
Semesteraufgabe Informatik 1

Abgabe spätestens bis: 10.02.2021

Allgemeine Aufgabenstellung

Sie können als freiwillige Zusatzaufgabe zuhause eine der unteren Systembeschreibungen implementieren. Die Aufgabe wird durch die in den Übungsgruppen vorhandenen Teams bearbeitet¹. Die Bewertung erfolgt durch eine Abnahme. Die Teams mit dem besten Programm zu einer Aufgabe erhalten eine Urkunde. Eine Teilnahme mit gutem Erfolg ist empfohlen, wenn man Tutor für Informatik 1 werden möchte.

Abgabe

Die Abgabe muss in folgender Form erfolgen:

- Upload der Materialien in den Hausaufgaben-Ordner der Digicampus-Veranstaltung **Informatik 1**.
- Die Upload-Materialien müssen enthalten:
 - Quellcode.
 - Hinweise zur Ausführung und Benutzung.
 - Auflistung der Personen, die an der Aufgabe gearbeitet haben.
 - Angabe von zeitlichen Überschneidungen am Tag der Abnahme (17.02.).

Allgemeine Hinweise

- Jedes Team bewertet die Lösung eines anderen Teams zu einer anderen Aufgabe:
 - Die Lösung des anderen Teams wird vom Lehrstuhl per Mail am 11.02.2021 zugeschickt.
 - Die Bewertung muss bis 15.02.2021 in den Hausaufgaben-Ordner der Digicampus-Veranstaltung **Informatik 1** hochgeladen werden (als normale Textdatei oder PDF).
- Abnahme am 17.02.2021:
 - Bei der Abnahme müssen alle Teammitglieder anwesend sein.
 - Die Teammitglieder führen ihr Programm vor und beantworten Fragen zur Funktionalität und zur Programmierung.
 - Die Abnahme erfolgt digital über Zoom.
- In der Vorlesung wurden noch nicht alle notwendigen Anforderungen besprochen. Sie sollten bereits jetzt mit der Programmierung der bereits behandelten oder aus dem Vorkurs bekannten Konzepte beginnen, und diese dann nach und nach um die weiteren Anforderungen ergänzen und verbessern, sobald diese in der Vorlesung besprochen wurden.

¹Eine Zusammenarbeit mit Personen außerhalb des eigenen Teams ist erlaubt, wenn die eigenen Teammitglieder kein Interesse an der Bearbeitung der Semesteraufgabe haben.

Technische Mindestanforderungen

- Das Programm muss in der Programmiersprache C erstellt worden sein, muss sich ohne Fehler und Warnungen mit den Compilerschaltern `-ansi -pedantic -Wall -Wextra` compilieren lassen, und muss fehlerfrei ausführbar sein.
- Es dürfen bei der Implementierung keine anderen Bibliotheken als die C89-Standardbibliothek benutzt werden.
- Der Code muss sich im Wesentlichen an den Vorgaben der Vorlesung orientieren, zum Beispiel:
 - Programmier-Konventionen.
 - Ausführliche Fehlerbehandlung von Funktionsaufrufen.
 - Sinnvolle Aufteilung des Quellcodes auf verschiedene Übersetzungseinheiten.
 - Sinnvolle Verwendung von lokalen, statischen und globalen Variablen.
 - Sinnvolle Verwendung von symbolischen Konstanten und Makros.
 - Vermeidung von Speicherlecks.
- Vorgaben für die textuelle Benutzerschnittstelle:
 - Ausführliche Benutzerführung (Informationen über die möglichen Eingaben, verständliche Status- und Fehlermeldungen).
 - Übersichtlich formatierte Ausgaben.
- Sie müssen alle Implementierungsentscheidungen begründen können.
- Die Anforderungen der spezifischen Aufgabenstellung müssen erfüllt sein.
- Der Code muss eigenständig erstellt worden sein.

