

# Zusammenfassung UNIX/LINUX-Einführungskurs -Aufbaukurs und -Shell-Programmierung

Version 1.22 — 19.2.2016

© 2005–2016 T. Birnthal, OSTC GmbH

Die Informationen in diesem Skript wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Der Autor übernimmt keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte vorbehalten einschließlich Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Für Dokumente und Programme unter dem Copyright der OSTC GmbH gilt:

- Dürfen heruntergeladen und im privaten Bereich frei verwendet werden.
- Kommerzielle Nutzung bedarf der vorherigen Zustimmung durch die OSTC GmbH.
- Titelseite und Copyright-Hinweise darin dürfen nicht verändert werden.

Hinweise auf inhaltliche Fehler, Schreibfehler und unklare Formulierungen sowie Ergänzungen, Kommentare, Wünsche und Fragen können Sie gerne an den Autor richten:

OSTC Open Source Training and Consulting GmbH  
Thomas Birnthal  
E-Mail: [tb@ostc.de](mailto:tb@ostc.de)  
Web: [www.ostc.de](http://www.ostc.de)

## Inhaltsverzeichnis

<b>1</b>	<b>Schreibweise in dieser Zusammenfassung</b>	<b>6</b>
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Datei-Inhalt ansehen . . . . .	7
2.2	Hilfestellung abrufen . . . . .	7
2.2.1	man-Abschnitte (Sections) . . . . .	7
2.2.2	man-Seitenaufbau . . . . .	8
2.3	Wichtige Steuertasten . . . . .	8
2.4	Wichtige LINUX-spezifische Steuertasten . . . . .	8
2.5	Wichtige KDE-spezifische Steuertasten . . . . .	9
2.6	Maus-Bedienung unter X Window . . . . .	9
2.7	Die wichtigsten UNIX-Kommandos . . . . .	9
<b>3</b>	<b>Benutzer- und Gruppenverwaltung</b>	<b>10</b>
3.1	Kommandos zur Benutzerverwaltung . . . . .	10
3.2	Kommandos zur Gruppenverwaltung . . . . .	10
<b>4</b>	<b>Dateisystem</b>	<b>11</b>
4.1	Auskunftsfunktionen . . . . .	11
4.2	Dateitypen (Kommando ls -l) . . . . .	11
4.3	Verändernde Kommandos . . . . .	12
4.4	Besitzverhältnis und Zugriffsrechte . . . . .	12
4.4.1	Kommandos . . . . .	12
4.4.2	Zugriffsrechte für Dateien . . . . .	12
4.4.3	Zugriffsrechte für Verzeichnisse . . . . .	13
4.4.4	Dateirechte in oktaler + symbolischer Form . . . . .	13
4.4.5	LINUX-Sonderrechte . . . . .	13
4.5	Dateien und Verzeichnisse . . . . .	14
4.5.1	Standard-Verzeichnisnamen . . . . .	14
4.5.2	UNIX Standard-Verzeichnisse . . . . .	14
4.5.3	LINUX-spezifische Standard-Verzeichnisse . . . . .	15
4.5.4	Die wichtigsten Gerätedateien von LINUX . . . . .	15
4.5.5	Wichtige zentrale Konfigurationsdateien . . . . .	16
4.5.6	Wichtige zentrale Netzwerk-Konfigurationsdateien . . . . .	16
4.5.7	Wichtige lokale Konfigurationsdateien/Verzeichnisse . . . . .	16
<b>5</b>	<b>Prozeßverwaltung</b>	<b>17</b>
5.1	Kommandos zur Prozeßverwaltung . . . . .	17
5.2	Kommandos zur Hintergrund-Prozeß-Verwaltung . . . . .	17
5.3	Die wichtigsten Signale . . . . .	18
<b>6</b>	<b>Shell</b>	<b>18</b>
6.1	Shell-Typen . . . . .	18
6.2	Shell-Konfigurationsdateien . . . . .	18
6.2.1	Konfigurationsdateien der Bourne-Shell . . . . .	19
6.2.2	Konfigurationsdateien der Bash . . . . .	19

6.2.3	Konfigurationsdateien der C-Shell	19
6.2.4	Konfigurationsdateien der TC-Shell	19
6.2.5	Konfigurationsdateien der Korn-Shell	19
6.3	Shell-Operationen	20
6.4	Pfadsuche (bei Aufruf von CMD)	20
6.5	Shell- und Umgebungs-Variablen	20
6.5.1	Kommandos für Shell-Variablen	20
6.5.2	Kommandos für Umgebungs-Variablen	21
6.5.3	Variablen-Kommandos unter der (T)C-Shell	21
6.5.4	Einige Standard-Variablen	21
6.5.5	Eingabeprompt-Definition (bash)	21
6.5.6	Eingabeprompt-Definition (tcsh)	22
6.6	Kommando-Wiederholung (bash + ksh)	22
6.7	Dateinamen-Vervollständigung (bash + ksh)	23
6.8	Aliase	23
6.8.1	Nützliche Aliase	23
6.9	Funktionen	24
6.10	Ein/Ausgabe-Umlenkung	24
6.11	Spezielle Ein-/Ausgabeumlenkung	24
6.12	Here-Dokument	24
6.13	Dateinamen-Expansion	25
6.14	Kommando-Substitution	25
6.15	Shell-Quotierung	25
6.16	Kommando-Listen	25
<b>7</b>	<b>Sonstige Kommandos</b>	<b>26</b>
7.1	Wichtige Format-Angaben zu date	26
<b>8</b>	<b>Vi</b>	<b>27</b>
8.1	Die wichtigsten Vi-Befehle	27
8.2	Weitere wichtige Vi-Befehle	28
8.3	Wichtige allgemeine Vi-Optionen	28
8.4	Wichtige Vi-Optionen für Programmierer	28
8.5	Wichtige Vim-Optionen	29
8.6	Nützliche Vi-Makros	29
<b>9</b>	<b>Drucken</b>	<b>30</b>
9.1	BSD-Variante der Druck-Kommandos	30
9.2	System V-Variante der Druck-Kommandos	30
9.3	lpc-Kommandos	30
<b>10</b>	<b>Reguläre Ausdrücke</b>	<b>31</b>
10.1	Standard-Metazeichen	31
10.2	Erweiterte Metazeichen	31
10.3	Metazeichen im Ersetzungsmuster	31
10.4	Escape-Sequenzen	31
10.5	perl-Metazeichen	32

<b>11 UNIX-Werkzeuge</b>	<b>32</b>
11.1 strings, wc, head, tail, tee	32
11.2 cmp, diff	32
11.3 cut, paste, join, tr, split	33
11.4 sort, uniq	33
11.5 grep	33
11.5.1 Die wichtigsten Optionen von grep	34
11.6 ed, ex, sed, awk, perl	34
11.6.1 Sed-Kommandos	35
11.7 find, locate	35
11.7.1 find-Bedingungen	36
11.7.2 find-Aktionen	36
11.7.3 find-Beispiele	36
11.8 tar, cpio	37
11.9 compress, gzip	37
<b>12 System-Administration</b>	<b>38</b>
12.1 fdisk, fdformat, mkfs, fsck, dd	38
12.2 mount, umount	38
12.2.1 Mount-Optionen	39
12.2.2 Samba-Mount-Optionen	39
12.3 at, crontab	39
12.3.1 at-Befehle	39
12.3.2 crontab-Befehle	39
12.3.2.1 crontab-Felder	39
12.3.2.2 Erlaubte crontab-Feldwerte	40
12.4 mail	40
12.4.1 Die wichtigsten Mail-Befehle	40
<b>13 Shell-Programmierung</b>	<b>41</b>
13.1 Shell-Aufrufarten	41
13.2 Shell-Optionen	41
13.3 Spezielle Shell-Variablen (Parameter)	41
13.3.1 Bedingte Auswertung von Shell-Variablen	42
13.4 Bedingungen des test-Kommandos	42
13.4.1 Textvergleiche	42
13.4.2 Dateiattribut-Vergleiche	42
13.4.3 Datei-Vergleiche	43
13.4.4 Numerische Vergleiche	43
13.4.5 Logische Verknüpfungen	43
13.4.6 Shell-Optionen testen	43
13.5 Kontrollstrukturen	43
13.5.1 Benutzereingabe einlesen	43
13.5.2 Verzweigung	44
13.5.3 case-Mehrfachverzweigung (Text-Vergleich mit Shell-Metazeichen)	44
13.5.4 for-Schleife (Liste bzw. Shell-Aufrufargumente abarbeiten)	44
13.5.5 while und until-Schleife (Bedingung prüfen)	44

13.5.6 Funktions-Definition und -Aufruf . . . . .	45
<b>14 Awk . . . . .</b>	<b>45</b>
14.1 AWK-Optionen . . . . .	45
14.2 GNU-AWK-spezifische Optionen . . . . .	45
14.3 Konstanten . . . . .	46
14.3.1 Zahlen . . . . .	46
14.3.2 Zeichenketten . . . . .	46
14.3.3 Escape-Sequenzen . . . . .	47
14.3.4 Reguläre Ausdrücke . . . . .	47
14.3.5 Reguläre Ausdrücke . . . . .	47
14.3.5.1 Metazeichen in Regulären Ausdrücken . . . . .	47
14.3.5.2 POSIX-Zeichenklassen ([:class:]) . . . . .	48
14.3.6 Operatoren . . . . .	49
14.3.7 Ausdrücke . . . . .	49
14.4 Programmaufbau . . . . .	49
14.4.1 Regel . . . . .	49
14.4.1.1 Muster . . . . .	49
14.4.1.2 Aktionen . . . . .	50
14.4.2 Funktionen . . . . .	50
14.4.2.1 Definition . . . . .	50
14.4.2.2 Aufruf . . . . .	50
14.4.3 Zeilenumbrücken sind erlaubt nach . . . . .	51
14.4.4 Felder, Variablen und Arrays . . . . .	51
14.4.4.1 Defaultwerte für Feld- und Satztrenner . . . . .	51
14.4.4.2 Felder . . . . .	51
14.4.4.3 Variablen . . . . .	51
14.4.4.4 Arrays . . . . .	51
14.4.4.5 Vergleichstypen . . . . .	52
14.4.5 Vordefinierte Variablen und Arrays . . . . .	52
14.4.6 Vordefinierte Funktionen . . . . .	53
14.4.6.1 Arithmetik-Funktionen . . . . .	53
14.4.6.2 Bit-Funktionen . . . . .	54
14.4.6.3 Zeichenketten-Funktionen . . . . .	54
14.4.6.4 Array-Funktionen . . . . .	54
14.4.6.5 Ein/Ausgabe-Funktionen . . . . .	55
14.4.6.6 getline-Rückgabewerte . . . . .	55
14.4.6.7 printf-Formatumwandlungen . . . . .	56
14.4.6.8 printf-Zusatzangaben zu Formatumwandlungen . . . . .	56
14.4.6.9 Zeit-Funktionen . . . . .	56
14.4.6.10strftime-Formatangaben . . . . .	57
14.4.6.11Internationalisierung-Funktionen . . . . .	57
14.4.6.12Typ-Funktionen . . . . .	57
14.5 GAWK-Spezialdateien . . . . .	58

## 1 Schreibweise in dieser Zusammenfassung

Folgende groß geschriebenen, englischen Namen werden in den Befehlen als Platzhalter verwendet. Sie sind jeweils durch passende echte Namen zu ersetzen (ebenso wie die anderen GROSS geschriebenen Namen in Befehlen).

ARCH	Archiv-Datei [ <b>archive</b> ]
CMD	Kommando-Name [ <b>command</b> ]
DEST	Ziel-Datei/Verzeichnis [ <b>destination</b> ]
DEV	Gerät [ <b>device</b> ]
DIR	Verzeichnis [ <b>directory</b> ]
EXPR	Ausdruck (numerischer oder logischer) [ <b>expression</b> ]
FILE	Datei-Name
GID	Gruppen-Nummer [ <b>group identity</b> ]
GROUP	Gruppen-Name [ <b>group</b> ]
HOST	Rechner-Name
ID	Nummer [ <b>identity</b> ]
JOBNR	Hintergrundprozeß/Druckauftrag-Nummer
MOVE	Bewegungs-Kommando im Vi
NN	Zahl
NUM	Zahl [ <b>number</b> ]
OPT	Options-Name
PATH	Pfad zu einer Datei
PATTERN	Muster
PID	Prozeß-ID
REGEX	Regulärer Ausdruck [ <b>regular expression</b> ]
SCRIPT	Skript-Datei (Awk, Sed, Shell, ...)
SRC	Quell-Datei/Verzeichnis [ <b>source</b> ]
TEXT	Beliebiger Text ( <i>meist in Anführungszeichen zu setzen!</i> )
TYPE	(Daten)Typ
UID	Benutzer-Nummer [ <b>user identity</b> ]
USER	Benutzer-Name
VALUE	Beliebiger Wert
VAR	Variablen-Name

Optionale Angaben werden in eckige Klammern [...] eingeschlossen.

<...>	Taste
Strg-<...>	Steuerung/Control-Taste + Taste gemeinsam drücken [ <b>control</b> ]
CR RETURN	Return-Taste [ <b>carriage return</b> ]
ESC	Escape-Taste
NL / LF	Return-Taste [ <b>newline, linefeed</b> ]
TAB	Tabulator-Taste
DEL	Entfernen-Taste [ <b>delete</b> ]

## 2 Grundlagen

### 2.1 Datei-Inhalt ansehen

<code>more FILE</code>	Datei <code>FILE</code> seitenweise anzeigen [ <b>mehr</b> ]
<code>less FILE</code>	Datei <code>FILE</code> seitenweise anzeigen ( <i>verbesserte Version!</i> ) [ <b>weniger</b> ]

<code>&lt;SPACE&gt;</code>	Um eine Seite weiterblättern (Leertaste)
<code>b</code>	Zurückblättern ( <i>nur less!</i> ) [ <b>back</b> ]
<code>&lt;RETURN&gt;</code>	Um eine Zeile weiterblättern
<code>/TEXT &lt;RETURN&gt;</code>	<code>TEXT</code> suchen
<code>n</code>	Letzte Suche wiederholen [ <b>next</b> ]
<code>q</code>	<code>more/less</code> verlassen [ <b>quit</b> ]
<code>h</code>	Hilfe anzeigen [ <b>help</b> ]

### 2.2 Hilfestellung abrufen

<code>man [[-s] NN] CMD</code> <code>man -f CMD</code> <code>man -k TEXT</code> <code>man -t CMD   lpr</code> <code>whatis CMD</code> <code>apropos TEXT</code>	Beschreibung zu <code>CMD</code> aus <code>man</code> -Abschnitt <code>NN</code> [ <b>section</b> ] Einzeilige <code>man</code> -Beschreibungen zu <code>CMD</code> ausgeben [ <b>find</b> ] Einzeilige <code>man</code> -Beschreibungen mit <code>TEXT</code> darin ausgeben [ <b>key</b> ] <code>CMD</code> -Beschreibung mit <code>troff</code> setzen und ausdrucken [ <b>troff</b> ] Analog <code>man -f CMD</code> Analog <code>man -k TEXT</code>
<code>CMD --help</code> <code>CMD -?</code> <code>CMD -h</code> <code>CMD -.</code>	Usage-Meldung zu <code>CMD</code> (GNU-Programme) Usage-Meldung zu <code>CMD</code> (einige Programme) Usage-Meldung zu <code>CMD</code> (einige Programme) [ <b>help</b> ] Usage-Meldung zu <code>CMD</code> (einige Programme)
<code>help</code> <code>help CMD</code> <code>hilfe</code> <code>info CMD</code>	Alle eingebauten Shell-Kommandos ausgeben ( <code>bash</code> ) Usage-Meldung zu eingebautem Shell-CMD ausgeben Browsergestütztes SuSE-Hilfesystem aufrufen GNU-Beschreibung zu <code>CMD</code> anzeigen (interaktiv, Hypertext)

#### 2.2.1 man-Abschnitte (Sections)

1	Ausführbare Programme oder Shellbefehle
2	(C-)Systemaufrufe (Kernelfunktionen)
3	(C-)Bibliotheksaufrufe (Funktionen in System-Bibliotheken <code>/usr/lib</code> )
4	Gerätedateien (gewöhnlich in <code>/dev</code> )
5	Formate und Bedeutung von Konfigurations-/Logdateien (z.B. <code>/etc/passwd</code> )
6	Spiele
7	Makropakete und Konventionen (z.B. <code>man(7)</code> , <code>groff(7)</code> )
8	Systemadministrationsbefehle (in der Regel nur für <code>root</code> )
9	Kernelroutinen (kein Standard)
1	Lokale Erweiterungen

## 2.2.2 man-Seitenaufbau

NAME	Name
SYNOPSIS	Syntax-Zusammenfassung
DESCRIPTION	Beschreibung
OPTIONS	Optionen
OVERVIEW	Übersicht
DEFAULTS	Normaleinstellungen
ENVIRONMENT	Umgebung (Variablen)
FILES	(Konfigurations)Dateien
EXAMPLES	Beispiele
NOTES	Bemerkungen
SEE ALSO	Siehe auch ( <i>Verweise auf verwandte Kommandos</i> )
REFERENCES	Verweise auf verwandte Kommandos
BUGS	Fehler
AUTHOR	Autor
HISTORY	Entwicklungs-Geschichte

## 2.3 Wichtige Steuertasten

<RETURN>	Eingegebenes Kommando ausführen
<BACKSPACE>	Letztes eingegebenes Zeichen entfernen
<Strg-C>	Laufendes Kommando abbrechen [ <b>break</b> ]
<Strg-D>	Abmelden/Dateiende anzeigen [ <b>end of file</b> ]
<Strg-Z>	Aktuellen Prozeß stoppen + in Hintergrund stellen
<Strg-S>	Terminal-Scrolling anhalten [ <b>stop</b> ]
<Strg-Q>	Terminal-Scrolling laufen lassen [ <b>quit</b> ]
<Strg-U>	Ganze Eingabezeile löschen [ <b>undo</b> ]
<Strg-W>	Letztes Wort auf der Eingabezeile löschen [ <b>word</b> ]
<Strg-H>	Backspace
<Strg-I>	Tabulator [ <b>indent</b> ]
<Strg-V>	Nächstes Steuer-Zeichen schützen (Escape) [ <b>verbose</b> ]
<Strg-G>	Klingel
<Strl-L>	Bildschirm neu aufbauen [ <b>load/leeren</b> ]

## 2.4 Wichtige LINUX-spezifische Steuertasten

Cursor-Auf/Ab	In alten Kommandos blättern
Cursor-Links/Rechts	In aktuellem Kommando bewegen
<DEL>/<BACKSPACE>	In aktuellem Kommando löschen
Shift-Cursor-Auf/Ab	Zeilenweise in Kommandoausgabe blättern
Shift-Bild-Auf/Ab	Seitenweise in Kommandoausgabe blättern
Alt-F1...F6	Auf Textterminal 1..6 umschalten
Strg-Alt-F1...F6	Auf Textterminal 1..6 umschalten (in Grafikoberfläche)
Alt-Cursor-Links/Rechts	Auf vorheriges/nächstes Textterminal umschalten
Strg-Alt-F7	Auf Grafikoberfläche umschalten
Strg-Alt-BACKSPACE	Grafikoberfläche beenden
Strg-Alt-+/-	Bildschirmauflösung verändern (+/- auf <i>Zahlentastatur</i> !)



