

Comal 80 — die universelle Programmiersprache

Hand aufs Herz: Wie oft haben Sie sich als C 64-Programmierer schon über die Unzulänglichkeiten des eingebauten Basic 2.0 geärgert und sich ein komfortableres Basic gewünscht? Diese Zeiten sind jetzt endgültig vorbei. Mit Comal 80 nutzen Sie die enormen Hardware-Fähigkeiten des C 64 voll aus, und zwar ohne Maschinensprache, ohne PEEKs und POKEs, dafür schneller, komfortabler und übersichtlicher, als Sie es sich je vorgestellt haben. Comal vereinigt die leichte Erlernbarkeit von Basic mit der übersichtlichen Strukturierung von Pascal und den grafischen Fähigkeiten von Logo. Das Ergebnis ist eine neue Sprache, die alles, was es bisher an Programmiersprachen für den C 64 gibt, in den Schatten stellt.

64 KByte im ROM-Modul

Comal 80 ist für den C 64 nur als Steckmodul erhältlich. Dieses Modul enthält 64 KByte ROM, die in vier Speicherbänken zu je 16 KByte organisiert sind. Die eingebaute Bank-Auswahl-Logik schaltet ganz nach Bedarf zwischen den vier Comal-ROM-Bänken, dem Basic-ROM und dem darunterliegenden RAM hin und her. Bild 1 zeigt die Speicheraufteilung des C 64 unter Comal 80. Der gesamte adressierbare Speicherbereich (RAM und ROM) beträgt 152 KByte (!), für Comal-Programme stehen davon 30 KByte zur Verfügung. Das entspricht dem auch bei Basic-Erweiterungen wie Simons Basic oder Exbasic Level II vorhandenen Programm-Speicherplatz. Dazu kommen nochmals 16 KByte RAM, die für selbstgeschriebene Spracherweiterung (User Packages) verfügbar sind. Wenn dieser Speicherplatz noch nicht ausreicht, kann sich Comal-Erweiterungen mit einem Umfang von bis zu 32 KByte entweder in Assembler oder in Comal selbst schreiben und in ein EPROM brennen — das Comal-Modul enthält zu diesem Zweck einen freien EPROM-Steckplatz.

Der bildschirmorientierte Editor des C 64 wurde in vielen Punkten noch weiter verbessert. Zahlreiche

Vergessen Sie alles, was Sie über die Programmierung Ihres C 64 wissen! Denn mit Comal 80 verwandelt sich der C 64 in einen neuen Computer.

Control-Funktionen wie »lösche ab Cursor-Position bis Zeilenende« oder »Cursor auf vorhergehendes Wort setzen« erleichtern die Programmeingabe. Spezielle Comal-Kommandos wie DEL (Zeilenbereich löschen), AUTO (automatische Zeilennummern-Vorgabe) oder RENUM (Neumerierung eines Comal-Programms) bringen großen Komfort bei der Programm-Eingabe und -Änderung. Die Funktionstasten sind mit sinnvollen Kommandos belegt, können aber vom Anwender beliebig umdefiniert werden.

Komfortabler Editor

Comal ist genaugenommen weder ein Interpreter noch ein Compiler, sondern eine Zwischenstufe. Comal-Programme werden vor der Ausführung (die interpretativ erfolgt) in zwei Stufen »vor-übersetzt«. Die erste Stufe ist die Umsetzung der Comal-Schlüsselwörter in sogenannte »Token«, also meist aus einem Byte bestehende Befehls-Abkürzungen. Eine solche Vorübersetzung findet auch beim normalen Basic 2.0 statt. Comal geht aber noch weiter: Wird ein Comal-Programm gestartet, dann führt Comal vor dem eigentlichen Programmablauf eine semantische Analyse durch, in deren Verlauf die effektiven Adressen von Variablen, Prozeduren und Funktionen ermittelt und gespeichert werden. Während des Programmablaufs braucht Comal daher nicht lange in Variablen-Tabellen zu suchen, sondern findet die gesuchte Adresse praktisch unmittelbar vor. Dadurch werden Ausführungsgeschwindigkeiten erreicht, die in der Regel erheblich über denen vergleichbarer

Basic-Programme liegen. Einen Compiler kann man bei Comal getrost vergessen, er würde keinen nennenswerten Geschwindigkeitsvorteil mehr bringen.

