Jens Reinecke

Disassemblieren in den AIM-Texteditor

Vor allem aus Platzgründen werden Maschinenprogramme oft nur als Hex-Dump und nicht als Assembler-Sourcelisting abgedruckt. Wenn man dann versucht, solche Programme zu ändern, z. B. in einen anderen Adressenbereich zu schieben oder einzelne Befehle noch zusätzlich einzufügen, hilft das folgende Programm für AIM-65 und PC-100: Es verwandelt Objektcode in assemblierfähigen Source-Code, der dann im Texteditor beliebig geändert werden kann.

Oft steht man vor dem Problem, daß man ein Programm, welches nur als Objektcode vorliegt, in einem Adressenbereich laufen lassen möchte, für den es nicht geschrieben ist.
Wenn keine Absolutsprünge, hierzu zählen auch Unterprogrammaufrufe die auf Adressen innerhalb des Programms zeigen, vorkommen, so kann man sich ein kleines Verschiebeprogramm schreiben und dann mit dessen Hilfe das Programm da hineinschieben, wo man es laufen lassen möchte. Tauchen aber die schon erwähnten Absolutsprünge auf, wird der Aufwand schon umfangreicher, da man die Sprungadressen entsprechend der neuen Startadresse des Programms ändern muß; man kann sich leicht vorstellen, daß bei größeren Programmen der Aufwand erheblich steigen kann. Das größte Problem aber wird auftreten, wenn man Befehle oder ganze Programmabschnitte einfügen oder streichen will, da man dann nicht nur die Absolutsprünge ändern muß, sondern auch eventuell betroffene Relativsprünge (bedingte Sprünge); das bedeutet, daß man in mühevoller Arbeit die Sprungweite errechnen muß.

Der Komfort-Disassembler schreibt in den Editor

Dieses Programm (*Bild 1*) arbeitet mit dem im AIM befindlichen Disassembler zusammen, übersetzt den Maschinencode in Assemblerformat und schreibt ihn dann in den AIM-Editorbereich. In mehreren Durchläufen werden dann alle Adressen von Absolutsprüngen, Subroutinenaufrufen und Relativsprüngen durch Labels ersetzt, ebenso steht das entsprechende Label vor dem Befehl, der angesprungen werden soll. Nach Durchlauf des Komfort-Disassemblers kann man wie von AIM gewohnt mit der Taste T in den Editor springen und braucht nur noch die neue Startadresse einzugeben. Nach dem dann folgenden Assemblerdurchlauf hat man das Programm dann an der gewünschten Stelle im RAM-Bereich. Natürlich ist es jetzt auch kein Problem mehr, im Editor Befehle oder ganze Programmabschnitte einzufügen bzw. zu löschen.
Nach dem Start bei hex 0200 fragt das Programm zunächst den Benutzer, ob es den Source-Code in den eventuell schon vorher vom Monitorprogramm aus mit E initialisierten Editor-Adressenbereich schreiben soll oder ob ein neuer Bereich angegeben werden soll (FROM/TO).
Dann kann man den gewünschten zu disassemblierenden Adressenbereich eingeben, worauf darauf zu achten ist, daß dort auch wirklich ein Maschinenprogramm und nicht irgendwelche Text- oder Datentabellen existieren, da sonst Fehler auftreten (*Bild 2*).
Während der ersten „Bearbeitungsphase" sieht man im Display die disassemblierten Befehle. Später beseitigt das Programm noch unnötige Leerräume am Zeilenbeginn und -ende, wobei WAIT im Display erscheint. Nach READY kann man irgendeine Taste antippen, und man ist wieder im Monitor.

