

Übungsblatt 5: Shell-Skripte 2

23. November 2018

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist **Freitag, 30.11.2017, 11:00 Uhr**
- Schickt die Lösungen bitte per Email an Euren Tutor:
 - Montag 14:00–15:30: Grant Cates (gcates@icp.uni-stuttgart.de)
 - Dienstag 9:45–11:15: Kai Szuttor (kai@icp.uni-stuttgart.de)
 - Dienstag 15:45–17:15: Julian Michalowsky (jmichalowsky@icp.uni-stuttgart.de)
 - Mittwoch 15:45–17:15: Michael Kuron (mkuron@icp.uni-stuttgart.de)
 - Donnerstag 9:45–11:15: Johannes Zeman (zeman@icp.uni-stuttgart.de)
- Die Übungen sollen in Gruppen von jeweils *zwei bis drei* Leuten bearbeitet werden. Abgaben von Einzelpersonen werden nicht akzeptiert. Bitte gebt *nur eine Lösung pro Gruppe* ab und nennt in eurer Abgabe alle Mitglieder eurer Gruppe!

Aufgabe 5.1: Dateinamen mit Zeitstempel versehen (2 Punkte)

Schreibe ein Skript, das als einzigen Parameter den Pfad zu einem Zielverzeichnis (ohne ‘/’ am Pfadende!) übergeben bekommt. Das Skript soll innerhalb dieses Verzeichnisses sämtliche Dateien derart umbenennen, dass zwischen ursprünglichem Dateinamen und Dateiendung ein Zeitstempel nach folgendem Schema eingefügt wird: “-<zweistellige Jahreszahl><zweistellige Monatszahl><zweistellige Tageszahl>-<zweistellige Stundenzahl>:<zweistellige Minutenzahl>”. Das folgende Listing demonstriert das gewünschte Verhalten:

```
$ ls testdir
123456.dat '2. Eine Buch.txt' fuer_elise.mp3
$ date
Fri Nov 23 11:32:01 CET 2018
$ ./add_timestamp.sh testdir
$ ls testdir
123456-181123-11:32.dat
'2. Eine Buch-181123-11:32.txt'
fuer_elise-181123-11:32.mp3
```

Hinweise:

- Zum Testen Deines Skripts kannst Du Dir das Verzeichnis `/group/cgl/2018/05/testdir` kopieren.
- In dieser Aufgabe kann `sed` mit *back references* helfen.

Aufgabe 5.2: Netzwerkdiagnose (4 Punkte)

Schreibe ein Skript, das als einzigen Parameter den Namen eines Rechners oder dessen IP-Adresse bekommt. Dieses Skript soll:

- **5.2.1** (1 Punkt) Zunächst überprüfen, ob ein Rechnername/eine IP-Adresse übergeben wurde. Falls nicht, soll das Skript mit einer entsprechenden Fehlermeldung abbrechen.
- **5.2.2** (2 Punkte) In einer Dauerschleife immer jeweils *ein* ping-Paket an den Zielrechner schicken und dann eine Sekunde lang warten. Leite die Standardausgabe des ping-Befehls nach “/dev/null” um. Benutze den von ping zurückgegebenen Fehlercode, um zu entscheiden ob der Zielrechner erreichbar war. Falls der Zielrechner nicht erreicht werden konnte, soll eine entsprechende Fehlermeldung ausgegeben werden und das Skript beendet werden.
- **5.2.3** (1 Punkt) Erweitere das Skript so, dass es falls der Zielrechner nicht erreicht wurde die Fehlermeldung außerdem an (D)eine im Skript vordefinierte E-Mail-Adresse verschickt. Im Betreff der Mail soll der Name des Zielrechners genannt werden (beispielsweise “Machine cip20 down”).

Hinweise:

- Um Dein Skript zu testen, kannst Du als Zielrechner giraffe (IP-Adresse 129.69.120.105) angeben. Dieser ist so konfiguriert, dass er eingehende ping-Pakete mit einer Wahrscheinlichkeit von 10% ignoriert.

Aufgabe 5.3: Würfelspiel (4 Punkte)

In Bash-Skripten können sogenannte “Funktionen” genutzt werden um Code wiederverwendbar zu machen. Als einfaches Beispiel um den Gebrauch von Funktionen zu verdeutlichen, sieh Dir folgendes Skript an:

```
#!/bin/bash
# Script name: example.sh

function complicated {
    echo In here very comlicated things happen.
    # Add them if you like ...
}

complicated
complicated
complicated
```

Ausführen dieses Skriptes führt zu folgender Ausgabe:

```
$ ./example.sh
In here very comlicated things happen.
In here very comlicated things happen.
In here very comlicated things happen.
```

In dieser Aufgabe soll ein einfaches Würfel-Ratespiel simuliert werden, bei dem der Spieler jeweils einen Tip abgibt, welche Zahl als nächstes gewürfelt wird. Gespielt wird so lange, bis die vorhergesagte und die gewürfelte Zahl übereinstimmen – damit ist das Spiel gewonnen. Schreibe ein Skript nach folgenden Vorgaben:

- **5.3.1** (1 Punkt) Schreibe zunächst eine Funktion `roll`, die mit jeweils gleicher Wahrscheinlichkeit eine Zahl zwischen eins und sechs ausgibt.
- **5.3.2** (3 Punkte) Schreibe nun die Dauerschleife, die das Spiel simuliert. Bei jedem Durchlauf soll der Spieler aufgefordert werden einen Tip abzugeben und seine Eingabe in einer Variable zwischengespeichert werden. Durch Aufruf der Funktion `roll` wird ein Würfelwurf simuliert, dessen Ergebnis mit der Eingabe verglichen wird. Sind beide gleich, wird eine Gewinn-Nachricht ausgegeben und das Skript beendet – anderenfalls beginnt die nächste Runde.

Hinweise:

- Benutzereingaben können mit `read` in eine Variable gespeichert werden.