



Wettstreit der Assembler

64'er
Test

Wer den C 64 bis aufs letzte ausreizen möchte, muß ihn leider in Maschinensprache programmieren. Wir testen das nötige Handwerkszeug, zwei der besten Assembler. Sie sind die idealen Werkzeuge für Programmier-Profis.

In den letzten Monaten sind die Assembler in der 64'er etwas zu kurz gekommen. Es hat sich auch nicht viel auf dem Markt getan. Das einzige größere Ereignis war das Erscheinen von TurboAss, der von den Entwicklern als der schnellste Assembler für den C 64 bezeichnet wird. TurboAss hat sich in den letzten Monaten glänzend bewährt, während ein »Oldie« die Bühne der Assemblerwelt wieder betreten hat. Dabei ist das Wort Oldie in keiner Weise abfällig gemeint, denn schon im 64'er Ausgabe 1/85 konnte ASSI/M ein hervorragendes Testergebnis erzielen. Im Laufe der Monate sind an ASSI/M einige Veränderun-

gen vorgenommen worden, so daß jetzt eine erheblich verbesserte Version vorliegt, die sicherlich einiges verspricht.

Eines gleich vorweg: Wir können die beiden Assembler nicht direkt miteinander vergleichen und einen Testsieger ermitteln. Dazu sind beide Assembler von ihrem Konzept her zu unterschiedlich aufgebaut. TurboAss ist ein auf Geschwindigkeit und Speicherbedarf hin optimierter Assembler, der meistens ohne jeden Diskettenzugriff auskommt. Bei kleineren Programmen sind die Assemblierungszeiten derart kurz, daß man sich fragt, ob überhaupt assembliert wurde. Auch längere Pro-

gramme zu assemblieren, ist eher eine Frage von Sekunden als Minuten. Ein anders Konzept verfolgt ASSI, das Flaggschiff der Assembler. Er arbeitet rein diskettenorientiert und ist dadurch zwangsläufig um ein Vielfaches langsamer als TurboAss. Doch das Hauptargument für ASSI/M ist nicht seine Geschwindigkeit sondern seine Vielzahl von gut durchdachten Assembler-Anweisungen, die das Programmieren erleichtern und übersichtlicher machen. Deswegen muß jeder Anwender und Programmierer für sich selber entscheiden, welches Programm seinen Anforderungen besser angepaßt ist.

Beide Assembler werden zusammen mit einem Editor ausgeliefert, so daß der normale Basic-Editor des C 64 nicht verwendet werden muß. Beide sind sogenannte Full-Screen-Editoren: Der eingegebene Quelltext kann mit den Cursor-Tasten nach oben und unten gescrollt werden, Änderungen im Quelltext werden ausgeführt, ohne daß man <RETURN> drücken muß.

Beide Editoren bieten derart viele ähnliche Funktionen, daß selbst eine Kurzbeschreibung mehrere Seiten benötigt. Wir können hier aus Platzgründen leider keine kompletten Befehlslisten drucken — es gibt einfach zu viele Anweisungen. Es fehlt

keine wichtige Funktion, praktisch alle Wünsche des Benutzers werden erfüllt. Deswegen sind sich die Editoren auch in mancher Hinsicht ähnlich. Ihr größter Unterschied liegt jedoch in der Speicherung des Quelltextes, denn hier zeigt sich auch das unterschiedliche Konzept der beiden Assembler. TurboAss hält aus Geschwindigkeitsgründen Editor, Assembler, Quelltext und den Objectcode gleichzeitig im Speicher des C 64. Der Quelltext wird deswegen in gepackter Form gespeichert, da sonst zu wenig Platz für den Objectcode wäre. Also wird jede Zeile Quelltext bei der Eingabe »tokenisiert«, das heißt, daß die Zeile gleich auf korrekte Syntax überprüft und dann in einen kompakten Code übersetzt wird. Dieser Code benötigt wesentlich weniger Speicher als die ASCII-Codes der Zeile und kann vom Assembler auch um ein Vielfaches schneller abgearbeitet werden. Anders verhält es sich beim ASSI/M: Editor und Assembler sind zwei völlig voneinander getrennte Programme. Der Quelltext wird mit dem Editor geschrieben und dann auf Diskette gespeichert, der Assembler holt sich den Quelltext während der Assemblierung von der Diskette. Dadurch kann der Quelltext praktisch beliebig lang werden. Der Editor mit dem Namen FSE übersetzt den Quelltext also nicht in einen Code, sondern behandelt ihn als ganz normales ASCII-File. Das hat den Vorteil, daß der FSE universell einsetzbar ist. So kann man ihn auch im beschränkten Maße zur Textverarbeitung verwenden. Der TurboAss-Editor weist dagegen alle für ihn unsinnigen Zeilen mit einer Fehlermeldung zurück. Das hat natürlich den Vorteil, daß beim eigentlichen Assembliervorgang keine Syntaxfehler auftauchen können, da diese ja schon bei der Quelltexteingabe eliminiert werden. Beim ASSI kann man sich schon mal, ohne es zu merken, vertippen und muß dann nach dem Assembliervorgang den Editor laden, den Quelltext laden, ändern und speichern, den Assembler laden und die As-

semblierung erneut starten. Bei der langsamen 1541 ist das sicher kein Vergnügen.