Von Basic zu Comal

In Comal 80 sind die besten Eigenschaften der drei Programmiersprachen Basic, Pascal und Logo zu einem sehr komfortablen Programmier-System zusammengefaßt. Auf der Kommando-Ebene braucht der von Basic kommende Comal-Anwender nur wenig umzulernen: Programme werden mit LOAD geladen, mit SAVE gespeichert und mit LIST gelistet. Auch Befehle wie NEW, RUN, AUTO, RENUM oder MERGE dürften dem Basic-Programmierer nicht unbekannt sein.

Viele im Programm-Modus verwendete Befehle sind an Basic angelehnt. Das geht so weit, daß Comal vielfach Basic-Befehle akzeptiert, wo eigentlich spezielle Comal-Schlüsselwörter stehen müßten. Ein Beispiel ist die FOR-Schleife, die in Comal statt mit NEXT mit ENDFOR abgeschlossen werden muß. Tippt man jedoch versehentlich »NEXT«, so macht das nichts aus: Das Comal-System ist intelligent genug, um zu erkennen, was der Benutzer mit NEXT meinte und ersetzt das NEXT stillschweigend durch ENDFOR.

Strukturiert wie Pascal

Comal 80 für den C 64 ist in vieler Hinsicht mit Pascal vergleichbar, ohne allerdings dessen in vielen Fällen unnatürliche Strenge und Kompliziertheit zu übernehmen. Comal versucht stets, dem Benutzer dort Arbeit abzunehmen, wo dies von einem intelligenten System auch getan werden kann. Beispielsweise zwingt Pascal dazu, alle Variablen, Konstanten oder Prozeduren vor der ersten Verwendung des entsprechenden Namens zu definieren. Bei Comal dagegen ist es gleichgültig, wo eine Prozedur im Programm definiert wird. Bevor das RUN-Kommando ausgeführt wird, ordnet Co-


```

25 INPUT ZAHL
30 PRINT SQR(ZAHL)
35 HANDLER
40 CASE ERR OF
45   WHEN 206
50     PRINT "nur Zahl einge-
ben !"
55   WHEN 1
60     PRINT "negati-
ve Zahl nicht erlaubt !"
65   OTHERWISE
70     PRINT ERR, ERRTEXT$
75 ENDCASE
80 ENDTRAP
    
```

Dieses Programm wartet auf die Eingabe einer Zahl und berechnet danach die Quadratwurzel aus dieser Zahl. Zwei Fehlermöglichkeiten sind dabei offensichtlich: Zum einen kann der Benutzer irrtümlich irgendwelche Buchstaben mit eingeben (Fehler 206, »numeric constant expected«), zum anderen darf das Argument der Quadratwurzel-Funktion nicht negativ werden (Fehler 1, »Argument Error«). Durch den TRAP-HANDLER werden diese beiden Fehler abgefangen und mit entsprechenden Meldungen quittiert. Falls wider Erwarten irgendein anderer Fehler auftreten sollte, so wird in Zeile 70 einfach die Fehlernummer samt dazugehörigem Fehler-Text übergeben. Wenn Sie sich an den englischen Fehlermeldungen stören sollten — Comal ermöglicht es Ihnen, alle Fehlermeldungen neu zu formulieren und abzuspeichern.

Comal und die 1541

Ein Schwachpunkt im C 64-System ist bekanntlich die Floppy-Station 1541. Die meisten C 64-Besitzer haben sich wohl zähneknirschend daran gewöhnt, daß das Auflisten des Directorys mit Programmverlust be-

straft wird und daß man zur Anzeige der DOS-Fehlermeldungen ein Programm benötigt. Mit Comal 80 hat es ein Ende mit dem Zähneknirschen: DIR oder CAT zeigen das Directory, PASS sendet Kommandos an die 1541. Der Fehlerkanal wird entweder sofort angezeigt — wobei dies mit Hilfe des TRAP-Handlers gesteuert werden kann — oder kann durch STATUS abgefragt werden. CREATE legt eine relative Datei an, mit GET\$ kann eine beliebige Anzahl von Bytes von der Floppy-Station eingelesen werden.

