

COMMODORE CW WELT

KLAMM- HEIMLICH

Die Floppy 1541 gibt es
nicht mehr, der Nachfolger
bietet erheblich mehr

BASIC- TOOL

39 neue Befehle
für Ihren C 16

SPORT:

Rennerprobt: Der C 64
als Zeitnehmer
im Eiskanal

SERIE

Es muß nicht
immer Basic sein

EDV

Computer-Wissen
für Einsteiger

Wer hat Angst vor der CeBIT 1988?

Dem Erfinder fällt nichts mehr ein. Den Hersteller drückt die Konkurrenz. Der Händler rauft sich die Haare. Der Programmierer ist wütend. Der Amateur reagiert frustriert, der Einsteiger bloß noch ratlos.

Wenn all diese Umstände an einem Tag zusammenreffen, hat der Münchner den Föhn als Erklärung. Wie lautet die Entschuldigung bei der Computerindustrie?

Die Tage bis zur nächsten CeBIT, dem „Welt-Centrum der Büro- und Informationstechnik“ in Hannover (16.-23. März), sind gezählt.

Was wird es Neues geben, wer wird wo welche Rückzieher machen? Gerüchte und Trendmeldungen gehören zum Vorfeld jeder Fachmesse wie das Klappern zum Handwerk.

Grundlegende Frage: Wird der 16-Bit-Computer seinem Vorgänger endgültig Rang und Käufermärkte ablaufen? Ein neuer C64 mit einem eingebauten Amiga soll marktreif sein. Pech für viele, die sich bei der zeitlich rasanten Entwicklung auf

dem Computermarkt in der Situation sehen, ihre Programme, Disketten, ihre Einrichtungen alle zwei Jahre auf den Müll werfen zu müssen.

Gemach: Für den Amiga gibt es einen C64-Emulator, bei der entsprechenden maschinellen Einrichtung können alle C64-Programme verwendet werden. Der Emulator erspart also den Kauf eines zusätzlichen Rechners. Was aber, wenn der schon vorhanden ist? Lassen die Hersteller den Verbraucher dann im Regen stehen?

Der „Knüller“ zur CeBIT soll bei Commodore ein neuer C64 mit 1541- oder 1581-Laufwerk sein – mit 5.25- oder 3.50-Zoll-Floppys, das Ganze in neuem Design, versteht sich.

Bleibt abzuwarten, ob das neue Gerät nur ein Verschnitt des SX64 ist, oder ob es die Technologie des Jahres 1988 aufweist: ein C64 mit besserem BASIC, fehlerfreiem Betriebssystem, höherer Taktfrequenz, mehr Speicher, schnelleren Disketten-Operationen und anderen eingebauten Vorteilen. COMMODORE WELT wird das Geheimnis für Sie lüften.

Auch bei anderen Novitäten hat COMMODORE WELT die Nase vorn. In diesem Heft finden Sie einen Testbericht über das neue Laufwerk 1541-II. Es ist Beweis dafür, daß die Acht-Bit-Ära noch nicht abgelaufen ist, – wenngleich die Technologie kräftig auf 16 und 32 Bit marschiert.

Das größte Programmangebot aller Computer der Welt ist auf den C64 zugeschnitten. Allein in der Bundesrepublik Deutschland gibt es anderthalb Millionen C64-Besitzer. Für viele war

und ist der C64 der Einstieg in die Computerwelt. Das wird sicher auch so bleiben.

Die Leistungsfähigkeit des C64 ist noch längst nicht ausgereizt. Das beweisen immer wieder Programmierer, die aus dem grauen Kasten Programme herauskitzeln, auf die ein 16-Bit-Gerät stolz sein könnte.

So ist es für die Hersteller auch eine Frage der Produktpolitik, ob sie sich aus breiten Märkten heraushehlen und die Käufer ihrem Dilemma überlassen wollen.

Kein Anwender, ob Profi oder Amateur, wird sich gegen modernere, komplexere Technologien wehren. Doch er erwartet – mit Recht – von den Herstellern Unterstützung. Die kann zum Beispiel schon bei Anleitungen und Handbüchern durchaus verbessert werden. Denn hier entsteht häufig der Eindruck, Hersteller würden überhaupt keinen Wert legen auf Newcomer in der Computerwelt.

Doch außer der „Bahnsteigkarte“ und besserer Technologie erwarten Anwender bedienungsfreundlichere Geräte, besseren Service bei Reparaturen, weniger Qualitätsstreuung, größere Kompatibilität und ausgereifte Modelle. Kein Anwender will unfreiwillig Versuchsperson für ein Entwicklungsmodell sein. Überzogene Forderungen? Wir meinen: Nein.

Denn, ins Stammbuch der Hersteller und ihrer Vertriebsfirmen geschrieben, der „User“ ist – der Käufer.

In diesem Sinne, bis zu den Berichten um die CeBIT Ihre Redaktion
COMMODORE WELT

IMPRESSUM

COMMODORE WELT

erscheint monatlich in der CA-Verlags GmbH (i.G.)

CHEFREDAKTEUR
(verantwortlich für den Inhalt): ANTON KULT

Ressort 16/116/P4:
Alfons Mittelmeyer

Ressort VC20/C64:
Anton Kult

Ressort 128:
Harald Beiler

Ressort Amiga:
Torsten Seibt

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

GESCHÄFTSFÜHRER
(und verantwortlich für Anzeigen):
Werner E. Seibt

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Postfach 1161
8044 Unterschleißheim
Telex: 5214428 cav-d
Es gilt Preisliste Nr. 8 vom 1.1.1988
Media-Unterlagen bitte anfordern.

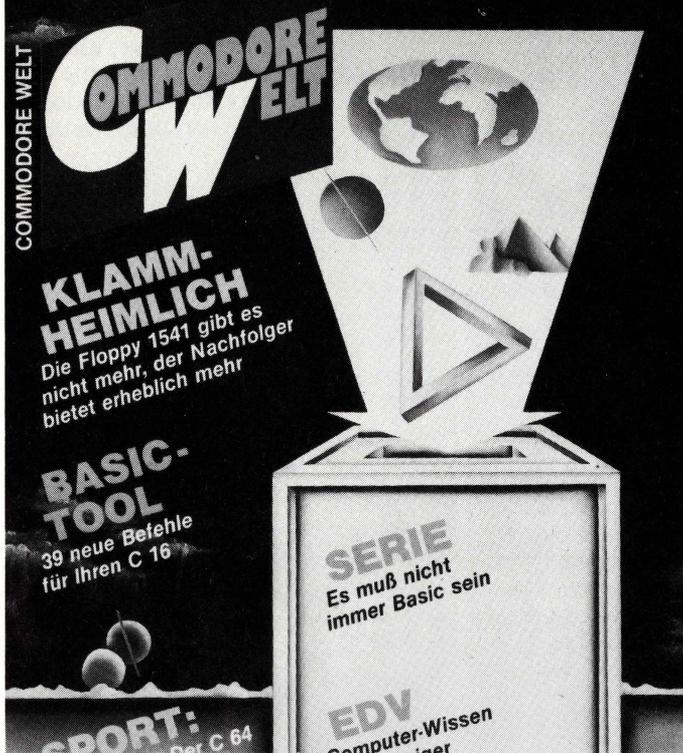
ANZEIGENREPRÄSENTANZ:
Verlagsbüro Wolf,
Veichenweg 2
7909 Dornstadt
Tel. 07348/21394

ANZEIGENSACHBEARBEITUNG:
Angelika Kilches-Wienstein
Tel.-Nr. der Anzeigenabteilung: 089/18 40 22

VERTRIEB:
Verlagsunion Wiesbaden

© 1988 by CA Verlags GmbH (i.G.)
Printed in Germany

Nr. 2/88 · Februar · DM 5,50 – ÖS 46 – SFR 5,50 · 5. Jahrgang



Titelfoto: Siehm und Partner

WISSENSWERTES

Vorschau zur CeBIT 1988

SERIE & REPORT

C64 Semiprofessionel
Der C64 im Einsatz als Zeitnehmer und Daten-Auswerter bei Internationalen Bob- und Rodelwettkämpfen im Eiskanal vom Königsee/Berchtesgaden
Seite 4

Bessere Berufschancen
EDV-Wissen ist gefragt denn je. Aber reichen die Anstrengungen in den Kultusministerien und Institutionen? Es liegt noch einiges im Argen
Seite 6

Plotter 1520
Das verkannte Genie im kleinen Kasten. Mit Ausstattungsmerkmalen, die ein Matrixdrucker nicht bieten kann
Seite 10

Deutscher C64
BASIC-Befehle des C64 in Deutsch – oder Fehlermeldungen in einer beliebigen Sprache
Seite 11

TIPS & TRICKS

KEY – das Zauberwort
Funktionstastenbelegung nach freier Wahl, auch auf dem C64 einfach zu realisieren. Mit Programm
Seite 13

Superbase Tricks
Erstellung von Bildschirmmasken bei Superbase. Freiprogrammierbare Daten-Felder endlich im Griff
Seite 15

Die Seite 16
Vorsicht vor ERR\$, ST und DS\$ – Floppy Tester und Programm – Sicherheit im Umgang mit der Floppy – High oder Double Sided – C64 > = < C16
Seite 16

Computer-Grundlagen
Lernen Sie die Prozessor-Befehle zur Realisierung einer Programmlogik kennen – Bescheid wissen = besser programmieren
Seite 18

Sortierroutine
Strukturiertes Sortieren: Aus einem bunt zusammengewürfelten Haufen eine mathematisch berechenbare Menge, auch für Nicht-Mathematiker
Seite 23

Schwarze Scheiben
Licht ins Disketten-Dickicht. Das kleine ABC beim Diskettenkauf
Seite 24

Suchen und Finden 128PC
Datei für den 128 im 80-Zeichen-Modus. Eingeben – Speichern – Anhängen – Ausgabe auf Monitor oder Drucker und einiges mehr an Möglichkeiten
Seite 26

Textverarbeitung
Vergleiche, Referenzlisten, T & T Konvertierungstabellen zu den Textverarbeitungsprogrammen SCRIPT
Seite 32

DAS WAR 1987
Jahresinhaltsverzeichnis der COMMODORE WELT nach Themen, Ausgabe, Seite und Stichwörtern
Seite 56

LISTINGS

BASIC TOOL C16
39 neue Befehle für Ihren C16/Plus4, zum Beispiel OLD – WINDOW – KILL – Editier-, Sound- und Grafikhilfe
Seite 35

BASIC TOOL DEMO Seite 37
BASIC TOOL PROGRAMM Seite 38

RETOUR BRETTSPIEL C16
Eine Version von Reversi auf den 3.5-Computern von Commodore
Seite 46

RAINBOW C16
Unterhaltsames Strategiespiel für Schnelldenker
Seite 50

TITELBILD-DESIGNER C16
Mit Hilfe eines Screen-Editors ein Titelbild entwerfen und in Ihr eigenes BASIC-Programm einbinden
Seite 53

PFADFINDER C64/C16
Auf der Suche nach verlorenen Texten – Textverarbeitungsprogramme auf dem Prüfstand – Hilfstool für Vergeßliche
Seite 59

VARIABLENSUCHE C64
Find C64, ein Programm-Utility zur Verhinderung von Variablenfehlern. Der Doppelvergabe auf der Spur
Seite 63

TEST & TECHNIK

BAD CAT
Spieletest zur Olympiade der Katzen. Exotische Disziplinen in der Stadt
Seite 65

BÖRSE
Für jeden Geschmack etwas. Unser beliebter, kostenloser Kleinanzeigen-Service
Seite 69

FORTRAN COMPILER
Keine Angst vor harten Nüssen – Programmierlösungen in FORTRAN
Seite 72

1541-II
Das Neueste aus der Commodore-Bauteile-Kiste, das 1541-II Floppy-Laufwerk
Seite 76

ZEITMESSELEKTRONIK MIT DEM C64

Alles hört auf den C64 - stop!

Jeder von Ihnen ist bestimmt schon voller Spannung vor dem Fernseher gesessen und hat ein mörderisches Skirennen auf eisglatter Piste verfolgt. Den anderen interessieren Bob- und Rodelwettkämpfe.

Eine in diesem Zusammenhang als selbstverständlich empfundene Serviceleistung hat **COMMODORE WELT** unter die Lupe genommen: die Zeitmessung und Auswertung der Ergebnisse.

nehmens steht außer Zweifel, schließlich war Dieter Preiss, der neue Leiter der Entwicklungsabteilung von Commodore, Schüler von E. Dolag. In wochenlanger, akribischer Kleinarbeit wurde ein hieb- und stichfestes System ausgeklügelt, das noch heute unverändert seinen Dienst tut.

TECHNISCHE RAFFINESSE: DIE ZEITMESSANLAGE

Die Zeitmeßelektronik hat verschiedene Aufgaben: Messen der jeweiligen Start-, Zwischen- und Einlaufzeiten, sie auszuwerten, festzuhalten und anzuzeigen. Entlang der

Die End- oder Zwischenzeiten eines Rennlaufes zu messen, zu ordnen und dann in der richtigen Reihenfolge auf die Ergebnistafel zu übertragen ist, obwohl die Vorgänge heutzutage reibungslos und meist fehlerfrei ablaufen, die Summe komplizierter rechnerischer Abläufe.

Unser Studienobjekt war die Bob- und Rodelbahn im Bundesleistungszentrum Königsee in Berchtesgaden. Dort werden die Zeitauswertungen durch Commodore-Computer vorgenommen, und zwar durch einen C64. E. Dolag, Inhaber eines bekannten Spezial-Elektronik-Unternehmens im Münchner Osten, lud uns ein, die Anlage zu besichtigen.

ES MUSS NICHT IMMER EIN TEURER PC SEIN

Der in Berchtesgaden installierte C64 ist kein Einzelfall: er steht auch an der Rodel- und Bobbahn Igl in Innsbruck; Verhandlungen mit dem Wintersportzentrum Winterberg in der DDR scheinen erfolgreich zu sein. Ja, der gute alte Brotkasten kann eben doch einiges mehr, als nur Weltraum-Invasoren abzuschießen. Ganz allein kann er's freilich nicht, er ist eher unverzichtbares Glied in einem ausgeklügelten Auswertungssystem. Der „Vater“ dieses Systems ist E. Dolag. Die Vorgeschichte: 1979 wurde die Bob- und Rodelbahn Königsee erbaut, die unvermeidliche Anzeigetafel (für die laufende Zeit und die Siegerreihenfolge) wurde von einer Schweizer Firma entwickelt, allerdings nur mit der Anzeige der ersten drei Bestplatzierten. Im Jahre 1983 war das dem zuständigen Komitee aus Königsee nicht mehr genug, es wollte die ersten fünf Sieger auf der Tafel sehen. Der Auftrag zur Änderung der Großanzeigetafel wurde an die Firma Edotronik vergeben. Das Know-how dieses Unter-



Rennbahn stehen 19 Lichtschranken, die den Meßimpuls auslösen, wenn sie durchfahren werden. Diese Signale werden einem Gerät zugeführt, das von allen recht salopp Klavier genannt wird – wohl wegen der Anordnung der Tastatur. Diese Eigenentwicklung von Edotronic arbeitet mit einer Prioritätslogik. Es werden Lichtschranken ausgewählt, die vor dem Start von der Aufsichtsperson im Starterhaus eingestellt werden (bei unserem Besuch in Königsee war es gerade der Neffe des Rodelweltmeisters im Zweier, Stangassinger). Diese Lichtschranken leiten die Zeiten an zwei Zeitmeßcomputer weiter, von denen ei-

ner im Starterhaus steht. Der zweite Computer steht an einem sicheren Ort (im Nebenraum), zur Sicherheit, falls der im Starterhaus ausfällt. Dieser Spezial-Computer ist natürlich kein C64, damit wäre er heillos überfordert. Allerdings würde auch kein

SICHER IST SICHER: ZEITEN WERDEN DOPPELT GEMESSEN

großer PC oder ein Amiga annähernd die Genauigkeit erreichen, mit der der Alge-Sport Timing-Zeitmeßcomputer aus Österreich arbeitet. Auf einem schmalen Papierstreifen werden die Zeiten zur Kontrolle ausgedruckt,

außerdem an einen Steuerblock der Anzeigetafel geschickt, der für die Anzeige der Startnummer und der aktuellen Zeit zuständig ist. Damit hat dieser Zeitmeßcomputer, der TDC 4000 heißt und immerhin 4000,- Mark kostet, der auch ohne Stromversorgung mit Batterie betrieben werden kann, seine Arbeit erfüllt. Allenfalls die Einblendungen der Startnummer und der laufenden Zeit, die Sie am Fernsehbildschirm verfolgen können, gehen auf sein Konto.

DIE ARBEIT DES C64 BEGINNT

Im folgenden Verlauf sind die weiteren Teile des Zeitmeßsystems gefordert, wobei der C64 die Hauptarbeit leistet. Dieser Commodore-Computer muß die so gewonnenen Zeiten ordnen und zur Reihung verarbeiten. Nicht zu vergessen: Diese Daten werden auf die große Anzeigetafel weitergeleitet, die für Fahrer, Trainer und Zuschauer eine ständige Kontrolle des momentanen Standes des Rennens gibt. Seine Daten erhält der C64 dafür von der Alge-Uhr durch ein Parallelkabel, das am Userport angeschlossen ist. Mit einer abgestimmten Software, die teils aus Assembler-routinen, teils aus compiliertem BASIC (hier erwies sich der gute, alte Pet Speed-Compiler für den C64 als am geeignetsten) besteht und als ROM-

DATENDISKETTE WIRD IM DIREKTZUGRIFF BESCHRIEBEN

Modul in den Erweiterungsport des C64 gesteckt wird, werden die Daten gesammelt, geordnet und für jede einzelne Startnummer mit Direktzugriffsbefehlen (B-R, B-W) auf Diskette geschrieben. Fast zur selben Zeit werden diese aufbereiteten Zeiten wieder weiter an die Anzeigetafel gesendet, auf der dann die momentan gültige Reihung der fünf führenden Fahrer erscheint.

Diese Aufgabe übernimmt ein zweiter Steuerblock, der dieselben Funktionen besitzt, nur daß dieser ausschließlich mit dem C64 kommuniziert. Dieser Steuerblock besitzt eine serielle Schnittstelle. E. Dolag gab zu, daß bei der Entwicklung der Software anfangs einige Probleme auftraten, Daten auf dem Userport parallel zu empfangen und praktisch zur selben Zeit seriell an die große Tafel weiterzugeben. Schließlich konnte aber auch dieses Problem zufriedenstellend gelöst werden. Überhaupt, die Software. Durch sie werden nicht nur die Daten der Zeituhr übernommen und bearbeitet, sondern an einen Drucker Epson



RX-80 weitergegeben, der auf Tastendruck die Ergebnislisten ausdrückt. Dieser gesamte Vorgang läuft automatisch ab und behindert den Starter im Starthaus bei seinen eigentlichen Aufgaben (Durchsagen, Rückstellen des „Klaviers“ und der Zeituhr) nicht.

Damit aber nicht genug. Auf besonderen Wunsch der Trainer und Fahrer hat E. Dolag etwa ein Jahr später einen zweiten C64 mit Monitor und Plotter BMC B-1500 dort aufgebaut

SOFTWARE IM ROM-EINSTECKMODUL

und eine Softwarediskette dazugeliefert. Die Ergebnisse, die vom ersten C64 auf die Datendiskette geschrieben werden, können nun umgerechnet und grafisch auf dem Bildschirm und/oder Farbplotter dargestellt werden. Es ist daraus leicht ersichtlich, wo Zeiten verloren oder gewonnen wurden, denn die Zwischenzeitstellen bestimmen die grafischen Koordinaten. Soviel zur Zeitmeßanlage, die aus Klavier, Zeitmeßuhr, Startuhren und C64 plus Floppy besteht. Bleibt nur noch ein kurzer Blick auf die etwa 4 Meter hohe Anzeigetafel. Die Buchstaben- oder Zahlen-

STEHT IM BLICKPUNKT: DIE ANZEIGETADEL

felder darauf bestehen aus einer 5x7-Matrix leuchtender Punkte und ist kompatibel zur Commodore-Tastatur. (Sie könnten damit auch

ohne weiteres den Klammeraffen oder den „Pfeil nach oben“ darstellen, was aber im Wettkampftag nie vorkommen wird.)

Zur Software im ROM-Modul ist zu sagen, daß als vorbereitende Arbeiten die Teilnehmerzahl, Startnummer, Titel des Wettkampfes, Datum, Jury-Mitglieder, Delegierte, Rennleiter, Bahnchef, ja sogar der Sanitätsdienst und die Witterung am Wettkampftag eingegeben werden können.

Sollte wirklich einmal eine Fehlfunktion auftreten (Absturz oder Stromausfall), ist selbstverständlich auch eine manuelle Eingabe der Zeiten möglich, hier muß man halt auf die vom Alge-Zeitcomputer auf dem Papierstreifen ausgedruckten Daten zurückgreifen. Etwas mehr Arbeit, aber der kontinuierliche Rennverlauf ist gewährleistet.

Im Vorfeld der Rodel-Europameisterschaften 1988 hielt das Organisationskomitee eine Besprechung ab, bei der auch ein Cheftechniker von Commodore anwesend war.

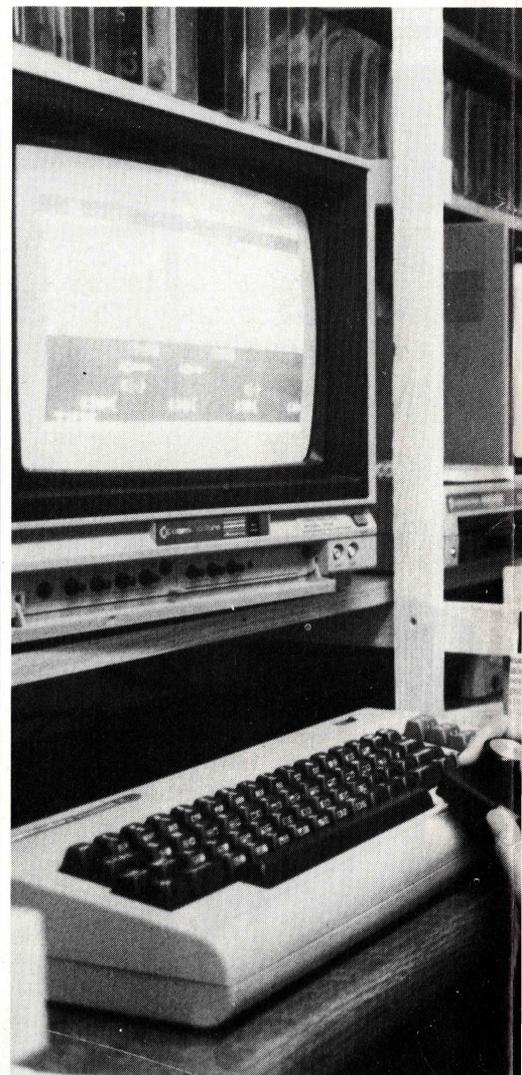
GEPLANT: COMMODORE-ÜWAGEN AUF DEM NEUESTEN STAND DER TECHNIK

Die Aufgabe dieses Mannes und seiner Crew ist die technische Betreuung von Sportereignissen in ganz Europa, bei denen Commodore die Ergebnisauswertung übernommen hat. Von ihm konnten wir auch erfahren, daß Commodore dabei ist, einen Übertragungslastwagen einzurichten, der mit etwa 20 PCs der neuesten Generation (PC 60-80) und diversen Amiga 2000 bestückt sein soll.

