

9. Kommandos für Files Teil 1
=====

Wie kann ich Files anschauen?

=====

Achtung! Es gibt Text- und Binärdateien.

Binärdateien sollte man nur sehr vorsichtig mit den dafür vorgesehenen Werkzeugen anschauen. Bildschirm kann eventuell verloren gehen.

Feststellen des Types eines Files:

```
file <filename>
```

Das Kommando "file" ist abhängig vom Betriebssystem!

POSIX:

```
file [-dh][-M file][-m file] file ...
```

Solaris:

```
file [-dh] [-m mfile] [-M Mfile] [-f ffile] file...
```

Linux:

```
file [-bchiklLNnprsvz0] [--apple] [--mime-encoding] [--mime-type]
[-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...
```

Sequentiell von vorn nach hinten (Text-Files)

Kommando:

more - file perusal filter for crt viewing

Syntax:

more [-dlfpsu] [-<num>] [+/<pattern>] [+<line>] [file ...]

Beschreibung:

More ist ein Filterprogramm, das einen Text Seitenweise auf einem Ausgabegerät anzeigt. Altes Werkzeug. Less ist besser, da leistungsfähiger. Nach jeder Seite wird eine Pause gemacht, die der Nutzer durch eine Eingabe beenden kann. Folgende Eingaben sind Möglich:

Leertaste - Seitenvorschub
Eingabetaste - Zeilenvorschub.
h - Ausgabe eines Hilfsmenü.
/<regulärer Ausdruck> - Vorwärtssuchen
! - Ausführung eines Kommandos

Weiteres: Siehe Hilfsmenü

Optionen:

```
-<num> - <num> Zahl der Zeilen auf dem Ausgabegerät
-d      - Ausschrift bei nichtdruckbaren Zeichen
        anstelle von Warnton
-l      - Pause bei jedem ^L (Seitenvorschub), sonst
        Warnsignal.
-f      - logisches Zählen, kein falten von langen Zeilen
-s      - unterdrücke mehrfache Leerzeilen
-u      - unterdrücke Unterstreichungen
+<pattern>
        - fange erst ab <Zeichenkette> an
+<line> - fange erst ab Zeil <num> an
```

Beispiele:

```
cd Texte
more unix
more -20 unix          # Zeilenzahl 20
more -s unix          # Leerzeilen zusammenfassen
more +/wenig unix     # ab "wenig"
more +30 unix         # ab 30.Zeile
more +5 /etc/passwd
more +/man /etc/passwd
```

Nocheinmal sequentiell von vorn nach hinten und zurück

Kommando:

less - opposite of more

Syntax:

```
less -?  
less --help  
less -V  
less --version  
less [-[+]aAbcCdeEfFgGiIjLmMnNqQrRsSuUVwWX~]  
      [-b space] [-h lines] [-j line] [-k keyfile]  
      [-K character set] [-{oO} logfile]  
      [-p pattern] [-P prompt] [-t tag]"  
      [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]  
      [-# shift] [+[[+]cmd] [--] [filename]...
```

Beschreibung:

Less ist ein Filterprogramm, das einen Texte Seitenweise auf einem Ausgabegerät anzeigt. Im Gegensatz zu more kann man mit diesem Programm vorwärts und rückwärts in dem angezeigten File blättern. Less versteht die Kommandos von more und vi.

Optionen:

```
less --help
```

SUMMARY OF LESS COMMANDS

Commands marked with * may be preceded by a number, N.

Notes in parentheses indicate the behavior if N is given.

h H Display this help.
q :q Q :Q ZZ Exit.

MOVING

e ^E j ^N CR * Forward one line (or N lines).
y ^Y k ^K ^P * Backward one line (or N lines).
f ^F ^V SPACE * Forward one window (or N lines).
b ^B ESC-v * Backward one window (or N lines).
z * Forward one window (and set window to N).
w * Backward one window (and set window to N).
ESC-SPACE * Forward one window, but don't stop at end-of-file.
d ^D * Forward one half-window (and set half-window to N).
u ^U * Backward one half-window (and set half-window to N).
ESC-) RightArrow * Left one half screen width (or N positions).
ESC-(LeftArrow * Right one half screen width (or N positions).
F Forward forever; like "tail -f".
r ^R ^L Repaint screen.
R Repaint screen, discarding buffered input.

