

Übungsblatt 0: Grundlagen

Keine Abgabe.

Hinweis: Das heutige Übungsblatt ist ein Anwesenheitsübungsblatt, das bedeutet: Sie müssen Ihre Lösung nicht abgeben und bekommen keine Punkte. **Da es aber eine Einführung in wichtige Befehle und Techniken gibt, die Sie für eine erfolgreiche Teilnahme an der Vorlesung benötigen, sollten Sie es trotzdem unbedingt bearbeiten!** Es beinhaltet vier Aufgaben, die Sie in Ihrer Übungsgruppe in Kleingruppen lösen können. Bringen Sie deshalb, wenn möglich, einen Laptop zur ersten Übungsstunde mit. Selbstverständlich dürfen Sie sich die Aufgaben auch schon zuhause anschauen. Ihr Tutor wird Ihnen bei Fragen jederzeit helfen und eventuell eine kurze Einführung geben.

Aufgabe 1: Benutzung von Moodle (0P)

Die Übungszettel und die Vorlesungsnotizen werden auf der Lernplattform Moodle im Kurs *Einführung in die Praktische Informatik (IPI)* (<https://elearning2.uni-heidelberg.de/course/view.php?id=19653>) zum Download bereitgestellt. Der Einschreibeschlüssel wird in der ersten Vorlesung bekannt gegeben und kann auch beim Tutor erfragt werden.

- (a) Melden Sie sich mit dem Einschreibeschlüssel für den Kurs an.
- (b) Laden Sie sich das PDF-Dokument `befehlsuebersicht.pdf` sowie den Programmcode `aufgabe3.cc` herunter, letzteren werden Sie in Aufgabe 3 brauchen.

Auf Moodle haben wir zusätzlich eine “Wechselbörse für Übungsgruppen” eingerichtet. Falls Sie Ihre Übungsgruppe wechseln möchten und die Übungsgruppe zu Ihrem Wunschtermin bereits voll ist, können Sie hier nach einem Tauschpartner suchen (z.B. ‘Biete Montag 14 Uhr || Suche Mittwoch 14 Uhr’).

Aufgabe 2: Erste Schritte unter Unix (0P)

Eine Alternative zur Maus und grafischen Oberfläche im Umgang mit dem Computer stellt die Kommandozeile dar, welche auch als Eingabeaufforderung oder Shell bezeichnet wird. Diese wollen wir in dieser Übungsaufgabe näher kennenlernen und ausprobieren. Alle dafür benötigten Befehle sind in der Datei `befehlsuebersicht.pdf` aufgelistet.

- (a) Öffnen Sie ein Terminal¹. Sie befinden sich nun in Ihrem Home-Verzeichnis. Legen Sie hier einige Verzeichnisse und Unterverzeichnisse an und wechseln Sie zwischen diesen hin- und her. Lassen Sie sich anzeigen, in welchem Verzeichnis Sie aktuell sind.
- (b) Wechseln Sie in das unterste Verzeichnis des Verzeichnisbaums (/ genannt). Schauen Sie sich die dort vorhandenen Verzeichnisse an. Wechseln Sie zurück in Ihr Home-Verzeichnis, indem Sie `cd` ohne weitere Argumente benutzen.
- (c) Starten Sie einen Editor, z.B. `gedit`, indem Sie den Namen in die Kommandozeile eingeben. Versuchen Sie (ohne den Editor zu schließen), einen weiteren Befehl in die Kommandozeile einzugeben. Was passiert? Schließen Sie den Editor wieder. Öffnen Sie erneut einen Editor, geben Sie aber nun `&` (also beispielsweise `gedit &`) hinter dem Editornamen ein. Versuchen Sie erneut, einen weiteren Befehl in die Kommandozeile einzugeben. Legen Sie einige Textdateien in Ihrem Home-Verzeichnis an. Lassen Sie sich den Inhalt Ihres Home-Verzeichnisses anzeigen. Versuchen Sie weitere Informationen (Lese- und Schreibrechte oder Änderungsdatum) anzeigen. Der `ls`-Befehl hat viele Optionen. Benutzen Sie den `man`-Befehl, um sich die Hilfeseite zum `ls`-Befehl anzuschauen und probieren Sie weitere Optionen aus.
- (d) Legen Sie mit dem Operator `>` eine Textdatei an, welche die *manpage* (also den Text, der von `man` angezeigt wird) eines Befehls enthält. Was macht der Operator `>`?
- (e) Kopieren Sie eine Ihrer selbsterstellten Textdateien in ein anderes Verzeichnis. Verschieben Sie eine Ihrer Textdateien in ein anderes Verzeichnis.
- (f) Löschen Sie eine Datei. Löschen Sie eines der zuvor angelegten Verzeichnisse. Was passiert, wenn das Verzeichnis nicht leer war (d.h. Dateien oder Unterverzeichnisse enthält)?

