

MC68000-Assembler: Beispiel

Beispielprogramm: Bubblesort

| Unterprogramm Sortieren (Bubblesort)
 | Feld wird aufsteigend der Größe nach sortiert

| Verwendung der Register:
 | D0 : Schleifenzähler äußere Schleife [LAENGE-1..0]
 | D1 : Schleifenzähler innere Schleife [D0..0]
 | A0 : Adresse des aktuellen Feldelements
 | D2 : aktuelles Feldelement, wird mit darunterliegendem verglichen

| Vereinbarungen
 10 #define LAENGE ... | Länge des Feldes (Anzahl der Elemente)
 11 #define FELD ... | Zeiger auf Ende des Feldes
 | (Adresse des letzten Feldelements)

```

20      MOVEQ   LAENGE,D0 | äußeren Schleifenzähler initialis.
21      SUBQ   #1,D0
22      BLE    Ende      | ein Element ist immer sortiert
23 Schl1: MOVE.W D0,D1   | inneren Schleifenzähler initialis.
24      MOVEA.L FELD,A0 | A0 auf letztes Feldelement setzen
25 Schl2: MOVE.W (A0),D2 | akt. Feldelement holen
26      CMP.W  -(A0),D2 | und mit darunterliegendem vergleichen
27      BGE    KeinT     | richtig angeordnet? -> kein Tausch
28 Tausch: MOVE.W (A0),2(A0) | sonst größeres Element nach oben,
29      MOVE.W D2,(A0)   | kleineres nach unten bringen
30 KeinT: SUBQ  #1,D1    | inneren Schleifenzähler dekrement.
31      BNE    Schl2     | evtl. nächsten Schleifendurchlauf
32      SUBQ  #1,D0      | äußeren Schleifenzähler dekrement.
33      BNE    Schl1     | evtl. nächsten Schleifendurchlauf
34 Ende: ...
    
```

MC68000-Assembler: Keller

Aufgabe: In einer Tabelle, die in einem vorgegebenen Speicherbereich abgelegt ist und die nachstehende Form hat, soll mittels einer Matrikelnummer die zugehörige Note gesucht werden.

	Note	Matrikelnummer	Name (Länge variabel, aber gerade)	Null-Byte
1. Datensatz	'1'	190000	Müller, Hans	0
2. DS	'3'	190001	Meier, Uwe	0
...
letzter DS	'2'	190255	Schmidt, Kalle	0
"EOF"	0			

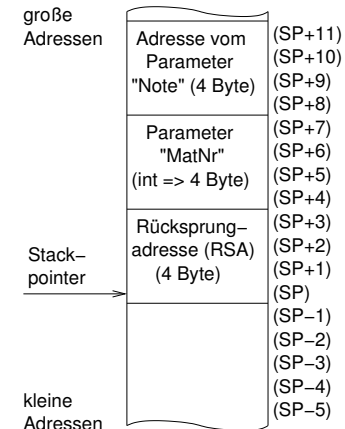
Prozeduraufruf über: void suche_note(int MatNr, char *Note)

Gegeben sind des weiteren:

TabAnfang, TabEnde, eine Zahl vom Typ int belegt 4 Byte
 Gerade Namenslänge, weil Zugriff auf „Longword“ nur an geraden Adressen;
 daher auch TabAnfang mit ungerader Adresse

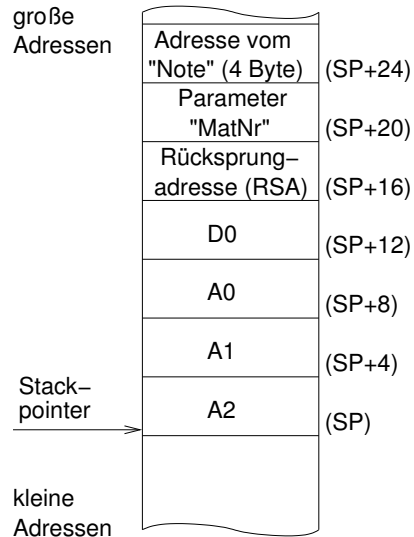
MC68000-Assembler: Unterprogrammaufruf I

gegeben: Aufgerufen wurde „void suche_note(int MatNr, char *Note)“ mit JSR, wobei int 4 Byte belegt und das Unterprogramm die Register D0, A0, A1 und A2 benutzt und deshalb sichert und zurücksichert.



