



Institut für Angewandte Informationsverarbeitung

Dr. Andreas F. Borchert
Stefan Lenz

22. November 2012
Blatt 5

Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 29. November. 2012, 16:00 Uhr

Lernziele:

- Structs und Enumerations
- Ein Blick auf eine Assemblersprache

Aufgabe 9: Redcode-Maschine (10 Punkte)

In dieser Aufgabe sollt Ihr eine virtuelle Maschine programmieren, die die künstliche Maschinensprache Redcode ausführt und in einem Wettstreit zwei Programme gegeneinander antreten lässt.

Ein Blatt mit allen Befehlen der *hier zu verwendenden* Version von Redcode findet Ihr auf der Vorlesungshomepage. Ebenso könnt Ihr dort einige bereits fertige Redcode-Programme sehen. Den Assembler-Text der Programme kann man sich in den Dateien mit Endung `.rc` anschauen. Das entsprechende Programm in „Opcode“ (abkürzend für operation code), also in für die geforderte Maschine verständlicher Sprache, ist in den Dateien mit Endung `.o` zu finden.

Die Maschine besteht aus einem virtuellen Prozessor und einem angeschlossenen Speicher.

Der Speicher hat folgende Eigenschaften:

- Er ist ringförmig, d.h. alle Adressen sind gültig und werden modulo Größe des Speichers genommen.
- Alle Elemente des Speichers sind Instruktionen. Daten werden ebenfalls durch eine Instruktion repräsentiert: Die Instruktion `DAT 0, #n` steht für die Ganzzahl n .
- Zu Beginn ist der Speicher mit `DAT 0, 0` gefüllt.

Der Prozessor führt die Anweisungen aus. Jede Anweisung besteht aus dem Opcode für den Befehl und den beiden Operanden. Diese bestehen ihrerseits aus einem Wert und dem

dazugehörigen Adressierungsmodus. (Hinweis: Diese Information soll dazu dienen, geeignete Datenstrukturen zu finden. Die Opcodes und die Adressierungsmodi sollen als Enumerations definiert werden.)

Folgende Begrenzungen sollen gelten:

- Der Speicher soll eine Größe von 8.000 Instruktionen haben.
- Programme sollen nicht mehr als 256 Anweisungen beinhalten dürfen.
- Der Abstand zwischen 2 Programmen im Speicher soll mindestens 300 Anweisungen betragen.

Das Programm `redcode.c` soll als Argumente zwei Dateinamen bekommen, in denen Opcode für die Redcode-Maschine steht. Die zwei Programme sollen dann im Wettstreit gegeneinander antreten. Die Regeln des Wettstreits („Core War“) sind:

- Für beide Programme wird zu Beginn zufällig ein sich nicht überschneidender Platz im Speicher ausgewählt.
- Jedes Programm hat seinen eigenen „Program-Counter“, der den nächsten Befehl angibt. In jeder Runde wird zuerst für das erste, dann für das zweite Programm die Anweisung, auf die der zugehörige Program-Counter verweist, ausgeführt.
- Das Programm, das zuerst versucht eine Dateninstruktion auszuführen, verliert.

Testet das Programm mit Beispielopcode. Der Gewinner soll ausgegeben werden. Zusätzlich sollte die Funktionsfähigkeit des Programmes aber noch mit anderen Ausgaben getestet werden, z. B. mit Ausgabe des jeweils ausgeführten Befehls.

```
submit ssl 9 redcode.c team
```

Viel Erfolg!