## 2.5 Wichtige KDE-spezifische Steuertasten

Alt-TAB	Zum nächsten Fenster springen
Shift-Alt-TAB	Zum vorherigen Fenster springen
Strg-F1..F8	Zur Arbeitsfläche Nummer 1..8 springen
Strg-TAB	Zur nächsten Arbeitsfläche springen
Shift-Strg-TAB	Zur vorherigen Arbeitsfläche springen
Alt-F1	Startmenü des aktuellen Fensters öffnen
Alt-F2	Fenster für Befehl ausführen öffnen
Alt-F3	Systemmenü des aktuellen Fensters öffnen
Alt-F4	Aktuelles Fenster schließen
Strg-Alt-ESC	Fenster abschließen ( <i>mit Totenkopf-Maus anklicken, ESC=Abbruch!</i> )
Strg-ESC	Systemüberwachungs-Programm starten (Prozeßliste)
Strg-F4	Alle Fenster minimieren

## 2.6 Maus-Bedienung unter X Window

Linke Taste	Text markieren durch Halten+Ziehen
Linke Taste 2x	Doppelklick markiert Wort, Halten+Ziehen markiert wortweise
Mittlere Taste	Markierten Text an Cursorposition einfügen (auch Linke+Rechte Taste)
Rechte Taste	Kontextmenü öffnen

## 2.7 Die wichtigsten UNIX-Kommandos

man CMD cd ls ls -l * exit	Beschreibung zu CMD aus man-Abschnitt NN [section] Verzeichnis wechseln [change directory] Verzeichnis-Inhalt anzeigen [list] Verzeichnis-Inhalt ausführlich anzeigen [long] Steht für alle Dateien in einem Verzeichnis Abmelden bzw. Terminal-Fenster schließen
clear date [+FORMAT] echo "TEXT" echo \$VAR	Bildschirminhalt des Terminals löschen Datum + Uhrzeit ausgeben (gemäß FORMAT, z.B. +%T) TEXT auf dem Terminal ausgeben Inhalt der Variablen VAR anzeigen
more FILE CMD   more grep "TEXT" FILE CMD   grep "TEXT" wc -l FILE CMD   wc -l lp(r) FILE CMD   lp(r)	Datei FILE seitenweise anzeigen [mehr] Ausgabe von CMD seitenweise anzeigen Nur Zeilen mit TEXT ausgeben [global regular expr print] Ausgabe von CMD seitenweise anzeigen Zeilen in Datei FILE zählen [word count/lines] Zeilen in Ausgabe von CMD zählen Datei FILE ausdrucken [line print] Ausgabe von CMD ausdrucken

### 3 Benutzer- und Gruppenverwaltung

#### 3.1 Kommandos zur Benutzerverwaltung

who	Alle angemeldeten Benutzer anzeigen
whoami	Eigene Kennung anzeigen
who am i	Eigene Kennung anzeigen (ausführlicher)
finger USER	Weitere Informationen über Benutzer beschaffen
su	In Kennung <code>root</code> wechseln [ <b>substitute user</b> ]
su -	In Kennung + Umgebung von <code>root</code> wechseln
su USER	In Kennung <code>USER</code> wechseln (alte Umgebung)
su - USER	In Kennung + Umgebung von <code>USER</code> wechseln
exit	Rücksprung zur vorherigen Kennung bzw.
exit	Abmelden/Fenster schließen
<Strg-D>	Abmelden/Fenster schließen
passwd	Paßwort ändern [ <b>password</b> ]
yppasswd	NIS-Paßwort ändern [ <b>yellow pages</b> ]
chsh	Login-Shell ändern (LINUX) [ <b>change shell</b> ]
chfn	Benutzer-Beschreibung ändern (LINUX) [ <b>change full name</b> ]
ypcat passwd	NIS-Benutzerdatenbank ausgeben [ <b>yellow pages catalog</b> ]
ypcat group	NIS-Gruppendatenbank ausgeben
ypcat hosts	NIS-Rechnerdatenbank ausgeben

#### 3.2 Kommandos zur Gruppenverwaltung

groups	Eigene Gruppenkennungen ausgeben
id [-a]	Benutzerkennung + alle Gruppenk. mit IDs anzeigen [ <b>identity, all</b> ]
newgrp GROUP	In Gruppe <code>GROUP</code> wechseln ( <i>Paßwort notwendig!</i> ) [ <b>new group</b> ] (legt Standardgruppe für Datei/Verzeichnis-Neuanlage fest)

## 4 Dateisystem

### 4.1 Auskunftsfunktionen

cat FILE1 FILE2...	Dateien aneinanderhängen + auf stdout ausgeben [ <b>concatenate</b> ]
cd PATH	In Verzeichnis PATH wechseln [ <b>change directory</b> ]
cd	In eigenes Heimat-Verz. wechseln (\$HOME, z.B. /home/USER)
cd /	In Root-Verzeichnis wechseln
cd ..	In Eltern-Verzeichnis des aktuellen Verz. wechseln
cd -	In vorheriges akt. Verz. wechseln (hin- und her, bash und ksh)
cd ~	In eigenes Heimat-Verz. wechseln
cd ~USER	In Heimat-Verz. von USER wechseln
df	Größe + Belegungsgrad aller Partitionen anzeigen [ <b>disk free</b> ]
df -k DIR	Größe + Belegungsgrad der DIR-Partition anzeigen [ <b>kilobyte</b> ]
du	Platzbedarf Dateibaum ab aktuellem Verz. anzeigen [ <b>disk used</b> ]
du DIR	Platzbedarf Dateibaum ab DIR (inkl. Unterverz.) anzeigen
du -s DIR	Platzbedarf Dateibaum ab DIR (nur Summe) anzeigen [ <b>sum</b> ]
du -k du -h	In Kilobyte bzw. für Menschen lesbar [ <b>kilobyte, human readable</b> ]
file FILE	Typ einer Datei anhand Datei-Inhalt ermitteln (/etc/magic)
less FILE	Datei seitenweise anzeigen (auch more, pg, page) [ <b>mehr</b> ]
ls	Dateinamen des akt. Verz. alphabetisch sortiert auflisten [ <b>list</b> ]
ls DIR	Dateinamen des Verzeichnisses DIR auflisten
ls -l	Dateinamen + Dateiattribut ... [ <b>long</b> ]
ls -d	Verzeichnis selbst, nicht seinen Inhalt ... [ <b>directory</b> ]
ls -l -d	Beide Optionen gleichzeitig (auch ls -ld)
ls -a	Auch versteckte Dateien (. . . .xxx) anzeigen [ <b>all</b> ]
ls -i	Inode-Nummer der Dateien mit ausgeben [ <b>inode</b> ]
ls -r	Sortierrichtung absteigend (Std: aufsteigend) [ <b>reverse</b> ]
ls -t	Dateien nach Änderungsdatum sortieren [ <b>time</b> ]
ls -F	Dateityp an Namen anhängen (/ = Verz, @ = Link, * = Exec, ...)
ls -R	Inhalt von Unterverzeichnissen mit auflisten [ <b>recursive</b> ]
ll	Häufig eine Abkürzung (Alias) für ls -l [ <b>list long</b> ]
more FILE	Datei seitenweise anzeigen (auch less, pg, page) [ <b>mehr</b> ]
pwd	Aktuelles Verzeichnis ausgeben [ <b>print working directory</b> ]

### 4.2 Dateitypen (Kommando ls -l)

-	Normale Datei
d	Verzeichnis [ <b>directory</b> ]
l	Symbolischer Link [ <b>link</b> ]
c	Zeichenorientiertes Gerät [ <b>character device</b> ]
b	Blockorientiertes Gerät [ <b>block device</b> ]
p	Named Pipe [ <b>pipe</b> ]
s	Socket [ <b>socket</b> ]

### 4.3 Verändernde Kommandos

cp SRC DEST	Datei SRC in Datei DEST kopieren ( <i>überschreiben!</i> ) [ <b>copy</b> ]
cp SRC... DIR	Dateien SRC... in Verzeichnis DIR kopieren
cp -f SRC DEST	Überschreiben von Dateien erzwingen [ <b>force</b> ]
cp -i SRC DEST	Rückfrage falls Dateien überschrieben werden [ <b>interactive</b> ]
cp -r SRC DEST	Dateibaum von SRC nach DEST kopieren [ <b>recursive</b> ]
ln SRC DEST	(Harten) Link namens DEST auf Datei SRC erzeugen [ <b>link</b> ]
ln -s SRC DEST	Symbol. Link namens DEST auf Datei SRC erzeugen [ <b>symbolic</b> ]
mkdir DIR	Verzeichnis anlegen [ <b>make directory</b> ]
mkdir -p PATH	Kompletten Verzeichnispfad anlegen [ <b>path</b> ]
mv SRC DEST	Datei SRC in DEST umbenennen [ <b>move</b> ]
mv SRC... DIR	Dateien SRC... in Verzeichnis DIR verschieben [ <b>move</b> ]
rm FILE	Datei löschen [ <b>remove</b> ]
rm -f FILE	Löschen von Dateien erzwingen [ <b>force</b> ]
rm -i FILE	Rückfrage pro zu löschende Datei [ <b>interactive</b> ]
rm -r DIR	Dateibaum komplett löschen [ <b>recursive</b> ]
rmdir DIR	Verzeichnis entfernen ( <i>muss leer sein!</i> ) [ <b>remove directory</b> ]
rmdir -p PATH	Kompletten Verzeichnispfad entfernen ( <i>muss leer sein!</i> ) [ <b>path</b> ]
touch FILE	Änd.Datum aktual. bzw. neue (leere) Datei anlegen [ <b>berühren</b> ]

### 4.4 Besitzverhältnis und Zugriffsrechte

#### 4.4.1 Kommandos

chgrp GROUP FILE	FILE der Gruppe GROUP zuordnen ( <i>Mitglied!</i> ) [ <b>change group</b> ]
chown USER FILE	FILE dem Benutzer USER zuordnen ( <i>nur root!</i> ) [ <b>change owner</b> ]
chown USER.GROUP	Benutzer + Gruppe gleichzeitig ändern ( <i>nur root!</i> )
chmod MODE FILE	Dateirechte ändern [ <b>change mode</b> ]
	MODE = [ugoa] [+ -=] [rwxst] oder (0) 000 - (7) 777 (s.u.)
chmod u+rwx FILE	Besitzer erhält alle Rechte an FILE [ <b>user</b> ]
chmod g-w FILE	Besitzergruppe wird Schreibrecht an FILE entzogen [ <b>group</b> ]
chmod o-rwx FILE	Andere haben keine Rechte mehr an FILE [ <b>others</b> ]
chmod a+r FILE	Alle haben Leserecht für Datei FILE [ <b>all</b> ]
chmod 644 FILE	Rechte rw-r--r-- für Datei FILE setzen
chmod -R ...	Rechte für ganzen Verzeichnisbaum ändern [ <b>recursive</b> ]
umask [NNN]	Standardrechte NNN für Datei/Verz-Neuanlage anzeigen/setzen, definiert die zu entfernenden(!) Rechte [ <b>usage mask</b> ]
umask	Gibt meist 022 aus (d.h. Schreibrecht für g+o entfernen)
umask 027	Schreibrecht für Gruppe + alle Rechte für Others entfernen

#### 4.4.2 Zugriffsrechte für Dateien

r	[ <b>read</b> ]	Datei-Inhalt lesbar
w	[ <b>write</b> ]	Datei-Inhalt veränderbar
x	[ <b>execute</b> ]	Datei als Programm ausführbar
s	[ <b>set user id</b> ]	Programmausführung unter der Besitzer-Kennung
s	[ <b>set group id</b> ]	Programmausführung unter der Besitzergruppen-Kennung
t	[ <b>sticky</b> ]	Programm nach dem Start im Speicher/Swap halten ( <i>veraltet!</i> )

#### 4.4.3 Zugriffsrechte für Verzeichnisse

r	read	Verzeichnis lesbar ( <code>ls</code> )
w	write	Verzeichnis veränderbar (Dateien anlegen/löschen/umbenennen verschieben, z.B. <code>cp</code> , <code>ln</code> , <code>mkdir</code> , <code>mv</code> , <code>rm</code> , <code>rmdir</code> , <code>touch</code> )
x	execute	Wechsel in Verzeichnis erlaubt ( <code>cd</code> )
s	set user id	(keine Funktion)
s	set group id	Neue Dateien/Verz. automatisch der gleichen Gruppe zuordnen, neue Verzeichnisse haben wieder set group id Recht gesetzt
t	sticky	Nur Verzeichnis-Eigentümer (oder <code>root</code> ) darf Dateien löschen

#### 4.4.4 Dateirechte in oktaler + symbolischer Form

0	---	
1	--x	1
2	-w-	2
3	-wx	2 + 1
4	r--	4
5	r-x	4 + 1
6	rw-	4 + 2
7	rwX	4 + 2 + 1

000	a-rwx	Keine Rechte
700	u+rwx,go-rwx	Nur Benutzer-Rechte
070	g+rwx,uo-rwx	Nur Gruppen-Rechte
007	o+rwx,ug-rwx	Nur Rechte für alle anderen
777	a+rwx	Rechte für alle
1000	u+s	User-Set-ID-Recht
2000	g+s	Group-Set-ID-Recht
4000	o+t	Sticky Recht

#### 4.4.5 LINUX-Sonderrechte

<code>lsattr FILE</code>	Sonderrechte auflisten [ <b>list attributes</b> ]
<code>chattr [+/-]OPT FILE</code>	Sonderrechte setzen/löschen [ <b>change attributes</b> ]

a	Nur Anfügen an Datei erlaubt [ <b>append</b> ]
A	Zugriffszeit nicht aufzeichnen ( <i>Zeitersparnis!</i> ) [ <b>access date</b> ]
c	Datei komprimieren ( <i>in ext2 nicht implementiert!</i> ) [ <b>compress</b> ]
d	Datei nicht sichern [ <b>dump</b> ]
i	Datei nicht veränderbar (Löschen/Umbenennen/Schreiben/Link) [ <b>immutable</b> ]
s	Datei sicher löschen (mit 0-Bytes überschreiben) [ <b>secure</b> ]
S	Änderungen synchron schreiben (ohne Pufferung) [ <b>synchron</b> ]
u	Datei wiederherstellbar nach Löschen ( <i>in ext2 nicht implementiert!</i> ) [ <b>undelete</b> ]

## 4.5 Dateien und Verzeichnisse

### 4.5.1 Standard-Verzeichnisnamen

/	Root-Verzeichnis
.	Aktuelles Verzeichnis
..	Eltern-Verzeichnis
/home	Standort der Heimat-Verzeichnisse aller Benutzer (auch /usr/home oder /export/home)
~	Eigenes Heimat-Verzeichnis (\$HOME, nur tcsh, bash, ksh)
~USER	Heimat-Verzeichnis von Benutzer USER (nur tcsh, bash, ksh)

### 4.5.2 UNIX Standard-Verzeichnisse

/bin	*	Grundlegende Befehle, zum Booten nötig [ <b>binary</b> ]
/dev	*	Geräte Dateien [ <b>device</b> ]
/etc	*	(Fast alle) Konfigurationsdateien [ <b>et cetera</b> ]
/home		Heimat-Verzeichnisse aller Benutzer
/lib	*	Grundlegende Systembibliotheken [ <b>library</b> ]
/lost+found		Für die von fsck [ <b>file system check</b> ] geretteten Dateien
/opt		Zusätzliche (kommerzielle) Software [ <b>optional</b> ]
/sbin	*	Befehle für Systemadministrator, zum Booten nötig [ <b>system binary</b> ]
/tmp		Temporäre Dateien aller Benutzer [ <b>temporary</b> ] (für jeden schreibbar, z.B. Editor-Zwischendateien)
/usr		Zweite UNIX-Dateihierarchie [ <b>unix system resources</b> ]
/usr/X(11R6)		X Window (Version 11, Release 6)
/usr/bin		Weitere Befehle
/usr/doc		Dokumentationen (veraltet)
/usr/etc		Konfigurationsdateien
/usr/local		Lokale Installationen (z.B. bin, lib, man)
/usr/man		Manual-Seiten (veraltet)
/usr/sbin		Weitere Systembefehle (nicht für Single-User-Mode nötig)
/var		Für vom System angelegte Zwischendateien [ <b>variable</b> ]

Mit \* gekennzeichnete Verzeichnisse werden zum Booten des Systems benötigt. Sie müssen daher auf der Root-Partition liegen (d.h. können nicht auf extra Partitionen liegen).

