C — Die professionelle Programmiersprache für den C 64

Ganz schlicht und einfach »C« heißt die Programmiersprache, die ursprünglich an den Bell Laboratories für das 16-Bit-Betriebssystem Unix entwickelt wurde. C ist eine Weiterentwicklung einer Spezialsprache mit Namen »B« von Digital Equipment, die wiederum auf eine Sprache namens (haben Sie's erraten?) »A« zurückgeht. B wurde um 1971 geschrieben, um das Betriebssystem Unix — zu der Zeit noch in Assembler programmiert — auf anderen Computersystemen lauffähig zu machen.

1972 erweiterte der bei Bell Laboratories angestellte Programmierer Dennis Ritchie die Sprache B und nannte sie kurzerhand C. Die Vorteile der Programmierung in C gegenüber Assembler waren derart überzeugend, daß 1973 Unix auf C umgeschrieben wurde. Unix war somit das erste Betriebssystem, das in einer höheren Sprache programmiert wurde. Die Tatsache allein, daß es möglich ist, Betriebssysteme in C zu schreiben, wirft ein deutliches Licht auf die Fähigkeiten dieser Programmiersprache.

C wurde von da ab vornehmlich auf 16- oder 32-Bit-Systemen für die verschiedensten Aufgaben verwendet. Für 8-Bit-Prozessoren war C bisher nur für den Z80 (unter CP/M) erhältlich.

Vor kurzem hat nun Data Becker eine C-Version speziell für den C 64 auf den Markt gebracht, den »C-Compiler-64«. Diesen ersten C-Compiler für den 6502/6510-Prozessor haben wir ausführlich getestet. Als erstes stellte sich uns natürlich die Frage, was C an Besonderheiten zu bieten hat und wie die Sprache aufgebaut ist.

Was kann C?

Wer sich an die Sprache C heranwagt, wird sich sehr schnell daran gewöhnen, seine Programme übersichtlich und strukturiert zu schreiben, denn C fordert geradezu dazu auf. C ist von der Syntax her sehr stark an Pascal angelehnt, bietet aber in der Struktur einige Vorteile.

Wie in Pascal können vor Programmbeginn Konstanten vereinbart werden, die sich während des Programmablaufs nicht ändern. Im

Eine Sprache, deren Programme fast mit der Geschwindigkeit von Assembler laufen, aber mit dem Komfort von Pascal erstellt werden, darf sich zu Recht »professionell« nennen. C für den C 64 ist eine solche Sprache.

Unterschied zu Pascal können diese Konstanten aber nicht nur numerische Werte oder Text beinhalten, sondern stehen wahlweise auch für ganze Anweisungsfolgen. Man nennt diese Konstanten dann auch »Makros«. Während der Compilation (Übersetzung) ersetzt der C-Compiler dann diese Makros durch die zugeordneten Anweisungen und bindet sie ins Programm ein.

Auch an Variablentypen hat C einiges zu bieten:

Neben den üblichen Datentypen wie Zeichen (char), 2-Byte-Integer (int) und Fließkommazahlen (float), hat man noch 1-Byte-Integer (short), 4-Byte-Integer (long), Absolut-Integer (unsigned) und doppelt genaue Fließkommazahlen (double) zur Verfügung. Wem das noch nicht ausreicht, der kann sich wie in Pascal auch eigene Variablentypen definieren.

Registervariable

Doch das ist noch nicht alles. Mit C ist es möglich, den Gültigkeitsbereich bestimmter Variablen festzulegen. Ob eine Variable nur in einer Funktion oder für das gesamte Programm gültig sein soll, ist in C schnell und leicht festgelegt.

Wenn es stark auf Geschwindigkeit ankommt, können Variablen auch als Registervariablen deklariert werden. Bei der Übersetzung versucht der Compiler dann, diese Variablen möglichst in den internen Registern der CPU zu halten. Die Verarbeitung erfolgt dadurch um einiges schneller. Diese Register-Option ist natürlich nicht unbedingt für den 6502-Prozessor mit seinen drei Registern (A,X,Y) gedacht.

Besonderes Augenmerk verdient auch die Tatsache, daß C die eigentlich assemblertypische Arbeit mit Zeigern durch eine eigene Arithmetik besonders unterstützt und dadurch auch recht unkonventionelle Wege der Problemlösung erlaubt.

