

Übung 1

Abgabe: 10.5.2018 8:30

Aufgabe 1.1: Bash: Grundlagen (0.5+0.5+1 = 2 Punkte)

- a) Geben Sie einen Kommandozeilen-Befehl an, mit dem in einer über Standard-Eingabe gelesenen Zeichenkette die erste (und nur die erste!) '6' durch eine '7' ersetzt wird. Beispiel (mein-befehl ist der hier zu definierende Befehl):

```
$ echo "BuS 2016: Abgabe der 1. Uebung am 6.5" | mein-befehl  
BuS 2017: Abgabe der 1. Uebung am 6.5
```

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

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

- c) Diesmal wollen wir uns den `grep`-Befehl genauer anschauen, lose an folgendem leicht abgewandelten `xkcd`¹ 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.

¹<https://xkcd.com/208/>

²Title Text: Wait, forgot to escape a space. Wheeeeeee[taptaptap]eeeeee.

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.

Aufgabe 1.2: Bash: Textverarbeitung (0.5+1+1 = 2.5 Punkte)

In dieser Aufgabe betrachten wir Textbearbeitung mittels `bash`. UNIX bietet hierzu eine Vielzahl kleiner Utility-Programme an, die zur Manipulation von Textfiles geeignet sind (z.B. `awk`, `grep`, `cut`, `paste`, `sort`, `uniq`, `diff`, `tr`, `wc`, `cat`, ...). Viele Aufgaben, zu denen man bei anderen Betriebssystemen komplette Programme schreiben müsste, lassen sich unter UNIX durch geschickte Verknüpfung dieser Tools über Pipes direkt auf der Kommandozeile einer Shell lösen.

Um die Aufgaben zu bearbeiten, ladet zunächst die folgende Textdatei aus 'The War of the Worlds' herunter: <http://www.openfst.org/twiki/pub/FST/FstExamples/wotw.txt>

Hinweis: `wget`

Löst die Aufgaben unter Verwendung der genannten UNIX-Tools. Gebt dabei neben der Lösung die verwendete Befehlsfolge an.

- a) Entfernt alle Sonderzeichen

" ? . ! : ; , + & `

aus `wotw.txt` und entfernt alle sich wiederholenden Leerzeichen.

- b) Wieviele Zeilen in `wotw.txt` enthalten die Zeichenfolge "road" nicht (dabei nicht nach Groß- und Kleinschreibung unterscheiden)?
- c) Welches sind die 10 häufigsten Wörter in `wotw.txt`? Gebt hierbei auch die absolute Häufigkeit dieser 10 Wörter an.

Aufgabe 1.3: Bash: Systemnah (0.5+2+1+2 = 5.5 Punkte)

- a) Beschreiben Sie knapp, was ein Systemcall (Systemaufruf, Syscall) ist.

- b) Beschreiben Sie in je einem Satz, was die folgenden vier wichtigen Syscalls tun:

`execve`, `open`, `stat`, `mmap`

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!

- d) Wir wollen `strace` nutzen, um das Core Util³ `ls` zu analysieren. Nutzen Sie dabei die Option `-C`. Da dies viel Output erzeugt, schränken wir unsere Betrachtung außerdem noch mit `-e trace=execve, stat, lstat, fstat, open, openat, getdents, readdir` auf wenige Syscalls ein. Analysieren und vergleichen Sie die folgenden Kommandos:

`ls /etc`

`ls -l /etc`

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

Analysieren sie die Ausgabe von `strace` des ersten Kommandos und vergleichen sie diese mit der Ausgabe von `strace` auf das zweite Kommando. Was fällt Ihnen in Bezug auf die Anzahl und Art der auftretenden Systemaufrufe auf? Erklären Sie den Sachverhalt kurz.

Aufgabe 1.4: Datentypen in C (0.5+0.5+0.5+0.5+0.5+0.5+0.5 = 3.5 Punkte)

Gegeben seien folgende Deklarationen:

```
int list[] = { 6, 4, 1, 7 };
char str[] = "123";
int i = 1;
int* pi;
```

Welche der folgenden Operationen sind möglich? Falls eine Operation nicht möglich ist: warum nicht? Falls sie möglich ist: was bewirkt sie? Gehen Sie davon aus, dass eine Operation das Ergebnis der darauf folgenden Operationen beeinflusst, wenn sie möglich ist.

- a) `str[1] = '0';`
- b) `i = *(list + 3);`
- c) `pi = &list[i];`
- d) `*pi = 42;`
- e) `list = pi;`
- f) `i = *(pi + 2);`
- g) `str[3]='4';`

Aufgabe 1.5: Typenkonvertierung in C (1+1+1 = 3 Punkte)

In der Vorlesung wurden verschiedenen Zahlentypen mit unterschiedlichen Größen vorgestellt. Des Weiteren wird zwischen impliziter und expliziter Typkonvertierung unterschieden. Implizite Typkonversionen stellen eine erhebliche Fehlerquelle dar. Man sollte daher die Konvertierung explizit durch sog. casting anweisen:

```
int i=3;
float f=2.5;
f = (float)i;
```

Im L2P kann die Quelltextdatei `c2f.c` heruntergeladen werden, in dem ein Programm angegeben ist, welches erlaubt Temperaturangaben von Celsius nach Fahrenheit zu konvertieren.

- a) Übersetzen Sie das Programm und testen Sie es mit unterschiedlichen Eingaben. Welches Problem besteht in dem Quelltext?
- b) Verändern Sie den Quelltext so, dass oben erkanntes Problem nicht mehr auftritt
- c) Ist es möglich das Problem ohne explizite Typkonvertierung zu lösen?

Aufgabe 1.6: Zeiger und Arrays in C (3.5 Punkte)

Betrachten Sie das folgende C-Fragment:

```
1  int a[6] = {2, 11, 23, 42, 13, 37};  
2  int *b = a; // b zeigt nun auf a[0]  
3  b++;  
4  int t1 = *b;  
5  int **c = &b;  
6  **c += 2;  
7  int t2 = **c + 1;  
8  int t3 = a[3] & (**c);  
9  b = a+5;  
10 *b = 10;
```

Welche Werte haben am Ende t1, t2 und t3? Welche Werte beinhaltet das Array a am Ende des Durchlaufs?

Begründen Sie Ihre Antwort in dem Sie zu jeder Zeile ab Nummer 3 einen kleinen Kommentar schreiben (je nach Komplexität 1-3 stichpunktartige Sätze). Für den einfachen Vorgang in Zeile 2 wäre bspw. der angegebene Kommentar ausreichend. Geben Sie zu jeder Zeile in dem sich der Inhalt des Arrays a ändert den vollständigen Inhalt von a erneut an.

Hinweis: Wir vergeben die Punkte für die Begründung, jedoch nicht für die ausschließliche Beantwortung der Frage nach dem Inhalt von t1, t2, t3 und a.