

Übungsblatt 4 - Betriebssysteme I

Aufgabe 1

Lesen Sie die Einführung in die bash (<http://m-jaeger.de/lv/bs1/skripten/bash/bash.pdf>) und lernen Sie den Umgang mit der bash durch interaktives Ausprobieren folgender Konzepte

- interne Kommandos
- Shell-Variablen und Umgebungsvariablen: Wertzuweisung, Einlesen, Zugriff auf den Wert
- Kontrollstrukturen: if, while, case, for
- Kommandoersetzung
- eval-Kommando

Lösen Sie nachfolgende Aufgaben (" \$" sei der Shell-Prompt).

- a) Welche Ausgabe ergibt die nachfolgende Kommandosequenz? Erläutern Sie das Zustandekommen der Ausgabe.

```
$ x="xwert"
$ export y="ywert"
$ bash
$ echo $x $y
$ export y="ywert2"
$ exit
$ echo $y
```

- b) Lesen Sie im Manual den Eintrag zum Kommando „file“. Erläutern folgendes Kommando:

```
$ ls | while read x; do file "$x"; done | grep text
```

Aufgabe 2

- a) Erläutern folgendes Kommando:

```
$ touch "$(date)" && ls -ltr | tail -1
```

- b) Erläutern folgende Kommandosequenz:

```
$ index=1
$ eval "x$index=1"
$ eval "echo \${x$index}"
```

Aufgabe 3

Schreiben Sie in C oder C++ eine Shell, die Pipelines mit 2 Prozessen ausführen kann. Der Shell-Hauptprozess soll warten, bis beide Pipeline-Teilnehmer fertig sind. Testen Sie, ob die Synchronisation bei blockiertem Leser bzw. Schreiber funktioniert.

Syntax: *Programmaufruf* | *Programmaufruf*

Beispiel:

```
ls -l /tmp | wc -l
```

Aufgabe 4

Schreiben Sie ein möglichst kurzes C-Programm, das genau die gleiche Funktion wie das Beispiel aus der vorherigen Aufgabe hat. In einem Subprozess wird „ls“ aufgerufen, das Zählen der Zeilen soll aber der Hauptprozess direkt (ohne „wc“) implementieren.