

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

Version 1.20 — 11.8.2007

© 2005–2007 T. Birnthaler, 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 Birnthaler
eMail: tb@ostc.de
Web: www.ostc.de

Inhaltsverzeichnis

1	Schreibweise in dieser Zusammenfassung	6
2	Grundlagen	7
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	9
2.5	Wichtige KDE-spezifische Steuertasten	9
2.6	Maus-Bedienung unter X Window	9
2.7	Die wichtigsten UNIX-Kommandos	10
3	Benutzer- und Gruppenverwaltung	10
3.1	Kommandos zur Benutzerverwaltung	10
3.2	Kommandos zur Gruppenverwaltung	11
4	Dateisystem	11
4.1	Auskunftsfunktionen	11
4.2	Dateitypen (Kommando ls -l)	12
4.3	Verändernde Kommandos	12
4.4	Besitzverhältnis und Zugriffsrechte	13
4.4.1	Kommandos	13
4.4.2	Zugriffsrechte für Dateien	13
4.4.3	Zugriffsrechte für Verzeichnisse	13
4.4.4	Dateirechte in oktaler + symbolischer Form	14
4.4.5	LINUX-Sonderrechte	14
4.5	Dateien und Verzeichnisse	14
4.5.1	Standard-Verzeichnisnamen	14
4.5.2	UNIX Standard-Verzeichnisse	15
4.5.3	LINUX-spezifische Standard-Verzeichnisse	15
4.5.4	Die wichtigsten Gerätedateien von LINUX	16
4.5.5	Wichtige zentrale Konfigurationsdateien	16
4.5.6	Wichtige zentrale Netzwerk-Konfigurationsdateien	17
4.5.7	Wichtige lokale Konfigurationsdateien/Verzeichnisse	17
5	Prozeßverwaltung	18
5.1	Kommandos zur Prozeßverwaltung	18
5.2	Kommandos zur Hintergrund-Prozeß-Verwaltung	18
5.3	Die wichtigsten Signale	19
6	Shell	19
6.1	Shell-Typen	19
6.2	Shell-Konfigurationsdateien	19
6.2.1	Konfigurationsdateien der Bourne-Shell	20
6.2.2	Konfigurationsdateien der Bash	20

6.2.3	Konfigurationsdateien der C-Shell	20
6.2.4	Konfigurationsdateien der TC-Shell	20
6.2.5	Konfigurationsdateien der Korn-Shell	21
6.3	Shell-Operationen	21
6.4	Pfadsuche (bei Aufruf von CMD)	21
6.5	Shell- und Umgebungs-Variablen	22
6.5.1	Kommandos für Shell-Variablen	22
6.5.2	Kommandos für Umgebungs-Variablen	22
6.5.3	Variablen-Kommandos unter der (T)C-Shell	22
6.5.4	Einige Standard-Variablen	23
6.5.5	Eingabeprompt-Definition (bash)	23
6.5.6	Eingabeprompt-Definition (tcsh)	24
6.6	Kommando-Wiederholung (bash + ksh)	24
6.7	Dateinamen-Vervollständigung (bash + ksh)	24
6.8	Aliase	25
6.8.1	Nützliche Aliase	25
6.9	Funktionen	25
6.10	Ein/Ausgabe-Umlenkung	26
6.11	Spezielle Ein-/Ausgabeumlenkung	26
6.12	Here-Dokument	26
6.13	Dateinamen-Expansion	26
6.14	Kommando-Substitution	27
6.15	Shell-Quotierung	27
6.16	Kommando-Listen	27
7	Sonstige Kommandos	27
7.1	Wichtige Format-Angaben zu date	28
8	Vi	29
8.1	Die wichtigsten Vi-Befehle	29
8.2	Weitere wichtige Vi-Befehle	30
8.3	Wichtige allgemeine Vi-Optionen	30
8.4	Wichtige Vi-Optionen für Programmierer	31
8.5	Wichtige Vim-Optionen	31
8.6	Nützliche Vi-Makros	31
9	Drucken	32
9.1	BSD-Variante der Druck-Kommandos	32
9.2	System V-Variante der Druck-Kommandos	32
9.3	lpc-Kommandos	32
10	Reguläre Ausdrücke	33
10.1	Standard-Metazeichen	33
10.2	Erweiterte Metazeichen	33
10.3	Metazeichen im Ersetzungsmuster	33
10.4	Escape-Sequenzen	34
10.5	perl-Metazeichen	34

