

# Eine kleine Linux-Kommandoreferenz mit Beispielen

Axel Pemann

7. Juli 2014

## Inhaltsverzeichnis

<b>1</b>	<b>Umgang mit der Shell</b>	<b>2</b>
1.1	Auffindung und Interpretation von Kommandos . . . . .	2
1.2	Für Shells scripting . . . . .	2
1.3	Hilfestellungen . . . . .	3
1.4	Grundeinstellungen . . . . .	3
<b>2</b>	<b>Hilfe-Kommandos</b>	<b>4</b>
<b>3</b>	<b>Verzeichnis-Kommandos</b>	<b>5</b>
<b>4</b>	<b>Datei-Kommandos</b>	<b>6</b>
4.1	Allgemeine Benutzerwerkzeuge . . . . .	6
4.2	Suchwerkzeuge . . . . .	7
4.3	Dateirechte verwalten . . . . .	8
4.4	Dateien/Verzeichnisse vergleichen . . . . .	8
4.5	Packprogramme . . . . .	8
4.6	Editor vim . . . . .	9
<b>5</b>	<b>Datei-Kommandos: Textfilter</b>	<b>11</b>
<b>6</b>	<b>Dateisystem-Kommandos</b>	<b>12</b>
6.1	Überblick: Linux Mount-Mechanismus . . . . .	14
<b>7</b>	<b>Prozessverwaltung</b>	<b>17</b>
<b>8</b>	<b>Benutzerverwaltung</b>	<b>20</b>
<b>9</b>	<b>Netzwerk-Kommandos</b>	<b>22</b>

# 1 Umgang mit der Shell

## 1.1 Auffindung und Interpretation von Kommandos

**bash** Die Bourne Again Shell ist der verbreitetste Interpreter für Unix/Linux-Kommandos. In Shell-Skripten wird gern das Vorgängermodell, die 'sh' verwendet. Welche Shells als Login-Shells benutzbar sind, beschreibt die Datei `/etc/shells`. Die 'bash' ist nicht nur Interpreter, sondern bietet auch Hilfestellungen mittels History, Kommandozeilen-Editor und Alias-Mechanismus. Weiteres unter: [http://de.linwiki.org/index.php/Linuxfibel\\_-\\_Shells\\_-\\_Bash](http://de.linwiki.org/index.php/Linuxfibel_-_Shells_-_Bash)

**enable** Ein Shell-Build-In, das alle internen Kommandos der Bash ausgibt

**type** Gibt Informationen zum Type des Kommandos aus und kann zur Ermittlung, ob ein Programm intern in der Shell eingebaut ist oder extern im Dateisystem gesucht wird, eingesetzt werden. Dabei durchforstet es den aktuellen Suchpfad (= Ausgabe: `echo $PATH`) nach dem Programm. Wenn es auffindbar ist, wird der Pfad angezeigt. Handelt es sich aber um ein internes Kommando der Shell, wird nur `is a shell builtin` ausgegeben. Des Weiteren werden Aliase als solche erkannt. Falls bei der Prüfung `is hashed` vorkommt, bedeutet dies, dass das Programm in dieser Bash-Sitzung schon einmal ausgeführt wurde und die Shell den Pfad aus dem Hash-Verzeichnis entnommen hat.

```
type date # -> date is /bin/date
type cd   # -> cd is a shell builtin
type ls   # -> ls is aliased to ls --color=auto
date && type date # -> date is hashed (/bin/date)
```

**wereis** Ausgeben von Binary- Source- und MAN-Pagepfad eines Kommandos

## 1.2 Für Shellscripting

**echo** Kopiert Zeichenketten vom Standard-Input auf den Standard-Output, wobei dabei per Default eine neue Zeile erzeugt wird.

```
echo 'Hallo an stdout!' # Simple ticks wegen dem Ausrufezeichen
echo -e "Text_\rmit_\rAlarmton\a"
echo "nameserver_194.25.2.129" > /etc/resolv.conf
```

**test** Kommando für Shell-Skripte, um verschiedenste Prüfungen vorzunehmen. Die Auswertung erfolgt über den Exit-Code (= Ausgabe von: `echo $?`; '0' bedeutet Erfolg, '1' bedeutet kein Erfolg):

```
test -f /etc # Ist /etc ein File?
echo $? # Die Zeile gibt 1 aus, Bedeutung: "Nein"
```

Ein komplexeres Beispiel: «Ist die Variable 'var' leer, weise ihr neuen Inhalt zu.»:

```
test -z 'echo $var' && var=NeuerInhalt
```

**read** Liest von der Tastatur und speichert die gelesenen Daten in einer Variablen

**source** Ein Shellsript ohne Subshell ausführen, wichtig für Ersetzungen von Variablen, die in der aktuellen Shellsitzung wirksam werden sollen. So kann man z.B. Profile-Dateien nach Änderungen ohne Reboot einlesen lassen und somit auf Funktion prüfen. Ein Kürzel für dieses Kommando ist ein Punkt, gefolgt von einem Leerzeichen.

```
source /etc/profile
source ~/.bashrc
. executable.file
```

### 1.3 Hilfestellungen

**history** Die Bash führt eine Liste bereits eingegebener Kommandozeilen, die man mit den Cursorstasten oder mit STRG+R durchsuchen kann. Mit dem Befehl 'history' kann man diese Liste direkt einsehen und mit '!zeilennummer' eine bestimmte Zeile wiederholt ausführen. Mit '!\$' kann man das letzte Argument der zuletzt ausgeführten Kommandozeile (z.B. ein langer Pfad) in der aktuellen Zeile erneut verwenden.

```
history 20 > theLastTwenty.hist # Zwanzig Zeilen in neue Datei
```

**Tabulator-Taste** Autovervollständigung für Kommandos, Datei- und Verzeichnispfade sowie Variablen- und Benutzernamen

**Kommandozeileneditor** Es existieren viele Shortcuts in der Bash (vom emacs-Editors geerbt), z.B. STRG + K für 'Löschen bis zum Zeilenende'

**alias** Um nicht immer lange Zeilen erneut eintippen zu müssen und um gefährliche Kommandos zu entschärfen, kann man Aliase definieren. Sollen sie nach einem Rechnerneustart automatisch zur Verfügung stehen, können sie z.B. in die Datei '~/.bashrc' geschrieben werden.

