Dateien und Verzeichnisse

Grundlagen

Öffne ein Terminal und gib folgendes ein:

```
pwd
```

Als Ausgabe am Bildschirm erhältst du beispielsweise:

```
/srv/samba/schools/default-school/teachers/sbel
```

Das ist dein "HOME Verzeichnis". Man sieht, dass das Verzeichnistrennzeichen eines Pfades unter Linux/Unix der Schrägstrich (englisch Slash) / (und nicht wie unter Microsoft Windows der Backslash \) ist.

Dadurch können wir auch einen neuen Begriff definieren: Ein **Pfad** ist eine Zeichenkette, bei der eine beliebige Anzahl von Verzeichnissen jeweils durch einen Schrägstrich getrennt sind.

Eine weitere Unterschied gegenüber Microsoft Windows ist, dass es keine Laufwerke (C:, D: usw.) gibt. Unter Unix bzw. Linux gibt es nur ein sogenanntes "root Verzeichnis" — im Deutschen auch mit **Wurzelverzeichnis** übersetzt. Die Bezeichnung Laufwerk wird unter Unix nicht verwendet.

Man spricht immer von Partitionen, Medien und Mountpunkten.

Unter Unix/linux wird, im Gegensatz zu Microsoft Windows, auch zwischen Groß– und Kleinschreibung unterschieden.

```
/HOME/laptop
```

und

```
/home/laptop
```

bezeichnen also zwei völlig verschiedene Verzeichnisse.

Eine Trennung nach Dateierweiterungen wie unter Microsoft Windows (z.B. *.exe oder *.doc ist unter Unix in manchen Gebieten zwar üblich (z.B. *.tex für LaTex Dokumente oder *.odt für LibreOffice Text-Dokumente) aber keineswegs zwingend — so haben ausführbare Dateien meist gar keine Endung (z.B. lautet der Programmname der meistverwendeten Shell einfach `bash`, der Programmname des Firefox-Browsers einfach `firefox`).

Autocomplete mit TAB

Bei der bash (und einigen anderen Shells) ist eine Autocomplete-Funktion. Wenn du einen Befehl oder einen Dateinamen unvollst ändig angibst und dann die TAB Taste drückst, wird der Name

vervollständigt, falls er eindeutig ist, d.h. wenn es nur eine Möglichkeit zur Vervollständigung gibt.

Gibt es mehrere Möglichkeiten, kannst du zweimal die TAB Taste drücken. Nun listet die bash sämtliche Möglichkeiten auf (ab einem voreingestellten Wert wird abgefragt, ob du wirklich alle Möglichkeiten sehen willst, weil die Liste dann teileweise sehr lang ist).

Arbeiten mit Verzeichnissen

Nach dem Einloggen befindest du dich in deinem HOME-Verzeichnis. Gib Folgendes ein:

```
pwd
```

Der Befehl pwd (print working directory) zeigt dir an, in welchem Verzeichnis du dich sich gerade befindest. Du erhältst jetzt eine Antwort, die die Ausgabe des Befehls zeigt, zum Beispiel:

```
/home/c102/c102mr
```

Wie auch unter Microsoft Windows dient zum Wechseln des Verzeichnisses der Befehl `cd`.

```
cd Dokumente
```

Nun gibst du wieder pwd ein. Die Ausgabe sollte in etwa wie folgt aussehen:

```
/srv/samba/schools/default-school/teachers/sbel/Dokumente
```

Nun wollen wir noch einmal ein `cd` ausführen. Gib ein

```
cd /etc
```

Wenn du jetzt pwd ausführst, lautet das Ergebnis:

```
/etc
```

Achte auf den feinen Unterschied zwischen den beiden Eingaben von cd. Beim zweiten Mal steht vor dem etc ein `/`. Du hast bereits das Trennzeichen `/` für Verzeichnisse kennengelernt - und genau das ist es hier.

Was wir hier lernen, ist der vielleicht bereits aus Microsoft Windows bekannte Unterschied zwischen absoluten und relativen Dateiangaben.