11 UNIX-Werkzeuge	34
11.1 strings, wc, head, tail, tee	34
11.2 cmp, diff	35
11.3 cut, paste, join, tr, split	35
11.4 sort, uniq	36
11.5 grep	36
11.5.1 Die wichtigsten Optionen von grep	36
11.6 ed, ex, sed, awk, perl	37
11.6.1 Sed-Kommandos	38
11.7 find, locate	38
11.7.1 find-Bedingungen	39
11.7.2 find-Aktionen	39
11.7.3 find-Beispiele	39
11.8 tar, cpio	40
11.9 compress, gzip	40
12 System-Administration	41
12.1 fdisk, fdformat, mkfs, fsck, dd	41
12.2 mount, umount	41
12.2.1 Mount-Optionen	42
12.2.2 Samba-Mount-Optionen	42
12.3 at, crontab	42
12.3.1 at-Befehle	42
12.3.2 crontab-Befehle	42
12.3.2.1 crontab-Felder	42
12.3.2.2 Erlaubte crontab-Feldwerte	43
12.4 mail	43
12.4.1 Die wichtigsten Mail-Befehle	43
13 Shell-Programmierung	44
13.1 Shell-Aufrufarten	44
13.2 Shell-Optionen	44
13.3 Spezielle Shell-Variablen (Parameter)	44
13.3.1 Bedingte Auswertung von Shell-Variablen	45
13.4 Bedingungen des test-Kommandos	45
13.4.1 Textvergleiche	45
13.4.2 Dateiattribut-Vergleiche	46
13.4.3 Datei-Vergleiche	46
13.4.4 Numerische Vergleiche	46
13.4.5 Logische Verknüpfungen	46
13.4.6 Shell-Optionen testen	47
13.5 Kontrollstrukturen	47
13.5.1 Benutzereingabe einlesen	47
13.5.2 Verzweigung	47
13.5.3 case-Mehrfachverzweigung (Text-Vergleich mit Shell-Metazeichen)	47
13.5.4 for-Schleife (Liste bzw. Shell-Aufrufargumente abarbeiten)	48
13.5.5 while und until-Schleife (Bedingung prüfen)	48

13.5.6 Funktions-Definition und -Aufruf	48
14 Awk	49
14.1 AWK-Optionen	49
14.2 GNU-AWK-spezifische Optionen	49
14.3 Konstanten	49
14.3.1 Zahlen	49
14.3.2 Zeichenketten	49
14.3.3 Escape-Sequenzen	50
14.3.4 Reguläre Ausdrücke	50
14.3.4.1 Metazeichen in Regulären Ausdrücken	50
14.3.4.2 POSIX-Zeichenklassen ([[<i>:class:</i>]])	51
14.3.5 Ausdrücke	51
14.4 Programmaufbau	51
14.4.1 Regel	51
14.4.1.1 Muster	51
14.4.1.2 Aktionen	51
14.4.2 Funktionen	52
14.4.2.1 Definition	52
14.4.2.2 Aufruf	52
14.4.3 Zeilenumbrücken sind erlaubt nach	53
14.4.4 Felder, Variablen und Arrays	53
14.4.4.1 Defaultwerte für Feld- und Satztrenner	53
14.4.4.2 Felder	53
14.4.4.3 Variablen	53
14.4.4.4 Arrays	53
14.4.4.5 Vergleichstypen	54
14.4.5 Operatoren	54
14.4.6 Vordefinierte Variablen und Arrays	55
14.4.7 Vordefinierte Funktionen	55
14.4.7.1 Arithmetik-Funktionen	55
14.4.7.2 Zeichenketten-Funktionen	55
14.4.7.3 Ein/Ausgabe-Funktionen	56
14.4.7.4 getline-Rückgabewerte	57
14.4.7.5 printf-Formatumwandlungen	57
14.4.7.6 printf-Zusatzangaben zu Formatumwandlungen	58
14.4.7.7 Zeit-Funktionen	58
14.4.7.8 strftime-Formatangaben	58
14.5 GAWK-Spezialdateien	59

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

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

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

2 Grundlagen

2.1 Datei-Inhalt ansehen

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

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

2.2 Hilfestellung abrufen

man [[-s] NN] CMD	Beschreibung zu CMD aus man-Abschnitt NN [section]
man -f CMD	Einzeilige man-Beschreibungen zu CMD ausgeben [find]
man -k TEXT	Einzeilige man-Beschreibungen mit TEXT darin ausgeben [key]
man -t CMD lpr	CMD-Beschreibung mit troff setzen und ausdrucken [troff]
whatis CMD	Analog man -f CMD
apropos TEXT	Analog man -k TEXT
CMD --help	Usage-Meldung zu CMD (GNU-Programme)
CMD -?	Usage-Meldung zu CMD (einige Programme)
CMD -h	Usage-Meldung zu CMD (einige Programme) [help]
CMD -.	Usage-Meldung zu CMD (einige Programme)
help	Alle eingebauten Shell-Kommandos ausgeben (bash)
help CMD	Usage-Meldung zu eingebautem Shell-CMD ausgeben
hilfe	Browsergestütztes SuSE-Hilfesystem aufrufen
info CMD	GNU-Beschreibung zu CMD 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 /usr/lib)
4	Geräte-dateien (gewöhnlich in /dev)
5	Formate und Bedeutung von Konfigurations-/Logdateien (z.B. /etc/passwd)
6	Spiele
7	Makropakete und Konventionen (z.B. man(7), groff(7))
8	Systemadministrationsbefehle (in der Regel nur für root)
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
<Ctrl-C>	Laufendes Kommando abbrechen [break]
<Ctrl-D>	Abmelden/Dateiende anzeigen [end of file]
<Ctrl-Z>	Aktuellen Prozeß stoppen + in Hintergrund stellen
<Ctrl-S>	Terminal-Scrolling anhalten
<Ctrl-Q>	Terminal-Scrolling laufen lassen
<Ctrl-U>	Ganze Eingabezeile löschen [undo]
<Ctrl-W>	Letztes Wort auf der Eingabezeile löschen [word]
<Ctrl-H>	Backspace
<Ctrl-I>	Tabulator
<Ctrl-V>	Nächstes Steuer-Zeichen schützen (Escape) [verbose]
<Ctrl-G>	Klingel
<Str1-L>	Bildschirm neu aufbauen

