

## 4. Übung

# Algorithmen & Programmierung

J. Akhundov

J. Pönisch

M. Reißner

M. Richter

### Aufgabe 1 (zur eigenständigen Vorbereitung)

Machen sie sich mit der Kommandozeile von Linux vertraut. Sollte Ihnen kein eigener Linux-Rechner zur Verfügung stehen, können Sie die öffentlichen Computer-Pools [1] oder die Login-Server [2] des Rechenzentrums verwenden.

- Tippen sie dann das Kommando `date` ein und drücken sie die Return-Taste. Was ist die Ausgabe des Kommandos? Welche Funktionen haben die Cursor-Tasten? Was passiert am unteren Bildschirmende?
- Machen sie sich mit dem Kommando `man` vertraut. Was bewirken nach dem Start des Programms die Leertaste, die Taste 'b', die Taste 'f', die Taste 'g' und die Taste 'q' ?
- Machen sie sich mit den Kommandos `cd` und `pwd` vertraut. In welchem Verzeichnis befinden sie sich nach der Anmeldung? Was bewirkt der Aufruf `cd .`? Was bewirkt der Aufruf `cd ..`? Was bewirkt der Aufruf `cd /`? Wie können sie in das Verzeichnis `/tmp` wechseln?
- Machen sie sich mit dem Kommando `ls` vertraut. Wie kann man in der Ausgabe Dateien und Verzeichnisse unterscheiden? Wo steht die Dateigrösse? Wie kann man die Darstellung der Ausgabe verändern?
- Machen sie sich mit den Kommandos `cat` und `less` vertraut. Inspizieren sie einige Dateien im Verzeichnis `/etc`, indem sie in das Verzeichnis navigieren, die Namen der dort enthaltenen Dateien ermitteln und dann deren Inhalt ausgeben.
- Informieren Sie sich darüber, was eine Manpage ist. Machen Sie sich mit den Programmen `man`, `whatis` und `apropos` vertraut.
- Finden Sie mit Hilfe der obigen Kommandos heraus, in welche Abschnitte (Sections) die Manpages gegliedert sind. Welche(r) Abschnitt(e) werden für die Veranstaltung A&P wohl am ehesten von Interesse sein?
- Was bewirken die folgenden Kommandos: `hostname`, `arch`, `uname -a`, `uptime`, `who`, `id`, `who am i`, `cal`, `cal 9 1752`, `cal 2014`, `history`, `time sleep 5`, `which`, `top`

- In der UNIX-Kommandozeile kann '\*' als Platzhalter für eine beliebige Menge an Zeichen verwendet werden, während der Platzhalter '?' für ein einzelnes Zeichen steht. Kombinieren sie ihr bisher gesammeltes Wissen mit diesen Platzhaltern, um eine Liste aller Dateien im Verzeichnis '/etc' auszugeben, die mit dem Buchstaben 'p' beginnen.
- Erkunden sie das Verzeichnis '/proc'. Geben sie den Inhalt verschiedener Dateien aus, z.B. von 'cpuinfo' oder 'meminfo'. Woher kommen diese Informationen?
- Lernen sie das Kommando `file` kennen. Probieren sie es mit Dateien in '/bin' und in ihrem Startverzeichnis aus.
- Lernen sie das Kommando `grep` kennen. Verwenden sie auch den '\*' Platzhalter, um das Kommando auf mehrere Dateien anzuwenden.
- Lernen sie das Kommando `find` kennen. Welche Dateien (keine Verzeichnisse) befinden sich im Verzeichnis '/home'? Welche Dateien im System sind grösser als 50MB?
- Probieren sie das Kommando `set` aus. Führen sie anschließend `echo $PATH` aus. Was bewirken beide Kommandos?
- Ermitteln sie, wie mit Hilfe des Kommandos `ssh` Befehle auf anderen Computern ausgeführt werden können.

## Aufgabe 2

Implementieren Sie folgende Algorithmen als C-Funktionen. Rufen Sie ihre Funktionen mit geeigneten Testparametern in der `main()`-Funktion auf.

- Die Funktion `unsigned int sum(unsigned int n)` soll die Summe der ersten  $n$  natürlichen Zahlen berechnen.
- Die Funktion `unsigned int ggt(unsigned int a, unsigned int b)` soll den größten gemeinsamen Teiler von  $a$  und  $b$  bestimmen. Verzichten Sie dabei auf die Rekursion.
- Die Funktion `unsigned int sum2(unsigned int n)` soll die Summe der ersten  $n$  ungeraden natürlichen Zahlen berechnen.
- Die Funktion `unsigned int fak(unsigned int n)` soll die Fakultät von  $n$  berechnen.
- Die Funktion `int pow(int a, unsigned int b)` soll  $a^b$  berechnen. Sie können dabei *binäre Exponentiation* verwenden.