Der neueste Stand der Computertechnik steht also auch im Dienste des Sports. Dabei sollten wir nicht unbeachtet lassen: Commodore erhält keinerlei Leasing- oder Leihgebühren, egal, wo dieser Ü-Wagen dann aufgestellt wird. Lediglich eine gewisse, von einer internationalen Kommission vorgeschriebene Zahl von Fernseh-Einblendungen des Firmenemblems ist gestattet, so daß die Kosten für diese mobile Fernsehstation wohl als Werbungskosten abzuschreiben sind.

Inzwischen ist abzusehen, daß die Tage des guten, alten C64 als Zeitenumrechner und -verteiler an der Königseer Bob- und Rodelbahn nach vier, fünf Jahren treuer Dienste doch gezählt sind. Die beiden Acht-Bit-Computer sollen im Laufe des nächsten Jahres von einem Amiga 2000 abgelöst werden. E. Dolag arbeitet bereits an der entsprechenden Software. hb □

Nach Schätzungen wird bereits im Jahr 1990 jeder zweite Arbeitnehmer ohne Computer-Wissen nicht mehr auskommen, an 13 Millionen Arbeitsplätzen wird ein PC-Terminal mit Bildschirm installiert sein. Welche rasante Entwicklung dieser Trend seit 1985 gemacht hat, soll Ihnen unsere Tabelle verdeutlichen. Immer weniger Arbeitnehmer werden künftig zu Block, Kugelschreiber oder gar zur Schreibmaschine greifen, denn der Computer bietet da mit einigen Tastendruckungen gleich mehr Möglichkeiten: Text- oder Datenverarbeitung, Grafik-De-



sign, komplizierte Rechengänge werden schneller und exakter gearbeitet, als es Menschen je könnten.

STIEFKIND DES STUNDENPLANES: EDV-UNTERRICHT

Ist eigentlich jeder Schulabgänger, der sich ins Berufsleben stürzen will, für diese Anforderungen gerüstet? Wohl kaum, denn nicht jeder besitzt einen ernstzunehmenden Computer

**Haben
Sie Fragen ?**

**Nutzen Sie die
COMMODORE
WELT-
HOTLINE**

**Jeden Mittwoch
15 - 19⁰⁰
Tel .: (089)1298013**

BESSERER BERUFSEINSTIEG MIT EDV-WISSEN

Können wird besser bezahlt

Industrie, Verwaltung, Dienstleistungsbetriebe werden in ständig zunehmendem Maße von einem prägnanten Schlagwort beherrscht: Elektronische Datenverarbeitung (EDV). In naher Zukunft wird ohne Computer-Technik nichts mehr laufen. Ein nicht zu unterschätzendes Argument für alt und jung, sich beizeiten mit diesem Medium auseinanderzusetzen.



zu Hause (wir sprechen hier nicht von Homecomputern). Vor allem mangelt es an vielen Schulen recht krass an einem Pflichtfach Informatik – von den fehlenden Geräten (PCs zum Üben und Lernen und Lehrern) gar nicht zu reden. Ohne Eigeninitiative von Schulleitern, Lehrern, Eltern und Schülern, die dafür ihre Freizeit opfern, ginge hier derzeit gar nichts.

Fachleute, die den Trend erkannt haben, fordern eine staatlich reglementierte Bestückung der Schulen mit Computern, und zwar aller Schulen. Wenn heutzutage überhaupt Informatikunterricht im Stundenplan vorgesehen ist, so ist er immer nur auf die Oberstufe der Gymnasien beschränkt. An den Hauptschulen findet er so gut wie gar nicht statt. So erreichte uns unlängst der Bittbrief eines Rektors, der unsere Redaktion um ausgemusterte Commodore-Homecomputer bat. Wir haben ihn mit seiner Bitte an Commodore Deutschland verwiesen.

Finden Sie nicht, daß es schon ein Armutszeugnis für die Kultusmini-

DIE SCHEUKLAPPEN DES KULTUSMINISTERIUMS

sterien oder die dafür zuständigen staatlichen Institutionen darstellt, daß sich dieser Schulleiter mit seiner fortschrittlichen Entscheidung, an seiner Schule Computer zu installieren und Informatik-Unterricht zu geben, nicht vertrauensvoll an seine vorgesetzte Dienststelle, sondern an private Verlage oder an den Hersteller selber wenden muß? Ist die Anschaffung von Turngeräten oder Basketball-Körben soviel wichtiger? Hier hätte unser Rektor in keinem Fall Schwierigkeiten gehabt.

Die Verantwortlichen der Kultusministerien sollten endlich anfangen, die Scheuklappen abzulegen, um den ihnen für zehn bis dreizehn Jahre anvertrauten Schülern alle Voraussetzungen mitzugeben, um ihnen eine Lehrstelle oder einen Beruf zu ermöglichen.

Schon bei Einstellungsgesprächen für Auszubildende wird immer häufiger die Frage gestellt, wie's denn eigentlich mit den Computer-Kenntnissen aussieht. Wer hier passen muß, kann sich selbst ausrechnen, ob er den Ausbildungsplatz bekommt oder ein anderer, der sich in der EDV auskennt. Sicher hat sich die EDV-Ausbildung in den letzten Jahren in verschiedenen Bundesländern verbessert. Doch immer noch mangelt es an spezifischer, also schulgerechter Software sowie an qualifizierten Lehrkräften für den Informatikunterricht. Wie

ZUNAHME DER EDV IM BÜRO

Geräte-Art	Anzahl	
	1985	1987
Bildschirmtext	20.000	40.000
Teletext	13.000	18.000
Telefax	26.000	64.000
Personal-Computer	350.000	900.000

Das Super- Sonderheft für alle Amiga-Freaks

AMIGA AKTIV

AMIGA AKTIV

8

ÖS 124
DM 14,80
SFR 14,80

**DAS
AMIGA
MAGAZIN**

VERGLEICH:
Verarbeitung
Made in
Germany



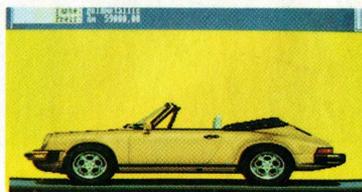
VORGESTELLT:
Preiswerte
Spiele



VOLUMINÖS:
Große
Buchübersicht

VERSCHIEDEN:
Alle Amiga-
Modelle
im Test

VIRTUOS:
Soundpro-
gramme im
Vergleich



COMMODOREWELT SPECIAL 8/87

**Jetzt an Ihrem Kiosk und
im Bahnhofs-Buchhandel
Durchgehend in Farbe!**

sieht es denn in der Praxis aus? Meist vermittelt der Fachlehrer für Mathe und Physik nachmittags in einer Extrastunde interessierten Schülern seine Kenntnisse, die nicht unbedingt aus einem Informatik-Studium stammen. Ein Amateur gibt also sein autodidaktisch erworbenes Wissen an interessierte Amateure weiter. Eine Situation, über die man lachen könnte, wenn sie nicht so gravierend wäre. Erschwerend kommt noch ein menschliches Problem dazu, das es schon seit Inbetriebnahme der ersten Computer vor einigen Jahrzehnten gibt: Die Angst, vom Kollegen Rechner aus dem Job gedrängt zu werden. Mit ein Grund, warum viele Lehrkräfte nichts mit diesem Teufelszeug zu tun haben wollen. Auch deshalb ist Informatik als Pflichtfach an Schulen schwer durchzusetzen.

AUSBILDUNG AM COMPUTER? DAVON STEHT NICHTS IN DER VORSCHRIFT

Die Hoffnung, daß einem am Ausbildungsplatz das Computern beigebracht würde, ist schlichtweg falsch. Heutzutage sind in fast allen Unternehmen mit mehr als 500 Mitarbeitern Computer installiert, etwa 70 Prozent der mittleren und kleinen Betriebe arbeiten mit der EDV. Diese Tatsachen allein garantieren noch keinen Ausbildungs- oder Arbeitsplatz.

In der Praxis sieht's anders aus: Jede Firma wird sich hüten, einen Berufsanfänger oder Auszubildenden an ihre Computeranlage zu lassen: Das ist in der Ausbildungsverordnung nicht für den Lehrbetrieb verpflichtend vorgesehen. Alles weitere beruht auf freiwilliger Basis des Lehrherrn. Es gibt natürlich Ausnahmen, wie im Banken- und Versicherungsgewerbe, in der Hotel- und Reisebüro-Branche und selbstverständlich für den, der den Beruf eines EDV-Kaufmanns ergreifen will.

Die Industrie- und Handelskammern sträuben sich aber, die EDV-Schulung für Auszubildende in den Pflichtkatalog aufzunehmen, denn dies wäre auch für kleinere Betriebe bindend, die keine EDV-Anlage besitzen oder keine brauchen. Die Folge wäre, daß von kleinen Betrieben keine Ausbildungsplätze mehr zur Verfügung gestellt würden, eine Verschärfung der sowieso bereits recht angespannten Situation auf dem bundesdeutschen Ausbil-

dungs- und Arbeitsmarkt. Darum folgen die IHKs dem lobenswerten Beispiel vieler großer Unternehmen,

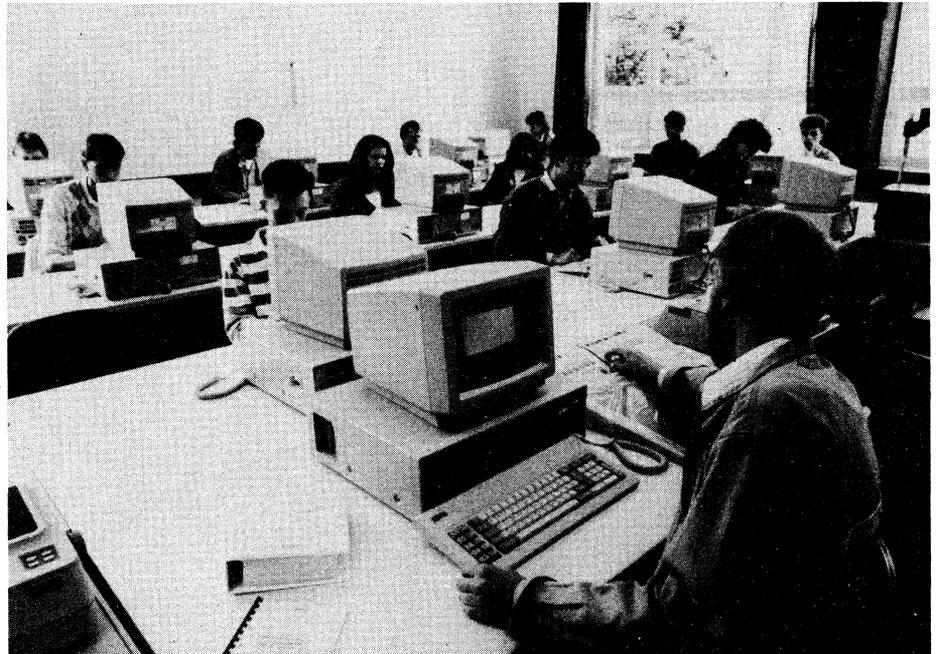
UM EINEN AUSWEG AUS DEM DILEMMA BEMÜHT

EDV-Unterricht für Lehrlinge auf freiwilliger Basis einzurichten, wovon sehr rege Gebrauch gemacht wird. Auf solchen Lehrgängen wird dem Azubi zwar vor allem die Bedienung der Geräte beigebracht, doch bemüht man sich auch, zu vermitteln, wofür das gut ist. Kenntnisse über Software und die Arbeitsweise des Computers werden zusätzlich zur reinen Tastatur-Lehre vermittelt. Informationen über an-

dere Computersprachen (Pascal, Fortran, C usw.), Datenschutz, sowie Beispiele aus der Praxis (Textverarbeitung, Lohnabrechnung etc.) dürfen dabei nicht fehlen.

Im Vergleich zum Wissen, das vermittelt wird, sind die Kosten für so einen Lehrgang ein Klacks: etwa 50,- Mark für 35 Unterrichtsstunden. Weitere Auskünfte erhalten Sie bei der zuständigen Industrie- und Handelskammer.

Vielleicht ein Schritt, den sich ein Jugendlicher überlegen sollte, der den Vorsatz hat, in nächster Zeit als Azubi ins Berufsleben einzusteigen. Home-Computer-Kenntnisse auf dem C16 oder dem C64 werden den meisten Personalchefs nicht ge-



Informatikunterricht an Schulen gibt es bisher nur auf freiwilliger Basis. Die IHKs veranstalten EDV-Kurse.

nügen, wenn er Sie nach Ihrem EDV-Wissen fragt. Professionelle Arbeit mit der EDV hat nämlich recht wenig mit den Fähigkeiten eines Homecomputers gemeinsam . . .
her □

COMMODORE-WELT

Mailbox

24 Stunden online

Parameter 8N1

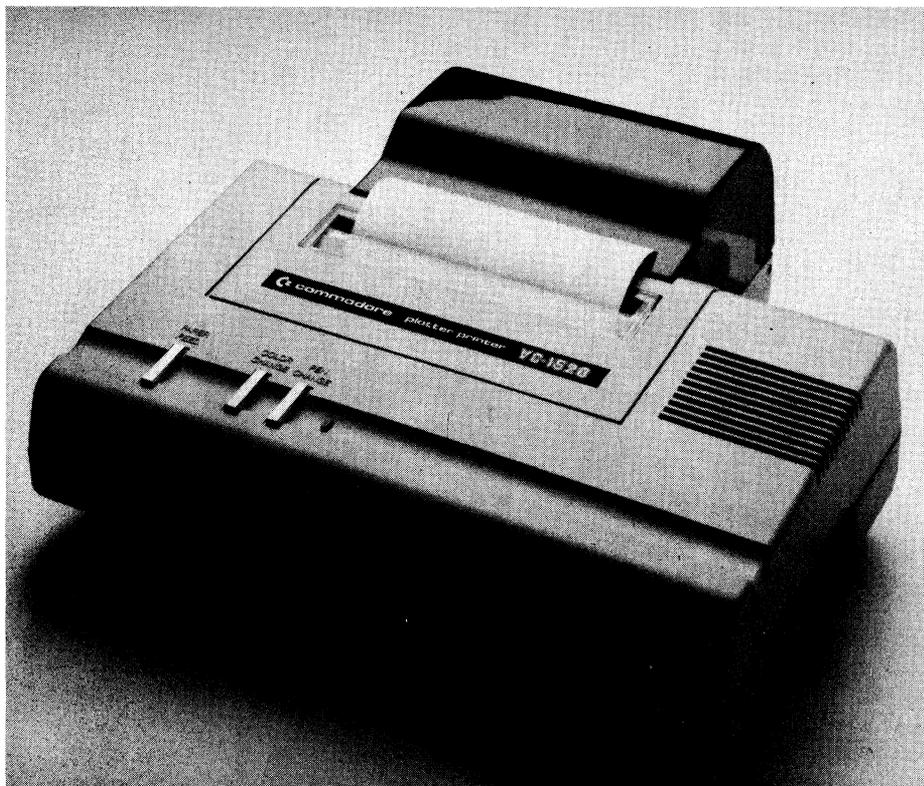
Tel.: (089) 18 39 51

1520 RICHTIG EINSETZEN

Das verkaufte Genie

Durch den immer rascheren Preisverfall der Computer bleibt es nicht aus, daß auch die Preise für die Zubehörgeräte in den Keller stürzten.

Mußte man noch vor Jahresfrist einige Hundertmarkscheine auf den Ladentisch legen, gibt es mittlerweile Geräte, die unterhalb des damaligen Einkaufspreises liegen. Dazu zählt auch der Commodore Plotter 1520. Obwohl er so unscheinbar aussieht, hat er doch einiges im Kasten.



Der Printer Plotter 1520 von Commodore, mit tollen Möglichkeiten.

Außer der Möglichkeit, grafische Darstellungen auf Papier zu bannen, bietet er doch einiges mehr: die 80-Zeichen-Darstellung eines Schriftstückes, das mit einem Textverarbeitungssystem erstellt wurde. Umlaute kann er zwar nicht, aber sonst können sich seine Leistungen sehen lassen.

Um die optimale Ausnutzung der Möglichkeiten des Plotters zu gewährleisten und die Funktionen zu verstehen, ist es wichtig, die allgemeine Arbeitsweise des Gerätes zu kennen.

Dazu ein Blick auf die wichtigsten Daten:

- Vier Kugelschreiberminen (schwarz, blau, grün, rot)
- Vier verschiedene Zeichengrößen (10, 20, 40, 80 Zeichen pro Plotterzeile)

- 13 Schriftzeichen pro Sekunde bei kleinster Schriftart
- 96 Zeichen
- 50 Zeichenschritte pro Zentimeter
- 480 Zeichenschritte in X-Richtung
- ± 999 Zeichenschritte in Y-Richtung
- 264 Zeichenschritte pro Sekunde
- Fünf Zentimeter Zeichenschritte pro Sekunde (X- oder Y-Richtung)
- Sieben Zentimeter Zeichenschritte pro Sekunde bei 45-Grad-Winkel
- 11,4 cm breites Papier.

Bei voller Ausnutzung ist der Plotter schon ein kleines Wunderwerk, wenn man bedenkt, daß auch mehrfarbige Grafik-Hardcopies möglich sind.

Auch die Zeichengeschwindigkeit läßt nicht zu wünschen übrig, wobei man den Preis und die Größe des Gerätes in Relation setzen muß. Für den Homecomputer-Besitzer ist die Leistung allemal ausreichend, obwohl doch die Papierbreite die Anwendungen etwas begrenzt. Genug der Theorie, kommen wir nun zu einigen grundlegenden Anwendungen, die dieses Gerät ermöglicht. Wie eingangs erwähnt, läßt es sich auch mit einer Textverarbeitung kombinieren. Auch als eigenständige Schreibmaschine, der nur die Tastatur fehlt, kann es eingesetzt werden.

Hier stoßen wir auch zum erstenmal auf die, vielen nicht ganz geheuren, Sekundäradressen, die so manchem Computeranwender den Angstschweiß auf die Stirn treiben, gelegentlich aber auch die Zornesröte, weil jeder Druckerhersteller seine eigene Meinung von diesen Adressen besitzt und halt nur die optimalen für seine Erzeugnisse benutzt.

SIEBEN VERSCHIEDENE SEKUNDÄRADRESSEN

Über die Sekundäradressen-Steuerungen des Plotters können die zum Drucker übertragenen Daten gesteuert werden. Auf diese Weise ist ein Drucken, Zeichnen und Ändern der Zeichenfarbe sowie weitere Steuerungen möglich.

- SA=0: Ausdruck von ASCII-Daten
- SA=1: Zeichnen von X-Y-Daten
- SA=3: Farbwahl
- SA=4: Zeichendrehung
- SA=5: Wahl der Linienart
- SA=6: Groß/Grafik-Klein/Groß-Umschaltung
- SA=7: Plotterrückstellung und Speicherlöschung.

Das übertragene Format sieht dann so aus:

```
OPEN lfn,6,0 oder OPEN lfn,6
PRINT #lfn,Daten PRINT #lfn,Daten
CLOSE lfn          CLOSE lfn
```

Wobei lfn für eine laufende Nummer steht, die zwischen eins und 255 betragen kann. Diese Nummer bezeichnet man als File-Nummer. Für die verschiedenen Sekundäradressen kommen noch die zusätzlichen Adressen, die dann die entsprechenden Befehle aufrufen. So werden über die weiteren Befehle die Zeichengröße, Farbe, Linienart, Zeichendrehung und Groß/Kleinschrift ausgewählt.

So betrachtet, verfügt der Plotter 1520 über eine breite Palette von Möglichkeiten, die es nur aus ihm herauszuholen gilt.

(Wird fortgesetzt, Red.)

bw

BASIC-BEFEHLE DES C64 – SELBST GEÄNDERT

Ihr C64 versteht auch Deutsch!

Das Basic 2.0 des C64 bietet trotz des Fehlens einiger komfortabler Anweisungen doch noch genügend Möglichkeiten, um recht ansehnliche Programme in BASIC schreiben zu können.

Den einen oder anderen wird es schon einmal gereizt haben, die Befehle oder sonstigen Anweisungen, die bei der Arbeit mit dem C64 ausgegeben werden, in Deutsch zu übersetzen. (Englisch

Schließlich muß er ja jedesmal in dieser Tabelle nachsehen, ob Ihre Befehlseingabe auch richtig und für ihn verständlich war. Wenn nicht, dann folgt der bekannte SYNTAX ERROR.

Die wichtigsten BASIC-Befehlswoorte, Schlüsselwörter zur Zahlenbehandlung sowie die Fehlermeldungen des BASIC

TABELLARISCH GESPEICHERT: BASIC-SCHLÜSSELWÖRTER

2.0 finden Sie in unserer Tabelle aufgelistet; der entsprechend belegte Speicherplatz wird von den beiden Dezimalzahlen vor dem Befehlswoort dargestellt.

Sie wissen alle, wie Sie den Speicherinhalt einer Adresse verändern können: Mit POKE Adresse, Wert (der die Zahl '255' nie übersteigen darf!).

Nun wäre es ja ganz einfach, aus dem Handbuch

drückt sind das die PEEK-Werte 76 (= L), 73 (= I), 83 (= S) und 212 (= T). Aber halt, in unserem Handbuch steht doch der Wert 84 für T!

Das hat seine Richtigkeit. Wie Sie nämlich aus der Tabelle ersehen, sind die Werte für die BASIC-Befehle im großen und ganzen fortlaufend hintereinander abgelegt. Der letzte Buchstabe so einer Anweisung ist zugleich für den Computer eine Art Ende-Erkennung. Er weiß

WORT-ENDE = ASCII-CODE PLUS '128'

damit, daß er das richtige Wort gefunden hat und



ist ja nicht jedermanns Sache.)

Wie wäre es, wenn Sie beispielsweise Ihrem C64 sagen könnte, LADE statt LOAD, ZEIG statt LIST oder NEU für NEW usw., usw.

Alle Anweisungen und Befehlswoorte, durch die Sie sich mit dem C64 unterhalten, sind an ganz bestimmten Speicherstellen abgespeichert.

Der C64 in seinem schlichten Kleid läßt nicht vermuten, was noch alles in ihm steckt. Der Brotkasten, wie er von Fans liebevoll genannt wird, hat trotz großer Konkurrenz aus dem eigenen Lager noch keine Probleme mit den Verkaufszahlen.

die entsprechenden ASCII-Codes rauszusuchen und sie anstelle der bereits vorgegebenen Zahlenwerte zu POKEn.

Nehmen wir etwa an, Sie möchten das Befehlswoort LIST in ZEIG ändern. Also suchen Sie sich aus der hier abgedruckten Tabelle die Speicheradressen heraus, in denen LIST steht (41229 bis 41232). In ASCII-Codes ausge-

es interpretieren kann. Daher haben sich die Software-Entwickler des C64 eines ganz einfachen Tricks bedient: Der letzte Buchstabe, also das Endezeichen, wird einfach um den Wert 128 erhöht. Das ist übrigens bei allen anderen Homecomputern von Commodore ebenso. Rechnen Sie nach: 84 + 128 ergibt 212. Normalerweise wird mit diesen um 128

REPORT

erhöhten Zeichen die Blockgrafik des C64 dargestellt, die mit der SHIFT-Taste zu erreichen ist. Oder die Großbuchstaben, wenn Sie den Klein/Großschriftmodus eingestellt haben.

