

C für Profis

Die Programmiersprache C erfährt in letzter Zeit eine immer größer werdende Beliebtheit. Mit dem Profi-C-System von Data Becker können jetzt auch Commodore-Besitzer alle Vorzüge dieser Sprache ohne Einschränkungen genießen.

Profi C gibt es in zwei Varianten, nämlich in einer C 64- und einer C 128-Version. Beide Versionen haben den gleichen Sprachumfang und sind auch sonst recht ähnlich. Erfreulicherweise wurden aber bei der C 128-Version konsequent die sich durch das bessere Betriebssystem und den größeren Speicher ergebenden Möglichkeiten voll ausgeschöpft. So kann der C 128-Besitzer mit Profi C selbstverständlich im 80-Zeichen-Modus arbeiten, C-Disketten mit Autostart ausrüsten oder beispielsweise auch im Fast-Modus arbeiten. Eine komplette Speicherbank kann beim C 128 darüber hinaus als RAM-Floppy genutzt werden, was die Geschwindigkeit beim Compilieren und Linken beträchtlich steigert. Natürlich ist in der C 128-Version auch das C-System selbst auf einer bootfähigen Diskette, so daß man nur noch die Diskette einlegen und den Computer einschalten muß. Kurz danach kann man mit Hilfe des umfangreichen Handbuchs bereits loslegen und sein erstes Programm schreiben.

Kommandos wie bei CP/M

Als erstes wird der eingebaute C-Kommandoprozessor in den Speicher des C 128/C 64 geladen. Dieser Kommandoprozessor ist nichts anderes als ein eigenes kleines Betriebssystem. Schon beim ersten Kontakt mit dem Kommandoprozessor fällt sofort eine starke Anlehnung an CP/M auf: Das C-System meldet sich mit »A:« und fordert zu Eingaben auf. Der Buchstabe »A« bedeutet dabei, daß Floppy-Laufwerk »A« (entspricht Gerätenummer 8) aktiv ist. Die Disket-

tenverwaltung erfolgt über völlig eigenständige Befehle, die im Kommandoprozessor integriert sind. Wie unter CP/M üblich, wird das Inhaltsverzeichnis mit <DIR> aufgelistet und der Inhalt einer Datei mit <TYPE> auf den Bildschirm gebracht.

Um die Quellprogramme zu erstellen, findet sich auf der Profi-C-Diskette ein externer Editor, der mit einer entsprechenden Anweisung im Kommandoprozessor aufgerufen wird. Dem Anwender präsentiert sich ein Texteditor, der nicht nur über ausgesprochene Textverarbeitungsfunktionen verfügt, sondern auch über eine Extra-Seite, die beispielsweise für die nach dem Compilieren erzeugte Fehlerdatei Platz bietet. An Textverarbeitungsfunktionen bietet der Editor Verschieben, Kopieren, Umschalten auf die Extra-Seite, Löschen und sogar eine Funktion zum Suchen und Ersetzen. Das Druckerlisting des gerade editierten Programms erwähnt selbst alte »Programmierhasen«. Am Anfang jeder Seite findet sich der Programmname, die Seitennummer und das Datum. Außerdem ist jede Zeile numeriert, was beim Editor nicht der Fall ist, der arbeitet wie die meisten Textprogramme mit der Anzeige von Spalte und Zeile in der Kommandoleiste. Der Editor ist wie alle anderen Teile des Softwarepakets in der Lage, mit der RAM-Floppy (nur C 128) zu kommunizieren. Falls das Standardschriftbild des Editors nicht gefällt, so ändert man dieses eben. Drei verschiedene Zeichensätze stehen zur Verfügung.

Um das fertige Programm zu compilieren, muß der Editor verlassen werden; es meldet sich automatisch wieder der Kommandoprozessor.

Nun erfolgt der Aufruf des Compilers, der die Eingabe der Quelldatei verlangt.