2.4 Wichtige LINUX-spezifische Steuertasten

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

2.5 Wichtige KDE-spezifische Steuertasten

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

2.6 Maus-Bedienung unter X Window

Linke Taste Linke Taste 2x Mittlere Taste Rechte Taste	Text markieren durch Halten+Ziehen Doppelklick markiert Wort, Halten+Ziehen markiert wortweise Markierten Text an Cursorposition einfügen (auch Linke+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 whoami who am i finger USER su su - su USER su - USER exit exit <Ctrl-D> passwd yppasswd chsh chfn	Alle angemeldeten Benutzer anzeigen Eigene Kennung anzeigen Eigene Kennung anzeigen (ausführlicher) Weitere Informationen über Benutzer beschaffen In Kennung root wechseln [substitute user] In Kennung + Umgebung von root wechseln In Kennung USER wechseln (alte Umgebung) In Kennung + Umgebung von USER wechseln Rücksprung zur vorherigen Kennung bzw. Abmelden/Fenster schließen Abmelden/Fenster schließen Paßwort ändern [password] NIS-Paßwort ändern [yellow pages] Login-Shell ändern (LINUX) [change shell] Benutzer-Beschreibung ändern (LINUX) [change full name]
ypcat passwd ypcat group ypcat hosts	NIS-Benutzerdatenbank ausgeben [yellow pages catalog] NIS-Gruppendatenbank ausgeben NIS-Rechnerdatenbank ausgeben

3.2 Kommandos zur Gruppenverwaltung

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

4 Dateisystem

4.1 Auskunftsfunktionen

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

4.2 Dateitypen (Kommando ls -l)

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

4.3 Verändernde Kommandos

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

4.4 Besitzverhältnis und Zugriffsrechte

4.4.1 Kommandos

chgrp GROUP FILE	FILE der Gruppe GROUP zuordnen (<i>Mitglied!</i>) [change group]
chown USER FILE	FILE dem Benutzer USER zuordnen (<i>nur root!</i>) [change owner]
chown USER.GROUP	Benutzer + Gruppe gleichzeitig ändern (<i>nur root!</i>)
chmod MODE FILE	Dateirechte ändern [change mode] MODE = [ugoa][+ -=][rwxst] oder (0)000-(7)777 (s.u.)
chmod u+rwx FILE	Besitzer erhält alle Rechte an FILE [user]
chmod g-w FILE	Besitzergruppe wird Schreibrecht an FILE entzogen [group]
chmod o-rwx FILE	Anderer haben keine Rechte mehr an FILE [others]
chmod a+r FILE	Alle haben Leserecht für Datei FILE [all]
chmod 644 FILE	Rechte rw-r--r-- für Datei FILE setzen
chmod -R ...	Rechte für ganzen Verzeichnisbaum ändern [recursive]
umask [NNN]	Standardrechte NNN für Datei/Verz-Neuanlage anzeigen/setzen, definiert die zu entfernenden(!) Rechte [usage mask]
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	[read]	Datei-Inhalt lesbar
w	[write]	Datei-Inhalt veränderbar
x	[execute]	Datei als Programm ausführbar
s	[set user id]	Programmausführung unter der Besitzer-Kennung
s	[set group id]	Programmausführung unter der Besitzergruppen-Kennung
t	[sticky]	Programm nach dem Start im Speicher/Swap halten (<i>veraltet!</i>)

4.4.3 Zugriffsrechte für Verzeichnisse

r	read	Verzeichnis lesbar (<i>ls</i>)
w	write	Verzeichnis veränderbar (Dateien anlegen/löschen/umbenennen verschieben, z.B. cp, ln, mkdir, mv, rm, rmdir, touch)
x	execute	Wechsel in Verzeichnis erlaubt (<i>cd</i>)
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 root) 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

