Assembler im Test

Teil 2

Bevor wir auf die vier diesmal getesteten Assembler zu sprechen kommen, vorher noch ein paar Bemerkungen über die Zielgruppe dieser Programm-Pakete.

Die in der letzten Ausgabe vorgestellten Assembler kann man sowohl vom Preis als auch von den Möglichkeiten her zu den professionellen Programmentwicklungssystemen zählen (vielleicht mit Ausnahme des T.EX.AS., der als Lehrsystem bezeichnet wird). In ihnen sind Funktionen enthalten, die selbst fortgeschrittene (Assembler-)Programmierer erst nach relativ langer Zeit und nach andauernder Benutzung beherrschen und anwenden. Dann allerdings möchte man auf den gebotenen Komfort und die mächtigen Befehle (zum Beispiel Makro-Bibliotheken, bedingte Assemblierung, etc.) nicht mehr verzichten.

Assembler, wie sie in dieser Ausgabe vorgestellt werden, wenden sich mehr an den Assembler-Anfänger und Gelegenheitsprogrammierer. Das heißt jedoch nicht, daß mit ihnen nicht auch vernünftig gearbeitet werden könnte. Für kleinere Projekte (klein bezieht sich nicht nur auf den Umfang, sondern auch auf die Komplexität) reichen diese Assembler nicht nur aus, sondern erfüllen durchaus ihren Zweck. Da es keine unnötigen Funktionen gibt, braucht man sich nicht mit einem Ballast von Kommandos herumzuschlagen. Die Geschwindigkeit ist (abgesehen vielleicht vom Mastercode-Assembler) völlig ausreichend. Mit Ausnahme vom Profimat-System sind in den anderen drei Programmen Editor, Monitor und Assembler zusammengefaßt, so daß Maschinensprache-Programme in einem Arbeitsgang geschrieben, assembliert und getestet werden können. Auch das ist ein Vorteil. Nun ist es aber nicht so, daß ein Assembler unbedingt notwendig zum Programmieren in Maschinensprache ist. Ein guter Monitor, wie zum Beispiel der SMON, stellt fast alle Hilfsmittel zur Verfügung (sehr wichtig zum Beispiel: Label). So wurde unter anderem das Programm HYPRA-LOAD aus Ausgabe 10/84 in monatelanger Arbeit, aber lediglich mit einem Monitor erstellt. Ein Assembler hat jedoch den Vorteil, daß ein Einfügen von Programmschritten einfacher und schneller möglich ist. Auch hat man die Möglichkeit, schon während der Programmierung Kommentare einzufügen, so daß man auch nach längerer Zeit seinen Entwurf noch verstehen und nachvollziehen kann. Nun aber zu den einzelnen Assemblern.

Mastercode

Mastercode ist ein kombinierter Editor/Assembler/Monitor und wird vom Verlag Markt & Technik AG vertrieben. Er ist auf Diskette und Kassette lieferbar.

Editor

Eine Äußerlichkeit hebt Mastercode von allen anderen Assemblern ab: Alle Funktionen werden über Menüs und Untermenüs abgewickelt. Dies ist gerade für einen Anfänger sinnvoll, der sich nicht eine Unmenge von Befehlen merken möchte.

Im Editor stehen die Unterpunkte Eingabe, Auflisten, Löschen, Umnumerieren, Speichern und Laden zur Verfügung. Dieser Editor ist völlig eigenständig, das heißt vom Basic-Editor unabhängig. Er arbeitet allerdings streng zeilenorientiert, so daß jede eingegebene Zeile mit einer Nummer beginnen muß. Man kann nicht, wie beim Editieren von Basic-Programmen, mit dem Cursor rauf und runter fahren.

Es lassen sich schon existente Bytefolgen in den Quelltext übernehmen, sie werden dann mit einem Pseudo-Opcode für Tabellen versehen. Dies ist aber nur nützlich bei frei verschieblichen Programmteilen, oder eben Tabellen, weil sonst alle Adressen von Hand angepaßt werden müßten.

