

MG DISKASSEMBLER - Assembler rückwärts

Mit der Besprechung dieses jüngsten Produktes der Millers Graphics Softwareschmiede wird die Beitragsreihe über die bemerkenswerten Programme dieser Firma beendet. Doch keine Angst - sobald das neue Extended Basic Modul Ver.120 von MG eingetroffen ist (ist bereits bestellt), kommt wieder Nachricht!

DISKASSEMBLER ist, wie auch die anderen beiden, DIAGS und EXPLORER, ein Programm, das nichts bringt, was im Grundsatz neu wäre.

Doch die Realisierung zeigt auch dieses Mal, daß man bei MG eine enorm anwenderfreundliche Synthese aller Merkmale bereits vorhandener Programme (hier Disassembler; davon gibt es für den TI eine ganze Menge!) fand und zudem einige neue Optionen einfügte.

Das soll ein kleiner Überblick verdeutlichen:

- DISKASSEMBLER kann entweder Speicherinhalte disassemblyn oder direkt von Diskette disassemblyn, also direkt im File.
- Speicherinhalte kann so gut wie jeder Disassembler analysieren, und wer von Diskette arbeiten will, der kann den 'Universal Disassembler' von Rene LeBlanc verwenden, der als Freeware verfügbar ist.
- Aber DISKASSEMBLER benötigt als einzige Information den Filenamen, wogegen man sich beim Universal Disassembler erst mühsam die Sektoren des Files zusammensuchen muß.
- DISKASSEMBLER kann das Ergebnis seiner Arbeit als File zur Verfügung stellen.
- Das kann der Weiand Disassembler V2.1 auch, der ebenfalls als eine Art Freeware verfügbar ist. Doch der kann keine Files disassemblyn.
- DISKASSEMBLER erlaubt das blockweise Disassemblyn eines beliebigen Maschinencodes. Das kann sonst kein anderes Programm. Bei allen anderen Programmen muß das in Handarbeit geschehen, indem abwechselnd Start- und Endadressen neu eingegeben und dazwischen Daten als Hexdump aufgelistet werden. DISKASSEMBLER macht das nach Vorgabe der Bereiche selbständig.

Soweit dieser kurze Vergleich, schließlich soll das eine Programmbeschreibung und keine Disassemblerübersicht werden.

WAS IST EIN DISASSEMBLER?

Doch halt! Die Kernfrage wurde bisher noch nicht beantwortet: Was ist denn überhaupt ein Disassembler?

Die Antwort klang in der Überschrift schon an und auch die Bezeichnung dieser Art Programm mag als Hinweis dienen.

Ein Disassembler macht die Arbeit eines Assemblers rückgängig,

indem er aus einem Maschinenprogramm wieder einen Quellcode erzeugt, der, je nach Qualität des Disassemblers, mehr oder weniger ähnlich dem Quellcode ist, den der Autor beim Assemblieren vorliegen hatte.

Normalerweise sieht man ja nur, was ein Programm macht, also den Output. Ein Disassembler zeigt, WIE es gemacht wird. Natürlich muß man zur Interpretation dessen, was ein Disassembler liefert, selbst in der Lage sein, Assemblerprogramme zu schreiben. Doch dadurch, daß einem ein Disassembler den Einblick in das Know-How anderer Programmierer erlaubt, kann man den einen oder anderen Kniff lernen. Und genau das ist m.E. die hauptsächliche Existenzberechtigung eines solchen Programms.

LEISTUNGSMERKMALE

Das hat wohl auch Thomas S. Freeman gedacht, als er DISKASSEMBLER schrieb, denn das Programm weist viele Punkte auf, die das Arbeiten mit ihm erleichtern.

So erkennt DISKASSEMBLER vollkommen selbstständig, welche Art von File vorliegt (Tagged Object Code oder Memory Image), wie lang es ist (bei Standard-Typen) und inwieweit der Maschinencode an feste Adressen gebunden ist.

Er erlaubt die Spezifikation von Datenfeldern, die keinen Maschinencode enthalten und daher nicht disassembliert werden müssen sowie die Angabe von besonderen Unterprogrammaufrufen, denen Datenworte folgen. Bei diesen erscheinen dann die DATAs als Daten und nicht als wenig sinnvolle Befehle.