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

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

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

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 (<code>lilo</code>)
/cdrom	Mount-Point für CDROM-Laufwerk (auch <code>/media/cdrom</code>)
/dvd	Mount-Point für DVD-Laufwerk (auch <code>/media/dvd</code>)
/floppy	Mount-Point für Floppy-Laufwerk (auch <code>/media/floppy</code>)
/mnt	Mount-Point für temporäre Dateisysteme [mount]
/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 [process]
/root	Heimat-Verzeichnis der <code>root</code> (evtl. auch <code>/</code>)
/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 [file system table]
/etc/group	Gruppenkonten (bis auf Paßwort)
/etc/gshadow	Gruppenpaßworte (verschlüsselt)
/etc/inittab	Bootvorgang steuern (Runlevel + Startup-Skripte) [initialization table]
/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 [line print demon]
/etc/hosts.allow	Zugriffsteuerung <code>tcpd</code> erlaubt diesen Rechner den Zugang
/etc/hosts.deny	Zugriffsteuerung <code>tcpd</code> verbietet diesen Rechner den Zugang
/etc/inetd.conf	Von <code>inetd</code> zu startende Netzwerkdienste
/etc/named.conf	DNS-Konfiguration
/etc/networks	Zuordnung Netzname ↔ IP-Adresse (Nameserver-Ersatz)
/etc/nsswitch.conf	Definiert die Reihenfolge der Namensauflösung [name service switch]
/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 <code>csh</code> ausgeführt
.emacs	Emacs-Konfiguration
.exrc	Vi/Ex-Konfiguration
.forward	Mail-Umlenkung
.login	Beim Start einer Login- <code>csh</code> ausgeführt
.logout	Beim Verlassen einer Login- <code>csh</code> ausgeführt
.netscape	Netscape-Einstellungen
.profile	Beim Start einer Login-Shell vom Typ <code>sh</code> , <code>bash</code> , <code>ksh</code> ausgeführt
.rhosts	Erlaubte Hosts für remote-Kommandos ohne Paßwort
.vimrc	Vim-Konfiguration
.xinitrc	X Window Start-Skript (<code>startx</code> → <code>xinit</code>)
.xsession	X Window Start-Skript (<code>xdm</code> oder <code>kdm</code>)

5 Prozeßverwaltung

5.1 Kommandos zur Prozeßverwaltung

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

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

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

5.3 Die wichtigsten Signale

SIGHUP	1	Konfigurationsdatei lesen [hangup]
SIGINT	2	Abbruch durch Eingabe von <Ctrl-C> [interrupt]
SIGQUIT	3	Prozeßende erreicht
SIGKILL	9	Bedingungsloser Prozeßabbruch [töten]
SIGTERM	15	Prozeß beenden, wird evtl. ignoriert [terminate]
SIGCONT	18	Ausgabe weiterlaufen lassen durch Eingabe von <Ctrl-Q> [continue]
SIGSTOP	19	Ausgabe anhalten durch Eingabe von <Ctrl-S>
SIGTSTP	24	In Hintergrund stellen durch Eingabe von <Ctrl-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 ~/.kshrc	Bestimmt die für eine interaktive ksh auszuführende Datei (Inhalt ~/.kshrc) [environment]

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

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

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

6.6 Kommando-Wiederholung (bash + ksh)

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

6.7 Dateinamen-Vervollständigung (bash + ksh)

In der `bash` sind verwendbar:

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

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

6.10 Ein/Ausgabe-Umlenkung

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

6.11 Spezielle Ein-/Ausgabeumlenkung

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

6.12 Here-Dokument

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

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 (sh)
[^abc] [^a-z]	Genau ein Zeichen nicht(!) aus der angegebenen Menge (csh)
~	Heimat-Verzeichnis des aktuellen Benutzers (tcsh, bash)
~USER	Heimat-Verzeichnis des Benutzers USER (tcsh, bash)
{abc,def,...}	Liste der angegebenen Zeichenketten (csh, ksh, bash)

6.14 Kommando-Substitution

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

6.15 Shell-Quotierung

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

6.16 Kommando-Listen

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

7 Sonstige Kommandos

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

7.1 Wichtige Format-Angaben zu date

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

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 (!= <i>erzwingen!</i>) [quit] Editierten Text (auf Datei FILE) schreiben [write]
:e[!] FILE CR :f FILE CR	Datei FILE statt aktueller editieren (!= <i>erzwingen!</i>) [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 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] 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 <Ctrl-F> <Ctrl-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 <Ctrl-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

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

8.3 Wichtige allgemeine Vi-Optionen

<code>errorbells</code>	Bei Eingabefehler piepsen
<code>ignorecase</code>	Groß/Kleinschreibung bei der Suche ignorieren
<code>list</code>	Zeilenenden durch \$ und Tabulatoren durch ^I anzeigen
<code>number</code>	Zeilen numeriert anzeigen
<code>report=NN</code>	Änderungen ab <code>NN</code> Zeilen in Statuszeile anz. (<i>1=ab 1 Zeile, 0 geht n.!</i>)
<code>showmode</code>	Vi-Modus in Statuszeile anzeigen (INSERT, APPEND, ...)
<code>wrapmargin=NN</code>	Automatischer Umbruch <code>NN</code> Zeichen vor Zeilenende (<i>0=kein Umbruch!</i>)
<code>wrapscan</code>	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	Automatisch einrücken (mit <Ctrl-D/T> ein/ausrücken)
shiftwidth=NN	Breite NN für nachträgliches Einrücken setzen (<< >>)
showmatch	Passende öffnende Klammer bei Eingabe der schließenden anzeigen
tabstop=NN	Tabulatorbreite auf NN Zeichen setzen

8.5 Wichtige Vim-Optionen

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

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
Ctrl-W	/\<	Suche nach Wort einleiten (danach Wort eintippen)
Ctrl-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 <Ctrl-V> <RETURN> oder im Vim durch <CR>.