Ebenfalls von Assembler abgeguckt sind die Inkrement- und Dekrementbefehle und die logischen Operatoren, die C zur Verfügung stellt. Neben logischem UND, ODER und EXKLUSIV-ODER hält C auch die Negation, das Einerkomplement und sogar Schiebeoperatoren für Variable bereit.

Selbstverständlich versteht C noch eine ganze Reihe anderer Anweisungen, die in Tabelle 1 zusammen mit den möglichen Datentypen und Speicherklassierungen aufgelistet sind. Betrachtet man diese Liste genauer, wird man viele Elemente der Sprache Pascal wiederentdecken.

Auf den ersten Blick erstaunlich ist, daß C nur relativ wenig Anweisungen versteht. Es zeigt sich hierbei jedoch die C-Philosophie, nur die notwendigsten Funktionen quasi als Grundbausteine zur Verfügung zu stellen und alle »höheren« Funktionen aus Software-Bibliotheken zu entnehmen. Man wundert sich, welch komplexe Probleme sich zusammen mit der mitgelieferten Funktions-Bibliothek lösen lassen.

Dies ist vielleicht das Bedeutende an der Sprache C. Sie ist aufgrund ihrer wenigen Anweisungen leicht zu lernen, kann aber wegen ihrer Flexibilität zu einer komplexen Struktur aufgebaut werden.

Es ist durchaus möglich, C ohne vorherige Programmierkenntnisse zu erlernen. Hat man jedoch zuvor schon in Assembler oder Pascal gearbeitet, wird man sich sicherlich schneller in C einarbeiten können.

Bild 1 zeigt ein kleines aber typisches Programm in C, das die Fakultät einer Zahl berechnet.

C-Compiler-64

Seit kurzem kann jeder C 64-Besitzer alle geschilderten Vorzüge von C genießen: Mit dem »C-Compiler-64« von Data Becker. Geliefert wird der Compiler im Data Becker-üblichen rot-weißen Ringbuch, das

eine 274 Seiten starke Anleitung und selbstverständlich die Diskette mit den notwendigen Programmen enthält.

Das umfangreiche Handbuch gliedert sich in zwei Teile, den Übungsteil und den Systemteil.

Der Übungsteil beschäftigt sich mit dem Einarbeiten und dem Anwenden der vier verschiedenen Teilprogramme des »C-Compiler-64« in recht ausführlicher Weise. Außerdem enthält er eine Einführung in die Programmierung der Sprache C, die (wie im Vorwort zu lesen ist) das nötige Grundwissen vermitteln soll.

Es ist sicherlich möglich, die Grundbegriffe von C mit dieser Einführung zu verstehen. Die Erklärung der Sprache C geschieht hier aber in sehr komprimierter Weise, so daß die Anschaffung eines speziellen Lehrbuches für den Anfänger doch empfohlen werden muß.

Zur Auffrischung bereits vorhandener C-Kenntnisse leistet der ins Handbuch integrierte C-Kurs allerdings recht gute Dienste.

Der zweite Teil des Handbuchs mit der Bezeichnung »Systemteil« zeigt noch einmal zusammengefaßt die vollständige Funktions- und Anwendungsweise der einzelnen Teilprogramme, wie sie im Übungsteil teilweise sehr ausführlich erklärt wird. Ebenso wie die anschließende Syntaxübersicht und Kurzerklärung der Befehle von C ist diese Zusammenfassung als eine Art Nachschlagewerk zum Compiler gedacht.

Standardbibliothek auf Diskette

Doch nun zur Programmdiskette: Auf der Diskette befinden sich neben dem C-Lader die Teilprogramme C-Copy, C-Editor, C-Compiler und C-Linker. Zusätzlich findet man noch die für C obligatorische Standardbibliothek und einige Beispielprogramme zum Experimentieren.

Zu Beginn hat man den C-Lader zu laden und zu starten. Man findet sich wieder in einem übersichtlichen Menü, über das man die vier Teilprogramme aufrufen kann. Über diesen Lader können auch übersetzte C-Programme geladen und gestartet werden.

C-Copy

Als besonderes Bonbon wird C-Copy im Vorwort des Handbuchs bezeichnet. C-Copy ist ein kleines

Datentypen	Speicher- klassierung	An- weisungen
char	auto	break
int	static	case
float	extern	continue
double	register	default
long		do
short		else
struct		for
union		goto
unsigned		if
typedef		return
		sizeof
		switch
		while

Tabelle 1. Die Standard-Anweisungen von C

erron	erhoff	nmion
nmioff	qerror	error
exit	fileopen	fileclose
putc	getc	puts
gets	putf	getf
move	cursor	strlen
strcpy	strcmp	strcat
alloc	free	

Tabelle 2. Funktionen von »stdio.1« (ohne Funktionsargumente)

Hilfsprogramm, das die Arbeit mit der Floppystation erleichtern soll. Es ist auch wichtig zum Erstellen einer Arbeitsdiskette für die eigenen C-Programme. Doch dazu später.