Besonderheiten des Programms

Wenn man die automatische „Leerraumbeseitigung" nicht haben möchte, z. B. um die Übersichtlichkeit des Listings zu erhöhen, kann man bei der Adresse 0501 drei NOP-Befehle einsetzen. Zu beachten ist, daß das Programm vorübergehend mehr Speicherplatz benötigt, als am Schluß im Editor steht, da ja die Leerräume erst in der letzten Phase beseitigt werden. Dies kann rund ein Drittel des Platzbedarfs ausmachen.
Selbstverständlich ist es auch möglich, mehrere Speicherbereiche unabhängig voneinander nacheinander zu disassemblieren, auf Kassette als Editor-File abzuspeichern und später in der gewünschten Reihenfolge aneinanderzuketten. Dazu ist in die GAP-Zelle A409 ein Wert von wenigstens 18 zu schreiben. Zu bedenken ist schließlich, daß das Programm nur solche Argumente in Labels umwandelt, die innerhalb des disassemblierten Objektcode-Adressenbereichs liegen; alle anderen erscheinen als hexadezimales Argument. Darauf ist besonders dann zu achten, wenn Sprünge zu einer Adresse außerhalb dieses Bereichs führen. In diesem Fall kann man die Labels von Hand mit dem Change-Befehl des Editors einsetzen.
In *Bild 3* findet sich schließlich – zur Kontrolle der richtigen Eingabe von Bild 1 – ein Prüfsummen-Programm, dessen eigene Prüfsumme 2C62 beträgt. *Bild 4* zeigt die damit errechneten Prüfsummen für unterschiedliche Adressenbereiche des Komfort-Disassemblers.

```
↑)=0200 A9 23 85 16 A9 00 85 19 85 1A 20 A3 E7 AD 1C A4
( ) 0210 85 12 8D 25 A4 85 1C AD 1D A4 85 13 8D 26 A4 20
( ) 0220 A7 E7 AD 1C A4 85 1E AD 1D A4 85 1F A2 00 20 7C
( ) 0230 04 20 3C E9 C9 4E F0 21 C9 59 D0 F5 20 44 EB 20
( ) 0240 A3 E7 AD 1C A4 85 E3 AD 1D A4 85 E4 20 A7 E7 AD
( ) 0250 1C A4 85 E5 AD 1D A4 85 E6 A5 E3 85 10 E6 E3 A5
( ) 0260 E4 85 11 D0 02 E6 E4 20 44 EB A9 00 A2 14 CA 9D
( ) 0270 38 A4 D0 FA A9 01 8D 19 A4 20 2B E7 AD 25 A4 A0
( ) 0280 00 C8 E6 1C C5 1C D0 F9 84 1D A2 00 BD 38 A4 E8
( ) 0290 C9 00 D0 F8 E8 A9 0D 9D 38 A4 CA A5 1D 09 30 9D
( ) 02A0 38 A4 CA A9 20 9D 38 A4 E8 E8 8A A8 38 8A E9 04
( ) 02B0 85 1B 98 88 88 88 BD 38 A4 29 7F 91 10 E0 0D D0
( ) 02C0 29 A9 01 C5 1D D0 06 A9 20 88 4C E8 02 BD 38 A4
( ) 02D0 C9 23 F0 04 C9 28 D0 0D A9 24 91 10 88 BD 38 A4
( ) 02E0 91 10 4C EA 02 88 A9 24 91 10 CA 88 D0 C8 18 A5
( ) 02F0 1B 65 10 85 10 90 02 E6 11 E6 19 D0 02 E6 1A 38
( ) 0300 AD 25 A4 E5 1E AD 26 A4 E5 1F B0 22 18 A5 10 69
( ) 0310 14 85 25 A5 11 85 26 38 A5 25 E5 E5 A5 26 E5 E6
( ) 0320 B0 03 4C 67 02 A2 14 20 7C 04 20 3C E9 00 AD 25
( ) 0330 A4 85 23 AD 26 A4 85 24 18 A5 10 69 06 85 E1 90
( ) 0340 07 A6 11 E8 8A 4C 4A 03 A5 11 85 E2 A0 06 A2 0D
( ) 0350 8D 88 05 91 10 E8 88 D0 F7 A2 2A 20 7C 04 A5 12
( ) 0360 85 10 A5 13 85 11 A5 E3 85 14 85 1E 85 20 A5 E4
( ) 0370 85 15 85 1F 85 21 A5 19 85 17 A5 1A 85 18 A2 00
( ) 0380 A0 00 18 A5 17 E9 00 85 17 B0 07 C6 18 10 03 4C
( ) 0390 9A 04 B1 14 20 84 EA C8 B1 14 20 84 EA 85 1B A2
( ) 03A0 0B BD EA 05 C5 1B F0 03 CA D0 F6 A0 00 B1 14 C8
( ) 03B0 84 22 C9 0D D0 F7 E0 00 D0 0D 98 18 65 14 85 14
( ) 03C0 90 02 E6 15 4C 7E 03 88 88 88 A5 16 91 14 98 48
( ) 03D0 48 A5 14 85 25 A5 15 85 26 A0 0A B1 14 20 84 EA
( ) 03E0 C8 B1 14 20 84 EA 85 1C A0 08 B1 14 20 84 EA C8
( ) 03F0 B1 14 20 84 EA 85 1D 68 A8 38 A5 1C E5 23 A5 1D
( ) 0400 E5 24 B0 0B 38 A5 1C E5 12 A5 1D E5 13 B0 0A A9
( ) 0410 20 91 14 20 8E 04 4C 68 04 20 8E 04 4C 45 04 A0
( ) 0420 00 A9 0D D1 1E F0 04 C8 4C 23 04 88 B1 1E 29 0F
( ) 0430 18 65 12 85 12 90 02 E6 13 C8 C8 98 18 65 1E 85
( ) 0440 1E 90 02 E6 1F A5 1C C5 12 D0 D4 A5 1D C5 13 D0
( ) 0450 CE A0 02 B1 1E C9 20 F0 09 AA 68 A8 8A 91 25 4C
( ) 0460 69 04 A5 16 91 1E E6 16 68 A5 20 85 1E A5 21 85
( ) 0470 1F A5 10 85 12 A5 11 85 13 4C 7E 03 20 44 EB BD
( ) 0480 B8 05 C9 04 F0 07 20 7A E9 E8 4C 7F 04 60 A5 22
( ) 0490 18 65 14 85 14 90 02 E6 15 60 A5 19 85 17 A5 1A
( ) 04A0 85 18 18 A5 17 E9 00 85 17 B0 02 C6 18 A9 30 85
( ) 04B0 1C A9 41 85 1D A5 E3 85 14 A5 E4 85 15 A9 20 A0
( ) 04C0 00 91 14 C8 91 14 C8 D1 14 D0 42 AA A9 0D C8 D1
( ) 04D0 14 D0 FB C8 84 22 88 88 8A 91 14 88 D1 14 F0 11
( ) 04E0 20 13 05 88 88 88 A9 20 D1 14 F0 05 91 14 88 D0
( ) 04F0 F7 20 8E 04 18 A5 17 E9 00 85 17 B0 C0 C6 18 10

( ) 0500 BC 4C 48 05 A2 24 20 7C 04 20 3C E9 00 20 13 05
( ) 0510 4C CB 04 A2 3A A5 1D 85 1F A5 1C 85 1E A9 22 18
( ) 0520 69 01 D1 14 F0 11 E6 1E E4 1E D0 F3 E6 1F 48 A9
( ) 0530 30 85 1E 68 4C 1F 05 88 88 A5 1F 91 14 C8 A5 1E
( ) 0540 91 14 C8 A9 20 91 14 60 A5 E3 85 14 85 10 A5 E4
( ) 0550 85 15 85 11 A0 00 B1 14 C9 0D F0 09 20 A1 05 20
( ) 0560 A8 05 4C 56 05 20 AF 05 B1 10 C9 20 F0 F7 20 A8
( ) 0570 05 B1 14 91 10 20 A8 05 20 A1 05 B1 14 C9 20 F0
( ) 0580 F7 B1 14 C9 0D F0 DE 91 10 20 A1 05 20 A8 05 C9
( ) 0590 00 D0 EE 20 AF 05 A5 10 85 E1 A5 11 85 E2 4C 04
( ) 05A0 05 E6 14 D0 02 E6 15 60 E6 10 D0 02 E6 11 60 18
( ) 05B0 A5 10 E9 00 85 10 B0 02 C6 11 60 45 44 49 54 4F
( ) 05C0 52 20 4E 45 55 20 3F 04 00 0D 44 4E 45 2E 04 45
( ) 05D0 44 49 54 4F 52 20 4F 56 45 52 46 4C 4F 57 04 52
( ) 05E0 45 41 44 59 04 57 41 49 54 04 EA 4C 90 B0 F0 30
( ) 05F0 D0 10 50 70 20 B0 F0 30 D0 10 50 70 20 20 10 50
```

