

Übungsserie 1

Abgabe: Dienstag 15.10.2024

Damit für alle die gleichen Regeln gelten, müssen Ihre Programme standardkonform und ohne Warnungen mit den Compileroptionen (GCC, Clang, MSVS vergleichbar): -Wall -Wextra -Werror -pedantic -std=c11 ohne Warnungen oder Fehler kompilieren. Vermeiden Sie insbesondere nicht autorisierte Bibliotheken oder Build-Environments. Reichen Sie Ihre Programme (Bitte nur die Quelltext-Dateien) als ZIP Archiv via Opal ein.

Aufgabe 1 In dieser Aufgabe richten Sie eine vollständige Entwicklungsumgebung für C auf Ihrem Rechner ein und lernen dabei, wie man typische Fehler bei der Übersetzung und Ausführung von Programmen erkennt und behebt. Dabei wird ein C-Compiler benötigt, der den C11-Standard unterstützt und POSIX-kompatibel ist.

1. Richten Sie Ihre Programmierungsumgebung ein:

- Installieren Sie einen C11-kompatiblen C-Compiler. Beispiele hierfür sind GCC (ab Version 4.9) oder Clang (ab Version 3.3). Zusätzlich können Sie auch Entwicklungsumgebungen wie Visual Studio Code (mit C/C++-Erweiterungen) oder CLion verwenden.
- Falls Sie keine Administratorrechte auf Ihrem Rechner besitzen, richten Sie eine virtuelle Maschine (VM) mit einem Ubuntu Linux-System ein, da dieses bereits POSIX-konform und gut für C-Programmierung geeignet ist.
- Falls nötig, installieren Sie auf Ihrer VM die benötigten Compiler und Entwicklungswerkzeuge, z. B. via `sudo apt-get install build-essential clang`.

2. Schreiben Sie das klassische "Hello World"-Programm in C und stellen Sie sicher, dass es korrekt kompiliert und ausgeführt werden kann.

```
#include <stdio.h>
```

```
int main(void) {
    printf("Hello, World!\n");
    return 0;
}
```

3. Fehler absichtlich einbauen: Um das Verhalten des Compilers, des Linkers und der Laufzeitumgebung zu verstehen, sollen verschiedene Fehlerarten gezielt erzeugt werden. Dokumentieren Sie die Fehlermeldungen und erklären Sie, was die jeweilige Meldung bedeutet und wie der Fehler behoben werden kann.

- Präprozessor-Fehler: Fügen Sie eine ungültige Präprozessoranweisung wie `#invalid_directive` in den Code ein und versuchen Sie das Programm zu kompilieren. Beobachten Sie die Fehlermeldung und erklären Sie, was der Fehler bedeutet.
- Compiler-Fehler: Fügen Sie einen Syntaxfehler ein, z.B. durch das Entfernen des Semikolons nach der `printf`-Zeile. Kompilieren Sie das Programm und dokumentieren Sie die Fehlermeldung.
- Linker-Fehler: Ersetzen Sie den Funktionsnamen `main` durch `meinmain` und versuchen Sie das Programm zu kompilieren und zu linkern. Beachten Sie die Fehlermeldung des Linkers.
- Laufzeitfehler (Runtime Error): Fügen Sie eine Division durch Null in das Programm ein:

```
int a = 1;
int b = 0;
int c = a / b; // Laufzeitfehler: Division durch Null
```

Führen Sie das Programm aus und dokumentieren Sie den Fehler.