

Kommandos

In diesem Teil werden die wichtigsten Kommandos vorgestellt. Die Kommandos werden in ihrer typischen Verwendung gezeigt und einige gebräuchliche Optionen dargestellt.

Die vollständige Beschreibung ist den jeweiligen Manual Seiten zu entnehmen.

- Dateiorientierte Kommandos (`more`, `od`, `tail`, ...)
- Verzeichnisorientierte Kommandos (`mkdir`, ...)
- Filesystemorientierte Kommandos (`mkfs`, ...)
- Programmorientierte Kommandos (`kill`, `nice`, `nohup`, ...)
- Sitzungsorientierte Kommandos (`su`, `newgrp`, ...)
- Remote Kommandos und Filetransfer (`rsh`, `rcp`, ...)

Inhalt

1.	Dateiorientierte Kommandos	7
1.1.	Dateiinhalte ansehen	7
1.1.1.	<code>more</code>	7

1.1.2.	cat	11
1.1.3.	head	13
1.1.4.	tail	14
1.1.5.	od	16
1.1.6.	hex	18
1.2.	Vergleich von Dateien	21
1.2.1.	comm	21
1.2.2.	cmp	23
1.2.3.	diff	26
1.3.	Modifikation von Dateien	31
1.3.1.	rev	31
1.3.2.	tr	32
1.4.	Umstrukturierung von Dateien	37
1.4.1.	uniq	37
1.4.2.	join	38

1.4.3.	sort.....	40
1.5.	Verschlüsselung von Dateien	46
1.6.	Suchen in Dateien	46
1.6.1.	grep, egrep, fgrep	46
1.7.	Informationen über Dateien	53
1.7.1.	df	53
1.7.2.	du.....	55
1.7.3.	find	57
1.7.4.	ls	78
1.7.5.	pwd	79
2.	Verzeichnisorientierte Kommandos	80
2.1.	mkdir	80
2.2.	rmdir	80
3.	Filesystemorientierte Kommandos	82
3.1.	mkfs.....	82

3.2.	mount	85
3.3.	umount.....	87
3.4.	Modifikation von Dateiattributen.....	90
3.4.1.	chmod	90
3.4.2.	chown	92
3.4.3.	chgrp	96
3.4.4.	ln.....	98
3.4.5.	mv.....	103
3.4.6.	touch.....	106
3.4.7.	rm	108
3.5.	Sichern und Zurückladen von Dateien	109
3.5.1.	cp	109
3.5.2.	cpio.....	112
3.5.3.	tar	119
3.5.4.	dd.....	122

3.5.5.	Konsistenzprüfung von Filesystemen.....	125
3.5.6.	fsck.....	125
3.5.7.	sync.....	128
4.	Sitzungsorientierte Kommandos	129
4.1.	Anmelden.....	129
4.1.1.	newgrp	129
4.1.2.	passwd	131
4.1.3.	su	131
4.2.	Informationen.....	134
4.2.1.	cal	134
4.2.2.	date	138
4.2.3.	id.....	143
4.2.4.	who.....	144
5.	Programmorientierte Kommandos.....	145
5.1.	kill	145

5.2. nice.....	147
5.3. nohup.....	149
6. Remote Kommandos und Filetransfer	151
6.1. telnet	151
6.2. ftp.....	152
6.3. r-Kommandos.....	154
6.3.1. rsh.....	156
6.3.2. rlogin	157
6.4. rcp	158

1. Dateiorientierte Kommandos

1.1. Dateiinhalte ansehen

1.1.1. more

BEZEICHNUNG

more - zeigt Dateien seitenweise

SYNTAX

more [-cdflsu] [-n] [+ Zeilennummer] [+ / Muster] [Name...]

BESCHREIBUNG

more gibt Textdateien seitenweise auf dem Bildschirm aus. Nach jeder Bildschirmseite wird die Ausgabe angehalten und auf eine Eingabe des Benutzers gewartet.

In der Shellvariablen MORE können Kommandozeilenoptionen für more gespeichert werden, die bei jedem Aufruf automatisch ausgeführt werden.

Am Dateiende wird more automatisch beendet.

Wenn more eine Datei liest, wird auf der letzten Bildschirmzeile im Prompt die prozentuale Position der aktuellen Bildschirmseite in der Datei angezeigt.

Wenn dieser Prompt angezeigt wird, erwartet more eine Eingabe des Benutzers. Die Eingaben bestehen in der Regel

aus einem einzigen Tastendruck. Einige Kommandos können mit einer Zahlenangabe N kombiniert werden.

Nleerzeichen

gibt die nächsten N Zeilen aus, oder einen kompletten Bildschirm, wenn keine Zahl angegeben ist

Nz das gleiche wie Leerzeichen; die Zahl N wird die neue Anzahl Zeilen pro Bildschirm

Ns überspringt die nächsten N Zeilen und zeigt die darauffolgenden Zeilen an

Nf überspringt die nächsten N Bildschirme, und zeigt die darauffolgenden Zeilen an

Nb springt N Bildschirmseiten rückwärts

Nctrl-B

das gleiche wie b

q oder Q

beendet more

= zeigt die aktuelle Zeilennummer an

v startet den Editor vi mit der aktuellen Datei in

der aktuellen Zeile

h das Hilfefkommando; gibt eine Übersicht über die more Kommandos

N/Ausdruck

sucht das N-te auftreten des Ausdrucks vom aktuellen Bildschirm an vorwärts

Nn sucht das N-te Auftreten des zuletzt gesuchten Ausdrucks vorwärts

' geht zurück an die Position, von der das letzte Suchkommando gestartet wurde

! Kommandozeile

startet eine Shell und führt die angegebene Kommandozeile aus

N:n springt zur nächsten Datei aus der Kommandozeilenliste, bzw. N Dateien weiter

N:p springt an den Anfang der aktuellen Datei oder in die vorhergehende Datei aus der Kommandozeilenliste bzw. N Dateien zurück

```
:f      zeigt den Namen der aktuellen Datei und die Posi-
        tion des aktuellen Bildschirms

:q oder :Q
        beendet more

.       (Punkt) wiederholt das letzte Kommando
```

OPTIONEN (siehe man pages)

Verwendung

Die Variable MORE wird gesetzt. Damit wird das Verhalten von more beeinflusst:

```
$ export MORE='-d'
$ more /etc/passwd
...
firewall:x:41:31:firewall account:/tmp:/bin/false
named:x:44:44:Nameserver Daemon:/var/named:/bin/bash
virtuoso:x:45:100:Virtuoso Admin:/opt/virtuoso-lite:/bin/bash
--Mehr-- (40%) [Leertaste zum Fortfahren, »q« zum Beenden.]
```

1.1.2. cat

BEZEICHNUNG

cat - gib Dateien aus

SYNTAX

```
cat      [-benstuvAET]      [--number]      [--number-nonblank]
         [--squeeze-blank]   [--show-nonprinting]  [--show-ends]
         [--show-tabs]     [--show-all] [Datei...]
```

BESCHREIBUNG

cat liest beliebige Dateien und schreibt sie ohne Veränderung in die Standardausgabe. Durch Umlenkung der Ausgabe auf eine Datei können so Dateien verkettet werden. Außerdem wird cat häufig benutzt, um Dateien an Programme zu übergeben, die nur von der Standardeingabe lesen. (Solche Programme werden im allgemeinen als Filter bezeichnet.) Für die Übergabe steht entweder die Ausgabeumlenkung der Shell mit > und >> zur Verfügung, oder die Ausgabe wird durch eine Pipeline | an den Filter weitergeleitet.

...

Verwendung

Zusammenfügen mehrerer Dateien:

```
$ cat Kap1 Kap2 Kap3 > Buch
$
```

Einfügen einer Zeile am Dateianfang:

```
$ cat .profile
set -o vi
alias ll='ls -al'
$
$ cat - .profile > tmpDat
tput clear
CTR_D
$ mv tmpDat .profile
$ cat .profile
tput clear
set -o vi
alias ll='ls -al'
$
```

cat als Editor:

```
$ cat > Text
Dies ist ein Text
mit 2 Zeilen
CTR_D
$
```

1.1.3. head

BEZEICHNUNG

head schreibt den Anfang einer Datei auf die Standardausgabe

SYNTAX

```
head [-c Anzahl [bkm]] [-n Anzahl] [-qv] [--bytes= Anzahl
[bkm]] [--lines= Anzahl] [--quiet] [--silent] [--verbose]
[Datei...]
```

```
head [-nr] [-bcklmqv] [Datei...]
```

BESCHREIBUNG

head schreibt die ersten (10) Zeilen von der Datei auf den Bildschirm. Wenn keine Datei oder - angegeben wird, liest head von der Standardeingabe. Wird mehr als eine Datei angegeben, so wird der Dateiname, in ==> und <== eingeschlossen, der Ausgabe vorangestellt.

OPTIONEN (siehe man pages)

Verwendung

Die ersten 5 Zeilen der Datei /etc/passwd:

```
$ head -2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
$
```

1.1.4. tail

BEZEICHNUNG

tail - zeigt das Ende einer Datei

SYNTAX

```
tail [-c [+] N [bkm]] [-n [+] N] [-fqv] [--bytes= [+] N
[bkm]] [--lines= [+] N] [--follow] [--quiet] [--silent]
[--verbose] [Datei...]
```

```
tail [{-,+}Nbcfklmqv] [Datei...]
```

BESCHREIBUNG

tail druckt die letzten (10) Zeilen einer Datei oder von der Standardeingabe, wenn keine Datei angegeben wird. Ein einzelnes - anstelle eines Dateinamens meint ebenfalls die Standardeingabe. Werden mehrere Dateien angegeben, so wird das Ende jeder Datei mit dem Dateinamen eingeschlossen in ==> und <== eingeleitet.

OPTIONEN (siehe man pages)

Verwendung

Die letzte Zeile der Datei /etc/passwd:

```
$ tail -1 /etc/passwd
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
$
```

Durch die Option `-f` kann eine Datei „überwacht“ werden:

1.1.5. od

BEZEICHNUNG

od - (octal dump) zeigt Dateien im oktalen, hexadezimalen oder in anderen Formaten an.

SYNTAX

```
od [-abcdfhiloxv] [-s [Länge]] [-w [Anzahl]] [-A Posi-
tionsformat] [-j Anzahl] [-N Anzahl] [-t Format] [--skip-
bytes= Anzahl] [--address-radix= Positionsformat] [--read-
bytes= Anzahl] [--format= Format] [--output-duplicates]
[--strings[= Anzahl]] [--width[= Anzahl]] [Datei...]
```

BESCHREIBUNG

od liest die angegebenen Dateien oder die Standardeingabe (wenn keine Datei oder anstelle einer Datei - angegeben ist), und gibt die Bytes formatiert und kodiert auf die Standardausgabe.

Jede Ausgabezeile enthält in der ersten Spalte die Positionsnummer des ersten in der Zeile dargestellten Bytes (vom Dateianfang gezählt). In den darauffolgenden Spalten werden die Daten aus der Datei in einem durch die Optionen kontrollierten Format angezeigt.

In der Standardeinstellung ohne Optionen gibt od die Position als 7-stellige Oktalzahl und die Daten in 8 Spalten zu je zwei Bytes in oktaler Darstellung aus.

OPTIONEN (siehe man pages)

Verwendung

Ausgabe mit verschiedenen Optionen -c:

```
$ cat text
1234567890
abcdefghij
$ od -c text
0000000  1   2   3   4   5   6   7   8   9   0  \n  a   b   c   d   e
0000020  f   g   h   i   j  \n
0000026
$
```

1.1.6. hex

```
NAME
    hex.c - dump a file to stdout in CP/M-like hex format
SYNOPSIS
    hex [-V] [-w] [-e] [-c] [-g] [-s] [filename...]
DESCRIPTION
    This program provides a facility for obtaining a dump of a
    binary file to stdout.  The default format is a CP/M style
    hex dump with byte offset in file, 16 bytes of hex and 16
    bytes of alpha representation with '.' for non-printables
    per line.  If no filename is given, hex reads from standard
    input.

    The program accepts the following options to control its
    output:

    -w -- set # of bytes dumped per line (default is 16)
```

```
-e -- display EBCDIC character assignments rather than
ASCII

-c -- display printable characters as text in-line

-g -- don't output mid-page gutter

-s -- dump file section

-V -- print version number and exit
```

The command line is scanned left-to-right for filenames and options, and each file dumped according to the format defined by preceding options. To turn off an option, precede the letter with a +. Options that take arguments may have them immediately following the option letter or whitespace-separated.

If the command-line arguments include two or more filenames (with - being regarded as a 'filename' for standard input), hex will emit a one-line header giving the name the file in front of each input file.

The -s option expects a start offset, optionally followed by a comma-separated count. Each of these should be a dec-

imal or hex integer (hex must be prefixed by x or h; leading zeros will be ignored).

The `-w` option expects a width, formatted as a single decimal or hex number in the style of an `-s` option argument. Odd widths turn on the `-g` option.

Verwendung

```
$ cat text
1234567890
abcdefghij
$ hex text
0000  31 32 33 34 35 36 37 38  39 30 0a 61 62 63 64 65  12345678 90.abcde
0010  66 67 68 69 6a 0a                                fghij.
$
```

1.2. Vergleich von Dateien

1.2.1. comm

BEZEICHNUNG

comm - vergleicht **zwei sortierte** Dateien

SYNTAX

comm [-{1,2,3}] Datei1 Datei2

BESCHREIBUNG

comm vergleicht zwei sortierte Dateien, und gibt die gemeinsamen und die verschiedenen Zeilen jeweils in Spalten aus. Die erste Spalte enthält die Zeilen, die nur in Datei1 enthalten sind. Die zweite Spalte enthält die Zeilen, die in beiden Dateien enthalten sind. Die dritte Spalte enthält schließlich die Zeilen, die nur in der zweiten Datei enthalten sind.

