

Rechnersysteme SS 2009

Aufgabenblatt 4

Vorzustellen vom 5.5.–11.5.

Aufgabe 1

- Was unterscheidet die befehlzählerrelative Adressierung von den anderen Adressierungstechniken? Wie werden bei dieser Technik die Adressen gebildet?
- Welcher positive wichtige Aspekt eines übersetzten Programms im Speicher wird durch die befehlzählerrelative Adressierung ermöglicht? Welcher Adressierungsteil darf nicht verändert werden, um eine eindeutige Adressierung zu gewährleisten?
- Mit welchen bekannten Adressierungsarten kann man die befehlzählerrelative Adressierungstechnik kombinieren?

Aufgabe 2 (Programmierung)

Es fehlt dem nachfolgend gezeigten Assembler-Code leider jegliche Funktionsbeschreibung. Zudem haben sich ein oder mehrere Fehler eingeschlichen. Finden Sie die Funktion heraus und debuggen Sie den Code!

```

                org      $0
                dc.l     $10000      * Stackpointer nach RESET
                dc.l     start      * Startadresse nach RESET

start           org      $1000
                lea     src,a0
                lea     dest,a1
                move.b  ctSrc,d0
                ext.w   d0
                jumpsr  ksp
                break

ksp            subq    #1,d0
schl           move.b  (a0)+,d1
                move.b  d1,(a1)+
                dbra   d0, schl
                rts

source        org      $2000
                dc.b    1,10,2,20,3,30,4,40,5,50,6,60
ctSrc         dc.b     12
destination   org      $3000
                ds.b    1
```

Aufgabe 3 (Unterprogrammtechniken)

1. Warum werden Unterprogramme (Subroutinen) bei der Programmierung eingesetzt?
2. Im 68k-Assembler gibt es die sehr ähnlichen Befehle JSR und BSR, um ein Unterprogramm aufzurufen. Was geschieht bei einem Aufruf? Wie unterscheiden sich die beiden Befehle?
3. Wie heißt das Gegenstück zu den 68k-Assemblerbefehlen JSR und BSR, um aus dem Unterprogramm zurückzukehren? Was geschieht beim Aufruf dieses Befehls?
4. Wie kann man im 68k-Assembler Parameter an ein Unterprogramm übergeben?