`	`l` / `@`
feste Links (hardlinks)	echte Duplizierung ein und derselban Datei an zwei verschiedene Stellen/Namen	`ln`	wie die verlinkte Datei!
fifo (named pipes)	First In First Out - virtuelle Dateien um Programmausgaben/-eingaben zu verknüpfen	`mkfifo`	`p` / `|`
character Devices	im /dev - Symbolisch für Systemkomponenten (z.B. /dev/console)	`mknod c`	`c` / [nichts]
block Devices	im /dev - Symbolisch für Systemkomponenten (z.B. /dev/hda)	`mknod b`	`b` / [nichts]

1.13.3 Links

- Im Prinzip ist ein Link nur ein weiterer Name für die Datei.
- Bei den Links wird zwischen dem harten Link und dem symbolischen (weichen) Link unterschieden.
- Ein Link kann dazu benutzt werden einem anderen Benutzer Zugriff auf eine Datei zu gewähren.
- Als Erstes bekommt der andere Benutzer Rechte auf die Datei und evt. auf das Verzeichnis.
- Danach kann er einen Link auf die Datei in seinem Heimatverzeichnis anlegen.
- Dadurch erhält er einen vereinfachten Zugriff auf den Inhalt der Datei.
- Durch die Benutzung von Links brauchen oft keine Kopien von Dateien erstellt werden.
- Dies spart Speicherplatz und erleichtert die Aufgabe des Systemadministrators alle Dateien auf dem neuesten Stand zu halten.
- Daneben können große Verzeichnisbäume übersichtlicher gestaltet werden.

1.13.3.1 Harte Links

- Immer wenn Sie eine Datei erzeugen wird auch ein Link zu dieser Datei erzeugt.
- Er ermöglicht den Zugriff auf die Datei, weil er direkt auf die Inode der Datei zeigt.
- Der Befehl `rm` löscht übrigens nicht direkt die Datei, sondern den Link, der auf die Inode der Datei zeigt.
- Einen Link können Sie nur auf eine existierende Datei anlegen.
- Er ist ein weiterer Name für die gleiche Datei.
- Dies wird deutlich wenn Sie sich die Links mit `ls -i` anzeigen lassen.
- Beide Links zeigen auf die gleiche Inode.
- Sie unterscheiden sich nur im Namen und/oder ihrem Ort im Verzeichnisbaum.

Beispiel:

- Auf die existierende Datei `eins` wird ein harter Link namens `zwei` angelegt.
- Wird nun `eins` editiert, so finden sich die Änderungen auch in `zwei` wieder.

```
[511] htor@archimedes:~/links$ ls -la
insgesamt 12
drwxr-xr-x    2 htor     users        4096 2003-03-15 15:54 ./
drwxr-xr-x   99 htor     users        8192 2003-03-15 15:54 ../
-rw-r--r--    2 htor     users           0 2003-03-15 15:54 eins
-rw-r--r--    2 htor     users           0 2003-03-15 15:54 zwei
[512] htor@archimedes:~/links$
```

- In der dritten Spalte sehen Sie die Anzahl der harten Links, die auf die Datei zeigen.
- Löschen Sie nun einen der beiden Links, so bleibt die Datei trotzdem bestehen.
- Erst wenn alle Links auf die Datei gelöscht sind, die Zahl also auf Null steht, wird die Datei auf wirklich gelöscht.
- Es gibt aber auch Einschränkungen für harte Links.
  - alle Links müssen sich auf dem gleichen Dateisystem befinden.
  - Außerdem ist es einem normalen Benutzer nicht möglich einen Hardlink auf ein Verzeichnis einzurichten. Dies ist nur `root` vorbehalten.

1.13.3.2 Symbolische Links

- Im Gegensatz zu dem harten Link zeigt der symbolische Link nur indirekt auf eine Datei, da er auf einen Namen (harten Link) der Datei zeigt.
- Deshalb ist er auch ein neuer Eintrag mit eigener Inode im Verzeichnisbaum.
- Symbolische Links werden dann eingerichtet, wenn die Grenzen der harten Links umgangen werden sollen.
- Symbolische Links können eingerichtet werden für
  - Verzeichnisse durch normale Benutzer
  - Dateien, die nicht existieren
  - Dateien auf einem anderen Dateisystem
- Wie im Beispiel unten zu sehen, ändert sich das Aussehen des symbolischen Links bei Verwendung des Befehls `ls -l`.
- Der Linkname wird durch einen Pfeil und den Namen der Zieldatei ergänzt.

```
[513] htor@archimedes:~/links$ l
insgesamt 12
drwxr-xr-x    2 htor     users        4096 2003-03-15 15:56 ./
drwxr-xr-x   99 htor     users        8192 2003-03-15 15:54 ../
lrwxrwxrwx    1 htor     users           1 2003-03-15 15:58 dir -> ./
-rw-r--r--    2 htor     users           0 2003-03-15 15:54 eins
lrwxrwxrwx    1 htor     users           4 2003-03-15 15:56 link -> eins
-rw-r--r--    2 htor     users           0 2003-03-15 15:54 zwei
[514] htor@archimedes:~/links$
```

- Die Funktion der Befehle `cd` und `pwd` kann sich bei gelinkten Verzeichnissen je nach Shell unterscheiden.
- In der `bash` zeigt, wenn wir dem Link `kurs` gefolgt sind, z. B. der Befehl `pwd` das Arbeitsverzeichnis über die verlinkte Struktur an.
- Der Befehl `cd ..` führt auch in das Verzeichnis mit dem Link zurück.
- In anderen Shells führt der Link an den Ort der normalen Struktur.
- Das aktuelle Verzeichnis ist dann nicht mehr durch den symbolischen Link gegeben, sondern durch den harten Link des Verzeichnisses.

1.13.4 Kommandos zur Dateiverwaltung

1.13.4.1 Das Kommando du

- Das Kommando `du` (*Disk Usage*) zeigt den Platz an, den eine Datei oder ein Verzeichnis mit seinen Dateien und Unterverzeichnissen einnimmt.
- `du [OPTIONEN] [DATEINAME]`
- Wird `du` ohne Dateinamen aufgerufen, dann zeigt es die Daten des aktuellen Arbeitsverzeichnisses an.

Optionen

`-a`	Zeige die Nutzung für einzelne Dateien
`-b`	Anzeige des Speicherplatzes in Bytes
`-c`	Zeigt die Summe an
`-h`	Zeigt durch einen Buchstaben die verwendete Einheit an
`-k`	Anzeige des Speicherplatzes in Kilobytes (normal)
`-m`	Anzeige des Speicherplatzes in Megabytes

`-l`	Zählt auch die Links
`-s`	Gibt nur die totale Summe aus

Ohne den Schalter `-a` zeigt `du` nur die Verzeichnisse.

Beispiel:

```
[580] htor@archimedes:~$ du -s tmp
155236  tmp
[581] htor@archimedes:~$
```

1.13.4.2 Das Kommando find

- Dieses Tool wird dazu benutzt das angegebene Verzeichnis und seine Unterverzeichnisse nach Dateien zu durchsuchen, die den mitgegebenen Kriterien entsprechen.
- `find [PFAD] [SUCHKRITERIEN]`
- Werden keine Kriterien angegeben, so zeigt `find` alle Dateien im Verzeichnis und seinen Unterverzeichnissen an.
- Wird kein Pfad angegeben, so dient das aktuelle Arbeitsverzeichnis als Startverzeichnis.
- Um die Anzahl der gefundenen Dateien einzuschränken, können Kriterien für die Eigenschaften der gesuchten Dateien angegeben werden.
- folgende liefert einen Überblick über die wichtigsten Kriterien.