Ein - anstelle eines Dateinamens steht für die Standard-eingabe.

```
OPTIONEN (siehe man pages)
  -{1,2,3}
```

Verwendung

Comm erzeugt folgende Ausgabe:

Spalte 1	Spalte 2	Spalte 3
Zeilen, die nur in Datei 1 vorkommen	Zeilen, die nur in Datei 2 vorkommen	Zeilen, die in beiden Dateien vorkommen

```

$ cat Datei1
gemeinsam
*****
1111111111
$
$ cat Datei2
=====
2222222222
gemeinsam
$
$ comm Datei1 Datei2
      =====
      2222222222
                                gemeinsam
*****
1111111111
$

```

1.2.2. cmp

BEZEICHNUNG

cmp - (compare) vergleicht zwei Dateien **byteweise**

SYNTAX

```

cmp [-cls] [--show-chars] [--verbose] [--silent] [--quiet]
    Datei1 [Datei2]

```

BESCHREIBUNG

cmp vergleicht zwei (binäre) Dateien und liefert die dezi-

male Position des ersten Bytes, in dem sich die Dateien unterscheiden, sowie die Zeilennummer zurück. Wird statt eines Dateinamens - angegeben, so wird von der Standard-eingabe gelesen; wird nur eine Datei benannt, so wird anstelle der zweiten ebenfalls von der Standard-eingabe gelesen.

OPTIONEN (siehe man pages)

Verwendung

```

$ cat d1
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
$
$ cat d2
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
$
$ cmp d1 d2
$ echo $?
0
$
$ cmp d1 d3
d1 d3 differ: char 4, line 1
$ echo $?
1
$

```

Durch die Option `-l` erfolgt die Ausgabe der Differenzen in der Form:

Nummer des Byte	Oktaler Wert in Datei1	Oktaler Wert in Datei2
-----------------	------------------------	------------------------

```
$ cat d1
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
$
$ cat d3
1 2-3 4 5 6 7 8 9 0
a b c-d e f g h i j
$
$ cmp -l d1 d3
      4  40  55
     26  40  55
$
```

1.2.3. diff

```
NAME
    diff - find differences between two files
SYNOPSIS
    diff [options] from-file to-file
DESCRIPTION
    In the simplest case, diff compares the contents of the
    two files from-file and to-file. A file name of - stands
    for text read from the standard input. As a special case,
    diff - - compares a copy of standard input to itself.

    If from-file is a directory and to-file is not, diff com-
    pares the file in from-file whose file name is that of to-
```

file, and vice versa. The non-directory file must not be
-.

If both from-file and to-file are directories, diff compares corresponding files in both directories, in alphabetical order; this comparison is not recursive unless the -r or --recursive option is given. diff never compares the actual contents of a directory as if it were a file. The file that is fully specified may not be standard input, because standard input is nameless and the notion of ``file with the same name'' does not apply.

diff options begin with -, so normally from-file and to-file may not begin with -. However, -- as an argument by itself treats the remaining arguments as file names even if they begin with -.

Options

Verwendung

Unterschiede zwischen Dateien:

```

$ cat d1
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
$
$ cat d2
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
$
$ diff d1 d2
$ echo $?
0
$ cat d3
1 2-3 4 5 6 7 8 9 0
a b c d e f g h i j
x x x x x x x x x x
$
$ diff d1 d3
1c1
< 1 2 3 4 5 6 7 8 9 0
---
> 1 2-3 4 5 6 7 8 9 0
2a3
> x x x x x x x x x x
$

```

Das diff Kommando erlaubt es, von verschiedene Versionen einer Datei nur die Unterschiede zu speichern und den jeweiligen Stand wider herzustellen:

1. Uhrversion

```
$ cat prog.1.c
#include <stdio.h>
main() {
    printf("Hello, world\n");
}
$
```

2. Änderung

```
$ cat prog.1.1.c
#include <stdio.h>
main() {
    printf("Hello, world\n Version 1.1\n");
}
$
```

3. Delta ermitteln

```
$ diff -eb prog.1.c prog.1.1.c > delta
$
$ cat delta
3c
    printf("Hello, world\n Version 1.1\n");
.
$
```

4. Löschen der geänderten Version

```
$ rm prog.1.1.c
$ ls
d1  d2  d3  delta  prog.1.c
$
```

5. Wiederherstellen der geänderten Version

```
$ echo "w prog.1.1.c" >> delta
$ ed - prog.1.c < delta
$
$ ls
d1  d2  d3  delta  prog.1.1.c  prog.1.c
$
$ cat prog.1.1.c
#include <stdio.h>
main() {
    printf("Hello, world\n Version 1.1\n");
}
$
```

1.3. Modifikation von Dateien

1.3.1. rev

NAME

rev - reverse lines of a file

SYNOPSIS

rev [file]

DESCRIPTION

The rev utility copies the specified files to the standard output, reversing the order of characters in every line. If no files are specified, the standard input is read.

Verwendung

```
$ cat d
123456
abcde
$
$ rev d
654321
edcba
$
```

1.3.2. tr

NAME

tr - translate or delete characters

SYNOPSIS

tr [OPTION]... SET1 [SET2]

DESCRIPTION

Translate, squeeze, and/or delete characters from standard input, writing to standard output.

-c, --complement

first complement SET1

-d, --delete

delete characters in SET1, do not translate

-s, --squeeze-repeats

replace sequence of characters with one

```
-t, --truncate-set1
    first truncate SET1 to length of SET2
```

```
--help display this help and exit
```

```
--version
    output version information and exit
```

SETs are specified as strings of characters. Most represent themselves. Interpreted sequences are:

```
\NNN character with octal value NNN (1 to 3 octal digits)
```

```
\\ backslash
```

```
\a audible BEL
```

```
\b backspace
```

```
\f form feed
```

```
\n new line
```

```
\r return
```

```
\t    horizontal tab
```

```
\v    vertical tab
```

```
...
```

Verwendung

Ersetzen von Zeichen durch korrespondierende Stelle:

```
$ cat d1  
Text mit Ziffern  
a b c d e f g  
$  
$ tr abcd 1234 < d1  
Text mit Ziffern  
1 2 3 4 e f g  
$
```

Ersetzen von Zeichen durch Bereiche:

```
$ cat d1
Text mit Ziffern
a b c d e f g
$
$ tr [a-d] [1-4] < d1
Text mit Ziffern
1 2 3 4 e f g
$
```

Ersetzen, wenn Strings unterschiedlich lang sind (letztes Zeichen von String2 wird verwendet):

```
$ cat d1
Text mit Ziffern
a b c d e f g
$ tr abcde 12 < d1
T2xt mit Ziff2rn
1 2 2 2 2 f g
$
```

Löschen von Zeichen:

```
$ cat d1
Text mit Ziffern
a b c d e f g
$
$ tr -d "ab" < d1
Text mit Ziffern
  c d e f g
$
```

Löschen aller Klein-Buchstaben:

```
$ cat d1
Text mit Ziffern
a b c d e f g
$
$ tr -d "[a-z]" < d1
T  Z
$
```

Löschen aller Groß-Buchstabe und Blanks:

```
$ tr -d "[A-Z]" < d1
extmitiffen
abcdefg
$
```

1.4. Umstrukturierung von Dateien

1.4.1. uniq

BEZEICHNUNG

uniq - löscht doppelte Zeilen aus einer sortierten Datei

SYNTAX

```
uniq [-cdu] [-f Nummer] [-s Nummer] [-w Nummer] [-.IRNum-
mer] [+.IRNummer] [--count] [--repeated] [--unique]
[--skip-fields= Nummer] [--skip-chars= Nummer] [--check-
chars= Nummer] [Eingabedatei] [Ausgabedatei]
```

OPTIONEN (siehe man pages)

Verwendung

Löschen von benachbart gleichen Zeilen:

```
$ cat d1
Dieser Text enthaellt 3 gleiche Zeilen
Zeile 123
123 Zeile
Zeile 123
Zeile 123
$
$ uniq < d1
Dieser Text enthaellt 3 gleiche Zeilen
Zeile 123
123 Zeile
Zeile 123
$
$ sort d1 | uniq
123 Zeile
Dieser Text enthaellt 3 gleiche Zeilen
Zeile 123
$
```

1.4.2. join

BEZEICHNUNG

join - verknüpft zwei Dateien nach Schlüsselfeldern

SYNTAX

```
join [-a 1|2] [-v 1|2] [-e Zeichenkette] [-o
Feldliste...] [-t Buchstabe] [-j [1|2] Feldnr] [-1
Feldnr] [-2 Feldnr] Datei1 Datei2
```

BESCHREIBUNG

join verknüpft zwei (alphabetisch) sortierte Dateien, indem je zwei Zeilen mit identischen Schlüsselfeldern zu einer Ausgabezeile verbunden werden.

Die Schlüsselfelder sind durch Leerzeichen voneinander getrennt. Führende Leerzeichen werden ignoriert. Wenn nicht anders angegeben, ist das erste Feld einer jeden Zeile Schlüsselfeld. Die Ausgabefelder sind ebenfalls durch Leerzeichen voneinander getrennt. Die Ausgabe besteht aus dem Schlüsselfeld, gefolgt von den übrigen Feldern der Datei1 und schließlich aller Felder der passenden Zeilen von Datei2 ohne das Schlüsselfeld.

OPTIONEN (siehe man pages)

Verwendung

```
$ cat telefon
elke 123
evi 124
hans 125
karl 126
$
$ cat cv
elke C Java
evi Cobol PS440
hans C++
karl C
heidi JavaScript
$
$ join telefon cv
elke 123 C Java
evi 124 Cobol PS440
hans 125 C++
karl 126 C
$
```

1.4.3. sort

BEZEICHNUNG

sort - sortiert die Zeilen einer Textdatei

SYNTAX

```
sort [-cmus] [-t Separator] [-o Ausgabedatei] [-bdfiMnr]
[+ POS1 [- POS2]] [-k POS1 [, POS2]] [Datei...]
```

BESCHREIBUNG

sort wird normalerweise zum Sortieren von Dateien verwendet. Es kann aber auch Dateien daraufhin überprüfen, ob sie sortiert sind; oder mehrere sortierte (oder unsortierte) Dateien zu einer sortierten zusammenfügen.

Dazu existieren drei Modi:

- der check Modus, der prüft, ob eine Datei bereits sortiert ist. Dieser Modus wird durch die Option -c eingeleitet
- der merge Modus, der mehrere vorsortierte Dateien zusammenfügt. Die Dateien so zusammenzufügen ist schneller, als sie komplett sortierten zu lassen. Dieser Modus wird durch die Option -m eingeleitet
- der Standardmodus ist sort

Wenn Schlüsselfelder bezeichnet sind, vergleicht sort die Schlüsselfelder in der Reihenfolge ihrer Bezeichnung, bis ein Unterschied gefunden wurde oder keine weiteren Felder vorhanden sind.

Wenn eine der globalen Optionen Mbdfinr benutzt wird, und

kein Schlüsselfeld angegeben ist, vergleicht sort die ganzen Zeilen.

OPTIONEN (siehe man pages)

Verwendung

Sortieren ohne Angabe von Schlüsselfeldern:

```
$ cat telefon
elke 23
hans 525
karl 426
elke 123
evi 124
$
$ sort telefon
elke 123
elke 23
evi 124
hans 525
karl 426
$
```

Prüfen, ob sortiert:

```
$ cat telefon
elke 23
hans 525
karl 426
elke 123
evi 124
$
$ sort -c telefon
sort: telefon:4: disorder: elke 123
$ echo $?
1
$
```

Sortieren nach Feldern von Pos1 bis Pos2 (Default Trennsymbol ' ', alternativ -t):

```
$ sort +1 -2 telefon
elke 123
evi 124
elke 23
karl 426
hans 525
$
```

Sortieren mit **numerischem** Schlüssel:

```
$ cat telefon
elke 23
hans 525
karl 426
elke 123
evi 124
$
$ sort +1n -2 telefon
elke 23
elke 123
evi 124
karl 426
hans 525
$
```

Sortieren nach mehreren Schlüsselfeldern:

```
$ cat telefon
elke 23
hans 525
karl 426
elke 123
evi 124
$
$ sort +0 -1 +1n -2 telefon
elke 23
elke 123
evi 124
hans 525
karl 426
$
```

Merg sortierter Dateien:

```
$ cat d1
123
234
345
$
$ cat d2
456
567
789
890
$
$ sort -m d1 d2
123
234
345
456
567
789
890
$
```

1.5. Verschlüsselung von Dateien

1.6. Suchen in Dateien

1.6.1. grep, egrep, fgrep

BEZEICHNUNG

grep - durchsucht Dateien nach Ausdrücken

SYNTAX

```
grep [-CVbchilnsvwx] [-Anzahl] [-AB Anzahl] [[-e] Ausdruck  
|-f Datei] [Datei...]
```

BESCHREIBUNG

grep durchsucht die angegebenen Dateien (oder die Standardeingabe) nach einem Ausdruck und gibt die entsprechenden Zeilen aus. Der Status von grep ist 0, wenn der Ausdruck gefunden wurde und sonst 1.

Als Ausdruck akzeptiert grep reguläre Ausdrücke mit den folgenden Steuerzeichen:

- c ein einzelner Buchstabe paßt auf sich selbst
- . ein Punkt paßt auf jeden Buchstaben außer auf das Zeilenende
- \? das dem Operator \? vorangehende Muster kann null oder einmal vorkommen
- * das dem Operator * vorangehende Muster kann 0 mal oder öfter vorkommen
- \+ das dem Operator \+ vorangehende Muster kann einmal oder öfter vorkommen

`\|` die durch den Operator `\|` verbundenen Argumente werden oder verknüpft

`^` (Caret) paßt auf den Zeilenanfang

`$` paßt auf das Zeilenende

`\<` paßt auf den Wortanfang

`\>` paßt auf das Wortende

[Buchstaben]

paßt auf alle Buchstaben; dabei können einzelne Buchstaben, aber auch Bereiche in der Form von-bis angegeben werden; wenn der erste Buchstabe nach [ein ^ ist, paßt der Ausdruck auf alle Buchstaben, außer den Aufgeführten

`\(\)` die Klammern fassen Ausdrücke zusammen; außerdem wird der auf den in Klammern eingeschlossene Teil des Musters passende Text markiert und mit einem folgenden `\N` Ausdruck referenziert (Tag)

`\N` referenziert die auf das in der N-ten runden Klammern eingeschlossene Muster passende Zeichenkette.