Daneben erlaubt das Programm die simultane Ausgabe des Ergebnisses auf Bildschirm, Drucker und Diskette (als File, das in vielen Fällen direkt wieder assembliert werden kann!). Dabei wird die Länge eines jeden Files auf maximal 65 Sektoren begrenzt und danach ein neues geöffnet. Auch macht es nichts, wenn die Diskette voll wird - das letzte File wird korrekt geschlossen und das Einlegen einer neuen Diskette abgewartet - kein Datenverlust!

Doch das Wichtigste ist: DISKASSEMBLER ist ein Zwei-Pass-Disassembler.

Das bedeutet, daß er sich zwei Mal um den Inhalt eines Files oder Speicherbereiches kümmert. Dabei 'merkt' er sich gewisse Parameter (genau genommen baut er eine Symboltafel ähnlich einem Assembler auf), mit deren Hilfe er im zweiten Pass (Durchlauf) richtige Labels vergibt. Damit ist das Ergebnis seiner 'Arbeit' näher am Original des Source-Codes als das irgendein anderer Disassembler erlaubt!

Aufgrund dieser Eigenschaft wird DISKASSEMBLER als 'Symbolischer Disassembler' bezeichnet, weil er in der Lage ist, auf Speicherinhalte nicht in direkter Weise, sondern eben über Symbole hinzuweisen, eine wichtige Voraussetzung zum Erreichen eines relocatiblen Codes!

Aber auch DISKASSEMBLER vollbringt keine Wunder. Denn daß der produzierte Quellcode fast immer direkt und ohne Fehler assembliert werden kann, hilft ja gar nichts, denn dann ist man ja so weit wie vorher. Wer irgendetwas ändern will, kommt um das Verständnis dessen, was DISKASSEMBLER liefert, nicht herum. Dazu gehört nach wie vor einige Erfahrung in der Programmierung von TMS9900-Maschinensprache.

Bei der Gelegenheit einige Worte zu dem zweideutigen Begriff 'Assembler'. In der deutschen Sprache gibt es ein passendes Beispiel, den 'Widerstand'. Einmal ist es eine Sachbezeichnung und dann eine Eigenschaft. Ähnlich beim Assembler. Gelegentlich wird die Maschinensprache einer beliebigen CPU als 'Assembler' bezeichnet, üblicherweise zusammen mit der vorangestellten

Bezeichnung der betreffenden CPU. Um das zeitraubende Bitgerummel der echten Maschinenbefehle zu umgehen, gibt man nun jedem Operationscode (Op-Code) der CPU einen Namen, das sogenannte Mnemonic. Mit diesen Namen und deren Zusätzen (der Syntax) schreibt man nun ein Programm in einer Sprache, die 'Assembler' genannt wird. Um diese Sprache auf das Bit-Niveau der CPU zu bringen, verwendet man einen 'Assembler' - einen 'Zusammenfüger'. Benommenen liegt aber diese Programmierung in Assembler schon einen Schritt über dem Prozessorniveau, so daß der Assembler, der Assemblerbefehle assembleert, eigentlich 'Compiler' heißen müßte. Da so aber nur solche Programme heißen, die eine noch weiter vom Maschinenniveau entfernte Hochsprache wie z.B. BASIC, PASCAL oder C in Maschinensprache übersetzen, blieb es beim doppeldeutigen Assembler. Das soweit zur Begriffsbestimmung.

DAS PROGRAMM

DISKASSEMBLER wird, wie üblich, als Diskette im Format 1S geliefert, also dem Format, was wohl alle TI-Systeme lesen können.

Der Loader von DISKASSEMBLER akzeptiert alle Controller, von Atronic über CorComp und Myarc bis TI-Original. Für den Geneva ist eine Sonderversion des Programms nötig, unabhängig vom eingesetzten Controller!

Für das CPS99 müssen ähnliche Patches im Loader vorgenommen werden, wie das beim EXPLORER notwendig war, damit das Programm startet.

Auf der Diskette befinden sich lediglich 3 Files. LOAD ist der Loader für den automatischen Start aus Extended Basic, DISKCONFIG ist das File, in dem sich die Optionen des Hauptprogramms befinden und DISKASSEM ist der universelle Loader für Editor/Assembler, Mini-Memory und Extended Basic.