Über einzelne Kommandozeichen lassen sich Fehlerkanal und Directory einlesen. Durch weitere Kommandos (Größer/Kleiner-Zeichen) kann man Files aller Art außer »REL« laden und wieder speichern. Außerdem besteht die Möglichkeit, die Gerätenummer der Floppy softwaremäßig einzustellen, um von einem Gerät zum anderen kopieren zu können.

Wie man sieht, ist C-Copy also ein recht hilfreiches Werkzeug zur Behandlung von C-Programmen und Files.

Text-Editor der Spitzenklasse

Kommen wir jetzt zum wichtigsten Teilprogramm für den C-Programmierer, zum Eingabe-Editor »C-Editor«.

Mit ihm werden die C-Programme eingegeben und editiert. »C-Editor« bietet hierfür eine überraschend große Anzahl von Kommandos und Befehlen, um dies möglichst komfortabel und einfach zu gestalten.

Zum Ersten fällt sofort der geänderte Zeichensatz ins Auge, der angenehm und gut zu lesen ist. Dies ist keine Spielerei, wie manche denken mögen, sondern notwendig, da C einige Zeichen verwendet (zum Beispiel die geschweiften Klammern), die der Standardzeichensatz des C 64 nicht besitzt. So haben einige Tasten neue Zeichen bekommen, an deren Belegung man sich aber sehr schnell gewöhnt.

Wie viele Textbearbeitungsprogramme hat auch C-Editor einen Kommandoblock am oberen Bildschirmrand, der die Cursorposition, den Filenamen, wichtige Meldungen und eine Tabulatorzeile anzeigt.

Es stehen zwei Textbereiche zur Verfügung, der normale Filetext und ein sogenannter Extratext. Über Tastendruck kann man zwischen den beiden »Textseiten« hin und her blättern.

Es ist möglich, das Format des Eingabetextes zwischen 40 und 80 Zeichen pro Zeile vorzuwählen. Wird ein Format größer 40 Zeichen gewählt, so wird der Bildschirm bei der Eingabe und beim Editieren auf angenehme Weise Zeichen für Zeichen nach links oder rechts weggeschrollt. Die Eingabe von Zeilen, die länger als 40 Zeichen sind, ist deshalb sehr komfortabel und übersichtlich (anders als bei manchen Textbearbeitungsprogrammen, die durch sprunghaftes Scrollen die Eingabe von Texten zu einem Abenteuer werden lassen). Während der Programmeingabe kann das Format leider nicht mehr verändert werden, da der Editor dann den kompletten Filetext löscht (new text-Befehl!).

Jede Zeile kann außerdem verschieden eingefärbt werden, was zur Übersichtlichkeit von C-Programmen viel beiträgt. Wie viele Textprogramme besitzt auch der C-Editor eine sogenannte letzte Zeile (ähnlich wie bei Vizawrite das »end of page«), welches das Textende anzeigt. Leider zeigt C-Editor diese Zeile im Text fatalerweise nicht an. Das Ende des Textbereiches bemerkt man erst, wenn man keine Zeichen mehr eingeben kann und oben im Editorkopf statt des angezeigten Editornamens »C-Editor 1.0« relativ unscheinbar die Meldung »last line« zu lesen ist. So kann das Textende ziemlich überraschend kommen; dann muß man erst wieder mit dem Zeileneinfügekommmando (F7) Platz schaffen, da C-Editor dies nicht automatisch macht! Diese Prozedur kann bei längeren Programmen nervtötend sein. Gott sei Dank besitzen aber alle Tasten (so auch F7) eine Repeatfunktion.

Abgesehen davon bietet C-Editor aber sehr viele Kommandos, die eher an ein Textbearbeitungsprogramm als an einen einfachen Eingabeeditor erinnern.

Abgesehen davon bietet C-Editor aber sehr viele Kommandos, die eher an ein Textbearbeitungsprogramm als an einen einfachen Eingabeeditor erinnern.

Abgesehen davon bietet C-Editor aber sehr viele Kommandos, die eher an ein Textbearbeitungsprogramm als an einen einfachen Eingabeeditor erinnern.