`\` jedes der Sonderzeichen kann, durch einen `\` (Backslash) eingeleitet, sich selbst suchen

`\b` paßt auf kein Zeichen, sondern auf den Anfang oder das Ende eines Wortes

`\B` symbolisiert den Raum innerhalb eines Wortes

`\w` paßt auf alle alphanumerischen Zeichen `[A-Za-z0-9]`

`\W` paßt auf alle nichtalphanumerischen Zeichen `[^A-Za-z0-9]`

Die Rangfolge der Operatoren ist (von der höchsten zur niedrigsten): `(,), ?, *, +` und `|`. Die anderen Operatoren sind mit den anderen Buchstaben gleichrangig.

OPTIONEN (siehe man pages)

BEISPIEL

Mit dem Kommando

```
grep '\( .... \).+\1' Handbuch.tex
```

wird in der Datei `Handbuch.tex` nach Zeilen gesucht, in denen ein Wort mit vier Buchstaben doppelt, aber nicht unmittelbar nach dem ersten vorkommt.

grep Global regular expression

Muster sind eingeschränkte reguläre Ausdrücke

fgrep fast grep

Muster sind nur konstante Zeichenketten (-> schneller)

egrep extended grep

besserer Algorithmus zum Finden der Muster und erweiterte reguläre Ausdrücke

Verwendung

Suchen aller Dateien im aktuellen Verzeichnis, die den String main enthalten:

```
$ grep -l "main(" *.c  
prog.1.1.c  
prog.1.c
```

Alle Zeilen Suchen, die auf n oder N enden:

```
$ cat d1
Dies ist ein Zeile mit n
Dies ist ein Zeile mit n.
Dies ist ein Zeile mit N
Dies ist ein Zeile mit N$
$
$ grep -i "n$" d1
Dies ist ein Zeile mit n
Dies ist ein Zeile mit N
$
```

Alle Zeilen die eine nicht leere Folge von „.“ enthalten:

```
$ cat d2
Zeile ohne Punkt
Zeile mit Punkt.
Zeile mit Punktfolge .....
$
$ egrep "(\\.)+" d2
Zeile mit Punkt.
Zeile mit Punktfolge .....
$
$ grep "(\\.)+" d2
$
```

Alle Zeilen, die mit mehr als zwei Ziffern beginnen:

```
$ cat d3
Zeile1
2. Zeile
12 letzte Zeile
$
$ egrep '^[0-9][0-9]' d3
12 letzte Zeile
$
```

Dateien mit vi bearbeiten, die printf enthalten:

```
$ vi `grep -l printf *.c`
```

1.7. Informationen über Dateien

1.7.1. df

BEZEICHNUNG

df - (disk free) zeigt den freien Festplattenplatz

SYNTAX

```
df [-aikPv] [-t fstype] [--all] [--inodes] [--type fstype]
  [--kilobytes] [--portability] [Pfad...]
```

BESCHREIBUNG

df zeigt den freien Festplattenplatz für das Dateisystem in dem das Verzeichnis Pfad angesiedelt ist. Wenn kein Verzeichnis angegeben ist, wird der freie Platz für alle aufgesetzten Dateisysteme angezeigt. Die Angabe erfolgt in Kilobyte, wenn nicht die Umgebungsvariable POSIXLY_CORRECT gesetzt ist. In diesem Fall wird der freie Platz in 512-Byte Sektoren angezeigt.

Wird als Pfad der absolute Name der Gerätedatei eines aufgesetzten Dateisystems angegeben, so wird der freie Platz auf diesem Gerät (Partition einer Festplatte) angegeben und nicht der des Dateisystem, in dem sich die Gerätedatei befindet. Der freie Platz auf einem abgesetzten Dateisystem ist auf diese Weise nicht zu ermitteln.

- a (all) zeigt alle Dateisysteme an, einschließlich der mit 0 Bytes Kapazität und der vom Typ ignore oder auto
- i (inodes) zeigt die Auslastung der Inodes anstelle der Blockauslastung
- k (kilobytes) zeigt den freien Platz in Kilo-byteblöcken, auch wenn die Umgebungsvariable POSIXLY_CORRECT gesetzt ist
- P (portability) erzwingt die Ausgabe in einer Zeile pro Dateisystem
- t Fstyp
(type) schränkt die Ausgabe auf die Dateisysteme vom Typ Fstyp ein

Verwendung

```

$ df
Dateisystem          1k-Blöcke    Benutzt Verfügbar Ben% montiert auf
/dev/hda3            19534576    5398792 13143372 30% /
/dev/hda1             23300       2739     19358 13% /boot
shmfs                 63300        0        63300 0% /dev/shm
$
$ df -m
Dateisystem          1M-Blöcke    Benutzt Verfügbar Ben% montiert auf
/dev/hda3            19077        5273     12835 30% /
/dev/hda1              23           3         18 13% /boot
shmfs                  62           0         61 0% /dev/shm
$

```

1.7.2. du

BEZEICHNUNG

du - (disk usage) zeigt die Verteilung des belegten Plattenplatzes auf die Verzeichnisse

SYNTAX

```

du [-abcklsxDLS] [--all] [--total] [--count-links] [--summarize]
    [--bytes] [--kilobytes] [--one-file-system]
    [--separate-dirs] [--dereference] [--dereference-args]
    [Verzeichnis...]

```

BESCHREIBUNG

du zeigt den belegten Plattenplatz für das Verzeichnis und für alle Unterverzeichnisse (in Kilobyte). Wenn die Umgebungsvariable `POSIXLY_CORRECT` gesetzt ist, wird die Menge

in 512 Byte Blöcken angegeben.

OPTIONEN (siehe man pages)

Verwendung

Anzeige der User, die den meisten Platz belegen:

```
linux:/users # du -cm | sort -n -r | head -8
702      insgesamt
702      .
258      ./as
253      ./opt
248      ./as/OpenOffice.org638
206      ./as/OpenOffice.org638/program
129      ./opt/mico
92       ./opt/mico/mico_src
linux:/users #
```

1.7.3. find

BEZEICHNUNG

find - sucht nach bestimmten Dateien

SYNTAX

```
find [Verzeichnis] [-Option...] [-Test...] [-Aktion...]
```

BESCHREIBUNG

find durchsucht eine oder mehrere Verzeichnishierarchien nach Dateien mit bestimmten Eigenschaften, und führt damit bestimmte Aktionen aus. Die Eigenschaften können durch Tests bestimmt werden.

Optionen, Tests und Aktionen können mit Operatoren zusammengefaßt werden. find bewertet für jede Datei in den Verzeichnishierarchien die Optionen, Tests und Aktionen von links nach rechts, bis ein falscher Wahrheitswert auftaucht oder die Kommandozeilenargumente zu Ende sind.

Das erste Argument, das mit einem -, einer Klammer (,), einem Komma , oder einem Ausrufezeichen ! beginnt, wird als Anfang einer Option oder Test interpretiert. Alle Argumente davor werden als Verzeichnisnamen interpretiert.

Wenn kein Verzeichnis angegeben ist, wird das aktuelle Verzeichnis genommen. Wenn keine Aktion angegeben ist, wird die Aktion -print ausgeführt.

Der Status von find ist Null, wenn alle Aktionen erfolgreich waren, im Fehlerfall ist der Status größer als Null.

OPTIONEN (siehe man pages)

Die Optionen bestimmen das allgemeine Verhalten des Kommandos, und beziehen sich nicht auf spezielle Dateien. Die Optionen sind immer wahr.

-daystart

mißt die Zeiten für die -amin, -atime, -cmin, -ctime, -mmin und -mtime Eigenschaften vom Beginn des aktuellen Tages anstelle der letzten 24 Stunden

-depth bearbeitet den Inhalt jedes Verzeichnisses vor dem Verzeichnis selbst

-follow

folgt den symbolischen Links; diese Option schließt -noleaf mit ein

-maxdepth Ebenen

steigt bis zu der gegebenen Zahl von Ebenen im Verzeichnisbaum auf (in der Hierarchie ab); bei 0 Ebenen werden die Tests nur auf die in der Kommandozeile übergebenen Dateien und Verzeichnisnamen angewendet

-mindepth Ebenen

steigt mindestens die gegebene Zahl von Ebenen im Verzeichnisbaum auf (in der Hierarchie ab); bei einer Ebene werden die in der Kommandozeile genannten Dateien und Verzeichnisnamen nicht bearbeitet

-noleaf

erzwingt die Bearbeitung aller Verzeichniseinträge; normalerweise kann davon ausgegangen werden, daß jedes Linuxverzeichnis wenigstens zwei (harte) Links enthält: das Verzeichnis . ist ein Link auf das Verzeichnis selbst, und jedes Unterverzeichnis enthält den Eintrag .. als Link auf das Oberverzeichnis; wenn find bei der Untersuchung eines Verzeichnisses zwei Unterverzeichnisse weniger untersucht hat, als das Verzeichnis Links zählt, kann deshalb normalerweise die weitere Suche beendet werden

-version

gibt die Versionsnummer auf die Standardfehlerausgabe

-xdev durchsucht keine Verzeichnisse in anderen Dateisystemen (auf anderen Partitionen)

Tests:

Alle numerischen Argumente können auf drei Arten angegeben werden:

+ N steht für alle Zahlen größer als N

- N steht für alle Zahlen kleiner als N

N steht für genau N

Alle Tests werden auf die Dateien in den angegebenen Verzeichnissen einzeln angewendet. Die Tests liefern einen Wahrheitswert von 0 (Wahr), wenn der Test erfolgreich war.

Die Tests auf die erweiterten Zeitmarken (Zugriff und Erstellung) werden nur in solchen Verzeichnissen korrekt behandelt, die auf einem der neuen Linuxdateisysteme angesiedelt sind (e2fs, xiafs, new minix). Auf den anderen Dateisystemen wird nur das Datum der letzten Änderung zuverlässig getestet. Das Ergebnis der anderen Tests hängt davon ab, ob der letzte Zugriff bzw. die letzte Änderung so kurz zurückliegen, daß die veränderte I-Node noch im Arbeitsspeicher (Cache) ist. Dann können auch für die Dateien der alten Dateisysteme alle drei Zeitmarken unterschieden werden.

`-amin N`
auf die Datei ist vor N Minuten zugegriffen worden

`-anewer Referenzdatei`
auf die Datei ist vor weniger Zeit zugegriffen worden, als seit der letzten Veränderung der Referenzdatei vergangen ist; im Zusammenhang mit `-follow` tritt `-anewer` nur in Effekt, wenn `-follow` vor `-anewer` in der Kommandozeile steht

`-atime N`
auf die Datei ist vor $N \cdot 24$ Stunden zugegriffen worden

`-cmin N`
der Status der Datei wurde vor N Minuten geändert

`-cnewer Referenzdatei`
der Status der Datei wurde vor weniger Zeit verändert, als seit der letzten Veränderung der Referenzdatei vergangen ist; zusammen mit `-follow` tritt `-cnewer` nur in Effekt, wenn `-follow` vor `-cnewer` in der Kommandozeile steht

`-ctime N`
der Dateistatus wurde vor $N \cdot 24$ Stunden geändert

-empty die reguläre Datei oder das Verzeichnis ist leer

-false ist immer falsch

-fstype Typ
die Datei ist in einem Dateisystem vom angegebenen Typ; unter anderem werden minix, msdos, ext und proc erkannt

-gid N die Datei gehört der Gruppe mit der Kennzahl N

-group Name
die Datei gehört der Gruppe Name

-inum IN
die Datei belegt die Inode mit der Nummer N

-links N
die Datei hat N (harte) Links

-lname Muster
die Datei ist ein symbolischer Link auf eine Datei oder ein Verzeichnis mit einem zum Muster passenden Namen

-mmin N

der Inhalt der Datei wurde vor N Minuten verändert

`-mtime N`

der Inhalt der Datei wurde vor IN*24 Stunden verändert

`-name Muster`

der Name der Datei paßt zu dem Muster

`-newer Referenzdatei`

die Datei ist später verändert worden als die Referenzdatei; zusammen mit `-follow` tritt `-newer` nur in Effekt, wenn `-follow` vor `-newer` in der Kommandozeile steht

`-nouser`

die Datei gehört keinem im System eingetragenen Benutzer

`-nogroup`

die Datei gehört keiner im System angemeldeten Gruppe

`-path Muster`

der Pfadname der Datei paßt zum Muster"

`-perm Modus`
die Zugriffsrechte auf die Datei entsprechen exakt dem Modus; der Modus kann als Oktalzahl oder mit den bei `chmod(1)` beschriebenen Kennungen beschrieben werden, die Kennungen werden auf Modus 000 bezogen

`-perm -Modus`
(mindestens) die Zugriffsrechte für den Modus sind gesetzt

`-perm +Modus`
die Zugriffsrechte entsprechen höchstens dem Modus (oder sind weiter eingeschränkt)

`-regex Muster`
der Pfadname paßt zu dem regulären Ausdruck Muster

`"size N[{c,k}]"`
die Datei belegt N Datenblöcke zu 512 Bytes, bzw. N Bytes und N Kilobytes mit nachgestelltem c oder k

`-true` ist immer wahr

`-type C`
die Datei ist vom Typ C; folgende Typen werden

unterschieden:

b gepufferte Gerätedatei für ein blockorientiertes Gerät

c ungepufferte Gerätedatei für ein zeichenorientiertes Gerät

d Verzeichnis

p benannte Pipeline (FiFo)

f normale Datei

l symbolischer Link

s Socket

-uid N die Kennziffer des Eigentümers ist N

-used N

auf die Datei ist N Tage nach der letzten Änderung
zugegriffen worden

-user Name

die Datei gehört dem Anwender Name

-xtype C

das gleiche wie -type für alle Dateien, die keine symbolischen Links sind; wenn die Datei ein symbolischer Link ist und die Option -follow nicht gesetzt ist, wird die Datei, auf die der Link zeigt, auf den Typ C geprüft; wenn die Option -follow gesetzt ist, ist der Test wahr, wenn C = 1 ist

Aktionen:

