

COMMODORE DISC C64/ C128

Nr. **17** DM 19,80 ÖS 168,- SFR19,80

Unverbindliche Preisempfehlung

C64:

**Zurück in die Pionierzeit:
Eisenbahn
Bildschirm total:
Rand & Sprites
Tauscht Grafiken:
Grafikkonverter
'C' mit dem C64:
Apfelberge
Prozessor-Sprache:
Assembler-Kurs 1
Zwölf Funktionstasten:
Key**

128 PC:

**Relationale Datei:
Check-Liste
Multi-Datenverwaltung:
Büro
Datei auf Diskette:
Stardat
Mailbox-Abenteuer:
Die verlorene Information**

**Zehn
Commodore
Programme
auf Disc
im Heft**

**164 KByte
ohne
Abtippen!**



**Alle Programme
auf Disc im Heft!**

INHALT COMMODOR

CD Intern

GUTEN TAG, LIEBE LESER DER COMMODORE DISC!

Sie laufen und laufen und laufen . . .

Frohe Kunde aus der Commodore-Hauptverwaltung in Frankfurt: Bislang werden kontinuierlich Monat für Monat etwa 20.000 neue C64 und rund 6.000 C128 D in Deutschland verkauft. Aufs Jahr hochgerechnet, ergibt das über eine Viertelmillion Commodore-Homecomputer der Acht-Bit-Klasse. Grund genug für uns, weiterhin an unserem Grundsatz festzuhalten: Software, Themen und Berichte nicht nur für Fortgeschrittene, sondern — und vor allen Dingen — für Einsteiger, die sich vorgenommen haben, ab sofort die Kommunikation mit dem Computer aufzunehmen. Daß es nicht immer nur BASIC sein muß, wird Ihnen schnell klar werden, wenn Sie im Begleitmagazin unsere ausführliche Themenauswahl über C, die Programmiersprache der Zukunft, gelesen haben. Dafür brauchen Sie auch keinen großen PC oder Amiga, es geht ebenso mit den „Kleinen“.

„C“ — DIE SPRACHE DER ZUKUNFT?

Noch eine andere Computer-Sprache soll uns ab Erscheinen dieser Ausgabe am Herzen liegen: Die „Ursprache“ des C64 oder 128PC, nämlich seine eigene, besser bekannt als Maschinensprache und Assembler. Einer un-

serer freien Autoren, Ralf Trabhardt, hat sich vorgenommen, allen wissensdurstigen Lesern mit einem leicht verständlichen Kurs auf der COMMODORE DISC die Sprache der Mikroprozessoren 65xx/85xx näherzubringen. Daß ihm das ausgezeichnet gelungen ist, erkennen Sie bereits im Teil 1 dieses Assembler-Kurses auf Diskette.

BASIC IST BEILEIBE NICHT OUT

Natürlich wird für alle, denen zunächst mal BASIC am Herzen liegt, auch weiterhin Wissenswertes auf DISC gespeichert und im Heft abgedruckt. Etwa, wie Sie Ihre BASIC-Programme so gestalten können, daß der BASIC-Interpreter weniger Arbeit hat und dadurch schneller wird.

DER COMPUTER SPRICHT

Wer's lieber mit „echter“, für das Ohr verständlicher, Sprache hält, sollte sich den Testbericht über ein neu auf dem Markt erschienenenes Digitizer-Modul für den C64 durchlesen, bei dem Sie mit Hilfe eines Kassettenrecorders und Audiokassette Töne und Sprache in erstaunlicher Qualität als Sound-File in den Computer bringen können. Auch das Wort „Dialog“ hat mit Kommunikation zu tun. Hier haben Sie als Leser die Möglichkeit, alles, was Ihnen an uns paßt (oder auch nicht) oder was es über Ihren Computer zu vermelden gibt, zur Sprache zu bringen. Bedenken Sie aber bei Ihrer Kritik und Ihren Verbesserungsvorschlägen, daß wir letztendlich alle unsere Leser zufriedenstellen wollen. Ob wir zu wenig C128-

Software oder zuviel C64-Programme veröffentlichten und ob es nur Programme für den 128PC im 80-Zeichen-Modus gibt, liegt nur in der Betrachtungsweise des Einzelnen. Was der eine als Mißstand betrachtet, findet der andere wieder ausgezeichnet. Der eine ist der Meinung, die COMMODORE DISC sei viel zu teuer. Ein anderer schreibt uns, gemessen an der Qualität verschiedener bisher erschienener Anwendungsprogramme für den C128 müsse er uns fairerweise mitteilen, daß die COMMODORE DISC viel zu preisgünstig sei. Und in diesem „Niemandland“ zwischen zwei total konträren Lesermeinungen sitzen die Redakteure. Kurzum, wir haben beschlossen, so weiter zu machen wie bisher. Allerdings: Die uns zugesandte Software, die dann auf DISC veröffentlicht wird, werden wir uns künftig noch genauer und intensiver unter die Lupe nehmen, als es bisher schon geschah, das versprechen wir Ihnen. Nichts ist ärgerlicher für Sie als Leser — und für uns als Redakteure —, wenn sich erst bei ganz bestimmten Konstellationen ein Programmfehler herausstellt, der nicht mal dem Programmeinsender aufgefallen ist. Obwohl er eigentlich der erste sein müßte, der so was merkt.

ZEHN PROGRAMME AUF DER DISC

Was gibt's noch auf der COMMODORE DISC? Zunächst mal eine hilfreiche Verbesserung des Lademodus, speziell für die Einsteiger unter unseren Lesern. Die Ladeanweisungen im Heft ge-

ben Ihnen erschöpfend Auskunft darüber, wie es künftig funktionieren wird.

Unsere Grafikfreunde dürfte das Programm „Apfelberge in C“ sowie der „Grafikkonverter“ interessieren, der C128-Grafiken zu C64-Bildern macht. Außerdem gibt es ein intelligentes Spiel von einem bewährten Mitarbeiter, Günter Kramer, der dazu den ganzen Bildschirm benutzt, also auch den Rahmen: PC-Feeling beim C64.

Die C128-Benutzer unter unseren Lesern sind überwiegend an Anwendungsprogrammen interessiert. Diesem vielfach geäußerten Wunsch haben wir selbstredend wieder Rechnung getragen. Die Datenverwaltungsprogramme Check-Liste, Büro und Stardat stellen dies deutlich unter Beweis. Beim zuletzt genannten Programm handelt es sich um ein Verwaltungsprogramm für relative Dateien, die nicht im Computerspeicher, sondern auf Diskette abgelegt werden. Nach geringfügigen Änderungen ist es auch im 80-Zeichen-Modus lauffähig.

Und zu guter Letzt gibt's zur Entspannung noch ein gelungenes Adventure-Spiel für den C128, das verblüffend echt einen Mailbox-Betrieb simuliert. Hoffentlich ist uns die Mischung wieder gelungen.

Ob's ein Erfolg war, werden, wie in der Vergangenheit, Ihre Anrufe zur Hot-Line-Zeit (jeden Mittwoch zwischen 15.00 und 19.00 Uhr) und Ihre Briefe beweisen. Es bleibt uns also nur noch, Ihnen viel Spaß zu wünschen.

Ihr
COMMODORE-DISC-
Team

DISC Nr. 17 INHALT

TEST & TECHNIK

D.A.I.S.Y.

Sprache und Klang digitalisieren und auf dem C64 erklingen lassen.

Seite 12

BÜCHER UNTER CP/M

Hilfreiche Tips & Tricks zum Arbeiten mit dem „dritten“ Betriebssystem des 128PC

Seite 23

AUF DISC IM HEFT

DIE VERLORENE INFORMATION – THE LOST INFORMATION

Ein Adventure, so ganz nach dem Geschmack von „Hackern“ und Mailbox-Freaks, jedoch ohne Host-Rechner.

Seite 4

STAR-DAT

Das Beispiel einer relativen Datei, das Ihnen obendrein noch seltene Autogramm-Adressen präsentiert.

Seite 4

BÜRO

Ein Multi-Dateiverwaltungsprogramm, aus mehreren unabhängigen Modulen zusammengesetzt.

Seite 7

EISENBAHN

Kehren Sie zurück in die Zeit der Eisenbahn-Pioniere und bauen Sie sich Ihre eigene Strecke.

Seite 8

APFELBERGE IN „C“

Grau ist alle Theorie, sehen Sie, was „C“ auf dem C64 kann.

Seite 10

RAND & SPRITES

So nutzen Sie den ganzen(!) Bildschirm beim C64.

Seite 11

GRAFIK-KONVERTER

128PC – C64

C128-Grafiken in den C64 laden und zur Weiterverarbeitung abspeichern.

Seite 11

ASSEMBLER-KURS 1

Der erste Teil unseres Maschinensprachekurses auf Diskette.

Seite 14

KEY

So können Sie die Funktionstasten mit zwölf verschiedenen Texten belegen.

Seite 15

CHECK-LISTE

Lesen und sehen Sie, was den Vorteil einer relationalen Dateiverwaltung ausmacht.

Seite 16

SERIE & SERVICE

LOAD & RUN

So laden und starten Sie Programme von der COMMODORE DISC – noch schneller, noch bequemer!

Seite 22

BASIC MUSS NICHT LANGSAM SEIN

Eine „Homage“ an eine Programmiersprache, die oft hochnäsiger abqualifiziert wird und für den Einsteiger doch die idealste ist.

Seite 24

„C“ – DIE SPRACHE DER ZUKUNFT?

Was ist eigentlich „C“, woher stammt es, was kann ich damit machen?

Seite 27

```

"BOOT.64",8,1:          PRG "CHECK-LISTE":      PRG
"DISCLADER64/128"     PRG "MENUE"              PRG
"-----"            USA "BUERO 128":          PRG
"EISENBAHN",8:        PRG "INFO BUERO"         SEQ
"RAND&SPRITE",8:     PRG "INFO DATEI"       SEQ
"R&S"                 PRG "INFO KOMBI"       SEQ
"EB"                  PRG "INFO COUNT"       SEQ
"GRAFIK128-64",8:    PRG "INFO TEXTE"       SEQ
"P128"                PRG "INFO SEQ.FILE"      SEQ
"STARTROTTER"        PRG "STARDAT":          PRG
"CONQUEST"           PRG "AUTOGRAMM.R"      REL
"E.T."               PRG "AUTOGRAMM.S"      SEQ
"APFELBERGE",8:      PRG "LOST INFORMAT.": PRG
"APFEL/C.OBJ"        PRG "-----"          USA
"ASSEMB.KURS1",8:    PRG "(C) COPYRIGHT BY" USA
"MC-ROUTINEN 1"      PRG "CA - VERLAG GMBH" USA
"KEY",8:             PRG "-----"          USA
"-----"            USA 23 BLOCKS FREE.

READY.
```

Die Diskette in diesem Heft ist weder list- noch kopiergeschützt. Aus verständlichen Gründen können wir daher bei Programmfehlern lediglich Umtauschrecht einräumen. Das Rückgaberecht gegen Kaufpreiserstattung ist ausgeschlossen! Sollte also eines der Programme auf Ihrer Diskette nicht laufen, senden Sie die Diskette an den Verlag zurück, Sie erhalten selbstverständlich eine korrigierte Fassung. Anschrift: Siehe Impressum.

THE LOST INFORMATIONS

Hacker auf Agenten- Jagd

In diesem gelungenen Mailbox-Adventure sind Ihre detektivischen Fähigkeiten gefordert. Spüren Sie als Hacker verlorengegangene CIA-Dokumente wieder auf.

Stellen Sie sich folgende Situation vor: Eines Abends schalten Sie Ihren C128 im 80-Zeichen-Modus ein, ein fröhlich Lied vor sich hinpfeifend – und nichts tut sich. Warten Sie aber noch einen Moment, bevor Sie den RESET-Taster betätigen, denn plötzlich bewegt sich doch etwas auf Ihrem Bildschirm. Neugierig geworden, lesen Sie den Text, der da auf einmal erscheint. Nun fällt es Ihnen siedendheiß ein: Sie haben gestern Abend bei Ihrem Streifzug durch verschiedene Mailboxen Ihr Modem nicht ausgeschaltet und jetzt hat sich eine fremde Organisation in Ihren C128 eingeschlichen.

Als Mailbox-Freak oder -User ist Ihnen dieser Vorgang nicht neu. Für alle anderen ist dieses Adventure-Spiel eine sehr gut gelungene Simulation eines Mailbox-Betriebes auf dem Computer. Sie sollen für den amerikanischen Geheimdienst CIA geheime Informationen, die verlorengingen, als Mailbox-Hacker zurückholen.

Nach dem Titelbild erscheint die Frage nach dem Logon-Paßwort, um zu prüfen, ob Sie den Anforderungen überhaupt gewachsen sind. Sie sind es bestimmt, wenn Sie HILFE eintippen. Das Paßwort wird Ihnen dann verraten.

Die verschiedenen Mailboxen, die das Programm simuliert, werden über ein Menü gesteuert.

Wenn Sie glauben, den Schuldigen gefunden zu haben, so rufen Sie die Hilfsseite aus dem Hauptmenü auf und starten den entsprechenden Punkt (4). Außerdem können Sie mit der Hilfsseite die Geschwindigkeit der Textausgabe, das Tempo und ähnliches einstellen.

TIPS ZUM SPIEL

Das erste wichtige Paßwort erfahren Sie, wenn Sie HILFE eintippen. Oft hilft auch, wie im realen Mailbox-Betrieb, die Eingabe von USER. Wenn Sie die LADA-Mailbox gewählt haben, sollten Sie sich bei der Frage nach dem Paßwort überlegen, was ein LADA überhaupt ist. Und schließlich ist es – wie im richtigen Leben – oft sehr hilfreich, eine „kleine Geldspende“ zu gewähren.

Mehr möchten wir nicht verraten, Sie werden sicher auch ohne unsere Hilfe dieses Mailbox-Adventure lösen. So schwer ist es auch wieder nicht.

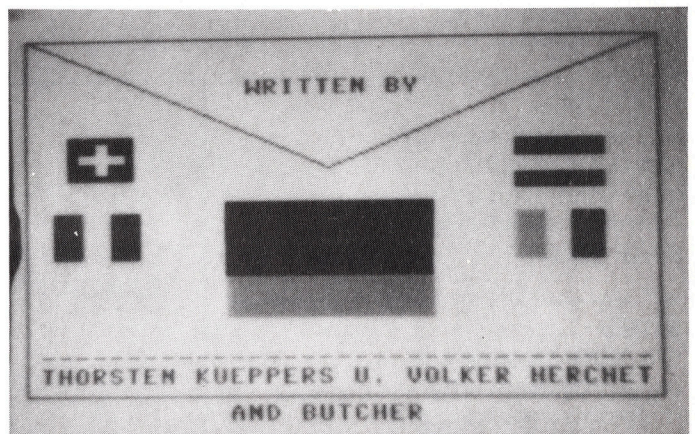
Georg Droschl/tc

STAR DAT

Kunst kommt von Können

Computern ist ein Hobby, Autogramme sammeln ein anderes. Wie Sie beides unter einen Hut bringen, soll Ihnen dieses Programm für den C128 im 40-Zeichen-Modus zeigen.

Zwei Schüler aus Mönchengladbach, außer C128-Freaks auch begeisterte Autogrammsammler, hatten eine gute Idee: Mit Hilfe ihres Computers schufen sie sich eine Datenbank von Autogramm-Adressen bekannter und beliebter Film- und Sport-Stars, von Sylvester Stallone bis Steffi Graf. Diese wollten sie interessierten Lesern der COMMODORE DISC zur Verfügung stellen. Ganz nebenbei entstand dadurch



noch eine recht komfortable Dateiverwaltung für den C128, die nach den Kriterien einer relativen Direktzugriffs-Datei auf Diskette programmiert wurde. Sie kann nicht nur für Adressen, sondern auch als Verwaltungsprogramm mit Möglichkeit zum Ausdrucken für Video- und Audiokassetten, Kunden- und Geschäftsadressen verwendet werden. Es sind lediglich einige kosmetische Eingriffe nötig, um sie auf die Bedürfnisse des einzelnen Anwenders anzupassen.

BEDIENUNGSANLEITUNG

Nach dem grafisch gut gelungenen Titelbild erscheint das Hauptmenü mit folgenden Auswahlpunkten:

- 1 Datei öffnen,
- 2 Dateneingabe,
- 3 Datenausgabe,
- 4 Datei schließen,
- E Programmende,
- D Directory,
- A Namensliste.

Durch Druck auf die entsprechende Taste werden die Funktionen ausgewählt.

1. Datei öffnen

Beim erstmaligen und bei jedem Arbeitsbeginn mit diesem Dateiprogramm ist unbedingt dieser Punkt anzuwählen. Es wird eine REL-Datei auf Diskette eröffnet: beim erstmaligen zum Schreiben und wenn sich die REL-Datei „Autogramm.R“ bereits auf Diskette befindet, zum Lesen.

Wählen Sie einen der anderen Punkte von Zwei bis Vier, ohne vorher eine Datei mit Punkt eins geöffnet zu haben, werden Sie keinen Erfolg haben, sondern nur eine Fehlermeldung erhalten.

2. Dateneingabe

Wenn Sie das erstmal dieses Programm benutzen, werden Sie jetzt diesen Menüpunkt anwählen. Es erscheint die Eingabemaske auf dem Bildschirm. Sie setzt sich aus den einzelnen Datenfeldern zusammen, zu denen Sie Ihre Eintragungen machen sollen. Natürlich läßt sich dieser Programmpunkt auch aufrufen, wenn sich, so wie in diesem Fall, bereits eine relative Datei (Autogramm.R) auf Diskette befindet. Die neuen Adressen, die Sie jetzt eingeben, werden einfach angehängt.

Eines müssen Sie bei der Dateneingabe beachten: Das Programm ist so konzipiert, daß es nur eine Länge von 25 Zeichen je Datenfeld (Name, Vorname, Straße) zuläßt. Sind es mehr, wird zunächst die Maske auf dem Bildschirm „zerhackt“, das heißt, die anderen Datenfelder scrollen eine Bildschirmzeile nach unten. Das ist aber kein Beinbruch. Mit Druck auf die RETURN-Taste erhält die Bildschirmausgabe wieder ihr vorheriges Aussehen, lediglich von Ihrem überlang eingegebenen Datenfeld werden nur die ersten 25 Zeichen verwendet. Wie Sie so eine verunglückte Eingabe eines Datenfeldes wieder korrigieren können, lesen Sie unter Menüpunkt drei. Daß Ihre Eingaben alle akzeptiert wurden, zeigt Ihnen die Statuszeile am unteren Bildschirmrand mit der Meldung „Datensatz ist gesichert“.

3. Daten ausgeben

Alles, was Sie in den C128 eingegeben haben, möchten Sie auch wieder lesen. Das geschieht mit dieser Funktion. Damit aber Ihr Computer weiß, welche Adresse oder welchen Datensatz er anzeigen soll, müssen Sie ihm ein Suchmerkmal mitteilen.

Wir haben uns beim Programm Stardat auf das Datenfeld eins (Name) festgelegt, was ja auch am naheliegendsten ist. Wenn Sie also die Autogrammadresse von Boris Becker erfahren möchten, so geben Sie bei der Frage nach dem Suchbegriff BECKER ein. Achten Sie auf die Großschreibung, da die Namen auch versal eingegeben wurden. Nun sucht der C128 in der gleichzeitig mit der relativen Datei angelegten sequentiellen Suchdatei, ob dieser Name vorhanden ist. Wenn nicht, erhalten Sie die Meldung „Datensatz nicht vorhanden“. Ansonsten liest der Computer aus der SEQ-Suchdatei „Autogramm.S“ die zum Namen gehörige Nummer des Datensatzes und kann ihn auf Grund dieser Zahl auf Diskette finden und auf dem Bildschirm anzeigen.

Steht die Adresse vor Ihnen, wird Ihnen auffallen, daß sich die Menüleiste am oberen Bildschirmrand verändert hat.

Folgende Möglichkeiten zur Weiterverarbeitung dieses speziellen Datensatzes stehen nun zu Ihrer Verfügung: *W = weitersuchen*: Damit können Sie sich einen anderen Datensatz zeigen lassen. Das ganze Spiel läuft

nun so ab, als hätten Sie aus dem Hauptmenü heraus 3. *Datenausgabe* gewählt.

E = editieren: Nach Wahl dieses Punktes läßt sich praktisch der gesamte Datensatz ändern oder gar neu überschreiben, so wie Sie es aus dem Hauptmenü mit Punkt 2 *Dateneingabe* gemacht hätten.

D = drucken: Stellen Sie sicher, daß Ihr Drucker (Commodore oder kompatibel) angeschlossen und eingeschaltet ist. Sie können die Sekundär-Druckeradresse ändern, voreingestellt ist 7 für Klein-/Großschrift. Nach Druck auf die RETURN-Taste beginnt der Ausdruck des Datensatzes inklusive der Datenfeld-Bezeichnungen.

Mit der Taste M gelangen Sie wieder ins Hauptmenü zurück, der bislang angezeigte Datensatz wird aber gelöscht.

4. Datei schließen

Diese Menüfunktion ist mindestens ebenso wichtig wie Punkt eins. Haben Sie neue Adressen eingegeben oder bereits bestehende verändert, eine existierende Datei erweitert – all Ihre Arbeit wird erst dann unwiderruflich auf Diskette dokumentiert, wenn zum Beenden Ihrer Dateiarbeit die Taste 4 gedrückt wird. Die REL-Datei wird ordnungsgemäß geschlossen und die SEQ-Suchdatei mit ihrem aktuell gültigen Inhalt auf Diskette zurückgeschrieben. Eine bislang existierende SEQ-Datei wird einfach gelöscht. Alle offenen Datenkanäle, drei an der Zahl, werden wieder geschlossen. Wählen Sie also nie E für *Programmende*, wenn Sie eine neue Datei entworfen oder eine bereits bestehende Datei überarbeitet haben, ohne vorher die Taste 4 gedrückt zu haben.

E. Programmende

Wann Sie diese Funktion nicht benutzen dürfen, haben wir bereits im vorhergehenden Punkt erwähnt. Allerdings ist dieser Menüpunkt sehr wohl nützlich, wenn Sie sich nur die Autogrammdatei geladen und einen oder mehrere Datensätze angesehen oder ausgedruckt haben, ohne an den bestehenden Daten etwas zu verändern. Dann genügt es vollkommen, wenn nur die geöffneten Files „Autogramm.R“ und „Autogramm.S“ geschlossen werden und das Programm in den Direktmodus zurückkehrt. Es wird aber kein RESET durchgeführt. Stardat befindet sich noch immer im Speicher, bis Sie es mit NEW löschen oder ein anderes BASIC-Programm laden.

