Freie Universität Berlin

# Übung 01

Rechnerarchitektur
Wintersemester 20/21
Prof. Dr.-Ing. Jochen Schiller

Abgabe 20.11.2019, 10:15

**Allgemeine Hinweise** Lesen Sie bitte folgende Hinweise aufmerksam durch. Sie gelten für alle Übungszettel.

- Melden Sie sich bitte im KVV unter **Section Info** zu einer Übungsgruppe an.
- Abgaben sind bis zum Ende der Vorlesung jeweils Freitags (10:15 Uhr) möglich.
- Bitte geben Sie zu jedem einzureichenden Zettel Ihre Beantwortung als PDF sowie den Quellcode kommentiert und unkompromiert im KVV ab. Die Abgabe in Papierform im Fach des Tutors ist optional.
- Beantworten Sie alle Aufgaben so verständlich wie möglich mit Ihren eigenen Worten. Abgaben sind in Englisch und Deutsch möglich.
- Falls Sie Quellen jenseits der Vorlesungsfolien verwenden, geben Sie diese an.
- Programmieraufgaben nutzen als Referenzsystem die Linux-Pools, werden also dort zum Testen kompiliert und ausgeführt.
- Zum Bestehen eines Zettels muss in jeder Aufgabe mindestens 1 Punkt erreicht werden, *und* mindestens 50 Prozent der Punkte auf dem gesamten Zettel
- Die Punkte sind wie folgt pro Aufgabe definiert:
  - 3 Punkte: Alles perfekt, die Klausur kann kommen.
  - 2 Punkte: Es funktioniert / ist im Wesentlichen korrekt, kleinere Mängel sind mit Kommentaren versehen.
  - 1 Punkt: Es funktioniert nicht, aber richtige Idee mit Fehlerbeschreibung (!) vorhanden oder Abgabe enthält grobe Fehler bzw. ist unzureichend beschrieben – jedoch erkennbarer Aufwand.
  - 0 Punkte: Aufgabe oder unabhängige Teilaufgabe nicht bearbeitet bzw. kein Arbeitsaufwand erkennbar.

    *Damit Zettel leider nicht bestanden.*

# Aufgabe 1: Das Von-Neumann-Rechnermodell

1946 wurde das von-Neumann-Rechnermodell vorgestellt, das die Rechnerarchitektur bis heute maßgeblich beeinflusst. Arbeiten Sie die grundlegenden Organisationsprinzipien und Besonderheiten dieses Modells heraus, indem Sie folgende Fragen möglichst prägnant und in eigenen Worten beantworten.

- Mit dem von-Neumann-Rechnermodell wurde erstmalig das Konzept für einen echten „general-purpose Computer“ vorgeschlagen. Was ist darunter zu verstehen?
- „Programme sind auch nur Daten“ ist eine grundlegende und eng mit dem von-Neumann-Rechnermodell verbundene Sichtweise. Was ist darunter zu verstehen?
- Das von-Neumann-Rechnermodell setzt sich aus vier Hauptbestandteilen zusammen. Welche Bestandteile sind dies und welchem Zweck dienen sie?
- Im von-Neumann-Rechnermodell gibt es einen Daten- und einen Befehlsprozessor als Bestandteil der CPU. Welche Aufgaben werden von welchen Komponenten dieser Prozessoren erfüllt?
- Das von-Neumann-Rechnermodell unterscheidet zwischen Daten- und Adressbus. Warum macht das Sinn?
- Die Arbeitsweise eines von-Neumann-Rechners wird durch die Bezeichnung SISD allgemein charakterisiert. Welches Prinzip verbirgt sich hinter dieser Abkürzung? Was für andere Kategorien gibt es nach Flynn?
- Bahnbrechend neu am von-Neumann-Rechnermodell war das Konzept einer quasi universellen Programmierbarkeit. Erörtern Sie in diesem Zusammenhang die Begriffe Maschinencode, Assemblersprachen sowie Ein- und Mehr-Adress-Befehle.
- Charakteristisch für das von-Neumann-Rechnermodell ist ein Zwei-Phasen-Konzept der Befehlsverarbeitung. Welches Problem wird damit auf welche Weise gelöst?
- Die Architektur eines klassischen von-Neumann-Rechners führte schon bald zu einem gewichtigen Problem, dem von-Neumannschen „Flaschenhals“. Was ist darunter zu verstehen und wie versuchte man später dieses Problem zunächst zu umgehen?

# Aufgabe 2: Gausssume

Schreiben Sie eine NASM-Funktion, welche die geschlossene Form der Gausssumme implementiert:

$$\frac{n(n+1)}{2} \quad \text{bzw.} \quad \frac{n^2+n}{2}$$

Machen Sie sich dazu mit Arithmetikbefehlen (ADD, SUB, MUL, DIV, IDIV, IMUL, NEG) in NASM vertraut. Wie genau funktionieren diese Befehle und warum gibt es zwei Multiplikations- und zwei Divisionsbefehle?

Die von Ihnen zu implementierende Funktion soll folgende Signatur haben:

```
uint64_t gauss(uint64_t n);
```

Machen Sie sich dafür mit Funktionsaufrufen auf Assemblerebene vertraut.

- Wo stehen die übergebenen Parameter?
- Wo muss der Rückgabewert hingeschrieben werden (Stichwort: Calling Convention)?
- Ein geeigneter C-Wrapper zum Ausführen der Funktion wird Ihnen im KVV gestellt. Warum ist ein solcher Wrapper nötig?

*Hinweis zu den C-Wrappern: Programmieren in C ist nicht Teil dieses Kurses, deswegen werden Ihnen die nötigen C-Programme / Programmteile für diesen Kurs gestellt. Im zweiten Teil dieses Moduls (Betriebs- und Kommunikationssysteme) müssen Sie selbständig in C programmieren. Es kann deswegen nicht schaden sich mit den C-Wrappern auseinander zu setzten bzw. auch mal einen selbst zu schreiben.*

*Hinweis: Um das Eintippen der Compile-Befehle zu vereinfachen sind Makefiles hilfreich.*