#### 4.5.3 LINUX-spezifische Standard-Verzeichnisse

/boot	Kern + Bootdateien ( <i>lilo</i> )
/cdrom	Mount-Point für CDROM-Laufwerk (auch <i>/media/cdrom</i> )
/dvd	Mount-Point für DVD-Laufwerk (auch <i>/media/dvd</i> )
/floppy	Mount-Point für Floppy-Laufwerk (auch <i>/media/floppy</i> )
/mnt	Mount-Point für temporäre Dateisysteme [ <b>mount</b> ]
/opt/kde[23]	Programme und Dateien für KDE-Oberfläche (Version 1/2/3)
/opt/gnome	Programme und Dateien für GNOME-Oberfläche
/proc	Pseudo-Verz. für Prozeß/Systeminformationen [ <b>process</b> ]
/root	Heimat-Verzeichnis der <i>root</i> (evtl. auch <i>/</i> )
/usr/src	Quellen der Systemsoftware (LINUX-Kern)
/usr/share	Dokumentationen (neu)
/usr/share/doc	Dokumentationen (neu)
/usr/share/man	Manual-Seiten (neu)
/usr/share/doc/howto	Howto-Seiten (neu)

#### 4.5.4 Die wichtigsten Gerätedateien von LINUX

/dev/cdrom	Link auf verwendetes CDROM Laufwerk
/dev/console	UNIX-Konsole
/dev/fd0 fd1	1./2. Diskettenlaufwerk
/dev/hda	Erste IDE-Festplatte
/dev/hda1 ... hda63	Partitionen der ersten IDE-Festplatte
/dev/hdb hdc ...	Zweite, dritte, ... IDE-Festplatte
/dev/mouse	Link auf von der Maus verwendete Schnittstelle
/dev/modem	Link auf angeschlossenes Modem (meist am COM-Port)
/dev/null	Verschluckt Eingaben, liefert EOF ( <i>Papierkorb, Schwarzes Loch!</i> )
/dev/psaux	PS/2-Maus
/dev/pts/NN	Virtuelle Konsole NN (in grafischer Oberfläche)
/dev/scd0	SCSI-CDROM ( <i>für CD-Brennen notwendig!</i> )
/dev/sda	Erste SCSI-Festplatte
/dev/sda1 ... sda15	Partitionen der ersten SCSI-Festplatte
/dev/sdb sdc ...	Zweite, dritte, ... SCSI-Festplatte
/dev/lp0 lp1 lp2	1-3. parallele Schnittstelle (LPT1 - LPT3)
/dev/ttyNN	Virtuelle Textkonsole NN
/dev/ttyS0 ... ttyS3	1-4. serielle Schnittstelle (COM1 - COM4)
/dev/zero	Liefert beliebig viele Nullbytes

#### 4.5.5 Wichtige zentrale Konfigurationsdateien

/etc/fstab	Beim Booten autom. zu mont. Dateisysteme [ <b>file system table</b> ]
/etc/group	Gruppenkonten (bis auf Paßwort)
/etc/gshadow	Gruppenpaßworte (verschlüsselt)
/etc/inittab	Bootvorgang steuern (Runlevel + Startup-Skripte) [ <b>initialization table</b> ]
/etc/magic	Datenbank mit Dateityp-Signaturen für file-Kommando
/etc/passwd	Benutzerkonten (bis auf Paßwort)
/etc/profile	Login-Skript der Shells sh, bash, ksh
/etc/printcap	Konfiguration der verfügbaren Drucker (lpd)
/etc/shadow	Benutzerpaßworte (verschlüsselt)
/etc/shells	Erlaubte Login-Shells (chsh)
/etc/XF86Config	X-Window Konfiguration

#### 4.5.6 Wichtige zentrale Netzwerk-Konfigurationsdateien

/etc/exports	Exportierte NFS-Laufwerke (zum Einhängen übers Netz)
/etc/hostname	Rechnername (evtl. HOSTNAME)
/etc/hosts	Zuordnung Rechnername ↔ IP-Adresse (Nameserver-Ersatz)
/etc/hosts.lpd	Erlaubte Hosts für Netzwerkdrucker-Zugriff [ <b>line print demon</b> ]
/etc/hosts.allow	Zugriffsteuerung tcpd erlaubt diesen Rechner den Zugang
/etc/hosts.deny	Zugriffsteuerung tcpd verbietet diesen Rechner den Zugang
/etc/inetd.conf	Von inetd zu startende Netzwerkdienste
/etc/named.conf	DNS-Konfiguration
/etc/networks	Zuordnung Netzname ↔ IP-Adresse (Nameserver-Ersatz)
/etc/nsswitch.conf	Definiert die Reihenfolge der Namensauflösung [ <b>name service switch</b> ]
/etc/protocols	Verfügbare Netzwerkprotokolle
/etc/resolv.conf	DNS-Suchreihenfolge + Nameserver festlegen
/etc/route.conf	Routing-Tabelle
/etc/services	Zuordnung Dienstname ↔ Portnummer

#### 4.5.7 Wichtige lokale Konfigurationsdateien/Verzeichnisse

.alias	Alias-Definitionen ( <i>bei SuSE von .bashrc gelesen!</i> )
.bashrc	Beim Start einer interaktiven Bash ausgeführt
.cshrc	Beim Start einer interaktiven csh ausgeführt
.emacs	Emacs-Konfiguration
.exrc	Vi/Ex-Konfiguration
.forward	Mail-Umlenkung
.login	Beim Start einer Login-csh ausgeführt
.logout	Beim Verlassen einer Login-csh ausgeführt
.netscape	Netscape-Einstellungen
.profile	Beim Start einer Login-Shell vom Typ sh, bash, ksh ausgeführt
.rhosts	Erlaubte Hosts für remote-Kommandos ohne Paßwort
.vimrc	Vim-Konfiguration
.xinitrc	X Window Start-Skript (startx → xinit)
.xsession	X Window Start-Skript (xdm oder kdm)



## 5 Prozeßverwaltung

### 5.1 Kommandos zur Prozeßverwaltung

ps ps -A ps -e ps -f ps -l ps -t TERM ps -u USER	Eigene Prozesse auflisten (inkl. ps selbst) <b>[process status]</b> Alle Prozesse <b>[all]</b> Alle Prozesse <b>[every]</b> Prozesse + alle Prozeßattribute <b>[full]</b> Prozesse + viele Prozeßattribute <b>[long]</b> Prozesse zu Terminal TERM anzeigen <b>[terminal]</b> Prozesse zu User USER anzeigen <b>[user]</b>
ps a ps f ps u ps w ps x	Prozesse aller Anwender (mit zugeordnetem Terminal) <b>[all]</b> Prozeßabhängigkeiten anzeigen <b>[forest]</b> Username mit anzeigen <b>[user]</b> Aufruf des Prozesses ausführlich anzeigen ( <i>mehrfach!</i> ) <b>[wide]</b> Prozesse ohne zugeordnetes Terminal (Dämonen) <b>[extended]</b>
kill [-SIGNAL] PID kill PID kill -9 PID kill -1 PID kill -l	Prozeß (durch Signal SIGNAL) beeinflussen <b>[töten]</b> Prozeß abbrechen (Standardsignal SIGTERM = 15, ignorierbar) Prozeß unbedingt abbrechen (SIGKILL = 9, nicht ignorierbar) Prozeß neu initialisieren (SIGHUP = 1) Liste aller Signal-Namen + Nummern ausgeben <b>[list]</b>
<Strg-C> (k)top ksysguard p(s)tree	Aktuellen Prozeß abbrechen (SIGINT = 2) <b>[break]</b> Prozeßliste + Prozeßattribute ständig anzeigen (LINUX) Analog (LINUX) Prozeßabhängigkeiten (Baum) anzeigen (LINUX) <b>[process tree]</b>

Die ps-Optionen der UNIX-Hauptlinien BSD und System V unterscheiden sich in ihrer Bedeutung (leider). Daher wird unter LINUX eine Option mit führendem - als BSD-Option und ohne führendes - als System V-Option interpretiert.

### 5.2 Kommandos zur Hintergrund-Prozeß-Verwaltung

CMD ... & <Strg-Z>	Kommando im Hintergrund starten Aktuellen Prozeß stoppen + in Hintergrund stellen (SIGTSTP = 20)
bg [JOBNR] fg [JOBNR] jobs kill %JOBNR nohup CMD...	Prozeß mit JOBNR im Hintergrund starten <b>[background]</b> Prozeß mit JOBNR in den Vordergrund holen <b>[foreground]</b> Hintergrundprozesse auflisten Hintergrundprozeß JOBNR beenden (% <i>notwendig!</i> ) CMD wird beim Abmelden nicht abgebrochen <b>[no hangup]</b> (Ein/Ausgabe von/auf Datei erhalten/machen, Std: nohup.out)
stop JOBNR	Prozeß mit JOBNR im Hintergrund anhalten (kill -STOP, ksh)

### 5.3 Die wichtigsten Signale

SIGHUP	1	Konfigurationsdatei lesen [ <b>hangup</b> ]
SIGINT	2	Abbruch durch Eingabe von <Strg-C> [ <b>interrupt</b> ]
SIGQUIT	3	Prozeßende erreicht
SIGKILL	9	Bedingungsloser Prozeßabbruch [ <b>töten</b> ]
SIGTERM	15	Prozeß beenden, wird evtl. ignoriert [ <b>terminate</b> ]
SIGCONT	18	Ausgabe weiterlaufen lassen durch Eingabe von <Strg-Q> [ <b>continue</b> ]
SIGSTOP	19	Ausgabe anhalten durch Eingabe von <Strg-S>
SIGTSTP	24	In Hintergrund stellen durch Eingabe von <Strg-Z>

## 6 Shell

Kommentar wird in der Shell durch # eingeleitet und gilt bis zum Zeilenende.

### 6.1 Shell-Typen

sh	Bourne-Shell	Ur-Shell, Skript-orientiert
csh	C-Shell	Angelehnt an C, Interaktions-orientiert
tcsh	Tenex-C-Shell	Erweiterung + Verbesserung der C-Shell (BSD)
ksh	Korn-Shell	Standard unter UNIX System V
bash	Bo(u)rn(e)-Again-Shell	Standard unter LINUX/GNU
ash	Anfang des Alphabets	Sehr einfache Shell, klein
zsh	Ende des Alphabets	Ultimative Shell, extremer Funktionsumfang

Shells sind sowohl zur *interaktiven* Benutzung auf der Kommandozeile als auch zur Ausführung von *Shell-Skripten* (Batch-Dateien) gedacht. Hier eine Bewertung ihrer Eignung für den interaktiven (I) bzw. den Batch-Gebrauch (B):

Name	SRC	Int	Batch
sh		--	+
csh		+	-
tcsh	csh	++	+-
ksh	sh	+	++
bash	sh	++	++
ash	sh	+-	+-
zsh	sh	++	++

### 6.2 Shell-Konfigurationsdateien

Es muss unterschieden werden zwischen Login-Shells (die beim Anmelden gestartet werden) und (interaktive) Sub-Shells, die zum Ausführen eines Shell-Skriptes oder Kommandos gestartet werden.

### 6.2.1 Konfigurationsdateien der Bourne-Shell

1	/etc/profile	Zentrales Login-Skript
2	~/.profile	Benutzerabhängiges Login-Skript
		Interaktive Bourne-Shells lesen keine Konf.dateien!
		Bourne-Shells führen beim Logout kein Skript aus!

### 6.2.2 Konfigurationsdateien der Bash

1	/etc/profile	Zentrales Login-Skript
2a	~/.bash_profile	Benutzerabhängiges Login-Skript oder
2b	~/.bash_login	Benutzerabhängiges Login-Skript oder
2c	~/.profile	Benutzerabhängiges Login-Skript
	~/.bashrc	Beim Start einer interaktiven Bash ausgeführt
	~/.alias	Alias-Definitionen (nur bei SuSE?)
	~/.bash_logout	Beim Verlassen einer Bash-Login-Shell ausgeführt

### 6.2.3 Konfigurationsdateien der C-Shell

1	/etc/cshrc	Beim Start einer Login-csh ausgeführt
2	~/.cshrc	Beim Start einer Login-csh ausgeführt
3	~/.login	Benutzerabhängiges Login-Skript
	~/.cshrc	Beim Start einer interaktiven csh ausgeführt
	~/.logout	Beim Verlassen einer Login-csh ausgeführt

### 6.2.4 Konfigurationsdateien der TC-Shell

1	/etc/csh.cshrc	Zentrales Login-Skript der tcsh
2	/etc/csh.login	Zentrales Login-Skript der tcsh
3a	~/.tcshrc	Beim Start einer Login-tcsh ausgeführt oder
3b	~/.cshrc	Beim Start einer Login-tcsh ausgeführt
4	~/.history	Benutzerabhängiges Login-Skript der tcsh
5	~/.login	Benutzerabhängiges Login-Skript der tcsh
6	~/.cshdirs	Benutzerabhängiges Login-Skript der tcsh
1	/etc/csh.cshrc	Zentrales Skript beim Start einer interaktiven tcsh
2a	~/.tcshrc	Beim Start einer interaktiven tcsh ausgeführt oder
2b	~/.cshrc	Beim Start einer interaktiven tcsh ausgeführt
1	/etc/csh.logout	Beim Verlassen einer Login-tcsh ausgeführt
2	~/.logout	Beim Verlassen einer Login-tcsh ausgeführt

### 6.2.5 Konfigurationsdateien der Korn-Shell

1	/etc/profile	Zentrales Login-Skript
2	~/.profile	Benutzerabhängiges Login-Skript
	\$ENV	Bestimmt die für eine interaktive ksh auszuführende
	~/.kshrc	Datei (Inhalt ~/.kshrc) <b>[environment]</b>

## 6.3 Shell-Operationen

1	History-Ersetzung (Teile alter Kommandos wiederverwenden)
2	Zerlegung in Token (Whitespace trennt die einzelnen Worte)
3	Quotierung berücksichtigen (@0@'...' und \)
4	Aliase ersetzen
5	I/O-Umlenkungen einrichten (< > >> 2> 2>> >& 1>&2  )
6	Variablen durch ihren Wert ersetzen (\$VAR)
7	Kommando-Substitution `...' \$(...) durchführen (durch Ergebnis ersetzen)
8	Dateinamen expandieren (Metazeichen * ? [] ~ {} ersetzen)
9	Kommando suchen (Pfadsuche anhand Variabler PATH)

## 6.4 Pfadsuche (bei Aufruf von CMD)

1	Built-in/Keyword (z.B. cd, echo, exit, ...)
2	Alias (z.B. alias ll='ls -lF')
3	Funktion (z.B. ll() { /bin/ls -lF \$*; })
4	Von links nach rechts in allen im Suchpfad PATH angegebenen Verzeichnissen (durch : getrennt) nach einer Datei namens CMD suchen; diese muss ausführbar (+ lesbar bei Shell-Skript) sein
5	Nicht gefunden → Meldung: error: command CMD not found

Standardmäßig wird nicht nach Kommandos im aktuellen Verzeichnis gesucht, außer im Suchpfad steht . (Punkt). Die Suche kann simuliert werden durch:

type CMD	Kommando-Typ ermitteln (Builtin, Alias, Funktion, Programm)
type -a CMD	Alle passenden Programme (nicht nur das 1.) ausgeben <b>[all]</b>
type -p CMD	Nur Suchpfad zu passendem Programm ausgeben <b>[path]</b>
which CMD	Entspricht type -p bei der (t) csh
whence CMD	Entspricht type -p bei der ksh

## 6.5 Shell- und Umgebungs-Variablen

### 6.5.1 Kommandos für Shell-Variablen

VAR="TEXT"	Erzeugt eine Shell-Variable und weist ihr den Wert TEXT zu (keine Leerzeichen um = erlaubt!)
set	Alle Shell-Variablen auflisten
\$VAR	Zugriff auf den Wert (Inhalt) einer Shell-Variablen
\${VAR}xxx	Alternative Variante falls direkt dahinter Text xxx steht
VAR=	Löschen einer Shell-Variablen (leer)
unset VAR	Löschen einer Shell-Variablen (undefiniert != leer)

### 6.5.2 Kommandos für Umgebungs-Variablen

export	Alle nicht exportierten Variablen auflisten
export VAR	Shell-Variable zu einer Umgebungs-Variable machen
export VAR="TEXT"	Analog + gleichzeitige Wertzuweisung (bash,ksh)
env	Alle Umgebungs-Variablen anzeigen

### 6.5.3 Variablen-Kommandos unter der (T)C-Shell

set VAR = "TEXT"	Erzeugt eine Shell-Variable ( <i>Leerzeichen um = nötig!</i> )
set VAR = (aaa bbb)	Array-Variable mit Werteliste füllen
set	Alle Shell-Variablen auflisten
unset VAR	Löschen einer Shell-Variablen (undefiniert != leer)
setenv VAR "TEXT"	Erzeugt eine Umgebungs-Variable ( <i>kein =!</i> )
setenv VAR (aaa bbb)	Array-Variable mit Werteliste füllen
printenv	Alle Umgebungs-Variablen anzeigen
unsetenv VAR	Löschen einer Umgebungs-Variablen

### 6.5.4 Einige Standard-Variablen

CDPATH	Suchpfad für <b>cd</b> [ <b>change directory path</b> ]
HOME	Standardverzeichnis für <b>cd</b> (Heimat-Verzeichnis)
EDITOR	Definiert den Kommandozeilen-Editor ( <i>vi/emacs</i> )
IFS	Whitespace-Zeichen für <b>read</b> [ <b>internal field separator</b> ]
LANG	Sprache für Fehlermeldungen einstellen [ <b>language</b> ]
LOGNAME	Aktueller Loginname (auf manchen Systemen nicht vorhanden)
MANPATH	Suchpfad für <i>man, whatis, apropos</i>
OLDPWD	Vorheriges aktuelles Verzeichnis [ <b>print working directory</b> ]
PAGER	Programm zum seitenweisen Blättern ( <i>more</i> )
PATH	Suchpfad für Kommandoaufruf
PS1	Shell-Prompt (\$ )
PS2	Fortsetzungs-Prompt (> )
PS3	Prompt der <i>select</i> -Menüabfrage
PS4	Debug-Prompt (+ )
PWD	Aktuelles Verzeichnis [ <b>print working directory</b> ]
RANDOM	Zufallszahl (0..32767)
SHELL	Name der Login-Shell
TERM	Terminaltyp (Editoren, <i>more</i> , <i>curses</i> -Bibliothek)
TZ	Zeitzone [ <b>time zone</b> ]
USER	Aktueller Loginname (auf manchen Systemen nicht vorhanden)

### 6.5.5 Eingabeprompt-Definition (bash)

Die Variable `PS1` legt das Aussehen des Eingabeprompts fest. Sie wird entweder für alle Benutzer gemeinsam in `/etc/profile` oder für einzelne Benutzer in `~/.profile` initialisiert. Folgende Sonderzeichen in `PS1` setzen spezielle (variable) Komponenten im Prompt ein:

\d	Datum in der Form Sun Dec 24 <b>[date]</b>
\H	Kompletter Rechner-Name (FQDN) <b>[host]</b>
\h	Rechner-Name bis zum ersten Punkt (.) <b>[host]</b>
\n	Zeilenvorschub <b>[newline]</b>
\r	Zeilenrücklauf <b>[carriage return]</b>
\s	Shell-Name <b>[shell]</b>
\t	Zeit in der Form hh:mm:ss <b>[time]</b>
\u	Benutzer-Name <b>[user]</b>
\w	Arbeitsverzeichnis <b>[working directory]</b>
\W	Arbeitsverzeichnis (nur Teil nach dem letzten /) <b>[working directory]</b>
\\$	\$-Zeichen für normalen Benutzer, #-Zeichen für root
\#	Nummer des aktuellen Kommandos (in der aktuellen Sitzung)
\!	History-Nummer des aktuellen Kommandos (über Sitzungen hinweg)
\[	Beginnt Terminalsteuersequenz (zählt nicht zur Kommandolänge mit)
\]	Beendet Terminalsteuersequenz (zählt nicht zur Kommandolänge mit)

### 6.5.6 Eingabeprompt-Definition (tcsh)

Die Variable `prompt` legt das Aussehen des Eingabeprompts fest. Sie wird entweder für alle Benutzer gemeinsam in `/etc/tcshrc.rc` oder für einzelne Benutzer in `~/.login` initialisiert. Folgende Sonderzeichen in `prompt` setzen spezielle (variable) Komponenten im Prompt ein:

%d/%D	Datum-Tag <b>[day]</b>
%m/%M	Datum-Monat <b>[month]</b>
%y/%Y	Datum-Jahr <b>[year]</b>
%t/%T	Uhrzeit <b>[time]</b>
%M	Kompletter Rechner-Name (FQDN) <b>[machine]</b>
%m	Rechner-Name bis zum ersten Punkt (.) <b>[machine]</b>
%n	Benutzer-Name <b>[name]</b>
%c	Arbeitsverzeichnis <b>[current directory]</b>
%C2	Arbeitsverzeichnis (nur 2 Teile) <b>[current directory]</b>
%#	\$-Zeichen für normalen Benutzer, #-Zeichen für root
%h	Nummer des aktuellen Kommandos <b>[history]</b>
%b/%B	Fettschrift an/aus <b>[bold]</b>
%s/%S	Hervorhebung an/aus <b>[standout]</b>

## 6.6 Kommando-Wiederholung (bash + ksh)

HISTFILE	Dateiname zur Speicherung der Kmdos. (Std: ~/.bash_history)
HISTSIZE	Anzahl zu merkender Kommandos (Std: 50/500)
history	Die letzten HISTSIZE Kommandos auflisten
history NN	Die letzten NN Kommandos auflisten
!NR	Kommando mit der Nummer NR wiederholen (bash)
!TEXT	Kommando mit Textanfang TEXT wiederholen (bash)
r NR	Kommando mit der Nummer NR wiederholen (ksh) <b>[repeat]</b>
r TEXT	Kommando mit Textanfang TEXT wiederholen (ksh)
^OLD^NEW	Letztes Kommando wiederh., zuvor OLD durch NEW ersetzen
!!	Letztes Kommando wiederholen (auch in der Form vi '!!')
!-2	Vorletztes Kommando wiederholen, usw.