Soweit wäre nun eigentlich alles in Butter, wenn ja wenn der C64 mitspielen würde. Aber Sie können die neuen Werte für das Wort ZEIG (90, 69, 73, 199) in den vorgesehenen Speicherplatz POKEN so oft Sie wollen, der Computer reagiert nach wie vor nur auf sein angestammtes Wort, nämlich LIST.

RAM IST NICHT GLEICH ROM

Was ist denn nun wieder los, werden Sie sich fragen. Die Antwort ist einleuchtend: Der Teil des Speichers, in den Sie so gerne hineinschreiben (POKEN) möchten, befindet sich im ROM (englisch Read Only Memory, zu deutsch Nur-Lese-Speicher). Die Bezeichnung sagt es schon: Lesen können Sie daraus, aber nichts hineinschreiben. Genau ausgedrückt: Das BASIC-ROM befindet sich exakt ab Adresse 40960 (\$A000) bis 49151 (\$BFFF). Ist es also „Essig“ mit der ersehnten Umwandlung der BASIC-Befehle?

Mitnichten. Die Erfinder des C64 waren nämlich wahre Meister im Verstecken von Speicherplatz. Den gibt's in der gleichen Größe und Ausdehnung nochmals, unter dem BASIC-ROM versteckt. Und das Allerwichtigste: Das ist ein RAM-Speicher (engl. Random Access Memory, zu deutsch Speicherplatz, der mit beliebigen Daten gefüllt werden kann). Also, haben wir es doch noch gefunden, unser Ei des Kolumbus!

Normalerweise ist dieser Bereich leer. Um aber damit etwas anfangen zu können, müssen wir den

exakten Inhalt des ROM vom BASIC-Interpreter dort hineinkopieren, Byte für Byte. Mit einer einfachen BASIC-Anweisung ist das möglich:

```
FORI = 40960 TO 49151
: POKE I, PEEK(I):NEXT
```

Schlafen Sie bitte nicht ein dabei, es dauert in BASIC eben seine Zeit, bis genau 8192 Byte von einer Baustelle zur anderen geschaufelt sind... Allerdings könnten Sie sich auch die einmalige Mühe machen und das hier abgedruckte Listing 'ROM-COPY' abtippen. Hier wird per Maschinenroutine das Kopieren in knapp einer Sekunde erledigt.

Normalerweise weiß dieser relativ doofe Computer namens C64 jetzt aber immer noch nicht, daß er sich seine Informationen aus dem kopierten RAM-Bereich holen soll (in dem Bereich, in dem Sie jetzt Ihre neuen Befehls Worte eingetragen haben).

DIE SCHALTSTELLE: ADRESSE 1

Aber auch daran haben die Väter des C64 gedacht: Es gibt einen Schalter in Speicherstelle 1. Je nach Wunsch kann das ROM oder RAM eingeschaltet werden, dazu ist nur der Speicherinhalt zu ändern.

Der Normalwert von Adresse 1 ist beim C64 immer 55, also die ROM-Bereiche vom BASIC-Interpreter und KERNAL, eingeschaltet. Beim C64 im 128PC übrigens ist dieser PEEK-Wert in 1 immer um 64 höher. Das hat mit der DIN-Taste zu tun, dadurch wird dem C128 im 64er-Modus mitgeteilt, daß diese spezielle Taste nicht gedrückt ist. Ansonsten reagiert diese Adresse aber genauso wie beim „reinen“ C64.

Um endlich in das RAM umzuschalten, das Sie soeben aus dem ROM kopiert haben, müssen Sie

ROM-COPY . 64

```
9 rem =====<nd>
10 rem rom-copy c 64 <pm>
20 rem (c) commodore welt/h.b. <dp>
30 rem $a000 - $ffff (40960-65535) <gb>
31 rem =====<kh>
40 dimhx(75):fori=0to9:hx(48+i)=i:
hx(65+i)=i+10:next <fj>
50 fori=49152to49230:readmc$:hi=asc(
left$(mc$,1)):lo=asc(right$(mc$,
1)) <md>
60 dz=16*hx(hi)+hx(lo):pokei,dz:ne
xt <b1>
70 sys49152:new <of>
80 data78,a2,00,86,62,a2,bf,86 <ch>
90 data63,a0,ff,a2,37,86,01,b1 <oh>
100 data62,a2,34,86,01,91,62,88 <fd>
110 datac0,ff,d0,ef,c6,63,a5,63 <no>
120 datac9,9f,d0,e5,a2,ff,86,63 <gp>
130 dataa0,ff,a2,37,86,01,b1,62 <ol>
140 dataa2,34,86,01,91,62,88,c0 <hn>
150 dataff,d0,ef,c6,63,a5,63,c9 <lf>
160 datadf,d0,e5,a2,37,86,01,58 <jd>
170 dataa5,01,29,fd,85,01,60 <jm>
```

ROM-COPY . 64

```
10 rem =====<jf>
20 rem zusatzprogramm rom-copy 64 <im>
30 rem aenderung der basic-befehle <na>
40 rem (c) commodore welt/h.b. <ee>
50 rem load = lade <le>
60 rem save = rett <ch>
70 rem list = zeig <pl>
80 rem error = fehler <lh>
90 rem ready. = okay!! <ck>
100 rem =====<id>
110 fori=41193to41200:reada:pokei,
a:next <fp>
120 fori=41229to41232:reada:pokei,
a:next <mh>
130 fori=41834to41839:reada:pokei,
a:next <ib>
140 fori=41848to41853:reada:pokei,
a:next <ca>
150 data76,65,68,197,82,69,84,212,
90,69,73,199 <gf>
160 data70,69,72,76,69,82,79,75,65,
89,33,33 <mn>
```


TIPS & TRICKS

KEY . 64

<pre> 10 rem key 64 11 rem funktionstastenbelegung 12 rem by commodore welt/h.b. 13 dimhx(75):fori=0to9:hx(48+i)=i: hx(65+i)=i+10:next 14 fori=49152to49661:readmc\$:hi=asc c(left\$(mc\$,1)):lo=asc(right\$(mc\$, 1)) 15 dz=16*hx(hi)+hx(lo):pokei,dz:ne xt 16 sys49172:new 17 data a2,0d,a0,00,a9,00,99,6c 18 data c1,ca,f0,08,98,18,69,14 19 data a8,4c,04,c0,78,a9,9d,8d 20 data 14,03,a9,c0,8d,15,03,a9 21 data 2b,8d,08,03,a9,c0,8d,09 22 data 03,58,60,20,73,00,c9,4b 23 data f0,06,20,79,00,4c,e7,a7 24 data 20,73,00,c9,45,f0,03,4c 25 data 32,c0,20,73,00,c9,59,f0 26 data 03,4c,32,c0,20,73,00,20 27 data 9e,b7,e0,00,d0,03,4c,1a 28 data c1,e0,0d,b0,3d,a9,00,ca 29 data f0,06,18,69,14,4c,5f,c0 30 data a8,84,fb,20,fd,ae,20,9e 31 data ad,20,a3,b6,c9,15,b0,22 32 data 85,fc,86,f7,84,f8,a6,fb 33 data a0,00,a5,fc,f0,0c,b1,f7 34 data 9d,6c,c1,e8,c8,c6,fc,4c 35 data 82,c0,a9,00,9d,6c,c1,4c 36 data ae,a7,4c,48,b2,a5,cc,f0 37 data 03,4c,31,ea,a5,cb,c9,04 38 data d0,05,a0,01,4c,cd,c0,c9 39 data 05,d0,05,a0,03,4c,cd,c0 40 data c9,06,d0,05,a0,05,4c,cd 41 data c0,c9,03,d0,05,a0,07,4c 42 data cd,c0,4c,31,ea,ad,8d,02 43 data c9,01,d0,04,c8,4c,e5,c0 44 data c9,02,d0,09,98,38,e9,01 45 data 4a,18,69,09,a8,84,f7,20 46 data 87,ea,a5,cb,c9,40,d0,f7 47 data a6,f7,a9,00,ca,f0,06,18 48 data 69,14,4c,f4,c0,a8,b9,6c 49 data c1,f0,14,c9,2a,f0,07,20 50 data d2,ff,c8,4c,fe,c0,a9,01 51 data 85,c6,a9,0d,8d,77,02,4c 52 data 31,ea,20,79,00,f0,03,4c 53 data 08,af,a9,00,85,f8,a2,01 54 data 86,f7,a9,0d,20,d2,ff,a9 55 data 00,a6,f7,20,cd,bd,a9,3e 56 data 20,d2,ff,a9,22,20,d2,ff 57 data a6,f8,bd,6c,c1,f0,06,20 58 data d2,ff,e8,d0,f5,a9,22,20 59 data d2,ff,a9,0d,20,d2,ff,e6 60 data f7,a5,f7,c9,0d,d0,03,4c 61 data ae,a7,a5,f8,18,69,14,85 62 data f8,4c,2f,c1,4c,4f,41,44 </pre>	<pre> <ff> <om> <kj> <hg> <oj> <cl> <dh> <ep> <mn> <kn> <be> <al> <cl> <ki> <jk> <mg> <ph> <hk> <dc> <co> <jn> <md> <kd> <ph> <ag> <me> <ei> <fk> <dj> <pl> <hd> <go> <cm> <bf> <kf> <fh> <pl> <cm> <im> <nk> <ea> <me> <ff> <df> <lj> <af> <ja> <hm> <ge> <on> <bi> <jj> </pre>	<pre> 63 data 00,00,00,ff,00,ff,00,ff <mh> 64 data 00,ff,00,ff,00,ff,00,ff <nb> 65 data 50,52,49,4e,54,43,48,52 <gp> 66 data 24,28,31,34,37,29,2a,00 <ii> 67 data 00,ff,00,ff,53,41,56,45 <ha> 68 data 00,00,00,ff,00,ff,00,ff <mo> 69 data 00,ff,00,ff,00,ff,00,ff <ei> 70 data 44,41,54,41,00,ff,00,ff <hn> 71 data 00,ff,00,ff,00,ff,00,ff <ed> 72 data 00,ff,00,ff,52,55,4e,2a <gf> 73 data 00,ff,00,ff,00,ff,00,ff <fe> 74 data 00,ff,00,ff,00,ff,00,ff <ml> 75 data 47,4f,54,4f,00,00,00,ff <bi> 76 data 00,ff,00,ff,00,ff,00,ff <jf> 77 data 00,ff,00,ff,4c,49,53,54 <eg> 78 data 2a,00,00,ff,00,ff,00,ff <nc> 79 data 00,ff,00,ff,00,ff,00,ff <dl> 80 data 47,4f,53,55,42,00 <mf> </pre>
---	---	--

oder sofort ausgeführt werden. Das sind vor allen Dingen diese Befehle, die in unserer Übersicht mit dem Zusatz '+ Return' gekennzeichnet sind. Diesen Belegungsvorschlag der verschiedenen F-Tasten brauchen Sie allerdings in keiner Weise einzuhalten, Sie können den Text, der auf einer Taste liegt, jederzeit ändern.

Löschen einer Belegung:
 KEY Nr. d. F-Taste, " " z.B. KEY1, " "

Belegen einer F-Taste:
 KEY Nr. d. F-Taste, "(Text)" z.B. KEY3, "LOAD" mit sofortiger Ausführung (Anhängen von Carriage Return): Dazu ist nötig, den 'Asterix' (das Sternchen), unmittelbar an den Text anzuhängen:
 z.B. KEY5, "LIST*"

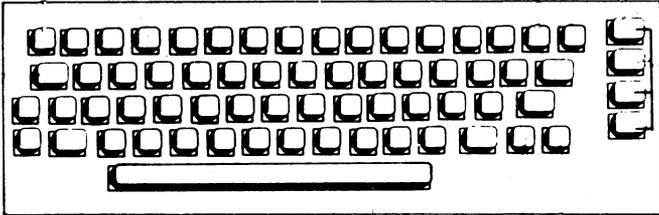
Bitte beachten Sie diese Syntax genau, damit keine Probleme entstehen. Wie Sie die Funktionstasten mit den geraden Ziffern (2, 4, 6, 8) erreichen, ist Ihnen sicher bekannt: Durch gleichzeitiges Drücken der SHIFT-Taste. Dieses Programm sieht noch ein weiteres Bonbon vor. Wenn Sie statt der SHIFT- die COMMODORE-Taste

drücken, können die vier gleichen Funktionstasten nochmals belegt werden, sie werden dann computerintern als F-Taste 9 bis 12 geführt und können von Ihnen auch mit Angabe dieser Zahlen belegt werden.

Mit der Eingabe "KEY0" (und dem Drücken der Taste 'Return') bekommen Sie Ihre Funktionstastenbelegung auf dem Bildschirm angezeigt. Bitte beachten Sie, daß der Text, den die F-Tasten dann ausgeben sollen, nie länger als 18 Zeichen pro Taste sein darf. Die Befehlserweiterung KEY ist nach dem Laden des BASIC-Ladeprogrammes und dem Start mit RUN nach einer kurzen Daten-Einlese-Zeit betriebsbereit, allerdings nicht gegen den NMI-Interrupt geschützt. Der entsteht, wenn Sie die Tastenkombination STOP/RESTORE drücken (oder einen RESET) ausführen. Durch 'SYS 49172' können Sie jedoch dieses Utility-Programm wieder aktivieren. Wenn Sie statt dessen 'SYS 49152' verwenden, so wird die momentan aktuelle Funktionstastenbelegung gelöscht, so daß Sie sie neu definieren müssen. hb □

TIPS UND TRICKS ZU SUPERBASE

Fehler erkannt Fehler gebannt



Superbase ist zweifellos die interessanteste und professionellste Datenbanklösung für die Commodore Computer C64/Plus4 und PC128. Leider scheint sich bei diesem Programm eine alte Weisheit zu bewahrheiten, wonach ein flexibles Programm schwieriger in der Einarbeitung ist, als ein einfaches, auf einen Datentyp festgelegtes Programm. Die meisten Fehler entstehen immer wieder bei der Erstellung der Bildschirmmasken.

Zum ersten ergeben sich Probleme mit der Bezeichnung eckige Klammer. Mit diesen eckigen Klammern sind die Klammern gemeint, die auf den Tasten mit dem Doppelpunkt oder dem Semikolon liegen. Werden andere Klammern benutzt, nimmt das Programm scheinbar die Berechnung an, führt aber, wie sich später herausstellt, die Berechnungen falsch durch. Auch die Meldung ungültige Berechnung kann einen schon zur Verzweiflung bringen, wenn man bei der Eingabe der Feldnamen nicht aufpaßt. Superbase nimmt zwar Feldnamen (Variablen) bis maximal zwölf Zeichen an, aber diese müssen dann auch hundertprozentig stimmen. Wobei zu beachten ist, daß Superbase die Feldnamen von rechts nach links bis zum ersten Blank zählt.

Beispiel:
Gesamt Zahlung
Letzte Zahlung

In diesem Beispiel haben beide Felder den Namen Zahlung. Um Abhilfe zu schaffen, können die Feldnamen logisch abgekürzt werden. Beispiel:
Ges.-Zahlung
Letzte-Zahlg.

Dabei wird die Bedingung der Einzigartigkeit erfüllt, die Superbase für die Feldnamen erwartet. Desweiteren treten Probleme auch bei der Eingabe der Daten auf, wobei nach RETURN Daten in alle anderen Felder kopiert werden. Auch dies hat mit dem Feldnamen zu tun. Meistens wurden dann die Felder in folgender Weise angelegt:

Vorname :
Name :
Straße :
Plz/Ort :

Haben Sie diese oder eine ähnliche Kombination für Ihre Maske gewählt, so werden nach Eingabe von Daten in das Feld „Vorname“ diese Daten sofort in alle anderen

Felder kopiert. Das Leerzeichen zwischen dem Namen und dem Doppelpunkt wird als Trennzeichen interpretiert.

SPEICHERPLATZ- BERECHNUNG

Vielen Superbase-Anwendern ist immer noch unklar, wie die Speicherplatz-Verteilung auf einer Datendiskette zu organisieren ist. Dabei scheiterte an diesem Problem schon so manche feine geplante Datenbank. Errechnen läßt sich der Bedarf auf folgende Weise:

Felder - 1 + Felderlänge
+ 5 = Speicherbedarf

Als kleinste Einheit reserviert Superbase 128 Byte auf der Diskette. Für jedes weitere Bit werden erneut 128 Byte reserviert.

EIGENE PROGRAMME

Wenn Sie sich schon ein wenig mit Superbase auskennen, werden Sie anfangen, eigene Programme zu schreiben. Dabei müssen Sie folgende Regel beachten:

Sie dürfen maximal 79 Zeichen pro Programmzeile eingeben. Da Superbase Befehle auch in gekürzter Form erkennt, diese aber nach erneutem Listen vollständig anzeigt, ist es ratsam, alle Befehle in der ausgeschriebenen Version zu benutzen. Dadurch wird erreicht, daß Sie jede Zeile vollständig auf dem Bildschirm sehen und kein Befehl verlorengeht.

DATEIEN LÖSCHEN

Bei Arbeiten mit Superbase kann es schon einmal vorkommen, daß eine größere Anzahl von Datensätzen gelöscht werden müssen. Man kann sich jeden Datensatz einzeln anzeigen lassen und dann mit delete löschen; bei einer größeren Anzahl von Datensätzen kann das

in Arbeit ausarten. Und wenn wir schon einen Computer benutzen, kann dieser die Arbeit doch viel besser übernehmen. Was benötigt wird, ist ein kleines Programm, welches im Programmiermodus eingegeben werden muß. Gestartet werden die Programme aus den beiden Hauptmenüs mit der Taste F7 oder aber dem Befehl execute. Nach einer Sicherheitsabfrage löscht das Programm alle Datensätze der entsprechenden Datei und zusätzlich auch die Dateidefinition.

```
10select first
20select delete:eof
   goto40
30goto 20
40ask "Datei löschen ?"
   :f$
50if f$ "" then menu
60select delete
70maintain other
80menu
```

GANZEN BEREICH LÖSCHEN

Das zweite Programm löscht Datensätze, die direkt hintereinander folgen. Voraussetzung ist, daß die Schlüsselfelder in einer ununterbrochenen Reihe sortiert sind.

```
10ask "erster Datensatz"
   :an$
20ask "letzter Datensatz"
   :en$
30select an$
40display
   <Schlüsselfeld>
50if "Schlüsselfeld" en
   then menu
60select delete:eof
   menu
70goto 40
```

AUSGEWÄHLTE DATENSÄTZE LÖSCHEN

Das dritte Programm ermöglicht das Löschen von Datensätzen, die nach bestimmten Kriterien sortiert in Schlüssellisten vorliegen.

```
10select from
   "Schlüsselliste"
20eol menu
30select delete
40goto 10
```

bw □

C64-PROGRAMME AUF DEM C16

Es erreichen uns immer wieder Zuschriften von Lesern, die wissen wollen, wie man die C64-Programme auf den C16 umschreiben kann. Um Ihnen diesen Software-Markt zu erschließen, veröffentlichen wir einige Tips, die Ihnen beim Umschreiben helfen.

Es ist freilich nicht möglich, Maschinenprogramme vom C64 auf den C16 zu übertragen. Lediglich bei BASIC-Programmen kann sich der Aufwand lohnen.

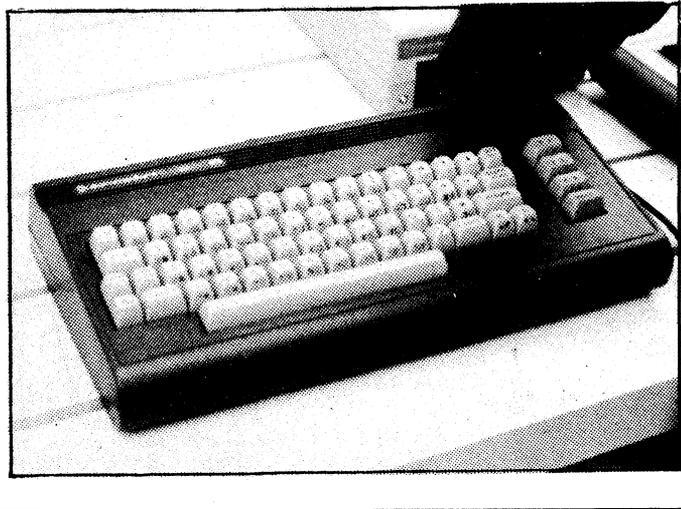
Das V3.5-BASIC des C16/116 ist vollkommen aufwärtskompatibel zum V2.0 des C64, das heißt, daß reine BASIC-Programme, welche nicht mit SYS, PEEK, POKE, USR oder WAIT, also mit Befehlen arbeiten, die auf bestimmte Speicherstellen wirken, vom C64 direkt auf den C16 übertragen werden können. Sie werden von Diskette ganz normal mit DLOAD...

geladen und laufen ohne Probleme, solange der Speicherplatz reicht. Wenn das Programm nur auf Kassette vorliegt, kann es nicht in den C16 eingelesen werden. Hier sollte man eine(n) gute(n) Freund(in) mit Floppy haben, damit das Programm auf dem Umweg über die Diskettenstation auf den C16 übertragen werden kann. (Wer keine solchen Freunde hat, kann sie über unseren Anzeigenservice finden!) Probleme gibt es vor allem bei den zahlreichen Pokes, die beim C64 nötig sind, da es nicht über ein so komfortables BASIC verfügt. Hier wollen wir Ihnen zunächst einmal ein paar der häufigsten Poke-Adressen auflisten, womit schon ein hoher Prozentsatz der Programme konvertiert werden kann:

POKE 53280,F
Hintergrundfarbe setzen;
POKE 53281,F

DIE SEITE 16 FÜR ALLE 16er!

Heute wieder einige Tips zum Umgang mit der Diskettenstation, letzte Meldungen von der 128-KByte-Erweiterung und schließlich die von vielen Lesern gewünschten Tips zur Umwandlung von C64-Programmen auf den C16.



Rahmenfarbe setzen;
POKE 646,F
Zeichenfarbe setzen.

Diese Pokes werden durch die entsprechenden Color-Befehle ersetzt. POKE198,0:WAIT198,1:GET A\$ (wartet auf Tastendruck) wird durch GETKEY A\$ ersetzt. Beschränken Sie sich bei Ihren Bemühungen, C64-Programme auf Ihren C16 oder Plus4 zu übertragen, für den Anfang auf einfache Programme ohne Grafik, veränderten Zeichensatz und Sound. Sind es nur wenige Pokes oder Wait-Befehle, so „verstecken“ Sie diese einmal probeweise durch REM und testen das Programm. Oft sind diese Tricks nur Verschönerung und das Programm läuft trotzdem. Für diejenigen Leser, die auch C64-Programme mit sehr vielen Pokes umwandeln wollen, arbeiten wir gerade an einer Poke-Vergleichstabelle, in der dann alle wichtigen Adressen enthalten sind. □

VORSICHT VOR ER, ST UND DS

Der C16 hat eine Reihe von Möglichkeiten, um Programme absturz sicher zu machen, indem durch den Trap-Befehl eine Fehlermeldung abgefangen und das Programm bei einem Fehler auf eine Fehlererkennungsroutine umgeleitet werden kann. Hier hat man die Möglichkeit, mit Hilfe der Variablen ER die Fehlernummer zu erfragen oder mit PRINT ERR\$(ER) den dazugehörigen Text auszugeben. Bei Fehlern mit der Diskettenstation braucht man die entsprechenden Variablen DS und DS\$, die die Fehlernummer oder den Fehler beinhalten, sowie die Statusvariable ST. Meist fragt man dann im Programm die Fehlernummer ab, gibt die entsprechende Anweisung an den Benutzer aus und testet dann erneut, ob alles in Ordnung ist.