◀◀◀ SSS ▶▶▶ Siggis Software Shop ▶▶▶ SSS ▶▶▶					
★ Knüllerpreise ★ Ein Preisvergleich lohnt sich immer ★ Knüllerpreise ★					
Spiele	Amiga/ST	Spiele	Amiga/ST	Spiele	Amiga/ST
Arctic Fox	54,50/54,50	Jump Jet	42,50/42,50	Superstar Icehock	64,50/64,50
Arkanoiid	64,50/42,50	Karting Grand Prix	26,50/26,50	Terramex	54,50/54,50
Backlash	45,50/44,50	King of Chicago	58,50/58,50	Test Drive	74,50/74,50
Bad Cat	49,50/54,50	Las Vegas	26,50/26,50	Tetris	54,50/54,50
BMX Simulator	42,50/44,50	Leaderboard Golf	64,50/64,50	Time & Magic	54,50/54,50
California Games	64,50/64,50	Lurkins Horror	74,50/74,50	Two & Two Basketb.	64,50/64,50
Chessmaster 2000	72,50/74,50	Moebius	64,50/64,50	Willy the Kid	26,50/26,50
Clever & Smart	54,50/56,50	Phantasia 3	54,50/58,50	Winterolympiade 88	56,50/56,50
Defender of Crown	66,50/64,50	Ports of Call	88,50/—	Wizball	54,50/54,50
Dungeon Master	64,50/64,50	Planetfall	—/—/74,50	Xenon	54,50/54,50
Eagles Nest	54,50/56,50	Rings of Zithin	64,50/64,50		
Emerald Mine	26,50/26,50	Roadwar 2000	54,50/54,50	Anwenderprogramme	Amiga/ST
Fam. Feuerstein	54,50/54,50	Roadwar Europa	68,50/68,50	C-64 Emulator	156,50/—
Flight-Simulator 2	99,50/98,50	Roadwars	54,50/54,50	Cambridge Lisp	—/—/358,50
Frightnigh	64,50/64,50	Rolling Thunder	64,50/54,50	Lattice C	528,50/—
Ferrari Formula 1	72,50/—	Sindbad	68,50/58,50	Lisp	478,50/—
Giana Sisters	52,50/54,50	Slaygon	54,50/54,50	Macro Assembler	178,50/—
Gridstart	26,50/26,50	Soccer King	26,50/26,50	MCC Pascal	—/—/248,50
Hotball	64,50/64,50	Strip Poker 2 Plus	42,50/44,50	Pascal	248,50/—
Iridon	42,50/—	Sub Battle Sim.	64,50/64,50	Pro Sprite Designer	—/—/99,50

S. Gebauer
Parkstr. 7a
5880 Lüdenscheid
Tel. (0 23 51) 2 45 02

◀◀ SSS ▶▶▶
◀◀ SSS ▶▶▶
◀◀ SSS ▶▶▶
◀◀ SSS ▶▶▶

Versandkosten: Vorkasse + DM 4,50 / Nachnahme + DM 7,50. Zur Auslieferung gelangt ausschließlich nur Originalware. Angebote freibleibend. Liefermöglichkeiten vorbehalten. Bei großer Nachfrage nicht jeder Artikel sofort lieferbar.

D. Directory

Zu dieser Funktion, die wohl jedem gut bekannt ist, brauchen wir sicher nichts weiter zu sagen. Allerdings: Da das Einlesen des DIRECTORY ebenfalls einen Datenkanal der Diskette benötigt, nach Wahl von Punkt eins (Datei öffnen) aber bereits die drei möglichen offen sind, sollten Sie diese Funktion nie anwählen, wenn Sie a) die Datei geöffnet oder b) mit Taste 4 noch nicht geschlossen haben. Dann erhalten Sie nämlich kein Disketten-Inhaltsverzeichnis, sondern nur Stuß.

A. Namensliste

Kein Mensch kann sich merken, welche Adressen er über einen längeren Zeitraum hinweg eingegeben und gespeichert hat. Trotzdem ist die Kenntnis der Suchbegriffe unabdingbar, will man sich die Datensätze zum Ansehen oder Ausdrucken auf den Bildschirm holen. Das erledigt diese Menüfunktion. Bei einer bereits bestehenden Datei geschieht das nach dem Öffnen (Punkt eins) automatisch.

Nachdem die SEQ-Suchdatei, auf der COMMODORE DISC „Autogramm.S“, eingelesen wurde, werden die relevanten Suchbegriffe auf dem Bildschirm ausgegeben. Damit wissen Sie genau, wie dieses Merkwort oder auch die Merzkahl lautet.

Vorher haben Sie noch die Wahl, ob Sie sich diese Daten sortiert (bei Namen sehr zu empfehlen) oder unsortiert, also in der Reihenfolge, wie sie gespeichert wurden, ausgeben lassen möchten. Wählen Sie S für „sortiert“, so wird der C128 in den FAST-Modus geschaltet, um den Sortiervorgang zu beschleunigen. Dadurch wird aber auch während dieser Zeit der Bildschirm abgeschaltet, Ihr Computer ist also nicht abgestürzt.

Das Programm verwendet die schnellste Sortieroutine in BASIC, Quicksort. Trotzdem dauert es je nach Größe der Datei seine Zeit, 100 Datensätze benötigen etwa eine Minute. Ist der Sortiervorgang beendet, meldet sich der Bildschirm wieder, und Sie bekommen die Namen, alphabetisch sortiert, aufgelistet. Diesen Ausgabevorgang können Sie jederzeit aus dem Hauptmenü, etwa zum Beispiel vor der Datenausgabe (Punkt drei), erneut aufrufen, diesmal aber ohne Sortierfunktion. Haben Sie die Daten bereits zu Anfang sortieren lassen, so werden sie später ohne Verzögerung auf Ihrem Bildschirm ausgegeben. Sollten Sie sich bereits zu Beginn für „unsortiert“ entschieden haben, ist allerdings eine nachträgliche Sortierung nicht mehr möglich.

WAS BRINGT EINE RELATIVE DATEI-VERWALTUNG?

Die relative Datei ist eine Art der Dateiverwaltung, die direkt auf Blöcke, Spuren und Sektoren der Diskette zugreift. Besser bekannt und auch einfacher zu programmieren ist sicher die Datenablage in einer sequentiellen Datei. Sie hat aber einen sehr großen Nachteil: Sie ist vom Speicherplatz des Computers abhängig. Zum Lesen oder Bearbeiten muß sie sich vollständig im Computerspeicher befinden, was sich natürlich auch beim Suchen recht zeitintensiv auswirken kann.

Eine REL-Datei hingegen schreibt jeden Datensatz nach ganz bestimmten Gesichtspunkten auf Diskette, je nach dem aktuellen Blockbelegungsplan (BAM) der Arbeitsdiskette beim Anlegen einer REL-Datei. Daraus können Sie schließen, daß sämtliche Blöcke

einer Arbeitsdisk als Speicher benutzt werden können. Bei einer Floppy 1541/1570 sind das immer 664 Blöcke (etwa 167.000 Bytes), bei der 1571 sogar das Doppelte. Eine sequentielle Datei diesen Ausmaßes ist völlig undenkbar, da sie nie in den doch wirklich großen Speicherplatz des C128 passen würde. Eine REL-Datei funktioniert wie ein Dateikasten, aus dem Sie die gewünschte Karteikarte jederzeit nach Bedarf ziehen können, ohne erst den gesamten Dateikasten durchsuchen zu müssen. Das trifft den Unterschied zwischen relativer und sequentieller Datei aufs I-Tüpfelchen.

Einen Nachteil wollen wir allerdings nicht verschweigen: SEQ-Dateien lassen sich jederzeit mit einem Kopierprogramm für einzelne Files auf eine andere Diskette übertragen, da es sich nur um eine Ansammlung von Bytes handelt, die an jeder Stelle auf der Diskette abgespeichert sein kann. Die REL-Datei dagegen sucht sich ganz bestimmte Stellen auf der Diskette aus. An diesen Plätzen, und nur an diesen, verewigt sie sich unauslöschlich; außer, Sie formatieren die Disk.

Wenn Sie allerdings ein Kopierprogramm für ganze Disketten verwenden, sieht die Sache anders aus. Denn damit erstellen Sie ein „spiegelbildliches“ Duplikat einer Diskette, die dieselbe Spuren-, Sektor- und Blockbelegung aufweist. Außerdem sollten Sie noch eines unbedingt beachten: Die REL-Datei „Autogramm.R“ auf dieser COMMODORE DISC kann jederzeit mit Stardat gelesen werden. Wollen Sie sie aber verändern oder weitere Adressen anhängen, so müssen Sie an der rechten Seite, neben dem Etikett, eine Schreibkerbe anbringen.

ANPASSUNG DES PROGRAMMES FÜR EIGENE VERWENDUNG

Autogramme sammeln ist nicht jedermanns Sache, das wissen wir. Auch ein Titelbild in HIRES-Grafik wird von manchem nur als störend empfunden. Als geübter Programmierer, und das sind Sie doch, wird es Ihnen sicher nicht schwerfallen, das Programm Stardat als andere Datei einzusetzen. Den Namen können Sie ändern.

Listen Sie zunächst das Programm mit LIST und ändern das Programmlisting, wie wir es Ihnen hier vorschlagen. Möchten Sie auf die Grafik im Titelbild verzichten, so löschen Sie einfach die Zeilen vier bis 2290 mit DELETE 4-2290. In Zeile zwei steht nun die Höchstmenge der Datensätze, die Sie eingeben können (DM=700). Wir glauben, daß wir hier nicht zu niedrig, eher ein wenig zu hoch gegriffen haben. Rein rechnerisch müßte es jedoch gehen, wenn Sie berücksichtigen, daß jeder Datensatz in diesem Programm genau 226 Bytes, also etwas weniger als einen Block belegt. DX\$(9) in derselben Zeile bedeutet, daß genau neun Datenfelder für jeden Datensatz (RECORD) verwendet wurden.

Wollen Sie mehr als zehn, so müssen Sie diese neue Menge unbedingt dort eintragen.

RE ist die Länge eines jeden Datenfeldes innerhalb des RECORDs. Auch diese Zahl läßt sich ändern. Aber bedenken Sie, daß das Produkt aus *Anzahl der Datenfelder x Länge pro Feld (=RE)* nie eine Gesamtlänge von 254 überschreiten darf.

In Zeile drei können Sie den Filenamen für das REL- und SEQ-File, das das Programm auf Diskette anlegt, nach Ihren Wünschen gestalten: etwa *Adressen*, *Video* oder ähnliches.

Schließlich und endlich können Sie noch in den Zei-

AUF DISC IM HEFT

len 20020 bis 20100 die Bezeichnungen der Datenfelder (DF\$(x)) Ihren Wünschen anpassen. Wenn Sie auch hier mehr als zehn definieren, bitte nicht vergessen, noch eine zusätzliche DIM-Anweisung für DF\$(x) in Zeile zwei unterzubringen.

NACH DIESER ÄNDERUNG AUCH IM 80-ZEICHEN-MODUS LAUFFÄHIG

Nicht unbedingt erforderlich, aber einer ansprechenden Bildschirmausgabe unwahrscheinlich zuträglich, ist ein Löschen des Bildschirms ganz zu Beginn des neuen, von Ihnen geänderten Programms (zum Beispiel SCNCLR in Zeile eins oder Zeile 20 stehenlassen).

Und das wird alle Leser beruhigen, die „nur“ einen 80-Zeichen-Monitor besitzen: Nach diesen eben geschilderten Änderungen läuft das Programm auch anstandslos mit dieser Bildschirm-Darstellung.

Volker Herchet/Thorsten Küppers/B.U. □

BÜRO 128

Manager

Büro 128 ist ein modular aufgebaut und menügesteuertes Programmsystem. Es ermöglicht Ihnen durch die Kombination aufeinander abgestimmter, flexibler und kompatibler Einzelprogramme in Verbindung mit der Floppy und einem Drucker eine optimale Nutzung des C128 zur Dateiverwaltung im 80-Zeichen-Modus.

Das Programmsystem *Büro 128* meldet sich nach dem Start und dem Drücken einer beliebigen Taste mit dem Zentralmenü, das Sie aus dem laufenden Programm heraus immer mit der Fragezeichen-Taste anwählen können.

Drücken Sie zunächst die ASCII/DIN-Taste, dieses Programm benötigt den DIN-Modus.

Darunter sind die anderen Programmteile dieses Multi-Dateiverwaltungs-Programms mit den zu drückenden Tasten aufgelistet:

* = Datei,
\$ = Kombi,
% = Count,
& = Texte,
0 = Zentrum,
. = Funktion.

Die Erläuterungen zu den einzelnen Programmteilen (Modulen) sind als sequentielles Textfile ebenfalls auf der COMMODORE DISC abgespeichert, gekennzeichnet mit dem Vorspann INFO. Um sich den Text dieser Erklärungen auf dem Bildschirm anzeigen zu lassen, wählen Sie bitte '0' (=Zentrale), anschließend 'B' (Groß-B, also mit SHIFT) und geben dann den entsprechenden Filenamen ein, zum Beispiel info buero.

DIE EINZELNEN MENÜPUNKTE

Die Zentrale – mit '.' plus entsprechender Taste lassen sich nur ihre Funktionen anwählen – ist die Basis für direkte Floppyfunktionen wie:

- w = Textfiles im Direktmodus auf Diskette schreiben oder mit
- r = von Disk lesen;
- n = Textfiles neu eingeben;
- h oder H oder K = eine Hardcopy vom aktuellen Bildschirm erstellen;
- R = eine Datei oder ein anderes File auf Disk umbenennen;
- F = Floppy-Fehlerkanal auslesen und anzeigen;
- D oder d = Anzeigen des DIRECTORY;
- A oder ' = Ihren gewünschten Absender oder mit 'a' für einen Fensterumschlag oder mit '>' Zwischentitel eingeben und mit '<' ausdrucken;
- X = Speicher löschen.

MODUL DATEI (*)

Legen Sie die Felderzahl pro Datensatz jeweils über 'A' fest und bestimmen Sie die Bezeichnung der Felder.

Im Prinzip können Sie beliebig viele Felder wählen und benennen, was natürlich Einfluß auf die Anzahl der Datensätze hat, die Sie insgesamt eingeben dürfen. Das Programm hat eine maximale Anzahl von 21 Datenfelder vorgesehen.

Die Eingabe wird mit 'e' ausgewählt. Hierbei sind alle Interpunktions-Zeichen erlaubt. Wenn Sie sich allerdings für '1' entschieden haben, dürfen Sie keine Interpunktions-Zeichen verwenden, falls Sie etwa nur Texttabellen anlegen möchten. Beide Eingabefunktionen lassen sich mischen. Angelegt wird die DATEI mit 'A'.

Cursorbewegungen sind nur innerhalb der aktuellen Zeile, nicht aber nach oben oder unten erlaubt. Sind alle Felder beschrieben, können Sie mit 'n' Korrekturen direkt vornehmen.

Haben Sie eine DATEI gespeichert und möchten Sie diese in den Rechner laden, werden mit dem Ladevorgang alle anderen Module automatisch dimensioniert (DIM-Anweisung). Es darf nicht über 'A' neu initialisiert werden. Mit 'X' (über die Zentrale) können Sie einen RESET durchführen. Ansonsten lassen sich jetzt die anderen Programme anlegen oder die Daten in den Rechner laden, ohne die Module vorher neu dimensionieren zu müssen. Sie können dann zwischen den Programmen beliebig hin- und herschalten, alle Datenwerte bleiben erhalten.

MODUL KOMBI (\$)

Dieses Modul besitzt zehn fest vorgegebene und vordefinierte Felder, die Sie im Listing des Programmes jedoch individuell ändern und anpassen können. Der Vorteil der Kombinationsdatei besteht vor allem in der Möglichkeit, auch hier zwischen allen anderen Dateien und Programm-Modulen hin- und herschalten zu können, wenn Sie erst einmal eine DATEI geladen haben. „Kombi“ bietet Ihnen eine sinnvolle Ergänzung zur Hauptdatei (DATEI).

**Jeden Mittwoch
15-19⁰⁰ Hotline**



089/129 80 13

AUF DISC IM HEFT

Auch läßt sich hier die Druckeransteuerung individueller gestalten. Im vorliegenden Programm werden beispielsweise die Telefonnummer und andere individuelle Daten nur auf dem Bildschirm, nicht aber über den Drucker, ausgegeben.

Übrigens: Sollten Sie einmal unfreiwillig aus dem Programm-Modus rauskommen, mit "GOTO 100" finden Sie jederzeit ohne Datenverlust zurück.

MODUL COUNT (%)

Dies ist ein einfaches, aber nützliches Zahlenbearbeitungs-Programm – zum Beispiel bei der Rechnungserstellung oder bei der Vorbereitung Ihrer Buchführung.

Haben Sie alle Beträge mit Hilfe dieses Programm-Moduls ausgedruckt, können Sie mit der Taste 'M' noch die Mehrwertsteuer über den Drucker auswerfen lassen.

Bei diesem Programmteil – er läßt sich auch mit allen anderen Modulen kombinieren – ist es wichtig, daß die Eingabe exakt in der begrenzten Spaltenzahl erfolgt.

Beispiel:

Tag = 01.

Monat = Jan. oder Juni, also jeweils drei Spalten für den Tag und vier für den Monat. Die Eingabegrenze für den Titel liegt bei 14 Spalten. Bei den Beträgen dürfen Sie Stellen nach dem Komma nie mit ',' abtrennen, sondern mit dem amerikanischen Dezimalpunkt '.'.

Falsch: 145.80

Richtig: 145.80

MODUL TEXTE (&)

„Texte“ wird ebenso wie die anderen Programmteile über 'A' angelegt. Wenn Sie einen gespeicherten Text von Diskette laden wollen, müssen Sie mit 'J' neu dimensionieren, falls Sie nicht vorher eine Datei geladen haben. In diesem Fall antworten Sie mit 'n', da sonst alle anderen Daten gelöscht werden. Bei der Eingabe in „Texte“ sind alle Interpunktionszeichen erlaubt, mit der RETURN-Taste schaffen Sie eine Leerzeile.

Auch hier können Sie den Cursor in der Zeile beliebig, nicht aber vertikal bewegen. Jede Texteingabe wird mit RETURN beendet. Korrekturen erfolgen über Auswahl der Zeilennummer. Wollen Sie die gesamte Eingabe Ihres Textes beenden, so drücken Sie in der dem Textende folgenden Leerzeile den Punkt '.', sonst sind die Daten später verloren.

SEQUENTIELLE FILES

Diese spezielle Funktion bietet Ihnen die Möglichkeit, Texte direkt auf Diskette zu schreiben oder von Diskette einzulesen. Sie können dann auch über die Hardcopy-Funktion ausgedruckt werden.

Über die Zentrale ('O' oder '.') können Sie mit der Funktion 'B' und anschließend 'w' direkt einen Text oder alle anderen Zeichen Ihrer Commodore-Tastatur auf Diskette schreiben. Über 'r' (Read) lassen sie sich jederzeit wieder einlesen oder mit 'n' neu eingeben. Während der Eingabe können Sie mit dem Cursor horizontal und vertikal beliebig hin- und herfahren.

Diese Cursorbewegungen erleben Sie aber auch später auf dem Bildschirm wieder mit, zum Beispiel, wenn

Sie ein Wort korrigiert haben. Für die Ausgabe auf den Drucker hat das allerdings keine Bedeutung. Beendet wird Ihre Eingabe mit dem Pfeil nach links. *Büro 128* stellt Ihnen ein komfortables Werkzeug zum Erstellen von Text- oder Text-/Zahldateien jeder erdenklichen Art zur Verfügung.

Es benutzt zum Aufruf der verschiedenen Funktionen recht ausgiebig die verschiedensten Zeichen Ihrer Commodore-Tastatur. In den einzelnen Menüs wird ausdrücklich auf die Funktionsweise dieser Tastenzeichen hingewiesen.

Eines müssen Sie aber dabei beachten: Finden Sie ein Tastenzeichen großgeschrieben, so müssen Sie dies auch mit der SHIFT-Taste zusammen drücken, sonst passiert bei diesem Menüpunkt gar nichts.

Helmut Behrens-Hohmann/bu □

EISENBAHN

BOSS

Mit diesem Spiel, an dem bis zu vier Personen teilnehmen können, gehen Sie zurück in die Zeit der Eisenbahn-Pioniere.

Zum exakten Funktionieren benötigt das Programm das Maschinensprache-File *R&S*, das vom Ladeprogramm automatisch nachgeladen wird, ebenso das Hauptprogramm *EB*, das eigentliche Spielprogramm, das aber die neuen BASIC-Befehle der Erweiterung *R&S* recht ausführlich nutzt.

Die BASIC-Erweiterung *R&S* (Rand & Sprites), die die Fähigkeiten des VIC-Chip beim C64 ganz erheblich ausdehnt, finden Sie an anderer Stelle im Heft ausführlich erläutert, so daß Sie sie auch für eigene Programme nutzen können.

SPIELANLEITUNG

Sie sollen in einer vom Zufall erstellten Landschaft Ihre Eisenbahnlinien aufbauen, um im zweiten Teil des Spiels Ihren Reichtum durch Frachtbeförderung zu mehren. Und das alles am besten im Wettstreit mit bis zu drei weiteren Spielpartnern.

Nehmen weniger als vier Personen am Spiel teil, geben Sie dies nach dem letzten Spieler durch Eingabe einer Null bekannt.

Haben Sie Ihre Teilnehmerliste bestätigt, baut das Programm eine Landschaft auf, in der Flüsse, Moore und Berge vorhanden sind. In der Mitte des Spielfeldes erscheint die Hauptstadt des Landes, gekennzeichnet durch einen reversen Stern, außerdem weitere 15 Städte mit den Kennzeichen A bis O, ebenfalls revers. Wenn die Hauptstadt immer in der Mitte an derselben Stelle ist, so werden die anderen Städte jedesmal an wechselnden Stellen plaziert, so daß sich von Fall zu Fall verschiedene Landschaften ergeben und sich wohl kaum jemals zwei Bilder gleichen.

Alle Spieler beginnen in der Hauptstadt damit, ihre Linien zu bauen. Die Hauptstadt gewährt Ihnen dabei einen Zuschuß von 100 Punkten.

Die Spieler beginnen der Reihe nach mit dem Bau.

AUF DISC IM HEFT

Inzwischen sind am oberen Rand die Nummern der Eisenbahnbauer sowie ihre Guthaben erschienen. Der Reihe nach werden die Spieler aufgefordert, eine Strecke nach ihrer Vorstellung zu bauen. Dabei ist es oberstes Ziel, die Städte miteinander zu verbinden. Jeder Spieler baut sechs Streckenabschnitte, dann ist der nächste an der Reihe. Es liegt an jedem selbst, die optimale Streckenführung auszusuchen. Gibt der Spieler „L“ ein, so kann er beginnen, bei „0“ ist der nächste an der Reihe. Es steht jedem frei, entweder auf den Bau zu verzichten, oder aber erst einige Abschnitte zu erstellen.