Aus unerfindlichen Gründen (nicht, daß sie nicht zu finden seien, sondern das Warum ist unerfindlich) springt der Loader bei einem TI mit PAD-RAM-Erweiterung direkt zum Titelbild. Diese sollte beim Booten abgeschaltet werden.

Wie üblich liegt ein ausführliches Manual bei (58 Seiten), welches auf 34 Seiten eine Programmanleitung bekannter Güte beinhaltet. Daran anschließend folgen die Anhänge, zuerst ein Anwendungsbeispiel von DISKASSEMBLER anhand der SAVE-Utility des EA-Modules (Seiten 34 bis 42), 2 Seiten Tips, wie man zu lange Program-Files in zwei kleine zerlegt und 10 Seiten Tabellen mit Speicherbelegungen, wie man sie unter den 3 verschiedenen Modulen vorfindet.

Nachdem der Loader gestartet wurde, lädt das Programm innerhalb weniger Sekunden. In dieser Zeit erscheint eine Copyright Meldung mit Firmen- und Autorennamen sowie dem Copyright-Datum. Dieser Warte-Bildschirm wird dann vom ersten Arbeitsbildschirm abgelöst.

Hier wird man zur Eingabe eines Filennamens aufgefordert. Löscht man diesen mit FCTN 3, so kann ein Speicherbereich disassembliert

werden. Dabei ist DISKASSEMBLER genauso kapriziös wie EXPLORER – alles, was in seinen Bereich fällt, wird nicht bearbeitet. Wird also Ihre Eingabe von Start- und Endadresse ignoriert, so fühlt sich das Programm 'bedroht', weil Sie es selbst disassemblieren wollten.

Da DISKASSEMBLER am oberen Ende der Speichererweiterung

'residiert' können andere Programme vorher geladen und diese dann direkt im Speicher untersucht werden, vorausgesetzt, sie belegen nicht den Speicherbereich von >3000 bis >3FFF und >D000 bis >FFFF.

Das ist vor allem dann empfehlenswert, wenn diese Programme externe Referenzen besitzen wie DSRLNK, VSBW, KSCAN etc. Da diese vom Loader des jeweiligen Moduls rückwärts aufgelöst werden, spart man sich so eine Menge Zeit. Näheres folgt weiter unten.

In der unteren Bildhälfte des ersten Arbeitsschirmes befinden sich zusätzliche Informationen zum Objekt der Untersuchungen.

TYPE	Gibt bei File-Disassemblerungen den Filetyp an.
CODE	Zeigt an, ob es sich um komprimierten, unkomprimierten oder Memory-Image Code handelt.
LENGTH	Zeigt die Anzahl Bytes im Programm an, die vom Loader an beliebige Stellen des Speichers geschrieben werden können (Relocatable Code). Eine Null hier sagt aus, daß das Programm an feste Adressen gebunden ist.
LOADER	Gibt an, ob das File mit einem normalen Loader irgend-eines Moduls geladen werden kann, oder ob dafür ein weiteres Anwenderprogramm nötig ist (Angabe Standard oder Custom).
LOADS AT	Gibt die Ladeadresse an. Der Zusatz RORG gibt an, daß die Ladeadresse im Prinzip beliebig ist und nur von DISKASSEMBLER willkürlich auf die erste freie Standard-adresse gelegt wurde (so, wie es jeder moduleigene Loader tut). AORG gibt wieder die Bindung an einen festen Adreßbereich an.
SECTORS	Taucht nicht bei DIS/FIX-Files auf. Gibt ansonsten die Länge eines Memory-Image-Typ Files (Program-File) an, so wie sie im File-Descriptor-Sektor des Files steht.
OPTIONS	Damit kann die Arbeitsweise von DISKASSEMBLER festgelegt werden. Folgende Optionen sind verfügbar:

- R Wie beim Assembler – stellt ein 'R' vor jede Register-nummer.
Wir diese Option nicht gewählt, so können auch Labels Namen wie z.B. 'R5' oder 'R0' erhalten.
- B Subtrahiert ggf. einen Basic-Screen-Offset von Texten, damit diese lesbar werden. Betrifft nur Programme, die aus einem der Basics heraus benutzt werden.
- T Gibt der Anzeige von Text-Statements den Vorzug gegenüber den DATAs. Siehe bei der Besprechung des Ausgabeformats.
- D Ist nur bei Untersuchung von Speicherbereichen notwendig und gibt an, daß eine DSR untersucht werden soll. Damit wird bewirkt, daß simultan zum Auslesen des spezifi-zierten Speicherbereichs bestimmte CRU-Bits gesetzt werden können.
- X Normalerweise vergibt DISKASSEMBLER an Adressen, die nicht direkt im Bereich des spezifizierten Speichers oder des Files liegen, keine Labels. Wird 'X' aktiviert, so macht er es trotzdem und fügt eine Equate-Liste der

- V Bei externen Referenzen wie oben angeführt (VSBW etc.) werden diese vom moduleigenen Loader rückwärts aufgelöst. Liegen derartige Referenzen vor, so muß diese Option beim zweiten Pass aktiviert werden, so daß die Referenzkette klar erkennbar wird. Ansonsten würden an den entsprechenden Stellen nur Nullen auftauchen. Ob

TI99/4A

TI-CLUB ERRORFREE INFO NR. 5

SEITE 7

- solche Referenzen existieren, wird nach dem ersten Pass angezeigt (Out of order origins).
- M Zeigt an, daß ein Myarc-Diskcontroller benutzt wird. Da sich dieses Gerät nicht an die Konventionen zur Benutzung des VDP-RAMs hält, muß dann an anderen Stellen nach den File-Informationen gesucht werden, die DISKASSEMBLER benötigt (genauer im System-RAM des Controllers ab >5000).
- G Übersteuert die Option 'M' und zeigt an, daß die Files von einer RAM-Disk geholt werden. Viele der vorhandenen RAM-Disks benutzen den VDP-Puffer überhaupt nicht, so daß DISKASSEMBLER den File-Status getrennt mittels der Level-1 Routine >14 bestimmen muß.

Alle diese Optionen können im File DISKCONFIG festgelegt werden, wobei M und G tunlichst in diesem File festgelegt werden sollten, da diese Optionen bereits beim Laden eines Files aktiv sind und man sich so Fehl-Ladevorgänge spart.

Es ist aber möglich, daß man erstmal keinen Filenamen eingeibt und sich bis zu den Optionen vorarbeitet und diese dann korrekt einstellt. Mit FCTN 9 kann dann erneut ein Filename eingegeben werden.

DISKCONFIG kann z.B. mit dem EA-Editor bearbeitet werden. Es ist immer genau 3 Zeilen lang. In der ersten Zeile steht die Einstellung der Bildschirmfarben. Sie werden als Hex-Ziffern eingegeben, wobei die erste Ziffer für die Vorder- und die zweite für die Hintergrundfarbe steht, z.B. F1 für Weiß auf Schwarz oder F4 für Weiß auf Dunkelblau, der Standard-Wert des Programms. In der zweiten Zeile können die Optionen eingegeben werden. Sie erscheinen später genau so im Options-Feld des Hauptprogramms. Die dritte Zeile enthält den Namen des Druckers. Dieser darf maximal 38 Zeichen lang sein. Laden Sie doch einfach das vorhandene File, womit Sie ein Beispiel haben!

DIE BEDIENUNG

Wie bereits mehrfach erwähnt, erlaubt DISKASSEMBLER die Disassembly von Speicher- und Fileinhalten. Wird im ersten Bildschirm ein Filename eingegeben, so wird dieses File disassembly. Wird kein Filename eingegeben, so wird ein anzugebender Speicherbereich im CPU-Adresse Raum disassembly. GRAM/GROM oder VDF können nicht untersucht werden. Da dort i.A. aber auch keine Maschinenprogramme zu finden sind, ist diese Einschränkung tolerierbar.

Mit der Tastenkombination **ctrl + shift + f** kann die Datei im Datei- oder Filenamespace CTRL + gedrückt werden. Das Programm kann nur so verlassen werden!