Hierbei wird oft eine sehr wichtige Tatsache übersehen. Wenn kein neuer Fehler auftritt, bleibt die Nummer des alten Fehlers erhalten und die Abfrage wird zu einer Endlosschleife, da das Programm durch die Abfrage des alten Fehlers nicht erkennen kann, ob der Fehler behoben wurde. Hierzu ein Beispiel:

Will man testen, ob die Diskettenstation eingeschaltet wurde, kann man durch Abfrage des Fehlerkanals (zum Beispiel mit A\$=DS\$), nachdem man einen Trap-Befehl auf die Folgezeile gerichtet hat, in der folgenden Zeile abfragen, ob ein Fehler erfolgte. Ist die Diskettenstation nicht eingeschaltet, steht in ER die Nummer 5 (DEVICE NOT PRESENT). Versucht man den Diskettenfehler abzufragen (IF DS=...) so kommt man in Schwierigkeiten, da ja dadurch wieder der Fehlerkanal angesprochen werden muß und das geht ja gerade in diesem Fall nicht. Also bleiben wir bei ER.

Nachdem durch eine IF-Abfrage erkannt wurde, daß die Diskettenstation nicht eingeschaltet ist (IF ER=5 THEN PRINT "EINSCHALTEN!":GETKEY A\$), springt man mit GOTO wieder auf die erste Zeile, in der ein erneuter Versuch unternommen wird. Inzwischen ist die Diskettenstation eingeschaltet, der Zugriff auf den Fehlerkanal erfolgt fehlerlos, aber leider wird ER nicht wieder zurückgesetzt und so ist die Abfrage nach ER wieder positiv. Das Programm befindet sich in einer Schleife. Leider kann man ER nicht wie bei anderen Variablen mit ER=-1 zurücksetzen, da ER eine reservierte Variable ist, die nur vom Interpreter verändert werden kann. Hier bietet sich als Ausweg die Statusvariable ST an. Diese ist -128, wenn die Diskettenstation ausgeschaltet ist. Sie hat

TIPS & TRICKS

FLOPPY-TEST

```

10 rem floppy-test =====16 <nb>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by schmid-fabian <lo>
50 rem heidelberg <kp>
60 rem <ah>
70 rem basic v3.5 <nl>
80 rem c16/116 p4 floppy <ce>
90 rem ===== <jg>
100 fe$=chr$(27)+"t":ho$=chr$(19) <mj>
110 re$=chr$(29)+chr$(29)+chr$(29) <dn>
120 sc$=chr$(19)+chr$(19)+chr$(147)
) <kf>
130 rem ----- <hh>
140 rem einschalttest <ho>
150 rem ----- <jj>
160 trap170:a=ds <le>
170 if st=-128 then print"Diskette
nstation einschalten":getkeya$:got
o160 <aa>
180 rem ----- <mp>
190 rem diskname testen <ep>
200 rem ----- <bf>
210 printsc$:char,5,3,fe$:input"Diskettenname";dn$:iflen(dn$)>16 then
n 210 <lf>
220 if len(dn$)<16 then dn$=dn$+"
":goto 220 <bi>
230 printfe$;:directory"# <ai>
240 f$="":printhe$re$;:open3,3:for
i=1to16:get#3,a$:f$=f$+a$:next:close3
<j>
250 bl$="":print:open3,3:fori=1to3
:get#3,a$:bl$=bl$+a$:next:close3:print:print
<ga>
260 trap 210 <gi>
270 if dn$<>f$ then print"Bitte richtige Diskette einlegen":getkeya$:
:print:goto230 <gg>
280 rem ----- <io>
290 rem filename testen <gi>
300 rem ----- <jl>
310 printsc$:char,5,3,fe$:input"Filename";p$:iflen(p$)>16 then 310 <ac>
320 trap330:a$=ds$:open8,8,8,p$ <hf>
330 if ds>0 thenprint"Floppyfehler
":printds$:getkeyq$:close8:goto310 <fc>
340 close8 <mm>
350 rem =====e=n=d=e===== <of>

```

den entscheidenden Vorteil, daß sie bei jedem Diskettenzugriff auf den entsprechenden Wert ge-

setzt wird und ohne Schwierigkeiten abgefragt werden kann. Vorsicht ist auch bei der Ausgabe des

Fehlers mit PRINT ERR\$(ER) geboten. Hier muß unbedingt eine Abfrage vorausgehen, ob überhaupt ein Fehler vorlag, da sonst ein ILLEGAL QUANTITY ERROR erfolgt, weil ER den Wert -1 hat. Am einfachsten macht man das so: IF ER THEN PRINT ERR\$(ER) □

SICHERHEIT IM UMGANG MIT DER FLOPPYSTATION

Besonders schlimm sind Programmabstürze, wenn sie nach längerer Arbeit beim Abspeichern der Daten erfolgen. Damit Ihnen das in Zukunft nicht mehr passiert, haben wir das Programm Floppy-Test für Sie geschrieben. Es beinhaltet alle möglichen Fehlerquellen.

Zunächst wird natürlich geprüft, ob die Diskettenstation eingeschaltet ist. Dann wird untersucht, ob die richtige Diskette einliegt. Hier kann man die Abfrage nach DN\$ im Programm ersetzen, wenn die Daten immer auf eine bestimmte Diskette geschrieben werden sollen. Wenn dies gleichzeitig ist, kann man den Teil weglassen.

Auch wenn Sie keine Abfrage nach dem Diskettennamen brauchen, sollten Sie sich den hier verwendeten Trick ansehen. Der Diskettenname wird nämlich quasi vom Bildschirm gelesen. Hierzu wird er mit DIRECTORY "#" angezeigt und der Cursor auf den Anfang des Namens positioniert. Dann wird der Bildschirm als Eingabekanal geöffnet (OPEN3,3) und die folgenden 16 Zeilen mit GET#3,A\$ gelesen (Zeile 250).

Zum Schluß wird ab Zeile 310 noch gezeigt, wie man prüfen kann, ob ein bestimmter Filename auf Diskette steht (immer zusammen mit dem Einschalttest). Dazu wird das File zum Schreiben geöffnet und gleich wieder geschlossen. Ist das

File nicht vorhanden, wird der Fehler in Zeile 330 angezeigt.

WAS IST BESSER: HIGH ODER DOUBLE?

Viele Neulinge im Umgang mit der Diskettenstation sind unsicher, welche Diskette sie verwenden sollen. Hier gibt es SS (Single Sided: nur eine Seite), DS (Double Sided: beide Seiten nutzbar), SD (Single Density: einfache Schreibdichte) oder DD (oder auch 2D, Double Density: doppelte Schreibdichte). Am billigsten ist es, Single Sided und Single Density zu kaufen und durch eine zweite Kerbe (symmetrisch zur ersten) auch die zweite Seite zu nutzen (aber nur bei solchen Disketten, die nicht allzu häufig benutzt werden).

Teurer sind die DD-Disketten, da ihre Beschichtung eine höhere Schreibdichte erlauben. Hier gehen die Meinungen über die Verbesserung der Datensicherheit beim C16/P4 auseinander. Wenn DD besser ist als SD, wie gut muß dann erst High-Density sein? Diese Disketten sind für Computer gedacht, die eine noch höhere Datendichte auf der Diskette abspeichern (zum Beispiel IBM-AT). Versuche mit diesen Disketten bei der Floppy 1541/1551 bringen aber meist nur das Ergebnis, daß sich diese Disketten nicht einmal formatieren lassen. Die Beschichtung ist nämlich für kleinere, enger gebündelte Magnetfelder ausgelegt.

Wird mit den normalen Commodore-Diskettenstationen beschrieben, so kommt es zwischen den Spuren zu Übersprechern. Also bleiben Sie bei den preiswerten Disketten. □

100 PLATINEN SUCHEN 100 BASTLER

Wir haben in den vergangenen Ausgaben mehrmals über die 128-KByte-Erweiterung des Herrn

Herrn aus Weinheim berichtet. Wie wir kürzlich erfahren, wurde die Produktion der Erweiterungskarten eingestellt. Der Preis ist offensichtlich so niedrig, daß sich kein vernünftiger Gewinn erwirtschaften läßt und zu einem höheren Preis läßt sich die Erweiterung nicht verkaufen.

Wer nun noch schnell seinen C16 auf 128 KByte aufmotzen will, muß schon selbst zum LötKolben greifen. Es gibt etwa 100 unbestückte Platinen (durchkontaktiert, doppelseitig beschichtet), die zusammen mit dem Bestückungsplan, einer ausführlichen Bauanleitung und dem Programm RAMFLOPPY (siehe CW 12/87) für 28,- Mark per Nachname (oder 30,- Mark Scheck/

bar) abgegeben werden. Also ran an den LötKolben — aber vorsichtig. Die Bauteile, mit denen die Platine bestückt werden muß, sind leicht in Elektronikgeschäften zu beziehen, wobei die RAM-Bausteine den größten Teil des Preises ausmachen. Aber verwenden Sie bitte keinen „RAMsch“, sonst gibt es mehr Ärger als Freude. Achten Sie vor allem darauf, daß nur die in der Bauanleitung beschriebenen wirklich schnelle RAMs sind. Auch sollten Sie eine gute Lötstation und etwas Erfahrung im Löten besitzen. □

Bezugsadresse für die 128-KByte-Platinen:

*Wolf Dieter Herr
Bischofsgasse 9
6940 Weinheim*

MIKROCOMPUTER-GRUNDLAGEN

Bescheid wissen ■ besser programmieren

Wer komfortabel in Maschinensprache programmieren will, benötigt hierzu einen Assembler. Sich selbst einen solchen in BASIC zu schreiben, ist nicht einmal besonders schwer. Neben der Arbeitsweise des Assemblers lernen Sie die CPU-Befehle zur Realisierung einer Programmlogik kennen.

Anstatt einen Maschinencode-Interpreter in BASIC zu programmieren, machen wir einen großen Sprung nach vorwärts aus folgenden Überlegungen: Ein vollständiger Interpreter müßte jeden CPU-Befehl interpretieren können. Ein eigenes Unterprogramm für jeden Befehl würde das Programm ziemlich lang machen. Wir kennen auch bereits die quälend langsame Geschwindigkeit der Abarbeitung. Bei einer langen Liste von Befehlen würde sich die Geschwindigkeit noch weiter vermindern. Die Lösung ist: Wir schrei-

ben uns selbst ein Assembler. Unser nicht zu interpretierendes, sondern assembliertes Programm besitzt die größtmögliche Ablaufgeschwindigkeit. Außerdem ist ein Assembler relativ einfach zu

DIE ARBEITSWEISE UNSERES ASSEMBLERS

realisieren, da die Befehle leicht in einigen Befehlsgruppen zusammengefaßt werden können. Wir brauchen also nur ein paar Listen von Befehlen anzufertigen und zu jeder Liste eine Tabelle, die die

richtigen Befehlscodes enthält.

In Datazeilen steht unser Quellprogramm. Es enthält, durch Komma getrennt, wie es der Natur von Daten entspricht, Assembleranweisungen, CPU-Mnemonics, Adressierungskennzeichen und Daten in Form von Zahlen oder als Text. Liest das Assemblerprogramm nun einen Befehl, so wird das gelesene Wort mit den Elementen der verschiedenen Listen verglichen. Als Ergebnis dieses Vergleichs erhalten wir die Nummer der Liste und des gefundenen Listenelementes.

i = Nummer der Liste
j = Nummer des Elementes.

Mit diesem Wissen ausgerüstet, fällt es nicht schwer, in das dazugehörige Unterprogramm zu verzweigen, und dort entweder den Operationscode zu besorgen oder aber eine weitere Verzweigung vorzunehmen.

BEFEHLSGRUPPEN

Wir unterscheiden zwei Gruppen von Befehlen:

1. Assembleranweisungen
2. CPU-Befehle.

Assembleranweisungen machen ein komfortables Arbeiten möglich. Anstatt per Hand alle möglichen Adressen auszurechnen, setzen wir nur entsprechende Marken. Die eigentliche Arbeit nimmt uns dann der Assembler ab. Daß wir ohne CPU-Befehle nichts zu erreichen vermögen, versteht sich wohl von selbst. Uns erschien folgende Unterteilung der CPU-Befehle recht vielversprechend:

1. Implizite Adressierung
2. Relative Sprünge
3. Absolute Sprünge
4. Mehrfache Adressierung.

IMPLIZITE ADRESSIERUNG

Das ist die einfachste Befehlsart. Dem Operationscode, kurz Opcode, fol-

gen keinerlei Parameter. Es braucht also nur der ausfindig gemachte Code an die durch den Schreibzeiger bestimmte Stelle in den Hauptspeicher gepoke, und anschließend der Zeiger um eins erhöht werden. Einige Beispiele:

- INX erhöht den Inhalt des X-Registers
- DEX erniedrigt ihn
- INY erhöht den Inhalt des Y-Registers
- DEY erniedrigt ihn
- RTS entspricht einem RETURN in BASIC
- SEI unterbindet den Systeminterrupt
- CLI läßt ihn wieder zu
- SEC setzt das Carry-Flag
- CLC löscht es
- SEB (Set Borrow) ist kein extra CPU-Befehl, sondern entspricht einem CLC. Wir definierten ihn, um den Übergang auf andere CPUs zu erleichtern, bei denen im Unterschied zur 6502 oft ein SEB einem SEC entspricht.
- CLB (Clear Borrow) entspricht bei der 6502 einem SEC.

RELATIVE SPRÜNGE

Ein wichtiges Element der BASIC-Programmierung bilden die IF-Anweisungen, die bereits ausreichend wären, um eine Programmlogik zu realisieren. Auf FOR-NEXT-Schleifen könnte eigentlich verzichtet werden, wenn diese nicht schneller als IF-Anweisungen in Verbindung mit GOTO-Sprüngen abgearbeitet würden. Das Commodore-BASIC muß bei Sprüngen nämlich vom Programmanfang an alle Zeilen durchsuchen, bis es die gewünschte Zeilennummer findet. Bei Sprüngen in Maschinencode gibt es diese Problematik nicht. Der Sprung kann direkt auf die gewünschte Programmstelle erfolgen. Wir unterscheiden absolute und relative Sprünge. Bei absoluten Sprüngen folgt dem Op-

TIPS & TRICKS

code die Sprungadresse. Da zwei Byte zur Adressenbildung nötig sind, und diese auch von der CPU gelesen werden müssen, dauert ein solcher Sprung länger, als wenn dem Opcode nur ein Byte folgen würde. Ein weiterer Nachteil des absoluten Sprunges ist, daß Programme mit absoluten Sprungadressen nicht ohne entsprechende Überarbeitung im Hauptspeicher verschoben werden können. Lag nämlich vorher eine Sprungadresse zum Beispiel bei \$2000 und das Programm wird um \$1000 nach oben verschoben, so muß die Sprungadresse nun auf \$3000 abgeändert werden. Relative Sprünge umgehen dieses Problem. Dem Opcode folgt nicht eine absolute Adresse, sondern die Sprungweite. Die Anweisung „Springe 100 Byte vorwärts“ besitzt auch noch nach einer eventuellen Verschiebung des Programmes ihre Gültigkeit. Wenn die Sprungweite auf einen Bereich von 256 Byte beschränkt ist, so genügt bereits ein einziges Byte zur Angabe der Sprungweite, was sich vorteilhaft auf die Abarbeitung auswirkt. Damit nicht nur Vorwärts-, sondern auch Rückwärtssprünge erfolgen können, wurde folgende Vereinbarung getroffen: Zahlen von 0 bis 127 seien positive Zahlen und bewirken einen Vorwärtssprung. Zahlen von 128 bis 255 seien negative Zahlen und bewirken einen Rückwärtssprung. Da $255+1=256$, was der Computer als Null ansieht, entspricht die 255 einer -1. Analog gilt: 128 entspricht -128. Für relative Sprünge muß unser Assembler überprüfen, ob die Sprungweite in dem Bereich von -128 bis +127 liegt und bei negativem Vorzeichen durch die Subtraktion von der Zahl 256 den einzusetzenden Wert berech-

nen. Nach dem Schreiben des Opcodes und der Sprungweite muß der Schreibzeiger um den Wert zwei erhöht werden.

DAS FLAGREGISTER

Das Flagregister vereinfacht die Syntax bei bedingten Sprüngen. Je nachdem, wie das Ergebnis bei einer Operation ausfällt, werden bestimmte Bits des Flagregisters gesetzt. Der Zustand der jeweiligen Flags entscheidet darüber, ob ein bedingter Sprung nun ausgeführt wird oder nicht. Hier die Flags, die für unsere bedingten Sprünge interessant sind:

Zero-Flag (Null-Flag)
Carry-Flag (Übertrags-Flag)
Signum-Flag (Vorzeichen-Flag)
Overflow-Flag (Überlauf-Flag)

Zero-Flag: Ist das Ergebnis einer Operation Null, wird die Zero-Flag gesetzt, ist es nicht Null, wird es gelöscht.

Carry-Flag: Bei Schiebepfeilen ist das nach links oder recht herausgeschobene Bit nicht so gleich verloren, sondern setzt, wenn es sich um eine eins handelt, die Carry-Flag, bzw. löscht sie. Die Carry-Flag wird des weiteren durch Additions-, Subtraktions- und Vergleichsbefehle beeinflusst. Ist das Ergebnis einer Addition größer als 255, so wird der dabei entstandene Übertrag durch das Setzen der Carry-Flag kenntlich gemacht. Ist bei einer Subtraktion oder einem Vergleich das Ergebnis kleiner als Null, so wird der Übertrag ebenfalls durch ein entsprechendes Löschen oder Setzen der Carry-Flag erfaßt. Ob das Carry nun gesetzt wird oder nicht, ist von CPU zu CPU verschieden. Die 6502 und ihre Abkömmlinge vermerken den durch Subtraktion oder Vergleich entstandenen Übertrag (Borrow) durch das Löschen der

Carry-Flag. Hier entspricht also ein gesetztes Borrow einem gelöschten Carry und umgekehrt. Bei anderen CPUs wie der Z80 oder 8086 entsprechen Carry und Borrow einander. Um das Umschreiben von Programmen auf eine andere CPU zu vereinfachen und Verwechslungen zu vermeiden, haben wir unserem Assembler spezielle Befehle für eine fiktive Borrow-Flag mitgegeben, welche natürlich nichts anderes als unsere inverse Carry-Flag ist.

Signum-Flag: Die Signum- und die Overflow-Flag sind eigentlich nicht unbedingt nötig, erleichtern aber mitunter das Rechnen mit vorzeichenbehafteten Zahlen. Ist das Ergebnis einer Operation größer als 127, so wird die Signum-Flag als Kennzeichen einer negativen Zahl gesetzt.

Overflow-Flag: Die Signum-Flag allein ist zur Kennzeichnung einer vorzeichenbehafteten Zahl nicht ganz ausreichend. Entsteht durch Addition zweier positiver Zahlen eine Zahl größer als 127, so dürfen wir wohl kaum von einer negativen Zahl sprechen. Eine gesetzte Overflow-Flag sagt uns, daß die Signum-Flag den Sachverhalt nicht richtig wiedergibt. Dasselbe geschieht auch bei einer Subtraktion, wenn das Ergebnis kleiner als -128 sein sollte, was bedeutet, daß eine Zahl von 0 bis 127 dabei herauskommt. Die Signum-Flag würde uns vorgaukeln, es handle sich um eine positive Zahl. Die Overflow-Flag sagt uns, daß das Gegenteil der Fall ist.

BEDINGTE SPRÜNGE

Da vier Flags existieren, die jeweils zwei unterschiedliche Werte annehmen können, benötigen wir acht bedingte Sprungbefehle. Die sind:

BEQ (Branch Equal: Zero=1)

BNE (Branch Not Equal: Zero=0)
BCS (Branch Carry Set: Carry=1)
BCC (Branch Carry Clear: Carry=0)
BMI (Branch Minus: Signum=1)
BPL (Branch Plus: Signum=0)
BVS (Branch Overflow Set: Overflow=1)
BVC (Branch Overflow Clear: Overflow=0).

Aus bereits erwähnten Gründen führen wir zusätzlich ein:

BBS (Branch Borrow Set: = BCC)
BBC (Branch Borrow Clear: = BCS).

Bei der 6502 und ihren Abkömmlingen sind die bedingten Sprünge relative Sprünge, wobei das dem Opcode folgende Byte die Sprungweite angibt. Soll ein bedingter Sprung auf eine Adresse außerhalb des möglichen Sprungbereichs von 127 Byte vorwärts und 128 Byte rückwärts erfolgen, so ist dieses nur durch eine Kombination eines bedingten Sprunges mit einem unbedingten absoluten Sprung möglich. Bevor wir hierzu übergehen, wollen wir uns anhand eines Beispiels den Unterschied zu BASIC verdeutlichen.

Laufschleife in BASIC:
 $30 x=x-1$
 $40 \text{ if } x < > 0 \text{ then } \dots$

Laufschleife in Assembler:
dex
bne ...

Jede Zeile benötigt in BASIC bereits fünf Byte. $X=X-1$ sind bereits weitere fünf Byte. Sollte der Sprung auf eine zweistellige Zeilennummer erfolgen, so benötigen wir für die IF-Anweisung noch einmal acht Byte. Zusammen wären dieses bereits 23 Byte, wogegen die Assemblerschleife mit 3 Byte auskommt. Ermöglicht wird dieses durch einen speziellen Decrement-Befehl und dem Flag-Konzept. Viel wird dadurch gespart, daß nicht, wie in

TIPS & TRICKS

BASIC, eine speicherplatzfressende Zeilenstruktur benötigt wird.

ABSOLUTE SPRÜNGE

Die unbedingten Sprünge unserer CPU sind absolute Sprünge. Uns stehen drei unterschiedliche Sprungbefehle zur Verfügung:

JMP Jump: direkter Sprung, entspricht in BASIC einem GOTO

JSR Jump Sub-Routine: Routinenaufruf, entspricht in BASIC einem GOSUB

@JMP Indirekter Sprung: mit ihm lassen sich Sprunglisten realisieren, wie sie uns bereits von BASIC her durch die Befehlsfolge:
ON N GOTO 100,
200,300
bekannt sind.