-exec Kommando;

führt das Kommando aus; die Aktion ist wahr, wenn das Kommando einen Status von Null liefert; alle auf den Kommandonamen folgenden Argumente bis zu einem Semikolon ; werden als Kommandozeilenargumente für das Kommando interpretiert; das Semikolon kann nicht weggelassen werden, und es muß durch mindestens ein Whitespace von der letzten Option getrennt werden; die Konstruktion {} wird durch den Pfadnamen der Datei ersetzt; die Klammern und das Semikolon müssen in der Kommandozeile für find quotiert werden, damit sie nicht von der Shell bearbeitet werden

-fprint Ausgabedatei

schreibt den Pfadnamen der getesteten Datei in die

Ausgabedatei; wenn die Ausgabedatei nicht existiert, wird sie erzeugt, sonst wird sie erweitert; die Standardausgabe und die Standardfehlerausgabe werden als /dev/stdout und /dev/stderr angesprochen

-fprintf0 Ausgabedatei

schreibt den Namen der getesteten Datei in die Ausgabedatei und schließt die Ausgabe mit einem Nullbyte ab, wie -print0

-fprintf Ausgabedatei Format

schreibt den Namen der getesteten Datei in die Ausgabedatei und benutzt dabei das Format mit Sonderzeichen wie bei printf

-ok Kommando;

wie -exec, vor der Ausführung des Kommandos wird aber noch eine Bestätigung erwartet; nur eine Eingabe, die mit einem B oder einem y beginnt, führt zur Ausführung des Kommandos

-print gibt den vollständigen Pfadnamen der getesteten Datei auf die Standardausgabe

-print0

gibt den Pfadnamen der getesteten Datei, von einem Nullbyte abgeschlossen, auf die Standardausgabe; auf diese Weise können auch Pfadnamen korrekt weiterverarbeitet werden, die ein Zeilenende enthalten

-printf Format

gibt für die getestete Datei die Zeichenkette Format auf der Standardausgabe aus; Format kann verschiedene Sonderzeichen und Platzhalter enthalten, die von find bearbeitet werden:

\a	Alarmton
\b	Rückschritt
\c	Abbruch der Ausgabe
\f	Seitenvorschub
\n	Zeilenvorschub
\r	Wagenrücklauf
\t	horizontaler Tabulator
\v	vertikaler Tabulator

\\ der Backslash selbst

ein Backspace gefolgt von irgendeinem anderen Zeichen wird als normales Zeichen interpretiert und einfach ausgegeben

%% das Prozentzeichen selbst

%a die Zeit des letzten Zugriffs auf die Datei,
in dem Format der ctime Funktion

%A k die Zeit des letzten Zugriffs auf die Datei,
in dem von k bestimmte Format; k hat dabei
das gleiche Format wie der entsprechende
Parameter der strftime Funktion in C:

@ Sekunden seit dem 1.1.1970 0 Uhr GMT

H Stunde (00 bis 23)

I Stunde (01 bis 12)

k Stunde (0 bis 23)

l Stunde (1 bis 12)

M Minute (00 bis 59)

p	PM oder AM
r	Zeit, 12 Stunden (hh:mm:ss: AM/PM)
S	Sekunden (00 bis 61)
T	Zeit, 24 Stunden (hh:mm:ss)
X	Zeit (H:M:S)
Z	Zeitzone, oder nichts
a	abgekürzter Wochentag
A	ausgeschriebener Wochentag
b	abgekürzter Monatsname
B	ausgeschriebener Monatsname
c	Datum und Zeit
d	Tag im Monat
D	Datum (mm/dd/yy)

h	das gleiche wie b
j	der Tag im Jahr
m	die Zahl des Monats
U	die Nummer der Woche, Sonntag als erster Wochentag
w	die Zahl des Wochentags
W	die Nummer der Woche, Montag als erster Wochentag
x	Datum (mm/dd/yy)
y	die letzten beiden Stellen der Jahreszahl
Y	die Jahreszahl
%b	die Dateigröße in 512 Byte Blöcken (aufgerundet)
%c	das Datum der letzten Statusänderung im For- mat der C ctime Funktion

%Ck	das Datum der letzten Statusänderung im Format der BR strftime " Funktion; Parameter wie oben"
%d	die Höhe der Datei im Verzeichnisbaum; Null bedeutet, daß die Datei Kommandozeilenargument ist
%f	der Name der getesteten Datei, ohne Verzeichnisse
%g	der Gruppenname der getesteten Datei oder die Kennzahl, wenn die Gruppe nicht eingetragen ist
%G	die Gruppenkennzahl
%h	die Verzeichnisnamen des Pfadnamen der getesteten Datei
%H	das Kommandozeilenargument (Test), mit dem die Datei gefunden wurde
%i	die Nummer der Inode der getesteten Datei
%k	die aufgerundete Größe der getesteten Datei

in Kilobytes

- %l das Objekt, auf die ein symbolischer Link zeigt; leer, wenn die getestete Datei kein symbolischer Link ist
- %m die Zugriffsrechte als Oktalzahl
- %n die Anzahl der harten Links auf die getestete Datei
- %p der Pfadname der Datei
- %P der Pfadname und das Kommandozeilenargument (Test), mit dem die Datei gefunden wurde
- %s die Größe der getesteten Datei in Bytes
- %t die Zeit der letzten Änderung, im ctime Format
- %Tk die Zeit der letzten Änderung, im strftime Format (siehe oben)
- %u der Name des Eigentümers der getesteten Datei oder die Kennzahl, wenn der Benutzer

nicht eingetragen ist

%U die Benutzerkennzahl des Eigentümers der
getesteten Datei

-prune wahr, wenn die Option -depth gesetzt ist; sonst
falsch

-ls zeigt das Verzeichnis in dem die getestete Datei
gefunden wurde mit ls -dils an

Operatoren:

Die Optionen, Tests und Aktionen können mit Operatoren
verknüpft werden. Die Bearbeitung erfolgt prinzipiell von
links nach rechts.

(Ausdruck)

die Klammern fassen den Ausdruck zu einer Operation
zusammen

! Ausdruck

:RI "ist wahr, wenn der " Ausdruck " falsch ist"

-not Ausdruck

ist ebenfalls wahr, wenn der Ausdruck falsch ist

Ausdruck1 Ausdruck2

UND Verknüpfung; wenn Ausdruck1 wahr ist, wird Ausdruck2 bewertet (ausgeführt)

Ausdruck1 -a Ausdruck2

auch eine UND Verknüpfung

Ausdruck1 -and Ausdruck2

auch eine UND Verknüpfung

Ausdruck1 -o Ausdruck2

ODER Verknüpfung; Ausdruck2 wird bewertet (ausgeführt), wenn Ausdruck1 falsch ist

Ausdruck1 -or Ausdruck2

auch eine ODER Verknüpfung

Ausdruck1 , Ausdruck2

Liste; beide Ausdrücke werden immer bewertet (ausgeführt); der Wahrheitswert des gesamten Ausdrucks entspricht dem von Ausdruck2

Verwendung

Auflisten aller Dateien, die mit .html enden:

```
$ find / -name "*.html"
...
$
```

Auflisten aller Dateien, die in den letzten 2 Tagen modifiziert wurden:

```
$ touch d1
$ touch -t 03281313 d2
$ ls -l
insgesamt 0
-rw----- 1 as users 0 Apr 6 10:08 d1
-rw----- 1 as users 0 Mär 28 13:13 d2
$ find . -mtime -2
.
./d1
$ find . -mtime -20
.
./d1
./d2
$
```

Löschen aller Log-Dateien, die seit 2 Tagen nicht mehr modifiziert wurden:

```
$ ls -l *.log
-rw----- 1 as users 0 Mär 28 13:13 d3.log
-rw----- 1 as users 0 Apr 6 10:11 d4.log
$
$ rm -i `find . -name "*.log" -mtime +2 `
rm: »./d3.log« entfernen? y
$
```

Ändern und Anzeigen der Zugriffsrechte aller Log-Dateien:

```
$ ls -l
insgesamt 0
-rw----- 1 as users 0 Apr 6 10:08 d1
-rw----- 1 as users 0 Mär 28 13:13 d2
-rw----- 1 as users 0 Mär 28 13:13 d3.log
-rw----- 1 as users 0 Apr 6 10:11 d4.log
$
$ find . -name "*.log" -print -exec chmod 660 {} \;
./d3.log
./d4.log
$ ls -l *.log
-rw-rw---- 1 as users 0 Mär 28 13:13 d3.log
-rw-rw---- 1 as users 0 Apr 6 10:11 d4.log
$
```

1.7.4. ls

BEZEICHNUNG

ls - (list) zeigt den Inhalt eines Verzeichnisses

SYNTAX

```
ls [-abcdgiklmnpqrstuxABCFLNQRSUX1] [-w Spalten] [-T Spal-
ten] [-I Muster] [--all] [--escape] [--directory]
[--inode] [--kilobytes] [--numeric-uid-gid] [--hide-con-
trol-chars] [--reverse] [--size] [--width= Spalten]
[--tabsize= Spalten] [--almost-all] [--ignore-backups]
[--classify] [--file-type] [--ignore= Muster] [--derefer-
ence] [--literal] [--quote-name] [--recursive]
[--sort={none,time,size,extension}] [--format={long,ver-
bose,commas,across,vertical,single-column}]
[--time={atime,access,use,ctime,status}] [Pfad...]
```

BESCHREIBUNG

ls gibt den Inhalt der Verzeichnisse des Dateisystems an.

Das Standardausgabeformat von ls hängt vom Typ des Ausgabegerätes ab. Auf einem Terminal ist die mehrspaltige Ausgabe das Standardformat. In allen anderen Fällen wird die Ausgabe einspaltig ausgeführt.

Das Verhalten des ls Kommandos läßt sich nicht mehr durch Umbenennen in ll dir vdir etc. verändern. Stattdessen sind die Kommandos dir und vdir als separate Binärdateien mit entsprechenden Standardformaten verfügbar.

1.7.5. pwd

BEZEICHNUNG

pwd gibt den Namen des aktuellen Verzeichnisses aus

SYNTAX

pwd

BESCHREIBUNG

pwd ist in den meisten Shells ein integriertes Kommando (z.B. in der bash(1))

Hinweis: Die meisten Unix Shells haben eine eingebaute pwd-Funktion, welche die gleichen Funktionen hat. Bei interaktiven Kommandos wird in der Regel die pwd-Funktion der Shell benutzt.

2. Verzeichnisorientierte Kommandos

2.1.mkdir

BEZEICHNUNG

mkdir - erzeugt ein leeres Verzeichnis

SYNTAX

mkdir [-p] [-m Modus] [--path] [--mode= Modus] Verzeichnis

...

OPTIONEN (siehe man pages)

2.2.rmdir

BEZEICHNUNG

rmdir - löscht Verzeichnisse

SYNTAX

rmdir [-p] [--path] Verzeichnis ...

BESCHREIBUNG

rmdir löscht leere Verzeichnisse. Es gibt keine Möglichkeit, Verzeichnisse zu löschen, die noch normale Dateien enthalten.

OPTIONEN (siehe man pages)

Verwendung

Versuch ein nicht leeres Verzeichnis zu löschen:

```
$ rmdir v  
rmdir: »v«: Das Verzeichnis ist nicht leer  
$
```

3. Filesystemorientierte Kommandos

3.1.mkfs

```
NAME
    mkfs - build a Linux file system
SYNOPSIS
    mkfs [ -V ] [ -t fstype ] [ fs-options ] filesystem [ blocks ]
DESCRIPTION
    mkfs is used to build a Linux file system on a device,
    usually a hard disk partition. filesystem is either the
    device name (e.g. /dev/hda1, /dev/sdb2) or the mount
    point (e.g. /, /usr, /home) for the file system. blocks
    is the number of blocks to be used for the file system.

    The exit code returned by mkfs is 0 on success and 1 on
    failure.

    In actuality, mkfs is simply a front-end for the various
    file system builders (mkfs.fstype) available under Linux.
    The file system-specific builder is searched for in a num-
    ber of directories like perhaps /sbin, /sbin/fs,
    /sbin/fs.d, /etc/fs, /etc (the precise list is defined at
    compile time but at least contains /sbin and /sbin/fs),
    and finally in the directories listed in the PATH enviro-
    nment variable. Please see the file system-specific
    builder manual pages for further details.
```

Achtung:

Durch mkfs wird die alte Information auf dem Datenträger zerstört (vgl. DOS format)

Verwendung

Anlegen eines Unix Filesystems auf Floppy

```
$ mkfs /dev/fd0
mke2fs 1.24a (02-Sep-2001)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 inodes, 1440 blocks
72 blocks (5.00%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
184 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
$
```

3.2.mount

NAME

mount - mount a file system

SYNOPSIS

mount [-lhV]

mount -a [-fFnrsvw] [-t vfstype]

mount [-fnrsvw] [-o options [,...]] device | dir

mount [-fnrsvw] [-t vfstype] [-o options] device dir

DESCRIPTION

All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the file system found on some device to the big file tree. Conversely, the umount(8) command will detach it again.

The standard form of the mount command, is

```
mount -t type device dir
```

This tells the kernel to attach the file system found on device (which is of type type) at the directory dir. The previous contents (if any) and owner and mode of dir become invisible, and as long as this file system remains mounted, the pathname dir refers to the root of the file system on device.

```
Three forms of invocation do not actually mount anything:  
    mount -h  
prints a help message;  
    mount -V  
prints a version string; and just  
    mount [-l] [-t type]  
lists all mounted file systems (of type type). The option  
-l adds the (ext2, ext3 and XFS) labels in this listing.  
See below.
```

...

Verwendung

Anlegen eines Mount-Punktes und Mounten der Diskette im Floppylaufwerk A:

```

$ mkdir /fdA
$
$ mount /dev/fd0 /fdA
$
$ cd /fdA
$ touch d1
$ ls -l
insgesamt 17
drwxr-xr-x    3 root    root      1024 Apr  6 10:35 .
drwxr-xr-x   25 root    root      4096 Apr  6 10:34 ..
-rw-----    1 root    root         0 Apr  6 10:35 d1
drwxr-xr-x    2 root    root     12288 Apr  6 10:33 lost+found
$ pwd
/fdA
$

```

3.3.umount

```

NAME
    umount - unmount file systems
SYNOPSIS
    umount [-hV]

    umount -a [-nrv] [-t vfstype]
    umount [-nrv] device | dir [...]
DESCRIPTION
    The umount command detaches the file system(s) mentioned

```

from the file hierarchy. A file system is specified either by giving the directory where it has been mounted, or by giving the special device on which it lives.