Kriterien	Aktion
`-atime N`	Findet Dateien auf der Basis des letzten Zugriffs
`-ctime N`	Findet Dateien auf der Basis der letzten Änderung des Verzeichniseintrags
`-exec BEFEHL`	Ausführen eines `BEFEHL` für jede gefundene Datei
`-group GRUPPE`	Findet Dateien die zu einer bestimmten `GRUPPE` (GID) gehören
`-inum N`	Findet Dateien mit der Inode `N`
`-links N`	Findet Dateien mit `N` Links
`-mount`	Findet Dateien nur im gleichen Dateisystem
`-mtime` **`Fehler! Textmarke nicht definiert.`** `N`	Findet Dateien auf der Basis der letzten Änderung
`-name MUSTER`	Findet Dateien, auf deren Dateiname das `MUSTER` zutrifft.
`-newer DATEI`	Findet Dateien, die jünger als die angegebene `DATEI` sind.
`-ok BEFEHL`	Ausführen eines `BEFEHL` für jede gefundene Datei mit vorheriger Sicherheitsabfrage
`-perm NNN`	Findet Dateien, auf die Rechtemaske `NNN` zutrifft.
`-print`	Zeigt die gefundenen Dateien mit ihrem Pfad an. (Standard)
`-size N[c]`	Findet Dateien in Abhängigkeit von ihrer Größe (`N` Blocks oder `N` Zeichen).
`-type C`	Findet Dateien vom gegebenen Typ: b = Blockgerät, c = Zeichengerät, d = Verzeichnis, l = Link oder f = normale Datei
`-user BENUTZER`	Findet die Dateien des angegebenen Benutzers

- Einzelne Suchkriterien sind logisch verknüpfbar, d. h., es können Suchkriterien aus verschiedenen Teilbedingungen zusammengesetzt werden.
- Numerische Angaben als Bestandteil eines Suchkriteriums lassen sich auf drei Weisen benutzen:
  - `N`, `+N` und `-N`, wobei `N` eine ganze Zahl ist.
  - Das Kriterium `-links` veranlaßt z. B. eine Suche nach Dateien mit einer bestimmten Anzahl von Links und erfordert eine entsprechende numerische Angabe. So bedeuten

`-links 3`	Suche nach Dateien mit 3 Links
`-links +3`	Suche nach Dateien mit mehr als 3 Links
`-links -3`	Suche nach Dateien mit weniger als 3 Links

Beispiele:

`find -name "test*"`
Findet alle Dateien im aktuellen Verzeichnis und darunter, die mit `test` beginnen.
Dabei sollte das Suchmuster in Anführungszeichen stehen.

`find /usr -type d -print`
sucht nach allen Verzeichnissen, die unterhalb von `/usr` existieren und zeigt ihre Pfadnamen an.

`find / -name tty* -print`
liefert die Pfade aller Dateien des gesamten Dateisystems, die mit `tty` beginnen.
`find / -user 406 -ok rm {} \;`
sucht im gesamten Dateisystem nach Dateien des Benutzers mit der uid `406`. Gefundene Dateien werden bei positiver Beantwortung einer Sicherheitsabfrage gelöscht.

`find . -newer .version_time -print`
zeigt alle Dateien des aktuellen Verzeichnisses, die jüngeren Datums sind als die Datei `.version_time`.

`find / -size +10k -user htor -print`
zeigt alle Dateien die größer als 10 kB sind und dem Benutzer `htor` gehören.

`find ~ -atime -10 -print`
findet alle Dateien, auf die innerhalb der letzten 10 Tage zugegriffen wurde.

Bei Angabe von mehreren Kriterien wird immer von einer UND-Verknüpfung ausgegangen.
`find / -size +10k -user htor -print`
zeigt alle Dateien die größer als 10 kB sind und dem Benutzer `htor` gehören.
Dies kann durch `-a` betont werden.
`find / -size +10k -a -user tapico -print`

Eine ODER-Verknüpfung erreicht man durch `-o`.
`find ~ -name '*.htm' -o -name '*.html' -print`
sucht nach allen Dateien, die auf `.htm` oder `.html` enden.

Die Negation eines Kriteriums wird durch das `!` erreicht.
`find / -name !'*~' -print`
findet alle Dateien, die nicht mit einem Tilde `~` enden.

1.13.4.3 Das Kommando mkfifo

- Erzeugen kann man FIFOs mit dem Befehl `mkfifo` (*make fifo*).
- `mkfifo [OPTIONEN] NAME`
- Die Befehlszeile
- `mkfifo testpipe`
- würde z. B. eine Named Pipe mit Namen `testpipe` erzeugen.
- Genutzt werden könnte sie dann wie folgt:
- `ls /dev > testpipe & less < testpipe`
- `ls /dev` zeigt die Namen aller Gerätedateien (genauer: aller Dateien des Verzeichnisses `/dev`).
- In diesem Fall werden die Daten in die Named Pipe umgeleitet statt auf dem Bildschirm ausgegeben.
- Der Befehl `less` liest die Daten aus `testpipe` und gibt sie bildschirmseitenweise aus.
- Das `&` zwischen den Befehlen sorgt dafür, daß beide Teilbefehle nacheinander als Prozesse gestartet werden, allerdings ohne daß auf ihr Ende gewartet werden müßte.
- Insgesamt werden in diesem Beispiel also die Namen aller Gerätedateien bildschirmseitenweise angezeigt.

- Dieses Beispiel stellte die Named Pipes nur prinzipiell vor.
- Als Benutzer wird man in der gegebenen Situation eine gewöhnliche Pipe vorziehen: `ls /dev | less`

1.13.4.4 Das Kommando ln

- Das Kommando `ln` legt einen harten oder einen symbolischen Link an.
- `ln [OPTIONEN] DATEINAME NEUERLINK`
- Ist `NEUERLINK` ein Verzeichnis und die Optionen `-d` und `-F` sind nicht gegeben, so wird in dem Verzeichnis ein Link mit dem Namen `DATEINAME` angelegt.

Optionen

`-d`	Legt einen Link auf ein Verzeichnis an (Nur Superuser)
`-F`	Legt einen Link auf ein Verzeichnis an (Nur Superuser)
`-s`	Legt einen symbolischen Link an

Einen harten oder symbolischen Link können Sie durch den Befehl `rm` entfernen

1.13.5 Dateisicherung und Archivierung

- Es gibt eine große Anzahl von Programmen für die Datensicherung.
- Einige davon sind große mächtige kommerzielle Lösungen, andere sind klein und gehören zur Linux-Grundausstattung.
- Zwei davon sind `tar` und `cpio`.

1.13.5.1 tar

- Das Programm `tar` (*tape archive*) wird dazu benutzt um mehrere Dateien zu einer Archivdatei zusammenzupacken.
- Dabei wird die Verzeichnisstruktur beibehalten.
- Obwohl `tar` entwickelt wurde um Daten auf Magnetbänder zu schreiben, kann ein solches `tar`-Archiv auf jedem beliebigem Medium gespeichert werden.
- Daneben kann `tar` die Archive bei der Erstellung auch gleichzeitig mit `gzip` komprimieren.
- `tar [OPTIONEN] [TARARCHIV] [DATEILISTE]`
- In den Optionen bilden die Schalter eine besondere Gruppe.
- Es kann immer nur ein Schalter zur Zeit verwendet werden, während die anderen Optionen kombinierbar sind.

Schalter

`A`	Hängt das ARCHIV2 and das Ende von ARCHIV1
`c`	Legt ein neues Archiv an
`d`	Vergleicht den Inhalt des Archivs mit anderen Dateien
`r`	Fügt die neuen Dateien an das Ende eines bestehenden Archivs an
`t`	Zeigt eine Liste aller Dateien im Archiv
`u`	Fügt nur neue oder veränderte Dateien zum Archiv hinzu
`x`	Extrahiert Dateien aus dem Archiv

Optionen

`b`	Definiert die Blockgröße
`e`	Verhindert das Aufsplittern von Dateien über Archiv-Volumes
`f`	Name des Archivs mit Pfad oder Gerätename
`m`	Übernimmt nicht die Änderungszeit der Datei aus dem Archiv
`n`	Das Gerät ist kein Bandgerät
`p`	Die Originalrechte werden übernommen
`v`	Zeigt die Liste der Dateien an, die hinzugefügt werden
`w`	`tar` arbeitet interaktiv
`z`	Benutzt `gzip` zur Kompression
`F`	Am Ende jedes Mediums das angegeben Skript ausführen

L	Länges des Bandes in kByte
M	Das Archiv ist in mehrere Teile (Volumes) aufgeteilt.
W	Überprüft die Dateien, nachdem sie zum Archiv hinzugefügt worden sind.

Beispiele:
Erstellung eines Archivs (`archiv.tar`) über alle Dateien im aktuellen Arbeitsverzeichnis und dessen Unterverzeichnissen.
```
tar cf archiv.tar .
```

Als Ziel kann neben einer Datei auch ein Gerät angegeben werden. Der folgenden Befehl speichert das komplette Dateisystem auf dem Gerät `/dev/tape`.
```
tar cf /dev/tape /
```

Dabei wird der Inhalt des Magnetbandes überschrieben. Meistens müssen Sie bei einem Magnetband die Blockgröße mit angeben. Die Blockgröße definiert dabei die Menge an Daten (in Einheiten zu 512 Byte), die zur gleichen Zeit geschrieben werden können. Die Angabe erfolgt durch die Option `b`.
```
tar cfb /dev/tape 20 /
```

Wie Sie sehen, wird der Wert für `b` nicht wie üblich direkt hinter die Option geschrieben, sondern erst werden die Optionen aufgeführt und dann in der Reihenfolge die Werte.