Dem Opcode für den Sprung folgen zwei Adreß-Byte, bei der CPU 6502 zuerst das LOW-Byte, dann das HIGH-Byte. LOW- und HIGH-Byte LB und HB errechnen sich folgendermaßen aus der Adresse AX:

HB=INT(AX/256):
LB=AX-256*HB

Diese Werte poken wir hinter den Opcode des Sprungbefehles und erhöhen den Schreibzeiger um den Wert drei. Problematisch allerdings ist bei Sprungbefehlen die Bestimmung der Sprungadressen, wenn wir uns nichts besonderes einfallen lassen. Denn das Programm von vorne nach hinten Byte für Byte nach Sprungadressen durchzählen, bereitet recht viel Mühe und führt bald zur Frustration. Wenn wir zuletzt feststellen, daß wir in unserem Assemblerprogramm noch irgendeinen Befehl vergessen haben, dürfen wir noch einmal von vorne mit der Bestimmung der Sprungadressen beginnen, weil diese sich nun fast alle verschieben, eine wahre Sysiphus-

arbeit, die diejenigen sicher kennen, die schon Maschinenroutinen mit dem eingebauten Maschinenmonitor des C16/116/Plus4 oder des C128 entwickelt haben. Diese Arbeit jedoch können wir uns ersparen.

SPRUNGADRESSIERUNG DURCH LABEL

LABEL heißt soviel wie Etikett. Beim Programmieren verstehen wir darunter das Anbringen von Sprungmarken. Die meisten Assembler erlauben uns, irgendeiner Programmstelle, die wir anspringen wollen, einen Namen zu geben. Für unsere Zwecke mag es vorläufig genügen, wenn wir die Sprungmarken lediglich numerieren. Ein kleines Beispiel:

```

LDX,#,3
LAB,0, ASL,A
DEX
BNE,0
    
```

Mit LAB,0 veranlassen wir den Assembler, die Programmadresse, an welcher der Shift-Befehl steht, in seiner LABEL-Liste zu vermerken. Der Assembler kann daraufhin beim bald folgenden BRANCH-Befehl die Sprungweite zum LABEL Nr.0 berechnen und den entsprechenden Wert einsetzen. Als Resultat wird der Akku-Inhalt um drei Stellen nach links geSHIFTet, was einer Multiplikation mit acht entspricht.

Problematisch ist folgendes Beispiel:

```

Hauptprogramm
LDX,#,3
JSR,0
...
    
```

```

Unterprogramm
LAB,0, JMP,2
LAB,1, ASL,A
LAB,2, DEX
BPL,1
RTS
    
```

Diese beiden Programmteile haben dasselbe Ergebnis wie unser erstes Beispiel. Durch den Aufruf des Unterprogrammes lassen sich, sofern mehrfach ein Verschieben im

Lesen Sie bitte weiter a. S. 23

```

10 rem assembler=====cbm <k1>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by alfons mittelmeyer <cg>
50 rem <pd>
60 rem <ah>
70 rem basic v2.0 <nc>
80 rem c64/c16/c116/plus4/c128 <of>
90 rem ===== <jg>
100 rem ----- <op>
110 rem zu assemblieren <cp>
120 rem ----- <po>
130 data =adr,2,3072:rem screen (c
64/128 =adr,2,1024) <nj>
140 data org,14336 <og>
150 data sei <kg>
160 data lab,3, jsr,0 <hc>
170 data inc,$,1 <bh>
180 data bne,3 <me>
190 data cli <oo>
200 data rts <od>
210 data lab,0, ldx,#,610,2 <pj>
220 data stx,$,61,0 <gg>
230 data lda,#,6h0,2 <dl>
240 data sta,$,62,0 <pp>
250 data lda,$,1 <cg>
260 data ldy,#,4 <ei>
270 data ldx,#,0 <fg>
280 data lab,1,adr,0, sta,$x,0 <dh>
290 data inx <gg>
300 data bne,1 <oi>
310 data dey <ia>
320 data beq,2 <lk>
330 data inc,$,62,0 <lk>
340 data jmp,1 <hn>
350 data lab,2, rts <co>
360 data adr,1,db,0 <ol>
370 data end <dm>
380 rem ----- <kn>
390 rem e n d e <bk>
400 rem ----- <fc>
1000 rem ===== <np>
1010 rem assemblerprogramm <ko>
1020 rem ===== <ge>
1030 poke55,0:poke56,56:clr:goto10
80 <eb>
1040 hb=int(ax/256):lb=ax-256*hb:r
eturn <bd>
1050 rem ----- <im>
1060 rem restore programdatas <jp>
1070 rem ----- <an>
1080 fori=1to10000 <n1>
1090 readx$:ifx$<>"end"thennext <cm>
1100 i=10000:next <fa>
1110 rem ----- <kh>
1120 rem befehlssatz einlesen <jn>
1130 rem ----- <aa>
1140 dimbf$(100) <hj>
    
```

TIPS & TRICKS

1150 j=0:for n=1to5:for i=0to100	<fa>	1670 rem -----	<jf>
1160 readbf\$(j)	<fe>	1680 onjgoto1690,1700,1710,1720,17	
1170 ifbf\$(j)<>"... "thenj=j+1:next		30,1740,1750,1780,1810,1820,1790,1	
1180 az(n)=i:i=100:next:nextn	<gk>	830,1850	<kj>
1190 fori=1to10:readac\$(i):next	<mc>	1690 readsz:nextpc:rem org	<hf>
1200 dimmn\$(az(5)):bj=j+1	<ip>	1700 readx:lb(x)=sz:nextpc:rem lab	<fo>
1210 forn=1to3:x\$=""	<hl>	1710 readx:readax:lb(x)=ax:nextpc:	
1220 fori=1toaz(n):readx	<pa>	rem =lab	<dm>
1230 x\$=x\$+chr\$(x):next	<jp>	1720 readx:readax:ks(x)=ax:nextpc:	
1240 op\$(n)=x\$:nextn	<bh>	rem =konst	
1250 forn=1toaz(5):x\$=""	<pf>	1730 readx:ad(x)=sz:nextpc:rem adr	<mp>
1260 fori=1to10:reada\$	<af>	1740 readx:readax:ad(x)=ax:nextpc:	
1270 x\$=x\$+chr\$(val(a\$)):next	<ig>	rem =adr	<ei>
1280 mn\$(n)=x\$:nextn	<gc>	1750 readx\$:l=len(x\$):ifpathen1770	
1290 rem -----	<ii>	:rem dt	<kh>
1300 rem assemblerprogramm	<oe>	1760 fori=0tol-1:pokesz+i,asc(mid\$(
1310 rem -----	<hi>	(x\$,i,1)):next	<gg>
1320 forpa=-1to0	<jg>	1770 sz=sz+1:nextpc	<bm>
1330 print"pass "pa+2:print	<be>	1780 readx:pokesz,x:sz=sz+1:nextpc	
1340 restore:forpc=0to2000	<bl>	:rem db	<fc>
1350 readx\$:printx\$	<jn>	1790 pc=2000:nextpc:nextpa:rem end	<lp>
1360 forn=0tobj	<ji>	1800 print"fertig":end	<om>
1370 ifx\$<>bf\$(n)thennext	<bo>	1810 readax:gosub1040:pokesz,lb:po	
1380 ifn=bjthenprint"unknown comma		kesz+1,hb:sz=sz+1:nextpc	<nn>
nd":end	<gf>	1820 printsz:nextpc	<jn>
1390 j=n:n=bj:next:az=0:forn=1to6	<ih>	1830 readl:ifpathen1770:rem rb	<ih>
1400 ax=az:az=az+az(n):ifj>=azthen		1840 fori=0tol-1:pokesz+i,0:next:g	
next	<mi>	oto1770	<bg>
1410 j=j+1-ax:i=n:n=6:next	<jb>	1850 readl:goto1770:rem rs	<cj>
1420 onigoto1460,1510,1600,1680,18		1860 rem -----	<ig>
90	<fa>	1870 rem mehrfache adressierung	<bp>
1430 rem -----	<nj>	1880 rem -----	<kl>
1440 rem implizite adressierung	<oe>	1890 readx\$:x=0:forn=1to10	<cg>
1450 rem -----	<pd>	1900 ifac\$(n)=x\$thenx=n:n=10	<gi>
1460 pokesz,asc(mid\$(op\$(i),j,1))	<od>	1910 next:ifx=0then2160	<gl>
1470 sz=sz+1:nextpc	<ne>	1920 a=asc(mid\$(mn\$(j),x,1))	<aj>
1480 rem -----	<ho>	1930 ifa=0then2160	<ep>
1490 rem bedingte spruenge	<ko>	1940 pokesz,a:readx\$:xl\$=left\$(x\$,	
1500 rem -----	<nb>	1):a=val(x\$):l=len(x\$)	<go>
1510 sz=sz+2:readx:ifpathennextpc	<jm>	1950 onxgoto1960,2040,2100,2150,21	
1520 x=lb(x)-sz	<gm>	00,2100,2100,2040,2040,2100	<hm>
1530 ifx<-128orx>127thenprint"labe		1960 ifxl\$="k"orxl\$="G"thenreada	<ah>
l out of range":end	<bh>	1970 ifpathen2030	<mk>
1540 ifx<0thenx=x+256	<nb>	1980 ifxl\$="k"thena=ks(a)	<in>
1550 pokesz-2,asc(mid\$(op\$(i),j,1)		1990 ifxl\$<>"G"then2020	<jd>
)	<kj>	2000 ax=ad(a)+val(right\$(x\$,1-2))	<ig>
1560 pokesz-1,x:nextpc	<lj>	2010 gosub1040:a=lb:ifleft\$(x\$,2)=	
1570 rem -----	<no>	"Gh"thena=hb	<aj>
1580 rem unbedingte spruenge	<df>	2020 pokesz+1,a	<dg>
1590 rem -----	<fh>	2030 sz=sz+2:nextpc	<co>
1600 sz=sz+3:readx:ifpathennextpc	<jh>	2040 ifxl\$="G"thenreada	<ej>
1610 pokesz-3,asc(mid\$(op\$(i),j,1)		2050 ifpathen2090	<gd>
)	<dk>	2060 ax=ad(a)	<eb>
1620 ax=lb(x):ifj=3thenax=ad(x)	<md>	2070 ifxl\$="G"thena=ad(a)+val(righ	
1630 gosub1040:pokesz-2,lb	<cj>	t\$(x\$,1-1)):ax=a	<cm>
1640 pokesz-1,hb:nextpc	<na>	2080 gosub1040:pokesz+1,lb:pokesz+	
1650 rem -----	<oi>	2,hb	<gj>
1660 rem assembleranweisungen	<ai>	2090 sz=sz+3:nextpc	<ih>

TIPS & TRICKS

2100 ifx1\$="G"thenreada	<ci>	2660 rem unbedingte spruenge	<ap>
2110 ifpathen2030	<pg>	2670 rem -----	<oi>
2120 a=ad(a):ifx1\$<>"G"then2020	<fk>	2680 data 076,032,108	<nd>
2130 a=a+val(right\$(x\$,l-1)):goto2020	<kp>	2690 rem -----	<jf>
2140 a=ad(a):goto1970	<fp>	2700 rem mehrfache adressierung	<kj>
2150 sz=sz+1:nextpc	<ij>	2710 rem -----	<mo>
2160 print"nicht implementiert":end	<nm>	2720 data 105,109,101,---,097: adc	<bf>
2170 rem -----	<cf>	2730 data 113,117,125,121,---	<ea>
2180 rem implizite adressierung	<ko>	2740 data 041,045,037,---,033: and	<dl>
2190 rem -----	<dg>	2750 data 049,053,061,057,---	<hb>
2200 data brk,clc,cld,cli,clv,dex	<ie>	2760 data ---,014,006,010,---	<asl>
2210 data dey,inx,iny,nop,pha,php	<ai>	2770 data ---,022,030,---,---	
2220 data pla,plp,rti,rts,sec,sed	<gl>	2780 data ---,044,036,---,---	<bit>
2230 data sei,tax,tay,tsx,txa,txs	<ml>	2790 data ---,---,---,---,---	<bp>
2240 data tya,seb,clb,...	<mk>	2800 data 201,205,197,---,001: cmp	<km>
2250 rem -----	<ib>	2810 data 209,213,221,217,---	<kp>
2260 rem bedingte spruenge	<lm>	2820 data 224,236,228,---,---	<ie>
2270 rem -----	<pc>	2830 data ---,---,---,---,---	<ff>
2280 data beq,bne,bbs,bbc,bcs,bcc	<lh>	2840 data 192,204,196,---,---	<cpy>
2290 data bmi,bpl,bvs,bvc,...	<cb>	2850 data ---,---,---,---,---	<ni>
2300 rem -----	<oc>	2860 data ---,206,198,---,---	<dec>
2310 rem unbedingte spruenge	<di>	2870 data ---,214,222,---,---	<bd>
2320 rem -----	<la>	2880 data 073,077,069,---,065: eor	<ab>
2330 data jmp,jsr,@jmp,...	<dn>	2890 data 081,085,093,089,---	<gh>
2340 rem -----	<dh>	2900 data ---,238,230,---,---	<inc>
2350 rem assembleranweisungen	<mh>	2910 data ---,246,254,---,---	<oc>
2360 rem -----	<eb>	2920 data 169,173,165,---,161: lda	<kl>
2370 data org,lab,=lab,=konst,adr	<ba>	2930 data 177,181,189,185,---	<oa>
2380 data =adr,dt,db,dw,?adr,end	<no>	2940 data 162,174,166,---,---	<ldx>
2390 data rb,rs,...	<ne>	2950 data ---,---,---,190,182	<gl>
2400 rem -----	<jm>	2960 data 160,172,164,---,---	<ldy>
2410 rem mehrfache adressierung	<hk>	2970 data ---,180,188,---,---	<be>
2420 rem -----	<ea>	2980 data ---,078,070,074,---	<lsr>
2430 data adc,and,asl,bit,cmp,cpx	<kg>	2990 data ---,086,094,---,---	<di>
2440 data cpy,dec,eor,inc,lda,ldx	<fd>	3000 data 009,013,005,---,001: ora	<ld>
2450 data ldy,lsr,ora,rol,ror,sbc	<nm>	3010 data 017,021,029,025,---	<mn>
2460 data sta,stx,sty,...	<mh>	3020 data ---,046,038,042,---	<rol>
2470 rem -----	<pd>	3030 data ---,054,062,---,---	<bn>
2480 rem adressierungsarten	<in>	3040 data ---,110,102,106,---	<ror>
2490 rem -----	<nf>	3050 data ---,118,126,---,---	<fl>
2500 data #,\$,%a,@x		3060 data 233,237,229,---,225: sbc	<ig>
2510 data @y,%x,\$x,\$y,%y	<if>	3070 data 241,245,253,249,---	<ki>
2520 rem -----	<nb>	3080 data ---,141,133,---,129: sta	<bb>
2530 rem implizierte adressierung	<ag>	3090 data 145,149,157,153,---	<on>
2540 rem -----	<ga>	3100 data ---,142,134,---,---	<stx>
2550 data 000,024,216,088,184,202	<gd>	3110 data ---,---,---,---,150	<lg>
2560 data 136,232,200,234,072,008	<gf>	3120 data ---,140,132,---,---	<sty>
2570 data 104,040,064,096,056,248	<lk>	3130 data ---,148,---,---,---	<cf>
2580 data 120,170,168,186,138,154	<jo>	3140 rem =====e=n=d=e=====	<hl>
2590 data 152,024,056	<on>		
2600 rem -----	<an>		
2610 rem bedingte spruenge	<pf>		
2620 rem -----	<kb>		
2630 data 240,208,144,176,176,144	<ie>		
2640 data 048,016,112,080	<ke>		
2650 rem -----	<gb>		

Fortsetzung von Seite 20.

Gesamtprogramm benötigt wird, die Anzahl der benötigten Byte in den Hauptprogrammen reduzieren. Das Unterprogramm ist nun allgemeiner ausgelegt. Es liefert auch bei X=0 ein richtiges Ergebnis. Dazu bedarf es aber eines Vorwärts-

sprunges. Dem Assembler ist jedoch, wenn er den Sprungbefehl bearbeiten will, noch nicht die Adresse von Label 2 bekannt. Die einfachste Art dieses Problem aus der Welt zu schaffen, ist der Zweipass-Assembler. Dieser arbeitet das zu as-

semblierende Programm nicht nur einmal durch, sondern in zwei Durchgängen. Im ersten werden Sprung- und sonstige Adressen noch nicht ins

DER TWO PASS ASSEMBLER

Programm eingetragen, sondern nur die Adressen der Marken in der LABEL- und Adressen-Liste vermerkt. Beim zweiten Durchlauf sind dem Assembler alle benötigten Adressen bekannt und er kann nun sein Werk ohne Probleme vollbringen.

SPRÜNGE IN DAS BETRIEBSSYSTEM

Wir haben bereits die Label-Definition kennengelernt, mit welcher wir Stellen in unserem zu assemblierenden Programm markieren können. Oft aber möchten wir auch andere bereits fix und fertig in unserem Rechner vorliegende Routinen anspringen können. Zu diesem Zweck führen wir eine weitere Labeldefinition ein, mit deren Hilfe wir direkt eine Sprungadresse übergeben können.

Programmbeispiel:

```
ORG,14336
=LAB,0,65508:REM
  ZEICHENEINGABE
=LAB,1,64490:REM
  ZEICHENAUSGABE
LAB,2, JSR,0
      JSR,1
      JMP,2
```

Mit =LAB, gefolgt von Labelnummer und Sprungadresse können wir nun auch Adressen von Betriebssystemroutinen in die Label-Liste eintragen. Die ORG-Anweisung sagt dem Assembler, wohin er das Maschinenprogramm schreiben soll. In unserem Beispiel können wir es anschließend mit SYS 14336 aufrufen. Eine Ausnahme macht allerdings der C128. Bevor wir hier ein in der BASIC-Bank liegendes Maschinenprogramm starten können,

müssen wir vorher mit BANK 0 die BASIC-Bank aktivieren.

Leider liegen aber die Betriebssystemroutinen in der Bank 15. Da wir an dieser Stelle aber noch nicht auf die Problematik des Bankswitching eingehen wollen, probieren Sie das Beispiel am besten im C64-Modus aus. In einer Endlosschleife werden hier die gedrückten Tasten eingelesen und dann auf den Bildschirm ausgegeben.

LABEL OUT OF RANGE

Diese Fehlermeldung bekommen Sie zu sehen, wenn die Sprungweite für einen bedingten Sprung zu groß ist, wenn die Sprungadresse also außerhalb des möglichen Sprungbereiches liegt. Ist in folgendem Beispiel die Sprungweite zu groß, so müssen wir das Programm, wie weiter unten gezeigt, abändern.

Ausreichende Sprungweite:

```
LAB,0, LDA,#,0
      ...
      BNE,0
      RTS
```

Abänderung bei zu großer Sprungweite:

```
LAB,0, LDA,#,0
      ...
      BEQ,1
      JMP,0
LAB,1, RTS
```

Wir ersetzen unseren bedingten Sprung durch einen unbedingten absoluten. Da dieser bei nicht zutreffender Bedingung nicht ausgeführt werden soll, müssen wir ihn in diesem Falle überspringen.

FORTSETZUNG FOLGT

In dieser Folge unserer Reihe Micro-Computer-Grundlagen haben wir die Sprungbefehle abgehandelt. Was uns jetzt noch fehlt, sind die Befehle zur Datenmanipulation und ihre Adressierungsarten. Weiter geht es damit im nächsten Heft. □

Alfons Mittelmeyer

STRUKTURIERTES SORTIEREN

Von Ast zu Ast

Der Sinn eines Dateiprogrammes besteht darin, Daten eingeben und auf Tastendruck einfach und unkompliziert wieder aufrufen zu können, stimmt's?

Allerdings kennen unsere Acht-Bit-Commodore-Homecomputer keinen Sortierbefehl wie zum Beispiel der Amiga, es ist auch keine Sortieroutine im Betriebssystem eingebaut. Das muß leider je nach Bedarf vom Anwender in sein Dateiverwaltungsprogramm integriert werden. Wir stellen Ihnen hier einen relativ unbe-

kannten, aber sehr effektvollen Sortieralgorithmus vor, den Heap-Sort. Die Bezeichnung stammt aus dem Englischen und bedeutet soviel wie Haufen. Das Programm, das Sie anhand der gravierenden Listingzeilen jederzeit als Unterroutine in eigene Programme einbauen können, macht aus einem bunt zusammenge-

```
10 rem sortieroutine <ab>
20 rem heap sort <ck>
30 for i=1 to 26: cr$=cr$+chr$(29)
: next <dn>
40 ci$=left$(cr$,12):rn$=chr$(18) <jn>
50 m=20:dims$(m) <eo>
60 scnlr: printrn$"unsortiert:
" <ak>
70 for i=1 to m: reads$(i): prints$(i): next: printchr$(19)ci$rn$"sorting: " <ip>
80 le=int(m/2)+1: ri=m <em>
90 if ri<=1 then 210 <b1>
100 if le<=1 then 130 <l1>
110 le=le-1 <nd>
120 z=le: goto 160 <bp>
130 s1$=s$(1): s$(1)=s$(ri): s$(ri)=s1$: printci$;ri; s$(ri) <ac>
140 ri=ri-1 <ef>
150 z=1 <le>
160 s2$=s$(z) <kp>
170 fl=0 <kd>
180 if z*2<=ri and fl=0 then 220 <b1>
190 s$(z)=s2$ <cm>
200 goto 90 <ak>
210 goto 280 <nk>
220 z1=z*2 <pd>
230 if z1<ri then if s$(z1)<s$(z1+1) then z1=z1+1 <jk>
240 if s2$>=s$(z1) then 270 <ol>
250 s$(z)=s$(z1) <aa>
260 z=z1: goto 180 <ag>
270 fl=1: goto 180 <cb>
```

```

280 printchr$(19)cr$rn$"sortie
rt:" <ol>
290 forj=1tom:printer$s$(j):ne
xtj <gk>
300 datapascal,diskette,floppy
,sprites,drucker,chip,ram,rom,
videochip,basic <fi>
310 dataassembler,maske,amiga,
datei,text,grafik,pixel,softwa
re,tastatur,monitor <ie>

```

würfelten Haufen eine mathematisch berechenbare Menge, bei der das größte Element immer an der Spitze steht.

Durch die entsprechenden Programmschritte wird dieses größte Element jeweils ausgefiltert und zur Seite gestellt. Die restliche Menge wird wieder so eingeteilt, daß das Element mit dem höchsten Wert wieder ganz oben steht. Am besten läßt sich diese Struktur mit einem Datenbaum vergleichen. Da durch die programmtechnische Einteilung des Datenfeldes bereits eine grobe Vorsortierung erreicht ist, sind die weiteren Sortierschritte ohne langwierige Suchzeiten durchführbar.

Der Algorithmus handelt nach folgendem Prinzip: Die rechte, obere Grenze des Datenfeldes ist immer gleich der Anzahl der einzelnen Datenelemente; die linke Grenze gibt den Wert der gerade gültigen Suchvariablen (geteilt durch zwei) an. Im Listing finden Sie dies durch die beiden numerischen Variablen *le* und *ri* definiert. Eine Flagge (*flag*) gibt an, ob der nächstzuzuweisen-

de Platz innerhalb des Sortierfeldes diesem gerade in Arbeit befindlichen Datenelement zugewiesen werden kann (definiert durch die Variable *fl*).