Der Compiler erzeugt ein sogenanntes Link-Modul und kein fertiges Maschinenprogramm. Bevor das C-Programm gestartet werden kann, ist es daher noch nötig das Programm zu Linken. Beim Linken wird der endgültig ablauffähige Code erzeugt, das heißt, das Programm erhält noch zusätzliche Module beispielsweise für die Ein-/Ausgabesteuerung.

Die RAM-Floppy

Die beim C 128 integrierte RAM-Floppy wird beim Laden des Kommandoprozessors automatisch in der Speicherbank 1 installiert und ist bereits mit den wichtigsten Dateien, die zum Compilieren notwendig sind, geladen. Das bedeutet im Klartext, daß für die Programmierung nur noch die 64 KByte der Speicherbank 0 zur Verfügung stehen. Trotzdem wird man nach einiger Zeit die RAM-Floppy nicht mehr missen wollen. Sofort in die RAM-Floppy geladen werden die C-Standardbibliothek »Stdio.h«, in der wichtige Terminalfunktionen definiert sind, sowie die für den Linklauf nötigen Libraries. Eigene Programme können ebenfalls in der RAM-Floppy gespeichert werden. Das hat vor allem in der Testphase den Vorteil, daß das zu testende Programm sehr schnell in den Editor geladen werden kann. Auch die anschließende Compilierung läuft unter Verwendung der RAM-Floppy wesentlich schneller ab. Der gesamte Inhalt der RAM-Floppy kann mit einem speziellen Kommandoprozessor-Befehl auf Diskette gespeichert und auch wieder geladen werden. Allerdings wird der Inhalt als eine komplette Datei abgelegt. Wenn diese Datei wieder eingelesen wird, entwirrt der dazu verwendete Befehl die Daten wieder zu den ursprünglichen Dateien. Will man einen Quellcode von der RAM-Floppy auf Diskette bringen, so kann man dies ohne Probleme über den Editor verwirklichen.

Der Sprachumfang

Profi C unterstützt fast den gesamten Sprachumfang von C einschließlich selbstdefinierter Datentypen und enthält zusätzlich einige gerätespezifische Erweiterungen. So können beispielsweise beim C 128 die beiden Kontrollregister des VDC-Chips, der für die 80-Zeichen-Steuerung verantwortlich zeichnet, über eigene Befehle angesprochen werden. Was vom Programmierer von Dateiverwaltungen schmerzlich vermißt wird, ist der wahlfreie Zugriff auf Daten. Die Handhabung von sequentiellen Dateien dagegen ist vollständig implementiert. Sogar das Auslesen des Fehlerkanals macht anhand der anschaulichen Beispiele im Handbuch keine Schwierigkeiten. Ansonsten steht dem Anwender eine umfangreiche Bibliothek an Funktionen zur Verfügung, mit der es durchaus möglich ist, auch größere Programmier-Probleme zu lösen.

Was auf jeden Fall sehr positiv zu Buche schlägt, ist die umfangreiche Bibliothek an Grafikbeehlen. Das auf Diskette mitgelieferte Demo-Programm zeigt nach dem Übersetzen eine kleine Auswahl der umfangreichen Grafikmöglichkeiten (Balkendiagramm, Analoguhr). Wer also gerne strukturiert programmiert und dazu noch Freude an Computer-Grafik hat, ist mit Profi-C bestens bedient. Der Systemprogrammierer schätzt an C vor allem die Möglichkeit, mit Zeigern arbeiten zu können. Mit Zeigern können konkret Speicherstellen angesprochen und bearbeitet werden, was C sehr stark in die Nähe der Maschinenspracheprogrammierung rückt. Wer also gerne strukturiert und trotzdem maschinennah programmieren möchte, hat hier eine leistungsfähige Methode in der Hand, diesen alten Traum zu verwirklichen. Allerdings sei hier darauf hingewiesen, daß beim Arbeiten mit Zeigern größte Sorgfalt anzuwenden ist. Werden Speicherzellen direkt manipuliert, kann es bei Unachtsamkeit leicht vorkommen, daß das Programm zerstört wird.

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-