Neben dem Kopieren, Verschieben und Löschen von Textblöcken gibt es auch, wie schon bemerkt, eine Such- und Ersatzoption und ein Einfärbekommando, mit dem man Blöcke in verschiedene Farben tauchen kann.

Schließlich läßt sich auch der Fehlerkanal der Floppy und das Directory einlesen. Doch Vorsicht! Das Directory wird in den Text mit eingebunden und muß dann erst wieder gelöscht werden.

Es empfiehlt sich, vorher auf die Extraseite umzuschalten, wo es nicht stören kann. Natürlich fehlt auch die Printeroption zum Ausdrucken des Textes und die Lade- und Speichermöglichkeit nicht.

C-Editor ist insgesamt sehr komfortabel und steht manchen Textverarbeitungsprogrammen in der Anzahl der Kommandos und besonders deren Ausführungsgeschwindigkeit nicht im geringsten nach, eher ist das Gegenteil der Fall.

Der Compiler

Nun zum Kernstück des »C-Compiler-64«, dem Compiler selbst.

Wie bei vielen Compilern gibt es auch bei diesem für den Anwender wenig zu tun. Die Arbeit leistet nämlich der Compiler selbst, und zwar relativ schnell, wie es bei der Sprache C üblich ist.

Nach der Eingabe des Namens des zu compilierenden Programms und dem Filenamen des compilierten Programms, heißt es für den Programmierer beobachten und hoffen, daß kein Fehler in seinem Programm steckt.

Zu sehen gibt es allerdings nicht viel. Es wird nur angezeigt, bei welcher Funktion oder welchem Zusatzfile der Compiler gerade ist.

Hat er einen Fehler entdeckt, so gibt er eine komplette Fehlermeldung mit betreffender Zeilennummer (wichtig fürs spätere Editieren) auf dem Bildschirm aus.

Ein Fehlerprotokoll auf dem Drucker ist leider nicht möglich.

Man muß jetzt jedoch nicht wie wild alle auftretenden Fehler mit-schreiben, denn das übernimmt der Compiler selbst. Während des Compilierens werden alle auftretenden Fehler in einem Extrafile (»error-c«) mitprotokolliert. Dieses File kann später mit dem Editor in den Extratext geladen werden, um dort bei der Fehlersuche und deren Behebung behilflich zu sein.

Gehört man zu den seltenen Glückspilzen, deren Programme

keine Fehler haben, ist die Arbeit allerdings noch nicht zu Ende. Denn mit dem vom Compiler erstellten sogenannten Linkfile kann man noch nichts anfangen. Es muß erst vom sogenannten Linker mit anderen notwendigen Programmteilen verbunden und zu einem lauffähigen Programm zusammengebastelt werden. Diese Arbeit besorgt der »C-Linker«, das vierte Teilprogramm des C-Compiler-64.

Der Linker

Dieser Teil des C-Compilers-64 erzeugt, wie schon erwähnt, ein lauffähiges C-Programm aus dem vom C-Compiler gelieferten Linkfile.

Mit ihm können Sie nun getrennt compilierte, aber zusammengehörige C-Programme aneinanderbinden.

Haben Sie Funktionen aus der Standardbibliothek in Ihrem C-Programm verwendet (was meistens der Fall ist), so muß auch das Linkfile der Standardbibliothek dazugebunden werden.

Der Linker sucht sich dann aus der Bibliothek die verwendeten Funktionen und fügt sie dem eigentlichen C-Programm hinzu. Sie können aber auch selbstgeschriebene oder erweiterte Funktionssammlungen mit dem Linker an Ihr C-Programm binden. Wichtig ist nur, daß sich alle notwendigen Files auf der verwendeten Diskette befinden. Deshalb ist es ratsam, vor der Arbeit mit dem Compiler und dem Linker mit C-Copy eine Arbeitsdiskette anzufertigen, die das Linkfile der Standardbibliothek und eventuell eigene Funktionssammlungen enthält.

Mit dem C-Linker können maximal sieben verschiedene Linkfiles zu einem lauffähigen C-Programm zusammengebunden werden. Dabei steht es dem Benutzer frei, zu wählen, ob das fertige C-Programm später von Basic oder nur vom C-Lader aus geladen und gestartet werden soll.

Nebenbei ist es noch möglich, die Speicherbergrenze festzulegen, um eventuell obere Speicherbereiche vor den C-Programmen zu schützen.