## 6.7 Dateinamen-Vervollständigung (bash + ksh)

In der `bash` sind verwendbar:

NNN <TAB>	Kommando-Namen NNN mit passend. Kdo. vervollständigen ( <i>1. Wort!</i> )
NNN <TAB><TAB>	Liste der Alternativen bei mehrdeutigem Namen NNN anzeigen
CMD NNN <TAB>	Datei-Namen vervollständigen ( <i>nicht 1. Wort!</i> )
\$NNN <TAB>	Variablen-Namen vervollständigen ( <code>bash</code> )
~NNN <TAB>	Benutzer-Namen vervollständigen ( <code>bash</code> )
@NNN <TAB>	Rechner-Namen vervollständigen ( <code>bash</code> )

In der `ksh` sind verwendbar:

<ESC> *	Aktuelles Wort zu allen passenden Dateinamen expandieren
<ESC> \	Aktuelles Wort durch eindeutigen passenden Dateinamen ersetzen
<ESC> /TEXT	Kommando mit TEXT darin suchen und ausführen

Funktioniert für Kommandos (1. Wort) und Dateinamen (ab 2. Wort).

## 6.8 Aliase

alias ll='/bin/ls -lF'	Alias ll definieren ( <i>Rekursion vermeiden!</i> )
ll *.c *.h	Wird zu /bin/ls -lF *.c *.h expandiert
alias	Alle Aliase anzeigen
alias ll	Alias ll anzeigen
unalias ll	Alias ll löschen

### 6.8.1 Nützliche Aliase

c	clear	Bildschirm löschen
cx	chmod u+x	Ausführungsrecht setzen
h	history	Die letzten Befehle anzeigen
hg	history   grep	In den letzten Befehlen suchen
ld	ls -l   grep '^d'	Verzeichnisse des aktuellen Verzeichnisses ausgeben
lt	ls -ltr	Dateien nach Änderungsdatum sortieren (neueste zuletzt)
lss	ls -l   sort +4 -5	Dateien nach Größe sortieren (größte zuletzt)
m	less	Dateien seitenweise anzeigen
mfl	mount /floppy	Floppy einhängen
umfl	umount /floppy	Floppy aushängen
va	vi \$HOME/.alias	Eigene Aliase editieren
sa	. \$HOME/.alias	Eigene Aliase aktivieren

## 6.9 Funktionen

<pre>ll() {     /bin/ls -lF } ll() { /bin/ls -lF; } typeset -f typeset -f ll unset -f ll</pre>	<p>Funktion <code>ll</code> definieren (<i>Rekursion vermeiden!</i>)</p> <p>Analog in einer Zeile (<code>;</code> + Leerzeichen nach <code>}</code> sind nötig!)</p> <p>Alle Funktionen anzeigen</p> <p>Funktion <code>ll</code> anzeigen</p> <p>Funktion <code>ll</code> löschen</p>
--	---

## 6.10 Ein/Ausgabe-Umlenkung

<code>CMD &lt; FILE</code>	stdin von Datei <code>FILE</code> lesen
<code>CMD &gt; FILE</code>	stdout auf Datei <code>FILE</code> schreiben ( <i>vorher gelöscht!</i> )
<code>CMD &gt;&gt; FILE</code>	stdout an Datei <code>FILE</code> anhängen
<code>CMD 2&gt; FILE</code>	stderr auf Datei <code>FILE</code> schreiben ( <i>vorher gelöscht!</i> )
<code>CMD 2&gt;&gt; FILE</code>	stderr an Datei <code>FILE</code> anhängen
<code>CMD1   CMD2</code>	stdout von <code>CMD1</code> mit stdin von <code>CMD2</code> verbinden [pipe]

## 6.11 Spezielle Ein-/Ausgabeumlenkung

<code>CMD 1&gt;&amp;2</code>	stdout(1) zu stderr(2) hinzufügen (kombinieren)
<code>CMD 2&gt;&amp;1</code>	stderr(2) zu stdout(1) hinzufügen (kombinieren)
<code>CMD &amp;&gt; FILE</code>	stdout und stderr auf Datei <code>FILE</code> schreiben ( <i>vorher gelöscht!</i> ) entspricht <code>&gt; FILE 2&gt;&amp;1</code> oder <code>2&gt; FILE 1&gt;&amp;2</code>
<code>CMD &gt;&amp; FILE</code>	stdout zu stderr hinzufügen (csh)
<code>CMD &gt;  FILE</code>	stdout auf Datei <code>FILE</code> schreiben ( <i>erzwingen bei noclobber!</i> )
<code>CMD1  &amp; CMD2</code>	stdout und stderr an <code>CMD2</code> übergeben

## 6.12 Here-Dokument

<code>CMD &lt;&lt;TEXT</code>	stdin folgt, bis <code>TEXT</code> in einer Zeile für sich steht (Ersetzung von <code>\$VAR</code> , <code>`CMD`</code> , <code>\$(CMD)</code> und <code>\</code> wird durchgeführt)
<code>CMD &lt;&lt;-TEXT</code>	Analog, aber Tabulatoren am Zeilenanfang ignorieren
<code>CMD &lt;&lt;"TEXT"</code>	Analog, aber keine Ersetzung von <code>\$VAR</code> , <code>`CMD`</code> , <code>\$(CMD)</code> und <code>\</code>
<code>CMD &lt;&lt;'TEXT'</code>	Analog, aber keine Ersetzung von <code>\$VAR</code> , <code>`CMD`</code> , <code>\$(CMD)</code> und <code>\</code>
<code>CMD &lt;&lt;\TEXT</code>	Analog, aber keine Ersetzung von <code>\$VAR</code> , <code>`CMD`</code> , <code>\$(CMD)</code> und <code>\</code>



### 6.13 Dateinamen-Expansion

/	Verzeichnis-Trenner ( <i>nicht in Dateinamen verwendbar!</i> )
?	Genau ein beliebiges Zeichen
*	0 oder mehr beliebige Zeichen ( <i>Wh. von Z. ist nicht möglich!</i> )
\X	Das Metazeichen x selbst (der Backslash \ quotiert)
[abc] [a-z]	Genau ein Zeichen aus der angegebenen Menge
[!abc] [!a-z]	Genau ein Zeichen nicht(!) aus der angegebenen Menge ( <i>sh</i> )
[^abc] [^a-z]	Genau ein Zeichen nicht(!) aus der angegebenen Menge ( <i>cs</i> )
~	Heimat-Verzeichnis des aktuellen Benutzers ( <i>tcsh, bash</i> )
~USER	Heimat-Verzeichnis des Benutzers USER ( <i>tcsh, bash</i> )
{abc,def,...}	Liste der angegebenen Zeichenketten ( <i>cs, ksh, bash</i> )

### 6.14 Kommando-Substitution

`CMD`	CMD ausführen und durch Ergebnis ersetzen (alte Form, in jeder Shell)
\$(CMD)	Analog (neue Form, nur in <i>bash/ksh</i> , ähnelt Variablen-Ersetzung)

### 6.15 Shell-Quotierung

'...'	Sämtliche Sonderzeichen abschalten [ <b>single quote/tick</b> ]
"..."	Alle Sonderzeichen bis auf \$, `...`, \$(...) und \ abschalten [ <b>double quote</b> ]
\X	Genau ein (das folgende) Sonderzeichen x abschalten [ <b>backslash</b> ]

### 6.16 Kommando-Listen

(CMD; ...)	Kmdos in neuer Shell ausführen (gemeinsam Umlenken, Hintergrund)
{CMD; ...; }	Kmdos in akt. Shell ausführen (gemeinsam Umlenken, Hintergrund)
CMD1 ; CMD2	CMD1 ausführen, danach CMD2 ausführen ( <i>wartet auf Ende von CMD1!</i> )
CMD1 & CMD2	CMD1 starten, danach CMD2 starten ( <i>wartet nicht auf Ende von CMD1!</i> )
CMD1 && CMD2	CMD1 ausführen, bei Erfolg CMD2 ausführen (CMD1 hat Exit-Status 0)
CMD1    CMD2	CMD1 ausführen, bei Mißerfolg CMD2 ausführen (CMD1 hat Exit-Status 1)

## 7 Sonstige Kommandos

clear	Löscht den Bildschirminhalt eines Terminals
echo TEXT1 ...	Gibt TEXT1 ... auf Terminal aus (durch je 1 Leerzeichen getrennt)
echo \$SHELL	Name der Login-Shell ausgeben
banner "TEXT"	TEXT groß geschrieben ausgeben
date [+FORMAT]	Gibt Datum + Uhrzeit aus (gemäß FORMAT, s.u.)
cal [MON] YEAR	Jahres/Monatskalender ausgeben ( <i>Sep 1752 ist ungewöhnlich!</i> )
sleep [NN]	1 (NN) Sekunden warten
fmt -w 70 FILE	Absätze (Leerzeilen trennen) auf 70 Z. Breite umbrechen [ <b>format</b> ]
pr FILE	Datei FILE seitenweise für Ausdruck aufteilen [ <b>print</b> ]
spell FILE	Datei FILE korrekturlesen [ <b>buchstabieren</b> ]
troff FILE	UNIX-Textformatierung von FILE [ <b>text raster output file format</b> ]
write USER@HOST	Nachricht (stdin) auf Bildschirm von USER schreiben
wall	Nachricht (stdin) auf Bildschirm aller User schreiben [ <b>write all</b> ]
talk USER@HOST	Mit USER telefonieren (zweigeteilter Bildschirm)

### 7.1 Wichtige Format-Angaben zu date

%H	Stunde [ <b>hour</b> ]
%M	Minute [ <b>minute</b> ]
%S	Sekunde [ <b>second</b> ]
%T	Uhrzeit hh:mm:ss [ <b>time</b> ]
%d	Tag [ <b>day</b> ]
%M	Monat [ <b>month</b> ]
%Y/%Y	Jahr zweistellig/vierstellig [ <b>year</b> ]
%D	Datum dd/mm/yy [ <b>date</b> ]
%a	Wochentagname (Sun, ..., Sat)
%w	Wochentagnummer (0=Sonntag, ..., 6=Samstag) [ <b>weekday</b> ]
%b	Monatsname (Jan, ..., Dec)
%j	Tag des Jahres (1..366)
%W	Wochennummer des Jahres (1..53) [ <b>week</b> ]

## 8 Vi

### 8.1 Die wichtigsten Vi-Befehle

~/.exrc ~/.vimrc ESC ESC :q! CR	Vi/Vim-Konfigurationsdatei [ex/vim resource] Kommando-Modus beenden (2x → piepst + ignoriert) Vi sicher verlassen (ohne Änderung an Datei)
vi [FILE] CR :wq CR :x CR ZZ :q CR :q! CR :w CR :w FILE CR	Datei FILE bzw. leere Datei (:w FILE <i>nötig!</i> ) editieren Editierte Datei schreiben + Vi verlassen [write+quit/exit] Editierte Datei nicht schr. + Vi verlassen (!=erzwingen!) [quit] Editierten Text (auf Datei FILE) schreiben [write]
:e[!] FILE CR :f FILE CR	Datei FILE statt aktueller editieren (!=erzwingen!) [edit] Aktuelle Datei mit dem Namen FILE benennen [file]
:n CR :prev :N VIM! :args CR :rew CR	Nächste Datei editieren (falls mehrere angegeben) [next] Vorherige Datei editieren [previous/Next] Alle beim Aufruf des Vi angeg. Dateien anzeigen [arguments] Zur 1. beim Aufruf angegeb. Datei zurückspringen [rewind]
a TEXT... ESC A TEXT... ESC i TEXT... ESC I TEXT... ESC o TEXT... ESC O TEXT... ESC	Nach Cursorposition TEXT... einfügen [append] Am Zeilenende TEXT... einfügen [Append] Vor Cursorposition TEXT... einfügen [insert] Am Zeilenanfang TEXT... einfügen [Insert] Neue Zeile TEXT... unter aktueller Zeile einfügen [open] Neue Zeile TEXT... über aktueller Zeile einfügen [Open]
x X dd dw dMOVE	Ein Zeichen unter/vor Cursor löschen [crossout] Zeile/Wort/gemäß Bewegung MOVE löschen [delete]
rX R TEXT... ESC s TEXT... ESC cc cw cMOVE ESC yy yw yMOVE ESC p P	Ein Zeichen unter Cursor mit X übertippen ( <i>kein ESC!</i> ) [replace] Ab Cursorposition mit TEXT... übertippen [Replace] Ein Zeichent unter Cursor mit TEXT... übertippen [substitute] Zeile/Wort/gemäß Bewegung MOVE ändern [change] Zeile/Wort/gemäß Bew. MOVE in Zw.speicher kopieren [yank/copy] Zwischensp. nach/vor akt. Zeichen/Zeile einfügen [put]
h l j k w b e W B E 0 ^ \$ NN  <Strg-F> <Strg-B> G lG NNG mX 'X `X	Zeichen links/rechts (auch <BACKSPACE>/<SPACE>) Zeile ab/auf (auch +/-<RETURN>) Wort vor/zurück/Wortende [word/backword/endword] WORT vor/zurück/WORTende ( <i>inkl. Ziffern, Satzzeichen!</i> ) Zum Zeilenanfang/1.Buchstaben/Zl.ende/Spalte NN springen Bildschirmseite auf/abblättern [forward/backward] Zum Dateiende/Dateianfang/Zeile NN springen [Go] Marke X (a-z) setzen, Zeile/Zeichen mit M. X anspringen [mark]
/SUCH CR ?SUCH CR :%s/SUCH/TEXT/gc CR n N	Text SUCH vorwärts/rückwärts suchen Text SUCH überall in Datei durch TEXT ersetzen (%=1, \$=alle Zeilen, s=substitute, g=global, c=confirm) Letzte Suche vor/zurück wiederholen [next]
u U <Strg-R> VIM! .	Letzte Änderung/alle Änd. einer Zeile zurücknehmen [undo] Letzte Änderung wiederherstellen (2x u im Vi) [redo] Letztes Edit-Kommando (Änderung) wiederholen [Punkt]

Mit VIM! markierte Befehle sind nur im Editor Vim möglich.

## 8.2 Weitere wichtige Vi-Befehle

% ~ J << >> xp dwWP ddp v V <Strg-V> VIM! :e# <Strg-6>	Passende Klammer zu { [ ( < > ) ] } suchen Groß/Kleinschreibung von aktuellem Zeichen vertauschen Nachfolgende Zeile an aktuelle anhängen [ <b>join</b> ] Aktuelle Zeile einrücken/ausrücken (Option <code>shiftwidth</code> ) Zwei Zeichen/Worte/Zeilen vertauschen Block zeichen/zeilen/spaltenorientiert markieren [ <b>visual</b> ] Vorherige Datei editieren
"XyMOVE "XdMOVE "XP "Xp	Von MOVE ausgewählten Text in Puffer x (a-z) kopieren [ <b>yank</b> ] Von MOVE ausgewählten Text in Puffer x (a-z) verschieben [ <b>del</b> ] Inhalt von Puffer x (a-z) vor Cursor/aktueller Zeile einfügen [ <b>put</b> ] Inhalt von Puffer x (a-z) nach Cursor/aktueller Zeile einfügen
:map MACRO TEXT :unmap MACRO :map	MACRO mit Ersatztext TEXT für Kommando-Modus definieren MACRO mit Ersatztext TEXT für Kommando-Modus löschen Alle Makros für Kommando-Modus anzeigen
:map! MACRO TEX :unmap! MACRO :map!	MACRO mit Ersatztext TEXT für Eingabe-Modus definieren MACRO mit Ersatztext TEXT für Eingabe-Modus löschen Alle Makros für Eingabe-Modus anzeigen
:ab ABBREV TEXT :unab ABBREV :ab	Abkürzung ABBREV für Eingabe-Modus definieren ( <i>Leerzeichen!</i> ) Abkürzung ABBREV für Eingabe-Modus löschen Alle Abkürzung für Eingabe-Modus anzeigen

## 8.3 Wichtige allgemeine Vi-Optionen

errorbells ignorecase list number report=NN showmode wrapmargin=NN wrapscan	Bei Eingabefehler piepsen Groß/Kleinschreibung bei der Suche ignorieren Zeilenenden durch \$ und Tabulatoren durch ^I anzeigen Zeilen numeriert anzeigen Änderungen ab NN Zeilen in Statuszeile anz. ( <i>1=ab 1 Zeile, 0 geht n.!</i> ) Vi-Modus in Statuszeile anzeigen (INSERT, APPEND, ...) Automatischer Umbruch NN Zeichen vor Zeilenende ( <i>0=kein Umbruch!</i> ) Suche über Dateiende/anfang hinweg fortsetzen
--	---

Optionen mit `:set OPTION` setzen und mit `:set noOPTION` (*zusammengeschrieben!*) löschen. In Konfigurations-Datei `.exrc/.vimrc` den Doppelpunkt vor `set` weglassen. Kommentare in `.exrc/.vimrc` durch `"` einleiten (*keine Leerzeilen erlaubt!*).