Ansonsten stehen folgende Funktionen zur Verfügung (Reihenfolge wie im Handbuch):

FCTN 1,2 wie üblich zum Löschen und Einfügen eines Zeichens, FCTN 3 zum Löschen einer Eingabezeile, FCTN 4 bricht komplett ab und führt zur Eingabe des Filenamens, FCTN 6 zeigt den folgenden Bildschirm an (es gibt insgesamt 4), FCTN 9 führt schrittweise

jeweils einen Bildschirm zurück und CTRL i läuft durch die verschiedenen Möglichkeiten von Bildschirm-Vorder- und Hintergrundfarben.

Daneben sind die Cursortasten aktiv, mit denen man zwischen den Eingabefeldern herumfahren kann.

T 199/48 TI-CLUB ERRORFREE INFO NR. 5 SEITE

DISASSEMBLIEREN VON SPEICHERINHALTEN

Wird die Filenameneingabe mit einem Leerstring und ENTER verlassen, so erscheint ein fast leeres Informationsfeld, unter dem Start- und Endadresse eingegeben werden können. Dabei ist die Endadresse, wie auch bei allen anderen Hexadezimal-Eingaben, nicht inclusive!

Wurde im Optionsfeld ein 'D' eingegeben (DSRs disassemblieren), so erscheint zusätzlich eine Zeile mit CRU-Adressen. Dabei kann eine Softwarebasisadresse angegeben werden, an der das CRU-Bit 0 (relativ) gesetzt wird, der Speicherbereich in einen Puffer geholt und dieses Bit anschließend wieder rückgesetzt wird. Es können bis zu 4 Bits simultan geschaltet werden. Nullen als Adresse werden ignoriert. So muß z.B. für die Untersuchung des Diskcontrollers als Adresse >1100 eingegeben werden, bei einer Startadresse von >4000 und einer Endadresse von >6000, effektiv also >5FFE. Für die zweite RDM-Bank muß als zweite CRU-Adresse >1116 eingegeben werden.

Wurden alle Eingaben gemacht oder FCTN 6 gedrückt, so erscheint der zweite Bildschirm. Ab jetzt unterscheiden sich Speicher- und File-Disassembly im weiteren Ablauf nicht mehr.

DISASSEMBLIEREN VON DISKETTENFILES

Nachdem der Name eines existenten Files eingegeben wurde (z.B. DSK1.DISKASSEM - das geht!), wird geprüft, von welchem Typ das File ist. Program-Files werden direkt geladen, sofern sie nicht länger als 47 Sektoren sind. DIS/FIX-Files werden immer nur satzweise geladen.

Unterhalb der Informationszone erscheinen die Standard-Adressen der Loader für die drei Module, mit denen DISKASSEMBLER geladen werden kann. Sofern das File nicht adreßgebunden ist, setzt DISKASSEMBLER einen Default-Wert für LOAD AT ~~00000000~~ an.

Werte, die größer als 16.000 betragen, kann DISKASSEMBLER nicht akzeptieren. Längen manipuliert werden können (soweit, wie UG-SPEZIFIKATION zulässt!), was aber nicht immer sinnvoll ist. Die Vorgabewerte können mit ENTER akzeptiert werden, FCTN 6 führt zum folgenden Bildschirm.

BILDSCHIRM 2 - BLOCK-DISASSEMBLIERUNG

Hier können Sie innerhalb des Adreßbereiches des Files oder Speicherbereichs bestimmte Zonen als Daten- bzw. Textblöcke deklarieren, die nicht explizit disassembliert werden, sondern mit vorangestelltem 'TEXT' oder 'DATA' erscheinen. Das spart enorm Platz auf der Diskette, die das Ausgabefile enthalten soll und beschleunigt die Disassembly. Eingegeben werden Start- und Endadresse eines jeden Blocks, die Endadresse wiederum nicht inclusive!

Mit ENTER gelangt man von Feld zu Feld, FCTN 6 führt zum dritten Bildschirm.

BILDSCHIRM 3 - SUBROUTINEN MIT DATAS

Eine recht elegante Art der Parameterübergabe bei TMS9900

Maschinencode ist die, hinter einem Subroutinenaufruf mehrere Datenworte als Konstanten zu übergeben, auf die dann mittels Register 11 oder 14, je nach Aufruf, zugegriffen werden kann. Diese Daten können einen Disassembler recht ordentlich durcheinander bringen, weshalb DISKASSEMBLER darauf achtet, diese

DATAs zu übergehen.