Mag auch die Funktionsweise dieser Sortieroutine nicht gleich auf den ersten Blick verständlich sein, so zeigt Ihnen doch das abgedruckte Beispielprogramm nach dem Start mit *Run*, daß *Heap Sort* zu den schnellsten Sortieroutinen gehört. Als *Datas* wurden von uns zufällig ausgewählte Begriffe rund um den Computer gewählt, die Sie zunächst in unsortiertem Zustand auf dem Bildschirm links angezeigt bekommen. Nach Durchlauf des Sortierprogrammes, dessen Arbeitsweise Sie in den Hauptschritten verfolgen können, wird die exakt sortierte Reihenfolge rechts ausgegeben. Bitte, denken Sie immer daran: Ein Dateiverwaltungsprogramm ist nur dann sinnvoll, wenn die von Ihnen gewünschten Daten innerhalb kürzester Zeit wieder abgerufen werden können. Das ist ohne effektives Sortierprogramm kaum realisierbar. *tc*

LICHT IM DISKETTEN-DICKICHT

Lassen Sie sich kein X für ein U vormachen

„Nehmen Sie am besten XY-Disketten im Format 5,25 Zoll. Und achten Sie darauf, daß es sich um die Qualitätsstufe Double Sided, Double Density bei 48 TPI handelt. Viel Spaß noch mit Ihrem neuen Computer.“ . . . und so nimmt sich **COMMODORE WELT** diesem Kauderwelsch, der Computerfachsprache, an. **COMMODORE WELT** erklärt, was die verwirrenden Bezeichnungen bedeuten.

Ein neuer Computer steht zu Hause. Voll ausgerüstet mit Floppy, Drucker und RGB-Monitor: Der ganze Stolz von Friederich Bechtelsheimer. Tag und Nacht sitzt er vor dem Gerät. Programmlisting nach Programmlisting tippt er ein und so kommt es, wie es kommen muß:

Die Diskette ist voll. Also muß eine neue, leere Diskette her; am besten gleich ein ganzer Zehnerpack. Also kratzt Friederich 32,47 Mark seiner letzten Ersparnisse zusammen und geht zum Computershop, drei Straßen weiter. Kaum im Laden, sieht er schon ein Regal voller Diskettenschachteln. Weil beide anwesenden Verkäufer bereits Kundschaft bedienen, die Kasse frei ist und Friederich möglichst schnell zu seinem Computer zurück will, geht er aufs Diskettenregal zu, um sich einen ZehnerPack zu nehmen. Doch bei näherer Betrachtung erkennt er: Da sind Dutzende von verschiedenen Diskettensorten und -marken aufgereiht. Und die Preise reichen von 12,95 Mark bis über 50 Mark. Also wartet er auf Verkäufer. Bereits acht Minuten später

steht der ersehnte Berater bei Friederich und erklärt den Diskettenwirrwarr: Am einfachsten ist die Unterscheidung von Disketten nach deren Größe, gemessen in Zoll. Früher waren Acht-Zoll-Disketten weit verbreitet; mittlerweile wurde diese Größe

„BEZEICHNENDE BEZEICHNUNGEN“

durch 5,25-Zoll-Disketten abgelöst, die ihrerseits in Zukunft den 3,5-Zoll-Disketten Platz machen. Anhand der Größe der Diskettenhülle ist zu erkennen, um welchen Typus es sich handelt. Die Größe der benötigten Disketten ist abhängig vom Laufwerk.

Außer physikalischen Unterschieden sind Disketten in unterschiedliche Qualitätsstufen eingeteilt. Dabei gilt: Je leistungsstärker das Diskettenlaufwerk, um so hochwertiger müssen die verwendeten Magnetscheiben sein.

Ein Beispiel: Betreiber der Floppy 1541 können Disketten sehr niedriger Qualitätsstufen nutzen, da der Floppy-Schreib-Lesekopf zirka 170 KByte auf Diskette überträgt.



BÜRGERKRIEG IM LIBANON

Menschen in Not brauchen Hilfe: zuverlässig, schnell, wirksam. Die beiden kirchlichen Hilfswerke nehmen ihren Auftrag ernst.

Deutscher Caritasverband, Konto 202, Postgiro Karlsruhe oder Banken und Sparkassen.

Diakonisches Werk, Konto 502, Postgiro Stuttgart oder Banken und Sparkassen.

Kennwort: BÜRGERKRIEG IM LIBANON

Dabei ist es uninteressant, ob die benutzte Diskette für diese Kapazität oder für Datenspeicherungen von 500 KByte und mehr ausgelegt ist. Floppy 1541-Besitzer verwenden demnach Disketten geringer Qualität. Anders sieht es bei SFD 1001-Nutzern aus. Dieses Laufwerk speichert zirka 1 MByte Daten, das sechsfache einer 1541-Station. Dementsprechend sind Disketten mit hohen Kapazitäten zu verwenden. Andernfalls ist mit Aufzeichnungsfehlern und Datenverlusten zu rechnen. Vor allem nach längerer Nutzungsdauer steigen Read- oder Write-Error-Zahlen an.

WAS IST WAS?

Welche Aufzeichnungskapazität eine Diskette besitzt, ist unter verschiedenen Bezeichnungen ersichtlich. Zum einen gibt es Unterschiede in der *Aufzeichnungsdichte*, bezeichnet als Single Density und Double Density oder, in Deutsch, einfache Dichte und doppelte Dichte.

Mehrere Hersteller verwenden Kurzbezeichnungen: sd für einfache Dichte und dd für doppelte Dichte. Grundsätzlich sind Disketten doppelter Dichte zu bevorzugen. Der Unterschied zwischen sd und

dd: dd-Disketten durchliefen erfolgreich einen Qualitätstest, bei dem doppelt so viele Daten auf Magnetscheibe gespeichert und wieder gelesen wurden, als bei einer sd-Diskette. Oftmals sind sd-Disketten bei dd-Disketten-tests „durchgefallen“ und haben sich erst bei sd-Tests zum Verkauf qualifiziert.

Weitere Qualitätsunterschiede sind durch die Bezeichnungen ss und ds festgehalten: ss steht für single sided, zu deutsch, einseitig und ds für double sided, zweiseitig. Viele Computernutzer lesen aus diesen Worten falsche Bedeutungen heraus: Eine einseitige Diskette ist nicht etwa nur auf einer Seite mit einer Magnetschicht versehen, sondern nur auf einer Seite getestet.

Mit anderen Worten: Der Hersteller garantiert bei einseitigen Disketten hundertprozentige Funktionsicherheit der Schreib-/Leseseite, während er bei doppelseitigen Disketten diese Funktionsgarantie auch auf Seite zwei ausweitet. Erfahrungsgemäß sind einseitige Disketten ohne Probleme durch Diskettenlocher zweiseitig zu benutzen. Bei Preisunterschieden über 20 Prozent sind einseitige Disketten vorzuziehen. Ausnahmen sind wichtige Arbeitsdisketten.

Fachleute werfen gerne mit Fachbegriffen um sich. Bei Disketten gehört TPI, Tracks Per Inch, zu diesen gerne benutzten Ausdrücken. Im 5,25-Zoll-Bereich sind 48- und 96-TPI-Disketten, im 3,5-Zoll-Bereich 57,5- und 135-TPI-Disketten handelsüblich. TPI gibt an, wieviel Datenringe innerhalb eines inch liegen dürfen. Auf Disketten sind Daten ringförmig um den Mittelpunkt angeordnet. Je mehr Ringe innerhalb einer Strecke vom Mittelpunkt nach außen führend Platz finden, um so höher sind Datendichte der Diskette und die TPI-Zahl.

WAS BEDEUTET TPI?

Benutzer leistungsfähiger Laufwerke sind auf Disketten mit hohen TPI-Werten angewiesen. Nutzt man ein Laufwerk oder Diskettenbetriebssystem, das die garantierte TPI-Zahl einer Hochleistungsdiskette nicht erreicht, sind Magnetscheiben niedriger Qualität sinnvoll.

Benutzer leistungsfähiger Laufwerke sind auf Disketten mit hohen TPI-Werten angewiesen. Nutzt man ein Laufwerk oder Diskettenbetriebssystem, das die garantierte TPI-Zahl einer Hochleistungsdiskette nicht erreicht, sind Magnetscheiben niedriger Qualität sinnvoll.

ÜBER MARKEN UND SANDKÖRNER

Diskettenmarken gibt es wie Sand am Meer. Manchmal ähneln sich Disketten verschiedener Marken genauso wie Sandkörner. Nicht selten beliefert ein Diskettenher-

steller mehrere verschiedene Markendiskettenvertriebe mit ein und derselben Diskettenserie. Alles, was da noch an Unterschieden im Laden bleibt, sind

- Preise;
- Markenaufkleber;
- Verpackungen;
- und Werbesprüche.

Mehr und mehr ist daher der Trend zum Kauf günstiger No-Name-Disketten zu verstehen. Überdies bieten viele Markendisketten-Vertriebe ihren Händlern No-Name-Disketten zu günstigen Preisen an, deren einziger Unterschied zu den Markendisketten das Fehlen der Markenaufkleber ist. Auf diesem Weg versuchen diese Firmen, Überbestellungen möglichst rasch zu verkaufen, ohne dabei die Preise ihrer eigenen Marken zu drücken.

WAS HÄNDLER NICHT VERRATEN

Jedoch: Computerhändler verkaufen gerne Markendisketten. Sie versuchen damit, Konkurrenten wie Kaufhäusern zu kontern, welche No-Name-Produkte zu ruinösen Preisen anbieten. Außerdem: 10 Prozent Gewinn an zehn Mark teuren Disketten sind eine Mark, während 10 Prozent von 40 Mark immerhin vier Mark Gewinn ergeben . . .

Sven Faulhaber □

Zwei Themen — ein Ereignis:

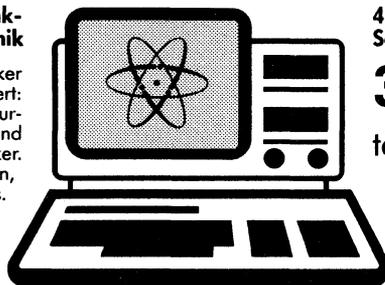
Hobby-tronic & COMPUTERSCHAU



11. Ausstellung für Funk- und Hobby-Elektronik

Die umfassende Marktübersicht für Hobby-Elektroniker und Computeranwender, klar gegliedert: In der Westfalenhalle 5 das Angebot für CB- und Amateurfunkler, Videospiele, DX-er, Radio-, Tonband-, Video- und TV-Amateure, für Elektro-Akustik-Bastler und Elektroniker.

Mit dem Actions-Center und Laborversuchen, Experimenten, Demonstrationen und vielen Tips. In der Westfalenhalle 6 das Superangebot für Computeranwender in Hobby, Beruf und Ausbildung. Dazu die Mikrocomputer-Beratung und die Stände der Computerclubs.



4. Ausstellung für Computer, Software und Zubehör

3.-7. Februar 1988

täglich 9-18 Uhr

Stark verbilligte Sonderrückfahrkarte an allen Bahnhöfen der DB — Mindestentfernung 51 km außerhalb VRR — plus Eintrittsermäßigung.

Messezentrum Westfalenhallen Dortmund

SUCHEN UND FINDEN

Machen Sie mehr aus Ihrem 128!

Kaum ein Home-Computer aus der Acht-Bit-Familie von Commodore eignet sich so uneingeschränkt zur Datenverwaltung und -bearbeitung wie der 128PC, nicht zuletzt durch seine Darstellungsmöglichkeit von 80 Zeichen pro Textzeile.

Von dem grandiosen Speicherplatz in Bank 0 und Bank 1 (Variablenspeicher) gar nicht zu reden.

Der Computer-Anwender kennt zwei Arten von Dateien: Zum einen die fertigen, bei denen die Reihenfolge, sprich Sortierung stimmt und die nach Möglichkeit nicht mehr verändert werden sollte. So eine fertige Datei dient als Nachschlagewerk zum Beispiel für Adressen, die sich auf dem Bildschirm und/oder auch dem Drucker gesamt oder einzeln pro Datensatz ausgeben läßt. Die Datensätze sind wie in einem Karteikasten ordentlich gereiht abgelegt, was zum Suchen und Finden natürlich sehr hilfreich sein kann.

Die andere Art, wollen wir sie einmal offene Datei nennen, besitzt den Charakter eines unendlichen Notizzettels (unend-

handelt und können nicht zu Berechnungen herangezogen werden. Kommt Ihnen das bekannt vor? Richtig, eine Menge Textverarbeitungsprogramme verfolgen dieses Datei-Prinzip.

Daß hier eine der üblichen Sortier Routinen, die Datenfelder zum Beispiel alphabetisch sortieren, fehl am Platze wäre, versteht sich von selbst, denn die Datenfelder sollten ihre Position im gesamten Datensatz so behalten, wie sie eingegeben wurden.

Nehmen wir doch gleich ein praktisches Beispiel: Als eifriger Leser unserer COMMODORE WELT haben Sie sich schon seit längerem mit dem Gedanken getragen, den Inhalt der einzelnen monatlich

IM PRINZIP UNSORTIERT: OFFENE DATEIEN

lich selbstverständlich nur bis zur Grenze des verfügbaren Speicherplatzes in Ihrem Computer), bei der ein Datensatz praktisch die Datei selbst ist. Ein Datenfeld umfaßt dabei der Einfachheit halber eine Textzeile auf dem Bildschirm (es ist selbstverständlich auch eine logische Eingabezeile möglich, beim C128 beträgt sie 160 Zeichen), und es lassen sich Textzeichen in beliebiger Vielfalt eingeben. Zahlen werden hierbei ebenfalls wie ein Text be-

EIN ANWENDUNGS- BEISPIEL

erscheinenden Ausgaben endlich einmal nach Sachgebieten aufgeteilt zu katalogisieren. Es ist doch zu dumm, immer sämtliche Exemplare eines Jahrgangs durchblättern zu müssen, nur um festzustellen, wo überall etwas zum Thema Sprites steht, weil Sie gerade darüber einiges nachlesen möchten. Auch wenn Sie sich nur durch die Inhaltsverzeichnisse arbeiten, bei zwölf oder mehr Heften kann das ganz schön frustrie-

SUCHDATEI 128

```

10 rem suchdatei =====128 <pg>
20 rem (c) commodore welt == <al>
30 rem ===== <ng>
40 rem by (hb) == <lo>
50 rem == <if>
60 rem == <nd>
70 rem version 7.0 80z/ascii == <on>
80 rem c 128 + floppy+ drucker == <po>
90 rem ===== <jg>
100 color6,7:color5,4 <om>
110 fast:sys51328:sys51372>window0
,0,79,24,1:poke2603,71:print:gosub
840:gosub850 <no>
120 fori=0to38:ln$=ln$+chr$(192):n
ext:fl=0 <mh>
130 n=750:dima$(n),s1$(n),s2$(n) <bo>
140 fori=0to76:11$(1)=11$(1)+chr$(
192):next <oo>
150 11$(2)=left$(11$(1),40) <do>
160 fori=0to76:11$(3)=11$(3)+chr$(
32):next <oj>
170 11$(4)=left$(11$(3),40) <mn>
180 lr$=""
":rem 25 spaces <fn>
190 um$(1)=chr$(176)+11$(1)+chr$(1
74) <pf>
200 um$(2)=chr$(221)+11$(3)+chr$(2
21) <fe>
210 um$(3)=chr$(173)+11$(1)+chr$(1
89) <gg>
220 ur$(1)=chr$(176)+11$(2)+chr$(1
74) <fl>
230 ur$(2)=chr$(221)+11$(4)+chr$(2
21) <pk>
240 ur$(3)=chr$(173)+11$(2)+chr$(1
89) <dd>
250 l=1:gosub660:char1,0,0:gosub84
0:gosub1300 <el>
260 getkeyq$ <ah>
270 ifq$="n"goto420 <gk>
280 ifq$="h"orq$="p"then1440 <hi>
290 ifq$="b"thengosub1070:goto250 <km>
300 ifq$="d"goto1130 <an>
310 ifq$="a"goto1820 <eg>
320 ifq$="e"thenend <ip>
330 ifq$="l"thengosub740:goto250 <ek>
340 ifq$="s"goto970 <lj>
350 ifq$="v"goto2030 <pf>
360 ifq$="c"thengosub560:gosub570:
goto250 <od>
370 ifq$="x"then2120 <je>
380 goto260 <ib>
390 rem ===== <ej>
400 rem = neueingabe = <ck>
410 rem ===== <bb>
420 l=1:gosub710 <an>
430 char1,1,1,"NEUEINGABE TEXT-DAT

```

TIPS & TRICKS

```

EI ":print:gosub830 <cc>
440 char1,1,4,"Dateiname: ":poke21 <on>
,64:inputcm$:poke21,0
450 window0,8,79,24,1:gosub670:gos <be>
ub580>window0,9,79,24 <fn>
460 ze=1:cz=0 <hk>
470 printtab(2):poke21,64 <po>
480 inputa$(ze):poke21,0:print <cb>
490 ifleft$(a$(ze),1)=chr$(42)then <of>
900 <ld>
500 ze=ze+1:cz=cz+1:ifcz>14thencz= <po>
14 <dl>
510 goto470
520 rem = unterprogramme ===== <mg>
530 poke208,0:wait208,1:return <no>
540 open127,4:poke768,185:print#12 <jc>
7:close127:poke768,63:ifst=-128the <bk>
nchar1,1,3,"Drucker nicht betriebs <dj>
bereit !!!":sleep2 <pb>
550 return <lk>
560 window0,0,79,24:char1,1,1,ur$( <el>
1):fori=2to21:char1,1,i,ur$(2):nex <cl>
t:char1,1,22,ur$(3):window2,2,41,2 <ho>
1,1:return <oc>
570 directory:ifst=64thenprint:gos <ce>
ub590:gosub530:return
580 printtab(1)ln$ln$:return <bd>
590 println$:return
600 print#4,tab(30);:print#4,"-";a <do>
f;"-":return
610 ifaf=2thenza=0
620 ifi=b1thenforab=1to8:print#4:n <gn>
extab:af=af+1:gosub600 <ek>
630 return <al>
640 window0,8,79,24,1:char1,0,0,cm <gn>
$:window0,9,79,24,1:gosub580:windo <ip>
w0,10,79,24:return <eh>
650 fori=1tozx:print#4:next:return <gn>
660 window0,0,79,24,1:char1,1,0,um <ip>
$(1):print:fori=1to22:char1,1,i,um <eh>
$(2):print:next:char1,1,23,um$(3): <gn>
print:return <ip>
670 window60,0,76,2>window0,8,79,2 <eh>
4:return
680 window0,0,79,24:char1,20,6,ur$( <gn>
1),1:print:fori=7to11:char1,20,i, <ip>
ur$(2),1:print:next:char1,20,12,ur <eh>
$(3),1:print
690 window21,7,60,11,1 <gn>
700 return
710 window0,0,79,24:char1,0,1,ur$( <ip>
1),1:print:fori=2to6:char1,0,i,ur$( <eh>
2),1:print:next:char1,0,7,ur$(3), <gn>
1:print
720 window1,2,40,6,1 <ip>
730 return <eh>
740 l=0:gosub710:char1,1,0,"D A T <gn>
E N loeschen.",1:print <mg>

```

rend sein, bis man fündig wird.

Kurzer Rede langer Sinn: Eine Datei namens **COMMODORE WELT** muß angelegt werden (das abgedruckte Listing ist das Programm dazu). Auch muß diese Datei jederzeit erweiterbar sein, schließlich erscheint jeden Monat eine neue Ausgabe, die ebenfalls katalogisiert werden muß. Auf einfache Weise sollte dann der gewünschte Suchbegriff eingegeben werden können und der Bildschirm und/oder Drucker nach angemessener Suchfrist alle Datenfelder ausspucken, die mit dem Suchbegriff irgendwie zu tun haben.

Auch wenn's so einfach klingt, ohne ein gewisses System, das bereits bei der Dateneingabe befolgt werden sollte, wird dabei nichts Vernünftiges rauskommen.

Falls Sie das Programm Suchen und Finden zu dem eben gemachten Vorschlag verwenden möchten (es lassen sich natürlich beliebige andere Inhalte von Zeitschriften

ERLEICHTERT DAS SUCHEN: FORMATIERTE EINGABE

oder auch Video- und Audio-Dateien erstellen), empfehlen wir Ihnen folgende Eingabe-Syntax für die Datenfeldzeile, da dann die Suche am wenigsten Zeit benötigt:

Heft/Nr. Suchbegriff/
Inhalt*Seite

CW01/87 GRAF/Grafik-
Booster.128*14
(7 Sp.) (1 Space) (ab 9.
Spalte Suchbegriff

.....Inhalt/ab
Seite.....

Schließen Sie diese Eingabe bitte mit RETURN ab.

Die Suchroutine im Programm ist so ausgerichtet, daß sie ab der neunten Spalte oder Eingabezeile eines Datenfeldes nach dem von Ihnen vorgegebenen Sammelbegriff sucht. Der darf nicht län-

ger als vier Zeichen sein (in unserem Beispiel GRAF für Grafik). Die Schreibweise des Suchbegriffes muß unbedingt identisch mit der sein, die Sie im jeweiligen Datenfeld verwendet haben. (Also nicht einmal graf und das nächstmal wieder GRAF oder gemischte Schreibart.)

Wenn Sie das beachten, so findet der C128 auch bei großen Datensätzen relativ schnell alle Datenfelder heraus, die mit dem Kennzeichen GRAF versehen sind und listet Sie Ihnen auf. Sollten Sie sich trotzdem einmal geirrt haben, so ist nichts verloren, denn nun schaltet der Computer auf die penible Methode. Er durchsucht jedes Byte jeden Datenfeldes, ob ir-

DIE LANGWIERIGERE METHODE: ALLE ZEICHEN WERDEN DURCHSUCHT

gendeine Gemeinsamkeit mit dem von Ihnen eingegebenen Suchbegriff besteht. Daß das selbstverständlich seine Zeit dauert, vor allem bei großen Dateien, muß wohl nicht eigens erwähnt werden, aber Sie kommen auf diese Weise zum Ziel. Machen Sie zum Spaß mal die Probe aufs Exempel und lassen Sie bei einer bereits erstellten Datei etwa nach dem Suchbegriff x suchen. Nach einiger Zeit werden Sie feststellen, daß Ihnen der C128 alle Datenfelder auflistet, in denen irgendwo in einem Wort ein x enthalten ist. Statt x können Sie auch einen anderen Buchstaben nehmen.

Soviel zur Dateneingabe und zum Aufruf des Suchbegriffes, der von Ihnen frei wählbar ist. Das Programm Suchen und Finden für den 128 PC läuft nur im 80-Zeichen-Modus und ist ein überarbeitetes Modul aus dem weitaus vielseitigerem Programm Multi-Datei (erschienen auf COMMODORE DISC Nr. 6).