Die Städte sind daran interessiert, daß sie an das Eisenbahnnetz angeschlossen werden und zahlen dem ersten, der bei ihnen einen Anschluß erstellt, eine Prämie von 50 Punkten. Der zweite erhält 30 und der dritte noch 10 Punkte. Alle weiteren Anschlüsse gehen leer aus.

Ein Spieler kann auch mehrere Prämien in der gleichen Stadt einheimsen, wenn er diese mehrmals anschließt.



Diese Prämien werden auch dringend benötigt, denn der Eisenbahnbau kostet viel Geld. Sie werden sehen, daß auf dem unteren Rand der vor Ihnen liegende Abschnitt beschrieben wird. Dabei wird auch angezeigt, was es Sie kostet, Ihre Strecke dort zu bauen. Der Baupreis schwankt je nach Landschaft zwischen zehn und 70 Punkten.

Es ist aber nicht immer möglich, in jede beliebige Richtung zu bauen. So können Sie niemals eine Kurve setzen, die im spitzen Winkel verläuft. Es ist auch nicht möglich, in einen Abschnitt hineinzubauen, in dem bereits eine eigene oder fremde Linie verläuft. Dagegen sind die Städte immer aus allen acht Richtungen zu erreichen. Ebenso können Sie sie nach allen Richtungen verlassen. Ein reverses Fragezeichen zeigt die möglichen Richtungen an, in denen gebaut werden kann. Dieses Fragezeichen können Sie durch Eingabe von „R“ rechts um Ihren Standort bewegen. Dabei sei noch auf die Möglichkeit hingewiesen, zwischen zwei schräg verlaufenden Bahnabschnitten eine Bahnlinie zu überqueren. Auf dieselbe Weise kann ein Fluß überbrückt werden, ohne viele Punkte für eine Brücke zu opfern.

Beim Bauen sollten Sie vor allem darauf achten, daß Sie nicht in eine Sackgasse geraten, aus der es kein

Entrinnen mehr gibt. Da es nicht möglich ist, auf Ihrer eigenen Spur zurückzubauen, bleibt Ihnen nichts übrig, als immer, wenn die Reihe an Ihnen ist, „0“ einzugeben und somit auf das weitere Bauen zu verzichten. Sie können dann erst wieder im nächsten Abschnitt aktiv ins Geschehen eingreifen.

VORSICHT BEI DER FRACHTBEFÖRDERUNG: DER GEGNER KASSIERT MIT!

Der Abschnitt des Eisenbahnbaues ist zu Ende, wenn alle Städte ans Netz angeschlossen sind und alle Spieler hintereinander vom ersten bis zum letzten durch „0“ bekanntgegeben haben, daß sie nicht mehr bauen wollen.

Nun beginnen die Fahrten mit den Zügen. Es erscheinen am unteren Rand acht Angebote für Frachtbeförderung. Da sind in der oberen Reihe die beiden Buchstaben der Städte, von denen die Fracht abgeholt und zu der diese gebracht werden soll. Darunter steht der Beförderungssatz, den der Spieler erhält, wenn er die Fracht ordnungsgemäß transportiert hat. Ordnungsgemäß heißt hier, daß der Zug zuerst die Stadt anfahren muß, von der die Fracht abzuholen ist, und darauf die Stadt, in der die Fracht abzuliefern ist. Man wird nun noch versuchen, möglichst über die eigenen Linien zu fahren, denn für Fahrten über fremde Strecken erhält der Besitzer dieser Strecke pro Abschnitt einen Punkt, der dem fahrenden Spieler abgezogen wird. Seine eigene Strecke erkennt jeder Spieler an der Farbe, die bei Strecke und Spielertabelle identisch ist. Der Spieler, der an der Reihe ist, sucht nun eine ihm genehme Fahrt am unteren Rand aus. Anfangs stehen die Züge aller Spieler in der Hauptstadt.

Durch „R“ wird wieder die Fahrtrichtung gewählt und durch Leertaste oder Feuer der Zug gestartet. Dieser fährt bis zur nächsten Stadt auf der ausgewählten Strecke und ist auch vorher nicht zu stoppen. In der nächsten Stadt kann er mit „R“ in die nächste Richtung dirigiert werden. Das geht so lange, bis der Zug erst den Anfangs- und danach den Zielbahnhof erreicht hat. Bis hier wurden für jeden Abschnitt sofort ein oder zwei Punkte abgezogen. Nach Ende der Fahrt wird der Frachtbetrag gutgeschrieben. Das Frachtfeld am unteren Rand erlischt und der Spieler kann eine weitere Fracht übernehmen, die nach der gleichen Regel verläuft. Jeder Spieler kann in einer Runde eine bis drei Frachten nacheinander abwickeln.

G. Kramer □

**Das neue
128/C64
Special
ist da!**

**An guten Kiosken und
im Bahnhofs-Buchhandel.**

APFELBERGE IN 'C'

Künstler

Es muß ja nicht immer das Sieb des Erathostenes sein, wenn es um die Computer-Sprache C und ihre Effizienz geht.

Da C die Muttersprache der sogenannten Apfelgrafiken ist – das ist eine andere Art von Fractals, die Sie sicher schon von den Mandelbrot-Fractal-Bildern kennen –, soll sie den C64-Fans nicht länger vorenthalten bleiben. Sicher, so wie auf dem 32-Bit-Computer im Goethe-Institut in Bremen, der unter dem Betriebssystem UNIX läuft, eine Bildschirmauflösung von 1280 mal 1024 Bildpunkten besitzt und

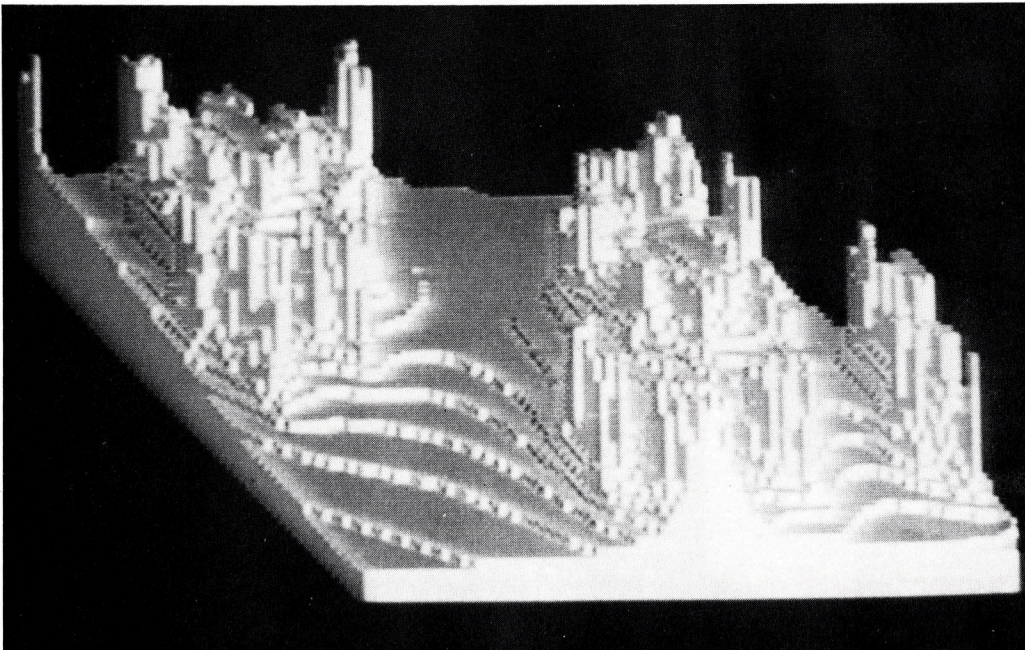
folgenden Funktionen, die mit den entsprechenden Tasten aufgerufen werden können:

0 = Eingabe

Es können für XL, XR, YU, YO und SUCHTIEFE die üblichen Apfelgrafik-Parameter eingegeben werden. Bei dem Punkt BALKENHÖHE sollten die Werte nicht höher als 25 sein, da sonst die Endpunkte außerhalb des Bildschirms liegen würden. Die Werte XC und YC können Sie zwischen Null und Eins legen, also von 0.0 bis 0.9. Achten Sie aber darauf, daß XC größer ist als YC.

1 = Laden und 2 = Speichern

Die hier verwendeten Lade- und Speicherroutinen stammen aus dem Handbuch zu *Profi-C64*, Seite 252 vom Data-Becker-Verlag, Düsseldorf. Wir möchten an dieser Stelle dem Verlag für die Genehmigung danken, diese Routinen innerhalb des C-Files „Apfel/C.OBJ“ auf der COMMODORE DISC zu veröffentlichen. Damit lassen sich Grafiken nach



Frappierende Ähnlichkeit mit Monument Valley, USA: Apfelmännchen-Grafik mit „C“

16 Millionen verschiedene Farbwerte zur Verfügung hat, wird's sicher nicht werden. Trotzdem lohnt es sich, dieses Programm auf Disc zu starten und das grafische Ergebnis abzuwarten, das hier nach wenigen Minuten vorliegt, im normalen BASIC 2.0 des C64 aber viele Stunden benötigen würde.

C MACHT'S MÖGLICH

Das Programmfile „Apfel/C.OBJ“ ist vollkommen in C geschrieben und wird vom BASIC-Lader „Apfelberge in C“ geladen und gestartet. Falls Sie auf dieses kleine Ladeprogramm in BASIC verzichten möchten, so müssen Sie „Apfel/C.OBJ“ absolut laden, also mit dem Zusatz „8,1“:
LOAD„APFEL/C.OBJ“,8,1

Die so entstehenden Apfelgrafiken werden dreidimensional dargestellt und haben eine verblüffende Ähnlichkeit mit dem Grand Canyon und Monument Valley in USA. Außerdem wird auch noch die Berechnungszeit zu Ihrer Information gestoppt. Nach dem Start meldet sich das Hauptmenü mit

Eingabe des Namens laden und speichern. Die Parameter, mit denen Sie die Grafik erstellt haben (siehe 0 = Eingabe), werden allerdings nicht gespeichert. Interessierte Leser sollten sie von Hand notieren – denn was nützt die schönste Grafik, wenn Sie nicht mehr wissen, aufgrund welcher Parameter Sie sie erstellt haben.

3 = Grafik sehen

Auch wenn das Programm nach der Fertigstellung der Apfelgrafik wieder ins Menü springt, bleibt der Bildschirmspeicher der Grafik nach wie vor erhalten und kann mit diesem Menüpunkt wieder angezeigt werden. Außerdem können Sie durch vorherige Eingabe der entsprechenden Farbwerte (Null bis Fünfzehn) jederzeit drei neue Multicolorfarben auswählen, wenn Sie mit den gerade aktuellen nicht einverstanden sind.

X = Ende

Durch Drücken der Tasten SHIFT und X wird ein RESET durchgeführt, und Sie befinden sich wieder

im normalen BASIC 2.0 des C64. Das Programm kann nur durch erneutes Laden wieder gestartet werden. Das Betätigen der Tastenkombination RUN-STOP/RESTORE wird abgefangen und verzweigt zu einer Abfrage, ob Sie das Programm wirklich beenden oder doch weitermachen möchten.

PROGRAMM-TECHNISCHES

Das Programm ist sehr einfach und verständlich gehalten. Als Listing können Sie es sich allerdings nur dann ansehen, wenn Sie im Besitz des entsprechenden Betriebssystems sind, etwa „Profi-C64“.

Die Grundlage ist ähnlich den sonstigen Apfel-Programmen. Als Abwandlung bestimmen die Iterationen nicht die Farbe, sondern die Balkenhöhe. Jeder Balken besitzt eine rote Vorder-, eine grüne linke und eine blaue Oberseite. Die Grafik wird von rechts nach links und von hinten nach vorne gezeichnet, dadurch entfällt eine Berechnung der verdeckten Linien. Eine perspektivische Darstellung wird durch eine Zeilenverschiebung um einen halben Bildpunkt (Pixel) nach rechts erreicht.

C-SPEZIFISCHES

Ein Wort zur System-Übertragbarkeit (Portabilität) dieses Apfelprogrammes in C: Es gibt mittlerweile auch für den Amiga und Atari ST C-Compiler-Grafik-Bibliotheken wie Aztec, so daß, abgesehen von Lade- und Speicherroutinen, keine nennenswerten Portabilitäts-Probleme entstehen dürften.

Und nun viel Spaß mit den „Apfelbergen in C“. Für experimentierfreudige Leser haben wir noch drei Parameter-Tabellen als Eingabebeispiel abgedruckt, die Sie ausprobieren sollten.

APFELBERGE IN 'C'

Beispiele für Parametereingaben
Menüpunkt 0 = Eingabe:

Beispiel 1:	XL =	-15
	XR =	.26
	YU =	.9
	YO =	.47
	SUCHTIEFE =	60
	BALKENTIEFE =	20
	XC =	1
	YC =	0
Beispiel 2:	XL =	-1.4
	XR =	1.4
	YU =	1.2
	YO =	- .9
	SUCHTIEFE =	40
	BALKENHÖHE =	22
	XC =	.77
	YC =	.6
Beispiel 3:	XL =	-1.2
	XR =	1.0
	YU =	- .7
	YO =	.5
	SUCHTIEFE =	50
	BALKENTIEFE =	20
	XC =	.78
	YC =	.45

Gerhard Szymanski/hb ☐

GRAFIK-CONVERTER C128-C64

Zeichner

Beide haben denselben Videochip für den 40-Zeichen-Modus, beide können dieselbe hochauflösende Grafik darstellen, und doch sind die damit erstellten Grafikbilder nicht kompatibel: Der C64 und der 128PC benutzen etwas verschiedene Speicherbereiche.

Nicht jeder C64-Besitzer ist Eigentümer eines guten Grafikprogramms zum Zeichnen und Malen auf dem Bildschirm, wie es beispielsweise *Hi-Eddi* ist. Grafiken mit den Befehlen und Anweisungen des BASIC 2.0 auf den Bildschirm zu bringen, ist ein schier aussichtsloses Unterfangen für einen Computerneuling. Leichter haben es da die C128-Anwender, die – nach intensivem Studium des Handbuchs – die recht komfortablen Zeichenbefehle wie DRAW, BOX, CIRCLE und so weiter benutzen können, sowohl in HIRES- als auch im Multicolor-Modus. Diese Grafikbilder lassen sich auch mit der simplen Anweisung BSAVE und den nötigen Parametern als PRG-File auf eine Diskette abspeichern und mit BLOAD zum Ansehen wieder laden.

Nur verwenden können Sie diese C128-Grafiken beim C64 nicht, das Bildschirm-RAM für die Farbgebung liegt leider an verschiedenen Stellen: beim C64 ab Adresse 1024 (\$0400) bis 2023, beim C128 dagegen von \$1C00 (7168) bis 8191. Gleich anschließend daran befinden sich die Bytes für das Grafikbild, die bei einer farbigen Grafik bis \$3FFF (16383) reichen.

Mit dem *Grafikkonverter* ist es ohne weiteres möglich, sich HIRES- oder Multicolor-Grafiken, in den C64 laden zu lassen, sie sich dort anzusehen und darüber hinaus noch als C64-Grafikbild im „Hi-Eddi“-Format auf eine Diskette zu speichern. So eine Grafik kann dann jederzeit in ein Zeichenprogramm wie „Hi-Eddi“ geladen und dort weiterverarbeitet werden. Aus einer C128-Grafik ist eine für den C64 geworden.

BEDIENUNGSANLEITUNG

Der *Grafikkonverter* besteht aus zwei Teilen, dem Haupt- und Steuerprogramm und dem File, in dem sich die benötigten Maschinenroutinen befinden: Einschalten der beiden Grafikmodi, HIRES oder Multicolor, Einlesen des Directory, Speichern des C128-Bildes als C64-Grafik, um nur die wichtigsten zu nennen.

Nach dem Laden des Hauptprogramms und dem Start mit RUN wird das Maschinsprache-File P128 nachgeladen. Dann erscheint ein einfaches Menü, aus dem Sie Ihre gewünschten Funktionen auswählen können:

- D = Disketten-Inhaltsverzeichnis,
- H = HIRES-Grafikbilder laden,
- M = Multicolorbilder laden.

Auf der COMMODORE DISC finden Sie folgende Grafik-Files, die reine C128-Grafiken sind:

- Startrotter (Multicolorbild),
- Conquest (Multicolorbild),
- E.T. (HIRES-Bild).

Bevor Sie so ein Bild-File laden, müssen Sie durch Druck auf die H- oder M-Taste festlegen, um welchen Modus es sich handelt, sonst sieht das nachgeladene Bild nicht so aus, wie Sie es gerne hätten.

Geben Sie dann den Bildnamen ein. Der C64 lädt nun das gewünschte C128-Bild und zeigt es Ihnen als C64-Grafik auf dem Bildschirm.

Drücken Sie jetzt die Leertaste, erscheint ein neues Menü:

S = Speichern als C64-Grafik im Hi-Eddi-Format,
A = anderes C128-Grafikbild ansehen,
E = Programmende.

Wenn Sie das Bild als C64-Grafik speichern möchten, so drücken Sie S. Der Speichervorgang beginnt unmittelbar, also ohne Sicherheitsabfrage. Als Filename wird der verwendet, den Sie bereits beim Laden der C128-Grafik benutzt haben. Achten Sie also darauf, daß sich auf der Diskette, auf die Sie das C64-Bild speichern möchten, nicht dieselben Filenamen oder die C128-Files befinden, sonst gibt's die Fehlermeldung: FILE EXISTS. Der Speichervorgang kann nicht durchgeführt werden.

Haben Sie A gewählt, so können Sie sich ein anderes Bild-File des C128 in den C64 holen.

HINWEISE ZUR WEITERVERARBEITUNG DER C64-BILDER MIT „HI-EDDI“

Falls Sie die bekannten Mal- und Zeichenprogramme „Hi-Eddi“ und „Hi-Eddi+“ für den C64 besitzen, kann dieses neu generierte Bild, das vordem eines für den C128 war, jederzeit mit den üblichen Ladebefehlen (CBM-Taste/L) geladen und dort weiter verarbeitet werden (Farben oder Zeichnung verändern und so weiter). So geändert, können Sie es erneut abspeichern, wie Sie's von einem Bild, das „nur“ mit „Hi-Eddi“ erstellt wurde, gewohnt sind.

Auf eines müssen wir Sie aber hinweisen: Das C128-Grafikfarbbild, und nur um solche handelt es sich, hat einen Speicherumfang von 9192 Byte. Diesem Umstand trägt der *Grafikkonverter* Rechnung, ein neu generiertes C64-Bild belegt 37 Blocks auf Diskette. Daher sollten Sie beim Lademenü dieses Zeichenprogramms immer F (für Farbbild) wählen. Welche Betriebsart Sie benutzen, 128 oder 192 (mit Farbménü), ist völlig egal.

MULTICOLOR GEHT DOCH

„Hi-Eddi“ ist normalerweise ein Zeichenprogramm, das mit Multicolorbildern nichts anfangen kann. Das heißt aber nicht, daß es nicht doch möglich sei. Es gibt zwei Versionen dieses Malprogramms: „Hi-Eddi“ und „Hi-Eddi plus“. Bei der ersten Version sollten Sie nur Betriebsart 128 anwählen. Dann bringen Sie, zum Einschalten des Multicolor-Modus, im Programm LADER folgende Zeile unter:

POKE 53270, PEEK (53270) OR 16

entweder gleich zu Beginn oder noch besser nach der INPUT-Abfrage „Betriebsart“. Die entsprechende Listingzeile müssen Sie eventuell trennen, da sie sonst zu lang wird.

Bei der zweiten Fassung, „Hi-Eddi-Plus“, sieht das etwas anders aus. Das Hauptprogramm besteht aus Maschinencode und einer BASIC-Zeile, in der der SYS-Befehl zum Programmstart steht. Der PIC.-LADER zum Ansehen der Grafiken auf dem Bildschirm ist jedoch in BASIC geschrieben. Hier sollten Sie folgende Zeilennummer anfügen:

555 POKE 43270, PEEK (53270) OR 16

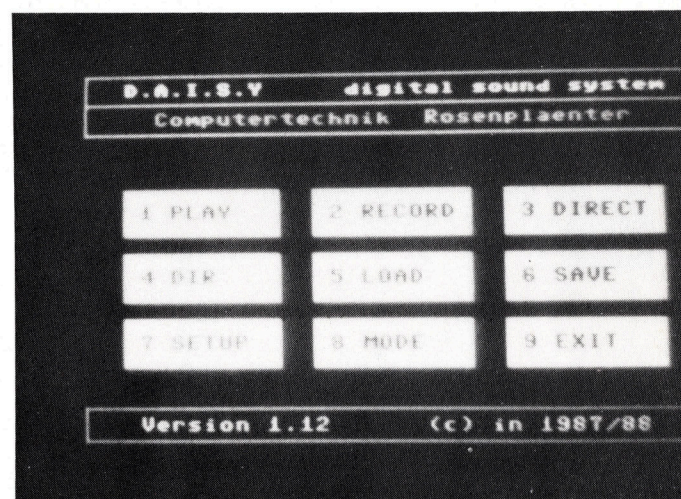
und dann dieses Ladeprogramm mit RUN starten. Nachdem Sie den auf dem Bildschirm erscheinenden Text gelesen haben, tippen Sie erneut ein: RUN 500. Jetzt werden Sie nach dem Filenamen des Multicolorbildes gefragt, das Sie laden möchten – und das vor vielen Byte-Einheiten einmal ein C128-Bild war. Das war's schon. hb □

D.A.I.S.Y

Hifi pur

Mit Walt Disney hat d.a.i.s.y. nichts zu tun. Es ist die Abkürzung für Digital Audio Interface System, ein Sprach- und Musik-Digitalisierer für den C64 mit außergewöhnlichen Qualitäten.

In den letzten Jahren ist eine Reihe von Sound-Digitalisierern auf dem Markt erschienen, die zwar preisgünstig sind, aber qualitativ doch viele Wünsche offen lassen. Die Sprache ist oft schwer zu verstehen und der Musik mangelt es an hohen Frequenzen. Neben einem Steckmodul und einem recht ausführlichen Handbuch werden eine Systemdiskette mit diversen Programmen sowie ein Kabelsatz mit drei verschiedenen Steckern (Cynch-, DIN- und 3.5-mm-Klinenstecker) geliefert.



Das Modul ist für den Erweiterungsport bestimmt und besitzt neben der Cynch-Buchse einen Regler und eine LED (Leuchtanzeige), die zum Einstellen der richtigen Aufnahmestärke dienen. Sound (Sprache oder Musik) können Sie über einen handelsüblichen Kassettenrecorder aufnehmen, der über das entsprechende Kabel mit dem C64 verbunden sein muß, auch wenn das Handbuch als optimale Signalquelle einen Compactdisc-Player mit Equalizer empfiehlt. So ein Gerät besitzt nicht jeder.

Das 28seitige Handbuch ist übersichtlich gestaltet und auch für Anfänger leicht verständlich. Es beschreibt die Steuersoftware und die BASIC-Erweiterung. Alle neuen Befehle sind mit einem vorangestellten Pfeil nach links aufzurufen. Zusätzlich erhalten Sie noch einige Tips zur Aufnahme.