Note that a file system cannot be unmounted when it is 'busy' - for example, when there are open files on it, or when some process has its working directory there, or when a swap file on it is in use. The offending process could even be umount itself - it opens libc, and libc in its turn may open for example locale files.

OPTIONS (siehe man pages)

Verwendung

Versuch, ein Dateisystem zu entfernen, das noch verwendet wird:

```
$ pwd
/fdA
$ umount /dev/fd0
umount: /fdA: Das Gerät wird momentan noch benutzt
$
```

Entfernen des Dateisystems:

```
$ cd /
$ ls -l /fdA
insgesamt 17
drwxr-xr-x    3 root    root      1024 Apr  6 10:35 .
drwxr-xr-x   25 root    root      4096 Apr  6 10:34 ..
-rw-----    1 root    root         0 Apr  6 10:35 d1
drwxr-xr-x    2 root    root     12288 Apr  6 10:33 lost+found
$
$ umount /dev/fd0
$
$ ls -l /fdA
insgesamt 8
drwx-----    2 root    root      4096 Apr  6 10:34 .
drwxr-xr-x   25 root    root      4096 Apr  6 10:34 ..
$
```

3.4. Modifikation von Dateiattributen

3.4.1. chmod

BEZEICHNUNG

chmod - (change mode) ändert die Zugriffsrechte auf Dateien und Verzeichnisse

SYNTAX

chmod [-Rcfv] Modus Datei ...

BESCHREIBUNG

chmod setzt oder ändert die Zugriffsrechte auf Dateien oder Verzeichnisse. Die Benutzung von chmod ist nur dem Eigentümer oder dem Superuser (root) erlaubt.

Die Zugriffsrechte werden als Modus bezeichnet. Der Modus kann entweder als (drei- oder vierstellige) Oktalzahl oder durch Buchstabenkennungen angegeben werden. Bei Angabe als Oktalzahl legen die letzten drei Ziffern jeweils die Rechte für den Besitzer, die Gruppe und die Anderen fest. Die einzelnen Bits der Oktalziffer stehen dabei für Lesen (4), Schreiben (2) und Ausführen (1).

Wenn vier Ziffern angegeben werden, so setzt die erste Ziffer spezielle Ausführungsmodi:

Wenn das erste Bit (4) dieser Zahl gesetzt ist, wird ein Programm mit der effektiven Benutzerkennung (BEUID für Effective User-ID) des Besitzers dieser Datei ausgeführt.

Wenn das zweite Bit (2) dieser Zahl gesetzt ist, wird ein Programm mit der Gruppenkennung dieser Datei anstelle der realen Gruppenkennung des aufrufenden Benutzers ausgeführt.

Das dritte Bit (1) schließlich hat unter Linux nur bei Verzeichnissen eine Bedeutung.

Die Buchstabenkennung setzt sich aus den folgenden Teilen zusammen:

```
[ugoa...] [[+ -=] [rwxstugo...]...] [, ...]
```

Dabei steht u (user) für Besitzer, g (group) für Gruppe, o (other) für Andere und a (all) für Alle. Die arithmetischen Symbole + -= geben an, ob eine Berechtigung hinzugefügt (+), gelöscht (-) oder gesetzt (=) werden soll. Die Berechtigungen sind r (read) für Lesen, w (write) für Schreiben, x (execute) für Ausführen. Die Option s (set user/group ID on execution) setzt die Identitätskennung bei der Programmausführung. Die Option t (text) schützt die Dateien eines beschreibbaren Verzeichnisses vor Löschung durch fremde Systembenutzer. Die nachgestellten u , g und o schützen die entsprechenden Rechte für Besitzer, Gruppe und Andere vor Veränderung

(zur Benutzung im Zusammenhang mit `-a`).

Die Rechte von symbolischen Links werden von `chmod` nicht geändert. Es gelten hier immer die Rechte der Datei, auf die der Link zeigt.

OPTIONEN (siehe `man pages`)

3.4.2. `chown`

NAME

`chown` - change file owner and group

SYNOPSIS

`chown` [OPTION]... OWNER[:[GROUP]] FILE...

`chown` [OPTION]... :GROUP FILE...

`chown` [OPTION]... --reference=RFILE FILE...

DESCRIPTION

This manual page documents the GNU version of `chown`. `chown` changes the user and/or group ownership of each given file, according to its first non-option argument, which is interpreted as follows. If only a user name (or numeric user ID) is given, that user is made the owner of each given file, and the files' group is not changed. If the user name is followed by a colon or dot and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well. If a colon or dot but no group name follows the user name, that user is made the owner of the files and the group of the

files is changed to that user's login group. If the colon or dot and group are given, but the user name is omitted, only the group of the files is changed; in this case, `chown` performs the same function as `chgrp`.

Change the owner and/or group of each FILE to OWNER and/or GROUP.

`-c, --changes`

like verbose but report only when a change is made

`--dereference`

affect the referent of each symbolic link, rather than the symbolic link itself

`-h, --no-dereference`

affect symbolic links instead of any referenced file (available only on systems that can change the ownership of a symlink)

`--from=CURRENT_OWNER:CURRENT_GROUP`

change the owner and/or group of each file only if its current owner and/or group match those specified here. Either may be omitted, in which case a match is not required for the omitted attribute.

```
-f, --silent, --quiet
    suppress most error messages

--reference=RFILE
    use RFILE's owner and group rather than the speci-
    fied OWNER:GROUP values

-R, --recursive
    operate on files and directories recursively

-v, --verbose
    output a diagnostic for every file processed

--help display this help and exit

--version
    output version information and exit

Owner is unchanged if missing.  Group is unchanged if
missing, but changed to login group if implied by a `:`.
OWNER and GROUP may be numeric as well as symbolic.
```

Verwendung

```
$ ls -l
insgesamt 0
-rw----- 1 as      users      0 Apr  6 10:58 d1
-rw----- 1 as      users      0 Apr  6 10:59 d2
$
$ su // root
Password:
#
# chown root d*
# exit
$ // as
$ ls -l
insgesamt 0
-rw----- 1 root    users      0 Apr  6 10:58 d1
-rw----- 1 root    users      0 Apr  6 10:59 d2
$
```

Nur der Eigentümer bzw. Root darf ändern:

```

$ ls -l
insgesamt 0
-rw----- 1 root users 0 Apr 6 10:58 d1
-rw----- 1 root users 0 Apr 6 10:59 d2
$
$ chown as d* // as
chown: Ändern des Eigentümers von »d1«: Die Operation ist nicht erlaubt
chown: Ändern des Eigentümers von »d2«: Die Operation ist nicht erlaubt
$

```

3.4.3. chgrp

BEZEICHNUNG

chgrp - (change group) ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnisses

SYNTAX

```
chgrp [-Rcfv] [--recursive] [--show-changes] [--silent]
      [--quiet] [--verbose] Gruppe Datei ...
```

BESCHREIBUNG

Der Befehl chgrp ändert die Gruppenzugehörigkeit einer Datei oder eines Verzeichnisses. Die Benutzung von chgrp ist nur dem Eigentümer und dem Superuser (root) erlaubt. Der Eigentümer kann eine Datei nur den Gruppen zuordnen, denen er selbst auch angehört.

Die Gruppe kann als ganze Zahl oder als Name angegeben werden. Die möglichen Gruppen und ihre Kennzahlen sind in

der Datei /etc/group festgelegt und können mit id
angezeigt werden.

der Datei /etc/group festgelegt und können mit id
angezeigt werden.

OPTIONEN (siehe man pages)

Verwendung

```
$ ls -l
insgesamt 0
-rw----- 1 as      users      0 Apr  6 11:06 d1
$
$ chgrp dba d1
$
$ ls -l
insgesamt 0
-rw----- 1 as      dba        0 Apr  6 11:06 d1
$
```

3.4.4. ln

BEZEICHNUNG

ln - (link) erzeugt einen Verzeichniseintrag einer existierenden Datei unter anderem Namen

SYNTAX

ln [Optionen] Quelle [Ziel]

ln [Optionen] Quelle ... Zielverzeichnis

BESCHREIBUNG

Jede Datei wird bei ihrer Erzeugung mit ihrem Namen in ein Verzeichnis eingetragen. Dieser Eintrag enthält außerdem einen Verweis auf eine Inode, in der die Zugriffsrechte auf die Datei, der Dateityp und gegebenenfalls die Nummern der belegten Datenblöcke eingetragen sind.

Mit dem `ln` Kommando wird ein neuer Eintrag in einem Verzeichnis angelegt, der auf die Inode einer existierenden Datei zeigt. Diese Art Link wird als Hardlink bezeichnet. Weil die Zugriffsrechte auf die Datei in der Inode bestimmt werden, sind die Zugriffsrechte auf alle Links einer Datei gleich.

Hardlinks können nur auf dem Datenträger angelegt werden, auf dem sich die Datei (und damit die Inode) selbst befindet. Um Links über die Dateisystemgrenzen hinweg anlegen zu können, bietet Linux die Möglichkeit symbolischer Links. In diesen Links ist der absolute Pfad gespeichert, auf dem die gelinkte Datei gefunden werden kann. Ein Zugriff auf diese Datei wird dann vom Betriebssystem automatisch auf die gelinkte Datei umgelenkt.

Ist das letzte Argument des Aufrufs ein existierendes Verzeichnis, so werden alle als Quelle aufgelisteten Dateien mit entsprechenden Namen im Zielverzeichnis verbunden. Wird nur eine einzige Quelle benannt, so wird ein Link unter diesem Namen im aktuellen Verzeichnis angelegt. Normalerweise löscht `ln` keine existierenden Dateien. Es werden standardmäßig „hardlinks“ angelegt. Links auf Verzeichnisse oder auf Dateien in anderen Dateisystemen können nur mit symbolischen Links realisiert werden.

Gelegentlich verändert sich das Verhalten eines Programms, wenn es durch einen Link unter einem anderen Namen aufgerufen wird. (Das funktioniert natürlich nur, wenn diese Änderung im Programm vorgesehen ist.)

OPTIONEN (siehe man pages)

Verwendung

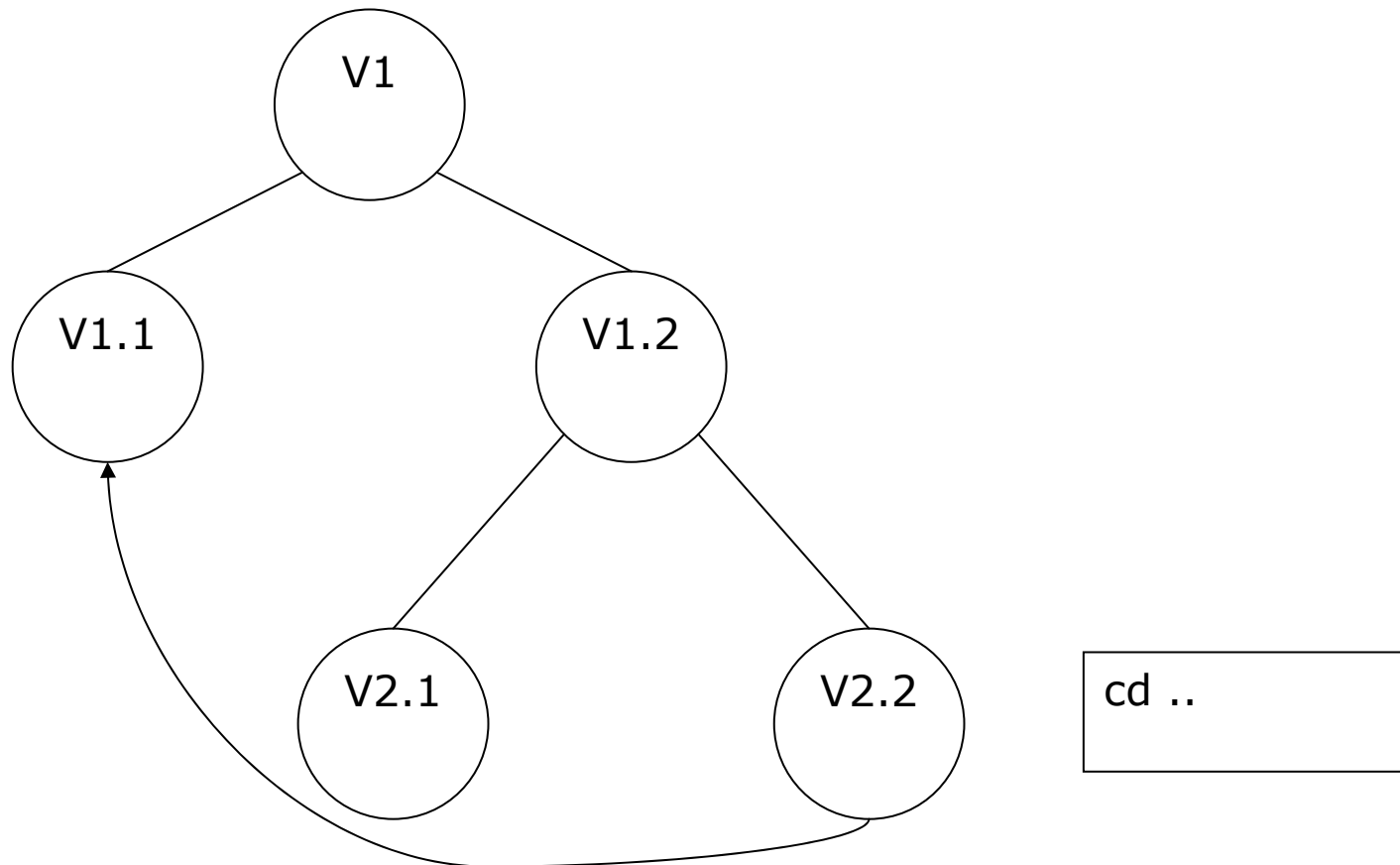
Hardlink auf /etc/passwd

```
$ pwd
/home/as/fhd/Vorlesungen/Unix/Kommandos/ln
$
$ ln /etc/passwd passwd
$
$ ls -l
insgesamt 8
-rw-r--r--    2 root    root      2905 Apr  2 19:11 passwd
drwx-----   4 as      users    4096 Apr  6 11:09 v1
$
```

Hardlinks auf Verzeichnisse nicht möglich:

```
$ pwd
/home/as/fhd/Vorlesungen/Unix/Kommandos/ln
$
$ ln /etc etc
ln: »/etc«: harte Verknüpfung für Verzeichnisse nicht erlaubt
$
```

Softlink auf Verzeichnis kann Probleme bereiten:



```
$ pwd
/home/as/fhd/Vorlesungen/Unix/Kommandos/ln/v1/v1.2
$
$ ls -l
insgesamt 4
drwx-----  2 as      users      4096 Apr  6 11:09 v1.2.1
lrwxrwxrwx   1 as      users           8 Apr  6 11:10 v1.2.2 -> ../../v1
$
$ cd ..
$
$ pwd
/home/as/fhd/Vorlesungen/Unix/Kommandos/ln/v1
$
```

3.4.5. mv

BEZEICHNUNG

mv - (move) verschiebt eine Datei oder benennt sie um

SYNTAX

mv [Optionen] Quelle Ziel

mv [Optionen] Quelle ... Verzeichnis

BESCHREIBUNG

mv verschiebt eine oder mehrere Datei(en) bzw. Verzeichnis(se) oder benennt sie um. Ein Verzeichnis kann nicht über die Grenzen eines Dateisystems hinweg verschoben werden.

