

Hardwarenahe Programmierung

Musterlösung zu den Übungsaufgaben – 7. November 2019

Aufgabe 1: Arrays mit Zahlen

Wir betrachten das folgende Programm
(Datei: [aufgabe-1.c](#)):

```
#include <stdio.h>

void f (int *s0, int *s1)
{
    while (*s0 >= 0)
    {
        int *s = s1;
        while (*s >= 0)
            if (*s0 == *s++)
                printf ("%d_", *s0);
        s0++;
    }
    printf ("\n");
}

int main (void)
{
    int a[] = { 10, 4, 3, 7, 12, 0, 1, -1 };
    int b[] = { 7, 14, 0, 8, 9, 22, 10, -1 };
    f (a, b);
    return 0;
}
```

- (a) Was bewirkt die Funktion `f`, und wie funktioniert sie? (4 Punkte)
- (b) Was passiert, wenn Sie beim Aufruf der Funktion für einen der Parameter den Wert `NULL` übergeben? Begründen Sie Ihre Antwort. (2 Punkte)
- (c) Was kann passieren, wenn Sie das Hauptprogramm wie folgt abändern ([aufgabe-1c.c](#))? Begründen Sie Ihre Antwort.

```
int main (void)
{
    int a[] = { 10, 4, 3, 7, 12, 0, 1 };
    int b[] = { 7, 14, 0, 8, 9, 22, 10 };
    f (a, b);
    return 0;
}
```

(2 Punkte)

Lösung

- (a) Was bewirkt die Funktion `f`, und wie funktioniert sie?

Die Funktion gibt alle Zahlen aus, die sowohl im Array `s0` als auch im Array `s1` vorkommen (Schnittmenge).

Dies geschieht, indem der Zeiger `s0` das gesamte Array durchläuft (äußere Schleife). Für jedes Element des ersten Arrays durchläuft der Zeiger `s` das gesamte zweite Array (innere Schleife). Auf diese Weise wird jedes Element von `s0` mit jedem von `s1` verglichen und bei Gleichheit ausgegeben.

Um die Schleifen abbrechen zu können, enthalten beide Arrays als Ende-Markierung eine negative Zahl (`-1`).

- (b) Was passiert, wenn Sie beim Aufruf der Funktion für einen der Parameter den Wert `NULL` übergeben? Begründen Sie Ihre Antwort.

In dem Moment, wo auf den jeweiligen Parameter-Zeiger zugegriffen wird (`while (*s0 >= 0)` für `s0` bzw. `int *s = s1; while (*s >= 0)` für `s1`), kommt es zu einem Absturz (Speicherzugriffsfehler). Die Dereferenzierung eines Zeigers mit dem Wert `NULL` ist nicht zulässig.

- (c) Was kann passieren, wenn Sie das Hauptprogramm wie folgt abändern ([aufgabe-1c.c](#))? Begründen Sie Ihre Antwort.

```
int main (void)
{
    int a[] = { 10, 4, 3, 7, 12, 0, 1 };
    int b[] = { 7, 14, 0, 8, 9, 22, 10 };
    f (a, b);
    return 0;
}
```

Durch die fehlenden Ende-Markierungen der Arrays laufen die Schleifen immer weiter, bis sie irgendwann zufällig auf Speicherzellen stoßen, die sich als Ende-Markierungen interpretieren lassen (negative Zahlen). Dadurch kann es zu einem Lesezugriff auf Speicher kommen, für den das Programm kein Lesezugriffsrecht hat, also zu einem Absturz (Speicherzugriffsfehler).

Aufgabe 2: Datum-Bibliothek

Schreiben Sie eine Bibliothek (= Sammlung von Deklarationen und Funktionen) zur Behandlung von Datumsangaben.

Diese soll enthalten:

- einen **struct**-Datentyp **date**, der eine Datumsangabe speichert,
- eine Funktion **void date_print (date *d)**, die ein Datum ausgibt,
- eine Funktion **int date_set (date *d, int day, int month, int year)**, die ein Datum auf einen gegebenen Tag setzt und zurückgibt, ob es sich um ein gültiges Datum handelt (0 = nein, 1 = ja),
- eine Funktion **void date_next (date *d)**, die ein Datum auf den nächsten Tag vorrückt.

Schreiben Sie auch ein Programm, das die o. a. Funktionen testet.

Lösung

Die Datei [loesung-2.c](#) enthält die Bibliothek zusammen mit einem Test-Programm.

Eine detaillierte Anleitung, wie man auf die Funktion [date_next\(\)](#) kommt, finden Sie im Skript zur Lehrveranstaltung, Datei [hp-2019ws.pdf](#), ab Seite 29.

(Die Vorgehensweise, die Bibliothek und das Hauptprogramm in dieselbe Datei zu schreiben, hat den Nachteil, daß man die Bibliothek in jedes weitere Programm, das sie benutzt, kopieren und auch dort aktuell halten muß. Eine sinnvollere Lösung wird demnächst in der Vorlesung vorgestellt werden.)