Dazu muß man sie ihm jedoch erst angeben. Dies wird i.A. beim ersten Durchlauf noch nicht bekannt sein, doch dient dieser erste Lauf ohnehin nur dem Überblick, zudem kann er ggf. wiederholt werden.

Eingegeben werden kann die Adresse der Subroutine und die Anzahl der DATAs, die dem Aufruf folgen. Achtung: Anzahl rechtsbündig schreiben!

FCTN 6 führt auch hier weiter, zum vierten Bildschirm.

BILDSCHIRM 4 - AUSGABEOPTIONEN

Dies ist die letzte Station, bevor DISKASSEMBLER mit der Arbeit beginnt. Die hier erlaubten Tastenfunktionen werden angezeigt. FCTN 7 (Select Printer Output) aktiviert die Druckerausgabe. Vorher wird noch der Name des Druckers abgefragt, mit dem Default-Wert aus DISKCONFIG. Wird hier der Name eines Diskfiles angegeben, so wird ein einziges File erzeugt, im Gegensatz zu FCTN 8 (Select Output Device), wo die Filelänge überwacht wird und mehrere, maximal 65 Sektoren lange Abschnitte erzeugt werden. Es empfiehlt sich, als Filename des Output Device einen Namen mit einem 'A' am Ende zu wählen. DISKASSEMBLER inkrementiert bei jedem Öffnen eines neuen Files den letzten Buchstaben, was beim Start mit 'A' immerhin 26 vernünftige Namen erzeugt. Hier darf kein Druckername eingegeben werden, da dann die Namensänderung ungültige Druckernamen erzeugen würde!

Die gute alte Quit-Taste ist hier entschärft und schaltet nur die

DER OUTPUT VON DISKASSEMBLER

Angenommen, Sie haben diese Anleitung bis hier hin befolgt. Dann beginnt DISKASSEMBLER mit dem ersten Durchlauf (Pass) seiner Untersuchungen. In diesem wie auch im zweiten Pass besteht der Output aus 6 Spalten.

Ganz links stehen die Adressen, denen jedoch 'LN' vorangestellt wurde. Damit sind diese Adressen gültige Labels für den Assembler des EA-Moduls. Ein derart aufgebautes File kann prinzipiell direkt anschließend wieder assembled werden. Da jedoch jede Zeile ein Label erhält, wird die Symboltafel des Assemblers sehr schnell überlaufen. Das Verfahren ist also nur bei kurzen Abschnitten sinnvoll.

In der zweiten Spalte von links stehen die Op-Codes (Mnemonics), welche die von der CPU auszuführenden Befehle darstellen. Je nach Option 'T' (an oder aus) folgt die ASCII- oder DATA-Interpretation der zweiten Spalte, also entweder Spalte 3 DATA, Spalte 4 TEXT oder Spalte 3 TEXT und Spalte 4 DATA. Spalte 5 (zweite von rechts) zeigt an, wieviele relocatable Worte sich in der Anzeigzeile befinden (RL1 oder RL2). Diese Information ist nur bei eingehenden Änderungen im untersuchten Programm von Bedeutung und deutet auf Label-Bezüge innerhalb relocatibler Segmente hin. Ganz rechts steht in Spalte 6 die laufende Adresse, bezogen auf den Anfang des Files/Speicherbereichs.

Während nun diese Informationen an Ihnen vorbeilaufen, stehen folgende Funktionen zur Verfügung:

FCTN 1,2 schalten die Option 'T' vorübergehend aus und an, übersteuern also diese Option während der Ausgabe. FCTN 4 stoppt die Ausgabe in Richtung Drucker (Off-Line-Cancel), FCTN 5 schaltet um

zwischen der Anzeige von Spalten 1 bis 3 und den Spalten 4 bis 6. FCTN 7 erlaubt das zwischenzeitliche Schließen bzw. Öffnen des Druckerkanals, so daß bestimmte Segmente nicht ausgedruckt werden. FCTN 8 macht dies analog mit dem Ausgabefile. FCTN 9 bricht die Arbeit ab, wobei alle Files geschlossen werden und zu Bildschirm 1 gesprungen wird. FCTN 0 erlaubt es, quasi mittendrin den momentan angezeigten Bildschirm auszudrucken oder abzuspeichern (Ausgabemedium wie FCTN 7!). Erscheint die Meldung 'Screen printed', so kann durch Druck auf ENTER die Ausgabe auf dem nun zugeschalteten Gerät fortgesetzt werden; FCTN 0 erneut gedrückt schließt die Ausgabe wieder. FCTN = schaltet wie angesprochen das Arbeitstempo um und CTRL 1 erlaubt die Einstellung der Bildschirmfarben. Den ersten Pass schickt man am besten über den Drucker.