9 Drucken

9.1 BSD-Variante der Druck-Kommandos

lpr FILE	Datei FILE drucken [line print]
CMD lpr	Ergebnis von Kommando CMD drucken
lpq	Abgesetzte Druckaufträge anzeigen [line print queue]
lprm JOBNR	Druckauftrag JOBNR löschen [line print remove]
lprm -	Alle Druckaufträge löschen (<i>nur root!</i>)
lpc [CMD]	Drucker verwalten [line print control]
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 [line print]
CMD lp	Ergebnis von Kommando CMD drucken
lpstat	Abgesetzte Druckaufträge anzeigen [line print status]
cancel JOBNR	Druckauftrag JOBNR löschen
-d PRINTER	Drucker festlegen [destination]
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 [destination]

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 [newline]
\(...\)	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 [up/lowcase]
\U \L	Alle folgenden Zeichen in Groß/Kleinschrift umwandeln
\E	Durch \U oder \L begonnene Umwandlung beenden [end]

10.4 Escape-Sequenzen

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

10.5 perl-Metazeichen

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

11 UNIX-Werkzeuge

11.1 strings, wc, head, tail, tee

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

11.2 cmp, diff

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

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 [delimiter,fields] Spalten L, M und N ausgeben [columns] Spalten NN-MM ausgeben [columns]
paste FILE1 FILE2 paste -d":@;" ... paste -s FILE paste -s - - -	Zeilen aus FILE1 und FILE2 nebeneinanderstellen Spaltentrenner :@; festlegen (\0=keiner) [delimiter] Alle Zeile von FILE in einer Zeile zusammenfassen [squeeze] 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>) [terminator] 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 [translate] Zeichen nicht(!) in SET1 ersetzen [complement] Zeichen in SET1 löschen [delete] Gleiche Zeichen hintereinander zusammenziehen [squeeze]
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) [lines] Stücke der Länge NN Bytes erstellen [bytes]

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

11.4 sort, uniq

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

11.5 grep

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

11.5.1 Die wichtigsten Optionen von grep

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

11.6 ed, ex, sed, awk, perl

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

11.6.1 Sed-Kommandos

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

11.7 find, locate

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

11.7.1 find-Bedingungen

-group NAME/ID	Besitzergruppe
-mtime [+ -]DAYS	Änderung vor mehr/weniger/genau DAYS 24h-Tagen [modification]
-name "PATTERN"	Dateinamen-Muster (* ? [...] ... , vor der Shell schützen)
-perm [-]RIGHTS	(Oktale) Rechte (NNN=exakt, -NNN=mindestens) [permission]
-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 [access]
-ctime [+ -]DAYS	Status-Änderung (Inode) vor mehr/... 24h-Tagen [change]
-iname "PATTERN"	Analog -name, aber Groß/Kleinschreibung egal [ignorecase]
-inum [+ -]NUM	Inode-Nummer mehr/weniger/gleich NUM [inode number]
-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 [directory]
find / -size 0 -ok rm {} \;	Leere Dateien löschen (mit Abfrage)
find / -type f \	Namen der Dateien die Text made enthalten [file]
-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>	<p>Archivdatei verwalten [tape archiver]</p> <p>ARCH erzeugen [create/file]</p> <p>ARCH in aktuellem Verz. auspacken [extract/verbose]</p> <p>ARCH mit Anzeige testen [test]</p> <p>Komprimiertes Archiv ARCH testen [test,ziped]</p>
<pre>cpio ... cpio -o ARCH cpio -id < ARCH cpio -tv < ARCH</pre>	<p>Archivdatei verwalten [copy input output]</p> <p>ARCH (Dateiliste von stdin) erz. auf stdout [output]</p> <p>ARCH ausp. (von stdin) mit Verz. erz. [input/directory]</p> <p>ARCH mit Anzeige testen (von stdin) [test/verbose]</p>

11.9 compress, gzip

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

12 System-Administration

12.1 fdisk, fdformat, mkfs, fsck, dd

fdisk DEV fdisk -l DEV	Gerät DEV menügesteuert partitionieren [format disk] Alle Partitionen von Gerät DEV auflisten [list]
fdformat DEV fdformat /dev/fd0u1440 fdformat /dev/fd0u1722	Floppy-Disk DEV formatieren [floppy disk format] (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 [make filesystem] MSDOS-Dateisystem auf Diskette anlegen Ext2-Dateisystem auf 1. Partition der 1. Platte anlegen
fsck DEV fsck /dev/hdb4	Dateisystem überprüfen [file system check] 4. Partition der 2. Festplatte überprüfen
dd if=FILE of=FILE bs=NN count=NN skip=NN seek=NN	Sektoren lesen/schreiben (Platte/Floppy) [disk duplicate] Eingabedatei [input file] Ausgabedatei [output file] Blockgröße NN [block size] 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 [montieren] 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 [all] Alle Geräte vom Typ TYPE in /etc/fstab einhängen [type]
umount PATH umount DEV umount -a mount -t TYPE	Gerät mit Mount-Point PATH aushängen [unmount] Gerät DEV aushängen Alle Geräte in /etc/fstab aushängen (bis auf root-Verz) [all] Alle Geräte vom Typ TYPE in /etc/fstab aushängen [type]