Interessant ist noch, daß beim Laden von Quelltexten immer ein MERGE durchgeführt wird. Dadurch lassen sich recht einfach zwei Quelltextdateien mischen.

Der nutzbare Speicher für den Quelltext ist enttäuschend knapp gehalten. Der maximale Speicherbereich für Quelltexte beträgt nämlich 15 KByte, das entspricht maximal 1000 Zeilen Assemblerquelltext, Kommentare nicht eingerechnet. Selbst diese Zahl ist bei der Verwendung von vielen Labeln nicht erreichbar. Das wäre ja alles nicht so schlimm, wenn es möglich wäre, Quelltextfiles miteinander zu verketten. Ein nur etwas längerer Quelltext muß dann jedoch in zwei Einzeltexte aufgeteilt werden, wobei im zweiten sämtliche benötigten Label noch einmal definiert werden müssen, da beide Teile nur unabhängig voneinander assembliert werden können.

Auch die geringe Verarbeitungsgeschwindigkeit macht die Arbeit nicht gerade zum Vergnügen.

Assembler

Label dürfen bis zu sechs Zeichen lang sein und können auch, wenn man sie als Variable, das heißt als Zwischenspeicher benutzt, erst nach dem Gebrauch definiert werden. Es dürfen maximal 100 verschiedene Label im Quelltext auftreten.

Die Ausgabe der Symboltabelle, also die Auflistung sämtlicher verwendeter Label, kann nach Beendigung des Assemblierens erfolgen.

Mastercode unterstützt die Zahlensysteme binär, oktal, dezimal und hexadezimal sowie die Eingabe im ASCII-Code. Rechnungen dürfen im Quelltext in den vier Grundrechnungsarten durchgeführt werden. Tabellen lassen sich entweder ganz normal als Byte-für-Byte-Tabellen oder als Adreßtabellen aufbauen.

Es ist möglich, den Objectcode direkt in den Speicher oder auf ein Peripheriegerät (Kassette, Floppy) auszugeben oder den Quelltext einfach nur auf Fehler zu untersuchen. Auf

Im zweiten Teil unseres Assembler-Tests kommen die Assembler unter 100 Mark an die Reihe. Wir stellen folgende Produkte vor: Mastercode-Assembler, Profimat, Profisoft und Maschine 64. Obwohl sie in der unteren Preisklasse angesiedelt sind, können sie eine wertvolle Hilfe beim Einstieg in die Maschinensprache sein.

dem Drucker erhält man im Anschluß an die Assemblierung ein sauber formatiertes Quelltext/Objektcode-Listing.

Als letztes sei noch erwähnt, daß Kommentare nur für sich alleine in Quelltextzeilen stehen dürfen und nicht, wie sonst üblich, auch durch Semikolon getrennt hinter einem Assembler-Befehl.

Monitor

Mastercode hält nur die Minimalfunktionen eines Monitors bereit. Dazu gehören das Disassemblieren, der Hexdump und das Verändern von Speicherbereichen, das Starten von Maschinenprogrammen, das Laden und Speichern von Objektcode sowie ein einfacher Einzelschrittbetrieb zum Austesten von Maschinenprogrammen.

Bei einem Hexdump macht es anfangs richtig Spaß, zuzusehen, wie die einzelnen Bytes schön nacheinander auf dem Schirm erscheinen und schon nach einigen Sekunden ein paar Zeilen auf dem Bildschirm stehen. Aber schon bald beginnt die extrem geringe Geschwindigkeit aller Monitorfunktionen zu stören, da man nicht jedesmal minutenlang auf ein Disassemblerlisting oder sonstiges warten möchte. Ein kurzer Blick mit Mastercode auf Mastercode bestätigte dann den Verdacht, daß man es hier mit compiliertem Basic zu tun hat, das es wohl kaum mit irgendeinem anderen Monitor, der in Maschinensprache geschrieben wurde, aufnehmen kann.