¹Eine Möglichkeit dafür ist die Tastenkombination `Strg+Alt+t`. Falls Sie unter Windows arbeiten, wäre eine Möglichkeit: `Win+r`, dann `cmd` eintippen und `Enter` drücken oder Sie nutzen *Cygwin*.

Aufgabe 3: Compiler

(0P)

In dieser Übungsaufgabe sollen Sie den Compiler kennenlernen, welcher den Programmcode einer Programmiersprache in Maschinencode übersetzt. In Aufgabe 1 haben Sie die Datei `aufgabe3.cc` heruntergeladen, welche einen Programmcode enthält, der die Zahlen 151 und 37 multipliziert und dividiert und das Ergebnis ausgibt. Sie müssen jetzt noch nicht alle Details des Programms verstehen.

```
#include <iostream>
using namespace std;

int main() {
    double i;
    i = 151 * 37;
    cout << i << endl;
    i = 151 / 37;
    cout << i << endl;
    return 0;
}
```

- (a) Kompilieren Sie das Programm. Dazu wechseln Sie in der Kommandozeile in den Ordner, der das Programm enthält und geben den Befehl

```
g++ aufgabe3.cc
```

ein. Der C++-Compiler `g++` erzeugt ein ausführbares Programm `a.out`, welches Sie mit dem Befehl `./a.out` starten. Schauen Sie sich die Programmausgabe im Terminal an.

- (b) Kompilieren Sie das Programm nun erneut, indem Sie den Befehl

```
g++ -o aufgabe3 aufgabe3.cc
```

eingeben. Nun erzeugt der Compiler ein ausführbares Programm mit dem Namen `aufgabe3`, da die Compileroption `-o` den Compiler anweist, die ausführbare Datei `aufgabe3` zu nennen.

Es gibt eine große Menge weiterer Compileroptionen, die Sie nachschlagen können. Eine wichtige Option ist `-Wall`, mit der der Compiler Sie vor einer Vielzahl von Fehlerquellen in Ihrem Programmcode warnen kann. Wenn Sie Fehler in Ihrem Programmcode haben, gibt der Compiler mehr oder weniger hilfreiche Fehlermeldungen aus. Probieren Sie aus, was passiert, wenn Sie

- (c) das Semikolon in Zeile 5 löschen,
- (d) in Zeile 8 die Zahl 37 durch 0 ersetzen,
- (e) in Zeile 5 `double i` durch `double j` ersetzen und
- (f) Zeile 10 komplett löschen.

Aufgabe 4: Binärdarstellung

(0P)

Ein Bit ist die kleinste Speichereinheit in der EDV. Die im Bit gespeicherte Information kann zwei Zustände haben, An oder AUS. Das Binärsystem besteht aus zwei Zahlen (z.B. 1 und 0). Es kann also hervorragend zur Beschreibung der beiden Zustände benutzt werden und ist daher für die Informatik von Interesse.

Berechnen Sie die Dezimaldarstellung von

(a) 10101101 und **173**

(b) 110. **1101110**

Berechnen Sie die Binärdarstellung von

(c) 23 und **10111**

(d) 44. **101100**