

Plus:

- vollständiger C-Sprachumfang
- umfangreiche Grafik-Bibliothek
- RAM-Floppy bei C 128-Version
- komfortabler Texteditor
- System um eigene Kommandos erweiterbar
- ausführliches und umfangreiches Handbuch
- sehr gutes Preis-Leistungs-Verhältnis
- Sicherheitskopie erhältlich

Minus:

- Diskette kopiergeschützt
- Programme nur mit Kommandoprozessor lauffähig

Stärken und Schwächen von Profi-C**Eigene Programme**

Wem die Befehle von Profi-C nicht ausreichen, der kann sich ohne weiteres seine eigenen schreiben. Die Sprache C wurde ja so konzipiert, daß man jederzeit eigene Befehle (in C Funktionen genannt) erstellen kann. Die zusätzlichen Befehle werden ähnlich wie bei CP/M in übersetzter Form auf Diskette gehalten und können wie normale, fest eingebaute Befehle verwendet werden.

Allerdings stellt man bald fest, daß die Profi-C-Programme nur innerhalb des Kommandoprozessors laufen. Man kann also leider keine C-Routinen programmieren, die dann selbständig laufen können und etwa per SYS-Befehl aufrufbar sind. Allerdings kann aus einem C-Programm ein selbststartendes Programm gemacht werden, das zunächst den Kommandoprozessor und dann das C-Programm lädt und ausführt. Leider ver-

mißt man in dem Handbuch die Erlaubnis, übersetzte C-Programme, die ja immer den Kommandoprozessor enthalten, weitergeben zu dürfen. Dies und die Tatsache, daß sich derart compilierte Programme immer mit einem aufwendigen Titelbild melden, setzt der Verwendbarkeit des Compilers über gewisse Spielereien hinaus doch recht enge Grenzen. Dies ist besonders schade, als es sich bei Profi-C wirklich um ein rundum gelungenes Softwareprodukt handelt, mit dem professionelles Arbeiten möglich ist.

Umfangreiches Handbuch

Das Handbuch für Profi-C ist sehr umfangreich (350 Seiten) und informativ ausgefallen. Dem Anfänger dient es gleichzeitig als Anleitung zum Erlernen von C, da ein ausführlicher Übungsteil vorhanden ist, dem es nicht an Beispielen mangelt. Sämtliche bereits vorhandenen Funktionen werden sehr an-

schaulich beschrieben. Eine genaue Anleitung zur Bedienung der RAM-Floppy und des Kommandoprozessors erleichtern die Arbeit beim ersten Kontakt mit Profi-C.

Erwähnenswert sind auch noch die abgedruckten Listings der Standard-C-Bibliotheken. Wer noch nie etwas von C gehört hat, sollte sich trotzdem ein eigenes C-Buch zulegen, da auch das Profi-C-Handbuch an verschiedenen Stellen Grundwissen voraussetzt.

Fazit

Abschließend kann gesagt werden, daß mit Profi-C ein sehr leistungsfähiges Entwicklungspaket für den C 64 und C 128 zur Verfügung steht, das zusätzlich durch Komfort und Geschwindigkeit besticht. Der Preis von 99 Mark ist angesichts der hohen Qualität von Software und Handbuch mehr als angemessen. (ev/nj)

Info: Data Becker, Merowinger Str. 34, 4000 Düsseldorf, Preis: 99 Mark

Small C**Der C-Compiler für Systemprogrammierer**

Die Sprache C verbindet die Vorteile einer Hochsprache in fast idealer Weise mit denen von Assembler. Mit Small C kann sich jeder CP/M-Anwender selbst davon überzeugen.

Geliefert wird Small C auf drei (!) Disketten im 1571-Format. Der Clou: Alle Programme werden mit Quelltext ausgeliefert. Wer Lust und Laune hat, kann sich so sehr schnell ein sehr eigenständiges System aufbauen, bestehend aus Compiler, Z80-Makro-Assembler, Linker und Text-Tools. Mit Hilfe des mitgelieferten Quellcodes können alle Bestandteile der

Software beliebig geändert werden. Man kann den Compiler erweitern, die Text-Tools mit eigenen Ideen bereichern oder Assemblerprogramme schreiben. Wie schon das Handbuch verrät, ist Small-C kein Compiler für große Anwendungen, sondern eher als Entwicklungssystem gedacht. In dieser Funktion aber ist es eine empfehlenswerte Alternative zur Z80-Assemblerprogrammierung.