Ist das getan, heißt es warten. Je nach Anzahl und Länge der einzelnen Linkfiles kann es mehr oder weniger lang dauern, bis der C-Linker in zwei Durchgängen (Fachwort: passes) alle notwendigen Funktionen und Files zu einem C-Programm »verschweißt« hat, das endlich lauffähig ist.

Selbstverständlich können auch hierbei Fehler auftreten, wenn zum Beispiel eine benötigte Funktion in keinem der angegebenen Linkfiles zu finden ist.

Gott sei Dank ist es möglich, den C-Linker wie auch alle anderen Teilprogramme des C-Compiler-64 durch gleichzeitiges Drücken von RUN/STOP und RESTORE zu unterbrechen. Mit einem kleinen Untermenü kann man dann auf Tastendruck einen Neustart wagen oder zum Lader zurückkehren, von dem aus alle anderen Teile des C-Compiler-64 geladen werden können.

Läuft alles zur Zufriedenheit (was selten auf Anhieb passiert), hat der C-Linker das lauffähige C-Programm auf der Arbeitsdiskette abgelegt. Zur Erinnerung, ob es sich um eine B-Version (nur von Basic aus zu laden und zu starten) oder um eine L-Version (Starten nur vom C-Lader« möglich) handelt, sollte man dies im Programmnamen vermerken. Denn eine L-Version läßt sich von Basic aus nicht starten und umgekehrt eine B-Version nicht über den Lader.

Vorsicht ist bei den B-Versionen geboten. Ist das C-Programm beendet, wird sofort ein Reset ausgelöst, und unser Computer meldet sich mit seiner bekannten Einschaltüberschrift. Ein Neustart ist dann nur mit dem erneuten Laden des C-Programms möglich (nervtötend bei umfangreichen Programmen).

Bei der L-Version wird man auch wenig Glück haben. Nach Beendigung des Programms springt der Computer zwar in den Lader zurück, doch warten muß man trotzdem, wenn man neu starten will. Das C-Programm wird nämlich nochmals geladen! Eine Sicherheitsabfrage sollte daher am Ende eines jeden C-Programms nicht vergessen werden.

Positiv dagegen ist, daß ein laufendes C-Programm (ob L- oder B-Version) wie der Compiler selbst mit RUN/STOP und RESTORE unterbrochen werden kann. Dann kann man ebenfalls per Untermenü entscheiden, ob mit dem Programm fortgefahren, wieder gestartet oder dieses beendet werden soll. Das gleiche geschieht, wenn ein sogenannter Laufzeitfehler auftritt. Wird im Programmablauf zum Beispiel eine Variable durch Null geteilt, stoppt das Programm mit der entsprechenden Fehlermeldung und dem bekannten Untermenü, das sich jedoch auch abschalten läßt.

```

page: 1      fakultaet.c
date: 09/07/85
1 typedef int file;
2
3 extern void printf(),scanf(),getc();
4
5 char inKey(fd)
6 file fd;
7 { char c;
8   while((c=getc(fd))!=0);
9   return c;
10 }
11
12 main()
13 {
14   int n,i;
15   double fact;
16   char z;
17   z="";
18   while (z!="q") {
19     printf("\nDieses Programm berechnet die Fakultaet von n");
20     printf("\nn= ");
21     scanf("%d",&n);
22     fact=1;
23     for (i=2; i<=n; i++) fact=fact*i;
24     printf("\nFakultaet von %3d = %f\n",n,fact);
25     printf("\nWeiter mit Taste\n");
26     z=inKey(0);
27   }
28 }
29 extern void error(),eroff();
30 extern printf(),exit();
31
32 main()
33 {
34   int a;
35   a=0;
36   eroff();
37   a=10/a;
38   a=qerror();
39   printf("error nr %d",a);
40   error();
41   exit();
42 }

```

**Ein Beispiel zur Programmierung in C.
Das Programm berechnet Fakultäten
natürlicher Zahlen. Fehler werden abgefangen.**

Die Sprache C besitzt in der Grundaufführung allgemein keine Eingabe- und Ausgabekommandos. Diese jedoch äußerst notwendigen Routinen sind in der bereits erwähnten Standardbibliothek enthalten. C-Compiler-64 besitzt deren drei:

»stdio.c« (Standard Input/Output) ist ein im C-Quelltext vorliegendes, also noch nicht kompiliertes Programm. Es deklariert alle in der Standardbibliothek vorkommenden Funktionen als extern, das heißt als nicht im Quelltext selbst befindlich. Der Compiler muß dies wissen, um ordnungsgemäß übersetzen zu können. Weiterhin enthält dieses File noch wichtige Konstanten.