Dokumentation

Als »Handbuch« erhält man knappe 50 Seiten im augenermüdenden Compactkassettenformat, was wohl von der ursprünglichen Vertriebsform auf Kassette herrührt. Inzwischen ist Mastercode aber auch auf Diskette erhältlich.

Dieses, aufgeklappt noch nicht einmal eine Postkarte bedeckende, Heftchen enthält aber, sauber gegliedert, alle Informationen, die zum Betrieb von Mastercode notwendig sind, diese allerdings etwas trocken und nicht gerade mit sinnvollen Beispielen aufgelockert. Im Anhang stehen Tabellen aller Opcodes des 6510 Mikroprozessors in alphabetischer wie numerischer Reihenfolge.

Als Fazit möchte ich ziehen, daß Mastercode sich wohl nur für diejenigen eignet, die kleine Aufgaben in Assembler lösen wollen und dabei nicht unter Zeitdruck stehen. Für einen Anfänger ist, wie schon erwähnt, die Menü-Struktur von Mastercode sehr nützlich.

Profimat

Das Assemblerpaket Profimat von Data Becker enthält den Assembler Profiass und den Monitor Profimon. Beschäftigen wir uns zunächst mit Profiass.

Assembler

Profiass-Quelltexte werden wie normale Basic-Programme eingegeben, können sogar in Basic-Programme eingebettet sein. Ein zusätzlicher Editor oder eine Befehlserweiterung zur leichteren Eingabe wird nicht mitgeliefert. Um gleich die Zusammenarbeit mit Basic-Programmen zu verdeutlichen: Direkt vor dem Quelltext muß Profiass mit SYS 32768 aufgerufen werden, ein Rücksprung ins Basic-Programm ist dann mit dem Pseudo-Opcode ».GTB« möglich.

Label dürfen bei Profimat bis zu acht signifikante Zeichen haben. Berechnungen können in den drei üblichen Zahlensystemen (Hex., Dez., Bin.) vorgenommen werden. Hier sind die Grundrechenarten sowie logische Funktionen und Schiebeoperationen erlaubt. Man scheint allerdings das logische NOT vergessen zu haben, dafür ist XOR enthalten. Klammern dürfen beliebig gesetzt werden.

Eine interessante Eigenschaft von Profiass ist, erstellte Symboltabellen speichern und wieder laden zu können. Sollten Sie öfters Routinen aus dem Betriebssystem benötigen, brauchen Sie die entsprechenden Label nicht für jedes Programm neu zu definieren, sondern können sie in die Symboltabelle laden.

Ebenso interessant ist, daß mehrere Assemblerbefehle durch Doppelpunkt getrennt in einer Zeile stehen dürfen.

Zu den üblichen Pseudo-Opcodes für Tabellen tritt ».FLP« mit dem es möglich ist, Zahlen im Commodore-Fließkommaformat im Speicher abzulegen. Dies ist brauchbar, wenn man in Assembler Fließkommaroutinen schreiben möchte.

Quelltexte können miteinander verkettet werden, das heißt, ein Quelltext ruft den nächsten zu assemblierenden auf. Somit lassen sich auch größere Programme assemblieren, insbesondere, weil Profiass den Objektcode direkt zur Diskette schicken kann.

Auch der Ausdruck von Listings ist steuerbar und kann auf beliebige Peripheriegeräte gesandt werden. Diese Listings werden voll formatiert und sogar seitenweise mit Kopf- und Fußzeilen ausgedruckt.

Makros

Profiass bietet auch Makros. Ein Makro ist eine meist kurze Befehlssequenz, die man nicht jedesmal neu eintippen möchte und deshalb mit einem eigenen Namen versieht, mit dem sie dann jederzeit aufgerufen werden kann. Im späteren Objektcode steht dann anstelle des Makroaufrufes jedesmal die vorher definierte Befehlssequenz. Es ist auch möglich, dieser Befehlssequenz von Mal zu Mal wechselnde Parameter zu übergeben.