TEST

VOLL MENÜGESTEUERT

Das menügesteuerte Hauptprogramm bietet außer den Standardfunktionen Aufnahme, Wiedergabe, Laden, Speichern und Directory noch die Möglichkeit, wichtige Parameter wie Start und Ende des Soundspeichers sowie Aufnahmegeschwindigkeit zu ändern. Diese Funktion ist auch nach der Aufnahme anwendbar, so daß Musikstücke „abgerundet“ werden können. Das heißt, daß Wiederholungen reibungslos ineinander übergehen, ohne Anfang und Ende störend erkennen zu lassen. Besitzer eines C128 haben die Möglichkeit, Aufnahmen im 2-MHz(FAST)-Modus zu generieren, wodurch sich die Aufnahmequalität nochmals verbessert. Die maximale Aufnahmezeit wird allerdings halbiert.

Zur Einstellung der idealen Aufnahmelautstärke hat der Benutzer die Möglichkeit, die am Modul anliegende Musik oder Sprache aus dem Monitor-Lautsprecher mitzuhören.

BEFEHLSERWEITERUNG – FREI ZUR WEITERGABE

Die Befehlsweiterung für das BASIC 2.0 des C64, das „d.a.i.s.y.-Basic-Tool“, funktioniert auch ohne das Einsteckmodul und darf von jedem Besitzer dieses Hard- und Software-Pakets weitergegeben werden, solange keine Veränderungen irgendwelcher Art daran vorgenommen worden sind. Mit diesem BASIC-Tool können Sie Sound-Files in eigene Programme einbauen. Dazu stehen Ihnen mehrere Befehle zur Verfügung.

Einige Beispiele:

- C x,y,z = Setzen der Bildschirmfarben
- D = Directory
- S x = Bildschirm ein-/ausschalten
- P n,x,y,z = Ausgabe eines Soundfiles aus der RAM-Disk.

Mit dem sogenannten Vorspann-Generator lassen sich auch längere Musikstücke ohne Modul abspielen.

NÜTZLICHES ZUSATZPROGRAMM

Wer seine eigenen „Maxis“ komponieren möchte, sollte sich für 34,50 Mark noch den „Sequencer 64“ besorgen. Durch An- und Teilspiel-Möglichkeiten von 31 Sounds können beispielsweise Effekte wie „H-H-H-Hallo“ erzeugt werden. Das ebenfalls menügesteuerte Programm besitzt einen Bildschirmeditor mit vielen Befehlen, etwa COPY/MOVE/INSERT/DELETE. d.a.i.s.y. ist voll kompatibel zu den gebräuchlichsten Floppy-Speedern wie *SpeedDOS*, *Prologic DOS*, *Turbo Trans* und *Professional DOS*. Damit Soundfiles schnell verfügbar sind, sollten Sie Ihren Speeder, falls Sie einen haben, auf alle Fälle nutzen.

FAZIT

d.a.i.s.y. ist ein recht leistungsstarker Sound-Digitizer mit beachtlicher Klangqualität. Der Preis ist allerdings nicht von Pappe: 178 Mark. Ob sich diese Ausgabe für Sie lohnt, könnten Sie zunächst mit einer bedeutend geringeren Investition testen. Gegen Einsendung von fünf Mark können Sie sich eine Testdiskette von Computertechnik Rosenplänter, Lange Straße 12, 3400 Göttingen, zuschicken lassen.

Sven Faulhaber/Reinhard Holler □

IMPRESSUM

COMMODORE DISC

C-DISC erscheint monatlich
in der CA-Verlags GmbH

VERANTWORTLICH
FÜR DEN INHALT:
Harald Beiler

REDAKTION UND
STÄNDIGE MITARBEITER:
Peter Basch, Harald Beiler,
Renate Huber, Lothar Miedel,
Alfons Mittelmeier, Michael
Reppisch, Rudolf Schmid-
Fabian, Torsten Seibt,
Hermann Wellesen, Bernd
Welte

GESCHÄFTSFÜHRER:
Werner E. Seibt

ANZEIGENVERWALTUNG:
ADV-Mediendienste,
Postfach 101124,
8900 Augsburg 1
Telefon: 0821/7904-227
Telex: 533502 adv
Teletex: 821887
Telefax: 0821/7904-243

Es gilt Preisliste Nr. 9 vom
1.4.1988.
Media-Unterlagen bitte
anfordern.

VERANTWORTLICH FÜR
DEN ANZEIGEN-INHALT:
Brigitte Kostic

ANSCHRIFT FÜR ALLE
VERANTWORTLICHEN:
Postfach 1107,
8044 Unterschleißheim
Telex: 5214428 cav-d

© 1988 by CA-Verlags GmbH
Heßstraße 90,
8000 München 40.
SPS und Autoren. Für unau-
gefordert eingesandte Manu-
skripte und Listings keine
Haftung. Bei Einsendung
von Texten, Fotos und
Programmträgern erteilt der
Autor dem Verlag die Geneh-
migung für den Abdruck
und die Aufnahme in den
Kassetten-Service zu den
Honorarsätzen des Verlages
und überträgt dem Verlag
das Copyright. Alle in dieser
Zeitschrift veröffentlichten
Beiträge sind urheberrecht-
lich geschützt. Jedwede Ver-
wendung ist untersagt.
Namentlich gezeichnete Bei-
träge unserer Mitarbeiter
stellen nicht unbedingt die
Meinung der Redaktion dar.

VERTRIEB:
Verlagsunion Wiesbaden

Printed in Germany by
ADV, 8900 Augsburg 1

**COMMODORE
DISC
Hotline
Jeden
Mittwoch
15 - 19⁰⁰
Telefon
089/129 80 13**

AUF DISC IM HEFT

ASSEMBLER-KURS C64 (Teil 1)

Dolmetscher

Das Höchste für den C64-Freak ist das Beherrschen der sogenannten Maschinensprache, auch Assembler genannt. Dieser Kurs unseres Autors Ralf Trabhardt soll allen BASIC-Programmierern den Einstieg in Assembler ermöglichen. Das Wort hat der Autor.

Ich werde mich darum bemühen, die komplizierten Fachbegriffe zu umgehen und leicht verständliche Erklärungen zu geben. Zunächst sollen innerhalb dieses Kurses alle Assembler-Befehle behandelt werden. In späteren Teilen wird auf die Verwendung von Kern-Routinen und die Interrupt-Technik eingegangen. Zu jedem Teil des Kurses befindet sich ein Programm auf der beiliegenden Diskette („Assemblerkurs 1“), das einige Assembler-Routinen enthält und Ihnen gleich die Auswirkungen der jeweiligen Befehle zeigen kann.

DAS WERKZEUG

Zum Programmieren in Assembler benötigen Sie unbedingt ein Programm, das die Befehle in reine Maschi-

STA (store accumulator) schreibt diesen Akkumulator-Inhalt dann in die angegebene Speicherstelle. Der Befehl BRK entspricht in etwa dem BASIC-Befehl STOP und sollte zunächst immer am Ende Ihrer ersten Programmversuche mit Assembler stehen. LDA und STA sind mit PEEK und POKE vergleichbar.

Zum Beispiel: POKE 8192,8

Dieser Befehl schreibt also den Wert Acht in die Speicherstelle 8192.

Eine Routine in Assembler, die die gleiche Aufgabe erfüllt, lautet: LDA #\$08

STA \$2000

BRK

Lassen Sie sich nicht dadurch verwirren, daß hier Hexadezimalzahlen (Erklärung im Programm) verwendet werden.

Das Zeichen „#“ in der ersten Zeile zeigt an, daß eine konstante Zahl und nicht der Inhalt einer Speicherstelle in den Akku geladen wird.

Die hexadezimale Zahl \$2000 entspricht der dezimalen 8192.

Zunächst wird also die Zahl Acht (bzw. S08) in den Akku geladen. In der zweiten Zeile wird dieser Wert in der Adresse 8192 (\$2000) abgelegt.

Ein anderes Beispiel:

Nach der Eingabe von POKE 12288, PEEK (8192) hätte die Adresse 12288 ebenfalls den Inhalt Acht.

UMRECHNUNGSTABELLE HEXADECIMALZAHLEN

\$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Aus dieser Tabelle ersehen Sie ohne Probleme die Hexadezimalumrechnung

nensprache umwandelt. Diese werden ebenfalls Assembler oder Monitore genannt. Ich empfehle Ihnen den „Supermonitor.64“ aus der COMMODORE DISC Nr. 9, selbstverständlich läßt sich aber jedes andere Monitor-Programm dazu verwenden.

LDA, STA UND BRK

Jetzt können wir mit den ersten Befehlen loslegen. Wir fangen an mit LDA (load accumulator). Der Akkumulator (oder kurz Akku genannt) ist eine Art Verteilerstelle. Mit diesem Befehl lädt man also eine Zahl in den Akku.

Die entsprechenden Assembler-Befehle lauten:

LDA \$2000

STA \$3000

BRK

Mit LDA wird diesmal keine Zahl, sondern der Inhalt der Speicherstelle \$2000 (ähnlich PEEK(8192)) in den Akku gebracht. STA legt den Akkumulator-Inhalt daraufhin bei \$3000 (dezimal 12288) ab.

DER UMGANG MIT DEM SUPERMONITOR

Eingegeben werden die Assembler-Befehle mit dem Supermonitor durch: A Speicherstelle Assembler-

Befehl (zum Beispiel: A C000 LDA#\$08). Wenn Sie Ihre eingegebenen Befehlsfolgen noch einmal betrachten wollen, so können Sie dies mit „D Speicherstelle“ (zum Beispiel D C000) tun.

Es wird Ihnen dann bei unserem Beispiel folgende Zeile gezeigt:

```
. C000 A9 08 LDA #$08
```

Die Zahl C000 ist die Speicherstelle, in der der Befehl steht. A9 ist die Codezahl für den LDA-Befehl. Die darauffolgende Zahl zeigt den Wert an, der in den Akku geladen wird. Uns brauchen zunächst nur die Assembler-Befehle im Klartext zu interessieren.

Laden Sie nun das Programm „Assembler-Kurs 1“ von der Diskette. Es wird Ihnen weitere Hilfen zu den ersten Assembler-Befehlen und zum hexadezimalen Zahlensystem geben. Ralf Trabhardt ☐

KEY 64

Schlüssel- Dienst

Funktionstasten und die Möglichkeit, sie jederzeit mit bestimmten Befehlen oder anderen Daten belegen zu können, sind eine feine Sache – leider nicht beim C64.

Das Betriebssystem und das BASIC 2.0 des C64 sieht eine andere Belegung der Funktionstasten nicht vor. Sicher, jede dieser vier Tasten – mit SHIFT gedrückt, sind es sogar acht – hat ihren eigenen CHR\$-Code (von F1 = CHR\$(133) bis F8 = CHR\$(140)). Aber das einzige, was Sie in dieser Normalbelegung des C64 damit anfangen können, ist, eine Abfrage zu programmieren, ob so eine Taste gedrückt wurde, um dann Ihr Programm entsprechend reagieren lassen zu können.

Ziemlich wenig, finden Sie nicht? Da ist der C128 viel besser dran. Seine acht Funktionstasten sind bereits beim Einschalten des Computers mit Texten belegt, die nach Druck darauf ausgeführt werden.

NEUER BEFEHL FÜR DEN C64: KEY

Nach dem Start des Programmes werden die nötigen Daten in den Speicherbereich von \$C000 (49152) bis \$C1FD (49661) gePOKEt und folgende Funktionstasten bereits mit diesem Text belegt:

```
F1 = LOAD
F2 = PRINTCHR$(147)+<RETURN>
F3 = SAVE
F4 = DATA
F5 = RUN + <RETURN>
F6 = GOTO
F7 = LIST + <RETURN>
F8 = GOSUB
```

Doch, damit nicht genug. Wenn Sie jetzt die Commo-

dore-Taste, links unten auf Ihrer Tastatur, zusammen mit der entsprechenden Funktionstaste F1 bis F7 benutzen, haben Sie noch vier Belegungen mehr zur Verfügung:

```
F9 = OPEN2,8,2
F10 = PRINT#
F11 = GET#
F12 = CLOSE 2
```

Es wäre auch noch möglich gewesen, diese vier Funktionstasten nochmals zu belegen, bei gleichzeitigem Drücken der F-Taste und der SHIFT- und Commodore-Taste. Aber zum einen sind dem Programmator keine passenden Belegungstexte mehr eingefallen, und zweitens wollte er Ihnen solche Fingerakrobatik nicht zumuten . . .

NÜTZLICHE BEFEHLE AUF TASTENDRUCK

Wie Sie sehen, sind das alles Anweisungen, die bei Ihrer täglichen Programmierarbeit mit dem C64 ständig vorkommen und nun lediglich per Druck auf die entsprechende Taste auf den Bildschirm geschrieben und zum Teil auch sofort ausgeführt werden. Es sind die Befehle, die in unserer Übersicht mit dem Zusatz „+ <RETURN>“ gekennzeichnet sind. Hier ist nämlich ein CHR\$-Code 13 (Carriage Return) angehängt worden.

JEDERZEIT VERÄNDERBAR

Diesen Belegungsplan der F-Tasten müssen Sie allerdings in keiner Weise einhalten, Sie können den Text jederzeit ändern.

Löschen

KEY Nr. d. F-Taste, “ “ z.B. KEY1, “ “

Belegen

KEY Nr. d. F-Taste, “(Text)“
z.B. KEY3, “LOAD“

Möchten Sie, daß der Befehl auch sofort ausgeführt wird (Carriage Return anhängen), so tippen Sie unmittelbar hinter den Text das Sternchen * (auch Asterix genannt); etwa KEY4, “RUN*“

Diese Befehlssyntax sollten Sie genau beachten, um etwaigen Problemen aus dem Weg zu gehen.

TASTENBELEGUNG ANZEIGEN

Mit der Befehlseingabe KEY0 und dem Drücken der Taste RETURN wird die aktuelle Funktionstasten-Belegung auf dem Bildschirm angezeigt.

Beachten müssen Sie aber auf alle Fälle, daß der Belegungstext nie länger als 18 Zeichen sein darf, sonst stimmen die Zeiger im Maschinenprogramm auf diesen Text nicht mehr.

Außerdem ist das Programm nicht gegen den „nicht maskierten Interrupt“ (NMI) geschützt, der entsteht, wenn Sie die RUN/STOP- und RESTORE-Taste gleichzeitig drücken, oder wenn Sie einen RESET ausführen.

Da aber das Maschinenprogramm im geschützten Speicherbereich ab 49152 steht, kann es jederzeit mit SYS 49172 wieder aktiviert werden.

Wählen Sie statt dessen den Aufruf SYS 49152, so funktioniert das Programm ebenfalls wieder. Aber die vorher geltende Funktionstasten-Belegung ist jetzt gelöscht, so daß Sie sie erneut definieren müssen. her ☐

RELATIONALE DATENBANK

Verwalter

Mit einer relativen Dateiverwaltung hat eine relationale Datenbank nichts zu tun. Im weiteren Sinne nicht mal etwas mit EDV.

Sogenannte *relationale Datenbanken* sind zwar kinderleicht zu bedienen, sparen aber dem Anwender trotzdem nicht die korrekte Gliederung der zu speichernden Daten.

Was ist eine relationale Datei? Das ist schnell erklärt: eine Tabelle.

- Die mit gewünschten markanten Bezeichnungen der Tabellenspalten sind die Datenfelder, die als Maskenzeilen ganz oben an der Tabelle stehen.
- Die einzelnen Bildschirmzeilen solcher Tabellen sind die Datensätze.
- Alle Datensätze zusammengenommen bilden die Datei.

Zur eindeutigen Identifizierung jedes Datensatzes dient ein Feld, das als Schlüssel deklariert wird, da es für jeden Satz einen markanten Inhalt darstellt, der sonst nicht vorkommt. Bei einer Adreßdatei etwa könnte das Datenfeld „Name“ so ein Schlüsselfeld sein, aber auch der „Wohnort“ könnte für den Anwender bedeutend sein. Es kommt immer darauf an, was dem einzelnen wichtig ist.

INVENTAR

Hersteller	Geraet	Geraetart	Grosshandel
Commodore	128 PC	Computer	COMP & Soehne Dortmund
Commodore	Amiga 500	Computer	Halb & Fett Muenchen
Schneider	CPC 664	Computer	Mies & Teuer Gronsdorf
Commodore	Disk Drive 1541	Floppy	Breit & Kurz Koeln
Commodore	C 64	Computer	COMP & Soehne Dortmund
STAR	LC 10	Drucker	Print & Stop Hamburg
Seikosha	VC 80	Drucker	Halb & Fett Muenchen
RKT	Printerface	Interface	Rolle Muenchen
Commodore	Plus/4	Computer	COMP & Soehne Dortmund
Commodore	1084	Monitor	COMP & Soehne Dortmund
Commodore	Disk Drive 1571	Floppy	Lang & Vorn Ruedesheim

Eine relationale Datenbank setzt sich aus mehreren als Tabellen definierte Dateien zusammen, die alle in irgendeiner Beziehung, einer *Relation*, zueinander stehen.

Durch dieses simple Prinzip ist es jedem EDV-Laien möglich, sich solche Datenbanken selbst zu erstellen. Er muß sich nicht wie ein PC-Profi mit hierarchisch organisierten Zugriffspfaden und anderen Details befassen, bei denen er unter Umständen nur „Bahnhof“ versteht.

TABELLEN-DATEI FÜR DEN 128PC: CHECK-LISTE

Daß sowohl vom Befehlsumfang als auch vom Speicherplatz her solche Dateien viel komfortabler auf einem Personal-Computer erstellt und programmiert

werden können, ist sicher jedem klar. Das einfache Beispiel *Check-Liste* für den C128 im 80-Zeichen-Modus wird Ihnen aber zeigen, daß trotz der geringeren Möglichkeiten auch diese Art der Dateiverwaltung auf dem Commodore-Homecomputer möglich ist.

DAS HAUPTMENÜ

Nach Laden und Starten durch RUN wird das Unterprogramm „Menü“ nachgeladen, das im Speicherbereich von \$1300 bis \$16A5 angesiedelt ist und nichts anderes tut, als bei Bedarf den Text des Hauptmenüs auf dem Bildschirm auszugeben. Dadurch ist das BASIC-Hauptprogramm aber kürzer, was sich auch auf die Verarbeitungs-Geschwindigkeit auswirkt.

Dieses Hauptmenü ist der Kreuzungspunkt, die Verteilerstelle für alle Dateioperationen. Nehmen wir uns die einzelnen Punkte, die jeweils mit der entsprechenden Zahlen- beziehungsweise Buchstabentaste anzuwählen sind, etwas genauer unter die Lupe.

1. Feldbezeichnungen (Maske) festlegen/laden

Zunächst erscheint die Frage, ob eine neue Maske erstellt oder eine bereits entworfene und abgespeicherte geladen werden soll. Diese Frage müssen Sie beim ersten Mal verneinen, denn ohne Maske kann keine Datei erstellt werden. In ihr werden die Bezeichnungen, die Abstände der Spalten sowie die Eingabelänge pro Datenfeld festgeschrieben.

Auf dem Definitions-Bildschirm wird nun gefragt,

wie viele Datenfelder Sie anlegen möchten. Nach der Eingabe der gewünschten Zahl müssen Sie die Bezeichnung für die einzelnen Datenfelder eingeben. Außerdem erscheint am unteren Bildschirmrand ein Entwurfsfeld, das Ihnen das aktuelle und, wenn Sie damit einverstanden sind, das spätere Aussehen der Bildschirmmaske anzeigt.

Die Ausdehnung und die Länge der Datenfelder bestimmen Sie, indem Sie nach der Eingabe der Feldbezeichnung mit der Leertaste oder dem Cursor weiter nach rechts fahren, bis Sie glauben, die Feldspalte sei breit genug. Zur Abgrenzung setzen Sie den Doppelpunkt. Obwohl die Eingabe von der C128-eigenen INPUT-Routine durchgeführt wird, die im Normalfall Leerzeichen hinter einer Zeicheneingabe ignoriert, nimmt sie der Computer durch den Trick mit dem Doppelpunkt an. Der ist zwar bei der Datenfeldbe-

AUF DISC IM HEFT

zeichnung nicht sichtbar, aber die Leerzeichen werden akzeptiert.

Nach jeder Eingabe eines Datenfeldes wird dies im Entwurfsfeld angezeigt, durch einen senkrechten Strich vom nächsten getrennt. Eines sollten Sie aber beachten: Eine Bildschirmzeile kann nie größer als 80 Zeichen sein, daher müssen Sie die Breite der Datenfelder (Spalten) entsprechend definieren. Auch hier hilft Ihnen der Computer. Falls die Gesamtlänge der Datenfelder die Grenze von 78 Zeichen überschreitet, macht Sie das Programm darauf aufmerksam. Die bisher beschriebene Entwurfszeile wird gelöscht und das Eingabe-Spiel beginnt von neuem. Manchmal kann es notwendig sein, eine kleiner Datenfeldanzahl einzugeben, oft genügt aber schon eine Reduzierung und Ausrichtung der Spaltenbreite mit Hilfe der Tasten SPACE oder DEL.

Hat die Datenfeldmaske endlich das gewünschte Aussehen, sollten Sie sie nach der entsprechenden Abfrage auf Diskette speichern. Ist Ihre Antwort „Nein“, kehrt das Programm zum Hauptmenü zurück. Sollten Sie in letzter Sekunde noch etwas ändern wollen, geht das über die Taste A.

COMP&SOEHNE

Hersteller	Geraet	Geraeteart	Grosshandel
Commodore	128 PC	Computer	COMP & Soehne Dortmund
Commodore	C 64	Computer	COMP & Soehne Dortmund
Commodore	Plus/4	Computer	COMP & Soehne Dortmund
Commodore	1084	Monitor	COMP & Soehne Dortmund

Beispiel einer Unterdatei, sortiert nach Datenfeld 4

Noch ein Wort zum Abspeichern: Der Name der Maske sollte identisch mit dem der Hauptdatei sein, die Sie eingeben möchten, also etwa „Adressen“. Es gibt keine Probleme, wenn Sie denselben Namen für Maske und Hauptdatei verwenden; das Programm hängt an den Dateinamen für die Maske selbständig den Zusatz „.M“ an. Verwenden Sie aber bitte keine Dateinamen, die länger als 14 Zeichen sind. Sonst könnte es passieren, daß der Masken- und Hauptdatei-File-name doch nicht mehr identisch sind, was sich negativ beim Menüpunkt drei auswirken könnte.