Die Standardbibliothek

Man kann diese einzelnen Deklarationen auch selbst vornehmen. Es ist jedoch einfacher, dies durch einen C-Befehl («#INCLUDE "stdio.c"») am Anfang seiner Programme zu tun. Der Compiler bindet dann »stdio.c« an dieser Stelle in den Quelltext ein. Das zweite File »stdio.l« ist im dritten File »stdio2.l« nochmals enthalten. Man benötigt also eigentlich nur

»stdio2.l«. Diese bereits compilierte Bibliothek enthält alle wichtigen Ein- und Ausgabefunktionen, die zur Bildschirm-, Tastatur- und besonders zur weiteren Peripheriebedienung unbedingt notwendig sind. Zusätzlich sind Funktionen zur Stringbehandlung und zur Laufzeitfehlerbehandlung vorhanden.

Es ist zum Beispiel möglich, mit bestimmten Funktionen die automatische Unterbrechung bei Laufzeitfehlern abzuschalten und den aufgetretenen Fehler aus einem Fehlerregister zu lesen, um eigene Fehler-routinen zu schreiben. Ebenso läßt sich der NMI (Nicht-maskierbarer Interrupt) an- und abschalten und schließlich selbst Laufzeitfehler auslösen, was recht unkonventionelle Problemlösungen zuläßt. Ebenso fehlen auch die für die Sprache C wichtigen Funktionen zur formatierten Ein- und Ausgabe nicht, die verschiedene Formate und dem Standard entsprechend verschiedene Zahlensysteme verarbeiten können.

Insgesamt bietet die Standardbibliothek des C-Compiler-64 von einem recht zufriedenstellenden Funktionsvorrat, der die für C not-

wendigsten Routinen enthält. Tabelle 2 zeigt den vollständigen Funktionsumfang von »stdio.l«.

Wer will, kann seine eigene Funktionssammlung aufbauen oder »stdio2.l« durch Zusatzfiles erweitern und so mit der Zeit seine Bibliothek mit Funktionen ausstatten, die sonst nur C-Compiler auf größeren Rechnern besitzen.

C-Compiler-64 beherrscht bis auf Bitfelder den kompletten Sprachumfang von C. Während der Testdauer konnten auch keinerlei Übersetzungsschwierigkeiten festgestellt werden. Allerdings muß man bei der Verwendung von Bibliotheksfunktionen aufpassen, wenn man Programme von anderen C-Versionen abtippt. Es gibt zwar nur selten Dialektunterschiede bei C, doch kann es vorkommen, daß bei bestimmten Compilern gewisse Funktionen andere Namen als üblich haben oder im C-Compiler-64 nicht enthalten sind. Diese müssen mit Makros undefiniert oder mit vorhandenen Funktionen zusammengesetzt werden.

Fazit

Insgesamt ist Data Becker mit dem C-Compiler-64 eine durchweg gelungene Implementierung der Sprache C gelungen. Bis auf die besagten Schönheitsfehler im ansonsten vorbildlichen Eingabe-Editor steht dem Anwender damit ein durchaus professionell zu nennendes Entwicklungssystem für C zur Verfügung. Natürlich kann man nicht erwarten, alle auf Großrechnern üblichen C-Bibliotheksprogramme auf der Diskette zu finden, bei einem Preis von immerhin 298 Mark hätte es aber schon ein wenig mehr sein dürfen. Für ungeduldige Anwender sind die relativ langen Ladezeiten des Compilers und des Linkers auf Dauer doch recht störend. Es wäre wünschenswert, wenn man sich bei Data Becker dazu entschließen könnte, die Schnelllade-Routinen des Profi-Pascal (Test Ausgabe 8/85) auch in den C-Compiler zu integrieren.

Für den Anwender, der sich ernsthaft mit der Sprache C befassen will, ist dieser erste C-Compiler für den C 64 sicher eine interessante Sache. Ganz ohne 16-Bit-Computer oder CP/M eröffnet sich hier die Welt der C-Programmierung. Man sollte aber nicht vergessen, als Anfänger ein geeignetes Lehrbuch zusätzlich zu kaufen, um die Sprache richtig beherrschen zu lernen.

(M. Thomas/ev)

Info: C-Compiler-64, Data Becker, Düsseldorf, 298 Mark