Makros bei Profimat dürfen beliebig viele interne Label verwenden. Diese müssen aber durch einen Punkt gekennzeichnet werden, dann erhalten sie automatisch eine Ordnungsnummer, so daß ein mehrmaliges Aufrufen des Makros möglich ist, ohne ein Label zweimal zu definieren. Zwei verschiedene Makros müssen aber verschiedene interne Label benutzen.

Makros dürfen hier nicht verschachtelt werden, das heißt ein Makro darf kein anderes Makro aufrufen.

Alle Makrodefinitionen müssen am Anfang des ersten zu assemblierenden Programms stehen, wenn Quelltexte verkettet werden sollen.

Damit wird das Aufbauen einer Makrobibliothek sehr erschwert.

Insgesamt sind gegenüber anderen, teureren Assemblern die Möglichkeiten der Makros eingeschränkt und wenig flexibel. Meiner Meinung nach hätte dieser Programmteil ruhig einer besseren Benutzerführung zum Opfer fallen können.

Monitor

Der Monitor Profimon enthält so ziemlich alle üblichen Befehle. Es können die Register sowie Speicherstellen in Hex-Dumps oder disassembliert angezeigt werden. Speicherbereiche können verschoben, mit anderen Bereichen verglichen oder mit bestimmten Werten gefüllt werden. Auch das Durchsuchen nach einer Bytefolge ist möglich, ebenso wie natürlich Laden und Speichern von Programmen. Zum Austesten von Programmen ist ein Breakpoint frei definierbar, bei dessen Erreichen in den vorhandenen Einzelschrittmodus umgeschaltet wird.

Mich stört allerdings, daß hier kein einfacher Assembler wie in anderen Monitoren vorhanden ist. Es ist eigentlich zu umständlich, für jede kleine Änderung im Programm den Quelltext erneut zu assemblieren, insbesondere, da Profimon ja eigentlich ein eigenständiges Programm sein soll. Eine Eigenschaft ist allerdings ungewöhnlich. Mit Profimon kann man auch auf das RAM unter dem Basic- und Kernal-ROM zugreifen oder das Charakter-ROM auslesen. Insgesamt gesehen, läßt es sich mit Profimon nicht anders als mit anderen handelsüblichen Monitoren arbeiten.

Dokumentation

Es befinden sich 34 Seiten Anleitung in einem DIN-A5-Ordner im typischen Data Becker-Design. Diese Anleitung hat aber einige Mängel. So wird man beispielsweise erst bei der Erklärung der Verkettung von Quelltexten mit der Tatsache überrumpelt, daß der Assembler mit SYS 32768 gestartet wird, was vorher nicht erwähnt wurde. Wer gezielt Informationen sucht, wird sie teilweise nicht finden, da sie manchmal in thematisch anderen Abschnitten versteckt sind. Das ist eigentlich schade, weil gerade auch der Profimat wohl die Erstkäufer eines Assemblers anspricht. Der Profimat ist nach Preis und Qualität für den Anfänger mit Aufstiegsambitionen geeignet. Für fortgeschrittene Programmierer bietet er allerdings zu wenig.

Profisoft-Assembler

Profisoft vertreibt ein kleines Programmpaket zur Erstellung von Assemblerprogrammen, das im folgenden in Ermangelung eines klangvollen Namens mit Profisoft-Assembler bezeichnet wird. Dieses Programmpaket umfaßt neben einem Assembler auch einen Re-Assembler, einen Mini-Monitor sowie einen Editor und ist sowohl auf Kassette als auch Diskette erhältlich.

Editor

Als Editor für den Quelltext wird der normale Basic-Editor verwendet. Um die Eingabe komfortabler zu gestalten, wurden einige Toolkitähnliche Funktionen in das Programm eingebaut. Diese stehen dann auch zum Schreiben und Editieren von Basic-Programmen zur Verfügung.

