

Systemprogrammierung - AIN/2

Sommersemester 2022

Einstieg in die Übungen

Als Voraussetzung für den Einstieg in die Übungen sollten Sie sich eine Linux-Arbeitsumgebung eingerichtet haben. Gehen Sie dann für den Einstieg wie folgt vor:

1. Richten Sie sich im Browser **Lesezeichen** für wichtige Webseiten ein. So haben Sie immer schnell Zugriff auf Hilfeinformation.
2. Legen Sie ein Arbeitsverzeichnis `sypr-uebungen` und darin ein Unterverzeichnis `Einstieg` an. Erstellen Sie im Unterverzeichnis eine Datei `einstieg.c` mit folgendem Inhalt:

Tippen Sie den Programmtext ab. Beim einfachen Kopieren mit der Maus lernen Sie keine C-Syntax!

```
/*
 * einstieg.c
 *
 * Das erste C-Programm gibt einen Text aus.
 *
 * Autor: ...
 * Erstellt am: ...
 */

#include <stdio.h>

int main(void)
{
    printf("Anzahl bisher geschriebener C-Programme eingeben: ");
    int anzahl;
    scanf("%d", &anzahl);

    printf("Vorname eingeben: ");
    char vorname[8];
    scanf("%s", vorname);

    printf("%ss %d. C-Programm funktioniert!\n", vorname, anzahl + 1);

    return 0;
}
```

3. Vergleichen Sie das gerade geschriebene C-Programm mit Ihrem entsprechenden Java-Programm aus dem ersten Semester.
Was ist (fast) gleich und was ist anders?

4. Übersetzen und binden Sie Ihr Programm im Terminal mit den Befehlen:

Diese und alle weiteren Befehle abtippen, nicht mit der Maus kopieren, wenn Sie etwas

lernen wollen!

```
gcc -g -fno-stack-protector -c einstieg.c
gcc einstieg.o -o einstieg
```

Bessern Sie nach, falls Fehlermeldungen kommen.

5. Versuchen Sie Ihr Programm zu starten:

```
einstieg
```

Der Kommandozeileninterpreter (die [Unix-Shell](#)) findet Ihr ausführbares Programm nicht. Wenn Sie beim Start eines Programms kein Verzeichnis angeben, sucht die Shell das Programm nur in den Verzeichnissen, die in der Umgebungsvariablen `PATH` aufgezählt sind. Die Umgebungsvariable `PATH` können Sie sich mit dem folgenden Kommando ansehen:

```
echo $PATH
```

Das aktuelle Arbeitsverzeichnis wird bei Linux mit einem Punkt angegeben und kann wie folgt in den Suchpfad aufgenommen werden:

```
export PATH=$PATH:.
```

Anschließend wird das Programm gefunden:

```
einstieg
```

6. Lassen Sie Ihr Programm unter Umgehung des Suchpfads laufen mit dem Befehl:

```
./einstieg
```

Das Programm fordert Sie auf, die Anzahl Ihrer bisher geschriebenen C-Programme sowie Ihren Vornamen einzugeben, und gibt anschließend einen Text aus.

Experimentieren Sie mit unterschiedlichen Eingaben:

- Vornamen verschiedener Länge, insbesondere sehr lange Vornamen, z.B. Friedrich-Wilhelm-August-Heinrich.
- bei der Anzahl bisher geschriebener Programme auch mal bewusst Falscheingaben, z.B. hallo statt einer Zahl, eine viel zu große Zahl oder Eingabeende `Strg-D`.

Funktioniert das Programm immer richtig?

Was liefert `scanf` als Rückgabewert? Recherchieren Sie!

Wie könnte man das Verhalten bei Falscheingaben verbessern?

Informationen zu Bibliotheksfunktionen wie `scanf` können Sie recherchieren, indem Sie in der Konsole `man 3 scanf` aufrufen.

Der Kommandoname `man` steht für Manual (engl. Handbuch) und die 3 für das Kapitel 3, in dem die C-Standardbibliothek beschrieben ist, zu der die Funktion `scanf` gehört.

Das Handbuch finden Sie auch im Internet unter <https://man7.org/linux/man->

[pages/index.html](#). Alternativ können Sie die Webseite <https://en.cppreference.com/w/c> oder notfalls Google für die Recherche verwenden.

7. Ändern Sie die Definition der Variablen `vorname` wie folgt:

```
char *vorname = (char*) malloc(8);
```

Der Speicher für den Vornamen wird nun auf dem Heap statt auf dem Stack reserviert. Vor dem Übersetzen müssen Sie noch eine weitere `#include`-Anweisung und eine `free`-Anweisung ergänzen:

```
...

#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    ...

    free(vorname);
    return 0;
}
```

Übersetzen Sie das geänderte Programm und experimentieren Sie wieder mit unterschiedlich langen Vornamen.

8. Führen Sie das geänderte Programm mit dem Speicherdebugger `valgrind` aus:

```
valgrind ./einstieg
```

Experimentieren Sie wie zu Beginn mit unterschiedlichen Eingaben.

Wie lang darf der Vorname werden, ohne dass eine Fehlermeldung kommt?

Welche Falscheingaben führen zu Fehlermeldungen?

Entfernen Sie die `free`-Anweisung aus dem Programm, und testen Sie, was `valgrind` von dieser Änderung hält.