

# Linux Kurs

## Blatt 3

Konstantin Schneider

21.03.2022

### bashfirst.sh

Schreiben Sie ein bash-Skript **bashfirst.sh**, das als Parameter einen Filenamen erwartet. Als erstes soll überprüft werden, ob dieser Eintrag überhaupt existiert. Wenn nein, Abbruch mit Meldung. Wenn ja, Überprüfung, ob File oder Directory.

```
#!/bin/bash

INPUT=$1

if [ -d "$INPUT" ]
then
    echo "Dies ist ein Ordner."
else
    if [ -f "$INPUT" ]
    then
        echo "Dies ist eine Datei."
    else
        echo "Datei oder Ordner existiert nicht!"
    fi
fi
```

### bashin.sh

Schreiben Sie ein bash-Skript **bashin.sh**, das zwei Namen von der Konsole einliest und diese in einem Array speichert. Erweitern Sie ihr Programm dann so, dass solange Namen eingelesen werden können, bis der User **n** eingibt. Anschließend sollen alle Namen ausgegeben werden.

```
#!/bin/bash

declare -a NAMES

read -p "Welcher Name soll eingelesen werden? " NAMES[0]

i=1
while true; do
```

```

read -p "Einen weiteren Namen einlesen? (y/n) " USER_INPUT
if [ $USER_INPUT = y ]; then
    read -p "Welcher Name soll eingelesen werden? " NAMES[$i]
    ((i=i+1))
else
    if [ $USER_INPUT = n ]; then
        break
    else
        printf "Fehler! Bitte nur y oder n eingeben!\n"
    fi
fi
done

echo
printf "Sie haben folgende Namen eingegeben:\n"
delim=""
for i in "${NAMES[@]}"; do
    printf "%s" "$delim$i"
    delim=", "
done
echo

```

## dusubdir

Sie wollen endlich mal Ihren Plattenspeicher ausmisten. Dazu hätten Sie ganz gerne eine Liste, die zu all Ihren Subdirectories die jeweilige Gesamt-Speicherbelegung ausgibt. Schreiben Sie ein bash-Skript `dusubdir`, das ein Startverzeichnis als Parameter erwartet. Wenn dieses nicht existiert, Abbruch, ansonsten Ausgabe wie z.B.:

```

pc58643:~> dusubdir ~/Themen.Fritz
1616 /home/wuf04055/Themen.Fritz/CKurs
13600 /home/wuf04055/Themen.Fritz/dbtrans
1136 /home/wuf04055/Themen.Fritz/demos.div

```

```

#!/bin/bash

DIR=$1

for i in $(find $DIR -mindepth 1 -maxdepth 1 -type d); do
    du -hs $i
done

```

## makesoftlist

Sie haben komplett den Überblick verloren, welche Programme auf Ihrem Rechner in `/usr/bin` vorhanden sind. Es gibt hier zu sehr vielen Binaries eine `man-page`. man mit der Option `-f` oder auch `whatis` liefert einen ganz kurzen Text, der hier ausreicht, z.B.

```

pc58643:~> man -f pdflatex
pdflatex (1) - PDF output from TeX

```

Erzeugen Sie also mit einem bash-Skript namens `makesoftlist` eine Liste aller Binaries aus `/usr/bin` mit entsprechender Kurzbeschreibung. Beispiele:

```
lynx (1) - a general purpose distributed information browser
lyx (1)  - A Document Processor
lz (1)   - gunzips and shows a listing of a gzip 'd tar 'd archive
```

Implementieren Sie zwei Besonderheiten:

- **Pflicht:** Da es in `/usr/bin` sehr, sehr viele Dateien gibt, geben Sie als ersten Parameter die Anzahl der Dateien an, die ausgewertet werden sollen, also z.B. `makesoftlist 20` Default sei 10.
- **Kür:** Wenn keine Manpage vorhanden ist, gibt der `man`-Befehl eine Fehlermeldung auf `stderr` aus. Fangen Sie diese Fehlermeldung ab und geben sie in Ihrer Liste eine eigene, kurze Anmerkung dazu aus.

```
#!/bin/bash

DIR="/usr/bin"
MAX_ROWS="${1:-10}"

k=0
for i in $(find $DIR -mindepth 1 -maxdepth 1 -type f); do
    man -f $i 2> /dev/null \
        || echo "<!!!> No manpages for $(basename $i) :( <!!!>"
    ((k=k+1))
    if [ $k -eq $MAX_ROWS ]; then
        break
    fi
done
```