OPTIONEN (siehe man pages)

Achtung:

Quell- und Zieldatei müssen im selben Dateisystem liegen, . Da nicht kopiert wird!.

Verwendung

Umbenennen von Dateien:

```
$ ls -l
insgesamt 4
-rw----- 1 as users 0 Apr 6 11:26 d_alt
drwx----- 2 as users 4096 Apr 6 11:26 v1
$ mv d_alt d_neu
$ ls -l
insgesamt 4
-rw----- 1 as users 0 Apr 6 11:26 d_neu
drwx----- 2 as users 4096 Apr 6 11:26 v1
```

Verschieben von Dateien:

```
$ ls -l
insgesamt 4
-rw----- 1 as users 0 Apr 6 11:26 d_neu
drwx----- 2 as users 4096 Apr 6 11:26 v1
$ mv d_neu v1
$ ls -l
insgesamt 4
drwx----- 2 as users 4096 Apr 6 11:26 v1
$ cd vi
$
$ ls -l
insgesamt 0
-rw----- 1 as users 0 Apr 6 11:26 d_neu
$
```

3.4.6. touch

BEZEICHNUNG

touch - ändert die Zeitmarkierung einer Datei

SYNTAX

```
touch [-acm] [-r Referenzdatei] [-t MMDDhhmm [[CC]YY] [.ss]] [-d Zeit] [--time= {time,access,use,mtime,modify} [--date= Zeit] [--file= Referenzdatei] [--no-create] Datei ...
```

BESCHREIBUNG

touch setzt die Zugriffs- und die Änderungszeit der Datei auf die aktuelle Zeit. Wenn die Datei nicht existiert, wird sie erzeugt.

Wenn als erster Dateiname ein gültiges Argument für die {it -t} Option übergeben wird, und keine der Optionen {it -d, -r} oder {it -t} ausdrücklich angegeben ist, wird dieses Argument als Zeitmarke für die folgenden Dateien verwendet.

OPTIONEN (siehe man pages)

Verwendung

Erzeugen einer Datei:

```
$ date
Sam Apr  6 11:31:17 CEST 2002
$ touch d1
$ ls -l
insgesamt 0
-rw-----  1 as      users      0 Apr  6 11:31 d1
$
```

Änderung des Zeitstempels einer Datei:

```
$ date
Sam Apr  6 11:34:28 CEST 2002
$ ls -l d*
-rw-----  1 as      users      0 Apr  6 11:33 d1
$
$ touch -t 01111818 d1
$ ls -l d1
-rw-----  1 as      users      0 Jan 11 18:18 d1
$
```

3.4.7. rm

BEZEICHNUNG

rm - löscht Dateien

SYNTAX

```
rm [-dfirvR] [--directory] [--force] [--interactive]
  [--recursive] [--verbose] Pfad ...
```

BESCHREIBUNG

rm löscht Dateien. Normalerweise werden die Verzeichnisse nicht mitgelöscht. Wenn eine Datei gelöscht werden soll, für die keine Schreibberechtigung besteht, muß der Befehl für diese Datei extra bestätigt werden. In Verzeichnissen, bei denen das Stickybit gesetzt ist, kann eine Datei nur von ihrem Eigentümer gelöscht werden.

Die Option -- zeigt an, daß die folgenden Argumente keine Optionen mehr sind. Dadurch ist es möglich auch Dateinamen, die mit einem - anfangen, zu löschen.

OPTIONEN (siehe man pages)

Verwendung

Löschen ohne Nachfrage, auch wenn Schreibschutz vorhanden:

```
$ ls -l d*
-r----- 1 as users 0 Apr 6 11:38 d1
$ rm d1
rm: schreibgeschützte Datei »d1« entfernen? n
$ ls -l
insgesamt 0
-r----- 1 as users 0 Apr 6 11:38 d1
$ rm -f d1
$ ls -l
insgesamt 0
$
```

3.5.Sichern und Zurückladen von Dateien

3.5.1. cp

BEZEICHNUNG

cp - (copy) kopiert eine oder mehrere Dateien

SYNTAX

cp [Optionen] Quelle Ziel

cp [Optionen] Quelle ... Verzeichnis

OPTIONEN (siehe man pages)

Achtung:

Da bei Kommandoaufrufen (wie `cp * /tmp`) das Metazeichen „*“ von der Shell expandiert wird, die Shell aber Punktdateien (z.B. `.profile`) nicht erfasst, werden diese versteckten Dateien nicht kopiert.

Sollen sie kopiert werden, muss man sie explizite angeben.

Verwendung

Kopieren aller Dateien, auch der versteckten Dateien:

```
$ cp * .* /tmp
```

cp mit einfachem Backup

```

$ ls -l
insgesamt 4
-rw----- 1 as users 0 Apr 6 16:59 d1
drwx----- 2 as users 4096 Apr 6 17:03 ziel
$ ls -l ziel
insgesamt 0
$
$ cp d1 ziel
$ ls -l ziel
insgesamt 0
-rw----- 1 as users 0 Apr 6 17:04 d1
$
$ cp --backup d1 ziel
$ ls -l ziel
insgesamt 0
-rw----- 1 as users 0 Apr 6 17:04 d1
-rw----- 1 as users 0 Apr 6 17:04 d1~ //backup
$

```

cp mit nummerierten Backups (VERSION_CONTROL=numbered)

```

$ echo $VERSION_CONTROL
numbered
$
$ cp --backup d1 ziel
$
$ ls -l ziel
insgesamt 0
-rw----- 1 as users 0 Apr 6 17:08 d1
-rw----- 1 as users 0 Apr 6 17:08 d1.~1~
$
$ cp --backup d1 ziel
$
$ ls -l ziel
insgesamt 0
-rw----- 1 as users 0 Apr 6 17:09 d1
-rw----- 1 as users 0 Apr 6 17:08 d1.~1~
-rw----- 1 as users 0 Apr 6 17:08 d1.~2~
$

```

3.5.2. cpio

```

NAME
    cpio - copy files to and from archives

SYNOPSIS
    cpio {-o|--create} [-0acvABLV] [-C bytes] [-H format] [-M
message] [-O [[user@]host:]archive] [-F
[[user@]host:]archive] [--file=[[user@]host:]archive]

```

```
[--format=format] [--message=message] [--null] [--reset-  
access-time] [--verbose] [--dot] [--append] [--block-  
size=blocks] [--dereference] [--io-size=bytes] [--quiet]  
[--force-local] [--help] [--version] < name-list [>  
archive]
```

```
cpio {-i|--extract} [-bcdfmnrtsuvBSV] [-C bytes] [-E file]  
[-H format] [-M message] [-R [user][:.][group]] [-I  
[[user@]host:]archive] [-F [[user@]host:]archive]  
[--file=[[user@]host:]archive] [--make-directories]  
[--nonmatching] [--preserve-modification-time] [--numeric-  
uid-gid] [--rename] [--list] [--swap-bytes] [--swap]  
[--dot] [--unconditional] [--verbose] [--block-  
size=blocks] [--swap-halfwords] [--io-size=bytes] [--pat-  
tern-file=file] [--format=format]  
[--owner=[user][:.][group]] [--no-preserve-owner] [--mes-  
sage=message] [--force-local] [--no-absolute-filenames]  
[--sparse] [--only-verify-crc] [--quiet] [--help] [--ver-  
sion] [pattern...] [< archive]
```

```
cpio {-p|--pass-through} [-0adlmuvLV] [-R  
[user][:.][group]] [--null] [--reset-access-time] [--make-  
directories] [--link] [--quiet] [--preserve-modification-  
time] [--unconditional] [--verbose] [--dot] [--derefer-  
ence] [--owner=[user][:.][group]] [--no-preserve-owner]  
[--sparse] [--help] [--version] destination-directory <
```

name-list

DESCRIPTION

This manual page documents the GNU version of `cpio`. `cpio` copies files into or out of a `cpio` or `tar` archive, which is a file that contains other files plus information about them, such as their file name, owner, timestamps, and access permissions. The archive can be another file on the disk, a magnetic tape, or a pipe. `cpio` has three operating modes.

In copy-out mode, `cpio` copies files into an archive. It reads a list of filenames, one per line, on the standard input, and writes the archive onto the standard output. A typical way to generate the list of filenames is with the `find` command; you should give `find` the `-depth` option to minimize problems with permissions on directories that are unwritable or not searchable.

In copy-in mode, `cpio` copies files out of an archive or lists the archive contents. It reads the archive from the standard input. Any non-option command line arguments are shell globbing patterns; only files in the archive whose names match one or more of those patterns are copied from the archive. Unlike in the shell, an initial ``.`` in a filename does match a wildcard at the start of a pattern, and a ``/'` in a filename can match wildcards. If no patterns are given, all files are extracted.

In copy-pass mode, cpio copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument. cpio supports the following archive formats: binary, old ASCII, new ASCII, crc, HPUX binary, HPUX old ASCII, old tar, and POSIX.1 tar. The binary format is obsolete because it encodes information about the files in a way that is not portable between different machine architectures. The old ASCII format is portable between different machine architectures, but should not be used on file systems with more than 65536 i-nodes. The new ASCII format is portable between different machine architectures and can be used on any size file system, but is not supported by all versions of cpio; currently, it is only supported by GNU and Unix System V R4. The crc format is like the new ASCII format, but also contains a checksum for each file which cpio calculates when creating an archive and verifies when the file is extracted from the archive. The HPUX formats are provided for compatibility with HPUX's cpio which stores device files differently.

The tar format is provided for compatibility with the tar program. It can not be used to archive files with names

longer than 100 characters, and can not be used to archive "special" (block or character devices) files. The POSIX.1 tar format can not be used to archive files with names longer than 255 characters (less unless they have a "/" in just the right place).

By default, cpio creates binary format archives, for compatibility with older cpio programs. When extracting from archives, cpio automatically recognizes which kind of archive it is reading and can read archives created on machines with a different byte-order.

Some of the options to cpio apply only to certain operating modes; see the SYNOPSIS section for a list of which options are allowed in which modes.

OPTIONS (siehe man pages)

Verwendung

Erstellen einer Sicherung (-c wg. Kompatibilität zu „älteren“ Unixen empfehlenswert):

```
$ find v1 | cpio -ovc > cpioArchive
v1
v1/d1
v1/v1.1
v1/v1.1/d2
v1/v1.2
v1/v1.2/d3
2 blocks
$
$ ls -l c*
-rw----- 1 as users 1024 Apr 6 13:04 cpioArchive
$
```

Lesen des Inhaltsverzeichnisses einer cpio Datei:

```
$ cpio -t < cpioArchive
v1
v1/d1
v1/v1.1
v1/v1.1/d2
v1/v1.2
v1/v1.2/d3
2 blocks
```

Wiederherstellen der Dateien:

```
$ ls -l
```

```
insgesamt 4
-rw----- 1 as      users      1024 Apr  6 13:04 cpioArchive
$
$ cpio -iv < cpioArchive
v1
v1/d1
v1/v1.1
v1/v1.1/d2
v1/v1.2
v1/v1.2/d3
2 blocks
$
$ ls -l
insgesamt 8
-rw----- 1 as      users      1024 Apr  6 13:04 cpioArchive
drwx----- 4 as      users      4096 Apr  6 13:08 v1
$
```

3.5.3. tar

BEZEICHNUNG

tar - (tape archiver) verwaltet Dateiarhive

SYNTAX

```
tar [-Acdrutux] [+delete] [-b N] [-BgGhiklmMoOpPPsSvwWz]
[-C Verzeichnis] [-f Datei] [-F Datei] [-K Datei] [-L
Länge] [-N Datum] [-T Datei] [-V Name] [-X Datei] [0-7]
[{\lmh}]
```

BESCHREIBUNG

tar ist ursprünglich ein Tool zur Verwaltung von Bandarchiven. Das GNU tar kann aber auch auf rohen Disketten oder in normalen Dateien Archive im tar Format anlegen und verwalten. Normalerweise werden Archive mit tar nicht komprimiert. Das GNU tar kann aber die Ein- und Ausgabe durch einen Kompressor leiten. Die neuen Versionen (ab 1.11.2) unterstützen sowohl compress als auch gzip.

Linux kann alle SCSI Streamer und mit Kernelpatches auch QIC-02 Streamer betreiben. Wenn keine Datei und kein Gerät angegeben ist, versucht tar auf einen Streamer zuzugreifen. Je nach Konfiguration ist das meist /dev/tape oder /dev/rmt0.