Auch in Richtung Programmier-technik hat man sich einiges einfalten lassen. Programmteile können mit CHAIN nachgeladen werden, wodurch Comal-Programme theoretisch bis zu 200 KByte lang sein können (30 KByte im RAM, 170 KByte auf Diskette). Außerdem können Comal-Prozeduren und -Funktionen als EXTERNAL deklariert werden. Dies bedeutet, daß im Programm nur der Prozedurkopf vorhanden sein muß. Die Prozedur selbst befindet sich in diesem Fall auf der Diskette und wird beim Aufruf automatisch nachgeladen und ausgeführt. Nach getaner Arbeit wird die Prozedur dann einfach wieder aus dem Speicher geworfen, um keinen Platz zu verschwenden.

Comal kann mit bis zu acht Laufwerken gleichzeitig arbeiten und unterstützt von vorneherein auch den Anschluß der »großen« Commodore-Laufwerke der 40xx- und 80xx-Serie.

Die Zusatz-Pakete

Natürlich läßt sich auch mit der Datasette arbeiten, allerdings mit den von Basic bekannten Einschränkungen hinsichtlich Bedienungskomfort und Leistungsfähigkeit.

Bisher war nur von dem international genormten Grundwortschatz

von Comal die Rede. Es ist jedoch auch möglich, systemabhängige oder anwenderspezifische Sprach-erweiterungen zu integrieren. Eine solche Spracherweiterung wird in Comal als »Package« bezeichnet. Packages (Pakete) sind nichts anderes als Sammlungen von Comal-Prozeduren oder -Funktionen, die meist in Maschinensprache geschrieben sind, aber natürlich auch in Comal verfaßt sein können. Eine Reihe derartiger Spezial-Pakete sind in Comal 80 bereits fest eingebaut. Es bereitet aber nur wenig Mühe, sich weitere Packages selbst zusammenzustellen.

Ein fertig programmiertes Package kann auf Diskette abgelegt und bei Bedarf in einen reservierten 16-KByte-RAM-Bereich geladen werden, belastet also nicht den 30 KByte großen Comal-Programmspeicher. Fest eingebaut sind bereits elf Packages, mit denen sich die Fähigkeiten des Comal-Systems nochmals potenzieren:

1. English

Nach »USE ENGLISH« gibt Comal alle Fehlermeldungen im englischen Klartext aus.

2. Danish

Hiermit erscheinen alle Fehlermeldungen im dänischen Original.

3. Graphics

Dieses Package erweitert Comal um eine Reihe von speziell auf den C 64 zugeschnittenen Grafik-Befehlen.

4. Turtle Graphics

An diesem Package werden Logo-Freunde ihre helle Freude haben. Alle Grafikbefehle von Logo stehen hiermit zur Verfügung.

5. Sprites

Alles, was der C 64 in dieser Richtung hardwaremäßig zu bieten hat, ist hier in Form komfortabler Comal-Befehle verfügbar.

6. Sound

Es gibt wohl keine Funktion des SID-Chips, die nicht durch spezielle Comal-Befehle unterstützt wird. Als besonderer Komfort können Noten als Strings codiert und abgespielt werden. Dies geschieht interruptgesteuert, so daß Hintergrundmusik sehr einfach programmiert werden kann.

7. Paddles

Dieses Package enthält Prozeduren und Funktionen zur Abfrage der Controlports.

8. Joysticks

Hiermit wird die Joystick-Kontrolle zum Kinderspiel.

9. Lightpen

Wer einen Lightpen sein eigen nennt, der findet hier alle Spezial-



Bild 2. Ein Beispiel für die grafischen Fähigkeiten von Comal. Verschiedene Windows mit Text und Grafik wurden definiert.

Fortsetzung auf Seite 164