2. Datensätze eingeben

Unter der soeben entworfenen oder dazugeladenen Maskenzeile erscheint beim jeweils aktuellen Datenfeld der blinkende Strichcursor, der Sie auffordert, die Spalte zu beschreiben. Haben Sie nichts einzutragen, drücken Sie nicht einfach RETURN, sondern setzen zuerst wieder unseren bereits bekannten Platzhalter, den Doppelpunkt. Das Betätigen der RETURN-Taste bringt den Cursor dann ins nächste Datenfeld und so fort, bis die Eingaben in der aktuellen Zeile beendet sind. Der erste Datensatz steht somit im Speicher und der Cursor am Beginn der nächsten Zeile, wo er auf Ihre Eingaben wartet, um den nächsten Datensatz zu schreiben.

Wenn Sie die Eingabearbeit beenden möchten, drücken Sie die Taste mit dem Stern (*). Darauf erscheint die Frage, ob Sie die Datei bereits auf Disket-

te abspeichern möchten. Hierzu müssten Sie den Dateinamen angeben, der ja derselbe wie der der Maske sein sollte. Haben Sie sich für „Nein“ entschieden, ist das kein Problem, aus dem Hauptmenü heraus läßt sich die gesamte Datei zu jeder Zeit speichern. Wenn Sie sie vorher alphabetisch sortieren lassen möchten, ist es sogar empfehlenswert, die Datei nicht sofort auf Diskette zu speichern.

3. Gesamte Datei inklusiv Maske laden

Jetzt wird klar, wieso wir Ihnen geraten haben, für Maske und Datei denselben Filenamen zu verwenden: Dieser Menüpunkt lädt beide Files auf einmal in den C128 und steht damit sofort zur Bearbeitung bereit. Den gemeinsamen File-Namen müssen Sie allerdings vorher noch eingeben . . .

4. Gesamte Datei speichern

Für alle Unentschlossenen hier die zweite Möglichkeit, die erstellten Daten auf Diskette auszulagern. Einen Vorteil hat sie auf alle Fälle gegenüber dem Sofort-Speichern: Sie könnten beispielsweise die Datei vor-

her mit Menüpunkt S sortieren lassen und erst jetzt, in geordneter Form, abspeichern.

5. Ausgabe auf Bildschirm

Die Datensätze (Zeilen), die sich gerade im Computer befinden, werden mit der entsprechenden Maske formgerecht auf dem Bildschirm aufgelistet. Wem's zu schnell geht, der möge die NO-SCROLL-Taste benutzen, so wie er es vom Stoppen eines Programmlistings kennt.

6. Ausgabe auf Drucker

Dies ist im Prinzip nichts anderes als Menüpunkt fünf, nur ist hier als Ausgabegerät Ihr angeschlossener Drucker angesprochen. Es sollte entweder einer von Commodore sein, oder einer, der dessen serielle Druckeranweisungen beziehungsweise das angeschlossene Druckerinterface versteht. Drucker, die mit einem Parallelkabel über den Userport des C128 mit dem Centronics-Anschluß des Druckers verbunden sind, können nicht benutzt werden – serieller Anschluß (die gebräuchlichste Art) ist erforderlich. Die Drucker-Sekundäradresse sieben (Klein-/Großschrift-Zeichensatz) ist voreingestellt. Bei der entsprechenden Abfrage können Sie aber jederzeit eine andere beliebige Sekundäradresse eintragen, falls Ihr Drucker oder Ihr Interface das braucht.

Mehr Auskunft gibt Ihnen Ihr Druckerhandbuch beziehungsweise die Bedienungsanleitung zu Ihrem Interface. Stellen Sie sicher, daß Ihr Drucker angeschlossen und vor allen Dingen eingeschaltet ist, sonst bricht das Programm mit einem „DEVICE NOT PRESENT“ ab. Eine Abfang-Routine für so einen

AUF DISC IM HEFT

Fehler ist nicht im Programm eingebaut, weil Sie jederzeit ein unfreiwillig unterbrochenes Programm erneut mit "GOTO 10000" und Druck auf RETURN erneut starten können. Die bisher eingegebenen Daten sind ebenfalls noch unversehrt erhalten. Aber bitte keinen Neustart mit RUN durchführen, sonst sind alle bisherigen Daten gelöscht.

Wohnort auch noch zu Beginn des Datenfeldes mit der Postleitzahl ausgestattet, so genügt die Eingabe dieser Zahl. Beispiel Datenfeld drei = Wohnort 8000 Muenchen 80
Hier würde die Eingabe „8000“ genügen, um alle Adressen auszugeben, die in München liegen. Wichtig ist, daß Sie sich bei der Dateneingabe strikt an die-

DRUCKER

Hersteller	Geraet	Geraeteart	Grosshandel
STAR	LC 10	Drucker	Print & Stop Hamburg
Seikosha	VC 80	Drucker	Halb & Fett Muenchen

7. Gesamte Datei erweitern

An jede sich gerade im Computerspeicher befindende Datei lassen sich damit neue Datensatz-Zeilen anhängen, bis zur kritischen Höchstgrenze des verfügbaren Speicherplatzes.

Ein Wort dazu: Es wurden zwar gleich zum Programmstart 700 Datenzeilen reserviert, also mit DIM dimensioniert. Da Sie aber davon ausgehen müssen, daß bei einer Sortierung neue Zwischenspeicherungs-Datenfelder angelegt werden, sollten Sie vorsichtshalber nie diese Höchstgrenze strapazieren. Lieber nach 500, 600 Datensätze aufhören und eine neue „Datei II“ anlegen.

Doch zurück zum Erweitern der Datensätze: Der letzte aktuelle Datensatz wird angezeigt, Sie können jetzt ebenso wie bei Menüpunkt zwei (Dateneingabe) verfahren.

8. Suchen nach Schlüssel-Datenfeld

Das ist der Kernpunkt des Programms *Check-Liste*. Jetzt müssen Sie sich entscheiden, welchen markanten Punkt, der für Sie interessant erscheint oder den Sie zu Ihrer Information benötigen, Sie aus dieser Gesamtdatei (egal, ob sortiert oder ungeordnet), herausgreifen möchten.

Es ist ohne weiteres möglich, durch Eingabe von „Huber Fritz“ sich nur die Adresse dieses speziellen Namens auf dem Bildschirm anzeigen zu lassen. Geben Sie jedoch nur „Huber“ ein, so werden Ihnen alle „Huber“ aufgelistet, also auch „Huber Hans, Huber Elke“ und so weiter. Geben Sie etwa nur „H“ ein, so erscheinen alle Namen, die mit „H“ beginnen. Daraus folgt: Relevant für das Programm sind immer die entsprechenden Zeichen bis zur Länge der Sucheingabe. Daher ist es sehr wichtig, beim Anlegen der Datei die Datenfelder entsprechend zu beschreiben. Es nützt Ihnen nichts, wenn Sie zum Beispiel bei der Namensangabe mit dem Vornamen beginnen. Die Sucheingabe „Fritz Huber“ bringt Ihnen zwar diesen Namen bestimmt auf den Bildschirm. Geben Sie aber nur „Huber“ ein, geschieht gar nichts, obwohl Sie vielleicht zehn Adressen mit diesem Familiennamen in der Datei stehen haben. Da sich die Zeichenfolge „Huber“ aber nicht am Anfang des eingegebenen Datenfeldes befindet, kann es der Computer nicht finden. Wenn Sie bereits bei der Dateneingabe planvoll vorgehen, gibt's diese Probleme nicht.

Natürlich können Sie nach jedem anderen Datenfeld suchen lassen, wenn Sie sich beispielsweise alle Adressen ausgeben lassen möchten, die in einem bestimmten Wohnort beheimatet sind. Haben Sie den

ses Schema gehalten haben. Eine so geschaffene Unterdatei kann wie gewohnt mit entsprechendem Filenamem (etwa „8000“ oder „Muenchen“) abgespeichert werden.

9. Directory

Ein Menüpunkt, der sicher keiner großangelegten Erklärung bedarf. Trotzdem ist es sehr hilfreich, aus einem Programm heraus das Disketten-Inhaltsverzeichnis einsehen zu können, denn ausgiebig und über einen längeren Zeitraum hinweg beschriebene Disketten haben so viele Hauptdateien, Masken-Files und Unterdateien gespeichert, daß Sie sich die exakten Filenamem sicher nicht alle merken können.

C. Datensätze ändern/editieren

Daten ändern sich, entsprechende Dateien sollten diesen Änderungen angepaßt werden können. Das geschieht mit diesem Menüpunkt, der Ihnen nach Aufruf die einzelnen Datensatzzeilen der aktuellen Datei im Computer auflistet, eine nach der anderen. Die folgende Zeile erhalten Sie durch Druck auf die Leertaste, dauernder Druck darauf blättert die Datenzeilen recht rasch durch, bis Sie bei der gewünschten angekommen sind. Beachten Sie, daß dies immer ab dem ersten bis zum letzten Datensatz geschieht, und zwar in dieser Reihenfolge.

Ist ausgerechnet der letzte Datensatz zu ändern, müssen Sie die gesamte Datei mit der Leertaste durchlaufen lassen, bis Sie beim letzten sind. Das wirkt sich aber nur bei der großen, umfangreichen Hauptdatei aus. Daraus erstellte Unterdateien (siehe Menüpunkt acht) sind in der Regel viel kürzer.

Haben Sie die gewünschte Datenzeile gefunden, drücken Sie die Taste A. Sie werden feststellen, daß zwischen den einzelnen Datenfeldern jeweils der Doppelpunkt erscheint, ebenso ein feststehender Vollcursor, der auf dem ersten Datenfeld steht. Jetzt können Sie wie bei der Eingabe die einzelnen Datenfelder ändern, neu beschreiben oder löschen. Verwenden Sie zur Eingabe aber nur die Cursortasten, zum Bewegen im Datenfeld, sowie die Leertaste zum Löschen von Zeichen. Bitte nie die DEL-Taste oder die SHIFT/INST-DEL-Taste zum Einfügen benutzen. Sie würden damit auch die Länge des einzelnen Datenfeldes verändern, was sich bei einer späteren Bildschirm- oder Drucker Ausgabe sehr unschön bemerkbar machen würde.

Haben Sie Ihre Änderungen beendet, so können Sie mit der SPACE-Taste weitere Datensätze zum Ändern heraussuchen oder mit der Stern-Taste den Än-

derungsvorgang nach einem Datensatz beenden. Auch hier sollten Sie nach einer Änderung die Datei wieder auf Diskette zurückspeichern.

U. Unterdatei laden

Unter Menüpunkt acht generierte Unterdateien lassen sich damit in den Computer laden. Diese Unterdateien haben keine anderen Merkmale als die Hauptdatei, aber ein Unterschied besteht doch: Der Filenamen für die Unterdatei wird in der Regel immer ein anderer sein als der für die dazu passende Maske. Daher können Sie hier nicht Menüpunkt drei verwenden (Datei inklusive Maske laden), sondern müssen die Filenamen sowohl für die Datei als auch für die Maske angeben. Dieser Menüpunkt kommt auch in Frage, wenn Sie eine Hauptdatei laden möchten, deren Dateiname sich vom Filenamen der Maske unterscheidet. Das wird in der Praxis aber nicht so oft vorkommen.

S. Sortieroutine

Das Programm benutzt die in BASIC schnellste Sortierart, Quicksort. Trotzdem kann es, je nach Umfang der Datei, einige Zeit dauern. Beachten Sie aber, daß für diese Sortieroutine nur immer das erste Datenfeld relevant ist. Es ist also nicht möglich, eine Hauptdatei nach anderen Datenfeldern als Nummer eins sortieren zu lassen. Der Sortiervorgang verfolgt das aufsteigende Prinzip, also von A bis Z; oder, wenn's Zahlen sind, von niedrigsten bis zur höchsten. Ideal ist die Sortieroutine dann, wenn das erste Datenfeld Namen oder auch zum Beispiel Artikelbezeichnungen enthält, die zur besseren Übersicht auf der Liste alphabetisch sortiert sein sollten. Nach dem Sortiervorgang ist die bislang ungeordnete Datei nicht mehr verfügbar. Wenn Sie also nach dem Sortieren den Menüpunkt vier (Speichern) wählen, so wird eine sortierte Datei auf Diskette gespeichert.

E. Programmende

Das Programm wird ohne Sicherheitsabfrage beendet. Vergewissern Sie sich also vorher, ob Sie die sich im Computer befindende Datei abgespeichert haben, falls dies nötig sein sollte. Falls nicht, kommen Sie mit "GOTO 10000" ins Programm zurück. Allzu viele Sicherheitsabfragen in einem Programm haben für mich immer den penetranten Beigeschmack einer gewissen Unfähigkeitsbescheinigung des Anwenders, ausgestellt vom Programmautor.

HINWEISE

Möchten Sie das Programm *Check-Liste* von der COMMODORE DISC auf eine Arbeitsdiskette übertragen, geht das mit dem BASIC-Hauptprogramm sicher einfach. Beim dazugehörigen File „Menü“ müssen Sie so vorgehen:
COMMODORE DISC 17 in die Floppy, mit Taste F8 oder der Anweisung MONITOR den Maschinensprache-Monitor des C128 aufrufen.
Jetzt eingeben:
L "MENUE" 08 <RETURN>
Damit wird dieses File an den entsprechenden Speicherbereich geladen.
Nun die andere Diskette einlegen, auf die gespeichert werden soll.
Jetzt eintippen:
S "MENUE" 08 1300 16A6 <RETURN>
Nach Beendigung des Vorganges ist dieses File exakt

so auf Ihrer Arbeitsdiskette abgespeichert wie auf der COMMODORE DISC.

Noch etwas Allgemeines zur Eingabe für dieses Programm: Es handelt sich hier um die ganz normale INPUT-Routine, die der C128 im Betriebssystem verankert hat, mit allen Vor- und Nachteilen. Das heißt, Sie dürfen im Eingabetext – egal, ob das die Datenfelder oder die Filenamen sind – nie die „verbotenen“ Zeichen Komma, Semikolon und Anführungsstrich verwenden. Daß der Computer bei einer INPUT-Eingabe den Doppelpunkt als Trennzeichen verwendet (alle Zeichen dahinter werden bei INPUT nicht angenommen), ist Ihnen nach dieser Programmierläuterung bestimmt geläufig.

Und noch eines kann das Programm nicht: Zahlen berechnen und Ergebnisse von Addition oder Subtraktion ausgeben. Zahleneingaben werden als Zeichenkette (String) behandelt, also als reiner Text.

Batcher □

RAND & SPRITES

Platz- Anweiser

Es gibt einige Programme für den C64, die es Ihnen ermöglichen, den oberen und unteren Bildschirmrand für Text oder Grafik zu nutzen. Das normale BASIC 2.0 schafft das aber nicht, dazu muß eine BASIC-Erweiterung her.

Rand & Sprites ist ein reines Maschinensprache-Programm, das absolut geladen werden muß:

```
LOAD"R&S",8,1
```

Geben Sie nach dem Laden bitte NEW ein, um die Zeiger für den BASIC-Anfang wieder richtig zu setzen, und starten Sie diese Erweiterung mit SYS 49152.

Mit diesem Befehl wird die Initialisierungs-Routine aufgerufen. Erst danach ist es möglich, die neuen Befehle zu benutzen. Doch es geschieht noch mehr: Der Zeichensatz wird nach 57344 (E000) verschoben und steht hier auch für Änderungen zur Verfügung. Dabei verschiebt sich auch der Bildspeicher nach 52224 (CC00) sowie die Spritezeiger nach 53240 – 47 (CFF8). Dadurch bleibt der gesamte BASIC-Bereich für das Anwendungsprogramm frei, es ist sogar möglich, den alten, nun freigewordenen Bildspeicher zu benutzen, indem man vor dem Laden die Befehlsfolge

```
POKE43,0:POKE44,4:CLR
```

eingibt. Damit sind noch weitere 1024 Bytes für BASIC gewonnen. Des Weiteren wird der VIC auf Normalwerte gesetzt und der neue Bildspeicher gelöscht sowie die drei Pseudo-VICs angelegt.

Nach der Rückkehr aus der Routine hat sich also außer einem nun leeren Bildschirm nichts Sichtbares ereignet.

Benutzen wir nun die ersten neuen Befehle. Deren Syntax ist bewußt einfach gehalten. Alle Befehle, die den Rand betreffen, beginnen mit den Zeichen ←R, gefolgt von einem weiteren Buchstaben, der den Be-

AUF DISC IM HEFT

fehl symbolisiert. Darauf folgen bei einigen Befehlen noch zwei durch Komma getrennte Parameter. Um die Randbenutzung einzuschalten, dient der Befehl \leftarrow RE ohne weitere Parameter. Es ist jedoch möglich, als erstes nach SYS12*4096 auch jeden anderen Befehl zu benutzen. Das ist sogar sinnvoll, wenn man Randfarben und Sprites vorbereiten will. Diese erscheinen aber erst, nachdem der Befehl \leftarrow RE gegeben wurde. Um Rand und Sprites wieder auszuschalten, dient ein weiterer Befehl: \leftarrow RA. Auch dieser ohne weitere Parameter. Damit werden zwar die Randbenutzung und auch die Sprites abgeschaltet, aber mit \leftarrow RE ist alles wieder vorhanden. Dazu noch eine Bemerkung: Alle neuen Befehle können natürlich auch im Direktmodus benutzt werden. Das sollte man auch zuerst tun, um sich mit deren Wirkung besser vertraut zu machen. Es gibt noch zwei weitere Befehle, die den Rand betreffen. Da ist zuerst der Befehl \leftarrow RC,f,n. Damit werden die Farben (C=Color) gesetzt. Der erste Parameter enthält die Farbe von Null bis 15. Im zweiten wird gesagt, was gefärbt werden soll. Dabei bedeutet:

- 1 = obere Ecken
- 2 = untere Ecken
- 4 = Seitenränder
- 8 = Hintergrund oben
- 16 = Hintergrund unten
- 32 = Hintergrund Mitte.

Sie können jede beliebige Kombination dieser Werte benutzen. Wenn die Seitenränder und der Hintergrund oben in der Farbe Schwarz erscheinen sollen, geben Sie den Befehl \leftarrow RC,0,12 (oder \leftarrow RC,0,4+8) ein. Damit können Sie die Pokes 53280/81 vergessen. Sie haben auch keinerlei Wirkung mehr, höchstens flimmert bei ihrer Benutzung das Bild ganz kurz.

Mit dem letzten Randbefehl benutzen wir die Eigenart des VIC, den Inhalt der letzten Speicherstelle als Bitmuster auf dem Rand abzubilden.

Der Befehl \leftarrow RF,n erzeugt eben dieses Bitmuster in Schwarz. Der Parameter kann Werte von Null bis 255 annehmen. Entsprechend erscheint der Rand bei Null leer, bei 255 Schwarz. Alle anderen Werte erzeugen ein Strichmuster. Wenn hier immer wieder vom Rand gesprochen wird, so ist damit der obere und untere Teil des Randes gemeint, der durch unsere Befehlsweiterung nutzbar gemacht wird.

Verschiedentlich habe ich schon auf die Benutzung der Sprites hingewiesen. Wir wollen ja auch Text und Grafik auf den Rand bringen. Dabei helfen uns eben diese Sprites. Wir können insgesamt 24 Sprites benutzen und zwar auf dem oberen und unteren Rand je acht, und acht in der Mitte. Für die acht Mittelsprites gilt jedoch eine Einschränkung: Ihre Y-Koordinate muß sich innerhalb der Werte 59 bis 200 bewegen. Das Programm achtet auf diese Konvention und beachtet auf diese Werte nach 200, zu niedrige nach 59. Für die 16 Sprites auf dem Rand ist die Angabe der Y-Werte nicht nötig, sie werden immer auf die richtige Höhe gebracht. Dabei sollte man darauf achten, daß die Y-Koordinate am oberen Rand variiert, wenn Sprites mit einfacher oder doppelter Höhe definiert werden. Rand und Sprites achtet immer darauf, daß der unterste Rand dieser Sprites mit dem normalen Bildrand übereinstimmt. Eine Korrektur der Y-Werte erfolgt aber nur mit dem noch zu erklärenden K-Befehl. Das heißt, daß nach einer Änderung der Höhe der oberen Sprites der K-Befehl benutzt werden sollte. Nun ist es grundsätzlich möglich, alle 24 Sprites

Rand:	\leftarrow RE	schaltet Randbenutzung ein
	\leftarrow RA	schaltet Randbenutzung aus
	\leftarrow RC,f,n	setzt Farben ins Bild
	\leftarrow RF,n	setzt Randmuster schwarz
Sprites:	\leftarrow SOE,n	schaltet Sprites ein
	\leftarrow SOA,n	schaltet Sprites aus
	\leftarrow SOC,f,n	färbt Sprites
	\leftarrow SOF,w,n	füllt oder löscht Sprites
	\leftarrow SOB,b,r,s	setzt Zeichen in Sprites (Bildschirmcode)
	\leftarrow SOR,r,s	rollt Zeichen rechts herum
	\leftarrow SOL,r,s	rollt Zeichen links herum
	\leftarrow SOX,b,s	für einfache oder doppelte Breite
	\leftarrow SOY,h,s	für einfache oder doppelte Höhe
	\leftarrow SOP,p,s	setzt Priorität
	\leftarrow SOM,m,s	Multicolor ein- oder ausschalten
Ende:	\leftarrow END	Programmende

selbst zu steuern. Sie müssen dazu allerdings die drei Pseudo-VICs benutzen, die Rand und Sprites angelegt hat. Die Adressen dieser VICs sind:

für den oberen Rand

= 52016 (CB30) A = 64512 B = 240–247

den unteren Rand

= 51968 (CB00) A = 65024 B = 248–255

die Bildmitte

= 52064 (CB60) A = 64000 B = 232–239

Die Adressen A in dieser Tabelle zeigen, wo sich die Abbilder der entsprechenden Sprites befinden, mit den dazugehörigen Blocknummern (B). Sie können die drei VICs benutzen wie den normalen VIC, der für Sie aber tabu sein sollte. Sie sollten aber auf jeden Fall die obengenannten Bedingungen für die Y-Koordinate im Mittelfeld beachten. Für den oberen Rand gilt bei einfacher Höhe der Wert 27, bei doppelter Höhe der Wert sechs. Für den unteren Rand ist Y = 248.

AUCH ANDERE WERTE MÖGLICH, ABER . . .

Natürlich kann Sie keiner daran hindern, andere Werte in die entsprechenden Register zu poken, doch werden Sie bald merken, daß sich die Abbilder der Sprites gegenseitig beeinflussen. Oder aber, daß sich der Rechner ganz aufhängt. Doch diese Unannehmlichkeiten vermeiden Sie sicher, wenn Sie die neuen S-Befehle benutzen. Auf jeden Fall können Sie aber die Spriteabbilder durch POKEN in obengenannte Adressen nach Belieben selbst füllen. Den originalen VIC bei Adresse 53248 sollten Sie auf keinen Fall benutzen, denn die Sprites lassen sich darüber nicht beeinflussen, und alle anderen Register werden vom Programm Rand und Sprites verwaltet und es gäbe ein Durcheinander, wenn Sie da hinein POKEN würden.