**clear** Den Bildschirminhalt des Terminals löschen (Entspricht STRG+L).

**reset** Löscht ebenfalls den Bildschirminhalt, setzt dabei aber das Terminal auf seine Voreinstellungen zurück. Es kann einfach blind eingetippt werden, falls die Ausgabe nicht mehr leserlich dargestellt werden kann.

```
alias subu='ssh -X ubu@192.168.12.100'
alias rm='rm -i'
```

### 1.4 Grundeinstellungen

**set** Mit diesem Kommando (set shell flags) lassen sich Shellvariablen ausgeben und ändern. So lässt sich z.B. das Verhalten im Fehlerfalle steuern.

```
set | more # Voreinstellungen ausgeben
set -x # Gibt die durchgefuehrten Ersetzungen aus
```

## 2 Hilfe-Kommandos

**man** Die Handbuchseiten (manual pages) für Linux-Kommandos, Konfigurationsdateien, Bibliotheken u.a.

```
man man      # Die Handbuchseite zum Hilfesystem
man cp       # Die Handbuchseite zum Kommando cp
man fstab    # Die Handbuchseite zur Datei /etc/fstab
man -k begriff # Suche nach einem Begriff im man-System (entspricht 'apropos')
man -a cp    # Alle Handbuchseiten zu einem Kommando oeffnen
```

**apropos** Suche nach Stichwörtern im man-System (in den whatis-Datenbankdateien, entspricht 'man -k', siehe oben)

**whatis** Gibt eine kurze Beschreibung über ein Kommando aus der whatis-Datenbank aus

**makewhatis** (Neu-)Aufbau der whatis-Datenbankdateien für das man-System

**help** Gibt eine Beschreibung über interne Kommandos der Bash aus

```
help enable
help cd
help alias
```

**info** Liest Dokumentationen, die im Info-Format vorliegen. Grundlegende Navigation:

- Scrollen: mit den Cursor-Tasten
- In Menüs stellen Sterne Hyperlinks dar, auf die man den Cursor stellen kann und danach mit Enter zum Linkziel springt.
- Mit 'l' (last) geht es wieder zum Ausgangspunkt zurück.
- Suchen kann man mit 'i' (index).
- Das Beenden des Programmes funktioniert wie bei 'man' mit: 'q'.

Bequemes Arbeiten mit info-pages ermöglicht der 'konqueror', beispielsweise mit der Eingabe von `info:/info` in der Adressleiste.

### 3 Verzeichnis-Kommandos

**ls** Auflisten (list) von Verzeichnisinhalten, zu allgemeinen Optionen siehe auch `ls` als Datei-Kommando. Um **nicht** die Inhalte eines Verzeichnisses aufzulisten, sondern lediglich die vergebenen Rechte am betreffenden Verzeichnis zu betrachten, ist die Option `'-d'` nützlich.

```
ls -ld /tmp # Nicht hineinschauen, nur das Verzeichnis selbst
```

**cd** Mit 'Change Directory' ist es möglich, in verschiedene Verzeichnisse zu wechseln. Danach befindet man sich in einem neuen Arbeitsverzeichnis.

```
cd / # In die System-Wurzel wechseln
cd - # Wieder ins vorherige Arbeitsverzeichnis wechseln
cd # Ohne jeden Parameter ins Heimatverzeichnis springen
cd .. # Ins Elternverzeichnis wechseln (eine Etage hoeher)
```

**pwd** Das aktuelle Arbeitsverzeichnis ausgeben (Print Working Directory)

**mkdir** Ein neues Verzeichnis erzeugen. Optional können mit `'-p'` nicht vorhandene Eltern-Verzeichnisse (Parents) erzeugt werden, mit `'-m'` lässt sich im gleichen Atemzug der Modus für die Zugriffsrechte, wie es das Kommando `chmod` ausführen würde, setzen.

```
mkdir -p auch/die/eltern/mit/erzeugen
mkdir -m 1777 /srv/austausch
```

**rmdir** Mit diesem Kommando lassen sich nur leere Verzeichnisse löschen. Damit sollte man stets arbeiten, wenn man sicher gehen will, dass Verzeichnisse wirklich leer sind. Ohne Rücksicht auf Verluste kann mit `rm -r` gelöscht werden.

## 4 Datei-Kommandos

### 4.1 Allgemeine Benutzerwerkzeuge

**ls -l** Mit `ls` und der Option `'-l'` (long) lassen sich ausführliche Informationen von Dateieigenschaften ausgeben.

```
ls -l ~/.bashrc # Eigenschaften der versteckten Datei .bashrc
ls -la # Alle Dateien im Home-Verz. auflisten, + versteckte
ls -ltr # Nach Aenderungszeit sortieren, juengstes unten
# --> Guenstig, wenn Verz. viele Dateien beinhalten.
ls -ltu # Nach der Zugriffszeit (use) sortieren
ls -ltc # Nach der Aenderungszeit der I-Node sortieren
ls -lS # Inhalt des aktuellen Verz. nach Dateigroesse sortieren
```

**file** Bestimmung des Dateitypes anhand eines Typvergleiches mit Hilfe von drei Tabellen. Wird `'data'` ausgegeben, konnte das Programm den Typ nicht ausmachen.

```
file /bin/date
```

**touch** Zeitstempel ändern oder, falls die Datei nicht existiert, eine neue Datei erzeugen. Linux-Dateisysteme arbeiten mit drei Werten: Zeit der letzten Änderung, Zeit des letzten Zugriffs und Zeit der letzten I-Node-Änderung. Das Kommando ändert ohne Option Änderungs- und Zugriffszeit, mit `'-m'` nur die Änderungszeit (Modification) und mit `'-a'` nur die Zugriffszeit (Access). Kommandos, die Zeitstempel auswerten können sind `ls` und `find`.

```
touch testdatei.txt # Aenderungs- und Zugriffszeit anpassen
```