TIPS & TRICKS

```

750 char1,1,1,"Sind Sie sicher (j/
n) ?",1:print <hl>
760 getkeya$ <dp>
770 ifa$="j"then780:else:return <en>
780 char1,1,2,"Bitte warten...",1:
print:i=0:dountili=ze:a$(i)="" :i=i
+1:loop:cm$="" <eg>
790 char1,1,4,"Arbeitsspeicher ist
jetzt leer.",1 <mj>
800 sleep3:return <gl>
810 printtab(20):i=0:dountili=36:p
rintchr$(96);:i=i+1:loop:print <gi>
820 return <pm>
830 char1,1,3,"(Eingabe beenden mi
t '*' ):print:return <fj>
840 char1,20,1,"SUCHEN UND FINDEN
Commodore 128":print:return <gj>
850 char1,21,2,"Offene Datei mit S
uchroutine":char1,22,4,"(C) COMMOD
ORE WELT / h.b.":return <jp>
860 window30,13,79,24:l=0:char1,0,
1,ur$(1):print:fori=2to10:char1,0,
i,ur$(2):next:print:char1,0,11,ur$(
3):window31,14,70,22:return <hi>
870 i=1:dountili=ze <lf>
880 ifa$(i)=""then1980:loop <md>
890 return <ii>
900 l=0:gosub680:char1,1,1,"Daten
speichern ? (j/n)",1:print <dm>
910 char1,1,3,"Falls noetig, Daten
disk einlegen !",1:print <hj>
920 getkeya$ <fb>
930 ifa$="j"then970:else:goto250 <hf>
940 rem ===== <im>
950 rem = daten speichern = <ap>
960 rem ===== <do>
970 l=0:gosub680:char1,1,0,"TEXT-D
ATEI speichern",1:print <ho>
980 char1,1,1,"File-Name oder -num
mer: "+cm$,1:print <kh>
990 char1,1,2,cm$+" wird jetzt ges
ichert",1:print:char1,1,3,"Bitte w
arten.",1 <co>
1000 dopen#2,"@"+(cm$),w <lb>
1010 print#2,ze <he>
1020 i=1:dountili=ze:print#2,a$(i)
:i=i+1:loop:print#2,chr$(10):close
2 <cc>
1030 gosub1930:goto1200 <df>
1040 rem ===== <op>
1050 rem = daten laden = <ca>
1060 rem ===== <gb>
1070 l=1:gosub680:char1,1,0,"TEXT-
DATEI laden.":print <fh>
1080 fori=0to20:a$(i)="" :next <no>
1090 char1,1,1,"Dateiname: ":poke2
1,64:inputcm$:poke21,0:print <fk>
1100 dopen#2,(cm$):char1,1,2,cm$+"
wird geladen.":input#2,ze:ifds<>0
thenchar1,1,3,ds$:sleep2:dclose:go
sub560:gosub570:goto1070 <kp>
1110 u=1:dountilu=ze+1:input#2,a$(
u):u=u+1:loop:dclose#2 <nj>
1120 gosub1930:return <gd>
1130 l=0:gosub710:char1,1,0,"Bildd
schirm-Ausgabe TEXT-DATEI ",1:print <cd>
1140 char1,1,2," E einzelschritt pe
r Tastendruck",1:print <ao>
1150 char1,1,4," G esamte Datei an
zeigen",1 <da>
1160 getkeya$ <fo>
1170 ifa$="e"then1660 <mn>
1180 ifa$="g"then1720 <kj>
1190 else:goto1160 <fb>
1200 l=0:gosub680:char1,1,1," D =
Drucken ",1:print:char1,1,3," M =
Menue",1 <ej>
1210 getq$ <dp>
1220 ifq$="d"thengosub1420:goto250 <n>
1230 ifq$="m"then250 <mn>
1240 goto1210 <ai>
1250 goto1210 <ei>
1260 else:goto1210 <ha>
1270 rem ===== <cm>
1280 rem = ein/ausgabemenue = <go>
1290 rem ===== <dl>
1300 print:printtab(20)chr$(18)" N
"chr$(146)"eue Datei eingeben":pr
int <bi>
1310 printtab(20)chr$(18)" S "chr$(
146)"peichern einer Datei":print <hc>
1320 printtab(20)chr$(18)" B "chr$(
146)"estehende Datei laden":print <md>
1330 printtab(20)chr$(18)" A "chr$(
146)"nfuegen an bestehende Datei"
:print <ga>
1340 printtab(20)chr$(18)" D "chr$(
146)"aten auf Bildschirm":print <jc>
1350 printtab(20)chr$(18)" H "chr$(
146)"ardcopy (Druckerausgabe)":pr
int <je>
1360 printtab(20)chr$(18)" L "chr$(
146)"oeschen von Daten":print <eb>
1370 printtab(20)chr$(18)" V "chr$(
146)"eraendern/editieren":print <ao>
1380 printtab(20)chr$(18)" X "chr$(
146)" Liste nach Suchbegriff":pri
nt <dk>
1390 printtab(20)chr$(18)" C " chr
$(146)"atalog/Directory";:printtab
(60)chr$(18)" E "chr$(146)"nde":pr
int <bd>
1400 gosub810:return <mb>
1410 rem ===== <ag>
1420 rem = daten ausdrucken = <pb>
1430 rem ===== <gk>

```

TIPS & TRICKS

```

1440 l=0:gosub710:char1,1,0,"TEXT-
DATEI drucken",1:print <fd>
1450 scnclr:print"S = Text im Spei
cher":print"L = laden" <mb>
1460 gett$:ift$="-s"then1490 <ff>
1470 ift$="l"thengosub1070:gosub53
0:goto1490 <mk>
1480 goto1460 <df>
1490 l=1:gosub680:char1,1,0,"Aende
rung der " :print <hf>
1500 char1,1,1,"Sekundaeradresse":
print <ib>
1510 printtab(20)chr$(145)"7"chr$(
157)chr$(157)chr$(157);:inputsa <pl>
1520 print" Wenn Drucker bereit is
t, <RETURN> " <bi>
1530 gosub530:gosub540 <gp>
1540 ifst=-128then250 <oa>
1550 af=1:za=0:open4,4,sa:cmd4 <hb>
1560 iffg=0andaf=1then1590 <ek>
1570 iffg=0andaf>1thengosub600:got
o1590 <lo>
1580 print#4,cm$:print#4 <ei>
1590 fori=1toze:b1=af*62-za <dg>
1600 gosub610 <dc>
1610 print#4,a$(i):next <kh>
1620 print#4:close4:fg=0:goto250 <cc>
1630 rem ===== <cd>
1640 rem = bildschirm-ausgabe = <mh>
1650 rem ===== <bi>
1660 gosub870 <bd>
1670 e=1:gosub640 <bk>
1680 printe;spc(2)a$(e):gosub530 <fb>
1690 ife=ze-1thenprint:gosub580:go
sub530:goto1200 <hc>
1700 e=e+1 <ng>
1710 goto1680 <cj>
1720 gosub870 <mp>
1730 s=1:gosub640 <ge>
1740 prints;spc(2)a$(s) <pe>
1750 ifs=ze-1thenprint:gosub580:po
ke208,0:wait208,1:goto1200 <np>
1760 s=s+1 <gf>
1770 goto1740 <jn>
1780 rem ===== <cm>
1790 rem = append (daten an- = <nh>
1800 rem = haengen) = <bh>
1810 rem ===== <ma>
1820 l=0:gosub710:gosub870:gosub18
90 <ig>
1830 char1,1,4,"Eingabe beenden mi
t '*'",1:ze=ze-1 <an>
1840 window0,0,79,24:char1,0,8,cm$
:gosub670:print>window0,9,79,24,1:
printtab(3)a$(ze) <nh>
1850 ze=ze+1:printtab(2);:poke21,6
4:inputa$(ze):poke21,0:print <co>
1860 ifleft$(a$(ze),1)=chr$(42)the

```

Fortsetzung von Seite 27

ANLEITUNG UND PROGRAMM-BESCHREIBUNG

Nach dem Laden und Star-
ten mit RUN erscheint
das Hauptmenü, das folgen-
de Möglichkeiten bietet:

- Neue Datei eingeben
- Speichern einer Datei
- Bestehende Datei laden
- Anfügen an Datei
- Datenausgabe auf Bildschirm
- Hardcopy (Ausgabe auf Drucker)
- Löschen einer Datei
- Verändern/Editieren
- Liste nach Suchbegriff
- Catalog/Directory (Anzeigen des Disketteninhaltsverzeichnis)
- Ende.

Aufrufen können Sie die einzelnen Funktionen durch Drücken des entsprechenden, revers kenntlich gemachten Buchstabens.
Hier eine Beschreibung der Optionen:

N Neue Datei eingeben

Links oben erscheint ein Window, das Sie nach dem Namen der Datei fragt, die Sie jetzt anlegen möchten. Sodann kommen Sie in den Eingabemodus, der blinkende Strichcursor zeigt es Ihnen an. Nun können Sie Ihre Eingaben in der Form wie vorher besprochen tätigen. Schließen Sie jede Eingabezeile mit RETURN ab, es spielt keine Rolle, wenn die Zeile kürzer als achtzig Zeichen ist. Beenden Sie die Eingabe und den Datensatz mit der Taste * (Stern). Danach haben Sie die Möglichkeit, Ihre bisherigen Einträge sofort auf Diskette abzuspeichern, ansonsten kommen Sie wieder ins Hauptmenü.

S Speichern

Der Speichervorgang wird hier ohne Sicherheitsabfrage ausgelöst. Alle Daten, die im Speicher stehen, werden als sequentielles File mit dem bei N angegebenen Dateinamen auf

Diskette gespeichert. Beachten Sie bitte, daß das Programm hierbei den Diskbefehl At Sign (den Klammeraffen) benutzt. Alle Dateien werden bei jedem Speichervorgang neu überschrieben, wenn Sie denselben Dateinamen aufweisen. Danach werden Sie in einem Fenster gefragt, ob Sie diese Datei ausdrucken lassen oder lieber zum Hauptmenü zurückkehren wollen.

B Bestehende Datei laden

Damit können Sie ein erstelltes Datenfile von Disk laden, vorher müssen Sie aber den Dateinamen angeben. Abkürzungen mit dem Jokerzeichen * sind hier nicht erlaubt, weil das Sternchen in diesem Programm eine andere Funktion hat. Falls Sie vor dem Ladevorgang den Dateinamen nicht mehr genau wissen, so benutzen Sie die Directory-Funktion mit der Taste C (vom Befehl CATALOG des BASIC 7.0).

A Anfügen an bestehende Datei

Da es sich hier, wie bereits erwähnt, um eine offene Datei handelt, sollten nach Möglichkeit Daten angefügt werden können. (Allerdings setzt uns auch hier der Speicherplatz Grenzen.) Die zu ergänzende Datei muß sich im Speicher des C128 befinden. Nachdem Sie diese Option ausgewählt haben, befinden Sie sich wieder im Eingabemodus (wie unter Menüpunkt N), zur Orientierung wird das letzte Datenfeld der Gesamtdatei angezeigt. Fügen Sie nun Ihre weiteren Datenzeilen an und beschließen Sie diese Eingaben wieder mit der Sternchen-Taste. Auch jetzt werden Sie gefragt, ob Sie die erweiterte Datei im gesamten wieder abspeichern möchten, was ja auch am naheliegendsten ist.

TIPS & TRICKS

```

n1880 <jc>
1870 goto1850 <dg>
1880 print:goto900 <eg>
1890 ifze=900then1980 <la>
1900 char1,1,0,"TEXT-DATEI ERWEITE
RN.",1:print <io>
1910 char1,1,1,"Datenfelder bisher
:"+str$(ze-1),1:print <dn>
1920 char1,1,2,"Letztes Datenfeld
wird angezeigt.",1:print:return <dj>
1930 rem ===== <kc>
1940 rem = disk-fehler ? = <nn>
1950 rem ===== <kp>
1960 ifds<>0thenprintds$:cm$="":go
to250 <dd>
1970 return <ab>
1980 l=0:gosub680:char1,1,2,"Kein
e Daten im Speicher !!!",1 <fo>
1990 char1,1,4,"<RETURN> = MENUE"
,1:gosub530:goto250 <fg>
2000 rem ===== <el>
2010 rem = text veraendern/edit. = <ia>
2020 rem ===== <np>
2030 ifa$(1)=""thengosub1980:goto2
50 <am>
2040 l=1:gosub710:char1,1,1,"TEXT-
Datei aendern/editieren.":print <mb>
2050 input"Datenfeld-Nr.":xe:wind
ow0,8,79,24,1:char1,1,0,"Edition T
EXT-Datei":print <ke>
2060 printtab(3)a$(xe):window0,0,7
9,24:char1,0,9 <ih>
2070 printtab(2);:poke21,64:inputa
$(xe):poke21,0:printchr$(13);:goto
900 <hj>
2080 rem ===== <gb>
2090 rem = auflistung nach = <cp>
2100 rem = suchbegriff = <je>
2110 rem ===== <nb>
2120 ifze=0thengosub1980:goto250 <bl>
2130 l=1:gosub710:char1,1,1,"Aufli
stung per Suchbegriff" <on>
2140 fori=1ton:s1$(i)="" :next:sn=0
:fl(1)=0 <nj>
2150 char1,1,2:input"Suchbegriff "
;sn$:char1,1,4,"Bitte warten..." <mj>
2160 fori=0toze <ik>
2170 iflen(sn$)=4andleft$(sn$,4)=m
id$(a$(i),9,4)thengosub2470 <ec>
2180 next:iff1(1)=1then2220 <hc>
2190 fori=0toze:fork=1tolen(a$(i)) <hf>
2200 ifsn$=mid$(a$(i),k,len(sn$))t
hengosub2470 <fl>
2210 nextk,i <il>
2220 window0,0,79,24,1:fori=0tosn:
prints1$(i):next:char1,70,23,"<TAS
TE>",1:gosub530 <nb>
2230 l=0:gosub680:char1,1,3,"D = d
rucken ",1:print:char1,1,4,"M = ME
NUE",1:char1,1,0,"N = Neufile (Suc
hliste)",1:print:char1,1,1,"A = AP
PEND (Suchliste)",1:print <dk>
2240 getx$:ifx$="m"then250 <gm>
2250 ifx$="n"thenvb=sn:gosub2510:g
oto2230 <id>
2260 ifx$="a"thengosub2620:goto223
0 <db>
2270 ifx$="d"then2300 <mj>
2280 goto2240 <jb>
2290 rem ===== <cm>
2300 rem = druckeroutine fuer = <hf>
2310 rem = suchliste = <jj>
2320 rem ===== <ma>
2330 l=1:gosub680:char1,1,0,"Such-
Liste drucken.",1:print <kh>
2340 char1,1,1,"Aenderung der" <ho>
2350 char1,1,2,"Sekundaeradresse:"
:print <bh>
2360 printtab(20)chr$(145)"7"chr$(
157)chr$(157)chr$(157); <en>
2370 inputsa <fe>
2380 char1,1,3,"Wenn Drucker berei
t ist, <RETURN>" <hc>
2390 gosub530:gosub540 <cn>
2400 ifst=-128then250 <bg>
2410 s=0:af=1:open4,4,sa:cmd4 <gk>
2420 print#4,tab(5)sn$,fl$:print#4 <fe>
2430 fori=1tosn:b1=af*62 <fl>
2440 gosub610 <pa>
2450 print#4,s1$(i):nexti <bd>
2460 print#4:close4:goto2230 <nk>
2470 s1$(sn+1)=a$(i):sn=sn+1 <po>
2480 fl(1)=1 <dn>
2490 return <bf>
2500 rem ===== <ko>
2510 rem = neufile liste per = <ki>
2520 rem = suchbegriff speichern = <md>
2530 rem ===== <np>
2540 l=0:gosub680:vb=sn+1:fori=1to
vb:s2$(i)=s1$(i):next <el>
2550 char1,1,2,"Saving "+sn$,1:cha
r1,1,3,"Bitte warten...",1 <fd>
2560 fi$=sn$:dopen#2,"@"+(fi$),w <ln>
2570 print#2,vb <dm>
2580 fori=1tovb:print#2,s2$(i):nex
t:print#2,chr$(10):dclose#2 <mg>
2590 fori=1tovb:s2$(i)="" :next:vb=
0 <pp>
2600 return <pc>
2610 rem ===== <mi>
2620 rem = append an liste per = <dj>
2630 rem = suchbegriff = <jj>
2640 rem ===== <lh>
2650 l=0:gosub680:char1,1,1,"Daten
werden an",1:print:char1,1,2,sn$+
" angehaengt.",1 <ng>

```

```

2660 dopen#2,(sn$):input#2,vb:fori
=1tovb:input#2,s2$(i):next      <ap>
2670 dclose#2                    <cl>
2680 append#2,(sn$)             <hp>
2690 forj=1tosn+1:s2$(vb)=s1$(j):p
rint#2,s2$(vb):vb=vb+1:next:dclose
#2:vb=vb-1:gosub2560            <em>
2700 return                      <ll>

```

ready.

D Daten auf Bildschirm

Die Datei im Computerspeicher läßt sich damit am Bildschirm auflisten, wobei Sie die Wahl zwischen zwei Vorgehensweisen haben: Gesamte Datei auflisten (zum Anhalten des Listvorgangs benutzen Sie bitte die NO-SCROLL-Taste) oder jede Datenzeile einzeln. Die Zahlen vor den einzelnen Zeilen bedeuten die Nummern der Datenfelder. Ein Druck auf eine beliebige Taste, am besten SPACE, bringt Ihnen die nächste Datenzeile auf den Bildschirm. Nachdem Sie am Datensatzende angelangt sind, fragt Sie ein Fenster, ob Sie ausdrucken lassen oder wieder ins Hauptmenü zurückkehren möchten.

H Hardcopy-Routine (Ausgabe auf Drucker)

Nach Wahl dieses Menüpunktes werden Sie zuerst gefragt, ob sich die ausdruckende Datei bereits im Speicher befindet oder ob Sie sie erst von Diskette nachladen möchten. Dann haben Sie in einem weiteren WINDOW die Möglichkeit, die Sekundäradresse zu ändern. Eingestellt ist die Nr. 7, die im Commodore-Modus für die Klein-/Großschrift zuständig ist. Diese Sekundäradresse kann je nach Drucker oder benutztem Interface verschieden sein, ziehen Sie dazu bitte Ihr Druckerhandbuch zu Rate. Wichtig zu erwähnen ist noch, daß der Drucker über ein serielles Kabel mit dem C128 verbunden sein muß,

mit einem Parallelanschluß über den Userport können Sie mit diesem Programm nichts ausdrucken.

L Löschen einer Datei

Nach einer Sicherheitsabfrage wird die ganze sich im Speicher befindliche Datei gelöscht.

V Verändern/Editieren

Oft ist es notwendig, Einträge eines Datenfeldes zu ändern. Geben Sie bitte zuerst die Nummer der Datenzeile ein (Sie sehen sie beim Auflisten der Datei auf dem Bildschirm, wie unter Menüpunkt D beschrieben), dann wird diese Zeile auf dem Eingabefeld gelistet, wobei sich der Cursor am Zeilenanfang befindet. So können Sie nun dieses Datenfeld verändern. Nach Druck auf die RETURN-Taste wird diese so geänderte Zeile übernommen und ersetzt das bisherige, unter dieser Nummer geführte Datenfeld.

X Liste nach Suchbegriff

Damit kommen wir zum Herzstück dieses Dateiprogrammes. Aus einer praktisch unsortierten Datei können Sie damit alle Datenfelder (=Zeilen) ausfiltern, die den von Ihnen festgelegten Suchbegriff aufweisen. Wo der stehen hat (ab welcher Spaltenposition), haben wir bereits erwähnt. Nach der Eingabe dieses Suchkennzeichens (das Programm ist auf vier Zeichen eingerichtet), dauert es ein wenig, dann werden die Datenzeilen in der Reihen-

folge, wie sie in der Gesamtdatei gefunden wurden, auf dem Bildschirm gelistet. Nach Druck auf irgendeine Taste erscheint wieder ein Fenster, das Ihnen folgende Weiterverarbeitungsmöglichkeit bietet:

1. Diese ausgefilterte Datei als Neu-File auf Diskette abzulegen (mit dem Dateinamen, den Sie als Suchbegriff eingegeben haben)
- oder
2. sie an eine bereits bestehende Filter-Datei auf Diskette anzufügen.

Damit ist es unter anderem möglich, aus zwei oder beliebig mehr Gesamtdateien solche herausgefilterten Suchdatensätze zu bilden und diese wiederum zu einer Gesamtdatei zu verbinden (die dann nur Datenfelder aufweist, die alle dasselbe Suchmerkmal aufweisen, also im Prinzip eine andere Art von Sortierung). Außerdem läßt sich diese spezielle Suchdatei, die Sie gerade auf dem Bildschirm aufgelistet bekommen haben, auch ausdrucken. Wenn Sie aber wieder ins Menü zurückkehren, werden Sie feststellen, daß die vorher geladene Gesamtdatei nach wie vor so im Speicher steht und von der gefundenen Suchdatei in keiner Weise verändert worden ist. (Die haben Sie ja hoffentlich bereits auf Diskette gespeichert, wenn nicht, müssen Sie eben wieder von vorne zu suchen anfangen . . .)

Die beiden abschließenden Menüpunkte C Catalog/Directory und Ende bedürfen wohl keiner ausführlichen Beschreibung. Bei der Arbeit mit der Suchdatei 128 sollten Sie immer den Speicherplatz vor Augen haben (es sind 700 bis 750 Datenfelder, also Zeilen, pro Datensatz möglich). Die entsprechend dafür eingesetzten Variablen wurden mit der DIM-Anweisung auf einen Wert von 750 festgelegt. Wir glauben, daß dies für

VARIABLE IN DIE BANK 1 DES 128PC

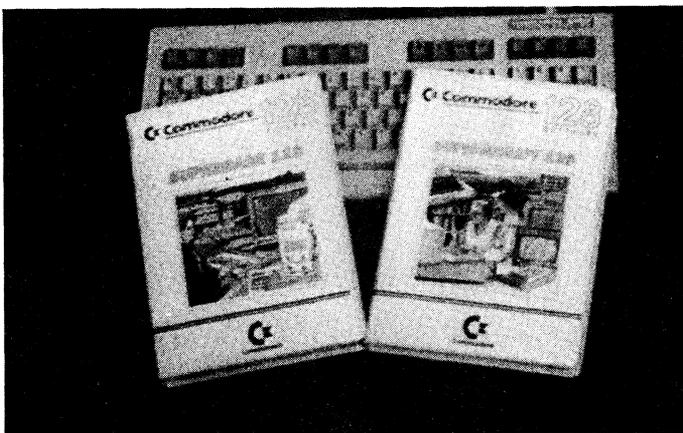
einen Datensatz ausreichend ist. Bedenken Sie aber bitte, daß alle diese Variablen in BANK 1 des C128 abgelegt werden, von den Variablen, die das Programm selbst benutzt, gar nicht zu reden. Da kann bei recht großen Dateien ein Speicher von 64 KByte doch auch einmal knapp werden, vor allem, wenn Sie die Suchfunktion aufrufen. Jedes gefundene Datenfeld muß dann nämlich in einer anderen indizierten Variablen (S1\$) festgehalten werden, was natürlich wieder Speicherplatz verbraucht. Andererseits wird kaum jedes von Ihnen eingegebene Datenfeld genau achtzig Zeichen lang sein; jedes kürzere wirkt sich platzsparend aus. Sie werden sicher selbst aufgrund Ihrer individuellen Bedürfnisse dahinterkommen, wann es ratsam ist, eine neue Datei oder einen neuen Gesamtdatensatz anzulegen. B.U. □

Nutzen Sie
unseren Klein-
Anzeigen-Service.
Kostenlos für
Privat-Anbieter!

