```
                         8.Elementare Kommandos für Direktories und Files
                         =====================
                         pwd, cd, ls, mkdir, ...
Was ist ein File?
-----------------
    File ist eine Folge von Bytes.
    Es gibt "besondere Files" - für das Betriebssystem (Directories,
    Geräte, ...) und gewöhnliche Files - für den Nutzer.
      - Gewöhnliche Files haben für das Betriebssystem keine Struktur.
              kein EOF, keine Zeilenstruktur, keine Blockstruktur.
      - Besondere Files haben für das Betriebssystem eine Struktur.
      - Files haben eine Länge.
      - Auf Files kann sequentiell oder direkt (Adresse des Bytes)
        zugegriffen werden.
      - Die Lage der Bytes eines Files auf den externen Medien ist für
        den Nutzer geheim.
      - Files werden durch einen Namen referenziert. Der Name muß in einem
        Directory verzeichnet sein!!!!!
        Es gibt keine namenlose Files (Systemfehler!!!!).
      - Files haben veränderbare Eigenschaften:
          Eigentumsrechte (Angabe des Besitzer):
               Eigentuemer
               Gruppe
          Zugriffsrechte:
              lesen, schreiben, ausführen
                 unterteilt nach
                       Eigentümer des Files
                       Gruppe des Files
                       den Rest der Welt (other)
           also 9 Zugriffsklassen
```





```
Die Files werden hierarchisch angordnet.
   Die Wurzel allen Übels ist das Wurzeldirectory -   "/".
   In einem Directory können Directories und Files
   enthalten sein - bunt gemischt.

   Dateibaum des UNIX:
                                                                  Wurzeldirectory
                                         /
                                         |                            Directories
  +----+----+----+-----+------+-----+------+--------+----+-----+----+----+
  |    |    |    |     |      |     |      |        |    |     |    |    |
  .    ..  etc home   bin    lib   opt    dev      proc sbin  var  vol  usr
                 |                                                  |
       +--+------+                                      +--+-------+-------+---+
       |  |      |                                      |  |       |           |
       .  ..   unixsoft                                 .  ..    fob-vol6
                 |                                                 |
       +--+----+-+--------+                         +--+-----+-----+------------+
       |  |    |          |                         |  |     |                  |
       .  ..  bell       Meyer                      .  ..   mi06              mi96
               |          |                                  |                  |
+--+--+--+--+--+--+--+    +--+                      +--+-----+-+------+--+  +--++
|  |  |  |  |  |  |  |    |  |                      |  |       |      |  |  |  |
.  .. d1 f1 f2 d2 f3 f4   .  ..                     .  ..    mueller        .  ..
      |        |                                               |
+-+---+-+-+    +--+                                    +--+---+--+--+--+--+
| |   |   |    |  |                                    |  |   |  |  |  |  |
. ..  f1  f9   .  ..                                   .  ..  f1 f2 d1 f3 d3
                                                                  |     |
```

```
Spezielle Directory-Namen:

   .   -   aktuelles Directory
   ..  -   übergeorgnetes Directory

Jeder Prozess (Kommando) "steht" in einem Directory - genannt
                               "Working Directory"
                               Aktuelles Directory
                               current directory
Nach dem Login (1.Prozess) steht man im Homedirectory.

Namensbildung:
    absolute Filenamen:
                Beginnen mit einem "/"   (Wurzelverzeichnis)
                enthalten die Namen aller Directories bis zum
                konkreten File
                 z.B.  /vol/fob-vol4/mi06/mueller/d1/f1
                       /vol/fob-vol4/mi96/../mi03/d1/f1
    relativer Filename:
                Geht vom "Working Directory" aus.
                   z.B.  "Working Directory":  /vol/fob-vol4/mi06
                                  d1/f1
                                  ./d1/f1
                                  f1
                                  ./f1
                                 ../mi96
Die Folge der Namen der Directories, die zu einem File führt wird als
  "Path" (Pfad) bezeichnet.
```





```
Elementare Kommandos für Direktories
====================================

Bestimmen des "Working Directory"

      Kommando:
                pwd   -  print name of current/working directory

      Synctax:
                pwd

      Beschreibung:
                Ausgabe des vollen Filenamens (beginnend mit "/") des
                aktuellen Directories.

      Beispiele:
                pwd
```

**Ändern des "Working Directory"**

**Kommando:**

```
cd,chdir  -  change working directory
```

**Syntax:**

```
cd     <Directory-Path>
chdir  <Directory-Path>
```

**Beschreibung:**

Ändern des Working Directories. Das Working Directory wird für den aktuellen Prozess (Shell) auf das angegebenen Directory gesetzt.