**rm** Dateien und Verzeichnisse löschen (remove). Mit der Option `'-r'` lassen sich Verzeichnisse löschen, vor allem wenn sie nicht leer sind. Leere Verzeichnisse entfernt man mit `rmdir`.

```
rm -i testdatei.txt # Datei mit Rueckfrage (interact.) loeschen
rm -r verzeichnis # Verz. samt Inhalt (recursive) loeschen
```

**mv** Dateien und Verzeichnisse verschieben oder Umbenennen. Das Kommando arbeitet immer rekursiv. Mit der Option `'-b'` werden Backupkopien angelegt, falls die Datei bereits schon existiert und überschrieben werden würde. Dies funktioniert nur mit Dateien. Die Option `'-i'` steht für interaktiv und startet eine Nachfrage, bevor es eine existierende Datei überschreibt.

```
mv hallo.txt hallo.txt
mv hallo.txt /tmp
mv Verzeichnisstruktur/ Backup.dir/
```

**cp** Dateien und Verzeichnisse kopieren und evl. zugleich Umbenennen. Das Kommando arbeitet nur mit der Option `'-r'` rekursiv. Mit der Option `'-b'` werden Backupkopien angelegt, falls die Datei bereits schon existiert und überschrieben werden würde. Dies funktioniert nur mit Dateien.

Die Option `-i` steht für „interactive“ und startet eine Nachfrage, bevor es eine existierende Datei überschreibt. Ein wichtiger Schalter ist `-a` (archiv). Damit bleiben Eigentümer-Kennungen erhalten und es werden eventuell vorhandenen symbolischen Links nicht gefolgt.

```
cp -i quelldatei.txt zieldatei.txt
cp datei1 datei2 datei3 verzeichnis/
cp -a /etc /root/etc_backup
```

**ln** Erzeugen von Links auf Dateien bzw. Verzeichnisse. Hardlinks müssen innerhalb einer physischen Partition liegen und funktionieren nicht mit Verzeichnissen. Symbolische Links (= herkömmliche Verknüpfungen) brechen diese Beschränkungen.

```
ln spiegel.txt mirror.txt # Ein zweiter Henkel fuer den Topf
ln -s /dev/ttyS0 /dev/modem
```

**cat** Mit conCATenate steht ein Werkzeug zur Verfügung, mit dem sich ein- oder mehrere Dateien nacheinander öffnen und ausgeben lassen. Es kann auch mit binären Dateien umgehen.

```
cat /etc/*.conf > multiple-configs.catenated
```

## 4.2 Suchwerkzeuge

**find** Mächtiges Kommando zur rekursiven Dateisuche innerhalb von Verzeichnisstrukturen. Es arbeitet mit verschiedensten Bedingungen und kann auch verschiedene Aktionen ausführen. Die allgemeine Syntax lautet:

```
find [pfad] [pfadtiefe] [beding1] [beding2] ... [aktion1] [aktion2] ...
```

Wird der Startpfad `'pfad'` nicht angegeben, wird im aktuellen Arbeitsverzeichnis mit der Suche begonnen.

```
# Nur bis zur Tiefe von 3 Ebenen nach Verz. mit 'sbin' suchen:
find / -maxdepth 3 -type d -name "*sbin*"
# Files suchen, die nicht aelter als 5 min sind:
find -cmin -5
# Files Suchen, die groesser als 10000 Kilobyte sind:
find -size +10000k
```

**locate** Schnelle Suche nach Dateien und Verzeichnissen in einer mit `'updatedb'` zu erzeugenden Index-Datenbank. Als einfacher Benutzer kann man den Index nur im eigenen Home-Verzeichnis erstellen lassen: `'updatedb -o ~/suchindex.db'`, die Suche kann dann mit `'locate -d suchindex.db suchpattern'` erfolgen.

```
locate /etc/ # Nach 'etc' als vollstaendigen Namensteil suchen
locate -r '\.iso\$' # Suche mit regulaerem Ausdruck
```

### 4.3 Dateirechte verwalten

**chown** Change Owner: Die Eigentumsverhältnisse für Dateien und Verzeichnisse ändern. Ein Kommando nur für den Superuser 'root'. Die rekursive Option lautet '-R'. Will man gleichzeitig Benutzer- und Gruppenzugehörigkeit ändern, hängt man dem Benutzernamen einen Punkt als Trenner und danach den Gruppennamen an.

```
chown max maxens-brief.txt
chown -R max.users /home/max
```

**chmod** Change-Modus: Datei-Zugriffsattribute (Detailrechte) ändern. Es gibt einen Oktal- und einen Verbalmodus. Der Oktalmodus beschreibt immer absolute Werte (von 0...7), der Verbalmodus dagegen arbeitet relativ zum vorher gesetzten Wert. Die rekursive Option lautet '-R'.

```
chmod -R 600 /srv/www/htdocs/private/
chmod ug+x unser-skript.sh
```

### 4.4 Dateien/Verzeichnisse vergleichen

**diff** Textdateien auf ihren Inhalt vergleichen, mit der Option '-r' auch Verzeichnisse samt ihrem Inhalt

**comm** Zwei sortierte Dateien Zeile für Zeile vergleichen

**cmp** Vergleicht zwei Dateien byteweise

### 4.5 Packprogramme

**gzip/gunzip** Komprimiert/dekomprimiert Dateien mit dem LZ77 Lempel-Ziv Algorithmus, kein Archivpacker, arbeitet nur an einzelnen Dateien, Dateiendung: .z, .gz

**bzip2/bunzip2** Frei verfügbares Kompressions-/Dekompressionprogramm ähnlich wie 'gzip', effizientere Kompression, aber langsamer

**zip/unzip** Komprimiert/dekomprimiert Dateien mit einem offengelegten Verfahren, es arbeitet mit einem Datencontainer, es kann sogar verschlüsselte Archive erstellen: 'zip -e -r verzeichnis.zip verzeichnis/'

**rar/unrar** Komprimiert/dekomprimiert Dateien; es bietet eine gute Kompressionsrate, ausgewählte Dateien werden nicht einzeln, sondern ähnlich wie bei einem Tarball gemeinsam