- Abbildung zeigt den Stack direkt nach dem Funktionsaufruf
- Das Unterprogramm sichert D0, A0, A1 und A2 auf dem Stack.
- D0, A0, A1 und A2 **müssen** vor dem Rücksprung mit RTS wieder vom Stack entfernt werden!
- Erinnerung: „Big endian“, d.h. höherwertigeres Byte an niedrigerer Adresse

MC68000-Assembler: Beispiel (Forts.)



Bemerkungen:

- Abbildung zeigt den Stack nach dem Aufruf von `suche_note` und dem Sichern der Register D0, A0, A1 und A2 auf dem Stack
- Die Parameter der Funktion können über das Adressregister SP indirekt mit Adressdistanz +20 (Matrikelnummer) und +24 (Adresse an dem das Ergebnis gespeichert werden soll) adressiert werden.
- D0, A0, A1 und A2 **müssen** vor dem Rücksprung mit RTS wieder vom Stack entfernt werden!

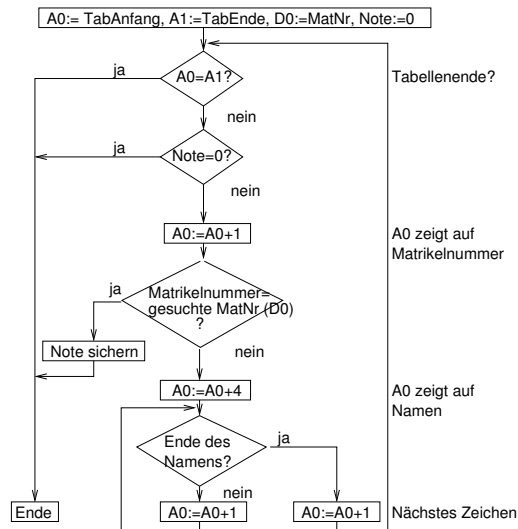
MC68000-Assembler: Beispiel (Forts.)

```

| A0 : Tabellenanfang
| A1 : Tabellenende
| A2 : Adresse des Ergebnisses (Note)
| D0 : Gesuchte Matrikelnummer
10 #define aMatNr 20(SP)
11 #define aNote 24(SP)
20 .EVEN
21 .GLOBAL _suche_note
30 _suche_note: MOVEM.L D0/A0-A2, -(SP) | Inhalt der benutzten Register sichern
31             LEA _TabAnfang, A0      | globale Variable in Register kopieren
32             LEA _TabEnde, A1        |
33             MOVE.L aMatNr, D0       |
34             MOVEA.L aNote, A2       |
35             MOVE.B #0,(A2)          | "Default"-Ergebnis 0 = nicht gefunden
...
| hier steht der "eigentliche" Code

60 Ende: MOVEM.L (SP)+, D0/A0-A2 | gesicherte Register zurueckholen
61 RTS
    
```

MC68000-Assembler: Bsp. (Ablaufplan) optional



MC68000-Assembler: Bsp. (Programm) optional

Dieser Code ist einzufügen in den vorigen Rahmen:

```

40 Schleife1: CMPA.L   A0,A1           | Ende der Tabelle erreicht?
41             BEQ     Ende           | Wenn ja: Ende

42             CMPI.B  #0,(A0)+       | Note=0 ("EOF")?
43             BEQ     Ende           | Wenn ja: Ende

44             CMP.L   (A0)+,D0        | gesuchte Matrikelnummer gefunden?
45             BEQ     Erfolg         | Wenn ja: Note kopieren

46 Schleife2: CMPI.B  #0,(A0)+       | Ende des Namens?
47             BEQ     Schleife1      | Wenn ja: naechster Datensatz

48             BRA    Schleife2       | Naechstes Zeichen

49 Erfolg:     MOVE.B  -5(A0),(A2)    | Note uebertragen
    
```