DER ZWEITE DURCHLAUF

Wurde der erste Durchlauf normal beendet (kein Abbruch durch FCTN 9), so wird nach Druck auf ENTER der zweite Pass aktiviert. Zuerst sieht es so aus, als ginge alles nochmal von vorne los, denn die 4 Arbeitsbildschirme werden erneut durchlaufen. Doch wird das File sofort wieder geladen. Auch alle Einstellungen auf Bildschirm 2 und 3 sind erhalten geblieben. Sie können diese jedoch ändern, sollte sich beim ersten Pass irgendeine Änderung als nötig erwiesen haben. Auch die Optionen können noch geändert werden. Normalerweise reicht es aber, 3 mal FCTN 6 zu drücken. Nach Spezifikation von Drucker und Ausgabefile beginnt der zweite Pass. Nun tauchen in Spalte 1 und 2 Labels auf, immer mit 2 Buchstaben oder ein Buchstabe und eine Zahl. Diese Namen werden von vorne nach hinten vergeben, d.h. sie laufen in der Reihenfolge, in der die Referenzen auf die Labels auftauchen. Damit sieht der Output schon fast wie ein richtiger Source-Code aus, bis eben auf die Namen der Labels. Der zweite Pass eignet sich für die Ausgabe in ein Diskettenfile.

EINIGE TIPS ZUM ARBEITEN MIT DISKASSEMBLER

Ein Programm wie DISKASSEMBLER stellt verschiedene Anforderungen. WER964A Benutzt TIEFLIEBENFEBROREREE groß NÄMMRVOß anderen AutBEMME zu 10 sehen, der kommt mit einem guten Grundlagenwissen in Assembler aus. Wer aber das analysierte Programm in eine Form bringen will, daß eigene Ideen eingebaut werden können, der muß fortgeschrittene Assemblerkenntnisse besitzen. Um z.B. DATAs hinter BL und BLNP korrekt zu zählen, muß erkannt werden, welche Befehle denn wirklich unsinnig sind, wie oft über R14 oder R11 mit Autoinkrement zugegriffen wird und wieviele Byte-Zugriffe dabei sind. Dann kann es passieren, daß Konstanten mit Werten, die rein zufällig im Adressbereich des Programms liegen, als Labels erkannt werden. Wird das nicht berücksichtigt, ist ein solches Programm nach Änderungen in der Regel nicht mehr lauffähig, auch wenn es fehlerfrei assembled wurde. Für all das können keine Kochrezepte gegeben werden - hier muß die Kombinationsfähigkeit und Erfahrung eingesetzt werden.

Ein Tip ist, den ersten Pass immer auf den Drucker zu schicken, als Referenz, und den zweiten Pass auf Diskette abzulegen. Zwar

kann die Größe eines Gryte-Programmes bis zu über 1000 Sekturen belegen, doch kann der Output so reduziert werden (Spalten 3 bis 6 sind meist überflüssig), daß der Umfang auf etwa ein Drittel zurückgeht. Aber eben erst hinterher.

Man sollte immer einige formatierte Disketten bereithalten,

besonders dann, wenn man nur über das nicht mehr zeitgemäße 90/180Byte-Format des TI-Diskcontrollers mit DOS-40 verfügt.

Zudem ist es unbedingt empfehlenswert, zwei Laufwerke zu besitzen. Ist nur eines vorhanden, so muß sich ein eventuelles Ausgabefile auf der gleichen Diskette wie das zu untersuchende Programm befinden, was einen Diskettenwechsel dann, wenn ein DIS/FIX 80 File untersucht wird, unmöglich macht.