**7z** Komprimiert/dekomprimiert Dateien mit der hoher Kompressionsrate (eingebauter Algorithmus: LZMA)

**xz** Neues Dateiformat mit noch höherer Kompressionsrate, basiert auf LZMA

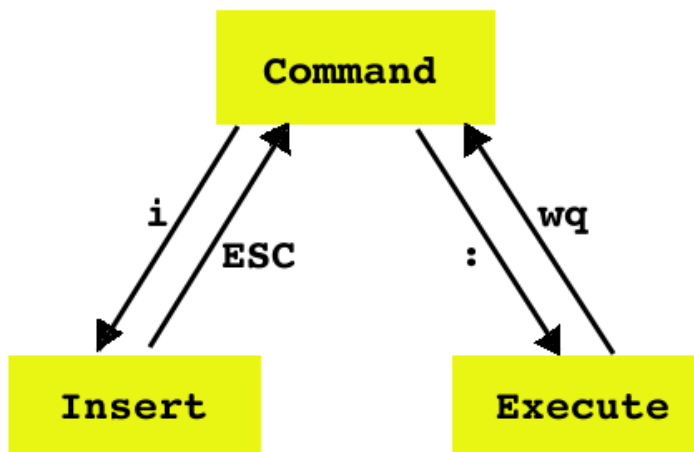


**tar** Verwaltet Dateiarhive (TApe aRchivierer); zusammen mit **gzip** (Schalter '-z') bzw. **bzip2** (Schalter '-j') oder **xz** (Schalter '-J') können Dateien/Verzeichnisse auch komprimiert werden, die gepackte Archivdatei nennt man 'Tarball' Hierzu folgende Beispiele mit **gzip** als Kompressionsverfahren:

```
tar -zcvf backup_etc.tar.gz /etc # Verz. /etc als kompr. Archiv
tar -zxvf backup_etc.tar.gz     # Das Archiv wieder entpacken
```

## 4.6 Editor vim

Der Editor vim ist eine erweiterte Version des einfachen Editors 'vi' (vi-iMproved), der zwar sehr leistungsfähig, aber auch etwas schwieriger zu handhaben ist. Wird eine Datei geöffnet, befindet man sich standardmäßig im **Command-Modus**, in dem nicht geschrieben wird, sondern typischerweise Kopier-, Ausschneide- und Einfügeoperationen durchgeführt werden. Die Grafik verdeutlicht dies und zeigt ein paar wichtige Tasten für den Moduswechsel:



Die folgenden Tabellen verstehen sich als 'Spickzettel', um wichtige Tastenkürzel schneller zur Hand zu haben:

## Command-Modus (Startmodus)

Kürzel	Bedeutung/Aktion
dd (delete)	Aktuelle Zeile in den Standard-Puffer verschieben
D	Von Cursorposition bis Ende der Zeile in den Puffer verschieben
yy (yank)	Aktuelle Zeile in den Standard-Puffer kopieren
3yy	Drei Zeilen kopieren (voran gestellte Zahlen erweitern die Befehle)
p (put)	Pufferinhalt nach Cursorposition einfügen
gg (goto)	Zum Dateianfang springen
G	Zum Dateiende springen
<i>zahl</i> G	Gehe zu Zeile (Sprung zu Zeile mit der Nummer <i>zahl</i> )
v (visual)	Visual-Modus: Textbereiche mit Cursortasten markieren
u oder STRG+r	Aktionen rückgängig machen (undo) oder wiederherstellen (redo)

## Insert-Modus (Textmodus)

o (open)	Eine Zeile unterhalb Cursor in den Textmodus schalten
i (insert)	Bei Cursor in den Textmodus schalten, links vom Zeichen
a (append)	Bei Cursor in den Textmodus schalten, rechts vom Zeichen
A	Am Ende der Zeile in den Textmodus schalten
r (replace)	Zeichen ersetzen (Replace-Modus)

## Execute-Modus (Last-Line-Modus)

:w (write)	Datei Speichern
:w <i>name</i>	Speichern unter ...
:q! (quit)	Beenden und Änderungen verwerfen
:wq!	Datei Speichern, wenn möglich über Dateirechte hinwegsetzen
!: <i>kommando</i>	Ausführen eines (beliebigen) Kommandos
/ <i>zeichenfolge</i>	Suchen nach <i>zeichenfolge</i> , mit n bzw. N weitersuchen
:1,\$s/ <i>heute</i> / <i>morgen</i> /g	Ersetzt <i>heute</i> durch <i>morgen</i> im gesamten Dokument

### Syntax-Highlighting aktivieren

In der Datei */etc/vim/vimrc* (oder */etc/vimrc*) den Eintrag «syntax on» aktiv setzen.

## 5 Datei-Kommandos: Textfilter

**head** Gibt standardmäßig die ersten zehn Zeilen einer Datei aus. Mit der Option '-n' kann man eine Zahl angeben, um die Standardeinstellung zu ändern.

**tail** Gibt standardmäßig die letzten zehn Zeilen einer Datei aus. Mit der Option '-n' kann man eine Zahl angeben, um die Standardeinstellung zu ändern. Interessant ist auch die Option '-f' (follow), die es dem Programm gestattet, einem sich veränderndem Datenstrom zu folgen.

```
tail -f /var/log/messages # Ein Ereignismonitor
```

**more** Ein Pager, mit dem seitenweise geblättert werden kann. Bedienung mit den Cursortasten.

**less** Ein erweiterter Pager, mit dem seitenweise auch rückwärts geblättert werden kann. Bedienung mit den Cursortasten, beenden mit 'q', suchen mit '/', weitersuchen mit 'n', rückwärts suchen mit 'N'.