## 8.4 Wichtige Vi-Optionen für Programmierer

autoindent shiftwidth=NN showmatch tabstop=NN	Automatisch einrücken (mit <Strg-D/T> ein/ausrücken) Breite NN für nachträgliches Einrücken setzen (<< >>) Passende öffnende Klammer bei Eingabe der schließenden anzeigen Tabulatorbreite auf NN Zeichen setzen
--	---

## 8.5 Wichtige Vim-Optionen

nocompatible	Nicht im Vi-Kompatibilitätsmodus arbeiten
ruler	Koordinatenanzeige (Zeile + Spalte) in Statuszeile [ <b>Lineal</b> ]
:syntax on	Syntaxcoloring an (abhängig von Datei-Extension)
wrap	Zu lange Zeilen umgebrochen darstellen
incsearch	Inkrementell suche [ <b>incremental</b> ]
hlsearch	Zur Suche passende Textteile markieren [ <b>highlight</b> ]
showcmd	Unvollständige Kommandos in Statuszeile anzeigen [ <b>command</b> ]
ff=unix	Dateiformat (dos, unix, mac) [ <b>fileformat</b> ]

## 8.6 Nützliche Vi-Makros

K	!}fmt -80 -u^M	Absatz (bis Leerzeile) auf 80 Zeichen Breite formatieren
D	:%s/ *\$/^M	Alle Leerzeichen am Zeilenende entfernen
Strg-W	/\<	Suche nach Wort einleiten (danach Wort eintippen)
Strg-W	\>/^M	Suche nach Wort durchführen ( <i>mit :map! definieren!</i> )
+	:n^M	Zur nächsten Datei springen (ohne Abspeichern)
#	:w^M:n^M	Zur nächsten Datei springen (aktuelle vorher speichern)
-	:prev^M VIM!	Zur vorherigen Datei springen (ohne Abspeichern)

Mit `:map NAME MACRO` definieren, `^M` ist einzugeben durch `<Strg-V> <RETURN>` oder im Vim durch `<CR>`.

## 9 Drucken

### 9.1 BSD-Variante der Druck-Kommandos

lpr FILE	Datei FILE drucken [ <b>line print</b> ]
CMD   lpr	Ergebnis von Kommando CMD drucken
lpq	Abgesetzte Druckaufträge anzeigen [ <b>line print queue</b> ]
lprm JOBNR	Druckauftrag JOBNR löschen [ <b>line print remove</b> ]
lprm -	Alle Druckaufträge löschen ( <i>nur root!</i> )
lpc [CMD]	Drucker verwalten [ <b>line print control</b> ]
PRINTER	Umgebungsvariable zur Definition des Standard-Druckers
lp	Name des Standard-Druckers (falls Variable PRINTER leer)
-P PRINTER	Statt Standard-Drucker den Drucker PRINTER ansprechen

### 9.2 System V-Variante der Druck-Kommandos

lp FILE	Datei FILE drucken [ <b>line print</b> ]
CMD   lp	Ergebnis von Kommando CMD drucken
lpstat	Abgesetzte Druckaufträge anzeigen [ <b>line print status</b> ]
cancel JOBNR	Druckauftrag JOBNR löschen
-d PRINTER	Drucker festlegen [ <b>destination</b> ]
LPDEST	Umgebungsvariable zur Definition des Standard-Druckers
lp	Name des Standard-Druckers (falls Variable LPDEST leer)
-d PRINTER	Statt Standard-Drucker den Drucker LPDEST ansprechen [ <b>destination</b> ]

### 9.3 lpc-Kommandos

status [QUEUE]	Status von Warteschlange QUEUE/allen Warteschlangen ausgeben
help [CMD]	Befehlsliste/Hilfe zu CMD anzeigen
quit/exit	lpc verlassen
enable QUEUE	Warteschlange QUEUE aktivieren
disable QUEUE	Warteschlange QUEUE deaktivieren
start QUEUE	Ausdruck von QUEUE starten
stop QUEUE	Ausdruck von QUEUE stoppen
up QUEUE	enable + start
down QUEUE	disable + stop
restart QUEUE	Dämon der Warteschlange QUEUE stoppen + neu starten
clean QUEUE	Zwischendateien der Warteschlange QUEUE entfernen
topq QUEUE NR	Job NR an die Spitze der Warteschlange QUEUE setzen

## 10 Reguläre Ausdrücke

### 10.1 Standard-Metazeichen

.	Ein beliebiges Zeichen
X*	0 oder mehr Wiederholungen des Zeichens x
^	Zeilenanfang
\$	Zeilenende
\X	Zeichen x quotieren
[abc] [a-z]	Menge von Zeichen ([a-z] = ASCII-Zeichenbereich)
[^abc] [^a-z]	Negierte Menge von Zeichen (auch [^A-Z], [^0-9])

### 10.2 Erweiterte Metazeichen

?	0 oder 1 Wiederholung des Teils davor (optional)
+	1 oder mehr Wiederholungen des Teils davor
	Entweder-Oder (alternativ)
(...)	Klammerung mehrerer Zeichen
\{M,N\}	M bis N Wiederholungen des Teils davor
\{M,\}	M oder mehr Wiederholungen des Teils davor
\{M\}	Genau M Wiederholungen des Teils davor
\n	Zeilenvorschub [ <b>newline</b> ]
\(...\)	Zeichenkette merken (in \1..\9)
\< \>	Wortanfang, Wortende (\b \B in perl)

### 10.3 Metazeichen im Ersetzungsmuster

\N	N-te per \(...\) gemerkte Zeichenkette (N=1..9)
&	Kompletten zu Muster gefundenen Text wieder einsetzen
~	Vorheriges Suchmuster verwenden
\u \l	Nächstes Zeichen in Groß/Kleinschrift umwandeln [ <b>up/lowcase</b> ]
\U \L	Alle folgenden Zeichen in Groß/Kleinschrift umwandeln
\E	Durch \U oder \L begonnene Umwandlung beenden [ <b>end</b> ]

### 10.4 Escape-Sequenzen

\a	[ <b>alert</b> ]	Akustisches Signal
\b	[ <b>backspace</b> ]	Zeichen vorher löschen
\f	[ <b>form feed</b> ]	Seitenvorschub
\n	[ <b>newline</b> ]	Zeilenvorschub
\r	[ <b>carriage return</b> ]	Wagenrücklauf
\t	[ <b>tabulator</b> ]	Tabulator
\v	[ <b>vtab</b> ]	Vertikaler Tabulator
\ooo	[ <b>octal</b> ]	Zeichen mit oktalem Wert 000–377
\xHH	[ <b>hexadecimal</b> ]	Zeichen mit hexadezimaalem Wert 00–FF

## 10.5 perl-Metazeichen

\A \Z		Zeilenanfang / -ende (bei mehreren Zeilen)
\b \B	<b>[break/no break]</b>	Wortgrenze/keine Wortgrenze
\s \S	<b>[space/no space]</b>	Leerraum / Kein Leerraum (Leerz., horiz. + vert. Tab. Zeilenvorschub, Wagenrücklauf, Seitenvorschub)
\d \D	<b>[digit/no digit]</b>	Ziffer (0–9) / Keine Ziffer
\w \W	<b>[word/no word]</b>	Buchstabe (a–zA–Z_0–9) / Kein Buchstabe

## 11 UNIX-Werkzeuge

### 11.1 strings, wc, head, tail, tee

strings FILE strings -a	In FILE nach allen ASCII-Textstücken suchen und sie ausgeben Alle Textstücke suchen (auch im Codesegment)
wc FILE wc -l FILE wc -w FILE wc -c FILE	Zeilen, Wörter und Zeichen in FILE zählen Nur Zeilen in FILE zählen <b>[lines]</b> Nur Wörter in FILE zählen <b>[words]</b> Nur Zeichen in FILE zählen <b>[characters]</b>
head FILE head -NN FILE	Die 10 ersten Zeilen von FILE ausgeben Die NN ersten Zeilen von FILE ausgeben
tail FILE tail -NN FILE tail +NN FILE tail -f FILE	Die 10 letzten Zeilen von FILE ausgeben Die NN letzten Zeilen von FILE ausgeben FILE ab der Zeile NN ausgeben Neue Zeilen am Ende von FILE permanent ausgeben <b>[follow]</b>
tee FILE tee -a FILE	stdin auf FILE schreiben und an stdout weitergeben stdin an FILE anhängen und an stdout weitergeben <b>[append]</b>

### 11.2 cmp, diff

cmp FILE1 FILE2	FILE1 und FILE2 binär vergleichen (Exit-Status)
diff FILE1 FILE2 diff -i ... diff -r ... diff -s ... diff -q ...	FILE1 und FILE2 zeilenw. vergl. und Unterschiede ausgeben Groß/Kleinschreibung ignorieren <b>[ignore case]</b> Dateien in Unterverz. ebenfalls vergleichen <b>[recursive]</b> Gleiche Dateien ebenfalls anzeigen <b>[same]</b> Nur Dateinamen, nicht die Unterschiede anzeigen <b>[quiet]</b>



### 11.3 cut, paste, join, tr, split

cut FILE cut -dC -fN-M  cut -cL,M,N cut -cNN-MM	Spalten oder Felder aus FILE schneiden (vertikal) Felder durch Trennzeichen C getrennt (Std: <TAB>) Felder N-M ausgeben [ <b>delimiter,fields</b> ] Spalten L, M und N ausgeben [ <b>columns</b> ] Spalten NN-MM ausgeben [ <b>columns</b> ]
paste FILE1 FILE2 paste -d":@" ... paste -s FILE paste -s - - -	Zeilen aus FILE1 und FILE2 nebeneinanderstellen Spaltentrenner :@; festlegen (\0=keiner) [ <b>delimiter</b> ] Alle Zeile von FILE in einer Zeile zusammenfassen [ <b>squeeze</b> ] Jeweils 3 Zeilen aus stdin in einer Zeile zusammenfassen
join FILE1 FILE2  join -tC ... join -1 M -2 N	Zeilen in FILE1 und FILE2 über Schlüsselspalte verknüpfen ( <i>beide Dateien müssen alphabetisch sortiert sein!</i> ) Feldtrenner C festlegen (Std: <SPACE>) [ <b>terminator</b> ] Feld M der 1. Datei mit Feld N der 2. verknüpfen
tr SET1 [SET2] tr -c SET1 tr -d SET1 tr -s SET1 SET2	Zeichen aus SET1 in korresp. aus SET2 umsetzen [ <b>translate</b> ] Zeichen nicht(!) in SET1 ersetzen [ <b>complement</b> ] Zeichen in SET1 löschen [ <b>delete</b> ] Gleiche Zeichen hintereinander zusammenziehen [ <b>squeeze</b> ]
split FILE [PREFIX] split -l NN ... split -b NN ...	FILE in Stücke xaa, xab, ... zerlegen (Std-Präfix: x) Stücke der Länge NN Zeilen erstellen (Std: 1000) [ <b>lines</b> ] Stücke der Länge NN Bytes erstellen [ <b>bytes</b> ]

tr liest nur von Standard-Eingabe und schreibt auf Standard-Ausgabe, eine Datei kann beim Aufruf nicht angegeben werden.

### 11.4 sort, uniq

sort FILE sort -n FILE sort -d FILE sort -f FILE sort -r FILE sort +N -M FILE sort -k N,M sort +0.N -0.M sort -tC FILE sort -oFILE FILE	Zeilen von FILE alphabetisch aufsteigend sortieren Numerisch sortieren [ <b>numeric</b> ] Lexikografisch sortieren (Std) [ <b>dictionary</b> ] Groß/Kleinschreibung ignorieren [ <b>fold</b> ] Sortierreihenfolge absteigend [ <b>reverse</b> ] Von Feld N+1 bis Feld M sortieren (N Felder überspringen) Von Feld N bis Feld M sortieren ( <i>neue Schreibweise!</i> ) [ <b>key</b> ] Von Spalte N+1 bis Spalte M sortieren Feldtrenner ist Zeichen C (Std: Tabulator) [ <b>terminator</b> ] Datei FILE direkt sortieren (ohne Kopie) [ <b>output</b> ]
uniq FILE uniq -c FILE uniq -d FILE uniq -u FILE	Doppelte Zeilen aus FILE entfernen ( <i>muss sortiert sein!</i> ) [ <b>unique</b> ] Vor Zeilen ihre Häufigkeit ausgeben [ <b>count</b> ] Nur mehrfach vorkommende Zeilen ausgeben [ <b>duplicates</b> ] Nur einfach vorkommende Zeilen ausgeben [ <b>uniques</b> ]

### 11.5 grep

grep "TEXT" FILE egrep "TEXT" FILE fgrep "TEXT" FILE	Zeilen mit Muster TEXT in FILE suchen (Standard-Regexp) Zeilen mit Muster TEXT in FILE suchen (Erweiterte-Regexp) Zeilen mit Text TEXT in FILE suchen (Feste Zeichenkette)
--	--

### 11.5.1 Die wichtigsten Optionen von grep

-c	Nur die Anzahl der passenden Zeilen ausgeben [ <b>count</b> ]
-h	Dateinamen nicht ausgeben (bei mehr als einer Datei) [ <b>hide/head</b> ]
-i	Groß/Kleinschreibung ignorieren [ <b>ignore case</b> ]
-l	Nur Dateinamen ausgeben (Muster paßt auf mind. eine Zeile) [ <b>list</b> ]
-n	Zeilennummer den passenden Zeilen voranstellen [ <b>number</b> ]
-v	Nach nicht(!) passenden Zeilen suchen [ <b>vice versa</b> ]

### 11.6 ed, ex, sed, awk, perl

cat SCRIPT   ed FILE	Ed-Kommandos SCRIPT auf FILE anwenden [ <b>editor</b> ]
cat SCRIPT   ex FILE	Ex-Kommandos SCRIPT auf FILE anw. [ <b>extended editor</b> ]
sed [-e] 'CMD' FILE	Sed-Kommandos CMD auf FILE anwenden [ <b>stream editor</b> ]
sed -f SCRIPT FILE	Sed-Kommandos aus SCRIPT auf FILE anwenden [ <b>file</b> ]
-n	Zeilen nicht automatisch ausgeben [ <b>noprint</b> ]
awk 'CMD' FILE	Awk-Befehle CMD auf FILE anw. [ <b>aho/weinberger/kernighan</b> ]
awk -f SCRIPT FILE	Awk-Kommandos aus SCRIPT auf FILE anwenden [ <b>file</b> ]
-F "TEXT"	Feldtrenner festlegen
-v VAR="VALUE"	Variable VAR mit Wert VALUE belegen
perl -e 'CMD;' FILE	Perl-Befehle CMD auf FILE anwenden [ <b>execute</b> ]
perl -f SCRIPT FILE	Perl-Kommandos aus SCRIPT auf FILE anwenden [ <b>file</b> ]
-a	\$_ automatisch in @F zerlegen [ <b>autosplit</b> ]
-n	while (<>) {...}-Schleife ohne Ausgabe [ <b>noprint</b> ]
-p	while (<>) {...}-Schleife mit Ausgabe [ <b>print</b> ]
-W	Alle Warnungen einschalten

### 11.6.1 Sed-Kommandos

a\ c\ i\ d	Nachfolgende Zeilen nach aktueller Zeile einfügen [ <b>append</b> ], alle Zeilen außer der letzten sind mit \ abzuschließen Aktuelle Zeile durch nachfolgende Zeilen ersetzen [ <b>change</b> ], alle Zeilen außer der letzten sind mit \ abzuschließen Nachfolgende Zeilen vor aktueller Zeile einfügen [ <b>insert</b> ], alle Zeilen außer der letzten sind mit \ abzuschließen Aktuelle Zeile löschen [ <b>delete</b> ]
s/REGEX/SUBST/ y/abc/ABC/	In aktueller Zeile REGEX durch SUBST ersetzen [ <b>substitute</b> ] In aktueller Zeile Zeichen a durch A, b durch B und c durch C ersetzen (analog dem Kommando tr) [ <b>yank</b> ]
n l p q =	Aktuelle Zeile ausgeben und die nächste Zeile einlesen [ <b>next</b> ] Aktuelle Zeile ausgeben (Steuer-Zeichen im ASCII-Code) [ <b>list</b> ] Aktuelle Zeile ausgeben [ <b>print</b> ] Aktuelle Zeile ausgeben und Sed-Skript abbrechen [ <b>quit</b> ] Nummer der aktuellen Zeile ausgeben
r FILE w FILE	Inhalt der Datei FILE nach aktueller Zeile einfügen [ <b>read</b> ] Aktuelle Zeile auf Datei FILE ausgeben [ <b>write</b> ]
b [LABEL] t [LABEL]  : LABEL {	Zu Marke LABEL (oder zum Skriptende) springen [ <b>branch</b> ] Zu Marke LABEL (oder zum Skriptende) springen, wenn seit dem letzten Einlesen oder seit dem letzten t-Kommando eine Ersetzung erfolgte [ <b>test</b> ] Marke für b- oder t-Kommando ( <i>max. 7 Zeichen!</i> ) [ <b>number</b> ] Kommandos bis zu } als Gruppe behandeln [ <b>group</b> ]
g h x	Aktuelle Zeile durch Zwischenpuffer ersetzen [ <b>get</b> ] Aktuelle Zeile in Zwischenpuffer kopieren [ <b>hold</b> ] Aktuelle Zeile und Zwischenpuffer vertauschen [ <b>exchange</b> ]
D G H N P	Aktuelle Zeile bis zum ersten Newline löschen [ <b>delete</b> ] Zwischenpuffer an aktuelle Zeile anhängen [ <b>get</b> ] Aktuelle Zeile an Zwischenpuffer anhängen [ <b>hold</b> ] Nächste Zeile an aktuelle Zeile anhängen [ <b>next</b> ] Akt. Zeile bis zum ersten Newline ausgeben (+ löschen) [ <b>print</b> ]

### 11.7 find, locate

find PATH COND ACTION	ACTION auf Dateien ab PATH mit Eigenschaft COND anwenden
locate "MUSTER"	Alle Dateien mit zu MUSTER passendem Namen suchen (LINUX)
updatedb	locate-Datenbank aktualisieren (LINUX)

### 11.7.1 find-Bedingungen

-group NAME/ID	Besitzergruppe
-mtime [+−]DAYS	Änderung vor mehr/weniger/genau DAYS 24h-Tagen [ <b>modification</b> ]
-name "PATTERN"	Dateinamen-Muster (* ? [...] ... , vor der Shell schützen)
-perm [−]RIGHTS	(Oktale) Rechte (NNN=exakt, −NNN=mindestens) [ <b>permission</b> ]
-size [+−]NUM[c]	Dateigröße mehr/weniger/genau NUM Blöcke (bzw. Zeichen)
-type TYPE	Dateityp (f=file, d=directory, c=character device, b=block device, l=symlink, p=named pipe, s=socket)
-user NAME/ID	Besitzer
-atime [+−]DAYS	Lesender Zugriff vor mehr/weniger/genau 24h-Tagen [ <b>access</b> ]
-ctime [+−]DAYS	Status-Änderung (Inode) vor mehr/... 24h-Tagen [ <b>change</b> ]
-iname "PATTERN"	Analog −name, aber Groß/Kleinschreibung egal [ <b>ignorecase</b> ]
-inum [+−]NUM	Inode-Nummer mehr/weniger/gleich NUM [ <b>inode number</b> ]
-links [+−]NUM	Link-Anzahl mehr/weniger/gleich NUM
-nogroup	Keiner Gruppe aus /etc/groups zugeordnet (ID ohne Name)
-nouser	Keinem Benutzer aus /etc/passwd zugeordnet (ID ohne Name)
\ (... \)	Klammerung (quotieren wegen Shell)
\!	Negation (quotieren wegen Shell)
−a	UND-Verknüpfung (Standard falls keine Verknüpfung angegeben)
−o	ODER-Verknüpfung

Die Tagesgrenze bei −a/c/mtime liegt auf dem Startzeitpunkt des find-Kommandos, der Tag 0 entspricht 0 bis 24 Stunden vorher, 1 entspricht 24-48h, ...

