

# Unix – ganz schmerzlos

## Lösungen zum Übungszettel 7

**Aufgabe 19:** Das folgende Skript erzeugt den gewünschten Countdown:

```
#!/bin/sh

for num in 10 9 8 7 6 5 4 3 2 1 Feuer; do
    echo $num
    sleep 1
done
```

**Aufgabe 20:** Dieses Skript zeigt die Befehle an, mit denen die angegebenen Dateien „versteckt“ werden. Wenn man die echo-Befehle entfernt, würden sie ausgeführt werden:

```
#!/bin/sh

# arbeite die Liste der Argumente ab
for file in $*; do
    # ueberpruefe, ob Zieldatei bereits vorhanden
    if [ -f .$file ]; then
        echo .$file existiert bereits, ueberspringe $file
    else
        # das ist nicht der Fall, also verstecke die Datei
        echo mv $file .$file
        echo chmod go-rwx .$file
    fi
done
```

Vorsicht: Das Skript kann nur mit Dateien im aktuellen Verzeichnis umgehen. Wenn der Dateiname ein Pfad ist, wird durch einfaches Voranstellen eines Punktes ein falscher Name erzeugt. Die Lösung dieses Problems wird auf dem nächsten Übungszettel besprochen.

**Aufgabe 21:** Dieses Skript erwartet zwei Zahlen als Argumente und gibt die größere der beiden wieder aus:

```
#!/bin/sh

if [ "$1" -gt "$2" ]; then
    echo $1
else
    echo $2
fi
```

Das Skript läßt sich leicht so erweitern, daß die größte von drei Zahlen ausgegeben wird:

```
#!/bin/sh

if [ "$1" -gt "$2" ]; then
    if [ "$1" -gt "$3" ]; then
        echo $1
    else
        echo $3
    fi
else
    if [ "$2" -gt "$3" ]; then
        echo $2
    else
        echo $3
    fi
fi
```

Dieses Verfahren ließe sich jetzt auch bei vier, fünf usw. Zahlen anwenden, allerdings ist dies nicht sonderlich praktikabel. Daher ist für den allgemeinen Fall eine andere Vorgehensweise sinnvoller:

```
#!/bin/sh

min=$1
max=$2

for zahl in $*; do
    if [ $zahl -lt $min ]; then
        min=$zahl
    fi
    if [ $zahl -gt $max ]; then
        max=$zahl
    fi
done

echo Minimum: $min, Maximum: $max
```

In den Variablen `min` bzw. `max` merkt sich das Skript das bisher gefundene Minimum bzw. Maximum. Wenn eine Zahl kleiner als das bisherige Minimum oder größer als das bisherige Maximum ist, merkt sich das Skript die neue Zahl. Um die beiden Variablen mit sinnvollen Werten zu initialisieren, wird zunächst die erste Zahl als aktuelles Minimum bzw. Maximum angenommen.

**Aufgabe 22:** Im ersten Schritt sieht das Skript so aus:

```
#!/bin/sh

for printer in $*; do
    lpstat -p $printer
done
```

Wenn keine Argumente angegeben werden, soll wieder eine vorgegebene Liste von Druckern verwendet werden. Das könnte z.B. so aussehen:

```
#!/bin/sh

if [ $# = 0 ]; then
    liste="w6ps w6color w7ps"
else
    liste=$*
fi

for printer in $liste; do
    lpstat -p $printer
done
```