**grep** Suche nach Zeichenketten in Textdateien, Einsatz von regulären Ausdrücken. Ein wichtiges Hilfsmittel, um zielgerichtet nach verschiedensten Informationen zu fahnden.

```
dmesg | grep sd # Nach einem neuen SCSI-Geraet (Stick) suchen  
grep -v '^#' /etc/hosts # Nur aktivierte Zeilen ausgeben
```

```
du -s /* | sort -n # Belegung der Verzeichnisse berechnen
```

**sed** Der 'Streaming EDitor' - Editor zur nicht-interaktiven, automatischen Textbearbeitung, im Gegensatz zum 'vi' kann dieser Editor Ersetzungen innerhalb einer Pipe, so zu sagen im Textstrom, vornehmen. Im folgenden Beispiel ist als Ausgangspunkt nur die Datei 'alt.txt' vorhanden, die an mehreren Stellen 'alt' beinhaltet. Die Datei 'neu.txt' wird nach der Ersetzung erzeugt.

```
sed 's/alt/neu/g' < alt.txt > neu.txt # Aus 'alt' mach 'neu'!
```

**wc** Word Count - Woerter, Zeichen und Zeilen zählen

**cut** Felder aus Dateien ausschneiden, eine Angabe des Feldtrenners ist moeglich

```
cut -d: -f1,3 /etc/passwd # Benutzernamen und UID extrahieren
```

**awk** Eine Zeichenketten verarbeitende Programmiersprache, oft verwendet, um Felder aus Dateien zu extrahieren, die als Trenner Tabulator- oder Leerzeichen verwenden.

```
cat /etc/fstab | awk '{print$1}' # Nur Geraetdateien ausgeben
```

**uniq** Duplikate Zeilen entfernen, Voraussetzung: der Zeichenstrom ist sortiert.

```
cat unsort-doppelt.txt | sort | uniq > sort-nicht-doppelt.txt
```

## 6 Dateisystem-Kommandos

**fdisk** Umfangreiches, schwieriger zu bedienendes Partitionierungsprogramm, aber gelistet ist schnell:

```
fdisk -l [/dev/sda] # Partitionen der ersten Platte auflisten
```

```
Platte /dev/sda: 82.3 GByte, 82348277760 Byte
255 Koepfe, 63 Sektoren/Spuren, 10011 Zylinder
Einheiten = Zylinder von 16065 x 512 = 8225280 Bytes
```

Geraet	boot.	Anfang	Ende	Bloecke	Id	System
/dev/sda1	*	1	1019	8185086	7	HPFS/NTFS
/dev/sda2		1020	1141	979965	82	Linux Swap
/dev/sda3		1142	1749	4883760	83	Linux
/dev/sda4		1750	10011	66364515	5	Erweiterte
/dev/sda5		1750	1871	979933+	83	Linux

**cdisk** Einfacheres Partitionierungsprogramm, evl. muss die Gerätedatei angegeben werden

**mkfs** Dateisysteme erzeugen, ein ext3-Dateisystem (incl. Journal) z.B. mit: mkfs.ext3, VORSICHT: Es wird ohne Rückfrage sofort ausgeführt!

```
mkfs.ext3 /dev/sda5 # Die erste log. Partition "formatieren"
```

**fsck** Prüfen von Dateisystemen auf Konsistenz; wenn die automatische Prüfung beim Booten fehlschlägt, kann sie durch Root später am ungemounteten Device erzwungen werden

```
fsck.ext3 -f /dev/sda5 # Erzwingen (force) eines Checks
```

**mkswap** Vorbereiten einer Partition zum dynamischen Auslagern von Daten aus dem RAM

```
mkswap /dev/sda2 # Zweite Partition fuer Swap formatieren
```

**swapon** Aktivieren und Benutzen der Swap-Partition, Erfolgskontrolle mit 'free'

```
swapon /dev/sda2 # Zweite Partition fuers RAM-Swapping zugaenglich machen
```

**swapoff** Deaktivieren der Swap-Partition, Erfolgskontrolle mit 'free '

```
swapoff /dev/sda2
```

**mount** Ohne weitere Argumente: Ausgabe aller gemounteten Ressourcen, ansonsten Montieren (Einhängen) von Datenträgern bzw. Netzressourcen (automatisch beim Systemstart mittels der Datei /etc/fstab)

```
mkdir /mnt/sda5 # Falls nicht vorhanden, leeren Ordner fuer Mountpunkt anlegen
mount /dev/sda5 /mnt/sda5 # Ein Device dahinein mounten
mount 192.168.12.100:/srv/pub /mnt/pub # Eine Netzressource mounten
```

**umount** Aushängen (Demontieren) von Datenträgern bzw. Netzressourcen

```
umount /mnt/sda5 # SDA5 wieder aushaengen
umount -a # Alle nicht verwendeten Partitionen aushaengen (Nur Root!)
```

**df** Ausgabe einer Belegungsübersicht der verwendeten Ressourcen (disk free), mit der Option '-h' wird das Format 'human readable' benutzt, dh. es werden die Einheiten automatisch in eine lesbarere Größe umgerechnet:

```
df -h # Wieviel Plattenplatz ist auf den Geraeten verfuegbar?
df -h / # Plattenplatz der Wurzel-Partition
df -h . # Sehr praktisch: Plattenplatz der Partition, auf der ich gerade
arbeite (echo $PWD)
```

**du** Berechnung der Größe eines Verzeichnisses oder mehrerer (disk usage); die Ausgabe gibt die Größe der belegten Blöcke an. Mit der Option '-h' wird es wieder lesbarer, mit '-s' wird nur die Summe für das angegebene Verzeichnis berechnet:

```
du -sh /home # Wie gross ist das Verzeichnis /home?
du -s /* | sort -n # Sortierte Summen-Ausgabe in der Wurzel (Speicherfresser
finden)
```

**dd** Mit dd (disk dump) können ganze Festplatten geklont werden (allerdings recht langsam, da jeder einzelne Block kopiert wird) oder nur einzelne Sektoren, um z.B. den MBR zu sichern:

```
mount /dev/sdb1 /mnt/sdb1 # USB-Stick mounten
dd if=/dev/sda of=/mnt/sdb1/mein-mbr.dd bs=512 count=1 # In die Datei 'mein-
mbr.dd' sichern
umount /mnt/sdb1 # USB-Stick umounten
```

**sync** Ausführen von gepufferten Schreiboperationen (Evl. letzte Rettung für die Daten, wenn das System nicht mehr sauber heruntergefahren werden kann, das bedeutet, die veränderten Daten sofort auf die Massenspeicher zu schreiben.)