Voraussetzung:
- Das angegebenen Directory exisiert
- Der Prozess hat die notwendigen Zugriffsrechte (Ausführen)

Der Directory-Path kann absolute oder relativ angegeben werden.

**Beispiele:**

```
cd /tmp/D1
cd ..
cd /tmp/D2
```





**Ich will wissen, was für Files in meinem Directory sind.**

**Directory besichtigen. Eigenschaften von Files feststellen.**

**Kommando:**

```
ls   -  list directory contents
```

**Syntax:**

```
ls [OPTION]... [FILE]...
```

**Beschreibung:**

Ausgabe einer Liste mit Informationen über die Files des aktuellen Directories (wenn nicht anders angegeben). Die Liste ist alphabetische sortiert (wenn nicht anders gefordert). ist ein Filename spezifiziert worden wird die Ausgabe auf dieses File beschränkt. Ist der Filename ein Directory-Name, wird der Inhalt der Directory ausgegeben.

**Optionen:**

```
--help
-a        -   zeigt alle Filenamen an (einschließlich
              Punkt-Files)
-lisa     -   zeigt alles an
-lisac    -   zeigt alles an, letzte Modifikationszeit
              wird als Zeitangabe benutzt
-lisact   -   wie -lisac aber nach Modifikationszeit
              sortiert
```

**Beispiele:**

```
ls                   # Files
ls -a                # Files + Punktfiles
ls -l                # Files + Informationen
ls -la               # Files + Punktfiles + Informationen
ls -lac              #   ... + Modifikationszeit
ls -lact             #   ... + Sortierung nach Zeit
ls -lisa             # Alles mit Erzeugungszeit
ls -lisat            #   ... + Sortierung nach Zeit
ls -lisact           # Alles mit Modifikationszeit
ls -g                # wie -l ohne Nutzername
ls -gG               # wie -l ohne Nutzername und Gruppenname
ls -gGct             # ... + Modifikationszeit und Sortierung
ls -lisau            # Alles mit Accesstime
```





**Erzeugen von Directories**

**Kommando:**

mkdir – make directories

**Syntax:**

mkdir [OPTION] DIRECTORY...

**Beschreibung:**

Erzeugen des angegebenen Directories. Der Directory–Name kann als relativer oder absoluter Filename angegeben werden Das Directory darf noch nicht existieren. Der Nutzer muß Schreibrechte in dem Directory besitzen, in dem das Directory angelegt werden soll

**Optionen:**

| | |
|---|---|
| -p, --parents | kein Fehler, wenn vorhanden; übergeordnete Verzeichnissen erzeugen, wenn notwendig |
| --help | diese Hilfe anzeigen und beenden. |

**Beispiele:**

```
cd /tmp
mdkir x1
mkdir x2
mkdir x3/x4
mkdir -p x3/x4
mkdir --help
```

**Welchen Inhalt haben meine Files?**

**Kommando:**

```
file - determine file type
```

**Syntax:**

```
file  [  -bcikLnNsvz  ] [ -f namefile ] [ -F separator ]
                        [ -m magicfiles ] file ...
file -C [ -m magicfile ]      !!!! nur für Profis
```

**Beschreibung:**

Das Programm "file" testet jedes File und versucht das File zu klassifizieren. Folgende Tests werden durchgeführt:

- Filesystem-Test – Directory ermitteln
- Magic-Number-Test – Filetype bestimmen ASCII, ps, pdf, ..., ausführbares Programm (eventuell Architektur) Type-File: /usr/share/misc/magic
- Sprachtest – Programmiersprache (C, perl,...), natürliche Sprache (deutsch, englisch) bestimmen.

**Beispiele:**

```
file /etc/passwd
file /etc/shadow
file /etc/init.d/autofs
```





**Kommando:**

touch - Erzeugen eines leeren Files, modifizieren der Zugriffszeit

**Syntax:**

```
touch [-a] [-c] [-d date] [-m] [-r referenzfile]
      [-t [[CC]YY]MMDDhhmm[.ss]][--help][--version]
       <filename> {<filename>}
```

**Beschreibung:**

Erzeugen eines leeren Files, falls dieses nicht vorhanden ist.
Setzen der Zugriffszeit.

