

# Übungsblatt 1

Abgabe: 05. Mai 2016

## Aufgabe 1.1: Linux-, Shell und C-Playground (0 Punkte!)

Wir fangen mit ein paar 'Übungen' zum warm werden an, um uns ein bisschen mit Linux vertraut zu machen. Achtung! Zu dieser Aufgabe gibt es nichts abzugeben und es werden somit auch keine Punkte vergeben.

### a) Get and install Linux

Zum Bearbeiten der Aufgaben benötigt ihr ein Linux – also einfach das aktuelle Windows-System plattmachen und eine Linux-Distribution runterladen und installieren. ;)

Um sich einen Überblick über verschiedene Distributionen zu verschaffen, kann man hier nachschauen: <http://distrowatch.com/dwres.php?resource=major>

Wenn man nicht weiß, welche Distribution man wählen sollte, sei Ubuntu empfohlen (oder ein Klon wie Kubuntu, Xubuntu oder Linux Mint). Letztlich ist es aber Geschmackssache.

Wer lieber bei seinem aktuellen System bleiben und/oder nicht gleich einen Dualboot anlegen will, kann eine vollständige Linux-Distribution auch einfach in einer virtuellen Maschine z.B. mittels VirtualBox installieren: <http://www.virtualbox.org/>. Es gibt bereits fertige VirtualBox-Images auf <http://virtualboxes.org/images/>, so entfällt die Installation in der VM.

Installiert auch die sogenannten GuestAdditions mit – sie erlauben z.B. die Einrichtung gemeinsamer Ordner zwischen der virtuellen Maschine und dem Gastsystem in VirtualBox.

### b) RTFM - Read The 'Friendly' Manual!!!

Wer Probleme mit den Shell- und teilweise auch den C-Aufgaben hat, kann natürlich Google oder Ähnliches zur Hilfe heranziehen – es geht aber auch professioneller. Öffnet dazu eine Shell und gebt die folgenden Befehle ein (ohne \$):

```
$ man man  
$ man grep  
...
```

**Hinweis:** Read them. Really!

## Aufgabe 1.2: Grundlagen der Betriebssysteme (1+1+1+1 = 4 Punkte)

- Bei der Von-Neumann-Architektur spricht man vom sogenannten Von-Neumann-Flaschenhals, um auszudrücken, dass das Verbindungssystem (Daten- und Befehlsbus) zum Engpass zwischen dem Prozessor und dem Speicher wird. Wieso ist dieser Von-Neumann-Flaschenhals in heutigen Rechnersystemen nur noch abgeschwächt von Bedeutung?
- Oftmals werden dem Anwendungsentwickler Funktionen in Standardbibliotheken bereitgestellt, die intern Systemaufrufe ausführen. Erläutern Sie, warum es für den Programmierer wichtig sein kann, zu wissen, ob eine verwendete Bibliotheksfunktion einen Systemaufruf durchführt oder nicht.
- Wann ist es sinnvoll, Polling einzusetzen? Wann macht der Einsatz von Interrupts mehr Sinn?
- Wie ist es einem Rechner mit nur einer CPU (und einem Prozessorkern) möglich, Prozesse scheinbar gleichzeitig auszuführen?

### Aufgabe 1.3: Bash: Grundlagen (1+1+3 = 5 Punkte)

- a) Geben Sie einen Kommandozeilen-Befehl an, mit dem in einer über Standard-Eingabe gelesenen Zeichenkette die erste (und nur die erste!) '5' durch eine '6' ersetzt wird. Beispiel (mein-befehl ist der hier zu definierende Befehl):

```
$ echo "BuS 2015: Abgabe der 1. Uebung am 5.5" | mein-befehl
BuS 2016: Abgabe der 1. Uebung am 5.5
```

- b) Erläutern Sie, was der folgende Befehl bewirkt:

```
$ cut -d ' ' -f 1 d*
```

- c) Diesmal wollen wir uns den `grep`-Befehl genauer anschauen, lese an folgendem leicht abgewandelten `xkcd`<sup>1</sup> orientiert:



Hilf Cueball und Megan, die Adresse in den E-Mails zu finden. Die E-Mails befinden sich in der Datei `emails`, die im Lernraum zur Verfügung steht. Eine Adresse ist dabei wie folgt aufgebaut:

Vorname Name  
Straße Hausnummer  
PLZ Ort

Nutzen Sie `grep`, um die gewünschten Zeilen herauszufiltern. Geben Sie als Lösung den Befehl an, mit dem die Adresse aus der Datei gefiltert wird. Geben Sie außerdem die Adresse an (Personenname, Straße und PLZ + Ort), sowie die Zeilennummern, in denen Sie diese Adresse gefunden haben. Geben Sie des weiteren Absender und Empfänger der gesuchten E-Mail an.