Wenn Sie das Archiv auf mehrere Geräte (z. B. Disketten) aufteilen wollen, müssen Sie die Größe der Archiv-Teile (Volumes) angeben. Dies erfolgt durch den Schalter `M`. So speichert der folgende Befehl den Inhalt des Verzeichnis `/home` auf das Diskettengerät `/dev/fd0` mit einer Größe von 1440 kB.
```
tar cfML /dev/fd0 1440 /home
```

Wenn Sie ein bestehendes Archiv nur auf den neuesten Stand bringen wollen, ohne alle Dateien erneut hineinzupacken, dann benutzen Sie den Schalter `u`.
```
tar uf archiv.tar .
```

Sie können Dateien auch an ein bestehendes Archiv anhängen.
```
tar rf archiv.tar neueDatei
```

Solche Archive können sehr groß werden, wenn Sie z. B. Logdateien speichern. Diese können sehr gut gepackt werden. Mit der Option `z` wird `tar` angewiesen, das entstandene Archiv auch gleich mit `gzip` zu packen.

Um sich die Dateiliste des Archivs anzeigen zu lassen, kann man den folgenden Befehl verwenden.
```
tar tf archiv.tar
```

Um eine Archiv zu entpacken (Schalter `x`) und dabei auch die Dateinamen zu sehen (Option `v`) können Sie den folgenden Befehl benutzen.
```
var xvf archiv.tar
```

Vor dem Entpacken, sollten Sie feststellen welche Verzeichnispfad gespeichert wurden. Dies können Sie mit dem Befehl
```
tar tvf archiv.tar
```
feststellen.

Die Option `v` erzeugt eine ausführlichere Anzeige als `t` alleine.

Schauen wir uns doch mal folgendes Beispiel an:
Aus dem Wurzelverzeichnis haben Sie die Daten im Verzeichnis `/home` mit folgendem Befehl gespeichert.
```
tar cf home.tar home/*
```

Daher wurde das Verzeichnis `home` in jedem Pfad mit eingefügt. Sind Sie allerdings ins Verzeichnis `/home` gewechselt und haben den Befehl als
```
tar cf home.tar .
```
ausgeführt, so ist das Verzeichnis `home` nicht mehr Bestandteil des Dateipfads.

Bei einem Backup sollten Sie die Dateien immer in dem gleichen Verzeichnis extrahieren in dem Sie auch das Archiv erzeugt hatten.

Um einzelne Dateien aus einem Archiv zu entpacken, können Sie zum einen den interaktiven Modus (Option `w`) von `tar` wählen. Bei dem folgenden Befehl werden Sie bei jeder Datei gefragt, ob Sie die Datei entpacken wollen.

```
tar xvwf archiv.tar
```

Allerdings hat diese Methode einen schwerwiegenden Nachteil. Auf diesem Wege eine Datei aus einem 10.000 Dateien-Archiv zu entpacken ist doch etwas mühselig. Hier bietet sich einfach an, die gewünschte Datei einfach anzugeben.

```
tar xf archiv.tar meineDatei.txt
```

1.13.5.2 cpio

- Ein weiteres Werkzeug für Arbeit mit Archiven ist `cpio` (*copy in and out*).
- Sie können damit nicht nur aus mehreren Dateien ein Archiv machen und die Dateien aus einem solchen Archiv wieder extrahieren, sondern auch komplette Verzeichnisstrukturen an einen anderen Ort kopieren.
- `cpio [OPTIONEN]`
- Wie auch bei `tar` gibt es auch hier Schalter und Optionen.

Schalter

`-o`	Legt ein neues Archiv an
`-i`	Extrahiert Dateien aus einem Archiv
`-p`	Kopiert komplette Verzeichnisstrukturen

Optionen

`-a`	Setzt die Zugriffszeit der Dateien nach dem Kopieren zurück.
`-d`	Erzeugt Verzeichnisse, wenn Sie benötigt werden (`-i` und `-p`)
`-E DATEI`	Name einer Datei mit zusätzlichen Mustern für die zu entpackenden Dateien
`-F ARCHIV`	Name des Archivs, was entpackt werden soll (`-i`)
`-m`	Erhält die Änderungszeit der Dateien (`-i`)
`-r`	Fragt nach einem neuen Namen für die Datei vorm Kopieren, wird kein Namen angegeben, wird die Datei nicht kopiert.
`-t`	Zeigt mit `-i` die Dateiliste des Archivs an
`-u`	Überschreibt existierende Dateien
`-v`	Zeigt die Dateinamen bei der Bearbeitung an

- Im Gegensatz zu `tar` arbeitet `cpio` nicht direkt mit den Dateien sondern übernimmt nur die Aufgabe der Archivierung.

Beispiele:
Um alle Dateien und Verzeichnisse im Verzeichns `/home` zu sichern, müssen die Dateinamen mit `find` übergeben werden und die Ausgabe in eine Datei umgeleitet werden.

```
find /home | cpio -o > home.cpio
```

Um nun zu prüfen welche Dateien in dem Verzeichnis sind, können Sie folgenden Befehl benutzen.

```
cpio -itF home.cpio
```

Um das Archiv zu entpacken, benutzen Sie den Befehl

```
cpio -iF home.cpio
```

Eine einzelne Datei läßt sich genau wie bei `tar` aus einem Archiv extrahieren.

```
cpio -iF home.cpio meineDatei.txt
```

1.13.5.3 Tarballs

- Was sind Tarballs? Diese Frage stellen sich viele Benutzer, die zum ersten Mal ein Programm aus dem Netz laden und nicht von der Distribution installieren wollen.
- Auf den meisten Seiten findet sich neben Bezeichnungen wie RPM- und GNU Debian-Paketen auch die Möglichkeit das Programm als „*tarball*“ herunterzuladen.
- Neben den schon fertig kompilierten Binärdateien erhalten Sie in der Open Source Gemeinde auch den Quellcode um ihr Programm selbst kompilieren und installieren zu können.
- Dabei wird der Quellcode nicht nur alleine vertrieben, sondern meist zusammen mit Konfigurationsdateien für die Installation und Dokumentationen zum Programm.
- Diese Dateien befinden sich meistens in einer Verzeichnisstruktur.
- Solche Strukturen werden am besten in einem `tar`-Archiv gesichert.
- Damit das entstandene Archiv auch schnell heruntergeladen werden kann, wird es mit `gzip` komprimiert.

1.13.5.4 Das Kommando gzip

- Das Tool `gzip` komprimiert den Inhalt einer Datei und erzeugt daraus eine neue Datei mit dem gleichen Namen und einem angehängten `.gz`.
- Im Gegensatz zu dem Programm `zip`, daß es auch unter verschiedenen Namen unter Windows gibt, kann `gzip` nur eine Datei packen und löscht standardmäßig auch die Originaldatei.
- `gzip [OPTIONEN] [DATEI]`

Optionen

`-c`	Zeigt den Inhalt an ohne die komprimierte Datei zu löschen; zusammen mit `-d` (*content*)
`-d`	Dekomprimiert die Datei (*decompress*)
`-n`	Speichert nicht die originalen Zeitstempel und Dateinamen(*no name*)
`-N`	Speichert die originalen Zeitstempel (Standard) (*!(no name)*)
`-q`	Unterdrückt Warnmeldungen (*quiet*)
`-r`	Verarbeitet auch die Dateien in Unterverzeichnissen (*recursive*)
`-t`	Test der Datenintegrität(*test*)
`-v`	Name und Kompressiongrad werden ausgegeben(*verbose*)
`-ZAHL`	Gibt den Kompressionsgrad mit ZAHL an; Wert zwischen 1 (niedrig) und 9 (hoch).

