

1 Intro: Umgebung aufsetzen

Aufgabe 1.1. (Arbeitsumgebung aufsetzen – mittel) Ihre erste Aufgabe ist, eine Arbeitsumgebung herzustellen, in der Sie C-Programme entwickeln und übersetzen (kompilieren) können.

Sie haben die Aufgabe geschafft, wenn Sie das folgende „Hallo-Welt-Programm“ übersetzen und ausführen können.

Codebeispiel 1: Hallo-Welt-Programm „hello.c“

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     puts("Hello World");
6     return (0);
7 }
```

Speichern Sie das Programm in einer Datei names `hello.c` und führen Sie dann den Befehl

```
$ cc -o hello hello.c
```

aus, um das Programm zu übersetzen. Erhalten Sie keine Ausgabe, so war der Übersetzungsvorgang erfolgreich. Führen Sie dann die erhaltene Programmdatei durch die folgende Eingabe aus, sie sollten die dargestellte Ausgabe erhalten.

```
$ ./hello
Hello World
```

Sollte dies nicht funktionieren, müssen Sie ggf. weitere Schritte zum Setup unternehmen. Diese sind abhängig vom Betriebssystem:

Linux Ggf. müssen Sie weitere Pakete installieren. Wenden Sie sich an Ihren Tutor für weitere Details

macOS Sie benötigen die Xcode-Werkzeuge, sowie für die spätere Projektarbeit die Homebrew-Paketverwaltung. Zur Installation der Xcode-Werkzeuge führen Sie den folgenden Befehl aus:

```
$ xcode-select --install
```

Zur Installation von Homebrew folgen Sie bitte der Anleitung auf der Projektseite: <https://brew.sh>.

Windows Installieren die WSL2-Umgebung. Öffnen Sie dazu die PowerShell oder die Eingabeaufforderung als Administrator und geben Sie folgende Befehle ein:

```
> wsl --install
> shutdown /r
```

Ihr System startet sich dann neu. Für den Rest des Kurses verwenden Sie bitte ausschließlich die WSL-Konsole; es ist nicht möglich, die Aufgaben dieses Kurses außerhalb von WSL zu bearbeiten. Windows wird Ihnen dafür eine Verknüpfung mit Titel „Ubuntu“ erstellen.

Anschließend müssen Sie in der WSL-Umgebung das Paket „build-essential“ installieren. Geben Sie dazu in die WSL-Konsole ein:

```
$ sudo apt install build-essential
```

weitere Betriebssysteme Wenden Sie sich bitte an Ihren Tutor.

Aufgabe 1.2. (Eine Arbeitsgruppe bilden – mittel) Suchen Sie sich zwei Kommilitonen, und bilden Sie zu dritt eine Arbeitsgruppe. Es können nach Absprache mit den Tutoren auch Zweier- oder Vierergruppen gebildet werden.

Mit Ihrer Arbeitsgruppe zusammen werden Sie im Verlauf des Kurses gemeinsam die Übungsaufgaben erledigen und später ein gemeinsames Programm für die Präsentationsprüfung entwickeln. Sie müssen sich bis Anfang der zweiten Woche entschieden haben, was für ein Projekt Sie gemeinsam entwickeln wollen.

In den folgenden Übungszetteln finden sich Aufgaben, die Sie zu verschiedenen Projektideen inspirieren sollen. Besprechen Sie Ihre Projektidee mit Ihrer Gruppe und dem Tutor.

Aufgabe 1.3. (Dokumentation – leicht) Das wichtigste Werkzeug eines jeden angehenden Programmierers ist die Systemdokumentation. Das System vor dem Sie sitzen ist mit einem umfassenden Online-Handbuch ausgestattet, das sämtliche Funktionen und Programme dokumentiert. Sie können das Handbuch mit dem Programm `man(1)` aufrufen. Tippen Sie

`man Abschnitt Seite`

um das Handbuch zu *Seite* in *Abschnitt* aufzurufen. Gibt es in allen Abschnitten des Handbuches nur eine Seite zu diesem Begriff, so können Sie *Abschnitt* weglassen. Wir schreiben im Weiteren `printf(3)` für den Begriff `printf` im Abschnitt 3.

Ziel dieser Aufgabe ist es, dass Sie sich mit dem Handbuch vertraut machen. Im weiteren Verlauf des Kurses werden wir annehmen, dass sie in der Lage sind, Einzelheiten zu Bibliotheksfunktionen selber nachzuschlagen. Folgen Sie dazu folgenden Leseaufträgen:

- Überfliegen Sie die Handbuchseite zu `man(1)` und merken Sie sich, was für Inhalte in welchen Abschnitten stehen. In welchen Abschnitten sind Befehle dokumentiert? In welchen Abschnitten Funktionen?
- Zu jedem Abschnitt hat das Handbuch eine Einführungsseite. Lesen Sie zumindest `intro(3)`, die Einführung der C-Standardbibliothek.
- Die Handbuchseite¹ `gcc(1)` dokumentiert den C-Kompiler `gcc`. Finden Sie heraus, was die Optionen `-E` und `-S` tun. Damit Sie nicht die gesamte Seite lesen müssen ist es hier sinnvoll, die Suchfunktion des Handbuchs zu verwenden. Tippen Sie `h`, um die Hilfe aufzurufen, hier wird Ihnen erklärt, wie man in Handbuchseiten sucht.
- Sollte Ihnen der Name eines Programmes oder einer Funktion entfallen sein, so können Sie mit `apropos(1)` nach Seiten mit passender Beschreibung suchen. Finden Sie so eine Funktion, die ein Zeichen in einen Großbuchstaben (engl. *upper case letter*) konvertiert.

Bitte bearbeiten Sie diese Rechercheaufträge durch Lesen des Handbuchs und zeigen Sie dem Tutor die Stellen, an denen Sie die Antworten gefunden haben. Die Antworten irgendwo im Internet zu finden, reicht nicht, um die Aufgabe zu bestehen.

¹Auf macOS lesen Sie stattdessen `clang(1)`