Der Compiler selbst namens »CCCOM« ist mit einem weiteren Programm »ARCOM« auf einer der drei Disketten zu finden. Mit

»ARCOM« können Archivdateien verwaltet werden. In einer Archivdatei sind mehrere Dateien in einer großen Datei zusammengefaßt. Das Archivverwaltungsprogramm »ARCOM« bearbeitet diese Datei nun im vom Anwender gewünschten Ausmaß. Es können jederzeit neue Dateien in das Archiv aufgenommen werden, ebenso kann nach Herzenslust gelöscht und hinzugefügt werden. Natürlich ist auch ein Zerlegen der Archivdatei in ihre Bestandteile möglich. Der praktische Nutzen eines Archivs liegt im Anlegen von Bibliotheken

für bestimmte Anwendungen. Die anwendungsspezifischen C-Funktionen, die der Anwender sich selbst erstellt hat, können so platzsparend und vor allem schnell auffindbar in Archivdateien gesichert werden.

Einen kleinen Nachteil des Compilers bemerkt man bereits beim ersten fehlerhaften Durchlauf: die sonst sehr eindeutigen Fehlermeldungen lassen völlig die Zeilennummer vermissen, in der der Fehler aufgetreten ist. Dafür wird aber die gesamte fehlerhafte Zeile angezeigt, wobei die genaue Fehlerposition markiert ist. Wie bei al-

len Bestandteilen des Small-C-Pakets hat man auch beim Compiler die Möglichkeit, verschiedene Parameter zu übergehen, sobald ein Fehler auftritt oder der Quellcode wird während der Compilierung auf einem Gerät nach Wahl ausgegeben. Der Compiler erzeugt 8080-Quellcode, der in eine Ausgabedatei vom Typ »MAC« (für Macro-Assembler) geschrieben wird.

Aufbau des Small-C-Pakets

Den vom Compiler in Assembler übersetzten Quellcode wandelt der auf der zweiten Diskette befindliche Macro-Assembler in eine für den Linklauf nötige Datei um. Auch beim Assemblieren können wieder verschiedene Parameter die Ausgabe auf dem Bildschirm beeinflussen. Selbstverständlich kann der Macro-Assembler auch völlig unabhängig vom C-System eingesetzt werden. Der Assembler kann wahlweise als 8080- oder als Z80-Assembler eingesetzt werden und erzeugt in beiden Fällen relocatiblen Objektcode, der vom Linker zusammen mit beliebigen anderen Modulen gebunden werden kann. Der Linker schließlich erweist sich als das schwächste Glied in der Umwandlungskette. Denn hier braucht selbst der erfahrene Anwender eine gehörige Portion Geduld, wenn der Linker alle benötigten Bibliotheksfunktionen nach und nach aus der Bibliothek »CLIB.REL« holt und zu einem ablauffähigen COM-File zusammenbindet. Dieser große Zeitbedarf von mehreren Minuten selbst bei kleinen Programmen geht allerdings zu einem guten Teil auf das Konto der für ein CP/M-System einfach zu langsamen Commodore-Laufwerke. Doch am Ende hat man auf jeden Fall ein voll ablauffähiges Programm zur Verfügung, dessen Geschwindigkeit doch recht beachtlich ist und über den Zeitverlust beim Linken hinwegtröstet (siehe Listing 1 und Tabelle 1).

Sehr lobenswert erscheint der Umfang der mitgelieferten Text-Tools. Da findet sich

neben einem Editor und einem Formatierprogramm auch jede Menge Nützliches zur Bearbeitung der Quell-

Die Text-Bausteine

texte (Tabelle 2). Weder ein Kopierprogramm noch ein Programm zum Auswählen der Schriftarten für Epson-Drucker fehlen hier. Wer viel sortiert kann dies mit einem leistungsfähigen Sortierprogramm tun. Sogar zwischen Quicksort und Shellsort kann man hier wählen. Einen kleinen Nachteil hat die ganze Sache allerdings: Sämtliche Tools müssen erst compiliert werden, da sie nur im Quellcode vorhanden sind. Allerdings stellt diese Aufgabe selbst den Anfänger vor keinerlei Probleme, da alle Einzelschritte ausführlich im Handbuch beschrieben sind. Eines sei noch gesagt: Wenn sehr lange Programme, wie zum Beispiel der Editor, compiliert werden, muß schon etwas Zeit investiert werden. Vor allem der Linklauf ist wie bereits angesprochen, oft entnervend und zeitraubend.

