

Übungsblatt 8

Einführung in C

Abgabe bis 13. Juni 2023 um 12:00 Uhr im Moodle

Cybersecurity im Sommersemester 2023

Thomas Eisenbarth, Jan Wichelmann, Anja Köhl

Einführung

Erinnern Sie sich noch an das neue Kontextwechsel-Abrechnungssystem? Abgesehen von einigen vernachlässigbaren technischen Details (Machbarkeit, Kundenbindung) verspricht dieses einen enormen finanziellen Vorteil gegenüber der Konkurrenz, weshalb das Management um jeden Preis umsteigen möchte. Während sich die Rechtsabteilung bereits an die Formulierung entsprechend verklausulierter Knebelverträge macht, wird die Entwicklungsabteilung mit der Programmierung des ganzen beauftragt.

Leider hat das Management vergessen, dass der Aufgabenbereich eben jener Abteilung eigentlich eher im User-/Frontend-Bereich angesiedelt ist, und nicht in der Modifikation von Betriebssystem-Routinen. Der Vorschlag eines Managers, doch einfach ein eigenes Betriebssystem in einer der bekannten Sprachen (Python, PHP, Javascript) zu implementieren, schafft es nicht durch das Teammeeting; stattdessen soll externe Hilfe zur Schulung in einer hardwarenäheren Programmiersprache hinzugezogen werden, kurz: Sie.

Aufgabe 1 C-Programm verstehen (10 Punkte)

Ein hardwareaffiner Praktikant hat bereits ein C-Buch ausgegraben und eine kurze Einführung in die wichtigsten Sprachkonstrukte gegeben. Im Anschluss wollte er eigentlich einen kleinen Beispielcode erklären, ist dort aber dann auf halbem Wege gescheitert. Helfen Sie ihm weiter?

Das Programm:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  static void copy_reverse(int *source, int *destination, int length)
5  {
6      int *srcPtr = &source[length];
7      int *destPtr = destination;
8      while(srcPtr > source)
9          *destPtr++ = *--srcPtr;
10 }
11
12 int main(int argc, char **argv)
13 {
14     printf("How many numbers? ");
15     int count;
16     scanf("%d", &count);
17
18     int *input = malloc(count * sizeof(int));
19     int *output = malloc(count * sizeof(int));
```

```

20
21     printf("Type %d numbers: \n", count);
22     for(int i = 0; i < count; ++i)
23         scanf("%d", &input[i]);
24
25     copy_reverse(input, output, count);
26
27     printf("Reversed: ");
28     for(int i = 0; i < count; ++i)
29         printf("%d ", output[i]);
30     printf("\n");
31
32     free(input);
33     free(output);
34
35     return 0;
36 }

```

Sie können den Quellcode auch im Moodle finden, falls Sie das Programm selbst ausführen möchten.

Erklären Sie für die folgenden Zeilennummern jeweils kurz, was in der jeweiligen Zeile passiert:

1. Zeilen 1/2
2. Zeile 16
3. Zeilen 18/19
4. Zeile 23
5. Zeile 25
6. Zeilen 6/7/8/9
7. Zeile 29
8. Zeilen 32/33
9. Zeile 35

Beantworten Sie außerdem die folgende Frage des Praktikanten:

10. Der Übersicht halber wäre es besser, die `main`-Funktion am Anfang des Programmes zu haben. Leider gibt es dann einen Compilerfehler, wenn die `copy_reverse`-Funktion einfach nach unten kopiert wird. Warum entsteht dieser Fehler, und wie lässt sich das Problem lösen?

Aufgabe 2 C-Programm schreiben (10 Punkte)

Als kleine Übung soll ein lang ignoriertes Problem gelöst werden: Über die Jahre haben immer wieder Kunden Server angemietet, wieder abgegeben, oder einen neuen Wartungstechniker zugewiesen bekommen. Hierüber wird natürlich streng Buch geführt, was jedoch zu dem Problem führte, dass die Liste nun völlig fragmentiert ist. Die Buchhaltung hätte nun gern eine nach Kunden, Technikern und IP-Adressen sortierte Liste. Das einfach mithilfe einer beliebigen Tabellenkalkulations-Software zu erledigen ist natürlich völlig abwegig. Die Implementierungsabteilung hat sich hier schon ins Zeug gelegt, hätte aber gern eine Musterlösung von Ihnen, um ihre eigene Lösung damit abgleichen zu können.

Das Eingabeformat sieht hierbei wie folgt aus:

- Eine Ganzzahl $0 < n \leq 10000$, die die Anzahl der Datensätze angibt
- n Zeilen mit jeweils vier durch Leerzeichen getrennten Einträgen:
 - Server ID (Ganzzahl, steigt mit jeder Zeile um 1 an)
 - Kundenname (ein Wort, max. 31 Zeichen)
 - IP-Adresse ($x.x.x.x$, max. 15 Zeichen), entspricht der Server ID plus einem Offset
 - Technikername (ein Wort, max. 31 Zeichen)

Das Ausgabeformat ist identisch, bis auf einer kleinen Änderung: Die Einträge sollen nun nicht nach ID, sondern absteigend nach den folgenden Kriterien geordnet sein:

- Kundenname
- Technikernamen
- IP-Adresse bzw. ID

Bearbeiten Sie die folgenden Schritte:

1. Laden Sie sich aus dem Moodle das Grundgerüst herunter.
2. Definieren Sie die Struktur `row_t`, sodass sie einen Datensatz bestehend aus Server ID, Kundennamen, IP-Adresse und Technikernamen aufnehmen kann.
3. Lesen Sie mithilfe von `scanf` die Anzahl der Zeilen aus der Standardeingabe und allokieren Sie ein Array `rows`, das entsprechend viele Einträge vom Typ `row_t` aufnehmen kann.
4. Lesen Sie nun die einzelnen Zeilen mit `scanf` in `rows` ein.
5. Den Sortieralgorithmus müssen Sie nicht selbst implementieren; C bietet hierfür die Standardfunktion `qsort`. Diese erwartet das Eingabearray, die Anzahl der Einträge, die Größe eines einzelnen Eintrags, und eine Vergleichsfunktion. Ergänzen Sie die fehlenden Parameter.
6. Die Vergleichsfunktion ist bereits teilweise vorgegeben; es fehlt nur noch der Vergleich an sich. Die Funktion sollte eine (beliebige) Zahl < 0 zurückgeben, wenn a kleiner als b ist; wenn a größer als b ist, sollte diese eine (beliebige) Zahl > 0 zurückgeben. Falls a und b identisch sind, sollte die Funktion 0 zurückgeben. Für Strings gibt es bereits eine Vergleichsfunktion `strcmp`, die sich derartig verhält; lesen Sie die Dokumentation zu dieser Funktion und nutzen Sie sie, um Kundennamen und Technikernamen nacheinander zu vergleichen. Fügen Sie einen Vergleich der Server-IDs hinzu, dessen Ergebnis nur dann zurückgegeben wird, wenn sowohl Kundennamen als auch Technikernamen identisch sind.
7. Geben Sie das sortierte Array im korrekten Format aus.
8. Geben Sie das vormals allokierte Array zum Schluss wieder frei.
9. Testen Sie Ihr Programm mit den beiliegenden Testfällen.

Laden Sie Ihr funktionierendes Programm im Anschluss in der entsprechenden Moodle-Aktivität hoch und evaluieren Sie es erneut mit den dort hinterlegten Testfällen.