

# Unix – ganz schmerzlos

## Übungszettel 8

**Aufgabe 23:** Schreibe ein Skript, das die Fakultät einer Zahl berechnet. Zur Erinnerung, die Fakultät  $n!$  einer natürlichen Zahl  $n$  ist definiert als

$$n! \stackrel{\text{def}}{=} n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$$

**Aufgabe 24:** Vervollständige das in der Vorlesung gezeigte Programmfragment zum Konvertieren von verschiedenen Grafikformaten. Verwende dazu folgende Hilfsmittel:

- Es gibt eine Reihe von Programmen, die GIF-, JPEG und TIFF-Bilder in ein gemeinsames Zwischenformat konvertieren, PPM genannt. Aus diesem Zwischenformat läßt sich mit dem Programm `pnmtopng` eine PNG-Datei erzeugen. Im einzelnen sind die Konversionsprogramme:

Format	Programm
GIF	<code>giftopnm</code>
JPEG	<code>djpeg</code>
TIFF	<code>tifftopnm</code>

- Alle Programme geben das konvertierte PPM-Bild auf der Standardausgabe aus, und `pnmtopng` kann von der Standardeingabe lesen. Um also ein GIF-Bild in einem Schritt in ein PNG-Bild umzuwandeln, kann man die folgende pipe verwenden:

```
giftopnm foo.gif | pnmtopng > foo.png
```

Schreibe ein Programm, das eine Liste von Bildern abarbeitet (die man als Argumente übergeben kann) und diese in PNG-Dateien umwandelt. Dabei soll der Dateiname erhalten bleiben, lediglich die Suffixes `.gif`, `.jpg` bzw. `.tif` sollen durch `.png` ersetzt werden. Wenn es bereits eine PNG-Datei mit dem Namen gibt (z.B. weil man die Bilder `foo.gif` und `foo.jpg` gleichzeitig konvertieren möchte), soll eine Warnung ausgegeben werden, und das Ausgangsbild übersprungen werden.

**Aufgabe 25:** Im Verzeichnis `/vol/lehre/Unix-Kurs/` befinden sich drei Versionen des in der Vorlesung gezeigten Programms zur Quadrat-Berechnung. Es sind ein Shell-Skript (`square.sh`), ein Perl-Programm (`square.pl`) und ein C-Programm (`square`, der Quelltext ist in `square.c` zu finden). Rufe alle drei Programme nacheinander auf und achte darauf, wieviel Zeit die Programme benötigen. Vergleiche Deine subjektiven Eindrücke.

Um das Laufzeitverhalten besser beurteilen zu können, gibt es das Kommando `time`, das man vor einen Programmaufruf setzen kann. Wenn das eigentliche Programm beendet ist, zeigt `time` eine Statistik an, wieviel Zeit benötigt wurde. Wiederhole den obigen Vergleich mit Hilfe des `time`-Kommandos.

Sieh Dir den Quelltext der drei Programme an. Welches Programm läßt sich am besten lesen, welches am schlechtesten?