**Als kleiner Tipp:** Es ist vielleicht hilfreich, sich bei der Suche nach Adressen auf eine der drei Zeilen zu konzentrieren. Postleitzahlen haben hier übrigens immer 5 Ziffern, und Ortsnamen bestehen aus höchstens zwei Wörtern. Bei `grep` hilft es manchmal enorm, sich mit `--color` die eigentlichen matchenden Bereiche farbig markieren zu lassen.

<sup>1</sup><https://xkcd.com/208/>

<sup>2</sup>Title Text: Wait, forgot to escape a space. Wheeeeeee[taptaptap]jeeeeee.

#### Aufgabe 1.4: Bash systemnah (0.5+2.5+1+3 = 7 Punkte)

- a) Beschreiben Sie knapp, was ein Systemcall (Systemaufruf, Syscall) ist.
- b) Beschreiben Sie in je einem Satz, was die folgenden fünf wichtigen Syscalls tun:  
exec, stat, ioctl, mmap, brk  
*Tipp:* Die zweite Section der man-Pages beschreibt Syscalls, siehe: man man
- c) Wozu dient das Programm strace? Beschreiben Sie seine Funktion.  
*Tipp:* strace hat eine man-page! Achtung: Du bist in Frankreich? Hier ist das Programm illegal!
- d) Wir wollen strace nutzen, um das Core Util<sup>3</sup> ls zu analysieren. Nutzen Sie dabei die Option -C. Da dies viel Output erzeugt, schränken wir unsere Betrachtung außerdem noch mit -e trace=stat,lstat,fstat,open,openat auf die Syscalls stat,lstat,fstat (zwei Varianten von stat) und open bzw. die Variante openat ein. Analysieren Sie die folgenden Kommandos:
- ```
ls /etc
ls -l /etc
ls -la /etc
```
- Vergleichen Sie die Ausgabe von strace bei Anwendung auf die drei Kommandos. Was fällt Ihnen in Bezug auf die Anzahl und Art der auftretenden Systemaufrufe auf? Erklären Sie den Sachverhalt kurz.

#### Aufgabe 1.5: Bash: Textverarbeitung (0.5+0.5+2+1 = 4 Punkte)

Im Folgenden wird die Textbearbeitung mittels bash betrachtet. Beantworten Sie dazu die folgenden Fragen. Alle diese Probleme können und sollen als "Einzeiler"-Shell-Skripte durch das Kombinieren verschiedener Kommandozeilenprogramme, aber ohne Schleifen und bash-Variablen gelöst werden!

- a) Schreiben Sie ein Skript, welches zu allen Dateien (und Verzeichnissen) im aktuellen Verzeichnis die Dateigröße und den Dateinamen (und auch nur diese Angaben) ausgibt. Die Ausgabe braucht nicht schön formatiert zu sein, beide Angaben können einfach durch ein Leerzeichen getrennt ausgegeben werden.
- b) Modifizieren Sie die Ausgabe aus dem vorherigen Aufgabenteil so, dass Datum und Dateiname in umgekehrter Reihenfolge ausgegeben werden.
- c) Bei der Anmeldung zu den Übungen konnte man einen String als "Teampräferenz" festlegen. Personen mit gleichem String wurden bevorzugt der gleichen Übungsgruppe zugeordnet. Im Lernraum finden Sie die Datei teamnamen.txt, die in jeder Zeile einen Teamnamen enthält, der von einem Studierenden vergeben wurde. Schreiben Sie ein Skript, das ausgibt, wie viele Dreiergruppen sich aus den Teampräferenzen ergeben. Wie müssten Sie ihr Skript ändern, um die Anzahl der Zweiergruppen auszugeben? Wie viele Einergruppen (Teamname, der nur einmal auftaucht) gibt es? Wie viele Studierende haben keine Teampräferenz (Leerzeile) angegeben?

**Hinweis:** uniq

- d) Schreiben Sie ein Skript, das mit einem Aufruf die Anzahl der Einer-, Zweier- und Dreiergruppen, wie sie im vorherigen Aufgabenteil definiert wurden, ausgibt. Dabei ist sowohl eine Ausgabe der Form "<Gruppengröße> <Anzahl>" (oder umgekehrt) in drei Zeilen erlaubt, als auch die reine Ausgabe der Anzahlen in aufsteigender Gruppengröße.

**Hinweis:** Die Anzahl der Studierenden ohne Teampräferenz ("Nullerguppen") darf ausgegeben werden, muss aber nicht.

<sup>3</sup><http://www.gnu.org/software/coreutils/coreutils.html>