Wie bei den R-Befehlen haben auch die S-Befehle eine einfache Syntax, allerdings mit einem Zeichen mehr. Jeder Befehl beginnt mit: \leftarrow S und zwei weiteren Buchstaben. Darauf können bis zu drei Parameter

AUF DISC IM HEFT

folgen. Der dritte Buchstabe definiert den Befehl selbst, also was geschehen soll. Der zweite Buchstabe gibt an, wo der Befehl wirken soll. Die sich bietenden Möglichkeiten sind:

O = oberer Rand
U = unterer Rand
M = Mitte
R = Rand oben und unten
A = Rand und Mitte (alle drei)

SPRITES MIT EIGENER ROUTINE FÜLLEN

Zur Erklärung gleich der Befehl, der die Sprites einschaltet:

←SAE schaltet alle Sprites ein
←SOE schaltet die acht oberen Sprites ein
←SME schaltet die acht mittleren Sprites ein usw.

Analog dazu schaltet der Befehl ←SOA die oberen Sprites aus. Also: dritter Buchstabe E = ein, A = aus. Doch es könnte ja sein, daß Sie nicht alle acht Sprites ein- oder ausschalten wollen. Nichts einfacher als das: Erweitern Sie den Befehl um ein Komma und die Nummer des Sprites (1 bis 8). So schaltet der Befehl ←SRE,5 im oberen und unteren Rand das fünfte Sprite ein. Auch hier wird der zweite Buchstaben richtig interpretiert.

Nun wollen wir den Sprites Farbe geben. Als drittes Zeichen geben wir jetzt C ein (Color). Die Syntax des Befehls ist:

←SOC,f,n

Hier, wie in allen ←S-Befehlen, ist der zweite Buchstabe durch den entsprechenden Zielbuchstaben (O, U, M, R oder A) zu ersetzen. Im ersten Parameter wird die Farbe (0 bis 5) abgelegt. Der zweite Parameter kann weggelassen werden, dann gilt der Befehl für alle acht Sprites. Oder wir geben die Nummer des gewünschten Sprites 1 bis 8 ein.

Als nächstes können wir einzelne oder alle Sprites mit einem Wert füllen. Der Befehl ist:

←SOF,w,n

Das Zeichen w steht für den Wert und kann von 0 bis 255 gehen. Bei Null werden die Sprites gelöscht, bei 255 gefüllt, während alle Zwischenwerte die Sprites mit Streifen füllen. Die Benutzung des dritten Parameters erfolgt analog zum Farbbefehl.

Selbstverständlich ist es immer noch erlaubt, die Sprites durch eine eigene Routine zu füllen, wie oben beschrieben. Unser Programm gibt uns aber die Möglichkeit, Zeichen aus dem Zeichensatz in die Sprites zu setzen. Das geschieht mit dem Befehl:

←SOB,b,r,s

Im ersten Parameter benutzen wir den Bildschirmcode des gewünschten Zeichens von 0 bis 127.

Soll das Zeichen revers erscheinen, addieren wir einfach 128 dazu, und falls wir den zweiten Zeichensatz benutzen wollen, werden nochmals 256 addiert. Auf dieser Art ist es sogar möglich, beide Zeichensätze gleichzeitig und nebeneinander darzustellen.

24 ZEICHEN NEBENEINANDER

Wenn wir uns die acht Sprites in einer Reihe nebeneinander vorstellen, können wir 24 Zeichen, eins neben dem anderen, unterbringen. Im Befehl geben wir als dritten Parameter an, in welcher Spalte (1 bis 24) das Zeichen erscheinen soll. Etwas anders ist es mit der Reihe. Bekanntlich enthält ein Sprite 21 Reihen. Wir entscheiden uns also für eine Reihe von 1 bis 14 und geben diese als zweiten Parameter ein. Da jedes Zeichen acht Reihen benötigt, belegt ein Zeichen in

Reihe 14 die Reihen 14 bis 21. Somit können wir ab Reihe 15 kein komplettes Zeichen mehr unterbringen. Doch der Befehl kann noch mehr: Wenn Sie den zweiten Parameter, die Reihe also, auf Null setzen, werden die acht Bytes des im ersten Parameter definierten Codes in den Speicherstellen 53232 bis 53239 abgelegt und sind durch einfaches PEEKen von BASIC aus zugänglich. Das erleichtert das Ändern einzelner Zeichen sehr, denn die Speicheradressen des Zeichensatzes ab 57344 sind nicht ohne Schwierigkeiten von BASIC auszulesen. Sie sollten diese Stellen aber sofort nach dem Befehl lesen, denn andere Befehle benutzen diese Adressen ebenfalls und damit wäre der Inhalt verändert und nützt nichts mehr. Mit den Parametern r und s arbeiten auch die beiden Rollbefehle. Mit diesen können Sie ein in den Sprites

NOCH DER WICHTIGSTE – DER K-BEFEHL

angezeigtes Zeichen nach rechts oder links rollen. Der Befehl R rollt das Zeichen nach rechts, und L nach links. Und so sehen die Befehle aus:

←SOR,r,s

←SOL,r,s

Es werden immer nur die acht durch r und s definierten Bytes gerollt, sie können immer nur ein Zeichen definieren. Wenn Sie jedoch als zweiten Buchstaben A oder R wählen, geschieht das in zwei oder allen drei Spritegruppen gleichzeitig.

Die nächsten vier Befehle sind vom Aufbau her einander ähnlich und sollen daher gemeinsam besprochen werden. Es sind die Befehle:

←SOX,b,s für einfache und doppelte Breite

←SOY,h,s für einfache oder doppelte Höhe

←SOP,p,s für Priorität der Sprites

←SOM,m,s für Multicolor-Sprites.

Der Parameter s gibt wieder an, welches Sprite betroffen werden soll, bei Null sind es alle acht. Bei b=0 hat das Sprite einfache, bei b=1 die doppelte Breite. So ist es auch bei h=0 oder h=1, nur daß es hier um die Höhe geht. Mit der Priorität ist es ähnlich: p=0 Sprite vor Hintergrund, p=1 Hintergrund vor Sprite. Wollen Sie Ihre Sprites in Multicolor erscheinen lassen, so geben Sie m=1 ein, normal ist m=0. Bei Multicolor müssen Sie die Sprites allerdings selbst laden.

Damit wären wir fast am Ende, aber einer der wichtigsten Befehle steht noch aus, das ist allerdings auch der letzte: K=Koordinaten. Wir wollen ja unsere Sprites nicht nur erzeugen, sondern auch zeigen und gegebenenfalls bewegen. Das ist der K-Befehl:

←SOK,x,y,n

Das ist ganz einfach. n ist wieder die Nummer des Sprites, das positioniert werden soll, also 1 bis 8. X und y sind die Koordinaten, die, wie bei Sprites gewohnt, anzugeben sind. Allerdings gelten die schon oben genannten Einschränkungen für die Y-Koordinate. Für den oberen und unteren Rand liegt diese Koordinate immer fest und man gibt hier irgend einen Wert von 0 bis 255 ein. Für die Mittelsprites liegt der Wert von 49 bis 200, darüber oder darunter liegende Werte werden automatisch korrigiert. Eines ist noch erwähnenswert: Wenn n=0 eingegeben wird, so werden x und y als Koordinaten des ersten Sprites interpretiert und alle Sprites erscheinen lückenlos in einer Reihe, so daß es einfach ist, ganze Spruchbänder über das Bild fahren zu lassen. Da die Abstände von Sprite zu Sprite je nach einfacher oder doppelter Breite verschieden sind, sollte gegebenenfalls vor dem K-Befehl der X-Befehl stehen, denn der K-Befehl wertet die Spritebreite aus.

G. Krämer □

So laden Sie Ihre Diskette

Bitte lesen Sie diese Ladeanweisung genau durch, dann gibt's beim Laden der COMMODORE DISC keine Probleme!

Die beiden Programme "Disclader.64" und "Disclader.128" befinden sich gleich als erste auf Ihrer neuen COMMODORE DISC.

Folgende Punkte sollten aber noch immer beachtet werden:

- 1) Schalten Sie Computer und Floppy ein.
- 2) Legen Sie die COMMODORE DISC in den Schlitz des Floppy-Laufwerks und verriegeln diesen.

Jetzt richtet sich Ihr weiteres Vorgehen danach, ob Sie C 64- oder C 128-Programme laden wollen.

Als C 64-Besitzer (oder Benutzer des C 64-Modus im C 128) geben Sie bitte jetzt ein:
LOAD".*";8

Meldet sich der Computer wieder mit "READY" und dem blinkenden Cursor, so tippen Sie das Wort "RUN" ein und schließen den ganzen Vorgang mit Druck auf die RETURN-Taste ab.

Besitzen Sie einen C 128 und befinden sich auch in diesem Modus, genügt es normalerweise, wenn Sie einmal kurz den RESET-Taster (seitlich rechts an Ihrem Gerät) betätigen oder den Befehl "BOOT" eingeben und

dann die RETURN-Taste drücken. (Beides erzeugt denselben Effekt, auf Ihrem Bildschirm erscheint die Meldung "BOOTING . . . DISCLADER.128")

Doch Achtung: wir haben vorher gesagt, "normalerweise". Falls jedoch aufgrund der Programmfülle auf der COMMODORE DISC der BOOT-Sektor von Daten bereits belegt ist, die ein anderes Programm

braucht, haben wir verständlicherweise auf die Installation dieses "Selbstlade-Sektors" verzichtet. Bei der COMMODORE DISC 6 ist dies z.B. der Fall. Laden Sie bitte den "Disclader.128" wie folgt: RUN "DISCLADER.128" (Nach dem Laden wird er sofort gestartet.)

3) Beim C 64 als auch beim C 128 erscheint nach einer kurzen Wartezeit das Anfangsbild des entsprechenden Disc-Laders, rufen Sie jetzt bitte nach Druck auf die Leertaste das Inhaltsverzeichnis der gerade aktuellen COMMODORE DISC auf.

4) Nachdem sich diese Programmliste auf dem Bildschirm aufgebaut hat (schön langsam,

zum Mitlesen!), erscheint vor den Programmnamen ein kleiner Pfeil, den Sie mit den Tasten CURSOR HOCH und CURSOR AB beliebig vor diesen Namen hin- und herbewegen können.

Als C 64-Benutzer können Sie nur C 64-Programme anwählen, als C 128-User eben nur die für diesen Computer.

- 5) Haben Sie sich für ein Programm entschieden, positionieren Sie den Pfeil vor dessen Namen und drücken die Taste "RETURN".
- 6) Das gewünschte Programm wird jetzt automatisch geladen und sofort gestartet.

Da die beiden DISC LADERS aber ebenfalls BASIC-Programme sind, werden sie von dem von Ihnen gewählten und so geladenen Programm gelöscht. Wollen Sie ein anderes Programm Ihrer COMMODORE DISC auf die gleiche Art, also unter Benutzung des DISC LADERS, in Ihren Computer holen, ist es notwendig, daß Sie zuerst auch wieder den DISC LADER auf die vorher beschriebene Art und Weise laden.

Diese Methode ist zwar relativ sicher vor Fehlern, die man machen könnte, aber - zugegeben - doch recht zeitraubend.

Darum für alle, die das Ganze "von Hand" erledigen wollen, Hinweise, wie die Programme der COMMODORE DISC noch geladen werden können:

- A) C 64-Benutzer
Punkt 1) und 2) (einschalten und Disk einlegen sollte klar sein), so daß es weitergeht mit
- 3) Geben Sie folgenden Befehl ein: LOAD "\$",8 und drücken Sie RETURN. Die Be-

zeichnung "\$" lädt das Inhaltsverzeichnis Ihrer Diskette.

- 4) Nachdem der Computer "READY" meldet, geben Sie LIST ein und drücken auch jetzt wieder RETURN. Nun erscheint auf dem Bildschirm das genaue Inhaltsverzeichnis Ihrer COMMODORE DISC.

5) Entscheiden Sie sich für das Programm, das Sie laden wollen.

6) Gehen Sie mit dem Cursor nach oben, bis dieser in der Zeile vor dem gewählten Programmnamen steht (an die Stelle, an der die Zahl der belegten Blocks eingetragen ist.)

7) Geben Sie an dieser Stelle den Befehl

LOAD ein. Bitte drauf achten, daß das erste „Gänsefüßchen“ vor dem Programm-Namen nicht überschrieben wird! (Die Zahl, die die Blocks angibt, dürfen Sie ruhig überschreiben.)

- 8) Gehen Sie nun mit dem Cursor hinter das zweite Anführungszeichen des Programmnamens und schreiben dort hin: ;8: (der Doppelpunkt dahinter ist sehr wichtig!)

9) Drücken Sie jetzt RETURN. Das Laufwerk beginnt nun, das gewählte Programm einzuladen.

Nachdem dies geschehen ist, starten Sie das Programm nur noch durch den Befehl RUN: und der Betätigung der RETURN-Taste.

Zur besseren Übersicht hier nun ein Beispiel, wie ein korrekter Ladebefehl in diesem Beispiel für das Programm "ROULETTE.64" auf dem Bildschirm aussehen muß:
LOAD"ROULETTE.64"

;8:
Hinweis: Bitte vermeiden Sie es, die Programme auf

**COMMODORE DISC:
KEIN ABTIPPEN**

Ihrer **COMMODORE DISC** mit dem Befehl **LOAD**" (Programmname) ",8,1 (oder ,8,8) zu laden, außer, es ist in der Programmbeschreibung ausdrücklich darauf hingewiesen.

Nun zu den Ladeanweisungen für den C 128 im 128er-Modus (für diejenigen, die auch auf den **DISC LADER.128** verzichten wollen):

- 1) Hier muß ebenso der Computer und das Laufwerk eingeschaltet werden.
- 2) Legen Sie auch hier die **COMMODORE DISC** in den Laufwerksschacht und verriegeln diesen.
- 3) Drücken Sie nun die Funktionstaste **F 3** (oben rechts, oberhalb der numerischen Tastatur). Mit dieser Taste bekommen Sie das Inhaltsverzeichnis der Diskette (**DIRECTORY**) angezeigt.
- 4) Wählen Sie ein Programm aus.
- 5) Gehen Sie mit dem Cursor vor den ausgewählten Programmnamen an die Stelle, an der die Blockanzahl steht, wie beim C 64, vor dem Anführungszeichen.
- 6) Tippen Sie ein **RUN** (Auch hier kann die Blockzahl ruhig überschrieben werden!)
- 7) Gehen Sie mit dem Cursor hinter das zweite Anführungszeichen des Programmnamens und machen dort mit der entsprechenden Taste einen Doppelpunkt. Hier bitte kein ",8" anfügen, sonst gibt's einen **"SYNTAX ERROR"**.
- 8) Drücken Sie **RETURN**
- 9) Nachdem das Laufwerk das Programm geladen hat, wird es sofort automatisch gestartet.

Auch hier ein Beispiel für eine korrekte Ladeanweisung (wie sie auf dem Bildschirm aussehen sollte) am Beispiel des Spiels **"MONOPOLY.128"**:

RUN"MONOPOLY.128":

Dies hört sich vielleicht etwas kompliziert an, ist es aber gar nicht! Diese Prozedur haben nicht wir uns ausgedacht, sondern die Entwickler des Computers. Es handelt sich dabei um die ganz normale Ladeanweisung für Programme von Diskette.

VIELE INFORMATIONEN

Übrigens: Das aufgelistete Inhaltsverzeichnis gibt Ihnen auch noch einige andere Informationen:

– Programme, die hinter Ihrem Namen kein **PRG** stehen haben, sondern beispielsweise ein **SEQ**, lassen sich mit den normalen Ladebefehlen (so wie vorher besprochen) gar nicht in den Computer einladen. Meist handelt es sich hier um **FILES**, die vom eigentlichen Programm nachgeladen werden und nur Daten enthalten. Dasselbe gilt für **FILES**, die mit **REL** (Relativ) oder **USR** (User) gekennzeichnet sind.

– Die Zahl vor den Programmnamen ist die Blockzahl. Sie gibt darüber Auskunft, welche Länge, in Diskettenblöcken, ein Programm besitzt.

– Außerdem erhalten Sie beim Inhaltsverzeichnis auch noch die Meldung **"XX BLOCKS FREE."**. Dies gibt darüber Auskunft, wieviel Platz (in Blöcken gerechnet!) noch auf der Diskette frei ist. (1 Block = 256 Bytes).

Und nun viel Spaß mit der neuen **COMMODORE DISC**!

Für Sie gelesen

Rodney Zaks:
CP/M Handbuch.
Düsseldorf: Sybex-Verlag, 1984.
356 Seiten.
ISBN 3-88745-053-1.
48 Mark

Dieses CP/M-Handbuch ist für Einsteiger geeignet, die Schritt für Schritt an die vielfältigen Möglichkeiten der Anwendung des CP/M-Betriebssystems herangeführt werden. An die Einführung in CP/M und MP/M schließt sich ein Kapitel über Funktionen, CP/M-Befehle, Dateinamen, Dateien, Systemgenerierung, Multitasking unter MP/M und CCP/M an, weiter geht es mit der Handhabung von Dateien und Diskettenoperationen. Der nächste Absatz enthält die Struktur von Betriebssystemen der CP/M-Familie wie Speicheraufteilung, Dateiverwaltung, BDOS-Operationen, Blocking und Deblocking, Multitasking, Rekonfiguration, Auto-Start und Anpassung von CCP/M. Eine Kurzbeschreibung der Befehle und Programme der verschiedenen CP/M-Versionen fehlt nicht.

Praktische Hinweise über Behandlung von Disketten, Benutzerdisziplin, Ausgabe, Dateien, Kommunikation zwischen CP/M-Systemen, Anpassung von CP/M-Software, Public-Domain-Software und Maßnahmen nach einem Systemfehler runden das Bild ab.

Zum Schluß folgt eine geschichtliche Betrachtung mit Entwicklungstendenzen des CP/M-Betriebssystems. Im Anhang finden sich noch einige Tabellen und Aufstellungen zu allgemeinen CP/M-Fehlermeldungen, Hexadezimal-Konversionstabelle, ASCII-Konversions-

tabelle, ED-Steuerbefehle, ED-Befehle, PIP-Gerätenamen, PIP-Gerätecodes, PIP-Parameter, CP/M- und MP/M-Befehletabelle, Steuerzeichen bei Befehlseingabe, Dateikennung, Begriffe und Abkürzungen. Also auch ein Nachschlagewerk für Fortgeschrittene. □

Bernhard Bachmann:
Arbeiten mit CP/M Plus. Commodore 128 PC.
Düsseldorf: Sybex-Verlag, 1987.
288 Seiten.
ISBN 3-88745-638-6
48 Mark

Ein Buch speziell für Commodore 128PC, Zilog Z80 und CP/M 3.0 von Digital Research. Wer seine Kenntnisse über CP/M in der vorgenannten Konfiguration beginnen oder gar erweitern will, erschließt sich den Zugang zu einer fast unbegrenzten Menge von meist professioneller Software. Nach der Systemeinführung geht es so richtig los mit Informationen über CP/M: geschichtliche Entwicklung, Bestandteile und Grundfunktionen von CP/M Plus, Speicheraufteilung – Banking, Diskettenorganisation – Dateisystem, die Hardwarekomponenten, fundamentale Charakteristika und Arbeitsschritte, transiente Kommando-Dateien und weitere Arbeitsprozesse unter CP/M plus. Es folgt das selbstverständliche Kapitel Befehlsübersicht mit 34 genau erklärten Befehlen, dazu Beschreibung, Eingabesyntax, Beispiel und Optionen. Mit dem tabellarischen Anhang, Steuerzeichen von CP/M Plus, ASCII-Tabelle in Dezimal/Hexadezimal/Binär, Umrechnungstabelle Hex-Dez, ADM 31-Terminal-Steuerzeichen, logischen und physikalischen Gerätenamen, reservierten Dateikennungen, reserviertem Zeichen und Stichwortregister endet das gelungene Buch.

Basic-Interpreter auf Trab gebracht

Jeder Computer versteht nur seine eigene Sprache, digitale Signale, die sich noch am besten als Binär- oder Hexadezimalzahl darstellen lassen. Sogenannte Hochsprachen wie BASIC, PASCAL, COBOL, LOGO und FORTRAN sind ihm fremd. Dazu benötigt er einen „Übersetzer“. Das kann ein Interpreter oder ein Compiler sein.

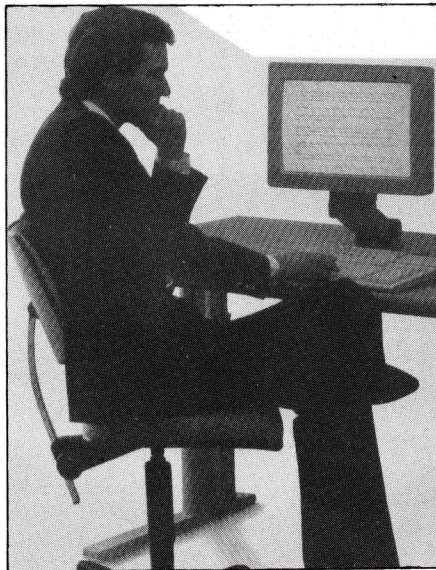
COMPILER LASSEN NICHTS ÜBRIG

Auf den Compiler möchten wir an dieser Stelle nicht ausführlich eingehen, nur soviel: Er übersetzt das ganze eingetippte Quellprogramm in eine dem Computer verständliche Maschinensprache, den „Objekt-Code“. Der Quelltext verflüchtigt sich nach der Compilierung aus dem Speicher.

Beim Interpreter sieht's anders aus, hier bleibt der Quelltext im Speicher bestehen. Wenn Sie nun „RUN“ eingeben, so wird oft – eigentlich fälschlicherweise – behauptet, das Programm würde damit gestartet. Stimmt nicht. Der Interpreter nimmt seine Arbeit auf und arbeitet den Programmtext, also das Listing, das im Speicher steht, Byte für Byte ab. Findet er nun ihm bekannte Byte-Folgen, wie die inzwischen hinlänglich bekannten BASIC-Befehle beziehungsweise deren Tokens (Abkürzungen), so springt er die dafür vorgesehenen Betriebssystem-Routinen an, um sie auszuführen. Die Tabellen, in denen der Computer vorher nachsehen muß, ob diese Anweisung überhaupt existiert, stehen im ROM (nicht veränderbarer Nur-Lese-Speicher).

Wenn in einer Schleife beispielsweise eine Programmzeile hundertmal durchlaufen wird, so muß auch das „Übersetzen“ hundertmal wiederholt werden. Der Compiler dagegen interpretiert nur einmal, diese Schleife wird gleich durch die entsprechende Routine in Maschinensprache übersetzt. Das ist auch der Grund, warum er um so schneller wird, je mehr Schleifen in Ihrem BASIC-Programm vorkommen, das Sie compilieren. Ein Nachteil soll allerdings nicht verschwiegen werden: Der entstandene Objekt-Code braucht in der Regel mehr Speicherplatz als das Original.