OPTIONEN (siehe man pages)

Verwendung

Erstellen einer Sicherung:

```
$ ls -l
insgesamt 4
drwx-----  4 as      users      4096 Apr  6 13:15 v1
$
$ tar cvf tarArchive *
v1/
v1/d1
v1/v1.1/
v1/v1.1/d2
v1/v1.2/
v1/v1.2/d3
$ ls -l
insgesamt 16
-rw-----  1 as      users     10240 Apr  6 13:16 tarArchive
drwx-----  4 as      users      4096 Apr  6 13:15 v1
$
```

Lesen des Inhaltsverzeichnisses einer cpio Datei:

```
$ tar tf tarArchive
v1/
v1/d1
v1/v1.1/
v1/v1.1/d2
v1/v1.2/
v1/v1.2/d3
$
```

Wiederherstellen der Dateien:

```
$ tar xvf tarArchive
v1/
v1/d1
v1/v1.1/
v1/v1.1/d2
v1/v1.2/
v1/v1.2/d3
$
$ ls -l
insgesamt 16
-rw-----  1 as      users      10240 Apr  6 13:22 tarArchive
drwx-----  4 as      users       4096 Apr  6 13:24 v1
$
```

3.5.4. dd

BEZEICHNUNG

dd - (disk dump) konvertiert Dateien für verschiedene Speichermedien

SYNTAX

```
dd [if= Datei] [of= Datei] [ibs= Bytes] [obs= Bytes] [bs= Bytes] [cbs= Bytes] [skip= Blöcke] [seek= Blöcke] [count= Blöcke] [conv= {ascii, ebcdic, ibm, block, unblock, lcase, ucase, swab, noerror, notrunc, sync}]
```

BESCHREIBUNG

dd liest eine Datei und schreibt den Inhalt mit wählbarer Blockgröße und verschiedenen Konvertierungen. Mit Hilfe dieses Kommandos können reguläre Dateien ebenso wie ganze Disketten oder Festplattenpartitionen kopiert werden.

OPTIONEN (siehe man pages)

BEISPIEL

Das Kommando

```
dd bs=8192 if=Image of=/dev/fd0
```

wird verwendet, um die fertig übersetzte Kerneldatei (Image) auf eine formatierte Diskette zu schreiben und so eine Bootdiskette zu erzeugen.

Mit dem Kommando

```
dd if=/dev/hda of=/dev/fd0 bs=512 count=1
```

kann der Superuser (root) eine Kopie des Festplattenbootsektors auf einer Diskette anlegen. Mit dieser Diskette kann die Festplatte gebootet werden, wenn der Festplattenbootsektor zerstört wurde.

Verwendung

Kopieren mit Konvertierung in Großbuchstaben:

```
$ cat d1
Dies ist ein Text
mit zwei Zeilen.
$
$ dd if=d1 of=d2 conv=ucase
0+1 Records ein
0+1 Records aus
$
$ cat d2
DIES IST EIN TEXT
MIT ZWEI ZEILEN.
$
```

Kopieren mit Formatumsetzung (ASCII->EBCDIC):

```
$ dd if=d1 of=d3 conv=ebcdic
0+1 Records ein
0+1 Records aus
$
$
$ cat d3
Ä%.ç@%ç£@. %@ã. $£%"%£@©|. %@é.%" .K%$
$
```

Achtung:

Normalerweise wird nicht wie in den o.a. Beispielen von Datei gelesen, sondern von einem Device, z.B. von Platte. Dann kann die Dateistruktur zerstört werden, wenn nicht die ganze Platte gelesen wird.

3.5.5. Konsistenzprüfung von Filesystemen

3.5.6. fsck

NAME

fsck - check and repair a Linux file system

SYNOPSIS

```
fsck [ -sACVRTNP ] [ -t fstype ] filesys [ ... ] [--] [
fsck-options ]
```

DESCRIPTION

fsck is used to check and optionally repair a one or more Linux file systems. filesys can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an ext2 label or UUID specifier (e.g. UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). The fsck program will try to run filesystems on different physical disk drives in parallel to reduce total amount time to check all of the filesystems.

The exit code returned by fsck is the sum of the following conditions:

- 0 - No errors
- 1 - File system errors corrected
- 2 - System should be rebooted
- 4 - File system errors left uncorrected
- 8 - Operational error
- 16 - Usage or syntax error

128 - Shared library error

The exit code returned when all file systems are checked using the `-A` option is the bit-wise OR of the exit codes for each file system that is checked.

In actuality, `fsck` is simply a front-end for the various file system checkers (`fsck.fstype`) available under Linux. The file system-specific checker is searched for in `/sbin` first, then in `/etc/fs` and `/etc`, and finally in the directories listed in the `PATH` environment variable. Please see the file system-specific checker manual pages for further details.

OPTIONS (siehe man pages)

Verwendung

```
# mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /boot type ext2 (rw)
shmfs on /dev/shm type shm (rw)
#
# fsck /dev/hda1
fsck 1.24a (02-Sep-2001)
e2fsck 1.24a (02-Sep-2001)
/dev/hda1 is mounted.

WARNING!!! Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? no

check aborted.
#
```

3.5.7. sync

BEZEICHNUNG

sync schreibt die gepufferten Blöcke auf die Platte

SYNTAX

sync

BESCHREIBUNG

Linux unterhält zur Optimierung der Festplatten und Diskettenzugriffe ein sogenanntes Blockdepot, in dem wie in einem Cache einige Datenblöcke des Massenspeichers im Arbeitsspeicher bereitgehalten werden. Veränderungen an diesen Daten werden zuerst nur im Arbeitsspeicher vorgenommen. Mit dem sync Befehl können die veränderten Daten sofort auf den Massenspeicher gesichert werden. Normalerweise wird in regelmäßigen Abständen automatisch vom update-Prozeß ein sync Systemaufruf ausgeführt.

Gelegentlich, zum Beispiel vor dem Ausschalten des Rechners, ist es sinnvoll, ein manuelles sync vorzunehmen. Wird der Rechner z.B. infolge eines Defektes ohne Datensynchronisation abgeschaltet, kommt es meist zu Datenverlusten und Fehlern im Dateisystem.

Verwendung

```
# sync  
#
```

4. Sitzungsorientierte Kommandos

4.1. Anmelden

4.1.1. newgrp

BEZEICHNUNG

newgrp - ändert die Gruppenkennung des aktuellen Benutzers

SYNTAX

newgrp [Gruppe]

BESCHREIBUNG

Das newgrp Kommando ändert die aktive Gruppenkennung des Anwenders. Ohne Angabe einer Gruppe wird zu der in der /etc/passwd Datei festgelegten Standardgruppe des Benutzers gewechselt. Sonst wird in die angegebene Gruppe gewechselt, wenn der Benutzer in der Datei /etc/group als Mitglied dieser Gruppe eingetragen ist. Das newgrp Kommando unterstützt keine paßwortgeschützten Gruppen.

Die Gruppe kann nur mit ihrem Namen angegeben werden.

Verwendung

Wechsel zu anderer Gruppe:

```
$ id
uid=500(as) gid=100(users) Gruppen=100(users),55(dba),101(ibs)
$
$ newgrp dba
Password:
$
$ id
uid=500(as) gid=55(dba) Gruppen=100(users),55(dba),101(ibs)
$
```

4.1.2. passwd

BEZEICHNUNG

passwd - ändert das Paßwort zum System

SYNTAX

passwd [Username]

BESCHREIBUNG

Die Paßwörter aller Benutzer werden in der Datei /etc/passwd gespeichert. Diese Datei ist lesbar aber schreibgeschützt. Um dem Benutzer die Möglichkeit zu geben, sein eigenes Paßwort zu ändern, läuft passwd SUID root. Deshalb hat der Anwender zur Laufzeit des Programms Rootprivilegien und darf in die Datei schreiben.

Bei einigen Linuxinstallationen wird das Benutzerpaßwort in einer separaten Datei namens shadow gespeichert, um den normalen Benutzern den Lesezugriff auf diese Daten zu verwehren. Die Einzelheiten zu diesem Paßwortsystem sind in den englischen Manualpages beschrieben.

4.1.3. su

BEZEICHNUNG

su - (superuser) ändert User und Gruppen ID

SYNTAX

su [-flmp] [-c Befehl] [-s Shell] [--login] [--fast]
[--preserve-environment] [--command= Befehl] [--shell=

```
Shell] [-] [Name [Argument...]]
```

BESCHREIBUNG

su startet eine neue Shell unter einer neuen Benutzerkennung (UID) und Gruppenkennung (GID). Wie bei einem neuen Login wird das Paßwort des Benutzers abgefragt. Wenn kein Name angegeben wird, ist das Standardargument root, man ändert die ID also in die des ``Superusers'`. Allein der Superuser kann mit dem su Kommando die Identität jedes beliebigen Users annehmen, ohne nach einem Paßwort gefragt zu werden. Damit (und nur damit) ist es möglich auch unter den Verwaltungsnamen wie {it bin, news} oder daemon zu arbeiten, die normalerweise durch ein Sperrpaßwort gesichert werden.

Das su Kommando setzt die Umgebungsvariablen HOME, SHELL, USER und LOGNAME entsprechend der Eintragungen in der Datei /etc/passwd.

Wenn außer dem Namen weitere Argumente in der Kommandozeile angegeben sind, werden sie der Shell übergeben.

OPTIONEN (siehe man pages)

Verwendung

User ändern, ohne Login Shell auszuführen:

```
$ su as  
Password:  
$
```

User ändern, Login Shell ausführen:

```
$ su - as  
Password:  
$
```

Befehl as andere User ausführen:

```
as@server> su -c "vi /etc/passwd" root
Password:
...
```

4.2. Informationen

4.2.1. cal

NAME

cal - displays a calendar

SYNOPSIS

```
cal [-m]y13] [[month] year]
```

DESCRIPTION

Cal displays a simple calendar. If arguments are not specified, the current month is displayed. The options are as follows:

- 1 Display single month output (use if cal was built with -3 as default to get older traditional output)
- 3 Display prev/current/next month output (use if cal was built with traditional -1 as default to get newer improved output)
- m Display Monday as the first day of the week. (The default is Sunday.)
- j Display Julian dates (days one-based, numbered from January 1).

`-y` Display a calendar for the current year.

A single parameter specifies the year (1 - 9999) to be displayed; note the year must be fully specified: ``cal 89'` will not display a calendar for 1989. Two parameters denote the month (1 - 12) and year. If no parameters are specified, the current month's calendar is displayed.

A year starts on Jan 1.

The Gregorian Reformation is assumed to have occurred in 1752 on the 3rd of September. By this time, most countries had recognized the reformation (although a few did not recognize it until the early 1900's.) Ten days following that date were eliminated by the reformation, so the calendar for that month is a bit unusual.

Verwendung

Aktuellen Monatskalender ausgeben:

```
$ cal
   April 2002
So Mo Di Mi Do Fr Sa
    1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

$
```

Bestimmten Monat ausgeben:

```
$ cal 01 1957
   Januar 1957
So Mo Di Mi Do Fr Sa
    1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

$
```

Jahr ausgeben:

\$ cal 2003

2003

Januar							Februar						März							
So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa
			1	2	3	4							1							1
5	6	7	8	9	10	11	2	3	4	5	6	7	8	2	3	4	5	6	7	8
12	13	14	15	16	17	18	9	10	11	12	13	14	15	9	10	11	12	13	14	15
19	20	21	22	23	24	25	16	17	18	19	20	21	22	16	17	18	19	20	21	22
26	27	28	29	30	31		23	24	25	26	27	28		23	24	25	26	27	28	29
														30	31					
April							Mai						Juni							
So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa
		1	2	3	4	5					1	2	3	1	2	3	4	5	6	7
6	7	8	9	10	11	12	4	5	6	7	8	9	10	8	9	10	11	12	13	14
13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	21
20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28
27	28	29	30				25	26	27	28	29	30	31	29	30					
Juli							August						September							
So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa
		1	2	3	4	5						1	2	1	2	3	4	5	6	
6	7	8	9	10	11	12	3	4	5	6	7	8	9	7	8	9	10	11	12	13
13	14	15	16	17	18	19	10	11	12	13	14	15	16	14	15	16	17	18	19	20
20	21	22	23	24	25	26	17	18	19	20	21	22	23	21	22	23	24	25	26	27
27	28	29	30	31			24	25	26	27	28	29	30	28	29	30				
							31													
Oktober							November						Dezember							
So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa
			1	2	3	4							1	1	2	3	4	5	6	
5	6	7	8	9	10	11	2	3	4	5	6	7	8	7	8	9	10	11	12	13
12	13	14	15	16	17	18	9	10	11	12	13	14	15	14	15	16	17	18	19	20
19	20	21	22	23	24	25	16	17	18	19	20	21	22	21	22	23	24	25	26	27
26	27	28	29	30	31		23	24	25	26	27	28	29	28	29	30	31			
							30													

4.2.2. date

BEZEICHNUNG

date - schreibt oder setzt die Systemzeit

SYNTAX

```
date [-u] [-s Datum] [+ FORMAT] [MMDDhhmm [[CC] YY] [.ss]]
```

BESCHREIBUNG

Ohne ein Argument gibt date die aktuelle Systemzeit aus. Wenn date mit einem Argument aufgerufen wird, dessen erstes Zeichen ein + ist, wird das Datum diesem Argument entsprechend formatiert. Dabei können alle Parameter benutzt werden, die von der strftime C-Bibliotheksfunktion verstanden werden. Alle Zeichen, die keine Parameter sind, werden unverändert ausgegeben.