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 [at queue]
atrm JOBNR	Auftrag mit JOBNR löschen [at remove]

12.3.2 crontab-Befehle

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

12.3.2.1 crontab-Felder

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

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 [head]
NN	Mail Nummer NN anzeigen
d	Aktuelle Mail löschen [delete]
d NN-MM	Mails NN-MM löschen [delete]
q	Mailprogramm verlassen [quit]
x	Mailprogramm ohne jede Änderung verlassen [quit]
<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 angebar
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) [noexec]
-u	Abbruch bei Zugriff auf nicht initialisierte Variable [unset]
-v	Gelesene Kommandos ausgeben [verbose]
-x	Wirklich ausgeführte Kommandos vorher ausgeben [execute]
-o OPT	Option OPT setzen
+o OPT	Option OPT zurücksetzen
-o ignoreeof	<Ctrl-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 <code>\${10}</code> ... 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>) [nonzero]
<code>-z "TEXT"</code>	TEXT ist leer [zero]
<code>"TEXT1" = "TEXT2"</code>	TEXT1 und TEXT2 sind gleich
<code>"TEXT1" != "TEXT2"</code>	TEXT1 und TEXT2 sind verschieden
<code>"TEXT1" \<< "TEXT2" K</code>	TEXT1 alphabetisch kleiner als TEXT2 (quot. wg. Shell)
<code>"TEXT1" \<> "TEXT2" K</code>	TEXT1 alphabetisch größer als TEXT2 (quot. wg. Shell)

13.4.2 Dateiattribut-Vergleiche

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

13.4.3 Datei-Vergleiche

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

13.4.4 Numerische Vergleiche

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

13.4.5 Logische Verknüpfungen

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

Die Reihenfolge von oben nach unten entspricht dem Vorrang.

13.4.6 Shell-Optionen testen

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

13.5 Kontrollstrukturen

13.5.1 Benutzereingabe einlesen

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

13.5.2 Verzweigung

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

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

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

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>	<p>Definition (<i>vor Aufruf!</i>) Extra Zeile! Einrücken sinnvoll Zugriff auf 1.Parameter Zugriff auf 2.Parameter Extra Zeile!</p>
<pre>func1 func1 100 Test</pre>	<p>Aufruf (<i>nach Definition!</i>) Aufruf mit Parametern</p>

14 Awk

14.1 AWK-Optionen

Option	Bedeutung
-F <i>sep</i>	Feldtrenner FS auf <i>sep</i> festlegen (Default: "_")
-f <i>progfile</i>	Programm von <i>progfile</i> einlesen (<i>mehrfach erlaubt</i>)
-v <i>var=text</i>	Variable <i>var</i> mit dem Wert <i>text</i> belegen (<i>vor BEGIN</i>)
-mf <i>n</i>	Maximale Anzahl Felder <i>n</i> festlegen (<i>im Gawk überflüssig</i>)
-mr <i>n</i>	Maximale Recordlänge <i>n</i> festlegen (<i>im Gawk überflüssig</i>)
--	Beendet die Optionen-Liste

14.2 GNU-AWK-spezifische Optionen

Gawk-Option	Bedeutung
--compat	Gawk-Erweiterungen abschalten
--traditional	Gawk-Erweiterungen abschalten
--copyleft	GPL (GNU General Public Licence) ausgeben
--copyright	GPL (GNU General Public Licence) ausgeben
--help	Gawk-Usage-Meldung ausgeben
--usage	Gawk-Usage-Meldung ausgeben
--version	Gawk-Versionsnummer ausgeben
--lint	Warnung bei gefährlichen/nicht portablen Konstrukten ausgeben
--lint-old	Warnung bei Konstrukten abweichend vom Ur-Awk ausgeben
--posix	Nur POSIX-Umfang zulassen
--re-interval	{ <i>n,m</i> }-Wiederholung in Regulären Ausdrücken erlauben
--source ' <i>program</i> '	Programmtext <i>program</i> auf der Kommandozeile angeben (<i>mehrfach erlaubt</i>)

14.3 Konstanten

14.3.1 Zahlen

```
123  -1  -3.141592  +.0125  1e12  -987.654E-321
```

14.3.2 Zeichenketten

```
"abc"
"Dies ist ein Text.\n"
"" (leere Zeichenkette)
```

14.3.3 Escape-Sequenzen

Gawk	Escape	Bedeutung
*	\a	Akustisches Signal [alert]
	\b	Backspace (Zeichen zurück)
	\f	Seitenvorschub [formfeed]
	\n	Zeilenvorschub [newline]
	\r	Wagenrücklauf [carriage return]
	\t	Tabulator
*	\v	Vertikaler Tabulator
	\ddd	Zeichen mit oktalem Wert <i>ddd</i> (Ziffern zwischen 0 und 7)
*	\xdd	Zeichen mit hexadezimalen Wert <i>dd</i> (Ziffern zwischen 0 und f/F)
	\"	Anführungszeichen
	\\	Backslash
	\/	Slash