## 6.1 Überblick: Linux Mount-Mechanismus

Die Kommandos *mount* und *umount* leisten die eigentliche Arbeit, wenn Dateisysteme oder Netzwerkreisourcen transparent in den Linux-Verzeichnisbaum einzuhängen sind.

Um aber Partitionen beim Booten automatisch in den Baum einzuhängen (wie z.B. die Wurzelpartition!) oder um festzulegen, dass normale Benutzer eine Ressource bei Bedarf einbinden dürfen, sind zwei Dateien von Belang:

*/etc/fstab* legt fest, welche Ressourcen wie zu mounten sind

*/etc/mstab* listet auf, welche Ressourcen aktuell gemountet sind

### Relevante man-Pages

In erster Linie sind die man-Pages von *mount* und *fstab* von Bedeutung. Weitere Man-Pages sind bei Bedarf für die verwendeten Dateisysteme zu konsultieren (siehe Abschnitt *Beispiele*).

### Die Datei */etc/fstab*

Abkürzung: FSTAB = **F**ile**S**ystem **T**ABLE (Dateisystemtabelle)

Diese einfache Textdatei spielt eine sehr wichtige Rolle, sie ordnet Gerätedateien oder Netzwerkfreigaben leeren Verzeichnissen, den Mountpunkten zu. Nach erfolgreichem Mountvorgang sind diese Mountpunkt-Verzeichnisse nicht mehr leer; in ihnen befindet sich dann der Inhalt z.B. einer Diskette, die mit Hilfe einer Gerätedatei */dev/fd0* (**f**loopy **d**isk**0**) angesprochen wird.

### Aufbau der Datei

Die Datei ist in Tabellenform organisiert:

**Eine Zeile** beschreibt jeweils das Einhängen einer Ressource.

**Sechs Spalten** beschreiben die Details für das Einhängen einer Ressource:

1. Ressource (z.B. */dev/fd0* oder *192.168.23.200:/home/pub*)
2. Mountpunkt (z.B. */media/floppy* oder */media/pub*)
3. Dateisystemtyp (z.B. *auto* für Autoerkennung oder *nfs*)
4. Verschiedene Optionen (z.B. *noauto*, *user* oder für NFS sinnvoll: *bg,soft*)
5. Datensicherung mit *dump* vornehmen? Eine Null steht für nein. Eine Eins für ja.
6. Reihenfolge der Datenkonsistenz-Prüfung beim booten? Eine Null steht für keine Prüfung. Eine Eins für ja, als erstes zu prüfen (Wurzelpartition). Eine Zwei für ja, später prüfen (Nicht-Wurzelpartition).

Findet das System in den Spalten fünf und sechs keine Einträge, gilt als Voreinstellung Null.

## Beispiele

**Wurzel-Partition** bei SuSE, zu den Optionen *acl,user\_xattr* siehe im Überblick vor allem <http://acl.bestbits.at/> und spezielles Manual *man -a attr*:

```
/dev/sda6 / ext3 acl,user_xattr 0 1
```

**Normale Diskette**, zu den Optionen *noauto,user,sync* siehe *man mount*:

```
/dev/fd0 /media/floppy auto noauto,user,sync 0 0
```

Mit «*mount /media/floppy*» kann nun jedermann eine Diskette einhängen und später mit «*umount /media/floppy*» wieder aus dem Dateisystem entfernen. Die Option *sync* soll veranlassen, dass Änderungen sofort auf das Medium geschrieben werden.

**NFS-Freigabe** (Network File System, Unix-Standard), zu den Optionen *soft,bg* siehe spezielles Manual *man nfs*:

```
192.168.23.200:/home/pub /media/pub nfs soft,bg 0 0
```

**Samba-Freigabe** (SMB = Server Message Block Protocol, Windows-Standard), zu den Optionen *username,password* siehe spezielles Manual *man smbfs*:

```
//192.168.23.200/doc /media/doc smbfs username=tux,password=x
```

## Virtuelle Dateisysteme

...benötigen keinen Festplattenplatz. Sie benutzen den Arbeitsspeicher, um die Kommunikation zwischen dem Kernel selbst und dem sogenannten Userspace (Benutzer-Bereich des Kernels = Schnittstelle nach außen) zu ermöglichen.

Aus einer Vielzahl hier eine Auswahl:

- Ein wichtiges virtuelles Dateisystem ist das unter */proc* eingehängene, das für jeden laufenden Prozess einen Verzeichnis-Eintrag führt sowie Hardwareinformationen, z.B. über die CPU, liefert: Siehe auch <http://www.linuxfibel.de/filesys.htm#virt>

```
cat /proc/cpuinfo
```

- Seit Kernelversion 2.6.x gibt es unter */sys* ein weiteres virtuelles Dateisystem. Hier befinden sich vor allem Hardwareinformationen und -einstellungen. Man kann sich z.B. die aktuelle CPU-Frequenz ausgeben lassen: Siehe auch: [http://de.gentoo-wiki.com/AMD\\_64](http://de.gentoo-wiki.com/AMD_64)

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
```

- Für automatische Mountvorgänge (USB-Sticks u.ä.) wird in neueren Distributionen *UDEV* verwendet. Mit Hilfe von Rules (Regeln) können Mountpunkte und Aktionen an persönliche Erfordernisse angepasst werden. Siehe dazu:

<http://de.wikipedia.org/wiki/Udev>

[http://de.wikipedia.org/wiki/Ger%C3%A4tenamen\\_unter\\_Linux](http://de.wikipedia.org/wiki/Ger%C3%A4tenamen_unter_Linux)

<http://wiki.ubuntuusers.de/udev>



## 7 Prozessverwaltung

**ps** Prozesse auflisten, meistens ist die Prozess-ID (PID) von Interesse. Wichtige Optionen: a, u, x, l (long); mit 'x' auch Daemons (Prozesse ohne Verbindung mit einem Terminal) anzeigen.