TurboAss und ASSI/M sind Zwei-Pass-Makro-Assembler. Sie bieten derart viele Funktionen, daß mancher Anfänger recht wenig mit ihnen anfangen kann. Selbst Profis werden die Assembler in den seltensten Fällen voll ausschöpfen können. Damit Sie einen Überblick über die speziellen Befehle der Assembler erhalten, haben wir in Tabelle 1 die Anweisungen der beiden zusammengefaßt. Eine Erklärung einzelner Spezialbegriffe, die bei anderen Assemblern meist nicht auftauchen, finden Sie im Textkasten. Im weiteren wollen wir uns nur auf die Merkmale beschränken, die die beiden aus der breiten Masse der Assembler herausheben.

Von Label, Makros, und Symbolen

Beide Assembler erlauben die Übergabe textueller Parameter in Makros. Ein Beispiel: Sie wollen in einem Makro einen Sprung ausführen, der allerdings vom Makroaufruf abhängig sein soll. Mit anderen Worten: Mal soll im Makro BEQ stehen, mal BNE und mal BVC. Das Makro kann so programmiert werden, daß der Sprungbefehl erst noch durch einen Parameter ergänzt werden muß. Als Parameter wird dann entsprechend »EQ«, »NE« oder »VC« übergeben, der das »B« im Makro zum entsprechenden Befehl vervollständigt. Diese komfortable Art der Makroprogrammierung findet man nur bei sehr wenigen Assemblern.

ASSI/M hat in diesem Bereich aber noch ein paar Funktionen, die TurboAss nicht kennt: Ein Parameter kann, wie ein String in Basic, vor der Assemblierung des Makros noch analysiert werden. Als nicht gerade sinnvolles, aber dafür umso interessanteres Beispiel befindet sich auf der ASSI-Diskette ein Makro, das arithmetische Ausdrücke compilieren kann. Bei Übergabe eines Parameters wie »4+12x23-7« wird ein Objectcode erzeugt, der genau diese Rechenoperation ausführt.

Um diese Makroprogrammierung aber sinnvoll anwenden zu können, muß man sich schon näher mit Maschinensprache-Programmierung beschäftigen haben.

Kurzes Fazit: Beide Assembler haben alle Standardanweisungen und dazu noch ein ganzes Paket an Extras, die andere Assembler nicht bieten. ASSI erscheint uns dabei noch komfortabler, da er einige Assemblerfunktionen mehr als TurboAss bietet, was man allerdings mit längeren Assemblierzeiten hinnehmen muß.

Beide Pakete werden zusammen mit einem Monitor ausgeliefert. Bei TurboAss ist es der X-MON, ein recht leistungsfähiger Monitor mit allen Standardbefehlen und eingebautem Diskmonitor. Leider fehlen beim X-MON wichtige Debugging-Befehle wie beispielsweise Trace. Dafür erhält man zum X-MON den kompletten Quelltext, so daß der Benutzer ihn an eigene Bedürfnisse anpassen und beliebig erweitern kann.

Zusammen mit ASSI erhält man den Demon, einen Monitor, der auch alle Standardbefehle bietet, darüberhinaus aber auf das »Debuggen« von Programmen abgerichtet ist. So bietet er verschiedene Trace-Modi und komfortable Breakpoint-Abfragen. Besonders interessant ist der Backtrace-Modus, der ein Programm schrittweise abarbeitet und in einem weiteren Speicherbereich eine Aufzeichnung der durchgeführten Schritte erstellt. Diese Aufzeichnung kann man dann später in Ruhe durchsehen und analysieren. Beim Tracen kann man sich übrigens auf dem Bildschirm die aktuellen Adressen, die oberen Bytes des Stacks und, vom Benutzer bestimmbare Adressen ständig anzeigen lassen, sodaß man stets volle Kontrolle über seine Programme hat. Ebenso komfortabel sind die User-Breakpoints, mit denen der Programmierer einen Testlauf machen kann, der unter definierten Bedingungen abgebrochen werden kann. So kann man testen, warum beispielsweise bestimmte Daten zerstört werden oder wieso eine Routine

in einer Endlosschleife endet. Demon bietet noch viele weitere Analyse-Möglichkeiten, die ihn zum besten Debugging-Monitor machen, den wir kennen.