Default "window" is the screen height.

Default "half-window" is half of the screen height.

.....

Beispiele:

```
cd Texte  
less unix
```

Nur Dateianfänge ansehen

Kommando:

head - output the first part of files

Syntax:

head [-NUMBER] [OPTION]... [FILE]...

Beschreibung:

head gibt die ersten 10 Zeilen der angegebenen Files auf der Standardausgabe. Vor jedem File wird der Filename als Überschrift ausgegeben, wenn mehr als ein File angegeben wurde.

Optionen:

-c, --bytes=GRÖSSE	erste GRÖSSE Bytes ausgeben (Linux)
-n, --lines=ANZAHL	erste ANZAHL Zeilen statt 10 ausgeben
-q, --quiet, --silent	nie Dateinamen vorab ausgeben (Linux)
-v, --verbose	immer Dateinamen vorab ausgeben (Linux)
--help	diese Hilfe anzeigen und beenden. (Linux)
--version	Versionsinformation anzeigen und (Linux) beenden.

Nur Dateienden ansehen

Kommando:

tail - output the last part of files

Syntax:

tail [-NUMBER] [OPTION]... [FILE]...

Beschreibung:

Ausgabe der letzten 10 Zeilen der angegebenen Files auf der Standardausgabe (Bildschirm). Wenn mehr als ein File angegeben wurde, wird der Filename mit ausgegeben.

Optionen:

--retry - Wiederholen der Open-Operation, wenn ein Fehler auftritt (zusammen mit -f).
-c N , --bytes=N (Linux)
- Ausgabe der letzten N Bytes eines Files.
-f, --follow, follow=<nr>
- Ausgabe der an ein File angefügten Daten, für die Betrachtung von Protokollfiles.
-F - identisch mit -f --retry

```
-n N, --lines=N, -N                                (Linux)
    - Ausgabe der letzten N Zeilen eines Files
--max-unchanged-stats=N
    - reopen eines Files nach N Zyklen, wenn sich
      der Status nicht geändert hat. Standard ist 5.
      Zusammen mit -f.
--pid=PID - tail beenden, wenn Prozess mit der Nummer PID
           beendet wird.
-q, --quiet, --silent
    - schweigsam

-s S, --sleep-interval=S
    - warten von S Sekunden zwischen zwei Versuchen.
      Zusammen mit -f.
-v, --verbose                                       (Linux)
    - immer Filenamen ausgeben
--help      - Hilfsinformationen anzeigen          (Linux)
--version   - Version ausgeben                     (Linux)
```

Beispiele:

```
cd Texte
tail unix
tail -10 unix
tail -n 10 unix      (Linux)
tail -c 20 unix      (Linux)
tail -v -c 20 unix   (Linux)
tail -f /var/log/warn (Linux)
tail -f /var/adm/messages (Solaris)
```

Wenn man es genau wissen will oder was die Welt im innersten zusammenhält.

Anschauen von Binär-Files

Kommando:

```
od - dump files in octal and other formats
```

Syntax:

```
od [OPTION]... [FILE]...
```

Beschreibung:

Ausgabe der binären Darstellung eines jeden Bytes eines Files. Dabei kann zwischen verschiedenen binären Formaten gewählt werden:

- oktal (Standard)
- hexadezimal
- dezimal
- Wortformate
- Kombinationen von obigem

Optionen:

<code>-A, --address-radix=RADIX</code>	entscheiden, wie Dateioffsets ausgegeben werden RADIX: x - hexadezimal d - dezimal o - oktal
<code>-j, --skip-bytes=BYTES</code>	BYTES Eingabebytes am Anfang jeder Datei übergehen
<code>-N, --read-bytes=BYTES</code>	Ausgabe auf BYTES Eingabebytes pro Datei begrenzen
<code>-s, --strings[=BYTES]</code>	Ketten mit wenigstens BYTES alphanumerischen Zeichen ausgeben
<code>-t, --format=TYP</code>	Ausgabeformat(e) wählen a ein benanntes Zeichen c ASCII-Zeichen oder Backslash-Escape d[ANZAHL] dezimal mit Vorzeichen, ANZAHL Bytes pro Zahl f[ANZAHL] Fließkomma, ANZAHL Bytes pro Zahl o[ANZAHL] oktal, ANZAHL Bytes pro Zahl u[ANZAHL] dezimal ohne Vorzeichen, ANZAHL Bytes pro Zahl x[ANZAHL] hexadezimal, ANZAHL Bytes pro Zahl

```
-v, --output-duplicates    nicht * benutzen, um Zeilenunterdrückung  
                           anzuzeigen  
-w, --width[=BYTES]       Anzahl BYTES pro Ausgabezeile ausgeben  
--traditional             Argumente in traditioneller Form akzeptieren  
--help                   diese Hilfe anzeigen und beenden.  
--version                 Versionsinformation anzeigen und beenden.
```

Traditionell spezifizierte Formatangaben:

```
-a    dasselbe wie -t a, benannte Zeichen wählen  
-b    dasselbe wie -t oC, Oktalbytes wählen  
-c    dasselbe wie -t c, ASCII-Zeichen oder Backslash-Escapes  
      wählen  
-d    dasselbe wie -t u2, dezimale Shorts ohne Vorzeichen wählen  
-f    dasselbe wie -t fF, Fließkommazahlen wählen  
-h    dasselbe wie -t x2, hexadezimale Shorts wählen  
-i    dasselbe wie -t d2, dezimale Shorts wählen  
-l    dasselbe wie -t d4, dezimale Longs wählen  
-o    dasselbe wie -t o2, oktale Shorts wählen  
-x    dasselbe wie -t x2, hexadezimale Shorts wählen
```

Standard ist `-A o -t d2 -w 16`.

Beispiele:

```
od -bc dat-file  
od -A d -bc dat-file  
od -A x -bc dat-file  
od -A o -bc dat-file  
od -A x -t cx1 dat-file
```

Erzeugen eines Files ohne Editor für geniale Programmierer.

Eingabe eines Files über Standardeingabe

Kommando:

cat - concatenate files and print on the standard output

Syntax:

cat [OPTION] [FILE]...

Beschreibung:

Verketteten von Files oder der Standardeingabe und Ausgabe auf der Standardausgabe. Wenn kein File angegeben wurde wird von der Standardeingabe gelesen.

Optionen:

-A, --show-all	äquivalent zu -vET	(Linux)
-b, --number-nonblank	nichtleere Ausgabezeilen nummerieren	
-e	äquivalent zu -vE	
-E, --show-ends	\$ am Ende jeder Zeile ausgeben	(Linux)
-n, --number	alle Ausgabezeilen nummerieren	
-s, --squeeze-blank	nie mehr als eine einzige Leerzeile	
-t	äquivalent zu -vT	
-T, --show-tabs	TAB-Zeichen als ^I ausgeben	(Linux)
-u	(wird ignoriert)	
-v, --show-nonprinting	^ und M- Notation benutzen, außer für LFD und TAB	
--help	Hilfe anzeigen und beenden.	(Linux)
--version	Versionsinformation anzeigen und beenden.	(Linux)

Beispiele:

```

cat dat-file          # normal
cat -T dat-file      # TAB's anzeigen
cat -A dat-file      # Anzeigen aller nichtdruckbaren Zeichen
                    # CNTL, LF, NL, TAB, Umlaute
cat > yyy            # Erzeugen eines neuen Files
                    # EOF <CNTRL>+D (^D)

```


Kopieren von Files

```
cp [-i|-p] <Quellfile> <Zielfile>
```

```
cp [-i|-p] <Quellfile1> {<Quellfile2>} <Zieldirektory>
```

```
cp [-i|-p] [-r|-R] <Quelldirektory> <Zieldirektory>
```

Kopieren eines oder mehrer Files in ein anderes File bzw. in eine andere Direktory. Direktories können rekursiv kopiert werden (Option -r).