```
ps aux # Alle Prozesse auflisten
ps l -C ssh # Aus den Prozessen solche mit 'ssh' herausfiltern
```

**pstree** Prozesse in Baumform ausgeben, dh. in Abhängigkeit zum Entstehungskontext mit PID und PPID (Parent Process ID).

```
pstree -p | head # Wer ist der Vater aller Prozesse?
init(1)-+-NetworkManager(4714)-+--{NetworkManager}(4777)
      |                               '-{NetworkManager}(6444)
      |-NetworkManagerD(4721)
      |-acpid(4559)
      |-apache2(5595)-+--apache2(6417)
                    |               |-apache2(6418)
                    |               |-apache2(6419)
                    |               |-apache2(6420)
                    |               '-apache2(6421)
      |-atd(5551)
```

**top** Selbst aktualisierendes Monitor- und Prozessmanipulationsprogramm. Es lässt sich interaktiv bedienen: 'u' lässt nach einem Usernamen sortieren, 'R' die Sortierung umkehren, 'k' einen Prozess killen.

**kill** Prozesse über ihre PID beenden, mit '-9' gewaltsam.

```
kill 6377 # Prozess mit der PID 6377 normal beenden
kill -9 6377 # Prozess mit der PID 6377 gewaltsam beenden
```

**killall** Prozesse über ihren Programmnamen beenden, mit '-9' gewaltsam.

```
ALT + F2 --> xterm # Ein X-Terminal starten
killall xterm # Das eigene Terminal beenden
```

**pgrep, pkill** Prozesse mittels übereinstimmender Teile ihres Programmnamens finden bzw. beenden

```
pkill fire # Den Firefox-Browser beenden und evl. weitere, die 'fire' im
          Namen haben
```

**xkill** Einzelne Fenster 'abschießen': Nach Ausführung dieses Kommandos verändert sich das Mauszeigersymbol; nun zeigt man mit der Maus auf ein grafisches Programm bzw. Desktopelement und kann es mit einfachem Linksklick beenden. Unter KDE gibt es dafür den Shortcut 'STRG + ALT + ESC'. Es droht u.U. Datenverlust! Mit der ESC-Taste kann die Aktion bei Bedarf abgebrochen werden.

**STRG + ALT + BACKSPACE** Gewaltsames Neustarten des X-Servers (Es droht u.U. Datenverlust!). Diese Funktion wurde in neueren Distributionen aus Sicherheitsgründen deaktiviert. Alternativ kann man als root den Displaymanager unter SuSE mit `'/etc/init.d/xdm restart'` neu starten oder unter Ubuntu mit `'/etc/init.d/gdm restart'`. Distributionen wie SuSE, die mit Runleveln arbeiten, können meist mit `'init 3'` die grafische Oberfläche beenden und mit `'init 5'` wieder starten.

**nice** Vorschalt-Programm um ein anderes mit veränderten Prioritäten zu starten. Der Benutzer darf nur Werte in positive Richtung setzen. Niedrigste Priorität liegt bei höchstem Nice-Wert (19) vor, höchste Priorität bei geringstem Nice-Wert (-20).

```
nice -n 19 updatedb -o ~/suchindex.db # Bitte langsamer!
nice -n -20 updatedb # Der Root darf alles: mehr Dampf geben!
```

**renice** Verhält sich ähnlich wie `'nice'`. Unterschiede: Man kann nachträglich die Priorität ändern, außerdem muss man über die PID gehen.

```
sleep 360 & # Testweise einen Prozess starten
ps l --sort nice # Prozesse sortiert nach nice-Wert ausgeben
renice 15 6523 # Nice-Wert fuer entsprechenden Prozess aendern
```

**sleep** Prozess starten, der nichts besseres zu tun hat, als zu schlafen. (Siehe oben bei `'renice'`)

**time** Vorschalt-Programm, das die Zeit, die ein Programm benötigt, misst (Stoppuhr).

**watch** Vorschalt-Kommando, das andere Programme wiederholt (Standard: alle 2s) aufruft

```
watch -d ps l --sort nice # Guenstig zum Testen von 'renice'
```

**nohup** Vorschalt-Kommando, das Programme trotz `'logout'` weiterarbeiten lässt

```
nohup find / -perm +4000 > progs-mit-suid-recht.log &
```

**fuser** Identifizierung von Prozessen, die von geöffneten Dateien bzw. Sockets herrühren

```
fuser -mv /media/stick # Gefahrloses Entnehmen moeglich?
```

**lsuf** 'List Open Files' - Ausgabe aller Dateien, die sich im geöffneten Zustand befinden. Nützlich für Diagnosezwecke zum Thema Sicherheit.

```
lsuf | grep '/home' # Alle auf '/home' geoeffneten Dateien
lsuf -i | grep LISTEN # Als Root: alle lokal laufenden Dienste
```

**cron** Zeitplanerdienst, der wiederholt Aufträge benutzerabhängig abarbeitet.

```
crontab -l # Ausgabe vorhandener Auftraege
crontab -e # Editieren des Zeitplaners: taeglich 2:00 Backup
# Min Std Tag Mon WoTag Kommando
0 2 * * * tar czf /root/etc-backup.tar.gz /etc
```

**at** Zeitgesteuerte Aufträge einmalig ausführen. Beispiel: eine Stunde Radiomitschnitt.

```
at now + 3 minute # In drei Minuten Radioaufnahme starten
warning: commands will be executed using /bin/sh
at> wget http://dradio-live.ogg.t-bn.de/dlf_low.ogg &
at> sleep 3600 && killall wget
at> # Mit STRG + d Programm verlassen
```

## 8 Benutzerverwaltung

**id** Gibt Informationen über einen bestimmten Benutzer aus oder über den aktuell angemeldeten.

**w** Gibt ausführliche Informationen über einen eingeloggten Benutzer und seine Prozesse aus.

**who** Zeigt in kürzerer Form, wer eingeloggt ist.

**finger** Gibt viele Daten über einen Benutzer aus: Loginname, voller Name, letzter Login, ...

**su** 'Switch User' - eine andere Identität annehmen, Prozesse laufen dann in anderem Kontext.

