

# Unix – ganz schmerzlos

## Lösungen zum Übungszettel 1

**Aufgabe 1:** Der `ls`-Befehl (wie viele andere Unix-Befehle) zeigt durch Angabe der Option `--help` eine Übersicht seiner Optionen an. Wahlweise kann man sich mit „`man ls`“ seine man-page anzeigen lassen. Auf diese Weise findet man die Option `-F`. Mit dieser Option werden spezielle Zeichen an die Datei- und Verzeichnisnamen angehängt:

- Verzeichnisse werden durch ein `/` gekennzeichnet
- ausführbare Programme durch einen Stern `*`
- normale Dateien werden nicht speziell gekennzeichnet

**Aufgabe 2:** Die Wildcard wird von der Shell interpretiert, d.h. die Shell ersetzt den Wildcard-Ausdruck durch eine Liste aller Dateien, die auf ihn passen. Wenn nur eine Datei auf das Muster passt, wird genau diese eine Datei eingesetzt. Wenn also das Muster `*.txt` z.B. nur auf die Datei `foo.txt` passt, erzeugt die Shell den Befehl

```
cp foo.txt texte
```

und führt ihn aus. Wenn es kein Unterverzeichnis `texte` gibt, legt der `cp`-Befehl einfach eine Datei `texte` an, die eine Kopie der Datei `foo.txt` ist.

Man kann dem `cp`-Befehl mitteilen, daß das Ziel ein Verzeichnis sein soll, indem man einen `/` an den Verzeichnisnamen anhängt:

```
juser@hobel> cp *.txt texte/
```

Selbst wenn das Muster jetzt wieder nur auf eine Datei passt, also die Anweisung

```
cp foo.txt texte/
```

entsteht, muss das Unterverzeichnis `texte` existieren. Der `cp`-Befehl weigert sich, eine Datei `texte` anzulegen, da ausdrücklich ein Verzeichnis gemeint war.

**Aufgabe 3:** Das Verzeichnis „`.`“ ist das jeweils aktuelle Verzeichnis, je nachdem, wo man sich gerade im Dateibaum aufhält. Man kann dieses Verzeichnis in Pfad-Angaben genauso verwenden wie jedes andere Verzeichnis auch. Anstatt

```
juser@hobel> xv bilder/versuch3.png
```

kann man auch

```
juser@hobel> xv ./bilder/versuch3.png
```

sagen. Aber man kann auch

```
juser@hobel> xv ./bilder/././versuch3.png
```

sagen, und greift doch jedesmal auf die gleiche Datei zu. Beachte: Im letzten Beispiel bezeichnen der zweite und der dritte Punkt ein anderes Verzeichnis als der erste Punkt.

Das Verzeichnis „.“ ist manchmal hilfreich, wenn man explizit zum Ausdruck bringen will „Folge einem Pfad, der in diesem Verzeichnis beginnt.“ Ausserdem kann man mit seiner Hilfe Dateien wie `-igitt` löschen.

Der naheliegende Versuch, die Datei `-igitt` zu löschen, schlägt fehl:

```
juser@hobel> rm -igitt
rm: invalid option -- g
Try `rm --help' for more information.
```

Der `rm`-Befehl erwartet nach einem Minus-Zeichen die Angabe von Optionen. Es sieht für ihn so aus, als ob er mit den Optionen `-i`, `-g`, `-i`, `-t` und nochmals `-t` aufgerufen würde. Die Option `-i` kennt der `cp`-Befehl, aber die Option `-g` ist ihm unbekannt. Daher kommt es zur Fehlermeldung.

Um die Datei aber dennoch zu löschen, muss man also den Dateinamen so angeben, dass er nicht mit einem Minus-Zeichen beginnt. Dies kann man am einfachsten mit dem „.“-Verzeichnis erreichen:

```
juser@hobel> rm ./-igitt
```

Nun kann der Dateiname nicht mehr irrtümlich als Folge von Optionen interpretiert werden, und die Datei wird anstandslos gelöscht.

Die andere, in der man-page beschriebene Möglichkeit, die Datei zu löschen, beruht auf einer Konvention, die von einer Reihe von Befehlen verstanden wird: Wenn man einem Befehl die Option „--“ mitgibt, werden alle nach dieser Option folgenden Worte als normale Argumente interpretiert, selbst wenn sie mit einem Minus-Zeichen beginnen. Die Datei könnte also auch mit dem Befehl

```
juser@hobel> rm -- -igitt
```

gelöscht werden, da `-igitt` jetzt nur noch als normaler Dateiname verstanden wird.