1.2.V Anwendung/Befehle – Versuch

1.2.V.1 Beispiel – Anlegen eines Verzeichnisbaums

Es sollen im persönlichen Verzeichnis folgende Verzeichnisse angelegt werden:

- `L12V1`
- `L12V1/unter1`
- `L12V1/unter2`
- `L12V1/unter1/uu`

Außerdem sollen im Verzeichnis `uu` die Dateien `neu1.txt` und `neu2.txt` angelegt werden. — Nichts einfacher als das!

Terminal

```
schueler@debian964:~$ cd                    # Gehe ins persoenl. Verz.
schueler@debian964:~$ mkdir L12V1 # Lege L12V1 an
schueler@debian964:~$ cd L12V1 # dort ist mehr zu tun: hineinwechseln
schueler@debian964:~/L12V1$ mkdir unter1 unter2 # mehrere Verz.
schueler@debian964:~/L12V1$ mkdir unter1/uu
schueler@debian964:~/L12V1$ touch unter1/uu/neu1.txt unter1/uu/neu2.txt
```

Man kann also mit den Befehlen `mkdir` und `touch` auf einmal mehrere Verzeichnisse und Dateien anlegen. Das ist typisch für Linux-Dienstprogramme, dass sie eine Massenverarbeitung von Dateien und Verzeichnissen erlauben. Dazu später mehr im Zusammenhang mit *Wildcards*.

1.2.V.2 Beispiel – Objektnamen mit Minuszeichen

Der Name einer Datei oder eines Verzeichnisses ist frei wählbar. Es gibt zwar bestimmte Sitten (in der Informatik *Konventionen* genannt) wie z. B. bestimmte Endungen für bestimmte Dateien, aber sie unterscheiden sich auf verschiedenen Plattformen, und längst nicht jeder hält sich daran. Und so kann es passieren, das jemand eine Datei genauso nennt wie die Option eines Befehls:

Terminal

```
schueler@debian964:~$ cd                  # ins persoenl. Verz.
schueler@debian964:~$ mkdir L12V2         # Lege ein Verz. an
schueler@debian964:~$ touch L12V2/-neu.txt # dort Datei -neu.txt anl.
schueler@debian964:~$ touch L12V2/-lustig # dort Datei -lustig anl.
schueler@debian964:~$ cd L12V2         # Wechsle in Verzeichnis
schueler@debian964:~/L12V2$ ls -l # Zeige Verz.-Inh. an
```

Nun sollen die Dateien `-lustig` und `-neu.txt` angezeigt werden:

Terminal

```
schueler@debian964:~/L12V2$ ls -lustig
insgesamt 0
3161438 0 -rw-r--r-- 1 schueler 0 Apr 23 18:29 -lustig
3161437 0 -rw-r--r-- 1 schueler 0 Apr 23 18:29 -neu.txt
```

Es erstaunt, dass die Namen und Eigenschaften beider Dateien angezeigt werden, obwohl nur eine genannt wurde. Und die Art der Anzeige ist auch etwas merkwürdig. Nun sollen beide Dateien, eine nach der anderen, gelöscht werden:

Terminal

```
schueler@debian964:~/L12V2$ rm -lustig
rm: Ungültige Option -- l
Versuchen Sie „rm ./-lustig“, um die Datei '-lustig' zu entfernen.
„rm --help“ liefert weitere Informationen.
```

Offenbar hält `rm` den Namen `-lustig` für eine Option oder sogar Reihe von Optionen. Das wird deutlich, wenn man die Fehlermeldung ansieht. Merke: **Datei- oder Verzeichnisnamen, die mit einem Minuszeichen beginnen, können ein Problem sein.**

Wie löst man nun dieses Problem? Eine Möglichkeit wird in der Fehlermeldung genannt: Man erweitert den Namen so, dass man einen Verzeichnisnamen davorschreibt. Da es sich um Dateien im aktuellen Verzeichnis handelt, kann man den Namen des aktuellen Verzeichnisses (er hat den Punkt als Namen) davorschreiben:

Terminal
```
schueler@debian964:~/L12V2$ rm ./-lustig
```

Man hätte auch den absoluten Pfadnamen verwenden können. Um die zweite Datei zu löschen, können wir spaßeshalber die andere Möglichkeit benutzen: Viele (nicht alle) Programme haben eine spezielle Option, mit der sie das Ende aller Optionen auf dieser Befehlszeile markieren, nämlich die Option `--`. Alle Namen, die nach dieser Option kommen, werden vom Programm nicht als Optionen, sondern als normale Parameter angesehen:

Terminal
```
schueler@debian964:~/L12V2$ rm  --  -neu.txt
```

In dieser Befehlszeile liegt der Dateiname `-neu.txt` nach der Option `--` und wird deshalb als Dateiname verstanden. So kann die Datei gelöscht werden.