```
/bin/su - # Auf sichere Weise Root werden
/bin/su - tux # Benutzer 'tux' werden
/bin/su -c "shutdown -h 15:00" # Command-Aktion als Root
```

**sudo** SSuper User Do Vorschalt-Kommando, um Benutzern für bestimmte, umfangreichere Aktionen Superuser-Rechte zu verleihen

**last** Zeigt eine Liste der zuletzt eingeloggten Benutzer.

**lastlog** Gibt einen Report über die jüngster Zeit am meisten vollzogenen Anmeldungen am System aus.

**useradd** Benutzer anlegen, mit '-m' (make) im gleichen Atemzug das Heimatverzeichnis erzeugen, mit '-s' eine Shell abweichend von Standardvorgaben, mit '-u' eine bestimmte UID zuweisen

```
useradd -D # Anzeige der Voreinstellungen
useradd -m tux -s /bin/bash # Benutzer 'tux' anlegen
```

**passwd** Passwörter vergeben/ändern sowie Gültigkeiteinstellungen für Benutzeraccounts vornehmen, ACHTUNG: ohne Übergabe eines Parameters ändert der Root sein eigenes!!

```
passwd tux # Fuer Benutzer 'tux' das Passwort setzen
```

**usermod** Einstellungen wie Gruppenzugehörigkeit für Benutzer verändern, mit '-g' ändert man die primäre Gruppe, mit '-G' kann man weitere, nicht primäre Gruppen angeben. ACHTUNG: es müssen immer alle sekundären Gruppen nochmals mit genannt werden.

```
usermod -g vip tux # Benutzer 'tux' wird versetzt
usermod -G audio,video,cdrom,plugdev,powerdev,users tux
```

**userdel** Löschen von Benutzeraccounts. Die Option '-r' (recursive) gestattet das gleichzeitige Löschen des Heimatverzeichnisses.

```
userdel -r tux
```

**groupadd** Eine neue Benutzergruppe anlegen, mit '-g' kann man eine bestimmte GID vorgeben.

```
groupadd -g 2000 vip
```

**vipw** Änderungen von Account-Einstellungen per Hand vornehmen, allerdings nicht direkt, sondern sicher an einer Kopie. Ohne Option aufgerufen, wird die Datei `/etc/passwd` editiert, die grundlegende Einstellungen über Benutzer gestattet, mit `'-s'` kann die Passwortdatei `/etc/shadow` geändert werden, mit `'-g'` die `/etc/group`, die für die Gruppdefinitionen verantwortlich ist.

## 9 Netzwerk-Kommandos

**Netzwerkarte und -treiber** Eine kleine Zeile klärt die vorrangigen Fragen, ob a) z.B. eine Ethernet-Karte gefunden wurde, b) unter welcher genauen Bezeichnung sie ansprechbar ist und c) ein Link vorliegt: `'dmesg | grep eth'`

**ifconfig** Werkzeug zur Kontrolle und zum Setzen der IP-Einstellungen

```
ifconfig                                # Zur Kontrolle
ifconfig eth0 192.168.2.2               # Adresse mit Standardmaske
    setzen
ifconfig eth0 192.168.2.2 netmask 255.255.255.240 # Adresse mit Subnetting
    setzen
```

**ip** Alternativ (anstelle der Verwendung von 'ifconfig' und 'route') mit den 'networking and traffic control tools' des Paketes 'iproute' die IP-Einstellungen vornehmen:

```
ip addr show                            # Ausgabe der Einstellungen
ip addr add 192.168.2.2 dev eth0         # Adresse setzen
ip route add default via 192.168.2.1    # Standardgateway setzen
```

**route** Routingtabelle ausgeben oder manipulieren

```
route -n # Routingtabelle ausgeben, ohne Namen aufzuloesen
route add default gw 192.168.2.1 # Setzen des Standardgateways
```

**DNS konfigurieren** Mit folgender Zeile kann die DNS-Konfiguration schnell mal in die verantwortliche Datei geschrieben werden (z.B. ein DNS-Server der Telekom):

```
echo 'nameserver 194.25.2.129' > /etc/resolv.conf
```

**iptables** Ausgabe und Manipulation von Firewallregeln, falls keine Regeln gesetzt sind, geben die Zeilen

```
iptables -L -n # Standard-Ketten auflisten
iptables -L -n -t nat # NAT-Ketten auflisten
```

jeweils nur 6 Textzeilen aus, es ist dabei je drei mal 'policy ACCEPT' zu lesen.

**smbclient** Tests und Verbindungsaufbau zu einem Samba-Server (Windows-NT-Server-Ersatz):

```
smbclient -L localhost -U% # Ein Lookup als anonymer Benutzer (-U%) nach
    eigenen Freigaben
smbclient \\\server-007\share-01 -U max # Verbindungsaufbau als max zu
    fremden Server
```

**rpcinfo** Test der Erreichbarkeit des RPC-Dienstes (Remote Procedure Calls, Portmapper), der als Voraussetzung fürs Mounten von NFS-Freigaben und auch für den NFS-Server selbst Voraussetzung ist:

```
rpcinfo -p 192.168.2.2
```

**showmount** Test der Erreichbarkeit und Ausgabe von Exporten eines NFS-Servers (Network File System = Unix-Freigabe)

```
showmount -e 192.168.2.2
```

**Dienste starten/restarten** Läuft irgendetwas nicht mit einem Dienst, kann es sein, dass er lediglich seine Konfigurationsdatei neu einlesen muss (mit '/etc/init.d/DIENSTNAME reload') oder vielleicht doch richtig neu zu starten ist (mit '/etc/init.d/DIENSTNAME restart'). Achtung: Diese Aktion ist nur temporärer Natur, soll der Dienst während des Bootvorganges automatisch gestartet werden, muss das in Konfigurationsdateien oder mit Hilfe von Runlevel-Links eingerichtet werden.

```
/etc/init.d/nmb restart    # Unter SuSE den nmbd restarten  
/etc/init.d/smb restart    # Unter SuSE den smbd restarten  
/etc/init.d/samba restart  # Unter Debian/Ubuntu nmbd UND smbd restarten
```