Der mitgelieferte Editor ist zwar nicht so komfortabel wie der CP/M-Editor ED, wenn man hier überhaupt von Komfort sprechen kann, aber die mitgelieferten Tools bieten hierzu sinnvolle Erweiterungen. Natürlich verarbeitet der Compiler auch Quellcodes, die mit dem regulären CP/M-Editor erstellt wurden.

Der Sprachumfang

Der Sprachumfang von Small-C kann sich für ein Entwicklungssystem durchaus sehen lassen. Zwar verfügt der Anwender nur über die Datentypen »char« und »int« sowie über entsprechende Pointer-Typen, doch dafür kann er aus einem großen Angebot von Funktionen wählen. Für den Systemprogrammierer interessant: Er kann auf eine Menge Funktionen zurückgreifen, die CP/M-Dateien manipulieren. So ist Small-C fast ideal zum Programmieren von CP/M-Utilities wie Kopierprogrammen etc. Sehr lobens-

wert ausgefallen sind auch einige Funktionen zur Programmsteuerung. So können beispielsweise Parameter aus dem Programmaufruf und der Programmname durch eine spezielle Funktion ermittelt werden. Funktionen zur Zeichenkettenbehandlung und Zeichenklassifizierung erleichtern die Arbeit wesentlich. Auch die Zeichenumwandlung stellt kein Problem dar. Etwas zu kurz geraten ist jedoch die Mathematikbibliothek: Sie besteht lediglich aus zwei Funktionen. Hier ist der experimentierfreudige C-Programmierer gefragt. Für die Formatierung von Integerwerten und Zeichenketten besticht Small-C mit einer sehr großen Auswahl. Umwandlungen von Zeichen nach Zahl und umgekehrt mit Rückgabe der Länge oder Umrechnungen von Hexadezimal- und Oktalzahlen werden voll unterstützt.

Compilierzeit	: 28,7 s
Assemblierzeit	: 33,1 s
Linkzeit	: 4 min. 40,4 s
Ausführungszeit	: 1,4 s

Tabelle 1. Zeiten für Übersetzung und Ausführung des Demo-Programms »Laufzeittest«

Fazit

Wer Spaß am Tüfteln hat und sich gerne eigene Systemprogramme schreibt oder mit der gelieferten Commodore CP/M-Software nicht zufrieden ist, dem bietet Small-C eine leicht zu erlernende Möglichkeit, seine Fähigkeiten unter Beweis zu stellen. Für einen Preis von 148 Mark erhält man ein leistungsfähiges Software-Paket, das alle Voraussetzungen zum Entwickeln von Systemsoftware und zum Einstieg in die Welt der C-Programmierung in sich birgt.

(Roland Fieger/bj)

```
type test.c
/* Small C Laufzeit-Test */
/* Herunterzaehlen von 10000 */

include "stdio.h"

main()
{
    int i;

    i=10000;
    printf ("\nS T A R T");
    getchar ();

    while (i) i--;

    printf ("\n\nF E R T I G");
    getchar ();
}
```

Listing 1. Laufzeit-Testschleife in »C«

CHG(Change)	Ersetzen von Zeichenketten in Textdateien
CNT(Count)	Zählen von Zeichen, Wörtern oder Zeilen
CPY(Copy)	Kopieren von Textdateien
CPT(Crypt)	Ver- und entschlüsseln von Dateien
DBT(Detab)	Ersetzen von Tab-Zeichen durch Leerzeichen
EDT(Edit)	Zeileneditor
ETB(Entab)	Gegensatz von DBT
FND(Find)	Suchen von Zeichenketten in Textdateien
FNT(Font)	Auswahl von Schriftarten des Epson-FX-80 und kompatible Drucker
FMT(Format)	Formatieren (Druck aufbereiten) von Textdateien
LST(List)	Ausgabe von Textdateien auf den Bildschirm
MRG(Merge)	Zusammenhängen von zwei sortierten Textdateien
PRT(Print)	Drucken von Textdateien
SRT(Sort)	Sortieren von Textdateien
TRN(Trans)	Kopiert Textdateien und ändert Zeichenketten

Tabelle 2. Die »Small Tools« zur Bearbeitung von Textdateien