Beispiel:

```
[507] htor@archimedes:~/test$ l
insgesamt 16
drwxr-xr-x    2 htor     users        4096 2003-03-16 11:57 ./
drwxr-xr-x   99 htor     users        8192 2003-03-16 11:35 ../
-rw-r--r--    1 htor     users         920 2003-03-16 14:03 test
[508] htor@archimedes:~/test$ gzip test
[509] htor@archimedes:~/test$ l
insgesamt 16
drwxr-xr-x    2 htor     users        4096 2003-03-16 14:03 ./
drwxr-xr-x   99 htor     users        8192 2003-03-16 11:35 ../
-rw-r--r--    1 htor     users         434 2003-03-16 14:03 test.gz
[510] htor@archimedes:~/test$
```

1.13.5.5 Das Kommando bzip2

- `bzip2` ist eine Weiterentwicklung des `gzip`-Kommandos, es erlaubt eine bessere Komprimierung der Dateien
- die Handhabung ist weitestgehend identisch, Unterschiede sind in der `man`-Page ersichtlich

Beispiel:

```
[595] htor@archimedes:~/test$ l
insgesamt 16
```

```
drwxr-xr-x    2 htor     users        4096 2003-03-16 14:05 ./
drwxr-xr-x   99 htor     users        8192 2003-03-16 11:35 ../
-rw-r--r--    1 htor     users         920 2003-03-16 14:03 test
[595] htor@archimedes:~/test$ bzip2 test
[596] htor@archimedes:~/test$ l
insgesamt 16
drwxr-xr-x    2 htor     users        4096 2003-03-16 14:05 ./
drwxr-xr-x   99 htor     users        8192 2003-03-16 11:35 ../
-rw-r--r--    1 htor     users         362 2003-03-16 14:03 test.bz2
[597] htor@archimedes:~/test$
```

1.13.5.6 Das Kommando zip/unzip

- `zip` dient zur Erzeugung eines Zip-Archives, welches unter Windows bzw. DOS weiterverarbeitet werden kann
- unzip kann Windows `zip`-Archive entpacken
- wenn Sie nur `zip` oder `unzip` eingeben erhalten Sie eine Kurzhilfe

Beispiel:
Eine neue zip-Datei erstellen:
`zip meinzip datei1 datei2 datei3`
packt den Inhalt von `datei1`, `datei2`, `datei3` in die zip-Datei `meinzip.zip`

Ein zip-Archiv extrahieren:
`unzip meinzip.zip`
entpackt die Datei meinzip.zip ins aktuelle Verzeichnis.

1.13.6 Mounten

- Da jede Partition und jedes Speichergerät sein eigenes Dateisystem besitzt, müssen diese an einer zentralen Stelle zusammengeführt werden.
- Bei Windows wird dies durch die Laufwerksbuchstaben realisiert, die unter *Arbeitsplatz* liegen.
- Bei Linux muß eine Partition immer als Wurzel `/` ansprechbar sein.
- Die anderen Partitionen werden dann in das Verzeichnissystem dieser Partition eingebunden.
- Das Verzeichnis, das die Wurzel des jeweiligen Dateisystems im Verzeichnissystem repräsentiert, wird als **Mount Point** bezeichnet und der Vorgang des Einbindens als **Mounten**.
- Dabei ist das Einbinden nicht nur auf die Linux-Dateisysteme wie ext2 beschränkt.
- Es können auch virtuelle Dateisysteme wie `/proc` eingebunden werden oder sogar Dateisysteme, die sich auf anderen Rechnern im Netz befinden.

1.13.6.1 Das Kommando mount

- Der `mount`-Befehl bindet ein Dateisystem in den aktuellen Verzeichnisbaum ein.
- `mount [OPTIONEN] [GERÄT] MOUNTPOINT`

Optionen

`-a`	Mounte alle Dateisysteme, die in `/etc/fstab` aufgeführt werden.
`-f`	Überprüfe ob das angegebene Dateisystem gemountet werden kann
`-n`	Schreibe die Mount-Informationen nicht in die Datei `/etc/mtab`
`-o OPTION`	Modifikatoren für den Mountvorgang (siehe folgende Tabelle)
`-r`	Mounten nur mit Leseberechtigung (read only)
`-t DSTYP`	Typ für das zu mountende Dateisystems
`-v`	Angabe der Mount-Informationen
`-w`	Mounten mit Schreibberechtigung (Standard)

Tabelle: Modifikatoren für `mount -o` und `/etc/fstab`

Option	Aktion
`async`	Der I/O-Zugriff erfolgt asyncron

atime	Der letzte Zugriff wird in der Inode festgehalten
auto	Kann mit der Option -a gemountet werden
defaults	Entspricht: rw, suid, dev, exec, auto, nouser und async.
dev	Interpretiert spezielle Zeichen- und Block-Geräte
exec	Binärdateien können ausgeführt werden
noatime	Der letzte Zugriff wird nicht in der Inode festgehalten (schnellerer Zugriff)
noauto	Wird nicht automatisch gemountet
nodev	Interpretiert nicht spezielle Zeichen- und Block-Geräte
noexec	Binärdateien werden nicht ausgeführt
nosuid	Das suid- oder sgid-bit wird nicht ausgeführt
nouser	Ein Benutzer kann das Dateisystem nicht mounten
remount	Mountet ein Dateisystem erneut (Wird verwendet um Modifikatoren zu ändern)
ro	Das Dateisystem wird nur lesbar gemountet (read only)
rw	Das Dateisystem wird schreibbar gemountet (read write)
suid	Erlaubt die Benutzung des suid- oder sgid-bits
sync	Alle E/A-Aktionen sollten syncron erfolgen
user	Ein Benutzer darf das Dateisystem mounten

- Der Aufruf von `mount` ohne Parameter zeigt die Liste aller Dateisysteme an, die in diesem Moment gemountet sind.

```
[531] htor@archimedes:~$ mount
/dev/hda4 on / type ext3 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /boot type ext3 (rw)
/dev/hdb5 on /windrives/e type vfat (rw,nosuid,nodev,umask=000,quiet)
/dev/hda5 on /windrives/d type vfat (rw,nosuid,nodev,umask=000,quiet)
shmfs on /dev/shm type shm (rw)
perikles:/mnt/hdb on /mnt/perikles/hdb type nfs
rw,nosuid,nodev,addr=192.168.100.254,user=htor)
/dev/hdc on /cdrom type iso9660 (ro,nosuid,nodev,user=htor)
/dev/hdd on /ddrom type iso9660 (ro,nosuid,nodev,user=htor)
[532] htor@archimedes:~$
```

- Beim Einbinden eines Dateisystems mit `mount` wird die Datei `/etc/fstab` ausgewertet. Wenn das Gerät in der Datei angegeben ist, so reicht für das Mounten die Angabe des Mount Points aus.
- `mount /floppy`
- Außerdem werden die Angaben für dieses Gerät aus der Datei entnommen. Änderungen der gegebenen Modifikatoren kann über den Schalter `-o` erfolgen.

Beispiel:

Ist der Mount Point in der `/etc/fstab` eingetragen, so reicht ein
`mount /a`
aus.
Ist dies nicht der Fall, so kann das Gerät über
`mount -t vfat /dev/fd0 /a`
gemountet werden. Dies ist aber nur `root` erlaubt.

1.13.6.2 Das Kommando umount

- Der `umount`-Befehl entfernt ein Dateisystem aus dem aktuellen Verzeichnisbaum.
- `umount [OPTIONEN] [GERÄT] [MOUNTPOINT]`

Optionen

`-a`	Unmounte alle Dateisysteme, die in `/etc/mtab` aufgeführt werden.
`-n`	Schreibe die Mount-Informationen nicht in die Datei `/etc/mtab`
`-t DSTYP`	Typ für das zu unmountende Dateisystems

- Durch den Befehl `umount` wird auch der Befehl `sync` aufgerufen, damit vor dem Entfernen des Dateisystems dieses mit dem Cache abgeglichen wird.

1.13.6.3 Der Befehl sync

- Durch den Befehl `sync` werden die sich im Cache befindlichen Daten auf die Platte geschrieben und damit die Festplatte auf den aktuellen Stand gebracht.
- `sync`
- Der `sync`-Befehl ruft einfach die Kernel-Prozedur `sync` auf.
- Diese wird auch vom Befehl `umount` zum sauberen Aushängen von Dateisystemen benutzt

1.13.6.4 Die Datei /etc/fstab

* Die Datei `/etc/fstab` wird von den Befehlen `fsck`, `mount` und `umount` benutzt.

```
[532] htor@archimedes:~$ cat /etc/fstab
/dev/hda4     /                    ext3       defaults                1 1
/dev/hda1     /boot                ext3       defaults                1 2
devpts        /dev/pts             devpts     mode=0620,gid=5         0 0
proc          /proc                proc       defaults                0 0
usbdevfs      /proc/bus/usb        usbdevfs   noauto                  0 0
# WINDOWS DRIVES
/dev/hda2  /windrives/c   ntfs   rw,auto,user,umask=000,exec,quiet   0   0
/dev/hdb5  /windrives/e   vfat   rw,auto,user,umask=000,exec,quiet   0   0
/dev/hda5  /windrives/d   vfat   rw,auto,user,umask=000,exec,quiet   0   0

# FLOPPY
/dev/fd0      /floppy                 auto            noauto,user 0   0
/dev/fd1      /gloppy                 auto            noauto,user 0   0

# ATAPI CDROM
/dev/hdc      /cdrom          auto        ro,noauto,user,exec 0   0
/dev/hdd      /ddrom          auto        ro,noauto,user,exec 1   0
...
```

- Die Spalten bedeuten von links nach rechts:
  1. Der physikalische Ort des Dateisystems oder Blockgeräts.
  2. Der Mount Point. An dieser Stelle wird die Wurzel des Dateisystems in die Verzeichnisstruktur eingebaut.
  3. Der Typ des Dateisystems
  4. Die Optionen, die beim Mounten des Systems benutzt werden.
  5. Die Nummer legt fest, ob das Dateisystem bei einem Backup durch `dump` gesichert werden soll.
  6. Die Nummer legt fest, in welcher Reihenfolge `fsck` die Dateisysteme prüft.