- i - interaktiv (beim Überschreiben von Files)
- p - Kopieren der Zugriffsrechte und Zeiten
- r - rekursives Kopieren
- R - rekursives Kopieren ohne Pipes

Besipiele:

```
cp  unix  unix.old    - Kopieren eines Files mit umbenennen  
cp  unix  save/      - Kopieren eines Files in ein Direktory
```

Wie werde ich Files los?

Löschen von Files

Kommando:

`rm - remove files or directories`

Syntax:

`rm [OPTION]... FILE...`

Beschreibung:

Löschen des angegebenen Files. Standardmäßig werden keine Directories gelöscht. Das File und das Directory, in dem das File steht, müssen schreibbar sein.

Optionen:

<code>-f, --force</code>	- File auch löschen, wenn schreibgeschützt, aber prinzipiell für den Nutzer möglich.
<code>-d, --directory</code>	- Löschen eines Directories (nur für SU) (Linux)
<code>-i, --interactive</code>	- interaktives Löschen, jedes File wird nachgefragt.
<code>-r, -R, --recursive</code>	- rekursives Löschen von Directories.
<code>-v, --verbose</code>	- nicht schweigsam (Linux)
<code>--help</code>	- Hilfetext auf Standardausgabe anzeigen
<code>--version</code>	- Versionsnummer anzeigen (Linux)

Beispiele:

(makedirs)

```
rm file1
rm file1 file2 file3
rm dir1
rm dir1/file1
rm dir1/file2
rm dir1/file3
rm dir1
rm -r dir1
rm -ri dir2
rm -f dir3
rm -r dir3
rm -fr dir3
```

```
rm --help
rm --version          (Linux)
```

Für misstrauische Nutzer:

```
shred [-f] [-n] [-u] [-v] [-z] <Filename>
```

shred überschreibt das File erst und löscht es dann.

Löschen von Files im grossen Stil.

Löschen von Directories

Kommando:

```
rmdir - remove empty directories
```

Syntax:

```
rmdir [OPTION]... DIRECTORY...
```

Beschreibung:

Löschen eines leeren Directories.

Optionen:

```
--ignore-fail-on-non-empty
```

Jeden Fehlschlag ignorieren, der nur daher rührt, dass ein Verzeichnis nicht leer ist.

```
-p, --parents
```

VERZEICHNIS entfernen, dann versuchen, jede Verzeichnis-komponente im Pfad zu entfernen; so ist »rmdir -p a/b/c« ist das gleiche wie »rmdir a/b/c a/b a«

```
-v, --verbose (Linux)
```

Diagnose für jedes bearbeitete Verzeichnis ausgeben

```
--help (Linux)
```

diese Hilfe anzeigen und beenden.

```
--version (Linux)
```

Versionsinformation anzeigen und beenden.

Beispiele:

```
rmdir rdir1  
rmdir -v rdir2  
rmdir -v rdir3  
rmdir --version
```

Anzeigen von binären Files

Kommando

`strings` - print the strings of printable characters in files.

Syntax

```
strings [-afovV] [-min-len]
        [-n min-len] [--bytes=min-len]
        [-t radix] [--radix=radix]
        [-e encoding] [--encoding=encoding]
        [-] [--all] [--print-file-name]
        [-T bfdname] [--target=bfdname]
        [--help] [--version] file...
```

Beschreibung

Anzeigen von Zeichenketten in binären Files. Es werden nur die druckbaren Zeichenketten eines Files ausgegeben. Nichtdruckbare Zeichen werden unterdrückt. Jede Zeichenkette wird auf einer Zeile ausgegeben.

Beispiel

```
strings - binaeresfile      # ganze File scannen
strings  binaeresfile      # Nur den Anfang scannen
```