Übungen Linux-Praktikum Blatt 9

Dr. Dirk Frettlöh (dfrettloeh@techfak.de) 19.12.2022

Aufgabe 1

Schreiben Sie eine Shell-Funktion `log_entry`, die einen gegebenen Text zusammen mit einem Zeitstempel und den Namen des Rechners ausgibt, auf dem sie aufgerufen wurde.

Die Nutzung und Ausgabe der Shell-Funktion soll wie folgt aussehen:

```
#!/bin/bash

function log_entry()
{
  # Diesen Teil müssen Sie schreiben
}

log_entry "Datenbank neu initialisiert"
sleep 3  # wartet 3 Sekunden
log_entry "Anfrage bearbeitet"
```

```
$ ./skript.sh
Nov 27 12:03:05 april: Datenbank neu initialisiert
Nov 27 12:03:08 april: Anfrage bearbeitet
```

Der Zeitstempel lässt sich mit Hilfe des `date`-Befehls erzeugen. Das genaue Ausgabeformat des Datums ist über einen Formatstring konfigurierbar; so gibt beispielsweise der folgende Formatstring den Namen und die Nummer des aktuellen Tages aus:

```
$ date "+%A der %d."
Donnerstag der 27.
```

Mehr Informationen zu dem Formatstring finden sie in der Manualpage zu `date`.
Den Rechnernamen liefert der Befehl `hostname`.

Aufgabe 2

Ein Tagesdatum sei als folgender Text gegeben: 06-12-2021

Geben Sie eine Kombination von Befehlen als Pipe an, um das Datum so an ein Array zu übergeben, dass die Werte für Tag, Monat und Jahr anschließend in den drei Array-Elementen liegen. Beispielaufruf:

`$ datum=` *Ausdruck zum Zerlegen und Zuweisen des Datums 06-12-2021*

```
$ echo ${datum[*]}
06 12 2021
```

Aufgabe 3

Gegeben sei das folgende Shellskript:

```
#!/bin/bash

while ! test -f /tmp/stop; do
   date
   sleep 1
done

echo "Angehalten!"
```

Wird das Skript aufgerufen, so gibt es jede Sekunde das aktuelle Datum und die Uhrzeit aus. Was muss man tun, damit sich das Skript mit der Meldung “Angehalten!” beendet?

Aufgabe 4

Unter der nachstehenden URL können Sie ein Archiv `dateien09.tar.gz` mit Dateien für dieses Übungsblatt herunterladen:

`https://www.math.uni-bielefeld.de/~frettloe/teach/unix/dateien09.tar.gz`

Die Datei `begriffe.txt` enthält mehrere Zeilen gegeneinander austauschbarer Satzkomponenten.

a) Schreiben Sie mit Hilfe von `while, read line` und einem Array ein Shellskript `zaehlen.sh`, das die Datei `begriffe.txt` einliest und zu jeder Zeile die Anzahl der in ihr enthaltenen Begriffe ausgibt:

```
$ ./zaehlen.sh
7: Wenige Zwei Drei Fünf Viele Schöne Wichtige
6: gelbe grüne rote blaue kleine rebellische
4: Kugeln Bälle Außerirdische Tassen
1: wurden
3: gestern erfolgreich versehentlich
5: versendet eingepackt verschenkt benutzt gesichtet
```

b) Ändern Sie Ihr Skript so ab, dass es aus jeder Zeile einen Begriff zufällig auswählt und ausgibt. Eine Zufallszahl im Bereich $0, \ldots, n-1$ kann mit dem Ausdruck `$((RANDOM%n))` erzeugt werden:

```
$ ./zufall.sh
Viele blaue Tassen wurden erfolgreich verschenkt.
$ ./zufall.sh
Wenige gelbe Bälle wurden gestern eingepackt.
$ ./zufall.sh
Schöne grüne Kugeln wurden gestern benutzt.
```

Aufgabe 5

Schreiben Sie ein Skript `addieren.sh`, das den Benutzer interaktiv beliebig viele Ganzzahlen eingeben lässt. Erfolgt als Eingabe das Gleichheitszeichen, so soll die Summe der eingegebenen Zahlen ausgegeben werden:

```
$ ./addieren.sh
Eingabe: 10
Eingabe: 20
Eingabe: 30
Eingabe: =
Summe = 60
```

Zusatzaufgabe

Das Archiv aus Aufgabe 4 enthält drei Bilddateien im `pgm`-Format, eine kleine und zwei größere. Das Format funktioniert ganz ähnlich wie das `pbm`-Format aus Aufgabe 3 von Blatt 8, nur dass es Graustufen zwischen 0 (schwarz) und 255 (weiß) gibt; statt 0 für weiß und 1 für schwarz.

- Schreiben Sie auch hierfür ein Skript, dass für eine gegebene `pgm`-Datei (z.B. `bild.pgm`) die Farben invertiert: Grauwert n wird zu Grauwert 255-n. Das invertierte Bild wird in eine Datei `i-`$\cdots$ geschrieben (im Beispiel `i-bild.pgm`).
- Schreiben Sie ein Skript `pgm2pbm`, dass eine gegebene pgm-Datei in eine pbm-Datei konvertiert. Die Grauwerte i sollen dabei bezüglich eines vorgegebenen Schwellenwerts s auf 1 (für $s < i$) bzw auf 0 (für $s \geq i$) gesetzt werden.
  Der Beispielaufruf `$ pbm2pgm bild.pgm 120` erzeugt also eine Datei `bild.pbm`, in der ein Pixel Farbe 1 hat, falls der entsprechende Pixel in `bild.pgm` Grauwert kleiner als 120 ist, und 0, falls der entsprechende Pixel in `bild.pgm` Grauwert größer oder gleich 120 hat.

Obacht: während die erste Zeile einer `pbm`-Datei so aussehen muss:

```
P1
```

muss die erste Zeile einer `pgm`-Datei so aussehen:

```
P2
```

Downloads (Folien, Übungsblätter)

`https://www.math.uni-bielefeld.de/~frettloe/teach/unix22.html`

Hinweise zu den Übungen

Die Übungen dienen dem Erlernen von Linux. Es gibt keine Abgabepflicht, es gibt überhaupt keine Abgaben. In der **A&D-Klausur** am Ende des Semesters werden allerdings Linux-Fragen vorkommen. Außerdem brauchen Sie im Verlaufe des Studiums solide Linuxkenntnisse. Daher ist es sinnvoll, dass Sie die Übungen entweder selbständig lösen, oder aber eines der Tutorien besuchen und die Übungen dort bearbeiten. Darüber hinaus können Sie in den Tutorien den Tutoren Fragen stellen zu Übungen und Vorlesung.