**Optionen:**

```
-a          nur die Zugriffszeit des Files ändern
-c          File nicht erzeugen
-d date     benutze "date" als Zeitangabe
-m          nur Modifikationszeit des Files ändern
-r referenzfile   benutze Zeitangaben des Referenzfiles
            als Zeit
--help      Hilfe
```

**Beispiele:**

```
touch xxx
touch yyy zzz tttt
touch -a *
touch -a -c xxx yyyy cccc
```

```
Elementare Kommandos für Files
==============================


Wie kann ich Files anschauen?

Achtung!!!  Es gibt ASCII- und Binary-Files.

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

Feststellen des Types eines Files:

      file <filename>       #    siehe oben
```





```
Sequentiell von vorn nach hinten (ASCII-Files).

    Kommando:
              more - file perusal filter for crt viewing

     Syntax:
              more [-dlfpcsu] [-<num>] [+/<pattern>] [+<line>] [file ...]

     Beschreibung:
              More ist ein Filterprogramm, das einen Texte 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

      Beispiele:
          more unix
          more -s unix           # Leerzeilen zusammenfassen
          more +30 unix          # ab 30.Zeile
          more +5 /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
```

**Beispiele:**

```
less  unix
less  /etc/passwd
```





**Auch Fileanfänge können interessant sein.**

**Kommando:**

```
head - output the first part of files
```

**Syntax:**

```
head [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
-n, --lines=ANZAHL      erste ANZAHL Zeilen statt 10 ausgeben
--help                  diese Hilfe anzeigen und beenden.
--version               Versionsinformation anzeigen und
```

**Beispiele:**

```
head /etc/passwd            # ersten 10 Zeilen
head -n 5 /etc/passwd       # ersten 5 Zeilen
head -c 20 /etc/passwd      # ersten 20 Bytes
head *                      # ersten 10 Zeilen von allen File
head -q *                   # ersten 10 Zeilen von allen
                            # Files ohne Filename
```

```
Ein schöner Rücken kann auch entzücken.
Nur das Ende von Files anschauen.

    Kommando:
              tail - output the last part of files

     Syntax:
              tail [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:
              -f, --follow, follow=<nr>
                        - Ausgabe der an ein File angefügten Daten,
                          für die Betrachtung von Protokollfiles.
              --help    - Hilfsinformationen anzeigen
              --version - Version ausgeben

     Beispiele:
              cd Texte
              tail unix
              tail -10 unix
              tail -n 10 unix
              tail -c 20 unix
              tail -v -c 20 unix
              tail -f /var/log/messages
```





```
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:
               --help   - Hilfe
                -b      - Binäre Ausgabe
                -c      - Character Ausgabe

     Beispiele:
            od -bc dat-file
            od -c  dat-file
```

**Wir erzeugen unser erstes File.**

**Eingabe eines Files über Standardeingabe.**

```
    Kommando:
              cat - concatenate files and print on the standard output

     Syntax:
              cat [OPTION] [FILE]...

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

     Optionen:
            --help                  Hilfe anzeigen und beenden.
            --version               Versionsinformation anzeigen und beenden

     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)
```





**Wie werde ich Files los?**

**Löschen von Files**

```
    Kommando:
              rm - remove files or directories
     Syntax:
              rm [OPTION]... FILE...

     Beschreibung:
              Löschen des angegeben Files. Standardmäßig werden
              keine Directories gelöscht. Das File und das Directory,
              in dem das File steht, müssen schreibar sein.

     Optionen:
              -f, --force       - File auch löschen, wenn schreibgeschützt,
                                  aber prinzipiell für den Nutzer möglich.
              -i, --interactive - interaktives Löschen, jedes File wird
                                  nachgefragt.
              -r, -R, --recursive
                                - rekursives Löschen von Directories.
              -v, --verbose     - nicht schweigsam
              --help            - Hilfetext auf Standardausgabe anzeigen
              --version         - Versionsnummer anzeigen
```

**Beispiele:**

```
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
```

**Für misstrauische Nutzer:**

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

**shred überschreibt das File erst und löscht es dann.**



8.Directory–und–Files–Kommandos 6.4.2010

**Löschen von Files im grossen Stil.**

**Löschen von Directories**

**Kommando:**
rmdir - remove empty directories

**Syntax:**
rmdir [OPTION]... DIRECTORY...

**Beschreibung:**
Löschen eines leeren Directories.

**Optionen:**
-v, --verbose
Diagnose für jedes bearbeitete Verzeichnis ausgeben
--help   diese Hilfe anzeigen und beenden.
--version
Versionsinformation anzeigen und beenden.

**Beispiele:**

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