### 11.7.2 find-Aktionen

−print	Ausgabe der gefundenen Dateinamen ( <i>Standard!</i> )
−ls	Ausführliche Ausgabe der gefundenen Dateinamen (ls −lsdi)
−exec CMD {} \;	Kommando CMD auf allen gefundenen Dateien ausführen ({} = gef. Dateiname, \; = Kommandoende, quotiert wegen Shell)
−ok CMD {} \;	Analog −exec, verlangt aber vorher Bestätigung mit y(es)

### 11.7.3 find-Beispiele

find / −name "*.c" −print	C-Dateien ab Root-Verzeichnis
find .. −mtime 1 −print	Gestern modifizierte Dateien ab .. (Eltern-Verz.)
find / −mtime −7 −print	In der letzten Woche modifizierte Dateien
find / −mtime +100 −print	Vor mehr als 100 Tagen modifizierte Dateien
find / −user tsbirn −print	Dateien des Anwenders tsbirn
find /usr −type d −name "*man*"	Verzeichnisse in /usr mit man im Namen [ <b>directory</b> ]
find / −size 0 −ok rm {} \;	Leere Dateien löschen (mit Abfrage)
find / −type f \	Namen der Dateien die Text made enthalten [ <b>file</b> ]
−exec grep −l "made" {} \;	
find / −user root −perm −002	Gehören root und sind für andere schreibbar

## 11.8 tar, cpio

<pre>tar   tar cf ARCH DIR/FILE   tar xvf ARCH   tar tf ARCH   tar tzf ARCH</pre>	Archivdatei verwalten [ <b>tape archiver</b> ] ARCH erzeugen [ <b>create/file</b> ] ARCH in aktuellem Verz. auspacken [ <b>extract/verbose</b> ] ARCH mit Anzeige testen [ <b>test</b> ] Komprimiertes Archiv ARCH testen [ <b>test,ziped</b> ]
<pre>cpio   ...   cpio -o ARCH   cpio -id &lt; ARCH   cpio -tv &lt; ARCH</pre>	Archivdatei verwalten [ <b>copy input output</b> ] ARCH (Dateiliste von stdin) erz. auf stdout [ <b>output</b> ] ARCH ausp. (von stdin) mit Verz. erz. [ <b>input/directory</b> ] ARCH mit Anzeige testen (von stdin) [ <b>test/verbose</b> ]

## 11.9 compress, gzip

<pre>compress FILE uncompress FILE uncompress -c</pre>	FILE komprimieren und ersetzen (Endung .z, veraltet) FILE dekomprimieren und ersetzen Ausgabe auf stdout, nicht ersetzen [ <b>console</b> ]
<pre>gzip FILE gunzip FILE gunzip -c</pre>	FILE komprimieren und ersetzen (Endung .gz, Standard) FILE dekomprimieren und ersetzen Ausgabe auf stdout, nicht ersetzen [ <b>console</b> ]
<pre>bzip2 FILE bunzip2 FILE bunzip2 -c</pre>	FILE komprimieren und ersetzen (Endung .bz2, noch besser) FILE dekomprimieren und ersetzen Ausgabe auf stdout, nicht ersetzen [ <b>console</b> ]
<pre>zcat FILE... zmore FILE zcmp FILE1 FILE2 zdiff FILE1 FILE2 zgrep "TEXT" FILE</pre>	Komprimierte Dateien FILE... aneinanderhängen Komprimierte Datei FILE anzeigen Komprimierte Dateien FILE1 und FILE2 vergleichen (binär) Komprimierte Dateien FILE1 und FILE2 vergleichen (zeilenw.) Zeilen mit Muster TEXT in komprimiertem FILE suchen

## 12 System-Administration

### 12.1 fdisk, fdformat, mkfs, fsck, dd

fdisk DEV fdisk -l DEV	Gerät DEV menügesteuert partitionieren <b>[format disk]</b> Alle Partitionen von Gerät DEV auflisten <b>[list]</b>
fdformat DEV  fdformat /dev/fd0u1440 fdformat /dev/fd0u1722	Floppy-Disk DEV formatieren <b>[floppy disk format]</b> (in Spuren + Sektoren einteilen) 1,44 MByte-Diskette formatieren (80 Sp, 18 Sekt) 1,72 MByte-Diskette formatieren (82 Sp, 21 Sekt)
mkfs -t TYPE DEV mkfs -t msdos /dev/fd0 mkfs -t ext2 /dev/hda1	Dateisystem TYPE auf Gerät DEV anlegen <b>[make filesystem]</b> MSDOS-Dateisystem auf Diskette anlegen Ext2-Dateisystem auf 1. Partition der 1. Platte anlegen
fsck DEV fsck /dev/hdb4	Dateisystem überprüfen <b>[file system check]</b> 4. Partition der 2. Festplatte überprüfen
dd if=FILE of=FILE bs=NN count=NN skip=NN seek=NN	Sektoren lesen/schreiben (Platte/Floppy) <b>[disk duplicate]</b> Eingabedatei <b>[input file]</b> Ausgabedatei <b>[output file]</b> Blockgröße NN <b>[block size]</b> Anzahl Blöcke NN NN Blöcke der Eingabe überspringen NN Blöcke der Ausgabe unterdrücken

### 12.2 mount, umount

mount DEV PATH mount PATH mount DEV mount mount /floppy mount /cdrom mount -a mount -t TYPE	Gerät DEV in Pfad PATH des Dateibaumes einhängen <b>[montieren]</b> Gerät gemäß /etc/fstab in Pfad PATH einhängen Gerät DEV in Pfad gemäß /etc/fstab einhängen Alle montierten Geräte auflisten Diskettenlaufwerk automatisch einhängen (/etc/fstab!) CDROM automatisch montieren (/etc/fstab!) Alle Geräte in /etc/fstab einhängen <b>[all]</b> Alle Geräte vom Typ TYPE in /etc/fstab einhängen <b>[type]</b>
umount PATH umount DEV umount -a umount -t TYPE	Gerät mit Mount-Point PATH aushängen <b>[unmount]</b> Gerät DEV aushängen Alle Geräte in /etc/fstab aushängen (bis auf root-Verz) <b>[all]</b> Alle Geräte vom Typ TYPE in /etc/fstab aushängen <b>[type]</b>

### 12.2.1 Mount-Optionen

defaults	Steht für: rw, suid, dev, exec, auto, nouser, async
ro	read only einhängen (write-Recht ignorieren)
rw	read write einhängen (Zugriffsrecht write zählt)
(no) auto *	Während dem Bootvorgang (nicht) automatisch einhängen
(no) user *	Normale User dürfen (nicht) einhängen
(no) atime	Inode-Access-Zeit (nicht) updaten bei Lesezugriffen ( <i>langsam!</i> )
(no) dev *	Block- und Zeichen-Geräte-dateien (nicht) ignorieren
(no) exec *	Programme können (nicht) ausgeführt werden
(no) suid *	Set-UID-Programme können (nicht) ausgeführt werden
(a) sync *	Ein/Ausgabe synchron (sofort) bzw. asynchron (gepuffert)
remount	Eingehängte Partition erneut einhängen ( <i>zum Optionen ändern!</i> )

Die mit \* gekennzeichneten Optionen sind in `defaults` enthalten.

### 12.2.2 Samba-Mount-Optionen

username=TEXT	Benutzer-Name für Anmeldung
password=TEXT	Benutzer-Paßwort für Anmeldung
uid=NUM	User-ID festlegen
gid=NUM	Gruppen-ID festlegen
fmask=NUM	Auszumaskierende Rechte für Dateien
dmask=NUM	Auszumaskierende Rechte für Verzeichnisse
umask=NUM	Auszumaskierende Rechte festlegen

## 12.3 at, crontab

### 12.3.1 at-Befehle

at HH:MM	Auftrag um HH:MM starten (Eingabe auf <code>stdin</code> )
atq	Alle Aufträge anzeigen [ <b>at queue</b> ]
atrm JOBNR	Auftrag mit JOBNR löschen [ <b>at remove</b> ]

### 12.3.2 crontab-Befehle

crontab -e	Vorhandene crontab-Tabelle editieren [ <b>edit</b> ]
crontab -l	Vorhandene crontab-Tabelle anzeigen [ <b>list</b> ]
crontab -r	Vorhandene crontab-Tabelle löschen [ <b>remove</b> ]
crontab FILE	crontab-Tabelle aus FILE erstellen

#### 12.3.2.1 crontab-Felder

1	Minute	0-59	Kleinste Einheit  Oder jan, feb, ..., dec 0=Sonntag, ..., 6=Samstag oder sun, ... sat Ausgabe auf Datei umlenken (sonst erfolgt mail)
2	Stunde	0-59	
3	Tag	1-31	
4	Monat	1-12	
5	Wochentag	0-6	
6-	Kommando		

In den zentralen `crontab`-Dateien kommt als 6. Feld der Benutzername dazu, unter dem das Kommando ablaufen soll. Erst ab dem 7. Feld folgt dann das Kommando.

### 12.3.2.2 Erlaubte crontab-Feldwerte

*	Alle möglichen Werte
NN	Wert NN
N1, N2, ...	Werteliste N1, N2, ...
NN-MM	Wertebereich NN-MM
NN-MM/SS	Werte NN, NN+SS, NN+2*SS, ... (Schrittweite SS)

## 12.4 mail

mail	Empfangene Mails lesen
mail USER... < brief	Mail brief an die Benutzer USER... verschicken
mail -s "SUBJECT" USER	Mail mit Titel SUBJECT an USER verschicken (von stdin)

### 12.4.1 Die wichtigsten Mail-Befehle

h	Kopfzeilen der (letzten 10) Mails anzeigen [ <b>head</b> ]
NN	Mail Nummer NN anzeigen
d	Aktuelle Mail löschen [ <b>delete</b> ]
d NN-MM	Mails NN-MM löschen [ <b>delete</b> ]
q	Mailprogramm verlassen [ <b>quit</b> ]
x	Mailprogramm ohne jede Änderung verlassen [ <b>quit</b> ]
<RETURN>	Nächste Mail anzeigen



## 13 Shell-Programmierung

### 13.1 Shell-Aufrufarten

	PATH durchsucht	x-Recht nötig	Sub-Shell	Neuer Prozeß	Aliase+ Funkt. vererbt	Rückkehr zum Aufrufer	Option angebbbar
cmd.sh	ja	ja	ja	ja	nein	ja	nein
sh [OPT] cmd.sh	nein	nein	ja	ja	nein	ja	ja
./cmd.sh	nein	ja	ja	ja	nein	ja	nein
. cmd.sh	ja	nein	nein	nein	ja	ja	nein
. ./cmd.sh	nein	nein	nein	nein	ja	ja	nein
exec cmd.sh	ja	ja	nein	nein	nein	nein	nein
exec ./cmd.sh	nein	ja	nein	nein	nein	nein	nein

Das Kommando `.` heißt `source`-Kommando, es liest die angegebene Datei in der aktuellen Shell ein (include-Anweisung).

### 13.2 Shell-Optionen

-n	Kommandos lesen aber nicht ausführen (Syntaxcheck) [ <b>noexec</b> ]
-u	Abbruch bei Zugriff auf nicht initialisierte Variable [ <b>unset</b> ]
-v	Gelesene Kommandos ausgeben [ <b>verbose</b> ]
-x	Wirklich ausgeführte Kommandos vorher ausgeben [ <b>execute</b> ]
-o OPT	Option OPT setzen
+o OPT	Option OPT zurücksetzen
-o ignoreeof	<Strg-D> ignorieren (d.h. exit notwendig)
-o noclobber	Umlenkung auf bereits existierende Dateien verhindern (> )

### 13.3 Spezielle Shell-Variablen (Parameter)

\$1	1. Kommandozeilen-Argument
\$2	2. Kommandozeilen-Argument
...	...
\$9	9. Kommandozeilen-Argument (Zugriff auf \${10} ... mit shift)
\$#	Anzahl Kommandozeilen-Argumente
\$*	Alle Kommandozeilen-Argumente
\$@	Alle Kommandozeilen-Argumente (einzeln quotiert bei "\$@")
\$0	Skript-Name (Aufrufform oder kompletter Pfad)
\$?	Exit-Code des letzten ausgeführten Kommandos
\$\$	Prozeß-ID des Shell-Skripts
\$_	Prozeß-ID des letzten im Hintergrund gestarteten Kommandos (CMD &)
\$-	Shell-Optionen

### 13.3.1 Bedingte Auswertung von Shell-Variablen

<code>\${VAR-TEXT}</code>	VAR zurück falls VAR definiert; sonst TEXT zurück
<code>\${VAR=TEXT}</code>	Analog + Zuweisung von TEXT an VAR
<code>\${VAR+TEXT}</code>	TEXT zurück falls VAR definiert; sonst nichts zurück
<code>\${VAR?}</code>	Ausgabe VAR: parameter not set + Abbruch falls VAR nicht definiert
<code>\${VAR?TEXT}</code>	Analog Ausgabe von VAR: TEXT; sonst VAR zurück

Ein Doppelpunkt : nach VAR verlangt, dass die Variable VAR nicht leer sein darf (`not null`). Ohne Doppelpunkt darf sie leer sein, muss aber definiert sein.

### 13.4 Bedingungen des test-Kommandos

Das Zeichen **K** heißt, der entsprechende Test ist nur in der `ksh` und in der `bash` vorhanden. Das Zeichen **B** heißt, der entsprechende Test ist nur in der `bash` vorhanden.

#### 13.4.1 Textvergleiche

<code>"TEXT"</code>	TEXT ist nicht leer ( <i>nicht mit ! -a -o kombinierbar</i> )
<code>-n "TEXT"</code>	TEXT ist nicht leer ( <i>mit ! -a -o kombinierbar</i> ) [ <b>nonzero</b> ]
<code>-z "TEXT"</code>	TEXT ist leer [ <b>zero</b> ]
<code>"TEXT1" = "TEXT2"</code>	TEXT1 und TEXT2 sind gleich
<code>"TEXT1" != "TEXT2"</code>	TEXT1 und TEXT2 sind verschieden
<code>"TEXT1" \&lt; "TEXT2" K</code>	TEXT1 alphabetisch kleiner als TEXT2 (quot. wg. Shell)
<code>"TEXT1" \&gt; "TEXT2" K</code>	TEXT1 alphabetisch größer als TEXT2 (quot. wg. Shell)

#### 13.4.2 Dateiattribut-Vergleiche

<code>-e FILE</code>	<b>K</b>	Datei FILE existiert [ <b>exists</b> ]
<code>-d FILE</code>		Datei FILE ist Verzeichnis [ <b>directory</b> ]
<code>-f FILE</code>		Datei FILE ist normale Datei [ <b>file</b> ]
<code>-r FILE</code>		Datei FILE ist lesbar [ <b>readable</b> ]
<code>-s FILE</code>		Datei FILE nicht leer [ <b>size</b> ]
<code>-w FILE</code>		Datei FILE schreibbar [ <b>writable</b> ]
<code>-x FILE</code>		Datei FILE ausführbar [ <b>executable</b> ]
<code>-b FILE</code>		Datei FILE ist blockorientiert [ <b>block device</b> ]
<code>-c FILE</code>		Datei FILE ist zeichenorientiert [ <b>character device</b> ]
<code>-L FILE</code>	<b>K</b>	Datei FILE ist ein symbolischer Link [ <b>link</b> ]
<code>-p FILE</code>		Datei FILE ist eine Named Pipe [ <b>pipe</b> ]
<code>-S FILE</code>	<b>K</b>	Datei FILE ist ein Socket [ <b>socket</b> ]
<code>-t DESC</code>		Deskriptor DESC ist ein Terminal (Std: 0=stdin) [ <b>terminal</b> ]
<code>-g FILE</code>		Datei FILE hat group id Recht gesetzt [ <b>group</b> ]
<code>-k FILE</code>		Datei FILE hat sticky Recht gesetzt [ <b>sticky</b> ]
<code>-u FILE</code>		Datei FILE hat user id Recht gesetzt [ <b>user</b> ]
<code>-G FILE</code>	<b>B</b>	Datei FILE gehört effektiver group id [ <b>Group</b> ]
<code>-N FILE</code>	<b>B</b>	Datei FILE seit letztem Lesen verändert [ <b>New</b> ]
<code>-O FILE</code>	<b>B</b>	Datei FILE gehört effektiver user id [ <b>Owner</b> ]
<code>-o OPTION</code>	<b>B</b>	Option OPTION eingeschalten

### 13.4.3 Datei-Vergleiche

FILE1 -nt FILE2	K	Datei FILE1 neuer als FILE2 [ <b>newer than</b> ]
FILE1 -ot FILE2	K	Datei FILE1 älter als FILE2 [ <b>older than</b> ]
FILE1 -ef FILE2	K	Dateien haben ident. device/inode-Nummer [ <b>equal file</b> ]

### 13.4.4 Numerische Vergleiche

NUM1 -eq NUM2	Zahl NUM1 gleich NUM2 [ <b>equal</b> ]
NUM1 -ne NUM2	Zahl NUM1 nicht gleich NUM2 [ <b>not equal</b> ]
NUM1 -le NUM2	Zahl NUM1 kleiner gleich NUM2 [ <b>less equal</b> ]
NUM1 -lt NUM2	Zahl NUM1 kleiner NUM2 [ <b>less than</b> ]
NUM1 -ge NUM2	Zahl NUM1 größer gleich NUM2 [ <b>greater equal</b> ]
NUM1 -gt NUM2	Zahl NUM1 größer NUM2 [ <b>greater than</b> ]

### 13.4.5 Logische Verknüpfungen

\( ... \)	Klammerung ( <i>quotiert wg. Shell!</i> )
\! EXPR	Negation von EXPR ( <i>quotiert wg. Shell!</i> ) [ <b>not</b> ]
EXPR1 -a EXPR2	EXPR1 und EXPR2 [ <b>and</b> ]
EXPR1 -o EXPR2	EXPR1 oder EXPR2 [ <b>or</b> ]

Die Reihenfolge von oben nach unten entspricht dem Vorrang.