Der BASIC-Befehl LIST bringt uns jedes Programm auf den Bildschirm, wo wir es Zeile für Zeile lesen oder überprüfen können. Für den Computer sieht so ein Listing aber ganz anders aus, denn er speichert es auf seine eigene Methode ab. Wenn Sie eine BASIC-Zeile eingetippt und RETURN gedrückt haben, so wird sie bereits „verarbeitet“. Zunächst werden alle Befehle und Anweisungen in die vorgesehenen Abkürzungen umgewandelt, auch wenn Sie schon beim Eintippen die im Handbuch aufgeführten Abkürzungen der BASIC-Schlüsselwörter benutzt haben (etwa ein „?“ für den Befehl PRINT). Das hat mit den Tokens nichts zu tun. Hier handelt es sich um genau 128 verschiedene Ein-Byte-Werte, die eine Zahl zwischen 128 und 255 annehmen können.



Den Vorteil dieser Tokens haben Sie sicher schon erkannt: Würde der Befehl GOSUB genau fünf Byte im Speicher belegen, so wird das durch das zugewiesene Token auf ein Byte reduziert – der Interpreter muß nicht so viel übersetzen und kann damit bereits schneller arbeiten. Ein Beispiel zum besseren Verständnis: Diese BASIC-Zeile besteht aus 23 einzelnen Zeichen (Byte):
`IF C = 0 THEN GOSUB 100:
 GOTO 20`

Nach der Umwandlung in Tokens, die der Computer ja automatisch nach der Eingabe durchführt, sind es nur noch zwölf.

So knausrig der BASIC-Interpreter bei Wörtern ist, mit Zahlen geht er ein wenig verschwenderischer um. Es bleibt ihm auch gar nichts anderes übrig, denn der jeweilige Mikroprozessor des Computers versteht eben nur Zahlenwerte. Jede BASIC-Zeile wird noch um drei Byte erweitert. Zunächst wird dieser Zeile ganz am Anfang das Low- und High-Byte der Speicheradresse vorangesetzt, in der die nächste, so bearbeitete BASIC-Zeile zu finden ist; die Link-Adresse. Ohne sie wäre etwa ein GOTO oder GOSUB gar nicht durchführbar, noch viel weniger ein RENUMBER.

BEI ZAHLEN RECHT GROSSZÜGIG

Am Ende jeder BASIC-Zeile trägt der Interpreter – wohlgemerkt, bereits bei der Eingabe mit RETURN – noch eine Zahl ein, nämlich Null. Beim Arbeiten weiß er somit, daß die nächsten folgenden zwei Byte schlicht und einfach die Link-Adresse für die nächste BASIC-Zeile sein müssen, die nun abzuarbeiten ist. Hinter diesen beiden Link-Byte folgt immer die Zeilennummer, die stets aus Low- und Highbyte besteht, auch wenn es sich um Programmzeile „1“, „2“ oder „9“ handelt: 09 00. Was hier wie ein Nachricht aussieht, wird spätestens bei Zeilennummern über 99 zum Vorteil: Die Zahl „100“ besteht aus drei, die Zeilennummer „60000“ sogar aus fünf Byte. Das ist dem Computer egal: für ihn sind das immer nur zwei Speicherstellen.

ZAHLEN IM LISTINGS SIND TEXT

Für Zahlen aber, die im Listingtext vorkommen, gilt das nicht. Ein „GOTO1“ benötigt zwei, ein „GOSUB 60000“ jedoch sechs Byte vom immer zu knappen Platz im BASIC-RAM. Schon aus diesem Grund sollten Sie beim Programmieren daran denken, oft aufzurufende Unterroutrinen ziemlich weit vorn in Ihrem Programm anzusiedeln, um die Sprungadressenzahl im Listingtext möglichst klein zu halten. Noch einen Grund gibt's, der für diesen Tip spricht: Beim Suchen eines Unterprogramms beginnt der BASIC-Interpreter immer am Anfang, also ab Zeile „0“ zu suchen (egal, ob es diese Zeilennummer gibt oder nicht). Es ist wohl jedem klar, daß er eine Subroutine, angesprungen mit GOSUB 10, schneller gefunden hat und ausführen kann, als ein „GOSUB 50000“.

TIPS & TRICKS

Wie so eine Suchroutine abläuft? Es wird von Link-Zeiger zu Link-Zeiger gesprungen und die dahinter eingetragene Zeilennummer überprüft. Dieses „Zeilenhüpfen“ kann der Computer so realisieren, daß er immer anhand einer Zwei-Byte-Addition den vorherigen Link-Zeigerwert zum nächsten addiert. Somit wird uns wieder eine neue Erkenntnis zuteil: Da nur immer die Link-Zeiger und die darauffolgende Zeilennummer verglichen wird, ist es dem Interpreter bei dieser Suchmethode doch völlig egal, ob die Zeile nur aus einer Variablen oder aus vielen Anweisungen, getrennt durch Doppelpunkte, besteht. Da liest man immer soviel vom übersichtlichen und strukturierten Programmieren in BASIC, gut und schön, doch wenn Sie jede Variablenzuweisung (A = 10) oder Anweisung (PRINT A) mit einer eigenen Zeilennummer versehen, dann wird Ihr Programm bei der Ausführung gewiß langsamer laufen. Darum unser Rat, auch wenn's vielleicht mancher BASIC-Schulmeister nicht gerne hört: Je länger die Zeile, vollgepackt mit Befehlen und Anweisungen bis hin zur möglichen Länge (80 Zeichen beim C64, 160 beim 128er), desto mehr sparen Sie Speicherplatz und beschleunigen Ihr Programm.

VERÄNDERLICH: VARIABLE

Ein nicht zu unterschätzender Faktor im gekonnten Dialog mit dem Interpreter ist die Variable. Damit können Sie ganz erheblich die Geschwindigkeit und den Platzbedarf Ihres BASIC-Programms im Speicher beeinflussen.

Ein BASIC-Programm teilt sich in vier Blöcke im Speicher:

- Programmtext;
 - einfache Variable und Stringzeiger;
 - Indizierte Variable (Felder, auch Arrays genannt) und
 - Zeichenkettenvariable (Strings).
- Alle diese Arten werden erst während des Programmablaufs erzeugt und dabei in eigens hurtig angelegten Tabellen abgelegt. Dabei erhält jede Variable drei Erkennungsmerkmale:

Name, Typ und Wert.

In der Regel benötigt der Variablenname zwei Byte, auch wenn er nur aus einem Zeichen besteht (zum Beispiel A, B). Das Betriebssystem hat grundsätzlich zwei Speicherstellen reserviert, bei einer Ein-Zeichen-Variablen wird automatisch das andere Byte mit einer Null belegt.

Drei Variablentypen lassen sich unterscheiden:

Integer, Real, String.

Etwas wichtiges, bevor wir uns den Typen zuwenden, wollen wir nicht unerwähnt lassen:

WER ZUERST KOMMT, MAHLT ZUERST

Variable werden nicht geordnet, sortiert, in der Tabelle abgelegt, sondern so wie sie auftreten. Damit ist klar, daß die erste am schnellsten gefunden wird. Verstehen Sie jetzt, warum umsichtige BASIC-Programmierer gleich zu Beginn nahezu alle Variablen definieren, die im Programmverlauf auftreten, und die am häufigsten verwendeten auch bei der Definierung an den Anfang setzen? Bestimmt ist es Ihnen schon ab und zu unsinnig erschienen, in einem Listing gleich in den ersten Zeilen zu lesen: A=0: B=0 usw., obwohl im Programmverlauf ganz klar ersichtlich war, daß diese beiden Variablen nie und nimmer den Wert „Null“, sondern andere, höhere Werte beinhalten würden. Jetzt wissen Sie, warum. Auch zu den Variablennamen noch eine Anmerkung: Obwohl unsere beiden Computer C64 und 128PC längere Namen ohne weiteres akzeptieren, werden nur die ersten beiden Zeichen berücksichtigt, es genügt also, wenn Sie sich auf zwei spezifische Zeichen beschränken. Vor allem ist dann die Gefahr von gleichen Namen schon auf ein Mindestmaß beschränkt, denn auch bei längeren Variablennamen bedeutet für den Computer DATEI dasselbe wie DARLEHEN. Bei anschließenden Berechnungen kann dann ganz schöner Staß herauskommen, wenn etwa die Nummer der aktuellen Datei als Darlehenssumme fungiert. Der Computer kann nichts dafür, denn beide heißen für ihn DA. Sie können auch alphanumerische Namen verwenden, also Buchstaben und Ziffern gemischt, beispielsweise A1, B2, C9, nie darf dabei aber die Zahl vor dem Buchstaben stehen, etwa 1A, 2B oder 9C. Das erzeugt sofort eine Fehlermeldung.

WELCHEN TYP HAT IHRE VARIABLE?

Doch kümmern wir jetzt um die Unterschiede des Variablentyps. Integer-Zahlen sind immer ganze Zahlen, also ohne Kommastellen. Ein Tip: Wenn Sie in einem Programm so eine Zahl brauchen, so hängen Sie dem Variablennamen noch ein Prozentzeichen % an, etwa

A%=10, denn damit wird dem Computer exakt der Variablentyp vermittelt und vor allem nur zwei Byte für den jeweiligen Wert reserviert.

„Real“ dagegen bedeutet soviel wie „wirklich, echt“. Damit meint der Computer, daß es für ihn viel natürlicher ist, eine Zahl mit möglichst vielen Nachkommastellen, was im Fließkommaformat solcher Werte begründet liegt (die Erklärung hierzu wäre einen eigenen Artikel wert), nur soviel dazu: Realzahlen rechnet der Computer ungleich genauer aus als Integer-Werte, oft sogar zu genau. Tippen Sie einmal in Ihren C64 ein: PRINT 7 ↑ 7. Sie werden staunen, welches Ergebnis Sie erhalten . . .

BYTE-MONSTER: STRINGS

Echte Speicherplatzfresser sind Zeichenketten, also Strings, Variable, die immer ein Dollar-Zeichen \$ nach dem Variablennamen besitzen. Schließlich kann ein einziger String eine maximale Länge von 255 Zeichen besitzen, die Leerzeichen (SPACES) mitgerechnet. Wie soll das denn gehen, werden Sie jetzt fragen, wenn doch nur 80 (eigentlich genau 88) Zeichen beim C64 beziehungsweise 160 beim 128er in eine Zeile passen? Ganz einfach: Durch die Verwendung des Pluszeichens. Definieren Sie Ihre Zeichenkette in der ersten Zeile, zum Beispiel so:

```
10 A$ = "(jetzt kommt eine Zeichenkette mit der erlaubten Zeilenlänge)"
```

```
20 A$ = A$ + "(und jetzt kommt der Rest, bis es höchstens 255 Zeichen sind!)"
```

Ganz einfach, nicht?

Allerdings wirken sich solche Ungetüme recht störend in unserer Variablentabelle aus. Integer- oder Realvariable belegen grundsätzlich dieselbe Byte-Anzahl, eine Zeichenkette kann jedoch, je nach Programmverlauf, stets seine Länge verändern, so daß der Interpreter Schwerstarbeit leisten und seine Variablentabelle ständig hin- und herschieben beziehungsweise anpassen muß, um die unbedingt nötige Ordnung zu halten. Das wirkt sich nun wirklich nicht fördernd auf die Ablaufgeschwindigkeit Ihres BASIC-Programmes aus. Wir schlagen Ihnen folgende Lösung als Tip vor: Dimensionieren Sie (mit der Anweisung DIM) zu Beginn des BASIC-Programms auch ganz einfache Stringvariablen wie A\$, G\$\$ oder welchen Namen Sie auch immer gewählt haben, denn mit DIM zwingen Sie Ihren

Computer, einen Platz mit fester Länge in der Variablen-tabelle zu reservieren. Dann sieht das so aus: In der Tabelle stehen Name, Typ und ein Ziffer von drei Byte. Im ersten Byte die Länge der Zeichenkette und die beiden folgenden bezeichnen als Low- und Highbyte die Adresse, wo dieser String im RAM gespeichert wurde, all das bewirkt die DIM-Anweisung. Trotzdem: Ändert auch so ein dimensionierter String im Programm seine Länge, müssen die übrigen leider wieder verschoben werden. Und wohin? Wo eben noch Platz ist. Genauer gesagt, der besagte String wird dahin geschoben. Der Computer ändert dazu nur die zwei genannten Adressbytes. So kann es aber passieren, daß eine sich ständig ändernde Stringvariable genauso oft im Speicher – immer an einer noch verfügbaren Stelle – steht, obwohl ja nur eine, die zuletzt abgespeicherte, aktuell ist.

COMPUTER-INTERNE MÜLLABFUHR

Alle anderen Variationen sind Müll, Schrott. Bevor jedoch dieser Abfall in ungeahnte Dimensionen wächst, haben die Software-Entwickler eine automatische Müllabfuhr eingebaut: Die „Garbage Collection“. Sie wird durch eine Betriebssystem-Routine erzeugt, die in einer ganz bestimmten Speicherstelle nachsieht, bis zu welchem Pegel die Strings bereits angewachsen sind (beim C64 beginnt der String-Speicher am obersten Ende des freien RAM-Speicher-Bereichs, also bei Adresse 40959 und bewegt sich abwärts, beim C128 ist dafür eine eigene Speicherbank, nämlich BANK 1 vorgesehen), um dann eingreifen zu können.

Dieses Interpreter-Unterprogramm filtert die gültigen Strings heraus und baut die Tabelle neu auf. Im schlimmsten Fall kann das sogar einige Minuten dauern.

ZAHLEN ALS VARIABLE DEFINIEREN

Viel Speicherplatz kann gespart werden (und Sie wissen ja inzwischen, daß das auch ein nicht unwesentlicher Faktor bei der Ausführungsgeschwindigkeit eines Programms ist), wenn Sie häufig gebrauchte Zahlenwerte in einer Variablen speichern. Benötigen Sie beispielsweise oft den Wert „1000“ für Rechenoperationen, so definieren Sie ihn bitte, am besten zu Beginn des Listings, mit einem Variab-

blennamen, also $W = 1000$. Der Grund ist einfach: „1000“ bedeutet für den Computer vier Byte Text. Der Interpreter erkennt erst am Variablennamen (ohne \$), daß es sich bei diesem Text eigentlich um eine Zahl handelt und muß ihn in sein berechenbares Fließkommaformat umwandeln. Das muß er jedesmal, wenn eine Zahl als Text auftaucht. Ist sie aber mit einem Namen in der Variablen-tabelle abgelegt, geht der Zugriff auf die Variable „W“ ungleich schneller vor sich als eine ständig sich wiederholende Umwandlung ins Fließkomma-Format.

HAUPTAUFGABE EINES INTERPRETERS: ZEICHENBYTE HOLEN

Sicher ist Ihnen beim Lesen von Fachartikeln oder -büchern schon der Begriff „CHRGET“ untergekommen. Das bedeutet „Hole das Zeichen“, diese Systemroutine liegt beim C64 beispielsweise bei Adresse \$0073 (115). Der Inhalt dieser Speicherstelle ist ein Zeiger (Pointer), der auf ein Zeichen des Listingtextes zeigt, den Sie im Begriff sind, einzugeben. Diese Routine wird, solange Listingtext vorhanden ist, aufgerufen, der Pointer um eins erhöht und das Zeichen ins Akkumulator-Register der CPU (Mikroprozessor) verlegt. Nun beginnen einige Vergleichsoperationen: war es ein Leerzeichen, ein SPACE, dann nächstes Zeichen holen – war es eine Null, in der nächsten Programmzeile fortfahren – war es ein Befehl respektive ein Token, dann gehe dahin, wo die Routine im ROM steht, um diesen Befehl auszuführen.

JEDER BEFEHL HAT EIN EIGENES PROGRAMM

Da beginnt nämlich die Hauptarbeit des BASIC-Interpreters. Zu jeder Anweisung, zu jedem BASIC-Befehl gibt es im ROM-Speicher Ihres Computers ein mehr oder weniger großes Maschinensprache-Programm. Wo und an welcher Speicheradresse dieses Unterprogramm beginnt, muß sich der Interpreter erst aus einer anderen Tabelle herausuchen. Erst dann wird die Adresse angesprungen und die Anweisung ausgeführt.

Nehmen wir ruhig als Beispiel einmal die OPEN-Anweisung zum Öffnen eines Kanals, etwa zur Floppy. Diese Systemroutine für OPEN benutzt jetzt nämlich erneut das CHRGET-Unterprogramm (es dürfte sowieso zu den meist gebrauch-

testen Routinen des BASIC-Interpreters gehören). Der Zeiger ist nun um eine Stelle weitergerutscht und gibt die Filennummer, etwa „2“ an. Von der OPEN-Routine werden diese Zahl und die folgenden beiden (Gerätenummer und Sekundäradresse) an diverse Speicherstellen in der Zeropage verbracht, wo sie dann beispielsweise von einer anderen Operationsroutine der Floppy ausgelesen und entsprechend berücksichtigt werden können. Wir könnten jetzt beliebig fortfahren, doch gehört dieses Thema eigentlich eher in die Beschreibung eines Programmierkursus für Maschinensprache.

MEHR ARBEIT FÜR DEN INTERPRETER HEISST LANGSAMER PROGRAMMABLAUF

Wo könnte noch an der Geschwindigkeit eines BASIC-Programms gefeilt werden? Zum Beispiel möglichst wenig Rechenarbeit für den Interpreter in die bekannten FOR-NEXT-Schleifen legen. Nicht so gut ist etwa:

```
FOR I = 1 TO 100
NP(I) = EP(I)*5/100
NEXT I
```

Zur Erläuterung: Das kleine Programm errechnet aus 100 Einzelpreisen, die vorher schon eingegeben wurden oder als Daten bereits dem Computer vorliegen, den jeweils gültigen Rabattbetrag von 5%.

Besser ist hier, den Faktor „4/5/100“ als Variable vorher festzulegen, etwa: RB, also $Rabatt = 5/100$ (oder auch 0.05 ist möglich). Das schnellere Programm sieht dann so aus:

```
RB = 6/100
FOR I = 1 TO 100
NP(I) = EP(I) * RB
NEXT I
```

BASIC MUSS NICHT LANGSAM SEIN

Sie sehen also, daß es eine ganze Menge Tricks gibt (es gibt sicher noch mehr), um auch ein Programm im oft so hochnäsigen geschmähten BASIC doch in der Ablaufgeschwindigkeit recht akzeptabel zu machen. Sicher, wenn Sie nur den einen oder anderen unserer Tips befolgen, wird sich das kaum meßbar auswirken, wohl aber in der Summe der genannten Punkte zur optimalen „Konversation“ mit dem BASIC-Interpreter Ihres Computers. *hb* □

C – DIE SPRACHE DER ZUKUNFT

Multi-Talent

Kaum eine andere Sprache fand in der professionellen Software-Entwicklung so viele Anhänger wie die Programmiersprache C. Programme wie MS-DOS 3.0, WORDSTAR2000 oder GEM sowie Teile des Amiga-Betriebssystems sind in C programmiert worden.

Mit PROFI-C von Data Becker steht nun auch den C64/C128-Besitzern die wohl zur Zeit portabelste Hochsprache mit maschinennaher Programmierung zur Verfügung. Grund genug, uns mit der Sprache zu befassen.

DIE „SCHÖPfungSGESCHICHTE“:

Die Wiege von C stand im Bell-Laboratorium in Murray Hill, USA. Ihre geistigen Väter, Dennis Ritchie und Brian Kernighan, entwickelten sie 1972 aus der Sprache B, einer 1970 von Ken Thompson überarbeiteten Version von BCPL (Basic Combined Programming Language), die heute noch zur Entwicklung von Compilern und Betriebssystemen verwendet wird.

1974 war das entscheidende Jahr für C, denn das Betriebssystem Unix wurde zu 80 Prozent in C geschrieben. Damit war der Beweis erbracht, daß sich auch mit einer Hochsprache ein leistungsfähiges Betriebssystem entwickeln läßt. Unix ist immerhin Multiuser und Multitasking-fähig).

Der erste damit ausgerüstete Rechner war ein PDP-11. Von nun an waren C und UNIX nicht mehr zu trennen.

Bis 1977 wurde die Sprache mehrmals überarbeitet und mit dem Erscheinen des Buches „The C-Programming Language“ von Dennis Ritchie und Brian Kernighan (in C-Kreisen die „Bibel“ genannt) war ein gewisser Standard vorgegeben, der auch heute noch bei der Entwicklung von Compilern und Bibliotheken maßgebend ist.

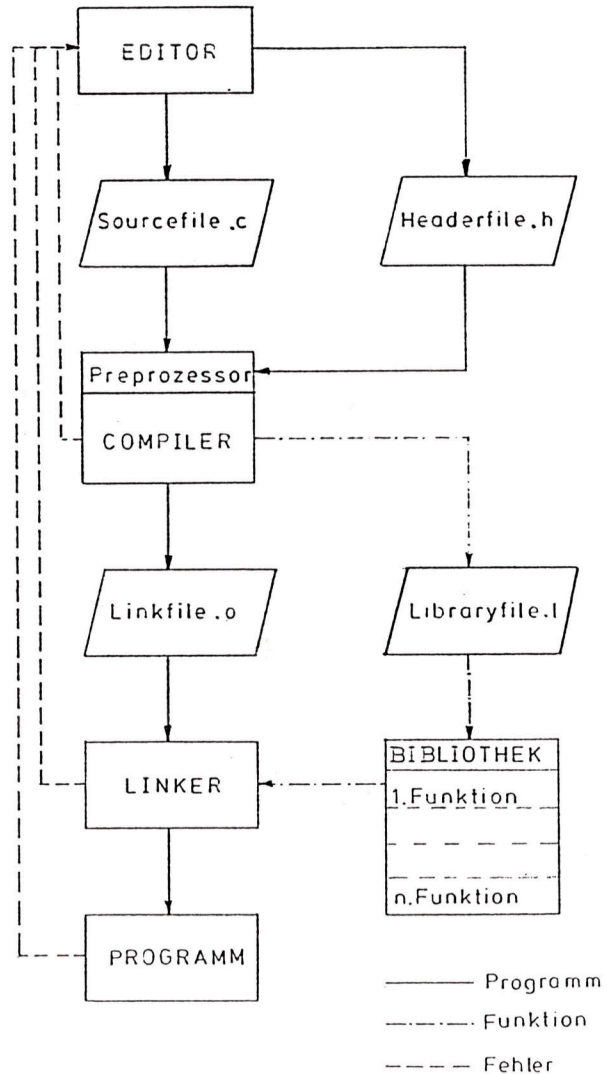
In den frühen 80er Jahren löste sich C von seiner UNIX-Umgebung (DEC-Rechner). Dies war endgültig der Durchbruch auf breiter Front. Mittlerweile gibt es C-Compiler auf fast allen Microcomputern und den gängigsten Betriebssystemen, zum Beispiel Acht-Bit-CP/M, CP/M 86, MS-DOS.