1.2.V.3 Beispiel – Objektnamen mit Leerzeichen und Sonderzeichen

Viele Menschen möchten ihre Datei- und Verzeichnisnamen mit Leerzeichen ausstatten. Für die Befehlszeile ist das aber ein gewisses Problem. Wir wollen ein Verzeichnis einrichten mit dem Namen `ITS 27.04.2021 3.+4. Stunde`:

Terminal
```
schueler@debian964:~$ cd
schueler@debian964:~$ mkdir L12V3
schueler@debian964:~$ cd L12V3
schueler@debian964:~/L12V3$ mkdir ITS 27.04.2021 3.+4. Stunde
schueler@debian964:~/L12V3$ ls -l
ls -l
insgesamt 16
drwxr-xr-x 2 schueler schueler 4096 Apr 23 18:47 27.04.2021
drwxr-xr-x 2 schueler schueler 4096 Apr 23 18:47 3.+4.
drwxr-xr-x 2 schueler schueler 4096 Apr 23 18:47 ITS
drwxr-xr-x 2 schueler schueler 4096 Apr 23 18:47 Stunde
```

Das ist nicht das, was wir wollten: Offenbar hat die Shell unseren Verzeichnisnamen falsch verstanden und vier Verzeichnisse angelegt. Nach jedem Leerzeichen in der Befehlszeile wurde ein neues Wort gefunden. Dieses Wort wurde als Wunschname für ein neu anzulegendes Verzeichnis interpretiert. Erst einmal müssen wir alle Verzeichnisse löschen:

Terminal
```
schueler@debian964:~/L12V3$ rmdir ITS 27.04.2021 3.+4. Stunde
schueler@debian964:~/L12V3$ ls -l
insgesamt 0
```

Das hat immerhin geklappt. Was können wir tun? Das Leerzeichen gilt in der Shell als ein *Sonderzeichen*. Sonderzeichen haben besondere Bedeutungen auf der Befehlszeile. Das Leerzeichen hat die Bedeutung eines Worttrenners. Es trennt den Befehlsnamen `rmdir` von den Parametern und die Parameter untereinander.

Wir haben aber einen Verzeichnisnamen, innerhalb dessen die drei Leerzeichen nicht diese Rolle spielen sollen. Innerhalb des Namens sollen die Leerzeichen versteckt werden, so dass sie nicht als Worttrenner dienen können. Man sagt, sie sollen *maskiert* werden. Netterweise gibt es dazu gleich drei Möglichkeiten:

a) Maskierung mit Backslash:
`ITS\ 27.04.2021\ 3.+4.\ Stunde` – jedes Leerzeichen wird durch einen vorangehenden Backslash (AltGr ß) unwirksam gemacht

b) Maskierung mit Hochkommata:
`'ITS 27.04.2021 3.+4. Stunde'` – alle Leerzeichen im Namen werden durch Umfangen mit Hochkommas (Shift ⇑ #) unwirksam gemacht

c) Maskierung mit Anführungszeichen:
`"ITS 27.04.2021 3.+4. Stunde"` – alle Leerzeichen im Namen werden durch Umfangen mit Anführungszeichen (Shift ⇑ 2) unwirksam gemacht

Alle drei Varianten sind gleich wirksam[1]. Deshalb können wir uns also eine beliebige Variante aussuchen, um das Verzeichnis doch noch erstellen zu können:

Terminal
```
schueler@debian964:~/L12V3$ mkdir "ITS 27.04.2021 3.+4. Stunde"
schueler@debian964:~/L12V3$ ls -1
'ITS 27.04.2021 3.+4. Stunde'
```

Die Hochkommata bei der Anzeige von `ls` zeigen an, dass im Namen Sonderzeichen vorhanden sind[2]. Sie zeigen *nicht* an, dass im Namen Hochkommata vorhanden sind.

[1] Den Backslash selbst kann man durch zwei Backslashs nacheinander darstellen. Ein Hochkomma kann man in der dritten Variante darstellen, ein Anführungszeichen in der zweiten.

[2] Und sie erlauben es, die Ausgabe von `ls` in in weiteren Programmen zu verwenden.