### 13.4.6 Shell-Optionen testen

-o OPT	K	Option OPT gesetzt [ <b>option</b> ]
--------	---	--------------------------------------

## 13.5 Kontrollstrukturen

### 13.5.1 Benutzereingabe einlesen

echo -n "Bitte eingeben: "	-n=kein Zeilenvorschub
read EINGABE	Benutzereingabe (bis RETURN)
echo "Eingegeben wurde: \$EINGABE"	(Testausgabe)

### 13.5.2 Verzweigung

<pre>if [ \$ZAHL -gt 2 ] then     echo "\$ZAHL ist größer als 2" elif [ \$ZAHL -lt 2 ] then     echo "\$ZAHL ist kleiner als 2" else     echo "\$ZAHL ist gleich 2" fi</pre>	<p>Leerzeichen um [, ] und -gt then muss auf neuer Zeile stehen (Testausgabe) else if/elsif geht nicht then muss auf neuer Zeile stehen (Testausgabe) else-Zweig (alle sonstigen Fälle) (Testausgabe) fi nicht vergessen (<i>neue Zeile!</i>)</p>
--	---

### 13.5.3 case-Mehrfachverzweigung (Text-Vergleich mit Shell-Metazeichen)

<pre>TEXT="Babel" case "\$TEXT" in     yes ja) echo "ja gefunden" ;;     n*)     echo "nein gefunden" ;;     *)      echo "Fehler" ;; esac</pre>	<p>Zum Testen Zeilen vertauschen in und ... nicht vergessen! ;; nicht vergessen! ;; nicht vergessen! else-Zweig (alle sonstige Fälle) esac nicht vergessen (<i>neue Zeile!</i>)</p>
--	---

### 13.5.4 for-Schleife (Liste bzw. Shell-Aufrufargumente abarbeiten)

<pre>for ELEM in abc defghi jklmnopqr do     echo "for: ELEM=\$ELEM" done</pre>	<p>ELEM enthält Werte aus Liste do muss auf neuer Zeile stehen (Testausgabe) done nicht vergessen (<i>neue Zeile!</i>)</p>
<pre>for ARG do     echo "for: ARG=\$ARG" done</pre>	<p>ELEM enthält Shell-Argumente do muss auf neuer Zeile stehen (Testausgabe) done nicht vergessen (<i>neue Zeile!</i>)</p>

### 13.5.5 while und until-Schleife (Bedingung prüfen)

<pre>ZAHL=3 while [ \$ZAHL -gt 0 ] do     echo "while: ZAHL=\$ZAHL"     ZAHL=`expr \$ZAHL - 1` done</pre>	<p>ZAHL initialisieren Durchlauf solange ZAHL größer 0 do muss auf neuer Zeile stehen (Testausgabe) Kommando-Subst (Leerz. um - nötig) done nicht vergessen (<i>neue Zeile!</i>)</p>
<pre>until [ \$ZAHL -eq 3 ] do     echo "until: ZAHL=\$ZAHL"     ZAHL=`expr \$ZAHL + 1` done</pre>	<p>Durchlauf bis ZAHL Wert 3 hat do muss auf neuer Zeile stehen (Testausgabe) Kommando-Subst (Leerz. um + nötig) done nicht vergessen (<i>neue Zeile!</i>)</p>

### 13.5.6 Funktions-Definition und -Aufruf

<pre>func1() {     echo "Funktion func1() aufgerufen"     echo "1.Parameter \$1"     echo "2.Parameter \$2" }</pre>	Definition ( <i>vor Aufruf!</i> ) Extra Zeile! Einrücken sinnvoll Zugriff auf 1.Parameter Zugriff auf 2.Parameter Extra Zeile!
<pre>func1 func1 100 Test</pre>	Aufruf ( <i>nach Definition!</i> ) Aufruf mit Parametern

## 14 Awk

### 14.1 AWK-Optionen

Option	Bedeutung
<code>-F <i>seplere</i></code>	Feldtrenner <i>FS</i> auf <i>seplere</i> festlegen ( <b>Default:</b> " ")
<code>-f <i>file</i></code>	Programmdatei <i>file</i> einlesen ( <i>mehrfach erlaubt</i> )
<code>-v <i>var=val</i></code>	Variable <i>var</i> den Wert <i>val</i> vor Programmstart zuweisen
<code>-mf <i>n</i></code>	Maximale Anzahl Felder <i>n</i> festlegen ( <i>im Gawk überflüssig</i> )
<code>-mr <i>n</i></code>	Maximale Recordlänge <i>n</i> festlegen ( <i>im Gawk überflüssig</i> )
<code>--</code>	Beendet Optionen-Liste (erlaubt Argumente mit führendem -)

### 14.2 GNU-AWK-spezifische Optionen

Gawk-Option	Bedeutung
--file/-f= <i>file</i> --source='code'	Programmdatei <i>file</i> einlesen ( <i>mehrfach erlaubt</i> ) Programmtext <i>code</i> auf der Kommandozeile angeben ( <i>mehrfach erlaubt, in Hochkomma setzen</i> )
--exec/-e= <i>file</i> --include/-i= <i>lib</i> --load/-l= <i>lib</i>	Analog --file/-f, aber letzte verarbeitete Option (für CGI-Skripte) Lädt eine Quellcode-Library <i>lib</i> (sucht gemäß AWKPATH) Lädt eine Shared-Library <i>lib</i> (sucht gemäß AWKLIBPATH)
--field-separator/-F= <i>sep/ere</i> --assign/-v <i>var=val</i>	Feldtrenner FS auf <i>sep/ere</i> festlegen ( <b>Default:</b> " ") Variable <i>var</i> den Wert <i>val</i> vor Programmstart zuweisen
--traditional/-c --posix/-P	Gawk-Erweiterungen abschalten Nur POSIX-Umfang zulassen (siehe unten).
--copyright/-C --help/-h --version/-V	GPL (GNU General Public Licence) ausgeben Gawk-Usage-Meldung ausgeben Gawk-Versionsnummer ausgeben
--lint/-L[= <i>wert</i> ]/  --lint-old/-t	Warnung bei gefährlichen/nicht portablen Konstrukten ( <i>wert=fatal</i> führt zu fatalem Fehler statt Warnung) ( <i>wert=invalid</i> : nur Warnung bei wirklich ungültigen Dingen) ( <i>wert=no-ext</i> : TODO) Warnung bei Konstrukten abweichend vom Ur-Awk
--dump-variables/-d[= <i>file</i> ] --gen-pot/-g --profile/-p[= <i>file</i> ] --trace/-I	Sortierte Liste globaler Var. mit Typ + Endwert ausgeben Aus lokalisierbaren Strings .po-Datei erzeugen ( <i>gentext</i> ) Profildaten in Datei <i>awkprof.out</i> bzw. <i>file</i> ablegen ( <i>pgawk</i> ) Gibt während Ausführung die Bytecode-Namen aus (mit + davor)
--csv/-k --characters-as-bytes/-b --re-interval/-r	CSV Verarbeitung aktivieren (Comma Separated Values) Eingabedaten als Einzelbyte-Zeichen betrachten { <i>n,m</i> }-Wiederholung in Regulären Ausdrücken erlauben ( <b>Default:</b> aktiv)
--optimize/-O --no-optimize/-s --pretty-print/-o= <i>file</i> --sandbox/-S	Default-Optimierungen aktivieren ( <b>Default:</b> an) Default-Optimierungen abschalten Formatierten Code auf <i>file</i> ausgeben ( <b>Default:</b> <i>awkprof.out</i> ) Im Sandbox-Modus laufen ( <i>system</i> , Umlenkung, Pipes inaktiv)
--bignum/-M --non-decimal-data/-n --use-lc-numeric/-N	Beliebig genaue Fließkommazahlen aktivieren (GNU MPFR, GMP) Oktal/Hexadezimalwerte in Eingabedaten erkennen ( <i>Vorsicht!</i> ) Dezimaltrennzeichen gemäß Locale-Einstellung beim Einlesen

## 14.3 Konstanten

### 14.3.1 Zahlen

```

123      -1      -3.141592   +.0125   1e12   -987.654E-321
0123     0777     07654321           # Oktale Werte
0xFF     0xAFFE    0X1234567890ABCDEF # Hexadezimale Werte
0xff     0xAffe    0x1234567890abcdef # Hexadezimale Werte

```

### 14.3.2 Zeichenketten

```

"abc"           # Ohne Zeilenvorschub
"Dies ist ein Text.\n" # Mit Zeilenvorschub
""              # Leere Zeichenkette

```

### 14.3.3 Escape-Sequenzen

Gawk	Escape	Bedeutung
*	\a	Akustisches Signal ( <b>alert</b> )
	\b	Backspace (Zeichen zurück)
	\f	Seitenvorschub ( <b>formfeed</b> )
	\n	Zeilenvorschub ( <b>newline</b> )
	\r	Wagenrücklauf ( <b>carriage return</b> )
	\t	Tabulator (horizontal)
*	\v	Vertikaler Tabulator
	\ddd	Zeichen mit oktalem Wert <i>ddd</i> (Zahlen zwischen 000 und 377)
*	\xdd	Zeichen mit hexadezimalen Wert <i>dd</i> (Zahlen zwischen 00 und ff/FF)
	\"	Anführungszeichen
	\\	Backslash
	\/	Slash

### 14.3.4 Reguläre Ausdrücke

```

/abc/          # Enthält "abc"
/^abc$/        # Enthält exakt "abc"
/(abc|def)+/   # Enthält "abc" oder "def" mind. 1x nacheinander
/[\t]+/        # Enthält Leerzeichen oder Tabulator
/\\r$/         # Enthält Carriage Return am Zeilenende

```

### 14.3.5 Reguläre Ausdrücke

```

"abc"          # Enthält "abc"
"^abc$"        # Enthält exakt "abc"
"(abc|def)+"   # Enthält "abc" oder "def" mind. 1x nacheinander
"[ \\t]+"      # Enthält Leerzeichen oder Tabulator
"\\r$"         # Enthält Carriage Return am Zeilenende

```

#### 14.3.5.1 Metazeichen in Regulären Ausdrücken

Gawk	Metazeichen	Bedeutung
	$(r)$	<b>Gruppierung:</b> Jede Zeichenkette, auf die $r$ paßt (höchster Vorrang)
	$c$ $\backslash c$ $\wedge$ $\$$ $\cdot$ $[abc]$ $[a-z]$ $[\wedge abc]$ $[\wedge a-z]$	Zeichen $c$ (kein Metazeichen) Escape-Sequenz oder (Meta)Zeichen $c$ wörtlich Zeichenketten/Zeilen-Anfang Zeichenketten/Zeilen-Ende 1 beliebiges Zeichen <b>Zeichenklasse:</b> 1 Zeichen aus Menge $abc$ bzw. $a-z$ <b>Invertierte Zeichenklasse:</b> 1 Zeichen <b>nicht</b> aus Menge $abc$ bzw. $a-z$
*	$\backslash y$ $\backslash B$	Wortgrenze / Wortinneres ( <b>Break</b> )
*	$\backslash <$ $\backslash >$	Wortanfang / Wortende
*	$[[:<:]]$ $[[:>:]]$	Wortanfang / Wortende
*	$\backslash w$ $\backslash W$	1 Wortzeichen ( $[A-Za-z_0-9]$ ) / kein Wortzeichen ( $[\wedge A-Za-z_0-9]$ )
*	$\backslash ' \backslash '$	Zeichenketten/Zeilen-Anfang/Ende (alternative Form)
	$r^*$ $r^+$ $r^?$	$0 - \infty$ aufeinanderfolgende Zeichenketten, auf die $r$ paßt ( <b>Closure</b> ) $1 - \infty$ aufeinanderfolgende Zeichenketten, auf die $r$ paßt ( <b>Positive Closure</b> ) $0/1$ Zeichenketten, auf die $r$ paßt ( <b>Option</b> )
*	$r\{m, n\}$	$m - n$ Wiederholungen der Zeichenkette, auf die $r$ paßt
*	$r\{m, \}$	$m - \infty$ Wiederholungen der Zeichenkette, auf die $r$ paßt
*	$r\{m\}$	$m$ Wiederholungen der Zeichenkette, auf die $r$ paßt (genau)
	$r_1 r_2$ $r_1   r_2$	<b>Verkettung:</b> Jede Zeichenkette $xy$ , wo $r_1$ zu $x$ <b>und</b> $r_2$ zu $y$ paßt <b>Alternative:</b> Jede Zeichenkette, die zu $r_1$ <b>oder</b> $r_2$ paßt (niedrigster Vorrang)

#### 14.3.5.2 POSIX-Zeichenklassen ( $[[:class:]]$ )

Gawk	Klasse	Bedeutung
*	alnum	Alphanumerische Zeichen (Buchstaben + Ziffern)
*	alpha	Buchstaben
*	blank	Leerzeichen oder Tabulator
*	cntrl	Control-Zeichen
*	digit	Dezimalziffern
*	graph	Alle druckbaren und sichtbaren Zeichen
*	lower	Kleine Buchstaben
*	print	Druckbare Zeichen (keine Kontroll-Zeichen)
*	punct	Satzzeichen
*	space	Whitespace (Leerzeichen, Tabulator, Zeilenvorschub, ...)
*	upper	Große Buchstaben
*	xdigit	Hexadezimalziffern



### 14.3.6 Operatoren

!C	Operator	Bedeutung
*	(...) \$n ++ --	Klammerung/Gruppierung <b>Feld-Zugriff</b> auf Feld $n$ (\$1, ..., \$NF) Inkrementieren, Dekrementieren ( <i>Präfix</i> und <i>Postfix</i> )
*r	^ ** + - ! * / % + -	Potenzierung $x^y$ (** nur <i>Gawk</i> ) <i>Unäres</i> Plus/Minus (Vorzeichen), logisch NICHT Multiplikation, Division, Modulo (Divisionsrest) Addition, Subtraktion
*	(space)	<b>Verkettung</b> von Zeichenket. ( <i>kein expliziter Operator!</i> )
*	&	Ein/Ausgabe per Pipe ( <i>getline</i> , <i>print</i> und <i>printf</i> )
	< > <= >=	Vergleich ( <i>Zahlen und Zeichenketten</i> ) Vergleich ( <i>Zahlen und Zeichenketten</i> )
	== !=	Vergleich ( <i>Zahlen und Zeichenketten</i> )
*	~ !~	Vergleich mit <b>Reg. Ausdruck</b> , negiert ( <i>matched by</i> )
*	in	Test ob <b>Index/Schlüssel</b> in Array vorhanden
	&&	Logisch UND ( <i>short-cut evaluation</i> )
		Logisch ODER ( <i>short-cut evaluation</i> )
r	?:	Bedingter Ausdruck ( <i>cond ? true-case : false-case</i> )
r	= += -= *= /= %= ^= **=	Zuweisung (inkl. Operation + - * / % ^ **)

### 14.3.7 Ausdrücke

```

1 + (2 * exp(3) + sqrt(100)) ^ 2      # Numerisch
substr($0, 1, 4) "String"             # Zeichenkette
i <= 10                                # Logisch
user !~ /Dieter/ && (age >= 18 || perm == "yes") # Logisch

```

## 14.4 Programmaufbau

### 14.4.1 Regel

```
MUSTER { AKTION }
```

#### 14.4.1.1 Muster

Muster	Bedeutung
BEGIN END	Preprocessing <i>vor</i> dem Einlesen aller Dateien Postprocessing <i>nach</i> dem Einlesen aller Dateien
BEGINFILE ENDFILE	Preprocessing <i>vor</i> dem Einlesen einer Datei Postprocessing <i>nach</i> dem Einlesen einer Datei
Ausdruck /REGEXP/	Logische Bedingung Matcht Eingabezeile
Ausdruck, Ausdruck /REGEXP/, /REGEXP/	Bereichsmuster auf Basis logischer Bedingungen Bereichsmuster auf Basis Regulärer Ausdrücke

### 14.4.1.2 Aktionen

!C	Anweisung
	break continue do <i>Anweisung</i> while ( <i>Ausdruck</i> ) exit [ <i>Ausdruck</i> ] for ( <i>Ausdruck1</i> ; <i>Ausdruck2</i> ; <i>Ausdruck3</i> ) <i>Anweisung</i> * for ( <i>Variable</i> in <i>Array</i> ) <i>Anweisung</i> if ( <i>Ausdruck</i> ) <i>Anweisung1</i> [else <i>Anweisung2</i> ] * next * nextfile (Gawk) return [ <i>Ausdruck</i> ] while ( <i>Ausdruck</i> ) <i>Anweisung</i>
*	<i>Ein/Ausgabe-Anweisung</i> (getline, print und printf) delete <i>Arrayelement/Array</i> <i>Ausdruck</i> (mit Konstanten, Variablen, Zuweisungen, Funktionsaufrufen, ...) <i>Funktion</i> ( <i>Argumente</i> ) { <i>Anweisung</i> ; ... } (Block)
*	switch ( <i>Ausdruck</i> ) { case <i>Wert Regex</i> : <i>Anweisung</i> ... default: <i>Anweisung</i> }

## 14.4.2 Funktionen

### 14.4.2.1 Definition

```
function FUNCNAME(PARAM1, PARAM2, ...)
{
    Anweisung1
    Anweisung2
    ...
    return RESULT
}
```

### 14.4.2.2 Aufruf

```
FUNCNAME(EXPR1, EXPR2, ...)
```

### 14.4.3 Zeilenumbrücken sind erlaubt nach

Token	Bedeutung
,	Komma (in <code>print</code> , <code>printf</code> , Funktionsdefinition und -aufruf)
{	Linke geschweifte Klammer (Blockbeginn)
?	Bedingte Operation (nur <i>Gawk</i> )
:	Bedingte Operation (nur <i>Gawk</i> )
&&	Logisch Und
	Logisch Oder
if (...)	Bedingung
else	Alternative
for (...)	Zählschleife
while (...)	Abweisende Schleife
do	Nicht abweisende Schleife
}	Rechte Klammer in einer <code>if</code> -, <code>for</code> - oder <code>while</code> -Anweisung

### 14.4.4 Felder, Variablen und Arrays

#### 14.4.4.1 Defaultwerte für Feld- und Satztrenner

Variable	Default	Bedeutung
FS	<code>\t</code>	<b>Whitespace</b> = beliebig lange Folge von Leerzeichen und/oder Tabulatoren; am Zeilenanfang/ende vorkommender Whitespace wird ignoriert [ <b>field separator</b> ]
RS	<code>\n</code>	Zeilenvorschub [ <b>record separator</b> ]

#### 14.4.4.2 Felder

Variable	Bedeutung
<code>\$0</code>	Aktueller Eingabesatz
<code>\$n</code>	Feld <i>n</i> des aktuellen Eingabesatzes ( <code>\$1..\$NF</code> )

#### 14.4.4.3 Variablen

```
v  v123  eine_variable  EineVariable  _var_  _VAR_  _Var_
```

#### 14.4.4.4 Arrays

```
arr[123]    = "Text1"
arr["abc"]  = 789
```

```
i = "A"; j = "B"; k = "C"
arr[i,j,k] = "hello world\n";  # Element "A\034B\034C" belegen
```

```

arr[i]
arr[i,j,k]

for (i in arr) ...      # OK
for (i,j,k in arr) ...  # Geht nicht!

if (i in arr) ...
if ((i,j,k) in arr) ... # OK

delete arr[i]
delete arr[i,j,k]

delete arr