14.3.4 Reguläre Ausdrücke

/abc/ Enthält "abc"
 /^abc\$/ Lautet exakt "abc"
 /(abc|def)+/ Enthält "abc" oder "def" mind. 1x nacheinander

14.3.4.1 Metazeichen in Regulären Ausdrücken

Gawk	Metazeichen	Bedeutung
	(<i>r</i>)	Gruppierung: jede Zeichenkette, auf die <i>r</i> paßt
	<i>c</i>	Zeichen <i>c</i> (kein Metazeichen)
	\ <i>c</i>	Eine Escape-Sequenz oder das (Meta)Zeichen <i>c</i> wörtlich
	^	Zeichenketten-/Zeilenanfang
	\$	Zeichenketten-/Zeilenende
	.	Jedes beliebige Zeichen
	[<i>abc</i>], [<i>a-z</i>]	Zeichenliste: jedes Zeichen in <i>abc</i> , <i>a-z</i>
	[^ <i>abc</i>], [^ <i>a-z</i>]	Negierte Zeichenliste: jedes Zeichen außer <i>abc</i> , <i>a-z</i>
*	\y \B	Wortgrenze / Wortinneres
*	\< \>	Wortanfang / Wortende
*	\w \W	Wortzeichen ([A-Za-z_0-9]) / kein Wortzeichen ([^A-Za-z_0-9])
*	\' \'	Zeichenketten-/Zeilenanfang / -ende (alternative Form)
	<i>r</i> *	0 – ∞ aufeinanderfolgende Zeichenketten, auf die <i>r</i> paßt (Closure)
	<i>r</i> +	1 – ∞ aufeinanderfolgende Zeichenketten, auf die <i>r</i> paßt (Closure)
	<i>r</i> ?	0/1 Zeichenketten, auf die <i>r</i> paßt (Option)
*	<i>r</i> { <i>m</i> , <i>n</i> }	<i>m</i> – <i>n</i> Wiederholungen der Zeichenkette, auf die <i>r</i> paßt
*	<i>r</i> { <i>m</i> , }	<i>m</i> – ∞ Wiederholungen der Zeichenkette, auf die <i>r</i> paßt
*	<i>r</i> { <i>m</i> }	Genau <i>m</i> Wiederholungen der Zeichenkette, auf die <i>r</i> paßt
	<i>r</i> 1 <i>r</i> 2	Verkettung: Jede Zeichenkette <i>xy</i> , wo <i>r</i> 1 zu <i>x</i> und <i>r</i> 2 zu <i>y</i> paßt
	<i>r</i> 1 <i>r</i> 2	Alternative: Jede Zeichenkette, die zu <i>r</i> 1 oder <i>r</i> 2 paßt

14.3.4.2 POSIX-Zeichenklassen ([[:class:]])

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

14.3.5 Ausdrücke

<code>1 + (2 * exp(3) + sqrt(100)) ^ 2</code>	numerisch
<code>substr(\$0, 1, 4) "String"</code>	Zeichenkette
<code>i <= 10</code>	logisch
<code>user !~ /Dieter/ && (age >= 18 permission == "yes")</code>	logisch

14.4 Programmaufbau

14.4.1 Regel

```
MUSTER { AKTION }
```

14.4.1.1 Muster

Muster
BEGIN
END
Ausdruck /REGEXP/
Ausdruck, Ausdruck /REGEXP/, /REGEXP/

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)

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
:	Bedingte Operation
&&	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	␣	Whitespace = beliebig lange Folge von Leerzeichen und/oder Tabulatoren; am Zeilenanfang/ende vorkommender Whitespace wird ignoriert [field separator]
RS	\n	Zeilenvorschub [record separator]