Insgesamt gibt es 16 neue Befehle, die alle mit einem »←« beginnen. Darunter befinden sich neben den Aufrufbefehlen für Assembler, Re-Assembler und Monitor, Befehle für automatische Zeilennumerierung und Zeilenumnumerierung (Renumber). Es ist ein Re-New vorhanden, das auch nach einem Reset wirksam ist und den Quelltext, sofern er nicht anderweitig zerstört wurde, wieder regeneriert. Dies ist besonders in der Testphase von Programmen nützlich, da man sich das ewige Abspeichern und Neuladen erspart.

Ein weiterer interessanter Befehl ist »←F«, mit dem man nicht nur Zeichenketten im Quelltext finden, sondern auch durch andere ersetzen kann. Ebenso ist das SAVEn bestimmter Speicherbereiche möglich, wie auch die Ausgabe der genauen Speicherbelegung der Quelltexte und Tabellen. Alles in allem ist das Erstellen von Quelltext sehr komfortabel und dürfte wohl auch höheren Ansprüchen genügen.

Assembler

Doch nun zum Assembler selbst. Hier darf ein Label aus maximal acht Zeichen bestehen, dabei sollten allerdings keine Leer- und Sonderzeichen verwendet werden. Tabellen werden mit drei Pseudo-Opcodes unterstützt, einer für Einzelbytes, einer für Texte und einer für Adressen. Während des zweiten Durchlaufs kann ein Listing wahlweise auf Bildschirm oder Drucker (nach OPEN) ausgegeben werden. Dieses Listing wird allerdings nicht formatiert, das heißt Labels und Kommentare stehen nicht geordnet untereinander. Dies läßt sich nur durch entsprechende Eingabe des Quelltextes erreichen. Einzelne Quelltextfiles auf Diskette/Kassette können beliebig aneinandergehängt beziehungsweise nachgeladen werden.

Re-Assembler

Ein sehr nützlicher Programmteil ist der eingebaute Re-Assembler. Dieser wird in der Anleitung fälschlicherweise als Disassembler bezeichnet. Im Gegensatz zu einem Disassembler, der Maschinenprogramme auf dem Bildschirm anzeigt oder auf dem Drucker ausgibt, erzeugt ein Re-Assembler wieder Quelltext, der dann nach Änderungen erneut assembliert werden kann. Der Vorteil liegt auf der Hand. Wenn Sie nachträglich in ein Programm etwas einfügen oder es umschreiben wollen, aber den Quelltext nicht besitzen, so können Sie sich diesen einfach regenerieren. Natürlich ist dieser Quelltext nicht gleich dem Original, denn woher sollte der Re-Assembler beispielsweise die Namen der einzelnen Labels kennen?

Bei einem vom Profisoft-Re-Assembler erzeugten Quelltext werden alle Speicherzugriffe und Sprünge über Labels abgewickelt. Um die Labels eindeutig zu halten, sehen sie folgendermaßen aus: Der erste Buchstabe ist ein L, gefolgt von der hexadezimalen Adresse des tatsächlichen Speicherplatzes, zum Beispiel LFFD2. Sollte der Re-Assembler auf einen undefinierten Opcode stoßen, so behandelt er ihn als Mini-Tabelle, bestehend aus einem Byte mit dem entsprechenden Pseudo-Opcode. In einem Durchlauf können maximal 4 bis 6 KByte re-assembliert werden; er dauert zirka 30 Sekunden.

Der Mini-Monitor

Nun noch einige Worte zum Mini-Monitor. Mit ihm lassen sich Speicherbereiche hexadezimal anzeigen, verändern und verschieben. Damit sind seine Möglichkeiten schon ausgeschöpft. Er ist allerdings insbesondere für das Verschieben des erzeugten Object-Codes notwendig, wenn der Quelltext an eine andere Stelle als der späteren tatsächlichen Startadresse assembliert wurde.