```

#### 14.4.4.5 Vergleichstypen

	str	strnum	num
str	str	str	str
strnum	str	num	num
num	str	num	num

#### 14.4.5 Vordefinierte Variablen und Arrays

Gawk	Variable	Bedeutung
*	ARGC ARGIND ARGV	Anzahl Kommandozeilen-Argumente (Programmname in ARGV[0]) Index der aktuell eingelesenen Eingabedatei in ARGV[1..ARGC-1] Array der Kommandozeilen-Argumente (ARGV[0..ARGC-1])
*	BINMODE	Binäre Ein/Ausgabe für Dateien (1/r = nur Eingabe-Dateien, 2/w = nur Ausgabe-Dateien, 3/rw = Ein+Ausgabe-Dateien)
*	CONVFMT	Format für automatische Umwandlung Zahl → Zeichenkette ( <b>Default:</b> "%.6g")
*	ENVIRON	Assoz. Array mit Umgebungsvar. und ihrem Wert ( <b>Environment</b> )
*	ERRNO	Systemfehlermeldung bei <code>getline</code> und <code>close</code>
*	FIELDWIDTHS FILENAME	Liste mit durch Leerz. getrennten festen Feldbreiten (statt FS benutzt) Name der aktuellen Eingabedatei
*	FNR FPAT FS FUNCTAB	Satznummer der aktuellen Eingabedatei Regex der Feldinhalte beschreibt (statt FS benutzt) Eingabefeldtrenner ( <b>Default:</b> "_" = Whitespace) Assoz. Array mit Funktionsnamen als Index und Funktionen als Wert
*	IGNORECASE	Groß/Kleinschreibung beim Vergleich mit Regulärem Ausdruck und in Zeichenketten-Funktionen ignorieren ( <b>Default:</b> false)
*	LINT	Schalter <code>--lint</code> dynamisch steuern (true=Lint Warnungen, false=keine, fatal=Abbruch bei Lint Warnung)
	NF NR	Anzahl Felder im aktuellen Satz Anzahl bisher eingelesener Sätze insgesamt
	OFMT OFS ORS	Ausgabeformat für Zahlen in <code>print</code> ( <b>Default:</b> "%.6g") Ausgabe-Feldtrenner in <code>print</code> ( <b>Default:</b> "_") Ausgabe-Satztrenner in <code>print</code> ( <b>Default:</b> "\n")
*	PREC PROCINFO ROUNDMODE	Fließkomma-Genauigkeit ( <b>Default:</b> 53 Stellen) Assoz. Array mit Informationen zum Awk-Prozeß (gid, uid, ...) Rundungsmodus ( <b>Default:</b> N = IEEE-754 roundTiesToEven mode)
	RLENGTH RSTART	Länge des passenden Textes nach <code>match</code> (-1 falls nicht gefunden) Beginn des passenden Textes nach <code>match</code> (0 falls nicht gefunden)
*	RS RT	Eingabesatztrenner ( <b>Default:</b> "\n") Zu RS passender Text nach dem Einlesen jedes Satzes
*	SUBSEP SYMTAB TEXTDOMAIN	Trennzeichen für mehrdim. Index [ <i>i, j, ...</i> ] ( <b>Default:</b> "\034") Assoz. Array mit Namen aller globalen Variablen/Arrays als Index Textdomäne des Programms für Übersetzung von Texten

#### 14.4.6 Vordefinierte Funktionen

##### 14.4.6.1 Arithmetik-Funktionen

Gawk	Funktion	Bedeutung
	<code>sin(x)</code> <code>cos(x)</code> <code>atan2(y, x)</code>	Sinus von $x$ ( $x$ in Radiant) Cosinus von $x$ ( $x$ in Radiant) Arcustangens von $y/x$ im Bereich $-\pi$ bis $\pi$ (in Radiant)
	<code>exp(x)</code> <code>log(x)</code>	Exponentialfunktion $e^x$ von $x$ Natürlicher Logarithmus (Basis $e$ ) von $x$
	<code>int(x)</code> <code>sqrt(x)</code>	Ganzzahliger Teil von $x$ Quadratwurzel von $x$
	<code>rand()</code> <code>srand([x])</code>	Zufallszahl $x$ , mit $0 \leq x < 1$ $x$ ist neuer Startwert von <code>rand</code> , gibt alten Startwert zurück (verwendet Uhrzeit ohne $x$ )

#### 14.4.6.2 Bit-Funktionen

Gawk	Funktion	Bedeutung
*	<code>and(a, b)</code>	<i>a</i> und <i>b</i> bitweise UND-verknüpfen
*	<code>or(a, b)</code>	<i>a</i> und <i>b</i> bitweise ODER-verknüpfen
*	<code>xor(a, b)</code>	<i>a</i> und <i>b</i> bitweise XOR-verknüpfen
*	<code>compl(a)</code>	<i>a</i> bitweise invertieren ( <b>Komplement</b> )
*	<code>lshift(a, n)</code>	<i>a</i> um <i>n</i> Bits nach links schieben
*	<code>rshift(a, n)</code>	<i>a</i> um <i>n</i> Bits nach rechts schieben

#### 14.4.6.3 Zeichenketten-Funktionen

Gawk	Funktion	Bedeutung
*	<code>gensub(r, s[, h[, t]])</code>	Ersetzt <i>r</i> in <i>t</i> durch <i>s</i> , gibt Ergebnis zurück; <i>h</i> ="g/G"(global) oder Zahl(=Position), sonst nur 1. Position; <i>t</i> bleibt unverändert, ohne <i>t</i> wird \$0 verwendet; ( ) ist in <i>r</i> und \1-\9 ist in <i>s</i> verwendbar
	<code>gsub(r, s[, t])</code>	Ersetzt <i>r</i> überall in <i>t</i> durch <i>s</i> ; gibt Anzahl Ersetzungen zurück ohne <i>t</i> wird \$0 verwendet <b>[global]</b>
	<code>sub(r, s[, t])</code>	Wie <code>gsub</code> , ersetzt aber nur die erste Teilkette
*	<code>length([s])</code> <code>length(a)</code>	Gibt Länge von <i>s</i> zurück, ohne <i>s</i> wird \$0 verwendet; Anzahl Elemente des Arrays <i>a</i> ermitteln
	<code>index(s, t)</code>  <code>match(s, r[, a])</code>	Gibt erste Position (1.. <i>n</i> ) von <i>t</i> in <i>s</i> zurück; oder 0 falls <i>t</i> nicht in <i>s</i> enthalten ist Überprüft, ob <i>s</i> eine zu <i>r</i> passende Teilkette enthält; gibt Position (1.. <i>n</i> ) oder 0 (kein Treffer) zurück, falls <i>r</i> nicht auf <i>s</i> paßt; belegt <code>RSTART</code> und <code>RLENGTH</code> (0 und -1 falls kein Treffer); füllt Array <i>a</i> mit in <i>r</i> geklammerten Treffern
	<code>patsplit(s, a[, r])</code>  <code>split(s, a[, r])</code>	Zerlegt <i>s</i> gemäß <i>r</i> in Arrayelemente, gibt ihre Anzahl zurück; legt diese in den Elementen 1 bis <i>n</i> des Arrays <i>a</i> ab; ohne <i>r</i> wird <code>FPAT</code> verwendet (verhält sich analog Eingabezeile-Zerlegung gemäß <code>FPAT</code> ) Zerlegt <i>s</i> gemäß <i>r</i> in Arrayelemente, gibt ihre Anzahl zurück; legt diese in den Elementen 1 bis <i>n</i> des Arrays <i>a</i> ab; ohne <i>r</i> wird <code>FS</code> verwendet; <i>r</i> ="" zerlegt <i>s</i> in Buchstaben (verhält sich analog Eingabezeile-Zerlegung gemäß <code>FS</code> )
*	<code>ord(s)</code>	Liefert ASCII-Code der 1. Zeichens von <i>s</i>
*	<code>sprintf(fmt, list)</code>	Gibt <i>list</i> formatiert gemäß <i>fmt</i> zurück (siehe <code>printf</code> )
*	<code>strtonum(s)</code>	Zeichenkette <i>s</i> in Dezimalzahl umwandeln (führende 0=als Oktalzahl, 0x/0X=als Hexadezimalzahl interpretiert)
	<code>substr(s, i[, n])</code>	Gibt Teilstück von <i>s</i> der Länge <i>n</i> ab Position <i>i</i> zurück; ohne <i>n</i> wird der Suffix ab Position <i>i</i> zurückgegeben; das erste Zeichen hat die Position 1
*	<code>toupper(s)</code>	Wandelt <i>s</i> in Großschreibung um
*	<code>tolower(s)</code>	Wandelt <i>s</i> in Kleinschreibung um

#### 14.4.6.4 Array-Funktionen

Gawk	Funktion	Bedeutung
*	<code>length(arr)</code>	Anzahl Elemente von Array <i>arr</i>
*	<code>delete arr</code>	Array <i>arr</i> vollständig löschen
*	<code>delete arr[ind]</code>	Arrayelement <i>arr[ind]</i> löschen
*	<code>asort(src[,dest[,how]])</code>	<b>Werte</b> von Array <i>src</i> bzw. von Array-Kopie <i>dest</i> gemäß <i>how</i> sortieren, indem die Original-Indices durch aufsteigend numerierte Indices 1, 2, ... ersetzt werden; Anzahl Elemente von Array <i>src</i> zurückgeben
*	<code>asorti(src[,dest[,how]])</code>	<b>Indices</b> von Array <i>src</i> bzw. von Array-Kopie <i>dest</i> gemäß <i>how</i> sortieren, indem die Original-Indices durch aufsteigend numerierte Indices 1, 2, ... ersetzt werden; Anzahl Elemente von Array <i>src</i> zurückgeben

#### 14.4.6.5 Ein/Ausgabe-Funktionen

Gawk	Funktion	Bedeutung
	<code>close(file)</code>	Datei <i>file</i> schließen
	<code>close(cmd)</code>	Kommando <i>cmd</i> (Pipe) schließen
*	<code>close(cmd)</code>	Koprozeß <i>cmd</i> (Pipe) schließen
*	<code>close(cmd, "to")</code>	Koprozeß <i>cmd</i> (Pipe hin) schließen
*	<code>close(cmd, "from")</code>	Koprozeß <i>cmd</i> (Pipe zurück) schließen
*	<code>fflush([file])</code>	Gepufferte Daten der Datei <i>file</i> schreiben
*	<code>fflush([cmd])</code>	Gepufferte Daten des Kommandos <i>cmd</i> schreiben
	<code>getline</code>	\$0 mit nächster Eingabezeile belegen; setzt NF, NR, FNR
	<code>getline &lt; file</code>	\$0 mit nächster Eingabezeile aus <i>file</i> belegen; setzt NF
	<code>getline var</code>	<i>var</i> mit nächster Eingabezeile belegen; setzt NR, FNR
	<code>getline var &lt; file</code>	<i>var</i> mit nächster Eingabezeile aus <i>file</i> belegen
	<code>cmd   getline</code>	Ergebnis von BS-Kommando <i>cmd</i> in \$0 ablegen; setzt NF
	<code>cmd   getline var</code>	... in <i>var</i> ablegen
*	<code>cmd  &amp; getline</code>	Ergebnis von BS-Koprozeß <i>cmd</i> in \$0 ablegen; setzt NF
*	<code>cmd  &amp; getline var</code>	... in <i>var</i> ablegen (Koprozeß)
	<code>print</code>	Aktuelle Eingabezeile \$0 ausgeben (mit Return)
	<code>print list</code>	Ausdrücke <i>list</i> ausgeben
	<code>print list &gt; file</code>	... auf <i>file</i> ausgeben ( <i>file</i> vorher löschen)
	<code>print list &gt;&gt; file</code>	... an <i>file</i> anhängen ( <i>file</i> vorher <i>nicht</i> löschen)
	<code>print list   cmd</code>	... an BS-Kommando <i>cmd</i> übergeben
*	<code>print list  &amp; cmd</code>	... an BS-Koprozeß <i>cmd</i> übergeben (läuft parallel)
	<code>printf fmt, list</code>	Ausdrücke <i>list</i> gemäß <i>fmt</i> formatieren und ausgeben
	<code>printf fmt, list &gt; file</code>	... auf <i>file</i> ausgeben ( <i>file</i> vorher löschen)
	<code>printf fmt, list &gt;&gt; file</code>	... an <i>file</i> anhängen ( <i>file</i> vorher <i>nicht</i> löschen)
	<code>printf fmt, list   cmd</code>	... an BS-Kommando <i>cmd</i> übergeben
	<code>printf fmt, list  &amp; cmd</code>	... an BS-Koprozeß <i>cmd</i> übergeben (parallel)
	<code>system(cmd)</code>	BS-Kommando <i>cmd</i> ausführen, Exit-Status zurückgeben

#### 14.4.6.6 getline-Rückgabewerte

Code	Bedeutung
1	Bei erfolgreichem Lesen eines Satzes
0	Am Dateiende
-1	Bei Fehler (z.B. Datei existiert nicht)

#### 14.4.6.7 printf-Formatumwandlungen

Gawk	Format	Bedeutung
	%c	Zeichen [ <b>character</b> ]
i	%d/i	Dezimalzahl mit Vorzeichen [ <b>decimal/integer</b> ]
E	%e/E	Gleitkommazahl [-] d. ddddddde [+ -] dd [ <b>exponent</b> ]; e klein oder E groß geschrieben
	%f	Gleitkommazahl [-] ddd. ddddddd [ <b>float</b> ]
G	%g/G	Verwendet e/E- oder f-Umwandlung, je nachdem welche kürzer ist; nicht signifikante Nullen werden unterdrückt [ <b>general</b> ]
	%o	Oktalzahl ohne Vorzeichen [ <b>octal</b> ]
*	%u	Dezimalzahl ohne Vorzeichen [ <b>unsigned</b> ]
	%s	Zeichenkette [ <b>string</b> ]
*	%x/X	Hexadezimalzahl ohne Vorzeichen [ <b>hexadecimal</b> ]; a-f klein oder A-F groß geschrieben
	%%/\%	Gibt ein % aus; verwendet kein Argument

#### 14.4.6.8 printf-Zusatzangaben zu Formatumwandlungen

Gawk	Parameter	Bedeutung
	-	Ausdruck linksbündig ausrichten ( <b>Default</b> : rechtsbündig)
	␣	Leerzeichen vor positiven Zahlen, - vor negativen
	+	+ vor positiven Zahlen, - vor negativen
	0	Führende Nullen statt Leerzeichen (nur bei <i>width</i> relevant)
*	#	Alternatives Format bei: o = Führende Null ausgeben x/X = Führendes 0x oder 0X ausgeben eEf = Dezimalpunkt immer ausgeben gG = Nullen am Ende nicht entfernen
	<i>width</i>	Ausgabe in Breite <i>width</i> (rechtsbündig mit Leerzeichen), eine führende 0 verwendet dazu Nullen
	<i>.prec</i>	Anzahl Nachkommastellen <i>prec</i> bei Zahlen, maximale Breite <i>prec</i> bei Zeichenketten
*	*	Statt <i>width</i> oder <i>prec</i> den nächsten Parameter in der <code>printf</code> -Argumentliste verwenden (nur <i>Gawk</i> )
*	<i>n</i> \$	<b>Positionsparameter</b> : <i>n</i> -tes Argument der Argumentliste einsetzen

#### 14.4.6.9 Zeit-Funktionen



Funktion	Bedeutung
<code>sys time()</code>	Aktuelle Zeit in Anzahl Sekunden ( <i>sec</i> ) seit 1.1.1970 1:00 UTC
<code>str time([fmt[, sec]])</code>	Sekunden <i>sec</i> gemäß <i>fmt</i> formatieren Defaultformat: "%a %b %d %H:%M:%S %Z %Y" Defaultzeit: <code>sys time()</code>
<code>mk time(datespec)</code>	Datum der Form "YYYY MM DD HH MM SS [DST]" (6/7 Elemente) in Zeitstempel (Anzahl Sek. seit 1.1.1970 1:00 UTC) konvertieren.

#### 14.4.6.10 strftime-Formatangaben

Format	Bedeutung
%A / %a	Wochentagname vollständig / abgekürzt auf 3 Zeichen
%B / %b	Monatsname vollständig / abgekürzt auf 3 Zeichen
%c	Datum + Zeit gemäß lokalem Standard
%d	Monatstag (01-31)
%H / %I	24 / 12-Stunde (00-23)
%j	Jahrestag (001-366)
%m	Monat (01-12)
%M	Minute (00-59)
%p	AM/PM
%S	Sekunden (00-61, wegen Korrektursekunden)
%U / %W	Wochennummer (00-53, erster Sonntag/Montag ist erster Wochentag 1)
%w	Wochentagnummer (0-6, 0=Sonntag)
%X / %x	Zeit / Datum gemäß lokalem Standard
%Y / %y	Jahr 4- / 2-stellig (1999/00-99)
%Z	Zeitzone
%%	% wörtlich

#### 14.4.6.11 Internationalisierung-Funktionen

Funktion	Bedeutung
<code>bindtextdomain(dirpath[, dom])</code>	Verzeichnis <i>dirpath</i> für .mo-Dateien festlegen gemäß Textdomäne <i>dom</i> ( <b>Default:</b> TEXTDOMAIN)
<code>dcgettext(s[, dom[, cat]])</code>	Übersetzung von <i>s</i> gemäß Textdomäne <i>dom</i> ( <b>Default:</b> TEXTDOMAIN) und Lokalkategorie <i>cat</i> ( <b>Default:</b> LC_MESSAGES)
<code>dcngettext(s, p, n[, dom[, cat]])</code>	Singularform <i>s</i> ( <i>n</i> = 1) oder Pluralform <i>p</i> ( <i>n</i> > 1) gemäß Zahl <i>n</i> übersetzen gemäß Textdomäne <i>dom</i> ( <b>Default:</b> TEXTDOMAIN) und Lokalkategorie <i>cat</i> ( <b>Default:</b> LC_MESSAGES)

#### 14.4.6.12 Typ-Funktionen

Funktion	Bedeutung
<code>isarray(x)</code>	Ergibt wahr, falls <i>x</i> ein Array ist, sonst falsch
<code>typeof(x)</code>	Gibt String mit Typ von Identifier <i>x</i> zurück (array, number, regexp, string, strnum, unassigned, undefined)
<code>mkbool(expr)</code>	Ergibt abhängig von <i>expr</i> den Wert „wahr“ (1) oder „falsch“ (0)

## 14.5 GAWK-Spezialdateien

Datei	Inhalt
/dev/pid	Aktuelle Prozeß-ID
/dev/ppid	Prozeß-ID des Elternprozesses
/dev/pgrpid	Aktuelle Prozeß-Gruppen-ID
/dev/user	Folgende 4 oder mehr Arrayelemente: 1. Real User-ID 2. Effective User-ID 3. Real Group-ID 4. Effective Group-ID 5.-n: Group-IDs