1.13.6.5 Die Datei /etc/mtab

- In dieser Datei werden alle gerade gemounteten Dateisysteme aufgelistet.
- Der Befehl `mount` ohne Parameter benutzt diese Datei für seine Ausgabe.
- Die Befehle `mount` und `umount` verändern die Datei.

```
[533] htor@archimedes:~$ cat /etc/mtab
/dev/hda4 / ext3 rw 0 0
proc /proc proc rw 0 0
devpts /dev/pts devpts rw,mode=0620,gid=5 0 0
/dev/hda1 /boot ext3 rw 0 0
/dev/hdb5 /windrives/e vfat rw,nosuid,nodev,umask=000,quiet 0 0
```

```
/dev/hda5 /windrives/d vfat rw,nosuid,nodev,umask=000,quiet 0 0
shmfs /dev/shm shm rw 0 0
perikles:/mnt/hdb /mnt/perikles/hdb nfs
rw,nosuid,nodev,addr=192.168.100.254,user=htor 0 0
/dev/hdc /cdrom iso9660 ro,nosuid,nodev,user=htor 0 0
/dev/hdd /ddrom iso9660 ro,nosuid,nodev,user=htor 0 0
[534] htor@archimedes:~$
```

- Die Ähnlichkeit mit der Ausgabe des Befehls `mount` ist nicht zu übersehen.

1.14 Weitere nützliche Kommandos

Folgende Kommandos sind zusätzlich auf den meisten, aber nicht allen Unix-Systemen verfügbar

1.14.1 Allgemeine Kommandos

1.14.1.1 Das Kommando true

- true macht nichts, ausser den Status „erfolgreich“ zurückzugeben

1.14.1.2 Das Kommando false

- false macht nichts, ausser den Status „fehlgeschlagen“ zurückzugeben

1.14.1.3 Das Kommando sleep

- Das Kommando sleep wartet die angegebene Zeit (in Sekunden), bevor es sich selbst beendet

1.14.1.4 Das Kommando su

- Das Kommando `su` erlaubt einen Wechsel der Benutzeridentität bzw. UID während einer Sitzung.
- `su [OPTIONEN] [BENUTZER]`
- Die einfache Eingabe von `su` ohne einen Benutzernamen bewirkt einen Wechsel zur Superuseridentität nach der Eingabe des Superuserkennworts.
- Der Wechsel der Identität wird durch das Starten einer neuen Shell realisiert.
- Im Gegensatz zum Einloggen, bei dem auch eine Shell gestartet wird, bleibt beim Wechsel der Identität mit `su` die Umgebung des alten Benutzers erhalten.
- Nur die Umgebungsvariablen `HOME` und `SHELL` werden auf die entsprechenden Werte des neuen Benutzers geändert.
- Jeder Benutzer kann seine Identität wechseln, solange er das Kennwort des neuen Benutzers kennt.
- Nur der Superuser kann seine Identität wechseln ohne ein Kennwort eingeben zu müssen.
- Mit dem Befehl `exit` oder drücken der Tastenkombination `<STRG>+<D>` kehren Sie zur alten Identität zurück.

Optionen

`-, -l`	Die Konfigurationsdateien des Benutzers werden ausgeführt
`-c`	Führt das angegebene Kommando unter dem neuen
`KOMMANDO`	Benutzer aus

Beispiele:

`su`
mit anschließender Kennworteingabe läßt den Benutzer nun als Superuser arbeiten.
`su testnutzer`
mit anschließender Kennworteingabe läßt den Benutzer nun unter der UID des Benutzers *testnutzer* arbeiten. Der Superuser benötigt keine Kennworteingabe um die Identität zu wechseln.
`su - testnutzer`
führt im Gegensatz zum obigen Beispiel dazu, daß die Konfigurationsdateien von *testnutzer* mit

ausgeführt werden. Daher ist z. B. das Heimatverzeichnis von *testnutzer* jetzt das aktuelle Arbeitsverzeichnis.

1.14.1.5 Das Kommando uname

- Das Programm `uname` zeigt Informationen über das installierte System und den Kernel an.
- `uname [OPTIONEN]`
- Die Verwendung von `uname` ohne Schalter ist wie `uname -s`.

Optionen

`-a`	Gibt alle Informationen aus (`--all`)
`-m`	Gibt den Hardwaretyp aus (`--machine`)
`-n`	Gibt den Netzwerkname des Rechners aus (`--nodename`)
`-r`	Gibt die Release-Nummer des Systems aus (`--release`)
`-s`	Gibt den Systemnamen aus (`--sysname`)
`-p`	Gibt den Prozessortyp aus (`--processor`)
`-v`	Gibt die Versionsnummer aus (*version*)

Beispiele:

Der Befehl ohne Schalter gibt einfach nur den Namen des Betriebssystems aus.

```
[534] htor@archimedes:~$ uname
Linux
[535] htor@archimedes:~$
```

Werden mehrere Schalter oder der Schalter `-a` verwendet, dann werden die Informationen in der Reihenfolge `-snrvmp` ausgegeben.

```
[535] htor@archimedes:~$ uname -a
Linux archimedes 2.4.19-4GB #1 Tue Oct 15 02:57:46 UTC 2002 i686 unknown
[536] htor@archimedes:~$
```

1.14.1.6 Das Kommando bc

- `bc` gehört zu den Interaktiven Konsolenprogrammen
- nach dem Start kann man einfache Formeln eingeben, die automatisch berechnet werden
- beendet wird `bc` mittels des Befehls `quit` oder `<STRG>-<D>`

Beispiel:

```
[526] htor@archimedes:~$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
400+5
405
10*57
570
quit
[527] htor@archimedes:~$
```

1.14.1.7 Das Kommando dos2unix

- dos2unix konvertiert DOS-Textdateien in das Unix Textformat

1.14.1.8 Das Kommando unix2dos

- unix2dos konvertiert unix Textdateien in das DOS Textformat

1.14.2 Netzwerkfunktionen

1.14.2.1 Das Kommando ftp

- ftp dient zum Dateitransfer über das Internet
- Haupanwendung ist das Hochladen von Internetseiten auf Webserver
- `ftp [OPTIONEN] [ZIELRECHNER]`
- wichtige ftp-Kommandos sind:
  - `ascii` – setzt den Transfermodus auf Text
  - `binary` – setzt den Transfermodus auf beliebige Dateien
  - `bye` – beendet die ftp-Sitzung
  - `cd` – wechselt Arbeitsverzeichnis auf Server
  - `delete [Datei]` – löscht die angegebene auf dem Server
  - `get [Datei]` – lädt die angegebene Datei vom Server herunter
  - `lcd` – ändert das Arbeitsverzeichnis des lokalen Rechners
  - `ls` – Zeigt den Inhalt des Verzeichnisses auf dem Server an
  - `mkdir` – legt ein neues Verzeichnis auf dem Server an
  - `put [Datei]` – lädt die lokale Datei auf den Server hoch
  - `pwd` – zeigt das aktuelle Arbeitsverzeichnis auf dem Server
  - `rmdir` – löscht ein verzeichnis auf dem Server

Beispiel:

```
[547] htor@archimedes:~/test$ l
insgesamt 16
drwxr-xr-x    2 htor     users        4096 2003-03-16 11:57 ./
drwxr-xr-x   99 htor     users        8192 2003-03-16 11:35 ../
-rw-r--r--    1 htor     users           6 2003-03-16 11:57 test
[548] htor@archimedes:~/test$ ftp perikles
Connected to perikles (192.168.100.254).
220 ProFTPD 1.2.2rc2 Server (powered by SuSE Linux) [perikles.tohoe.de]
Name (perikles:htor): htor
331 Password required for htor.
Password:
230 User htor logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mkdir test1
257 "/home/htor/test1" - Directory successfully created.
ftp> cd test1
250 CWD command successful.
ftp> pwd
257 "/home/htor/test1" is current directory.
ftp> put test
local: test remote: test
200 PORT command successful.
150 Opening BINARY mode data connection for test.
226 Transfer complete.
6 bytes sent in 0.000125 secs (47 Kbytes/sec)
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
-rw-r--r--   1 htor     users           6 Mar 16 11:01 test
226 Transfer complete.
ftp> bye
221 Goodbye.
[549] htor@archimedes:~/test$
```

1.14.2.2 Das Kommando host

- host dient zur Auflösung des Hostnamens oder der zugehörigen IP
- `host [HOSTANAME]`

Beispiel:

Auflösung der Internetadresse von www.google.com:

```
[552] htor@archimedes:~/test$ host www.google.com
www.google.com has address 216.239.53.99
```

Auflösung des Hostnamens der IP 195.185.147.155:

```
[555] htor@archimedes:~/test$ host 195.185.147.155
155.147.185.195.IN-ADDR.ARPA is a nickname for 155.128/27.147.185.195.IN-
ADDR.ARPA
155.128/27.147.185.195.IN-ADDR.ARPA domain name pointer warlord.unixer.de
[556] htor@archimedes:~/test$
```

1.14.2.3 Das Kommando ping

- ping dient zum Testen, ob der angegebene Rechner im Netz erreichbar ist
- `ping [RECHNERNAME]`

Beispiel:
test, ob www.sat.de noch erreichbar ist:

```
[556] htor@archimedes:~/test$ ping www.sat.de
PING www.sat.de (212.223.54.254) from 192.168.100.1 : 56(84) bytes of data.
64 bytes from www.sat.de (212.223.54.254): icmp_seq=1 ttl=115 time=80.8 ms
64 bytes from www.sat.de (212.223.54.254): icmp_seq=2 ttl=115 time=81.4 ms
64 bytes from www.sat.de (212.223.54.254): icmp_seq=3 ttl=115 time=167 ms

--- www.sat.de ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2020ms
rtt min/avg/max/mdev = 80.850/110.012/167.687/40.783 ms
[557] htor@archimedes:~/test$
```

Der Befehl läuft, bis man ihn mit `<STRG>-<C>` abbricht!

1.14.2.4 Das Kommando finger

- finger dient zur Anzeige der angemeldeten Nutzer an einem Unix System
- finger [USER][@RECHNERNAME]

Beispiel:

```
htor@aphrodite:~> finger @borkum
[borkum.informatik.tu-chemnitz.de]
Login    Name            Tty      Idle  Login Time   Office     Office Phone
htor     Torsten Hoefler    pts/0      1d  Feb 28 09:22 (warlord.unixer.de)
htor     Torsten Hoefler    pts/19    1:14  Feb 20 16:33 (warlord.unixer.de)
htor@aphrodite:~>
```

1.14.2.5 Das Kommando ssh

- `ssh` ist eine Weiterentwicklung von telnet
- es bietet eine sichere Möglichkeit, um sich auf anderen Unix-Rechnern anzumelden
- `ssh [USER@][RECHNERNAME]`

Beispiel:

```
[560] htor@archimedes:~$ ssh www.unixer.de
htor@www.unixer.de's password:
Last login: Sat Mar 15 17:14:24 2003 from archimedes
Welcome to aphrodite!
htor@aphrodite:~>
```

1.14.2.6 Das Kommando scp

- `scp` dient zum sicheren Dateitransfer zwischen Unix-Rechnern (Weiterentwicklung von `ftp`)
- `scp [Datei] [USER@][RECHNERNAME]:[VERZEICHNIS]`

Beispiel:
Kopiert die Datei test/test in das Home-Verzeichnis auf www.unixer.de:

```
[561] htor@archimedes:~$ scp test/test htor@www.unixer.de:~
htor@www.unixer.de's password:
test                          100% |*****************************|     6        00:00
[562] htor@archimedes:~$
```

1.14.3 Arbeit mit DOS Disketten

- falls es mal nicht möglich ist, eine Diskette zu *mounten*, um beliebige Zugriffe auf die darauf befindlichen Dateien zu machen, gibt Programme, um Dateioperationen auf Disketten auszuführen.
- Diese sind unter dem Namen *mtools* bekannt.
- Die wichtigsten Befehle werden hier kurz vorgestellt.
- Wenn als Argumente zu einem Befehl DOS-Dateinamen zugelassen sind, so können auch Jokerzeichen verwendet werden.
- Diese folgen jedoch den üblichen Unix-Regeln (`*` sind alle Dateien, `*.*` nur solche mit Erweiterung)!
- man muß jedoch die Jokerzeichen in Anführungszeichen einschließen, da sonst die Shell versucht, sie zu ersetzen.
- Als Separatoren zwischen Verzeichnisnamen eines Pfades kann man `/` verwenden.
- Bei DOS-Dateinamen wird, im Gegensatz zu Unix, nicht zwischen Groß- und Kleinschreibung unterschieden.
- Beim Kopieren von DOS nach Unix werden die Dateinamen auf Kleinschreibung umgesetzt.
- In der Gegenrichtung werden zu lange Dateinamen gekürzt, wobei Punkte nötigenfalls in `X` umgesetzt werden. (es ist ja nur ein Punkt erlaubt)

1.14.3.1 Das Kommando mdir

- dient zum Auflisten des Inhaltes von Disketten
- `mdir [DOS-Dateiname]`
- `mdir` listet den Inhalt des Diskette in dem von `DIR` (unter DOS) gewohnten Format auf.
- Wenn Parameter angegeben sind, so werden nur die davon erfaßten Dateien gezeigt.

1.14.3.2 Das Kommando mcd

- wechselt das Arbeitsverzeichnis auf der DOS Diskette
- `mcd [DOS-Verzeichnis]`
- Das gewünschte Verzeichnis wird in der Datei `.mcwd` im HOME-Directory abgespeichert und von nachfolgenden *mtools*-Befehlen verwendet.
- Analog gibt es `mmd` zum Erzeugen eines DOS-Verzeichnisses.

1.14.3.3 Das Kommando mcopy

- dient zum Kopieren von und auf DOS Disketten
- ist der wichtigste Befehl der *mtools*.
- `mcopy [-t] [DOS-Dateiname] [Unix-Ziel]`
- `mcopy [-t] [Unix-Dateiname] [DOS-Ziel]`
- Entweder alle Ursprungs-Dateinamen müssen DOS-Dateien sein und das Ziel ein Unix-Verzeichnis (oder -Dateiname, falls nur eine Datei kopiert wird), oder umgekehrt.
- Damit `mcopy` zwischen DOS- und Unix-Dateinamen unterscheiden kann, darf hier im Gegensatz zu den anderen *mtools*-Befehlen das `a:` nicht weggelassen werden!
- Es ist nicht möglich, Dateien innerhalb der DOS-Diskette zu kopieren. `mcopy` fragt nach, bevor es eine existierende Datei überschreibt.

Beispiel:
`mcopy a: dos`
kopiert alle Dateien im aktuellen (oder Wurzel-)Verzeichnis der 1440kB-DOS-Diskette in das Unix-Verzeichnis `dos` innerhalb des aktuellen Verzeichnisses.

Warnung:
Beim Kopieren auf die Diskette werden lange Dateinamen abgeschnitten.
Dadurch kann es zu Namensgleichheiten kommen, und in Dateien enthaltene Dateinamen (z.B. `#include` in C) stimmen nicht mehr!
Falls die Diskette wieder auf ein System mit langen Dateinamen gebracht werden soll, empflehlt es sich, die Dateien zunächst mit `zip` oder `tar` zu archivieren, um die Dateinamen zu erhalten.

1.14.3.4 Das Kommando mdel

- löscht DOS Dateien auf Disketten
- `mdel [DOS-Dateiname]`
- Die angegebenen Dateien werden von der Diskette gelöscht.

1.14.3.5 Das Kommando mtype

- zeigt den Inhalt einer DOS Datei an
- `mtype [-t] [DOS-Dateiname]`
- `mtype` zeigt die angegebenen DOS-Dateien auf dem Bildschirm an. (vgl. `cat`)
- Die Ausgabe kann auch auch in eine Datei oder in eine Pipe umgeleitet werden

Beispiel:
`mtype readme |more`

1.14.3.6 Übersicht über die mtools

`mattrib`	`[Datei]`	ändert die MS-DOS Dateiattribute.
`mcd`	`Verzeichnis`	wechselt das Arbeitsverzeichnis auf der DOS Diskette.
`mcopy`	`[-tnvm] Quelle Ziel`	kopiert MS-DOS Dateien von/nach Linux. Besonders interessant ist die automatische Konvertierung von CR/LF in LF mit der Option `-t`.
`mdel`	`[-v] Datei`	löscht eine MS-DOS Datei.
`mdir`	`[-w] Datei`	zeigt den Inhalt eines DOS Verzeichnisses an.
`mformat`	`[ -t Spuren] [ -h Köpfe] [ -s Sektoren] [ -l Label] Laufwerk:`	legt ein DOS Dateisystem auf einer Low-Level formatierten Diskette an.
`mlabel`	`[-v] Laufwerk:`	erzeugt ein Volume-Label für eine MS-DOS Diskette.
`mmd`	`[-v] Verzeichnis`	legt ein Verzeichnis auf einer DOS Diskette an.
`mrd`	`Verzeichnis`	löscht ein Unterverzeichnis auf

		einer DOS Diskette.
`mread`	`[-tnm] Quelle Ziel`	liest (kopiert) eine Datei "`roh"' von DOS nach Linux.
`mren`	`[-v] Alt Neu`	benennt eine existierende DOS Datei um.
`mtype`	`[-ts] Datei`	gibt den Inhalt einer DOS Datei aus.
`mwrite`	`[-tnvm] Quelle Ziel`	schreibt (kopiert) eine Datei „roh“ von Linux auf eine DOS Diskette.

