

Systemmanagement und Sicherheit

3. Übung

Aufgabe 1 (Shell-Programmierung)

- a) Schreiben Sie ein Shell-Skript, das eine beliebige Anzahl von Argumenten akzeptiert und diese ausgibt, nachdem der String „Hallo“ ausgegeben wurde. Beispiel:

```
./hallo1 Peter Stefan Michael  
Hallo Peter Stefan Michael
```

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

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

- c) Benutzen Sie eine `for`-Schleife, um mit einem Shellskript `mover` eine Liste von Files vom aktuellen Directory in ein anderes Directory zu verschieben. Gehen Sie beim Entwickeln schrittweise vor:

- statt des tatsächlichen Verschiebens geben Sie per `echo` nur jeweils das Kommando aus, das eine Datei verschieben soll
- die Liste von Files ist Teil des Shell-Skripts
- die Liste von Files steht in einer Datei `liste`

- d) Schreiben Sie ein Shell-Skript, das für zwei Zahlen a , b als Argumente alle ganzen Zahlen zwischen a und b druckt.

```
./counter 17 20  
17 18 19 20
```

- e) Starten Sie voriges Shell-Skript `counter` mit Hilfe von `sh`

```
sh counter  
sh -v counter  
sh -x counter
```

und beachten Sie die Unterschiede.

- f) Schreiben Sie ein Shell-Skript `good`, das abhängig von der aktuellen Uhrzeit entweder
- Guten Morgen (04:00-09:59) oder
 - Guten Tag (10:00-17:59) oder
 - Guten Abend (18:00-22:59) oder
 - Gute Nacht (23:00-03:59)

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

- g) Schreiben Sie ein Shell-Skript `calc`, das zwei Argumente a, b akzeptiert und den Wert

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

ausgibt (ganzzahlige Division).

```
./calc 3 3
11
```

- h) Schreiben Sie ein Shell-Skript `countc`, das die Anzahl der im aktuellen Directory vorhandenen C-Dateien (Endung `.c`) ausgibt (Hinweis: `ls` und `wc` benutzen).
- i) Erweitern Sie `countc` um die Fähigkeit, das Directory von der Standardeingabe zu lesen (Shellkommando `read`).
- j) Erweitern Sie `countc`, das es die Ausgabe „zu viele“ erzeugt, wenn es mehr als 10 C-Dateien sind.
- k) 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
```