

Systemmanagement und Sicherheit

4. Übung

Aufgabe 1 (Shell-Programmierung)

- a) Schreiben Sie ein Shell-Skript, das vor jedes Argument den String „Hallo“ setzt. Beispiel:

```
./hallo2 Peter Stefan Michael  
Hallo Peter  
Hallo Stefan  
Hallo Michael
```

- b) Schreiben Sie ein Shell-Skript `viewer`, der abhängig vom Art des Inhalts einer angegebenen Datei ein entsprechendes Programm zum Anzeigen der Datei aufruft. Falls die Datei eine Grafikdatei ist, soll beispielsweise `/usr/local/bin/xv` aufgerufen werden.

Die Unterscheidung der Inhaltstypen von Dateien können Sie treffen, indem Sie `file` aufrufen, wie im folgenden Beispiel:

```
$ file tomate.jpg  
tomate.jpg: JPEG image data, JFIF standard 1.01 ...
```

Unterscheiden Sie mindestens Bilddateien (`xv`), PDF-Dateien (`xpdf`), Textdateien (`less`) und Open-Document Texte (`libreoffice`).

- c) Schreiben Sie ein Shell-Skript `wavtomp3`, das WAV-Dateien in MP3-Dateien umwandelt. Hierfür können Sie `ffmpeg` benutzen.
- d) Schreiben Sie ein Shell-Skript `jpgtopng`, das JPEG-Dateien in PNG-Dateien umwandelt. Hierfür können Sie `djpeg` und `pnmtopng` benutzen.
- e) Schreiben Sie Shell-Skripte `counter1`, `counter2`, die für zwei Zahlen a , b als Argumente

- i) alle ganzen Zahlen
ii) alle 2^i -Potenzen 2^i zwischen a und b druckt. Beispiel:

```
./counter1 17 20  
17 18 19 20  
./counter2 15 100  
16 32 64
```

- f) Starten Sie vorigen Shell-Skripte mit Hilfe von `sh`

```
sh counter1
sh -v counter1
sh -x counter1
```

und beachten Sie die Unterschiede.

- g) Schreiben Sie ein Shell-Skript `good`, das abhängig von der aktuellen Uhrzeit entweder
- Guten Morgen (04:00-08:59) oder
 - Guten Tag (09:00-18:59) oder
 - Guten Abend (19:00-22:59) oder
 - Gute Nacht (23:00-03:59)

ausgibt. Hinweis: Benutzen Sie `date` und `cut`.

- h) Schreiben Sie zwei Shell-Skripte `calc1`, `calc2`, die zwei Argumente a, b akzeptieren und den Wert

$$\frac{a^5 - a}{30} + b$$

ausgeben (ganzzahlige Division).

```
./calc1 3 3
11
```

`calc1` soll mit `expr` arbeiten und `calc2` mit der arithmetischen Auswertung `$((...))`.

- i) Schreiben Sie ein Shell-Skript `chkfile`, das beliebig viele Dateien auf ihre Existenz überprüfen kann. Falls eine Datei keine reguläre Datei ist, so soll eine Fehlermeldung ausgegeben werden. Beispiel:

```
./chkfile a b c d
a existiert
b existiert nicht
c ist keine regulaere Datei
d existiert
```

- j) Schreiben Sie ein Shellskript, das das Kommando `which` emuliert, siehe Manualpage `which(1)`.
- k) Schreiben Sie ein Shellskript für beliebig viele auf der Kommandozeile angegebene Dateinamen. Das Skript soll alle diejenigen, die auf `.doc` enden, mit der Endung `.docx` versehen. Die anderen eventuell angegebenen Dateien sollen nicht umbenannt werden. Setzen Sie hierfür den `trim`-Operator (siehe Folie 195) für Suffixe ein (%). Die Erläuterung des `trim`-Operators finden Sie in der Manualpage von `sh` unter *Remove Smallest Suffix Pattern*.