Bild 1. Das Maschinenprogramm umfaßt ziemlich genau 1 KByte und disassembliert in drei Durchläufen. Dabei entsteht ein assemblierfähiger Quellencode im AIM-Texteditor

```
    =(L)
    /
    OUT=
       JSR $0513
    DEY
    DEY
    DEY
    LDA #$20
    A1 CMP ($14),Y
    BEQ    A0
    STA ($14),Y
    DEY
    BNE    A1
    A0 JSR $048E
    CLC
    LDA $17
    SBC #$00
    STA $17
    BCS $04BD
    DEC $18
    .END
```

Bild 2. So sieht der disassemblierte Speicherbereich 4E0...4FE aus. Um den entstandenen Quellencode wieder zu assemblieren, braucht man nur vor die erste Zeile noch eine Anfangsadresse zu schreiben

```
(M)=0000 20 A3 E7 B0
( ) 0004 FB AD 1C A4
( ) 0008 85 F2 AD 1D
( ) 000C A4 85 F3 20
( ) 0010 A7 E7 B0 FB
( ) 0014 A0 00 84 F0
( ) 0018 84 F1 18 B1
( ) 001C F2 65 F0 85
( ) 0020 F0 90 02 E6
( ) 0024 F1 E6 F2 D0
( ) 0028 02 E6 F3 A5
( ) 002C F3 CD 1D A4
( ) 0030 D0 E8 A5 F2
( ) 0034 CD 1C A4 D0
( ) 0038 E1 20 3E E8
( ) 003C A5 F1 20 46
( ) 0040 EA A5 F0 4C
( ) 0044 46 EA 00 00
(*)=10C

(/) 010C 4C 00 00
(C)FROM=0 TO=46   2C62
```

Bild 3. Um das Programm von Bild 1 fehlerfrei eingeben zu können, empfiehlt sich die Verwendung dieses Prüfsummen-Programms für AIM-65 und PC-100 (Start mit F1)

```
(C)FROM=200 TO=300   799F
(C)FROM=300 TO=400   697E
(C)FROM=400 TO=500   6667
(C)FROM=500 TO=5FE   5DD0
(C)FROM=200 TO=5FE   A754
```

Bild 4. Das sind die Prüfsummen für die Bytes von Bild 1, angegeben für unterschiedliche Adressenbereiche