1.14.4 Kommandos zu Systemstatistiken

1.14.4.1 Das Kommando hostname

- hostname gibt den Rechnernamen des Systems aus
- `hostname`

Beispiel:
zeigt den Rechnernamen an
```
[585] htor@archimedes:~$ hostname
archimedes
```

zeigt die IP Adresse des Rechners
```
[586] htor@archimedes:~$ hostname -i
192.168.100.1
```

1.14.4.2 Das Kommando dmesg

- Der Befehl `dmesg` gibt letzten Meldungen des Kernels aus.
- In den meisten Fällen wird er dazu benutzt um sie die Bootmeldungen nach dem Starten des Systems noch einmal in Ruhe anzuschauen.
- `dmesg [OPTIONEN]`

1.14.4.3 Das Kommando free

- Der Befehl `free` zeigt eine Übersicht über die Auslastung des Arbeitsspeichers.
- `free [OPTIONEN]`

Optionen

`-b`	Angabe in Byte (*byte*)
`-k`	Angabe in Kilobyte(*kB*)
`-m`	Angabe in Megabyte(*MB*)
`-o`	Kein Anzeigen der Puffer/Cache-Korrektur(*ohne*)
`-s ZEIT`	Kontinuierliche Anzeige im Abstand von ZEIT Sekunden(*switchtime*)
`-t`	Zeigt eine Zeile mit den Werten aller Speicher an(*total*)
`-V`	Versionsnummer()

Beispiel:
```
[589] htor@archimedes:~$ free
             total       used       free     shared    buffers     cached
Mem:        515028     489848      25180          0     116832     180560
-/+ buffers/cache:     192456     322572
Swap:            0          0          0
```

```
[590] htor@archimedes:~$
```

1.14.4.4 Das Kommando df

- Der Befehl `df` zeigt die Nutzung der Partition an.
- `df [OPTIONEN] [DATEINAME]`
- Wird ein Dateiname angegeben, so zeigt `df` die Daten der Partition an, auf der die Datei liegt.
- Ansonsten zeigt er eine Liste aller Partitionen an.

Optionen

`-a`	Liefert Informationen über alle Dateisysteme (normal) (*all*)
`-h`	Zeigt durch einen Buchstaben die verwendete Einheit an(*human readable*)
`-i`	Zeigt die Nutzung der Inodes an(*inode*)
`-k`	Anzeige des Speicherplatzes in Kilobytes (normal)(*kilobyte*)
`-m`	Anzeige des Speicherplatzes in Megabytes(*megabyte*)
`-t DSTYP`	Zeigt nur Dateisysteme vom Typ `DSTYP`(*typ*)
`-T`	Zeigt den Dateisystemtyp für jeden Eintrag(*typ*)
`-x DSTYP`	Zeigt keine Dateisysteme vom Typ `DSTYP`(*exclude typ*)
`--sync`	Startet erst das `sync`-Kommando

- Als Information liefert `df`:
  - Größe des Gerätes
  - Anzahl der freien Blöcke
  - Anzahl der belegten Blöcke
  - Anzahl der freien Blöcke in %
  - Mountpoint

```
[590] htor@archimedes:~$ df
Dateisystem           1K-Blöcke   Benutzt Verfügbar Ben% Eingehängt auf
/dev/hda4               7404040   4582832   2445100  66% /
/dev/hda1                 46636      9092     35136  21% /boot
/dev/hdb5              10480152   9932288    547864  95% /windrives/e
/dev/hda5              20689600  20689600         0 100% /windrives/d
shmfs                    257512         0    257512   0% /dev/shm
perikles:/mnt/hdb      13045432  12217004    828428  94% /mnt/perikles/hdb
/dev/hdc                 570532    570532         0 100% /cdrom
/dev/hdd                 713370    713370         0 100% /ddrom
[591] htor@archimedes:~$
```

1.14.4.5 Das Kommando last

- Der Befehl `last` zeigt die Benutzer an, die sich zuletzt eingeloggt haben.
- Er durchsucht dafür die Datei `/var/log/wtmp`.
- `last [OPTIONEN] [BENUTZER]`
- Daneben zeigt er auch an, seit wann die Datei `/var/log/wtmp` Informationen enthält.

```
htor@kongo:~> last -10
htor     pts/1        helena.lr3.depag Sun Mar 16 13:54   still logged in
stein    pts/5        stein.lr3.depag. Fri Mar 14 10:48 - 13:29  (02:40)
koenig   pts/4        koenig.delta.de  Fri Mar 14 10:43 - 13:39  (02:55)
jfriedri pts/3        jfriedrich.delta Fri Mar 14 10:09 - 16:51  (06:41)
jk       pts/1        alderaan.lr3.dep Fri Mar 14 09:00 - 11:02  (02:01)
ic2151   pts/4        scherf2          Fri Mar 14 08:57 - 10:40  (01:43)
stein    pts/3        stein.lr3.depag. Fri Mar 14 08:45 - 09:57  (01:12)
delta    pts/2        engelmann2.delta Fri Mar 14 08:38 - 14:55  (06:16)
koenig   pts/1        koenig.delta.de  Fri Mar 14 08:33 - 08:58  (00:24)
jk       pts/0        alderaan.lr3.dep Fri Mar 14 08:30 - 11:01  (02:31)

wtmp begins Thu Mar  6 12:37:25 2003
htor@kongo:~>
```

1.14.4.6 Das Kommando who

- Wer ist denn zur Zeit eingeloggt?
- Diese Frage stellt sich dem Administrator immer dann, wenn er den Rechner herunterfahren will.
- Der Befehl `who` zeigt alle eingeloggten Benutzer mit ihren Terminals an.
- `who [OPTIONEN]`

Optionen

`-H`	Zeigt eine Überschriftenzeile
`-i, -u`	Zeigt an, wie lange ein Benutzer nicht mehr gearbeitet hat
`-m`	Zeigt den aktuellen Benutzer an
`-q`	Zeigt die eingeloggten Benutzer und ihre Anzahl an

- Für die Ausgabe des Befehls wertet er die Dateien `/var/run/utmp` und `/var/log/wtmp` aus.
- Der Befehl kann aber auch auf eines der wichtigsten Probleme im Leben eines Administrators eine Antwort geben: „Wer bin ich?“. Einfach `who am I` eingeben und die Antwort ist da.
- Wer es liebevoller mag, dem sei die Version `who mom likes` an Herz gelegt.
- Im Endeffekt sind aber diese Erweiterungen nur andere Namen für `who -m`.

1.14.4.7 Das Kommando w

- Der Befehl `w` zeigt wie `who` die eingeloggten Benutzer an.
- Er liefert allerdings weitergehende Informationen wie Einlogzeitpunkt, CPU-Nutzung und was der Benutzer gerade ausführt.
- `w [OPTIONEN]`

```
htor@aphrodite:~> w
  1:58pm  up 83 days,  3:04,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
htor     pts/8    perikles         24Feb03  7days  0.67s  0.67s  -bash
htor     pts/11   archimedes       Sat 5pm  2:22m  0.43s  0.43s  -bash
htor@aphrodite:~>
```

1.14.5 Kommandos zur Prozeßverwaltung

- Um sich eine Übersicht über die Prozesse zu verschaffen und übelgelaunte Prozesse, die das System belasten, zu identifizieren, können eine Reihe von Tools verwendet werden.

1.14.5.1 Das Kommando ps

- Um sich eine Liste der Prozesse und deren PIDs anzeigen zu lassen, kann man den Befehl `ps` verwenden.
- `ps [OPTIONEN]`
- Die Optionen bei `ps` werden ohne führenden Bindestrich geschrieben, weil der Befehl keine Parameter kennt.
- Der Befehl `ps` zeigt normalerweise die unter der eigenen UID laufenden Prozesse an.
- Die Option `-e` bewirkt eine Anzeige aller laufenden Prozesse.