PROBLEME UND FEHLER

Nichts und niemand ist perfekt, auch DISKASSEMBLER nicht. So sieht man z.B. beim Output auf dem Schirm nur 76 der 78 Zeichen einer Zeile. Da pro Bildschirm nur 38 Spalten angezeigt werden, unterschlägt die Anzeige 2 Spalten in der Bildmitte. Ein, angesichts der sonstigen Professionalität des Programms, unverständliches Eingeständnis an schlecht eingestellte Monitore.

Ganz schlimm ist die Unsitte, bei einem Abbruch des ersten Laufes die Eintragungen in den Bildschirmen zwei und drei zu löschen. Es passiert doch gelegentlich, daß man Daten- oder Textblöcke Übersicht oder erst später merkt, welche Subroutinen wieviele DATAs benötigen. Will man einen vernünftigen Output erhalten, dann darf man immer wieder alles neu eingeben.

Daß man externe Referenzen selbst auflösen muß, ist einerseits verständlich (Speicherplatz), andererseits aber vermeidbar. Da die Symboltafel von DISKASSEMBLER über 900 Einträge verkraftet, wäre für eine Referenzkette auch noch Platz gewesen, gegebenenfalls als Option. Wer einmal ein langes File mit solchen Referenzen bearbeiten mußte, wird wissen, was dies an Arbeitserleichterung gebracht hätte.

Und dann ist da noch die mangelnde Übersicht, die DISKASSEMBLER bei der Loader-Erkennung von P-Files zeigt. Oft erkennt er einen Standard-Loader, den er nach dem Manual als Custom ansehen müßte. In diesen, nicht so seltenen Fällen unterschlägt er die ersten, oft wichtigen 3 Worte!

ANPASSUNG AN DAS CPS99

Auch dieser Patch bewirkt, daß der Loader keinen TI-Controller mehr erkennt. Daher sollten Sie vorher eine Kopie des Loaders DISKASSEM anfertigen (alle 3 Files lassen sich kopieren) und diese Kopie sicher aufbewahren. Der Eingriff ist recht einfach, da es nur einen Loader für alle Module gibt. Leider muß diese Änderung mit einem Disketten-Editor vorgenommen werden, da dieses File Steuercodes enthält, die abgeprüft werden und ein Reassemblieren nach Disassembly durch DISKASSEMBLER unmöglich machen. Folgende Angaben setzen die ORIGINALDISKETTE voraus!

Ändern Sie im Sektor 35 die Bytes 131 bis 134 von 1312 in 1000 und Byte 140 von 7 in F. Das ist alles. Hierzu empfiehlt sich DIAGS im ASCII-Modus.

BEZUGSQUELLE

DISKASSEMBLER kann, wie alle MG-Programme, bei LCC, Legio Computer

Centre, Haagweg 169, 2201 AG Rijswijk, Niederlande bearbeitet werden. Telefon 070-995757. Eine aktuelle Preisliste lag bei Redaktionsschluß nicht vor.

SCHLUSSWORT

Diese Reihe von drei Softwarevorstellungen hatte verschiedene Ziele. Ein Hauptziel war sicherlich, diese außergewöhnlichen Programme einer größeren Zahl von TI-Anwendern in angemessener Form vorzustellen, auch um zu zeigen, daß es für den TI Software gibt, die man gemeinhin dem PC-Bereich zuordnen könnte. Ebenso sollte jeweils eine Art deutsches Handbuch entstehen, das die vielfältigen Probleme, die beim Lesen der englischsprachigen Manuals auftreten können, wenigstens teilweise beseitigen hilft. Zudem war es ein Experiment, das zeigen sollte, welche Resonanz eine sehr ausführliche Softwarebesprechung ohne jeden graphischen Schnick-Schnack erhält. Doch in der Hauptsache sollte diese Reihe zeigen, daß in dem TI noch jede Menge ungenutzter Möglichkeiten stecken.

Natürlich können diese drei Beiträge kein Handbuch ersetzen - so war es auch nicht geplant. Aber wenn sie dem einen oder anderen einen Zugang zu DIABS, EXPLORER oder DISASSEMBLER verschafften, so waren sie nicht umsonst!

Christopher Winter (system-99 User-Group)

März 1988