Mit Editor, Assembler und Monitor gaben sich die Autoren der Programme aber nicht zufrieden. Auf den jeweiligen Disketten befinden sich noch andere nützliche Programme und Routinen.

Bei TurboAss erhält man beispielsweise einen leistungsfähigen Re-Assembler mitgeliefert. Er kann in über 90 Prozent der Fälle korrekt zwischen Datenbytes und echtem Programm unterscheiden. Zusammen mit einem Name-Replacer, der aus einem re-assemblierten Quelltext wichtige Speicherstellen erkennt und diese gegen vernünftige Label austauscht, ist dieser Re-Assembler das ideale Werkzeug zur Analyse von Programmen, zu denen man keinen Quelltext auftreiben kann. Leider ist beim ASSI/M kein solcher Re-Assembler enthalten.

Dafür gibt es beim ASSI/M umsonst eine Spezialversion, die Editor, Assembler und Monitor gekoppelt in einem Programm enthält. Diese hat aber die Tücke, das maximal bis zu 4 KByte Objectcode assembliert werden können. Dafür steigt die Arbeitsgeschwindigkeit aber gewaltig, da sämtliche Nachladezeiten von Diskette entfallen. Für kleinere Projekte ist die gekoppelte Version durchaus zu empfehlen.

Einer der großen Pluspunkte von ASSI sind aber die mitgelieferten Makro-Bibliotheken. Diese Bibliotheken sind eine wahre Fundgrube für jeden ambitionierten Programmierer, da er einerseits viele Tips und Tricks für den Einsatz von Makros erhält, andererseits aber auch wirklich sinnvolle Werkzeuge in die Hand bekommt. Immer noch ungeschlagen sind die Makros zur strukturierten Programmierung. Mit dieser Bibliothek kann man in Assembler mit Pascal-ähnlichen Strukturen wie REPEAT.UNTIL oder IF.ELSE.ENDIF programmieren, was einerseits die Übersicht über die Programme erhöht, andererseits aber auch das Program-

mieren sehr vereinfacht. Weitere Bibliotheken befassen sich mit der 16-Bit-Arithmetik und Input/Output-Routinen.

Sehr wichtig: Beide Programme sind nicht kopiergeschützt. Kopien des TurboAss funktionieren allerdings nur mit einem mitgelieferten Schutzstecker (Dongle).

ASSI/M ist nach dem »Of-

fenen Konzept« programmiert und dokumentiert worden. Für alle drei Programme werden wichtige Adreßbereiche kommentiert angegeben, so daß der Benutzer Assembler und Editor in Grenzen und den Monitor sogar komplett verändern und erweitern kann.

ASSI und TurboAss werden mit recht ausführlichen

und verständlichen Handbüchern ausgeliefert. Sie setzen allerdings schon gute Kenntnisse der Maschinensprache-Programmierung voraus.

Einige Leser wird sicherlich interessieren, daß es den ASSI/M für fast alle Commodore-Computer, vom VC 20 bis zum CBM 8000 gibt. Nähere Informationen

über die anderen Versionen erhalten Sie beim Autor, (Adresse siehe Tabelle 2).

Beide Assembler-Systeme haben sich schon viele Male im Praxis-Einsatz bewährt, so wurden die aktuellen Versionen mit Hilfe ihrer älteren Vorgänger programmiert. Wir können beide allen ernsthaften Programmierern empfehlen. (bs)

Kleines Assembler-Lexikon: TurboAss- und ASSI/M-Besonderheiten

Block — Ein Block ist ein lokaler Abschnitt eines Quelltextes. Alle Label und Symbole, die innerhalb eines Blocks definiert werden, sind außerhalb dieses Blocks unbekannt. Ein und dasselbe Label kann also, je nach Block, verschiedene Adressen kennzeichnen.

Bedingte Assemblierung — Ein Teil des Quelltextes wird nur dann assembliert, wenn bestimmte Bedingungen erfüllt sind. Sinnvoll insbesondere bei Makros, die sich selber aufrufen und bei Programmen, die während des Assemblierens an bestimmte Systeme angepaßt werden sollen.

Interaktive Assemblierung — Möglichkeit der Eingabe von Werten während des Assembliervorgangs. Der Quelltext kann also ohne Editor nachträglich vom Benutzer in Grenzen geändert werden. Sinnvoll, wenn ein und dieselbe Routine mehrere Male kurz hintereinander mit kleinen Änderungen assembliert werden soll.