Optionen

`a`	Zeigt alle Prozesse auf einem Terminal an
`e`	Zeigt die dazugehörigen Umgebungsvariablen
`f`	Zeigt die Ausgabe als Baum (Eltern- und Kindsprozesse)
`l`	Ausführliche Ansicht
`u`	Zeigt Benutzer und Startzeit an
`x`	Zeigt die Prozesse mit einem zugeordneten Terminal
`-e`	Zeigt alle Prozesse an

Beispiel:

```
[601] htor@archimedes:~$ ps
  PID TTY          TIME CMD
  956 tty1     00:00:00 bash
```

```
 4418 tty1     00:00:00 startx
 4419 tty1     00:00:00 tee
 4431 tty1     00:00:00 xinit4
 4449 tty1     00:00:05 fvwm2
 4463 tty1     00:00:00 FvwmButtons
 4464 tty1     00:00:00 FvwmTaskBar
 4467 tty1     00:00:12 xosview.bin
 4468 tty1     00:00:30 xosview.bin
 4469 tty1     00:00:01 FvwmPager
...
```

1.14.5.2 Das Kommando pstree

- Das Kommando `pstree` gibt einen Baum von allen Prozessen aus. Dadurch wird deutlich, welcher Prozeß von welchem anderen Prozeß gestartet wurde.
- `pstree [OPTIONEN] [PID]`

Optionen

`-a`	Zeigt die Kommandozeilenargumente an
`-c`	Verhindert die kompakte Darstellung von identischen Zweigen
`-l`	Zeigt lange Zeilen an, sonst werden die Zeilen für die Bildschirmbreite umgebrochen
`-n`	Sortiert die Prozesse nach PID und nicht nach Name
`-p`	Zeigt die PIDs zu den Prozessen an
`-u`	Zeigt an, wenn die UID von Kinds- und Elternprozeß sich unterscheidet

- Ohne weitere Parameter gibt `pstree` den Baum ab dem Prozeß `init` aus. Durch Eingabe einer PID kann die Ausgabe auf einen Teilast beschränkt werden.

Beispiel:

```
[607] htor@archimedes:~$ pstree -p 4449
fvwm2(4449)-+-FvwmButtons(4463)
            |-FvwmPager(4469)
            |-FvwmTaskBar(4464)
            |-sh(16783)---netscape(16788)---netscape(16803)
            |-xosview.bin(4467)
            |-xosview.bin(4468)
            |-xterm(4507)---bash(4509)---ssh(4514)
            |-xterm(14559)---bash(14561)
            |-xterm(14611)---bash(14613)
            |-xterm(14696)---bash(14698)---ssh(14857)
            |-xterm(14945)---bash(14947)---acroread(14978)
            |-xterm(15941)---bash(15943)
            |-xterm(15982)---bash(15984)
            `-xterm(17051)---bash(17053)---ssh(17058)
[608] htor@archimedes:~$
```

1.14.5.3 Das Kommando kill

- Es ist möglich mit den Prozessen über Signale zu kommunizieren.
- Das Signal kann eine Unterbrechung, eine illegale Anweisung oder andere Bedingungen anweisen.
- Das Kommando `kill` kann zur Sendung solcher Signale verwendet werden.
- Meistens wird es zum „töten“ eines Prozesses benutzt.
- `kill [OPTIONS] ID`
- Das Standardsignal ist das Beenden des Prozesses mit `SIGTERM`.
- Man spricht in diesem Fall auch von Terminieren.
- Nur der Superuser oder der Besitzer des Prozesses dürfen Signale senden.
- Die ID kann die PID, `%` (Das wäre der aktuelle Job), `%N` oder `%JOBNAME` sein.
- Eine Übersicht über die Signal liefert folgende Tabelle

Nr.	Langname	Kurzname	Bedeutung
1	SIGHUP	HUP	Hangup: Reinitialisierung des Prozesses
2	SIGINT	INT	Interrupt (wie <STRG>+<C>)
3	SIGQUIT	QUIT	Quit: Beenden
9	SIGKILL	KILL	Sofortiges Beenden des Prozesses (wird nicht ignoriert)
15	SIGTERM	TERM	Sofortiges Beenden des Prozesses (kann ignoriert werden)

1.14.5.4 Das Kommando killall

- Genau wie der Befehl `kill` leitet der Befehl `killall` Signale an Prozesse weiter.
- Allerdings wird bei diesem Befehl nicht die PID sondern der Name des laufenden Programm angegeben.
- Alle Prozesse, die diesem Namen zugeordnet sind, werden dann beendet.
- `killall [OPTIONS] ID`
- Die sonstige Arbeitsweise ist identisch mit `kill`.

1.14.5.5 Das Kommando nice

- Wie oft und schnell ein Programm CPU-Zeit bekommt hängt u. a. von seiner Prozeßpriorität ab.
- Dies Prozeßpriorität kann mit dem Befehl `nice` verringert werden.
- Nur der Superuser ist in der Lage einem Prozeß eine höhere Priorität zuzuordnen.
- `nice [OPTION] KOMMANDO`
- Das Kommando ist ein Befehl oder eine Kette von Befehlen, die mit der angegebenen Priorität ausgeführt werden können.
- Im Normalfall ist die Prozeßpriorität bei Verwendung von `nice` auf den Wert 10 festgelegt.
- Mit dem Schalter `-n ZAHL` kann eine Priorität zwischen `-20` und `19` angegeben werden.
- Dabei bedeutet eine kleinere Zahl eine höhere Priorität und eine große Zahl eine niedrige Priorität.
- Nur der Superuser ist in der Lage einen negativen Wert, und damit eine höhere Priorität als normal, einzustellen.

Beispiel:
```
nice -n 19 find / -name *.txt -print > textfdateien.txt
```
Dieser langwierige Prozeß bekommt eine sehr niedrige Ausführungspriorität zugewiesen. Er wird praktisch nur ausgeführt, wenn das System genug Zeit hat.

1.14.5.6 Das Kommando renice

- Der Befehl `renice` erlaubt eine Änderung der Prozeßpriorität im Gegensatz zu `nice` für laufende Programme.
- `renice PRIORITÄT PROZESS`
- Dabei kann der Zielprozeß durch Angabe der PID bestimmt werden.
- Mit den Schaltern `-u` und `-g` kann sich auf alle Prozesse einer Benutzers oder einer Gruppe bezogen werden und mit `-p` weitere PIDs angegeben werden.

1.14.5.7 Das Kommando top

- Ein anderer Weg die laufenden Prozesse zu betrachten ist der Befehl `top`.
- Er erlaubt daneben auch gleichzeitig Statistiken über den Speicher und die Swap-Datei. Systemlaufzeit, CPU-Status und Prozeßgröße sind weitere Angaben, die das Tool liefert.
- Im Gegensatz zu `ps` gibt `top` nicht nur einen Schnappschuß der Prozesse wieder, sondern erlaubt eine kontinuierliche Beobachtung.

Beispiel:
```
2:24pm  up 68 days, 23:23,  1 user,  load average: 0.00, 0.00, 0.00
51 processes: 50 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  0.6% user,  0.2% system,  0.0% nice,  0.3% idle
```

```
Mem:   130664K av,  119756K used,   10908K free,       0K shrd,   61824K
buff
Swap:  310456K av,   29200K used,  281256K free                   14132K
cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
11985 htor      12   0  1052 1052   832 R     6.3  0.8   0:00 top
    1 root       0   0   216  216   180 S     0.0  0.1   0:12 init
    2 root       0   0     0    0     0 SW    0.0  0.0   0:04 kflushd
    3 root       0   0     0    0     0 SW    0.0  0.0   3:07 kupdate
    4 root       0   0     0    0     0 SW    0.0  0.0   1:26 kswapd
    5 root     -20 -20     0    0     0 SW<   0.0  0.0   0:00 mdrecoveryd
  206 bin        0   0   424  420   344 S     0.0  0.3   0:00 portmap
  222 root       0   0   668  652   536 S     0.0  0.4   1:23 sshd
  230 root       0   0   592  592   484 S     0.0  0.4   0:03 syslogd
  234 root       0   0   880  876   348 S     0.0  0.6   0:00 klogd
  241 root       0   0     0    0     0 SW    0.0  0.0   0:05 rpciod
  242 root       0   0     0    0     0 SW    0.0  0.0   0:00 lockd
  259 at         0   0   352  328   264 S     0.0  0.2   0:00 atd
  293 root       0   0  1092 1092   884 S     0.0  0.8   0:00 safe_mysqld
  331 named      0   0  1756 1004   736 S     0.0  0.7   0:00 named
  336 mysql      0   0  4464 4464  2212 S     0.0  3.4   0:23 mysqld
```

1.14.6 Der Midnight Commander

- für Dateioperationen ist auch der Midnight Commander (`mc`) zu nennen, dieser ist jedoch nur auf wenigen Systemen verfügbar
- die Bedienoberfläche ist selbsterlärend, eine genaue Beschreibung ist in der `man`-Page verfügbar
- eine weitere Erläuterung wird deshalb nicht gegeben
- `mc`