Die folgenden Parameter werden unterstützt:

%% ein einfaches %

%n das Zeilenende

%t ein Tabulator

Die Zeitfelder:

%H Stunden 00 bis 23

%I	Stunden 00 bis 12
%M	Minuten 00 bis 59
%p	AM oder PM
%r	die Zeit, 12 Stunden (hh:mm:ss AM/PM)
%S	Sekunden 00 bis 59
%T	die Zeit, 24 Stunden (hh:mm:ss)
%X	die Zeit, 24 Stunden (%H:%M:%S)
%Z	die Zeitzone; oder nichts, wenn keine Zeitzone feststellbar ist

Die Datenfelder:

%a	der abgekürzte Wochentag
%A	der ausgeschriebene Wochentag
b	der abgekürzte Monat
%B	der ausgeschriebene Monat
%c	das Datum und die Zeit (Standardausgabeformat)

%d der Tag im Monat 00 bis 31
%D das Datum (mm/dd/yy)
%h das gleiche wie %b
%j der Tag im Jahr
%m der Monat 00 bis 12
%U die Nummer der Woche, Sonntag erster Tag
%w der Tag in der Woche 0 bis 6
%W die Nummer der Woche, Montag erster Tag
%x das Datum (mm/dd/yy)
%y die letzten beiden Stellen der Jahreszahl
%Y die ganze Jahreszahl

Wenn dem date Kommando ein Argument übergeben wird, das nicht mit einem + Zeichen beginnt, wird die Systemzeit diesem Argument entsprechend gesetzt. Das Argument muß

vollständig aus Ziffern bestehen. Die einzelnen Stellen der Zahl haben folgende Bedeutung:

MM	Monat
DD	Tag im Monat
hh	Stunde
mm	Minute
CC	die ersten beiden Stellen der Jahreszahl (optional)
YY	die letzten beiden Stellen der Jahreszahl (optional)
ss	die Sekunden (optional)

Nur der Superuser (root) darf die Systemzeit verändern.

OPTIONEN (siehe man pages)

Verwendung

Datum anzeigen:

```
$ date
Son Apr  7 11:40:55 CEST 2002
$
$ date +%d.%m.%Y
07.04.2002
$
```

Datum setzen (nur root):

```
$ su
Password:
# date 040712122002
Son Apr  7 12:12:00 CEST 2002
#
```

4.2.3. id

BEZEICHNUNG

id - gibt die reale und die effektive User ID und Gruppen ID aus

SYNTAX

```
id [-gnruG] [--group] [--name] [--real] [--user]
  [--groups] [Username]
```

BESCHREIBUNG

Das id Kommando zeigt die reale Benutzerkennzahl (BUID) mit dem Benutzernamen und alle Gruppen, in denen der Anwender eingetragen ist, mit ihren Kennzahlen (GID)s und Namen an. Wenn die effektive Benutzerkennung nicht der realen entspricht, wird die effektive Benutzerkennung ebenfalls angezeigt.

OPTIONEN (siehe man pages)

Verwendung

```
$ id
uid=500(as) gid=100(users) Gruppen=100(users),55(dba)
$
```

4.2.4. who

BEZEICHNUNG

who - gibt Information über die aktiven Anwender des Systems

SYNTAX

who [-imqsuwHT] [Datei] [ami]

BESCHREIBUNG

who zeigt die Namen, das Terminal und die Loginzeit der aktiven Benutzer. Außerdem gibt es Antwort auf die existentielle Frage wer bin ich?!

Wird eine Datei angegeben, so wird die Information über die aktiven Anwender aus dieser Datei genommen. Voreingestellt ist die Datei /etc/utmp

OPTIONEN (siehe man pages)

Verwendung

```
$ who
as          pts/0      Apr  6 09:33 (10.0.0.211)
$ whoami
as
$
```

5. Programmorientierte Kommandos

5.1.kill

BEZEICHNUNG

kill - beendet einen Prozess

SYNTAX

kill [-Signr] Prozeßnr

BESCHREIBUNG

kill wird benutzt, um außer Kontrolle geratene („aufgehängte“) Prozesse, die sich nicht mehr auf normale Art beenden lassen, zu terminieren (beenden). kill sendet dazu das Signal Signr an den Prozeß Prozeßnr. Standardwert ist SIGTERM (15) zum terminieren des Prozesses. Es können aber auch beliebige andere Signale gesendet werden. Weil das Signal SIGTERM nicht von allen Programmen bearbeitet wird, wird ein Prozeß manchmal erst mit dem Signal SIGKILL(9) vom Kernel beendet. Der „normalen“ Terminierung mit SIGTERM ist aber der Vorzug zu geben, weil dadurch dem Prozeß noch die Möglichkeit gegeben wird, die Bühne geordnet zu verlassen. Es können nur die eigenen Prozesse beendet werden.

In der bash ist ein kill Kommando eingebaut, daß dieses externe Programm verdeckt, wenn nicht ausdrücklich mit dem command Shellkommando das externe Programm aufgerufen wird.

OPTIONEN (siehe man pages)

Verwendung

```
$ sleep 200 &
[1] 14735
$ ps
  PID TTY          TIME CMD
11071 pts/0    00:00:00 bash
. . .
14735 pts/0    00:00:00 sleep
14736 pts/0    00:00:00 ps
$ kill -2 14735
[1]+  Interrupt                  sleep 200
$ ps
  PID TTY          TIME CMD
11071 pts/0    00:00:00 bash
. . .
14737 pts/0    00:00:00 ps
$
```

5.2.nice

BEZEICHNUNG

nice - läßt ein Programm mit veränderter Priorität laufen

SYNTAX

```
nice [-n Nettigkeit] [- Nettigkeit] [--adjustment= Nettigkeit] [Kommando [Argument...]]
```

BESCHREIBUNG

In einem Multitasking Betriebssystem wie Linux muß die Prozessorzeit auf verschiedene Prozesse verteilt werden. Dazu gibt es einen Scheduler der dafür sorgt, daß die Prozessorzeit möglichst optimal zugeteilt wird. Prozesse, die auf das Ergebnis eines anderen Prozesses, ein Ereignis oder ein Signal warten, können beispielsweise solange 'schlafen', bis das erwartete Ereignis eingetreten ist. Der Scheduler weckt die schlafenden Prozesse dann auf und teilt ihnen wieder Prozessorzeit zu. Trotzdem gibt es natürlich in der Regel mehr als einen lauffähigen Prozess. Und der Scheduler muß nach bestimmten Regeln den lauffähigen Prozessen Rechenzeit zuteilen. Dabei benutzt er unter anderem den nice Wert. Will ein Anwender nice, das heißt nett, zu den anderen Benutzern des Systems sein, startet er die Prozesse, die ruhig etwas länger dauern dürfen mit, dem nice Kommando. Ein negatives nice ist nur dem Superuser root erlaubt.

Wenn kein Wert angegeben ist, wird die Nettigkeit um 10

Punkte erhöht. Die maximale Nettigkeit ist 19 Punkte.
Nach unten kann der Superuser seine Nettigkeit bis -20
abkühlen.

OPTIONEN (siehe man pages)

Verwendung

Ausführen mit niedriger Priorität (=hohe nice Zahl)

```
$ nice man cpio > /dev/lp &  
$
```

Root darf hohe Priorität verwenden (nice Faktor erniedrigen):

```
# nice -20 tar cvf backup.tar /* &
#
```

5.3.nohup

BEZEICHNUNG

nohup - läßt ein Programm die Signale SIGHUP SIGINT SIGQUIT undSIGTERMignorieren

SYNTAX

nohup Kommando [Argument...]

BESCHREIBUNG

nohup schützt ein Programm vor den Hangup Signalen. Dadurch kann es im Hintergrund weiterlaufen, auch wenn der Benutzer sich ausloggt. Normalerweise würden mit der Loginshell alle Prozesse des Anwenders durch ein SIGHUP beendet.

Der Prozeß geht nicht automatisch in den Hintergrund, sondern muß mit einem & am Ende der Kommandozeile dorthin gebracht werden. Die Schedulerpriorität eines mit nohup gestarteten Programms wird um 5 erhöht. Wenn die Standardausgabe des Programms ein Terminal ist, so wird sie automatisch gemeinsam mit der Standardfehlerausgabe in die Datei nohup.out umgeleitet. Ist das aktuelle Verzeichnis schreibgeschützt, wird die Datei im HOME Verzeichnis angelegt.

Verwendung

```
# nohup nice -20 tar cvf backup.tar /* &  
# exit
```

6. Remote Kommandos und Filetransfer

6.1.telnet

NAME

telnet - user interface to the TELNET protocol

SYNOPSIS

```
telnet [-8EFKLacdfrx] [-S tos] [-X authtype] [-e escapechar] [-k realm]
        [-l user] [-n tracefile] [host [port]]
```

DESCRIPTION

The telnet command is used to communicate with another host using the TELNET protocol. If telnet is invoked without the host argument, it enters command mode, indicated by its prompt (telnet>). In this mode, it accepts and executes the commands listed below. If it is invoked with arguments, it performs an open command with those arguments.

OPTIONS (siehe man pages)

Verwendung

6.2.ftp

NAME

ftp - ARPANET file transfer program

SYNOPSIS

ftp [-v] [-d] [-i] [-n] [-g] [host]

DESCRIPTION

Ftp is the user interface to the ARPANET standard File Transfer Protocol. The program allows a user to transfer files to and from a remote network site.

Options may be specified at the command line, or to the command interpreter.

OPTIONS (siehe man pages)

Verwendung

```
$ ftp c3
Connected to c3.fbi.fh-darmstadt.de.
220 c3 FTP server (Version 1.7.212.1 Thu May 9 21:10:27 GMT 1996) ready.
Name (c3:swta10): swta10
331 Password required for swta10.
Password:
230 User as logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Nun kann man mit „get“ und „put“ Dateien transferieren.

```
ftp> cd /etc
250 CWD command successful.
ftp> get passwd localPWD
200 PORT command successful.
150 Opening BINARY mode data connection for passwd (627 bytes).
226 Transfer complete.
627 bytes received in 0.00 seconds (877.23 Kbytes/s)
ftp> by
221 Goodbye.
$
```

Damit man das Passwort nicht interaktiv angeben muss (in Skripten), kann man eine Passwortliste in der Datei „.netrc“ ablegen. Die Datei darf nur für den Besitzer lesbar und schreibbar sein.

```
$ cat .netrc
machine cs1 login as password XXX
machine cs2 login as password XXY
$
```

6.3.r-Kommandos

Die r-Kommandos dienen dazu, auf entfernte Rechner zuzugreifen, ohne sich explizite dort anmelden zu müssen.

Dazu ist es erforderlich, das man auf den Remote-Rechnern definiert, welcher andere Rechner als „befreundet“ gilt. Dies wird auf dem Remote-Rechner durch die Datei `.rhosts` im Home-Verzeichnis des Benutzers abgelegt.

Die Datei `.rhosts` enthält pro befreundetem Host eine Zeile mit

```
Hostname username # Comment
```

Beispiel:

```
$ hostname
trex
$ whoami
as
$ cat .rhosts
+
+ root
ulab ue48 # Kommentar
```

Damit hat

1. der User as von beliebigen Rechnern Zugriff
2. Root von beliebigen Rechner Zugriff
3. der User ue48 vom Rechner ulab Zugriff

6.3.1. rsh

NAME

rsh - remote shell

SYNOPSIS

```
rsh [-Kdnx] [-k realm] [-l username] host [command]
```

DESCRIPTION

Rsh executes command on host.

Rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. Interrupt, quit and terminate signals are propagated to the remote command; rsh normally terminates when the remote command does.

OPTIONS (siehe man pages)

Verwendung

Date Kommando auf Rechner trex als User ue42 ausführen:

```
as@linux>
as@linux> rsh -l ue42 trex date
Sun Apr 7 12:30:36 MET DST 2002
as@linux>
as@linux> rsh -l ue42 trex 'cat .rhosts'
+ +
as@linux>
```

Achtung: Metazeichen werden von lokaler Shell expandiert -> Quoten notwendig!

6.3.2. rlogin

```
NAME
    rlogin - remote login
SYNOPSIS
    rlogin [-8EKLDx] [-e char] [-k realm] [-l username] host
DESCRIPTION
    Rlogin starts a terminal session on a remote host host.

    Rlogin first attempts to use the Kerberos authorization mechanism,
    described below.  If the remote host does not supporting Kerberos the
    standard Berkeley rhosts authorization mechanism is used.
OPTIONS (sieh man pages)
```

Verwendung

Login auf entferntem Rechner ohne Passwort:

```
as@linux> rlogin -l ue42 trex
Last login: Sun Apr  7 12:28:49 from ulab.fbi.fh-darm
Sun Microsystems Inc.    SunOS 5.7          Generic October 1998
$ hostname
trex
$
```

6.4.rcp

NAME

rcp - remote file copy

SYNOPSIS

rcp [-px] [-k realm] file1 file2

rcp [-px] [-r] [-k realm] file ... directory

DESCRIPTION

Rcp copies files between machines. Each file or directory argument is either a remote file name of the form ``rname@rhost:path'`, or a local file name (containing no ``:'` characters, or a ``/'` before any ``:'`s).

- r If any of the source files are directories, rcp copies each subtree rooted at that name; in this case the destination must be a directory.
- p The -p option causes rcp to attempt to preserve (duplicate) in its copies the modification times and modes of the source files, ignoring the umask. By default, the mode and owner of file2 are preserved if it already existed; otherwise the mode of the source file

- modified by the `umask(2)` on the destination host is used.
- k The `-k` option requests `rcp` to obtain tickets for the remote host in realm `realm` instead of the remote host's realm as determined by `krb_realmofhost(3)`.
 - x The `-x` option turns on DES encryption for all data passed by `rcp`. This may impact response time and CPU utilization, but provides increased security.

If `path` is not a full path name, it is interpreted relative to the login directory of the specified user `ruser` on `rhost`, or your current user name if no other remote user name is specified. A path on a remote host may be quoted (using `\`, `"`, or `'`) so that the metacharacters are interpreted remotely.

`Rcp` does not prompt for passwords; it performs remote execution via `rsh(1)`, and requires the same authorization.

`Rcp` handles third party copies, where neither source nor target files are on the current machine.

Achtung: Metazeichen werden von lokaler Shell expandiert -> Quoten notwendig!

Verwendung

Kopieren der Datei `/etc/hosts` des Rechners `trex` als User `ue42` in das aktuelle Verzeichnis.

```
as@linux> ls -l
insgesamt 0
as@linux>
as@linux> rcp ue42@trex:/etc/hosts hosts
as@linux> ls -l
insgesamt 4
-r--r--r--    1 as          users          337 Apr  7 12:48 hosts
as@linux>
```
