24. November 2023

FAQ zur Programmierung in C für den MRT1 Versuch 3 (Animation)

Inhaltsverzeichnis

1 Wie kann ich sicher sein das meine Programmierumgebung korrekt funktioniert?

Kompilieren und Starten Sie den für den Versuch vorgegebenen Quellcode unverändert. Dieser ist fehlerfrei kompilierbar und das daraus entstandene Programm zeigt ein Testbild und eine Dialogbox.

2 Wie soll ich mit anderen Studenten zusammen am gleichen Quellcode arbeiten?

Nutzen Sie ein Quellcode-Verwaltungssystem. Im letzten Jahrzehnt hat sich das Werkzeug `git` etabliert. `Magit` ist ein komfortabler Ersatz für die Bedienung von `git` aus der Kommandozeile.

Durch Nutzung eines Hosters von Git-Repositories, ist es möglich Daten mittels `git` über unterschiedliche Rechner auszutauschen und im Team zu arbeiten.

Empfohlener Git-Repository-Hoster: `https://www.tu-chemnitz.de/urz/storage/gitlab/`

Andere Anbieter sind beispielsweise `Bitbucket`, `GitHub` und `GitLab` aber Achtung lesen Sie vor deren Nutzung die Geschäftsbedingungen.

3 Wo erfahre ich wie man eine Funktion aus der Standard-Library benutzt?

3.1 Dokumentation in Form von Manpages

Manpages sind die Standarddokumentation für Unix/Linux-Systeme und in der Regel überall installiert oder einfach nachinstallierbar.

Manpages sind durch ihren strukturierten Aufbau sehr hilfreich für die schnelle Recherche.

Beispiel: Hilfe zur Funktion `geltine`.
Geben Sie auf der Kommandozeile folgendes ein:

```
man -a getline
```

Es öffnet sich eine Dokumentation, welche Sie mit der Taste `q` beenden können. Manpages sind meist im gleichen Stil aufgebaut:

- `Synopsis` zeigt die Parameter und deren Datentyp, Hier findet man auch die Information, welche Header-Dateien in den eigenen Quellcode eingebunden (`#include`) werden müssen.
- `Description` gibt eine Funktionsbeschreibung
- `Return Value` erklärt welchen Wert und Datentyp die Funktion als Ergebnis liefert
- `Errors` gibt Hinweise wie welche möglicherweise auftretenden Fehler identifiziert werden können
- `Example` nicht immer vorhanden, aber wenn dann gibt es ein hier Quellcode als Anwendungs-Beispiel.

Dokumentation zur Bedienung von `man`:

```
man man
```

3.2 weitere Dokumentationen

Suchen Sie in der entsprechenden Kategorie mit dem Programm `zeal`.

4 Wie kann ich meinen Quellcode/Funktion/Modul testen ...

4.1 mein Quellcode hängt von der Ausgabe eines anderen Programmteils ab, der noch nicht fertig ist

Funktionen eines Moduls können leicht getestet werden indem Ergebnisse aus anderen Modulen, welche als Vorarbeit benötigt werden, einmal händisch berechnet und als konstante Eingaben in den Quelltext geschrieben werden. Vergessen Sie nicht die vorberechneten konstanten Eingaben beim Zusammenführen der Programm-Module zu entfernen oder auszukommentieren.

4.2 er ist viel zu kompliziert und lang

Wenn Sie unsicher sind, wie sich ein Teil ihres Programms verhält, dann schreiben Sie ein kleines separates Testprogramm in C, welches nur diese eine Sache ausführt und die Ergebnisse auf die Konsole ausgibt. So können Sie ihren Quellcode testen und verbessern. Der so entstandene Quellcode kann anschließend ins „große“ Programm übernommen werden.

Schreiben Sie kleine, kurze Funktionen mit aussagekräftigen Funktions- und Variablennamen. Dies senkt die Komplexität und hilft bei der Fehlersuche. (Stichwort: `Teile und Herrsche, Divide and Conquer`)

5 Wie bediene ich einen Debugger?

Häufig sind grafische Debugger-Tools nur Frontends für den `gdb`. Wenn man verstanden hat wie man den `gdb` in der Konsole/Terminal bedient, erschließt sich oft die Benutzung des jeweiligen Debugger-Tools.

Beej's Quick Guide to GDB

6 Kann ich auch ohne Debugger arbeiten?

Ja! Indem Sie den Quellcode lesen und verstehen. Dabei hilft eine Technik die häufig als `printf-Debugging` oder `Logging` bezeichnet wird. Fügen Sie `printf()` in Ihren Quellcode ein um Zwischenergebnisse auf der Konsole/Terminal auszugeben.

7 Das Fenster öffnet sich bei Programmstart nicht

Siehe mein Programm stürzt ab.

8 Das Fenster öffnet sich nur ganz kurz und schließt sich gleich wieder

Siehe mein Programm stürzt ab.

9 Mein Programm stürzt ab, warum?

Mit großer Wahrscheinlichkeit schreibt ihr Programm an eine Speicheradresse, deren Speicher schon durch andere Programmteile genutzt wird.

Das kann passieren, wenn eine Schleife zu spät terminiert, oder Sie nicht genügend Speicher – für den Speicherbereich in welchen kopiert – wird alloziiert haben. Das Programm stürzt jedoch nicht unbedingt an der Stelle ab, an der der Fehler im Programm auftritt, sondern erst später und oft auch an unterschiedlichen Stellen.

Das Programm kann auch abstürzen, wenn die Konfigurationsdatei nicht geöffnet werden konnte und ihr Programm trotzdem versucht weiterzuarbeiten. Fangen Sie solche Fehler in Ihrem Programm ab.

9.1 Möglichkeiten zur Fehlerbehebung

9.1.1 Vorarbeiten

korrigieren Sie Ihren Quelltext so, dass beim Kompilieren keine `Warnings` mehr auftreten.

9.1.2 Variante 1

Bauen Sie `printf()` Aufrufe in Ihr Programm ein. Lassen Sie Ihr Programm damit laufen. Wenn das Programm abstürzt wurde der fehlerhafte Quellcode zwischen Programmstart(!) und dem nachfolgenden `printf()` Aufruf ausgeführt.

Suchen Sie nach fehlerhaften Quellcode der oben erklärter Ursache für das Problem entspricht.

9.1.3 Variante 2

Nutzen Sie eine Software Bibliothek oder Programm, welche diese Art Speicherprobleme detektieren kann. z.B.:

- `valgrind` : Valgrind Quick Start Guide
- `electric-fence`
- `libdmalloc`
- `duma`

9.1.4 Variante 3

Nutzen Sie den Debugger, ßteppen"damit durch Ihr Programm und verifizieren Sie die Ergebnisse einzelner Funktionsaufrufe.

10 Bei Dateiende wird die vorherige Zeile/vorherigen Zeichen nochmal eingelesen

Das ist nicht der Fall. In den Ziel-Puffer wurde nichts geschrieben und dort steht immer noch das Ergebnis vom letzten erfolgreichen Auslesen.
Detektieren Sie das Ende der Datei korrekt.
Die Manpages zu `getline`, `gets`, `fscanf` usw. helfen beim Verständnis der Standard-Funktionen.

11 Meine Frage wurde nicht beantwortet.

Schreiben Sie ihre Frage per E-Mail an den Versuchsverantwortlichen für den MRT1 C-Versuch (Animation).
Die Kontaktdaten finden Sie auf der Webseite zum Praktikumsversuch.