Assemblerschleifen — Ein und dieselbe Befehlsfolge muß mehrere Male hintereinander im Objectcode stehen. Die Schleife assembliert den entsprechenden Quelltext also mehrmals hintereinander. Nützlich beispielsweise für schnelle Busroutinen.

Variable — Label, die immer neue Werte zugewiesen bekommen können. Nützlich beispielsweise als Zähler bei Makros, die sich selber aufrufen. (bs)

Tabellen	TurboAss	ASSI/M
Einzelne Bytes	.byte	.byte
ASCII-Texte	.text	.text
	.null	
	.shift	
Bildschirmcode	—	.scrn
16-Bit-Worte	.word	.word
		.dbyte
Rücksprungaddr.	.rta	—
Label/Symbol-Befehle		
Label umdefinieren	—	.reset
Variable definieren	.var	.var
Variableninh. ändern	.var	.set
Symboltabelle speichern	(im Editor)	.puts
Symboltabelle laden	—	.gets
Makro-Befehle		
Start Makro-Definition	.macro	.macro
Start Segment-Def.	.segment	.segment
Ende der Definition	.endm	.mend
Analyse der Makro-Param.	—	&len, &cmp, &instr
Bedingte Assemblierung		
Bedingungen:		
ungleich Null	.if	.if, .ifne
gleich null	.ifeq	.ifeq
positiv	.ifpl	.ifpl
negativ	.ifmi	.ifmi
Assembler-Pass 1	—	.ifpl
Pass 2	—	.ifp2
Ende If-Anweisung	.endif	.cend
Umdrehen der Bedingung	—	.else
Interaktive Assemblierung		
Ausgabe eines Textes	—	.prnt
Eingabe eines Symbolwertes	—	.inpt
Assemblerschleifen		
Start der Schleife	.lbl	—
Rücksprung	.goto	—
Listing-Ausgabe		
Listing ausgeben	.pron	.pron
Listing nicht ausgeben	.proff	.prof
Symboltabelle ausgeben	(im Editor)	.syms
Symbolt. nicht ausgeben	—	.nosy
Makros in voller Länge ausg.	.showmac	.xpn
nur Makroaufruf ausg.	.hidemac	.noxp
Leerzeichen ausg.	—	.skip
Seitenvorschub	—	.page
Seitenlänge definieren	—	.plen
Sonstiges		
Block öffnen	.block	.blck
Block schließen	.bend	.bend
Assembl. beenden	.end	.end
Offset	.offs	.offs
mit neuem File weiterassemblieren	—	.file
Sourcecode von Disk einf.	.include	.lib
Objectcode von Disk einf.	—	.link
neues Objektfile beginnen	—	.nobj
eigene Fehlermeldung ausg.	—	.perr
Umschaltung CBM-ASCII — »echtes« ASCII	—	.cset
Standard-Zahlenbasis setzen	—	.base
Prüfsumme Objektcode berechn.	—	.csum
ASSI-spezifische Syntax-Schalter		.mold, .mnew
		.bold, .bnew

Tabelle 1. Gegenüberstellung der Assembler-Anweisungen

	TurboAss	ASSI/M
Editor	Full-Screen-Editor mit eingebautem Tokeniser — nur für TurboAss-Sourcefiles verwendbar (in Assembler integriert)	Full-Screen-Editor (FSE) für alle ASCII-Dateien.
Monitor	Mitgeliefert: X-MON, Monitor mit Standard-Funktionen, kompletter Quelltext zum Erweitern	Mitgeliefert: Demon, Monitor mit allen Standard-Funktionen und sehr guten Debugging-Funktionen
Reassembler	Omikron-Reass mit automatischer Tabellen-Erkennung.	kein Reassembler im Lieferumfang enthalten
Quelltexte	einige Demos, kompletter Quelltext zum X-MON	ein Makro-Demo, mehrere sehr nützliche Makro-Bibliotheken für strukt. Programmieren, 16-Bit-Arithmetik und Input/Output-Funktionen
Handbuch	Handbuch mit Ergänzung, zusammen etwa 70 Seiten	mehrere Handbücher (Editor, Ass., Monitor, Quelltexte) zusammen etwa 100 Seiten, Reference Card (FSE)
Preis	139 Mark inkl. Dongle (Schutzstecker) und Reset-Taster	220 Mark (normal) 250 Mark (inkl. Editor-Erweiterung FSX)
Bezugsquelle	Omikron Software Erlachstr. 15 7534 Birkenfeld 2	Dirk Zabel Stresemannstr. 50 1000 Berlin 61

Tabelle 2. Das ist jeweils im Lieferumfang enthalten