

10. Übungsblatt: Programmierpraktikum II (SS 2003)

Abgabe: 14. Juli 2003

Bemerkungen:

- Alle nachfolgenden Aufgaben sind für den M68000-Assembler zu lösen. Die Abgaben müssen sich mit dem BSVC-Emulator im PI-Pool ohne Fehlermeldungen und Warnungen assemblieren lassen.
- Die Abgabe ist ausführlich zu kommentieren; unzureichend kommentierte Assembler-Programme werden nicht korrigiert.

Vorbemerkung: Zu Zeiten, als Pacman noch der Höhepunkt der Computerspiele war, als der PC noch "Homecomputer" hieß und 64 kByte Arbeitsspeicher hatte, als man noch Assembler programmieren musste, um wenigstens etwas Performance aus der Kiste herauszuholen - da gab es unter den Informatikern ein recht beliebtes Spiel¹: Core Wars.

Bei Core Wars treten zwei Assemblerprogramme im gleichen Adressraum gegeneinander an und versuchen, sich gegenseitig zum Absturz zu bringen. Auch wenn wir hier keine kompletten Core-Wars-Programme schreiben können, wollen wir doch einige grundlegend erforderliche Techniken lernen, wie das Verändern der eigenen Position im Speicher (Aufgabe 1) und das Entdecken und Angreifen "gegnerischer" Programme (Aufgaben 2 und 3)!

Aufgabe 1: Core Wars I

(8 Punkte, Abgabedatei: core1.s)

Schreibe ein vollständiges Assembler-Programm, das ab Adresse \$2000 abgespeichert ist. Nach Ausführung des Programmes soll es aber ab Adresse \$3000 stehen, und das ursprüngliche Programm soll gelöscht sein!

Aufgabe 2: Core Wars II

(7 Punkte, Abgabedatei: core2.s)

Schreibe ein vollständiges Assembler-Programm, das in einem fest vorgegebenen Speicherbereich nach allen Branch-Befehlen (Bcc, BRA und BSR) sucht. Jedesmal, wenn ein Branch-Befehl gefunden wird, wird der gesamte umgebende Block (d.h.

¹Damals gab es nämlich auch noch kein Internet und nur drei Fernsehprogramme, so dass man für seine Unterhaltung noch selbst sorgen musste.

alle vorne und hinten angrenzenden Worte bis zum ersten Nullwort) um den Branch-Befehl überschrieben, d.h. auf Null gesetzt.

- Der zu durchsuchende Speicherbereich wird unter den Adressen \$7000 und \$7001 beschrieben. Dabei enthält \$7000 die Adresse des ersten und \$7001 die des letzten zu betrachtenden Speicherwortes.
- Stößt das Programm im betrachteten Speicherbereich auf ein Wort, das die Struktur eines Branch-Befehls aufweist, so muss nicht geprüft werden, ob es sich tatsächlich um einen solchen handelt - das Wort wird unbesehen überschrieben.

Aufgabe 3: Core Wars II

(7 Punkte, Abgabedatei: core3.s)

Schreibe eine Abwandlung von Aufgabe 2. Es wird ebenfalls im vorgegebenen Speicherbereich nach Branch-Befehlen gesucht. Wird einer gefunden, so wird diesmal die dort enthaltene Adressdistanz so verändert, dass der Branch-Befehl zu dem Befehl verzweigt, dessen Adresse unter \$7002 gespeichert ist.