DIE GEGENWART

Die Sprache C ist eine sehr „kleine“, nicht problembezogene, maschinennahe Compilersprache. Es gibt im Sprachsatz von C keine I/O-Funktionen wie PRINT und INPUT im BASIC. Diese Funktionen sind in einer Standard-Bibliothek STDIO (STandarD Input/Output) abgelegt. Zu dieser Bibliothek gibt es ein Headerfile (stdio.h). In ihm sind die

Externen Funktionen aufgeführt. Dieses Headerfile wird vom Preprozessor in den Quelltext eingebunden und stellt das Bindeglied zwischen der Sprache C und dem System dar. Zur Verdeutlichung siehe Schema „Aufbau und Ablauf eines C-Systems“.

Aufbau u. Ablauf eines C-Systems



Zur Grundausstattung einer solchen Bibliothek gehören die Funktionen:

- printf() – formatierte Ausgabe Bildschirm
 - fprintf() – formatierte Ausgabe Datei
 - sprintf() – formatierte Ausgabe String
 - scanf(), fscanf(), sscanf(), als formatierte Eingabe sowie
 - putc(),getc(), getchar(), putchar(), gets(), fgets(), puts(), fputs(), fopen(), fclose(), um nur einige zu erwähnen. Sie gehören nicht unmittelbar zur Sprache C, sondern sind in C geschrieben.
- Das Headerfile und die Standard-Bibliothek werden in systemangepaßter Form mit dem Compiler zusammen ausgeliefert und können nach Belieben erweitert werden. Durch diesen Kunstgriff ist der Quelltext somit vom C64 bis zum SUPER CRAY ohne nennenswerte Anpassung übertragbar.

Durch dieses Konzept ist es auch möglich, kleine einheitliche Compiler zu verwenden – im Gegensatz zu Pascal, wo man sich immer mit sogenannten verbesserten Compilern herumschlagen muß. Dies ist auch der Grund, warum MODULA 2, aus demselben Hause wie Pascal, dieselben Wege wie C einschlägt.

DER PREPROZESSOR

Der Preprozessor ist eine Besonderheit von C. Er ist die dem Compiler vorgesetzte Instanz und verändert den Quellcode, bevor er vom Compiler bearbeitet wird. Durch spezielle Befehle, die immer ein „#“ vorangestellt haben, wird er aufgerufen. Einer der häufigsten ist der #include“stdio.h“. Hier wird durch den Preprozessor das Headerfile der Standardbibliothek eingesetzt. Es gibt noch weitere Befehle wie etwa die Möglichkeit, mit #define Markos definieren oder mit #undef wieder aufzuheben. Eine weitere Besonderheit ist die „bedingte Compilierung“. Hier können zum Beispiel verschiedene Versionen mit einem Quellcode erzeugt werden. Die Auswahl der zu compilierenden Quellcode-Bereiche erfolgt mit den Auswahlbefehlen: #if KONSTANTE. Wenn KONSTANTE ungleich Null ist, wird dieser Bereich übersetzt. Er muß dann mit #endif beendet werden.

Außerdem gibt es noch die Befehle #ifdef und #ifndef, die sich auch auf Makros beziehen. Der Preprozessor ist eine große Unterstützung bei der Arbeit und ein weiteres Plus für C.

DER SPRACHSATZ

Sehen wir uns den Sprachsatz von C genauer an. Es handelt sich in erster Linie um Datentypen und Kontrollstrukturen sowie eine reiche Auswahl an Operatoren.

1. Kontrollstrukturen:

Kontrollstrukturen definieren die Reihenfolge der Befehlsabläufe.

```
if else-Verzweigungen
if (Ausdruck)
    Anweisung 1
else
    Anweisung 2
```

C erlaubt hier eine Sonderform, nämlich die bedingte Bewertung:

```
max=(zahl1 < zahl2)?zahl1:zahl2;
```

switch:

Mit der „switch“-Anweisung kann man, im Gegensatz zu IF, unter mehreren Alternativen auswählen.

```
case Ausdruck:
    Anweisung 2;
    break;
case Ausdruck2:
    Anweisung2;
    break;
case Ausdruckn:
    Anweisungn;
    break;
default:
    Anweisungd;
```

Es wird der bei „switch“ angegebene Ausdruck ausgewertet und mit den „case“-Ausdrücken verglichen. Kommt es zu einer Übereinstimmung, so wird die nachfolgende Anweisung bis zur „break“-Marke ab-

gearbeitet. Sollte keine Übereinstimmung erfolgen, so wird die Anweisung nach „default“ ausgeführt.

for-Schleife:

Bei der „for“-Anweisung wird die Bedingung zum Schleifenanfang abgefragt.

```
for (Ausdruck1;Ausdruck2;Ausdruck3)
```

Anweisungen

Ausdruck eins initialisiert die Schleifenvariable. Ausdruck zwei liefert das Abbruchkriterium. Der letzte Ausdruck ist der Schleifenzähler.

while:

Die „while“-Schleife ist der „for“-Schleife sehr ähnlich. Auch hier wird die Schleifenbedingung am Anfang abgefragt.

```
while(Ausdruck)
```

Schleifenanweisung;

nächste Anweisung;

do...while-Schleife:

Der Unterschied zur „for“- und „while“-Schleife ist, daß die Schleifenbedingung erst am Ende abgefragt wird.

```
do
```

Schleifenanweisung;

```
while(Ausdruck);
```

nächste Anweisung;

continue:

Mit „continue“ kann man eine Schleife vorzeitig abbrechen und mit dem nächsten Durchlauf beginnen.

break:

Im Gegensatz zu „continue“ kann man mit „break“ eine Schleife vorzeitig verlassen, ohne mit dem nächsten Durchlauf zu beginnen.

return:

Ähnlich dem RETURN im BASIC beendet „return“ einen Funktionsaufruf. Der Rücksprung kann mit oder ohne Wertrückgabe erfolgen.

goto:

Als letzte Anweisung gibt es noch das verschmähte GOTO. Es sollte so selten wie möglich verwendet werden.

2. Datentypen:

In C existieren vier elementare Datentypen: char, int, float und double. Sie lassen sich noch mit den Attributen short, long und unsigned kombinieren. Dadurch ist die Größe des Speicherplatzes variabel.

char, unsigned char	–	Ein Zeichen aus einem Zeichensatz; ein Byte.
int, short, int, short	–	Eine Ganzzahl zwischen –32768 und +32767; zwei Bytes.
unsigned int, unsigned	–	Eine Ganzzahl zwischen Null und 65536; 2 Bytes.
long int, long	–	Eine Ganzzahl zwischen –214783648 und +2147483647; vier Bytes.
unsigned long int, unsigned long	–	Eine Ganzzahl zwischen Null und 4294967295; vier Bytes.

- float — Gleitkommazahl zwischen $\pm 9.09E-77$ und $\pm 6.78E+74$ auf sechs bis sieben Stellen genau; vier Bytes.
- long float, double — Gleitkommazahl zwischen $\pm 9.09E-77$ und $\pm 6.78E+74$ auf elf Stellen genau; sechs Bytes.

typedef:

Diese Anweisung ermöglicht die Definierung von „neuen Datentypen“. Ein schönes Beispiel haben wir in der Headerdatei „stdio.h“. In der Zeile sechs wird der Datentyp FILE definiert. Dadurch kann ihm die Funktion `io(FILE)` zugewiesen werden. Im Grunde ist das nichts anderes, als dem Kind einen neuen Namen zu geben.

void:

In neueren Versionen gibt es noch den Datentyp „void“ als Typ ohne Resultat. Dies kann man mit den Prozeduren in Pascal vergleichen.

3. Speicherklassen:

In C lassen sich Objekte in vier Speicherklassen vereinbaren:

auto:

Objekte der Speicherklasse „auto“ sind blockorientiert und werden nach dem Verlassen des Blocks wieder gelöscht.

static:

Objekte der Speicherklasse „static“ sind ebenfalls blockorientiert, behalten aber ihren Wert nach Verlassen des Blocks.

extern:

Objekte der Speicherklasse „extern“ sind programmorientiert und behalten ihren Wert auch bei getrennt übersetzten Funktionen.

register:

Objekte der Speicherklasse „register“ können direkt in die Prozessorregister gehalten werden, ansonsten verhalten sie sich wie „auto“.

4. Strukturen:

Wie in Pascal die Records-Strukturen, so lassen sich auch in C Strukturen vereinbaren. Hierzu stehen die Schlüsselwörter „struct“ und „union“ zur Verfügung. Im weiteren existiert in neueren Implementierungen das Schlüsselwort „enum“. Es ist dem Type bei Pascal vergleichbar.

C – DIE MASCHINENSPRACHE

Die nahe Maschinenprogrammierung wird bei C über das direkte Zugreifen mit Zeigern auf Adressen erreicht. Überhaupt haben die Zeiger (Pointer und Adressen) eine herausragende Bedeutung in C. Im Gegensatz zu Pascal, wo man erst spät oder gar nicht mit ihnen in Berührung kommt, sind sie in C ein unumgänglicher Bestandteil der Sprache. Sie sind transparent, dadurch gibt es kein „call by reference“ (Übergabe des Arguments selber), sondern nur „call by value“ (Übergabe des Argumentwerts). Die Möglichkeit der Zeigernutzung geht weit über an-

dere Sprachen hinaus. In C gibt es Zeiger auf alles, wie etwa auf Funktionen, Objekte und unbenannte Speicherbereiche von beliebiger Größe. Es läßt sich auch ein Datentyp-Zeiger auf einen Zeiger, der wiederum auf einen Vektor von Zeigern von Funktionen zeigt, deklarieren.

Hinzu kommen noch die Möglichkeiten der BIT- und ADRESSE-Manipulationen zum Beispiel UND, ODER, EXCLUSIV ODER und SHIFTEN (verschieben). Im weiteren lassen sich häufig benutzte Variablen in Prozessor-Registern halten, was den Zugriff darauf beschleunigt. Dieses trifft natürlich nicht für den 6502/10 Prozessor zu, da er nur zwei Register besitzt.

Ein derart umfangreiches Zeiger- und Bitverarbeitungskonzept ist zur Realisierung von systemnaher Programmierung unabdingbar.

C-Programme sind blockstrukturiert und bestehen aus mindestens einem Hauptprogramm `main()`, welches eine Reihe von Funktionen aufruft.

C – SICHER UND LEICHT?

Jetzt kommt der Punkt, an dem der C-Kritiker ins Lachen kommt: Programmieren in C ist so sicher wie ein Nitroglyzeringlas in der Hand. Zwar werden vom Compiler Syntaxfehler erkannt, dies sind aber nur die harmlosen Fehler.

In C kann man zum Beispiel das 50. Element eines Feldes mit 40 Elementen nicht nur lesen, sondern auch beschreiben:

Den Buchstaben a mit PI multiplizieren und mit dem Ergebnis eine Funktion aufrufen.

Mit C ist das alles möglich, die Liste ließe sich beliebig fortsetzen.

C besitzt keine Fehlerüberwachung und Plausibilitätsüberprüfung wie Pascal. Mag einem dies auf den ersten Blick unsinnig erscheinen, darf man nicht vergessen, daß eine gute Codeüberwachung die Effizienz eines Compilers stark beeinträchtigt. Auch unterbindet sie nicht nur manch Unnützes, sondern auch Nützliches.

C ist in erster Linie eine Sprache der Profis, da das Debugging der Programme (95 Prozent aller Fehler führen zum Absturz des Systems) äußerst schwer ist. Der vom Assembler her berühmte Griff zur RESET-Taste ist auch hier das Erkennungszeichen des C-Programmierers.

C UND DIE ZUKUNFT

C ist bis heute ohne Normung ausgekommen. Zwar gibt es einen Normvorschlag (ANSI X3J11), der auch über den von Ritchie und Kernighan gesetzten Standard hinausgeht, dieser ist aber in vielen Implementierungen schon vorhanden. Hierbei geht es um Zuordnung von Strukturen als Ganzes, die Einführung des Datentyps „void“ als Funktion ohne Rückwert und die Erweiterung des Preprozessor um die Möglichkeit, Strings zusammenzuziehen.

Während sich der Normungs-Ausschuß noch mit der Normung von C beschäftigt, steht schon ein neuer Stern am Himmel des Bell-Laboratoriums, C++. C++ bedeutet $c=c+1$, ein wesentliches Erkennungszeichen von C. Entwickelt wurde C++ 1983 von Bjarne Stroustrup. Es soll aufwärtskompatibel sein, das heißt, alte C-Quelltexte können übernommen werden. Wenn überhaupt, wird es wohl noch lange Jahre dauern, bis es auf breiter Basis durchbricht.

Gerhard Szymanski □

An uns – über uns

AUF DIE EINGABE KOMMT ES AN

Ich habe mir die COMMODORE DISC 14 gekauft und wollte mit UNI DATEI II arbeiten. Aber schon bei der Maske scheiterte ich, weil es da zum Beispiel heißt, man könne mit RETURN oder ENTER einen Datensatz abschließen, was mir leider nicht gelingen wollte! Nach mehreren Versuchen hatte ich auch ein paarmal einen SYNTAX ERROR im Programm. P.S. Ich würde mich auch freuen, wenn Sie mal ein in Deutsch übersetztes Mal- und Zeichenprogramm, das gut zu handhaben ist, auf DISC bringen würden.
Günther Rose,
Paderborn

Das Problem liegt unserer Meinung nach bei der Vorgehensweise zum Erstellen der Maske. Wenn das Programm fragt: „1. Datenfeld“, so geben Sie bitte darunter in folgender Schreibweise ein: Name des Datenfeldes, Länge als Zahl. Sehr wichtig ist dabei, daß Sie ein Komma zwischen beiden Angaben, eintippen. Diese drei Eingaben schließen Sie mit Druck auf die RETURN-Taste ab, und schon erscheint die Frage nach dem nächsten Datenfeld.

WIEDER DAS NEUE DOS 3.1!

Im Programm „Terminkalender“ hat sich ein Fehler eingeschlichen: In Zeile 1490 muß die Anweisung „DOPEN#7“ gestrichen werden, da die Datei „Tage“ bereits beim Einlesen des Programmes eröffnet wird. Sonst tritt ein Lesefehler im Unterprogramm „Datum suchen“ auf.

Ich hoffe, Ihr könnt mit meiner Information et-
was anfangen.
Werner Holstein,
Bremen

Das können wir ganz sicher, nur tritt dieser Fehler, wie bereits bei anderen Programmen auf verschiedenen COMMODORE DISCs auf, wenn der Benutzer des Programms einen 128 DCR, also ein Gerät neuerer Machart besitzt. Dem 128 D älterer Bauart, auf dem dieses Programm auch entwickelt und getestet wurde, würde das keine Schwierigkeiten bereiten, es tritt auch keine Fehlermeldung auf. Sie können sich helfen, indem Sie vor dem DOPEN-Befehl so ein File vorher schließen: DCLOSE#7.

VOLL DES LOBES

A priori möchte ich eines klarstellen: Die COMMODORE DISC ist die beste Erfindung, seit es Homecomputer gibt. Ohne Ihnen schmeicheln zu wollen, muß ich konstatieren, daß Sie auch an die C128-Besitzer gedacht haben und denken. Dafür gebührt Ihnen herzlicher Dank.

Ein kleines Problem hätte ich aber doch. Das Programm „Multi-Datei“ (COMMODORE DISC 6) ist sehr gut, hat aber einen Schönheitsfehler. Es druckt die deutschen Umlaute auf dem Commodore-Drucker MPS 1000 nicht korrekt aus. Ursache? (Vorausgesetzt, es wurde von mir kein Eingabefehler gemacht.)
Winfried Jäger,
Schweinfurt

Der MPS 1000 hat einen etwas anderen Druckerzeichensatz als andere, herkömmliche Drucker. Sie sollten das Listing „Multi-Datei“ folgendermaßen ändern und dann künftig die dafür vorgesehenen Tasten drücken.

Dann können die Umlaute auf dem Bildschirm genauso wie auf dem Drucker:

Zeile 60110

alt: C = 27

(SHIFT/Doppelpunkt = ä)

neu: C = 123

(COMM./F)

Zeile 60130

alt: C = 28

(englisches Pfundzeichen = ö)

neu: C = 124

(COMM./C)

Zeile 60210

alt: C = 29

(SHIFT/Semikolon = ü)

neu: C = 125

(COMM./X)

Zeile 60230

alt: C = 30

(Pfeil nach oben = ß)

neu: C = 126

(COMM./V)

Es handelt sich hier um dieselben Codes, die der DIN-Zeichensatz verwendet, Sie könnten dann auch praktisch von Haus aus die DIN-Taste einschalten. Allerdings sieht die Bildschirmausgabe bei den Menüs und Windows nicht besonders gut aus. Die Code für die großen Umlaute (ÄÜÖ) und der Paragraph können so bleiben, wie sie das Programm ursprünglich vorgesehen hatte.

AUSDRUCKEN GEHT OHNE WEITERES

Ich beziehe mich auf Ihre Ausgabe Nr. 15 der COMMODORE DISC und die „PC-Grafik“ auf Seite 12. Ich bin Neuling mit dem Drucker STAR NL-10 und würde gerne mein Gerät bei diesem Programm anwenden, es gelingt mir aber nicht. Bitte geben Sie mir einen Tip, welche Hardcopy-Routine ich wie einbauen kann.

Peter Aigner,
Neustadt 1



Ich bin Abonnent der COMMODORE DISC und möchte bei Ihnen zu

dem Heft Nr. 15 und zu dem beigegebenen Programm „PC-Grafik“ noch einige Fragen beantwortet haben. Ich bin noch nicht so vertraut mit dem C128 und möchte die Grafiken, die mit dem Programm erstellt werden, unbedingt auf meinen Drucker bringen. Es steht zwar im Programm, daß man ab Zeile 15000 eine eigene Hardcopy einbringen könne, aber wie mache ich das?
Ernst Mistler,
Köln

Als Abonnent besitzen Sie bestimmt auch die COMMODORE DISC 13, auf der sich ein Programm namens „HI-LO-PRINT“ befindet.

Laden Sie dieses Programm, bevor Sie mit „PC-Grafik“ arbeiten möchten, und starten Sie es mit RUN. Bringen Sie dann im Listing PC-GRAFIK, nachdem Sie es geladen haben, ab Zeile 15000 die Anweisung „SYS 4864“ unter und speichern Sie das geänderte Programm erneut ab. Wenn Sie jetzt aus dem Menü des Grafikprogrammes den Punkt „Drucken“ wählen, so findet der C128 im nun geänderten Listing die SYS-Anweisung und führt diese Druckroutine aus.

Allerdings nur, wenn Sie sie – wie bereits erwähnt – vorher geladen und mit RUN gestartet haben. Ist der Ausdruck beendet, kehrt das Programm wieder ins Menü zurück.

Falls der Drucker trotzdem störrisch sein sollte, so kann es eigentlich nur an der Sekundäradresse liegen.

Voreingestellt im Programm „HI-LO-PRINT“ ist die Eins für Epson- und kompatible Drucker. Wenn Ihr Drucker aber für den Ausdruck von hochauflösender Grafik die Null benötigt (zum

Beispiel STAR NL-10), so können Sie das leicht ändern; unmittelbar, nachdem Sie „HI-LO-PRINT“ geladen und mit RUN gestartet haben: POKE 5227,0.

DAS HEFT WAR NICHT DA!

Die Ausgabe Nr. 6 der COMMODORE DISC scheint es in Österreich nirgends gegeben zu haben. Als es zum gewohnten Termin noch nicht bei meinem Zeitschriftenhändler vorlag, erkundigte ich mich beim Österreich-Pressegroßvertrieb, der mir erklärte, es habe alles seine Richtigkeit. Die Nr. 6 erscheine näch-

sten Monat. Inzwischen ist aber das Heft Nr. 7 eingetroffen. Warum ist es nicht möglich, Ihre Zeitschrift auch in Österreich zu abonnieren?
Edmund E. Hofmannsrichter,
 A-1228 Wien

Auch wir wissen nicht, warum es ausgerechnet die COMMODORE DISC Nr. 6 nicht bei Ihrem Händler gegeben hat, aber das ist ausschließlich Sache des zuständigen Zeitschriften-Vertriebs. Die Ausgabe Nr. 6 können Sie aber noch bei uns erhalten. Aus Gründen der Zollbestimmung ist leider kein Abonnement ins Ausland möglich.

PROGRAMM STIMMT, ABER DER DRUCKER NICHT

Hiermit möchte ich Sie auffordern, das Programm „Disc-Box“ genauer zu überprüfen. Denn das Programm druckt die von mir erstellte Liste nicht auf meinem Drucker, STAR SG-10C, aus. Ich hoffe auf baldige Antwort.
Stefan Tysbierek,
 Castrop-Rauxel

Da brauchen wir nichts überprüfen; das Programm druckt tadellos aus, wenn Sie einen Epson-kompatiblen Drucker oder einen mit entsprechendem Interface benutzen. Wenn's bei Ih-

rem Drucker nicht klappt, so tut uns das leid. Aber bedenken Sie bitte, daß es sich wie bei allen anderen Programmen auf der COMMODORE DISC um Leserprogramme handelt, die halt nicht so bequeme Druckertreiber oder Anpassungsprogramme bieten. wie kommerzielle Software. Diese aber ist auch ganz erheblich teurer als die COMMODORE DISC mit all ihren Programmen. Falls Sie etwas von Maschinensprache verstehen, so müßten Sie mit einem Assembler oder Maschinensprache-Monitor in das Programm einsteigen und die entsprechenden Drucker-codes und -adressen anpassen.



SPAREN SIE 40 MARK! COMMODORE-DISC JETZT IM ABO

ABO-SERVICE

COUPON

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die ABO-Gebühr dem Verlag zugegangen ist

COMMODORE DISC
 ABO-Service 17
 Postfach 1161
 D-8044 Unterschleißheim

(gültig nur innerhalb der Bundesrep. Deutschland und Westberlin)

Ja, ich möchte von Ihrem Angebot Gebrauch machen. Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Wenn ich nicht vier Wochen vor Ablauf kündige, läuft diese Abmachung automatisch weiter.

WICHTIG! Sie können diesen Auftrag binnen einer Woche nach Zugang der Abo-Bestätigung widerrufen!

Es genügt die rechtzeitige Absendung

Name _____
 Vorname _____
 Straße / Hausnr. _____
 PLZ / Ort _____

Ich bezahle DM 200,- (inkl. Mehrwertsteuer) statt 237,60 für die nächsten 12 Ausgaben

- per beiliegendem Verrechnungs-/Euroscheck
- gegen Rechnung
- bargeldlos per Bankeinzug von meinem Konto bei Bank und Ort: _____
 Kontonummer: _____
 Bankleitzahl: _____
 (steht auf jedem Kontoauszug)

Unterschrift _____
 Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

2. Unterschrift _____

COMPUTERN LEICHT GEMACHT

Das PC-Magazin

Nr. 8/88 August/September · DM 7/6S 56/Sfr. 7



NEU

Jetzt an ausgewählten
Kiosken und im
Bahnhofs-Buchhandel