14.4.4.2 Felder

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

14.4.4.3 Variablen

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

14.4.4.4 Arrays

```
arr[123] = "Text1"
arr["abc"] = "Text2"
```

```
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) ...
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	num	strnum
str	str	str	str
num	str	num	num
strnum	str	num	num

14.4.5 Operatoren

!C	Operator	Bedeutung
	(...)	Klammerung
*	\$	Feld-Zugriff (\$1, ...)
	++ --	Inkrementieren, Dekrementieren (<i>Präfix</i> und <i>Postfix</i>)
*r	^ **	Exponentiation x^y
	+ - !	<i>Unäres</i> Plus, <i>unäres</i> Minus, logisch NICHT
	* / %	Multiplikation, Division, Modulo (Divisionsrest)
	+ -	Addition, Subtraktion
*	(space)	Verkettung von Zeichenketten (<i>kein expliziter Operator!</i>)
	< >	Vergleich (<i>Zahlen und Zeichenketten</i>)
	<= >=	Vergleich (<i>Zahlen und Zeichenketten</i>)
	== !=	Vergleich (<i>Zahlen und Zeichenketten</i>)
*	~ !~	Vergleich mit Regulärem Ausdruck, negiert (<i>matched by</i>)
*	in	Index in Array testen
	&&	Logisch UND (<i>short-cut evaluation</i>)
		Logisch ODER (<i>short-cut evaluation</i>)
r	?:	Bedingte Operation (<i>cond ? true-case : false-case</i>)
r	= += -= *= /= %= ^= **=	Zuweisung (inkl. Operation + - * / % ^ **)

14.4.6 Vordefinierte Variablen und Arrays

Gawk	Variable	Bedeutung
*	ARGC ARGIND ARGV	Anzahl Kommandozeilenargumente (Programmname in ARGV[0]) Index der aktuell eingelesenen Eingabedatei in ARGV[1..ARGC-1] Array der Kommandozeilenargumente (ARGV[0..ARGC-1])
*	CONVFMT	Format f. autom. Umwandlung Zahl → Zeichenkette (Default: "%.6g")
*	ENVIRON	Assoziatives Array mit Umgebungsvariablen und ihrem Wert
*	ERRNO	Systemfehlermeldung bei <code>getline</code> und <code>close</code>
*	FIELDWIDTHS	Liste mit durch Leerz. getrennten festen Spaltenbreiten (statt FS)
	FILENAME	Name der aktuellen Eingabedatei
	FNR	Satznummer der aktuellen Eingabedatei
	FS	Eingabefeldtrenner (Default: " " = Whitespace)
*	IGNORECASE	Groß/Kleinschreibung beim Vergleich mit Regulärem Ausdruck und in Zeichenketten-Funktionen ignorieren (Default: false)
	NF	Anzahl Felder im aktuellen Satz
	NR	Anzahl bisher eingelesener Sätze insgesamt
	OFMT	Ausgabeformat für Zahlen in <code>print</code> (Default: "%.6g")
	OFS	Ausgabe-Feldtrenner in <code>print</code> (Default: " ")
	ORS	Ausgabe-Satztrenner in <code>print</code> (Default: "\n")
	RLENGTH	Länge des passenden Textes nach <code>match</code> (-1 falls nicht gefunden)
	RS	Eingabesatztrenner (Default: "\n")
	RSTART	Beginn des passenden Textes nach <code>match</code> (0 falls nicht gefunden)
*	RT	Zu RS passender Text nach dem Einlesen jedes Satzes
	SUBSEP	Trennzeichen für mehrdim. Index [<i>i, j, ...</i>] (Default: "\034")

14.4.7 Vordefinierte Funktionen

14.4.7.1 Arithmetik-Funktionen

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

14.4.7.2 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 [global]
	<code>index(s, t)</code>	Gibt erste Position (1.. <i>n</i>) von <i>t</i> in <i>s</i> zurück; oder 0 wenn <i>t</i> nicht in <i>s</i> enthalten ist
	<code>length([s])</code>	Gibt Länge von <i>s</i> zurück, ohne <i>s</i> wird \$0 verwendet
	<code>match(s, r)</code>	Überprüft, ob <i>s</i> eine zu <i>r</i> passende Teilkette enthält; gibt Position (1.. <i>n</i>) oder 0 wenn <i>r</i> nicht auf <i>s</i> paßt zurück; belegt RSTART und RLENGTH
*	<code>ord(s)</code>	Liefert ASCII-Code der 1. Zeichens von <i>s</i>
	<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 FS verwendet; <i>r</i> =" " zerlegt <i>s</i> in Buchstaben
	<code>sprintf(fmt, list)</code>	Gibt <i>list</i> formatiert gemäß <i>fmt</i> zurück (siehe <code>printf</code>)
	<code>sub(r, s[, t])</code>	Wie <code>gsub</code> , ersetzt aber nur die erste Teilkette
	<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

14.4.7.3 Ein/Ausgabe-Funktionen

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

14.4.7.4 getline-Rückgabewerte

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

14.4.7.5 printf-Formatumwandlungen

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

14.4.7.6 printf-Zusatzangaben zu Formatumwandlungen

Parameter	Bedeutung
-	Ausdruck linksbündig ausrichten
␣	Leerzeichen vor positiven Zahlen, - vor negativen
+	+ vor positiven Zahlen, - vor negativen
#	Alternative Form 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 auf Breite <i>width</i> ausdehnen (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

14.4.7.7 Zeit-Funktionen

Funktion	Bedeutung
<code>systeme()</code>	Aktuelle Zeit in Sekunden (<i>sec</i>) seit 1.1.1970 1:00 UTC
<code>strftime([<i>fmt</i>[, <i>sec</i>]])</code>	Sekunden <i>sec</i> gemäß <i>fmt</i> formatieren Defaultformat: "%a %b %d %H:%M:%S %Z %Y" Defaultzeit: <code>systeme()</code>

14.4.7.8 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

14.5 GAWK-Spezialdateien

Datei	Inhalt
/dev/pid	Aktuelle Prozeß-ID
/dev/ppid	Prozeß-ID des Elternprozesses
/dev/pgrp	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