## Aufgabe 3

Implementieren Sie folgende Algorithmen als C-Funktionen. Rufen Sie ihre Funktionen mit geeigneten Testparametern in der `main()`-Funktion auf.

- Die Funktion `int fn1(unsigned int n)` soll  $\sum_{i=1}^n (-1)^i \cdot i$  berechnen.
- Die Funktion `unsigned int digits(unsigned int n)` soll die Anzahl der Ziffern der Dezimaldarstellung von  $n$  berechnen.

- c) Donald Knuth zahlte der ersten Person, die einen Bug in seinem Textsatzsystem `TEX` fand, `1c` als Belohnung aus. Für jeden weiteren Bug versprach er das Doppelte der Belohnung des letzten Bugs zu zahlen. Schreiben Sie eine Funktion `unsigned int bug(unsigned int n)`, die die Belohnung für den  $n$ -ten Bug zurückgibt.
- d) Schreiben Sie eine Funktion `unsigned int sumbug(unsigned int n)`, die berechnet, wieviel Geld Don Knuth bis zum  $n$ -ten Bug bereits an Belohnungen verteilt hat.
- e) Schreiben Sie eine Funktion `int polynom(unsigned int n, int x)`, die

$$x^n + x^{n-1} + \dots + x^2 + x + 1$$

berechnet.

- f) Schreiben Sie eine Funktion `int polynom2(unsigned int n, int x)`, die

$$(-1)^n \cdot x^n + (-1)^{n-1} \cdot x^{n-1} + \dots + x^2 - x + 1$$

berechnet.

#### Aufgabe 4

Übersetzen Sie das Programm aus Aufgabe 2a) mit Hilfe der Compiler-Toolchain GCC [6]. Vollziehen Sie dabei die einzelnen Übersetzungsschritte nach und machen Sie die Zwischenergebnisse sichtbar.<sup>1</sup>

Oft ist es nicht leicht Fehler im Code zu finden, weshalb sich die Verwendung eines *debuggers* empfiehlt. Machen Sie sich mit einem *debugger*, z.B. GDB [9] vertraut. Die Bedienung mag zunächst etwas gewöhnungsbedürftig erscheinen. Es gibt allerdings eine Menge Literatur über GDB. Zuallererst wäre hier das GDB-Handbuch [10] zu nennen. Eine kürzere Einführung zu GDB gibt es z.B. hier [11]. Hier sind vor allem die Kapitel 1, 2 und 4 für Sie relevant.

Sollten Sie eine GUI bevorzugen, müssen Sie eine Auswahl aus einer ganzen Reihe von Produkten treffen. Um nur ein paar zu nennen: [12, 13, 14, 15, 16]. Der GDB verfügt auch über ein eigenes Text-UI. Lesen Sie hierzu einfach Kapitel 25 im GDB-Handbuch nach, wenn Sie mehr darüber erfahren möchten.

## Literatur

- [1] <http://www.tu-chemnitz.de/urz/pools/index.html>
- [2] <http://login.tu-chemnitz.de/>, Login-Server des Rechenzentrums
- [3] <http://linux.die.net/man/>, Online-Sammlung verschiedener Linux-Manpages
- [4] <http://www.gnu.org/software/make/>, GNU Make
- [5] <http://www.ijon.de/comp/tutorials/makefile.html>, Tutorial zu Make

---

<sup>1</sup>Hier ist es ratsam, die Ausgabe des Kommandos `gcc --help` genauer unter die Lupe zu nehmen oder Abschnitt 3.2 im GCC-Handbuch [7] zu lesen. Noch ein Tipp: Für `gcc` gibt es selbstverständlich eine Manpage.

- [6] <http://gcc.gnu.org/>, Die GNU Compiler Collection
- [7] <http://gcc.gnu.org/onlinedocs/>, Dokumentation zum GCC
- [8] <http://www.mingw.org/>, Eine GNU-Umgebung für Windows
- [9] <http://www.gnu.org/software/gdb/>, Der GNU Debugger
- [10] <http://sourceware.org/gdb/current/onlinedocs/gdb/>, Dokumentation zu GCC
- [11] <http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html>
- [12] <http://www.gnu.org/software/ddd/>
- [13] <http://cgdb.sourceforge.net/>
- [14] <http://sourceware.org/insight/>
- [15] <http://www.kdbg.org/>
- [16] <http://projects.gnome.org/nemiver/>