Bewertung

- Die Bewertung erfolgt durch den Lehrstuhl auf Grundlage der Abnahme.
- Die Abnahme besteht aus einer 10-minütigen Präsentation des Programms (und zwar der Anwendung, nicht des Quellcodes) durch alle Teammitglieder, mit anschließender 5-minütiger Diskussion. Jedes Teammitglied muss einen Teil der Präsentation übernehmen.

Systembeschreibung A: Zahlencodierungen

Es soll eine Anwendung für Zahlensysteme mit beliebiger Basis zwischen 2 und 16 (jeweils einschließlich) und Zahlencodierungen (1-Komplement, 2-Komplement) erstellt werden, die mindestens folgende Funktionalitäten zur Verfügung stellt:

- Umrechnungen von positiven ganzen Zahlen und reellen Zahlen zwischen dem Dezimalsystem und anderen Zahlensystemen mit und ohne Darstellung der zugehörigen Rechnung
- Addition und Subtraktion von positiven ganzen Zahlen und reellen Zahlen in den verschiedenen Zahlensystemen mit und ohne Darstellung der zugehörigen Rechnung
- Codierung und Decodierung von ganzen Zahlen bzgl. der 1-Komplement- und 2-Komplement-Codierung mit und ohne Darstellung der zugehörigen Rechnung für folgende Anzahlen n von Bits:
 $n \in \{8, 16, 32\}$
- Addition und Subtraktion von ganzen Zahlen in der 1-Komplement- und 2-Komplement-Codierung mit und ohne Darstellung der zugehörigen Rechnung für folgende Anzahlen n von Bits:
 $n \in \{8, 16, 32\}$

Freiwillige Zusatzaufgabe

Interaktive Schritt-für-Schritt-Ausführung von Rechnungen.

Systembeschreibung B: Conway's Game of Life

Es soll das Spiel **Conway's Game of Life**² in C programmiert werden. Dabei sollen folgende Aspekte berücksichtigt werden:

- Die Anfangsgeneration kann entweder randomisiert erzeugt oder vom Benutzer definiert werden.
- Es kann eine beliebige Anzahl von Folgegenerationen simuliert werden.
- Das Spielfeld ist in Zeilen und Spalten unterteilt. Für den Rand des Spielfeldes kann zwischen unterschiedlichen Varianten gewählt werden (z.B. ist ein Torus-förmiges Spielbrett denkbar).
- Das Spielfeld wird auf der Kommandozeile angezeigt, überlegen Sie sich eine passende Ausgabe.
- Implementieren Sie verschiedene *interessante* Anfangskonfigurationen, die bei Spielstart ausgewählt werden können.

Freiwillige Zusatzaufgabe

Implementieren Sie einen oder mehrere der abweichenden Regelsätze und überlegen Sie sich interessante Anfangskonfigurationen.

Systembeschreibung C: Simulator für Maschinenprogramme

Wir hatten im Kapitel 2 unsere eigene Maschinensprache eingeführt. Entwickeln Sie ein C-Programm, welches die Ausführung von Programmen, die in dieser Maschinensprache geschrieben wurden, simuliert. Dabei sollen folgende Aspekte berücksichtigt werden:

- Es soll sichtbar sein, welcher Befehl gerade ausgeführt wird und welche Werte gerade in den Registern hinterlegt sind.
- Wenn das Maschinenprogramm Parameter (z.B. in `S1`) erwartet, soll der Benutzer diese eingeben können.
- Am Ende soll die Anzahl der benötigten Takte ausgegeben werden.

Freiwillige Zusatzaufgabe

Der Simulator gibt eine Beschreibung aus, welche Komponenten des von-Neumann-Rechners bei einem Befehl verwendet werden (ähnlich Übungsaufgabe 8) oder zeigt an, wie die Befehle den Fetch-/Decode-/Execute-/WriteBack-Zyklus durchlaufen (ähnlich Übungsaufgabe 7).

²https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens