

Vorlesung Rechnersysteme

Übungsblatt 8

ab 8. Juni 2010

Hinweis: Nutzen Sie bei der Erstellung der Programme den Easy68k. Bringen Sie das fertige Programm zur Demonstration auf einem USB-Stick mit.

Aufgabe 1

Entwickeln Sie ein 68k-Assemblerprogramm, das folgenden Programmablauf erzeugt: Wenn der Inhalt von Register D4 einer 2 entspricht, soll nur das Unterprogramm 1 einmal aufgerufen werden. Entspricht der Inhalt einer 3, wird nur Unterprogramm 2 einmal aufgerufen, und für alle anderen Werte in D4 soll nur Unterprogramm 3 einmal aufgerufen werden.

Schreiben Sie zunächst Pseudocode, um im nächsten Schritt daraus den Assemblercode zu entwickeln.

Aufgabe 2

Entwickeln Sie ein 68k-Assemblerprogramm mit dem Namen SRCT, das in einer Tabelle (Array), die im Speicher abgelegt ist, ein Zeichen sucht und die Vorkommen zählt. Dieses Zeichen ist unter der Adresse ACHAR vorgegeben. Die Tabelle beginnt bei der Adresse ATABEL. SRCT soll die Adresse des Speicherplatzes, unter der das zu suchende Zeichen erstmals gefunden wird, im Speicher unter der Adresse AADR ablegen. Die Anzahl der gefundenen Zeichen soll an der Adresse ACT im Speicher abgelegt werden. Die Länge der Tabelle befinde sich im Register D2. Wenn das Zeichen nicht gefunden wird, soll das Unterprogramm ERROR aufgerufen werden. Geben Sie das Ergebnis auf den Bildschirm aus.

Schreiben Sie zunächst Pseudocode, um im nächsten Schritt daraus den Assemblercode zu entwickeln.

Aufgabe 3

Entwickeln sie ein Assembler-Programm zur Berechnung der Fibonacci-Zahlen.

Aufgabe 4

Gegeben sei ein Auszug aus einem x86-Assemblerquelltext in Intel-Format, in dem eine Funktion aufgerufen wird.

```
...
push ebx
push dword 1312h
push eax
push ecx
call func
add esp, 12
L2:
...
func:
push ebp
move ebp, esp
sub esp, 12h
L1:
...
leave
ret
...
```

1. Beschreiben sie, wie viele *Parameter/lokale Variablen* die Funktion hat und was während des Funktionsaufrufs passiert.
2. Skizzieren sie abschließend den Stackaufbau beim Erreichen der Labels L1 und L2.

Aufgabe 5

1. Beschreiben sie die Vorteile die sich bei der Verwendung von positionsunabhängigem Code ergeben.
2. Welche Mechanismen zur Umsetzung von positionsunabhängigem Code existieren? Beschreiben sie diese genauer!