Dokumentation

Ein Programm, das gerade wegen seines Preises und seiner angepaßten Leistungen für Anfänger und

leicht Fortgeschrittene geeignet ist, benötigt natürlich auch eine gute Anleitung. Die vorliegende ist mit 16 Seiten DIN A5 etwas knapp. Sie setzt schon Kenntnisse der Maschinensprache und des Computers voraus. Es werden alle Funktionen hinreichend genau erklärt. Leider sind nur sehr wenige Beispiele abgedruckt. Ist man allerdings erst einmal mit dem Programm vertraut, wird man die Anleitung gerne weiterhin als schnelles Nachschlagewerk benutzen.

Insgesamt hat das Programm einen sehr guten Eindruck hinterlassen. Seine Bedienung ist einfach und doch komfortabel. Auch bei Fehlern stürzt der Profisoft-Assembler nicht ab (ein Totalabsturz ist nur bei mutwilligem Überschreiben des Assemblers selbst möglich). Mit dem Re-Assembler können auch fremde Programme leicht verändert werden. Das fordert direkt zum Experimentieren heraus. Ein kleiner Schwachpunkt ist lediglich der Monitor.

Maschine 64

Der letzte hier vorgestellte Assembler heißt Maschine 64 und wird von Dynamics vertrieben. Auch Maschine 64 erfüllt eine Vielzahl von Funktionen, es ist Assembler, Re-Assembler, Toolkit, DOS-Support, Monitor und Disk-Monitor in einem, und das bei nur 16 KByte Speicherverbrauch.

Editor

Genauso wie beim Profisoft-Assembler werden Quelltexte wie Basic-Programme eingegeben. Die Eingabe des Quelltextes wird hier von einem Toolkit unterstützt. Vorhanden sind hier die Befehle AUTO, APPEND, DELETE, FIND, RENUMBER. Zusätzlich gibt es dann noch die Befehle ASSEMBLER, REASSEMBLE, MONITOR, DISKMONITOR und BYE, welche die anderen Teile des Systems aufrufen, beziehungsweise Maschine 64 abschalten.

Alle diese Befehle sind auch zum Editieren ganz normaler Basic-Programme geeignet. Zusätzlich ist im Editor, wie auch in allen anderen Programmteilen, eine Diskettenunterstützung, ähnlich dem DOS 5.1, eingebaut.

Assembler

Über den Assembler selbst läßt sich folgendes sagen: Label dürfen beliebig lang sein, es werden allerdings nur die ersten 20 Zeichen unterschieden.

Sehr komfortabel ist das Anlegen von Texttabellen. Diese können nicht nur im ASCII-Code, wie üblich, sondern auch im Bildschirmcode und im invertierten Bildschirmcode angegeben werden. Dies ist sehr nützlich, wenn man Texte direkt in den Bildschirmspeicher schreiben und nicht über die Betriebssystemroutine »Zeichen ausgeben« arbeiten will.

Maschine 64 erlaubt auch Berechnungen im Quelltext. Diese sind allerdings auf Addition und Subtraktion sowie LO-Byte und HI-Byte-Isolierungen beschränkt. Es dürfen maximal acht Klammern gesetzt werden. Das Assemblerlisting am Ende des zweiten Pass wird teilweise formatiert und kann, ebenso wie die Symboltabelle, auch auf einem Drucker ausgegeben werden.