Der erste Befehl wechselt in das Verzeichnis "Dokumente" - und zwar abhängig vom aktuellen Verzeichnis. Da wir uns im HOME-Verzeichnis befinden, wechselt er dort in das Verzeichnis "Dokumente", falls es vorhanden ist. Wären wir beispielsweise in `/usr`, würde der Befehl versuchen dort in das "bin"-Verzeichnis zu wechseln (also nach `/usr/Dokumente`), was nicht klappen würde, weil es dieses Verzeichnis nicht gibt. Das nennt man **relative** Dateiangaben oder relative Pfadangaben.

Der zweite Befehl wechselt in das Verzeichnis `/etc`. Das ist ein **absoluter** Verzeichnisname (oder ein

"absoluter Pfad"). Unabhängig davon, wo wir uns gerade befinden, wechselt dieser Befehl nach `/etc`. Absolute Dateiangaben listen immer sämtliche Verzeichnisse im Pfad vom Root-Verzeichnis ausgehend bis zum Zielverzeichnis oder zur Zieldatei auf. Wollten wir in das "Dokumente"-Verzeichnis in unserem HOME-Verzeichnis wechseln, so müssten wir (entsprechend angepasst) eingeben:

```
cd /srv/samba/schools/default-school/teachers/sbel/Dokumente
```

Hier ist die Tab-Taste wieder dein Freund, denn das alles einzutippen macht wenig Freude.

Wechsle jetzt ins Root-Verzeichnis. Gib einfach ein:

```
cd /
```

Jetzt möchten wir versuchen, wieder in dein HOME-Verzeichnis zu kommen. Du könntest jetzt eingeben: `cd/srv/samba/schools/default-school/…`. Aber das ist ziemlich lang. Hier gibt es eine einfache Lösung. Du verwendest dazu das Tilde-Zeichen ~. Das steht immer für dein HOME-Verzeichnis. Du gibst also ein:

```
cd ~
```

Oben sind wir im `bin`-Verzeichnis in deinem HOME-Verzeichnis gelandet - das war etwas kompliziert, wie wir jetzt wissen. Es wäre viel einfacher gegangen:

```
cd ~/Dokumente
```

Eine Parallele zu Microsoft Windows ist die Verwendung von `..`, um das übergeordnete Verzeichnis zu bezeichnen. Du befindest dich jetzt im `Dokumente`-Verzeichnis in deinem HOME-Verzeichnis, wenn du die oben beschriebenen Schritte befolgt hast. Falls ein Verzeichnis namens "Desktop" existiert, kannst du dorthin wechseln, indem du Folgendes eingibst:

```
cd ../Desktop
```

Im Gegensatz dazu wird das aktuelle Verzeichnis mit `.` bezeichnet. Wenn sich in deinem Verzeichnis ein Programm namens `test` befindet, könntest du es durch die folgende Zeile ausführen:

```
./test
```

Jetzt weißt du also bereits, wie du von Verzeichnis zu Verzeichnis wechseln kannst.

Verzeichnisse erstellen

Der Befehl zum Erstellen von Verzeichnissen ist "mkdir" (make directory). Wenn du also in deinem HOME-Verzeichnis ein Verzeichnis namens "junk" erstellen möchtest, gib einfach ein:

```
mkdir ~/junk
```

Zum Entfernen von Verzeichnissen wird "rmdir" verwendet. Allerdings muss das Verzeichnis leer sein, damit du es entfernen kannst. Lass uns zuerst ein neues Verzeichnis erstellen:

```
mkdir ~/test
```

Dieses Verzeichnis benötigen wir jetzt nicht mehr. Wir entfernen es daher mit dem Befehl "rmdir" (remove directory):

```
rmdir ~/test
```

Du kannst das "junk"-Verzeichnis, wenn du möchtest, behalten. Es ist oft eines der Verzeichnisse, die in den meisten HOME-Verzeichnissen vorhanden sind (junk bedeutet auf Englisch so viel wie "Trödel" oder "Plunder"). Es wird in der Regel für Tests von Software und ähnlichem verwendet.

Zum Verschieben von Verzeichnissen kannst du den Befehl "mv" (move) verwenden. Erstelle also das "test"-Verzeichnis erneut und verschiebe es nach "Bloedsinn":

```
mkdir ~/test
mv ~/test ~/Bloedsinn
```

Wie wir später sehen werden, wird der "mv"-Befehl auf die gleiche Weise auch bei normalen Dateien verwendet.

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:git:shell:dateien_und_verzeichnisse:start?rev=1695278245

Last update: **21.09.2023 06:37**