Re-Assembler

Der Re-Assembler läßt kaum Wünsche offen. Er erzeugt aus Objektcode wieder Quelltext. Auch hier werden, soweit wie möglich, Sprünge und Speicherzugriffe über Label abgewickelt. Besonders komfortabel sind allerdings drei Punkte: Dem Re-Assembler kann vor dem Start mitgeteilt werden, wo Tabellen und wo tatsächliches Programm im Speicher stehen, so daß Tabellen und Programm in einem Arbeitsgang in Quelltext umgewandelt werden können. Sollte der Re-Assembler auf einen nicht als Programm identifizierbaren Bytewert treffen, wird aus ihm eine Ein-Byte-Mini-Tabelle mit angehängtem ERROR. Solche Zeilen können dann sehr schnell mit dem FIND-Befehl ausfindig gemacht werden. Als letztes ist es auch möglich, einen bestimmten Speicherbereich so zu reassemblieren, als ob er in einem anderen Bereich stehen würde.

Monitor

Besitzern des PET, dem Großvater der Home- und Personal Computer, wird der eingebaute Monitor unter dem Namen SUMO bekannt sein. Er kann Speicherbereiche anzeigen und ändern sowie durchsuchen, verschieben, vergleichen, füllen, laden und speichern. Ein Disassembler ist ebenso vorhanden wie auch ein einfacher Line-by-Line-Assembler, der sich gerade bei kleinen Änderungen an einem Programm bezahlt macht. Auch sind Umrechnungen Dez-Hex und umgekehrt möglich. Leider fehlt hier ein Einzelschrittmodus, mit dem man Maschinenprogramme Schritt für Schritt auf Funktionstüchtigkeit testen kann.

Disk-Monitor

Kurz angesprochen werden soll auch noch der Disk-Monitor. Beliebige Blöcke der Diskette können in den C 64-Speicher geholt und angezeigt, verändert und wieder zurückgeschrieben werden. Dabei ist es dann auch möglich, innerhalb eines Blocks Bytefolgen zu suchen oder zu verschieben oder gar den Block zu disassemblieren. Auch hier funktioniert dann der Line-by-Line-Assembler. Leider fehlen Befehle, mit denen man blockübergreifend arbeiten könnte, so daß man immer auf einen einzelnen Block bei der Arbeit fixiert ist.

Dokumentation

Die knapp 30 Seiten Anleitung im DIN-A5-Format machen im ersten Augenblick einen recht guten Eindruck, doch werden hier einige nützliche Details der einzelnen Programme verschwiegen, die man erst beim Probieren durch Zufall herausfindet. Ansonsten ist der Text sehr locker und leicht zu lesen. Leider sind aber auch hier die Beispiele sehr knapp gehalten. Ein Fehler im Handbuch soll nicht unerwähnt bleiben. Entgegen der Aussage, daß sich das Programm selbst starte, mußte RUN eingegeben werden. Das kann einen beim ersten Kontakt mit Maschine 64 doch leicht in Verwirrung bringen.

Insgesamt gesehen ist auch Maschine 64 ein sehr brauchbares, sicheres und bedienerfreundliches Programm. Gerade seine vielen kleinen Details und Zusatzfunktionen erleichtern die Arbeit ganz erheblich. Mit Maschine 64 wird man auch als fortgeschrittener Programmierer nicht so schnell den Wunsch nach einem anderen Assembler verspüren. (B. Schneider/gk)

Fazit

Die vier hier vorgestellten Assembler bieten alle eine dem Preis entsprechende Leistung. Man sollte sich jedoch vor einem Kauf genau überlegen, wie weit man in die Maschinensprache einsteigen will. Zum »Reinschnuppern« in die Assemblerprogrammierung genügen diese Programme allemal.

Mastercode Assembler: Markt & Technik, Hans-Pinsel-Str. 2, 8013 Haar bei München, Preis: Kassette 48 Mark, Diskette 63 Mark

Profimat: Data Becker, Merowinger Str. 30, 4000 Düsseldorf, Preis: Diskette 99 Mark

Profisoft Assembler: Profisoft, Sutthauser Str. 50-52, 4500 Osnabrück, Preis: Kass/Disk. 75 Mark

Maschine 64: Dynamics, Postfach 112005, 2000 Hamburg 11, Preis: Diskette 79 Mark