Was Textverarbeitungen wirklich leisten können

Vergleiche, Referenzlisten, Konvertierungstabellen, Tips und Tricks zu den Textverarbeitungsprogrammen Easyscript, Scriptplus und Superscript der Commodore-Rechner C64/Plus4/128PC, die nicht oder unvollständig in den Handbüchern stehen. Konvertierung der Textfiles von Rechner zu Rechner und Umarbeitung der Druckersteuerzeichen und Defaultslisten.

Commodore bietet in der Regel selbst ein Textverarbeitungsprogramm für seine Rechner an, so auch geschehen bei C64, Plus4 und 128PC. Wie die Programme auch genannt werden, ob Easyscript, Scriptplus oder Superscript, alle kommen aus dem Software-Haus Precision Software in Surrey, England, das für einige professionelle Programmentwicklungen berühmt geworden ist. Simon Tranmer, der Programmierer der Script-Serie, hat auch Superscript II für die Commodore-Rechner 2000, 3000, 4000, 8000, 600 und 700 geschrieben. In Kooperation mit Tom Cranstoun zeichnet er auch für Superbase Database verantwortlich, das auf C64, Plus4, 8096, 8296, 700 und Apple II läuft. Bei den Commodore-Rechnern ist aufgrund der unterschiedlichen Leistungsfähigkeiten im Hardwarebereich sowie bei der softwarespezifischen Entwicklung auch einiges in der Bedienung/Anwendung und im Ergebnis verschieden. Probleme der Konvertierung aus sequentiellen Dateien aller drei Textverarbeitungsprogramme können nur in Zusammenhang mit Sonderzeichen oder Drucker-Steuerzeichen entstehen, da deren Codierung intern unterschiedlich ist. Eine Anpassung ist ohne größere Schwierigkeiten möglich. Wenn Sie aber sequentielle Dateien mit zu großer Zeichenanzahl in einen kleineren Zeichenspeicher laden wollen, geht der



(1) Superscript und Superbase für den 128PC

Rest, der nicht mehr in den Zeichenspeicher hineinpaßt, verloren. Hier nun ein Speicherplatz-Vergleich aller drei Script-Programme, relativ zur Rechnerkonfiguration.

Mögliche Speichergrößen der Commodore Textverarbeitungsprogramme
EASYSRIPT - SCRIPTPLUS - SUPERSRIPT

TABELLE 1:

SUPERSCRIPT 128 PC	40/80 Display mit Bank 1
*40 Zeichen x 999 Zeilen	= 39980 Zeichen im Speicher
80 Zeichen x 726 Zeilen	= 58080 Zeichen im Speicher
120 Zeichen x 484 Zeilen	= 58080 Zeichen im Speicher
160 Zeichen x 363 Zeilen	= 58080 Zeichen im Speicher
200 Zeichen x 290 Zeilen	= 58080 Zeichen im Speicher
240 Zeichen x 242 Zeilen	= 58080 Zeichen im Speicher

* Nur in 40 Zeichen Modus möglich.

TABELLE 2:

SUPERSCRIPT 128 PC	40/80 Display mit Bank 2
*40 Zeichen x 432 Zeilen	= 17280 Zeichen im Speicher
80 Zeichen x 216 Zeilen	= 17280 Zeichen im Speicher
120 Zeichen x 144 Zeilen	= 17280 Zeichen im Speicher
160 Zeichen x 108 Zeilen	= 17280 Zeichen im Speicher
200 Zeichen x 86 Zeilen	= 17280 Zeichen im Speicher
240 Zeichen x 72 Zeilen	= 17280 Zeichen im Speicher

* Nur in 40 Zeichen Modus möglich.

TABELLE 3:

SCRIPT/PLUS C16/C116	40 Display mit BANK 1 mit mindestens 16 KByte
40 Zeichen x 91 Zeilen	= 3640 Zeichen im Speicher
80 Zeichen x 45 Zeilen	= 3600 Zeichen im Speicher
120 Zeichen x 30 Zeilen	= 3600 Zeichen im Speicher
160 Zeichen x 22 Zeilen	= 3520 Zeichen im Speicher
200 Zeichen x 18 Zeilen	= 3600 Zeichen im Speicher
240 Zeichen x 15 Zeilen	= 3600 Zeichen im Speicher

Eine ordentliche Textverarbeitung mit dieser Speichergröße ist nicht ratsam, kann aber durch linken (anhängen) verschiedener Textteile beim Ausdruck wieder teilweise kompensiert werden.

TABELLE 4:

SCRIPT/PLUS C16/C116/Plus4	40 Display mit BANK 1 mit mindestens 32 KByte
40 Zeichen x 501 Zeilen	= 20040 Zeichen im Speicher
80 Zeichen x 250 Zeilen	= 20000 Zeichen im Speicher
120 Zeichen x 167 Zeilen	= 20040 Zeichen im Speicher
160 Zeichen x 125 Zeilen	= 20000 Zeichen im Speicher
200 Zeichen x 100 Zeilen	= 20000 Zeichen im Speicher
240 Zeichen x 83 Zeilen	= 20040 Zeichen im Speicher

Ein Umschalten auf Bank 2 ist mit dieser Speichergröße nicht möglich.

SERVICE

TABELLE 5:

SCRIPT/PLUS C16/C116/Plus4 40 Display mit BANK 1/2 mit mindestens 64 KByte

40 Zeichen x 501 Zeilen	=	20040 Zeichen x 2	=	40080 Zeichen im Speicher
80 Zeichen x 250 Zeilen	=	20000 Zeichen x 2	=	40000 Zeichen im Speicher
120 Zeichen x 167 Zeilen	=	20040 Zeichen x 2	=	40080 Zeichen im Speicher
160 Zeichen x 125 Zeilen	=	20000 Zeichen x 2	=	40000 Zeichen im Speicher
200 Zeichen x 100 Zeilen	=	20000 Zeichen x 2	=	40000 Zeichen im Speicher
240 Zeichen x 83 Zeilen	=	20040 Zeichen x 2	=	40080 Zeichen im Speicher

TABELLE 6:

EASYSRIPT C64 40 Display

40 Zeichen x 764 Zeilen	=	30560 Zeichen im Speicher
80 Zeichen x 382 Zeilen	=	30560 Zeichen im Speicher
120 Zeichen x 254 Zeilen	=	30480 Zeichen im Speicher
160 Zeichen x 191 Zeilen	=	30560 Zeichen im Speicher
200 Zeichen x 152 Zeilen	=	30400 Zeichen im Speicher
240 Zeichen x 127 Zeilen	=	30480 Zeichen im Speicher

TABELLE 7:

Plotterausdruck
Tortengrafik Speicherplatz Vergleich in Prozent

SUPERSCRIPT	Bank 1 und 2
SUPERSCRIPT	Bank 1
EASYSRIPT	Bank 1
SCRIPTPLUS 64K	Bank 1 und 2
SCRIPTPLUS 32K	Bank 1
SCRIPTPLUS 16K	Bank 1

TABELLE 8:

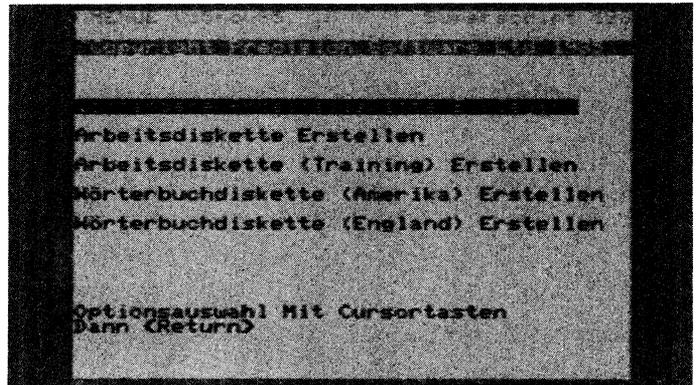
Plotterausdruck:
Balkendiagramm Speicherplatz Vergleich Zeichen ./, Zeilen

SUPERSCRIPT	Bank 1 und 2
SUPERSCRIPT	Bank 1
EASYSRIPT	Bank 1
SCRIPTPLUS 64K	Bank 1 und 2
SCRIPTPLUS 32K	Bank 1
SCRIPTPLUS 16K	Bank 1

Textverarbeitungsprogramme können nur so gut genutzt werden, wie auch der Anwender sie immer einsetzen will, wie dies der Computer, die Peripheriegeräte, das Programm und der Benutzer zulassen. Daß die Handbücher zu den Programmen ein zweites Mal gelesen oder erst gar nicht aus der Hand gelegt werden sollten, ist nicht nur den Einsteigern zu empfehlen.

SUPERSCRIPT

Bei Superscript für den 128PC beträgt der Bildschirm-ausschnitt in der Regel 80 Zeichen x 22 Zeilen = 1760 Zeichen bzw. 40 Zeichen x 22 Zeilen = 880 Zeichen, entsprechend dem gewählten Display-Modus. Größere Zeichenanzahl pro Zeile – 40 bis 240 sind möglich, das wird durch Scrollen des Bildschirms nach rechts oder links, nach unten oder oben bewerkstelligt. Dieser sogenannte Lupeneffekt – Sie sehen immer nur einen bestimmten Teilbereich eines Textes –, bedeutet bei Superscript 1760 Zeichen (80 Display) oder 880 Zeichen (40 Display) – bei 58080 Zeichen bzw. 39980 möglichen Zeichen im Speicher – gegenüber Scriptplus und Easyscript mit 1000 Zeichen



(2) Einschaltmeldung Superscript 40 Zeichen

(40 Display) bei 200040 Zeichen bzw. 30560 möglichen Zeichen im Speicher.
Zum Vergleich: Eine vollgeschriebene DIN A4-Seite mit 1/6-Zoll-Zeilensprung und 10 Zeichen pro Inch hat 70 Zeilen x 80 Zeichen = 5600 Zeichen pro Blatt. Superscript besitzt noch drei Statuszeilen als Kopfzeilen. In denen wird in der ersten Zeile der eingestellte Modus, zum Beispiel: rechts 'Edit W.I.N.M.1' links 'Z: 121 S: 1'. In der zweiten Zeile steht links der aktuelle, geladene Filename höchstens '16 Zeichen lang' und links der Hinweis, daß mit 'Menü: F1' das Hauptmenü aufgerufen werden kann. Die dritte Zeile zeigt ständig die vom Anwender eingestellten Tabulatoren-schritte mit Trennstrichen (/---/---/) an.

Tabelle 9:

EDIT	=	Editor Mode	Eingabemodus
W	=	Wordwrap on/off	Wortumbruch ein/aus
I	=	Insert on/off	Einfügen ein/aus
N	=	Numeric on/off	Zahleneingabe ein von Links/aus von Rechts
M	=	Merge Mode	Fülldateien Mode
1 od 2	=	Bank 1/2	Arbeitsspeicher/Textbereich 1 und 2
Z	=	Zeilennummer	in der sich der Cursor befindet
S	=	Spaltennummer	in der sich der Cursor befindet

Scriptplus und Easyscript haben nur eine Info-Zeile, sie wechseln den ganzen Bildschirm in die entsprechenden Hilfsmenüs. Ein Umschalten in die Bank 2 ist nur noch im Plus4 oder mit erweitertem C16/C116 möglich, der C64 bietet hier mit Easyscript diesen zusätzlichen Arbeitsspeicher nicht an.

Tabelle 10:

Einschaltmeldung SUPERSCRIPT 128 80 Zeichen Bildschirm:

MENUE V15Nov85

Superscript 128

Copyright Precision Software Ltd. 1985

Vorhandene Arbeitsdiskette Einlegen
Arbeitsdiskette(Training) Erstellen
Wörterbuchdiskette (Amerika) Erstellen
Wörterbuchdiskette (England) Erstellen

Optionauswahl Mit Cursortasten Dann (Return)

SERVICE

Ladeanzeige für Defaulttabelle und Beispiele der Belegung der Stop- bzw. Escape-Taste plus Befehlstaste (0-9 a-z A-Z), die nach Wunsch verändert werden können. Folgende Kennzeichnung ist zu beachten:
 /= Befehlsanfang F1, ^= Taste Control, ^m = Return.

Tabelle 11:

Anzeige	W. . . . l.		S:1 Kl:	
#ci10:	F1/L/E/C I	ci10 =	CHARACTER INCH	Zeichen pro Inch
#cm:		cm =	COMMENT	Kommentarzeile
#cn:	F1/L/S/C E	cn =	CENTER NO	Zentrieren nein
#cy:	F1/L/S/C S	cy =	CENTER YES	Zentrieren ja
#bm6:	F1/L/M/B	bm6 =	BOTTOM MARGINS	Untere Textgrenze
#eo0:	F1/L/M/E	eo0 =	EVEN ODD	Gerade/Ungerade aus
#em:	F1/D/M/C	em =	CONDITIONAL MERGE	Platzhalter B-Länge
#fe:	F1/L/E/F	fe =	ENTER FEATURE	Druckerbefehle send
#fn:	F1/L/S/F E	fn =	FOOT ALT	Fuß wechseln N
#fp0:	direkt	fp =	FORM FEED PAGE	Neue Seite
#fs:	direkt	fs =	FOOT ALT	2. Fußzeile
#ft:	F1/L/B/F	ft =	FOOT BORDER	1. Fußzeile
#fy:	F1/L/S/F S	fy =	FOOT ALT	Fuß wechseln J
#hd:	F1/L/B/H	hd =	HEAD BORDER	1. Kopfzeile
#hl:	F1/L/B/L	hl =	HEAD LEFT	Kopf linker Rand
#hn:	F1/L/S/H E	hn =	HEAD ALT	Kopf wechseln J
#hr:	F1/L/B/R	hr =	HEAD RIGHT	Kopf rechter Rand
#hs:	direkt	hs =	HEAD ALT	2. Kopfzeile
#hy:	F1/L/S/H S	hy =	HEAD ALT	Kopf wechseln N
#in:	F1/L/M/I	in =	INDENT MARGINS	Text einrücken
#jn:	F1/L/S/J E	jn =	JUSTIFY NO	Justieren nein
#jy:	F1/L/S/J S	jy =	JUSTIFY YES	Justieren ja
#li6:	F1/L/E/L	li6 =	LINES INCH	6 Zeilen pro Inch
#lk:	F1/L/L	lk =	LAYOUT LINK	Files verketteten
#lm1:	F1/L/M/L	lm1 =	LEFT MARGINS	Linker Rand 1
#nm:	F1/D/M/N	nm =	NAME MERGE	Fülldatei Namen
#np:	F1/L/S/N	np =	NEW PAGE	Umbruch Neue Seite
#of0:	F1/L/M/O	of0 =	OFFSET	Druckeroffset 0
#pd:		pd =	PRINTER DEFINATION	Druckerdefinition
#pl72:	F1/L/M/P	pl72 =	PAGE LENGHT	Papierlänge 72
#pn22:	F1/L/E/P	pn =	PAGE NUMBER	Seite Nummer 22
#ps:	F1/L/P	ps =	LAYOUT PAUSE	Pause Druckerausgabe
#rn:	F1/L/S/R E	rn =	JUSTIFY RIGHT N	Rechtsbündig nein
#rm00:	F1/L/M/R	rm00 =	RIGHT MARGINS	Rechter Rand 00
#ry:	F1/L/S/R S	ry =	JUSTIFY RIGHT Y	Rechtsbündig ja
#sc	,x,x,x,x,x,x,x	sc =	SET COLOR	Bildschirmfarben
#sk:	F1/L/E/S	sk =	SPACE	Leerzeilen
#sp0:	F1/L/S/S	sp0 =	SPACING	Zeilenvorschub 0/1/2
#tm0:	F1/L/M/T	tm0 =	TOP MARGINS	Obere Textgrenze
#0=dd		ESC 0 =	DOKUMENT DIRECTORY	
#1=ssf		ESC 1 =	SET SEARCH FILE	
#2=ssc		ESC 2 =	SET SEARCH CONTINUE	
#3=gd		ESC 3 =	GO DOWN	
#4=gu		ESC 4 =	GO UP	
#5=/fbs		ESC 5 =	FEATURE BOLD START	
#6=/fbe		ESC 6 =	FEATURE BOLD END	
#7=/fus		ESC 7 =	FEATURE UNDERLINE START	
#8=/fue		ESC 8 =	FEATURE UNDERLINE END	
#9=/duo		ESC 9 =	DOCUMENT UTILITY OTHER	
#a=/el		ESC a =	ERASE LINE	
#b=/ab		ESC b =	AERA BLOCK	
#i=/ai		ESC i =	AERA INSERT	
#m=/am		ESC m =	AERA MOVE	
#r=/ar		ESC r =	AERA REPEAT	
#lk:drucker def		lk =	LINK DOKUMENT	Verbinden Files

Wie Sie sehen, ist je nach Geschmack die Defaultliste (Computer-Grundeinstellung) den eigenen Wünschen und Bedürfnissen auf einfache Weise anzupassen und beliebig zu erweitern. Die in unserer Liste aufgeführten Befehlsketten sind nicht statisch, sondern vom Redakteur willkürlich ausgewählt worden, sie sollten

nach Häufigkeit des Bedarfs belegt werden. Mit der letzten Zeile der Defaultliste (Rechner-Grundeinstellung) *lk:drucker def wird die ausgewählte Drucker Definationssequenzen nachgeladen.

Tabelle 12:

#cmdrucker def		Zeile
#sk5;	Ihr Drucker ist definiert als :#sk2	
Mannesmann Tally 85 mit Centronics-Interface#sk2		
Ausgabe über Userport#sk2		
Weiter mit Leertaste#sk9		
#pd		
00:	Textweite für Bildschirm / Ändern mit Taste F1/S/T	1
0:	(0) für Centronics/ (2) für RS232/ (4) für Seriell/ Drucker Nummer	2
255:	(0/3/5/6/7/8/9/10) Sekundäradresse für Normaldruck/ (255) = keine	3
5:	Daten und Stop Bits für RS232	4
6:	Baud-Rate für RS232	5
1:	Parität für RS232	6
1:	(1) wenn Zeilenvorschub erwartet/ (0) wenn nicht / LINE FEED	7
0:	(1) wenn CBM Codes erwartet/ (0) wenn nicht	8
0:	(1) wenn 'cursor down mode'/ (0) wenn nicht	9
0:	(1) wenn Diablo Codes/ (0) wenn nicht	10
0:	(1) wenn Spinwriter Codes/ (0) wenn andere	11
1:	(1) wenn Weitschrift an/ (0) wenn nicht	12
0:	(1) wenn Drucker Fettschriftfähig durch Backspace (0) wenn nicht	13
0:	(1) wenn Doppeldruckfähig mit 1/120 Inch-Schritten (0) wenn nicht	14
0:	(1) wenn Drucker Unterstreichfähig durch Backspace (0) wenn nicht	15
1:	(x) Anzahl der Überschreibungen im Fettdruck	16
0:	(1) Wenn unten beschriebene Funktionen in CBM Code (0) wenn nicht	17
27,45,1:	Unterstreichen EIN (Underline)	18
27,69:	Fettdruck EIN (Enhanced)	19
27,71:	Doppeldruck EIN (Double Strike)	20
15:	Engschrift EIN (Compressed)	21
14:	Breitschrift EIN (Enlarged)	22
27,83,0:	Hochstellen EIN (Superscript)	23
27,83,1:	Tiefstellen EIN (Subscript)	24
27,45,0:	Unterstreichen AUS (Underline)	25
27,70:	Fettdruck AUS (Enhanced)	26
27,72:	Doppeldruck AUS (Double Strike)	27
18:	Engschrift AUS (Compressed)	28
20:	Breitschrift AUS (Enlarged)	29
27,84:	Hochstellen AUS (Superscript) mit CR	30
27,84:	Tiefstellen AUS (Subscript) ohne CR	31
27,82,0:	Sonderwunsch 1 (USA Zeichensatz)	32
27,82,1:	Sonderwunsch 2 (F Zeichensatz)	33
27,82,2:	Sonderwunsch 3 (D Zeichensatz)	34
27,82,3:	Sonderwunsch 4 (GB Zeichensatz)	35
27,77:	Sonderwunsch 5 (ELITE)	36
27,80:	Sonderwunsch 6 (PICA)	37
27,52:	Sonderwunsch 7 (Italic kursiv ein)	38
27,53:	Sonderwunsch 8 (Italic kursiv aus)	39
27,112,1:	Sonderwunsch 9 (Proportionalschrift ein)	40
27,112,0:	Sonderwunsch 0 (Proportionalschrift aus)	41
:	setze 0 Zeichen/Inch	42
:	setze 10 Zeichen/Inch PICA-Schrift	43
:	setze 12 Zeichen/Inch ELITE-Schrift	44
:	setze 15 Zeichen/Inch	45
:	setze 20 Zeichen/Inch	46
:	setze 4 Zeilen/Inch	47
27,50:	setze 6 Zeilen/Inch	48
27,48:	setze 8 Zeilen/Inch	49
27,49:	setze 12 Zeilen/Inch	50
:	Druckerinitialisierungssequenz ++++++	51
35:	Nummernzeichen #	52
36:	Dollar \$	53
64:	Paragraph § at sign @	54
91:	Klammer [55
35:	Pfund £	56
93:	Klammer]	57
94:		58
123:		59
124:		60

125: ä
 126: ö
 16,123: ã
 17,124: ö
 18,125: ü
 19,126: ß
 22,91: x
 23,92: ö
 24,93: ü

61
 62
 63
 64
 65
 66
 67
 68
 69

39 neue Befehle für Ihren C16/P4

BASIC-Tool erweitert den Befehlssatz Ihres C16/C116/Plus4 um 39 nützliche Tools, die mit SYS-Sequenzen aufzurufen sind. Den Überblick der Befehle verschafft Ihnen die nachfolgende Übersicht. Am Anschluß daran finden Sie weitere Tips zum Umgang mit dem Programm.

BEFEHLSÜBERSICHT:

Befehl: Old
Syntax: SYS 1630
Wirkung: Retten eines mit New oder Reset gelöschten Programmes.

Befehl: Kaltstart
Syntax: SYS 1642
Wirkung: Einschaltmeldung: BASIC-Speicher wird gelöscht

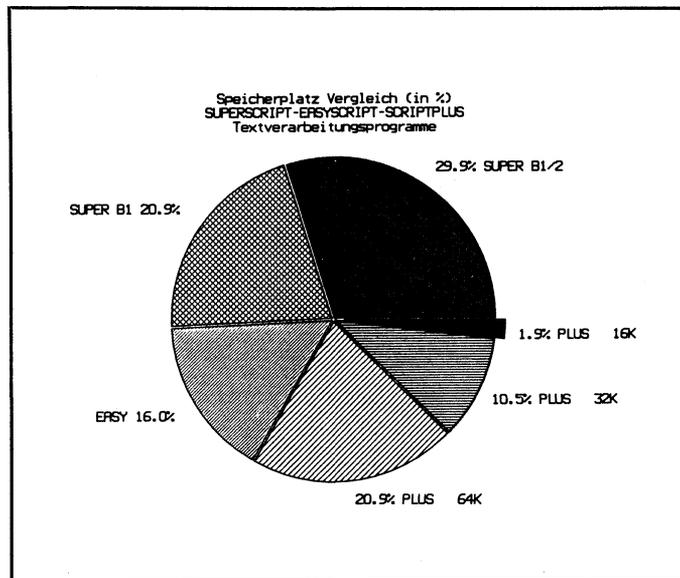
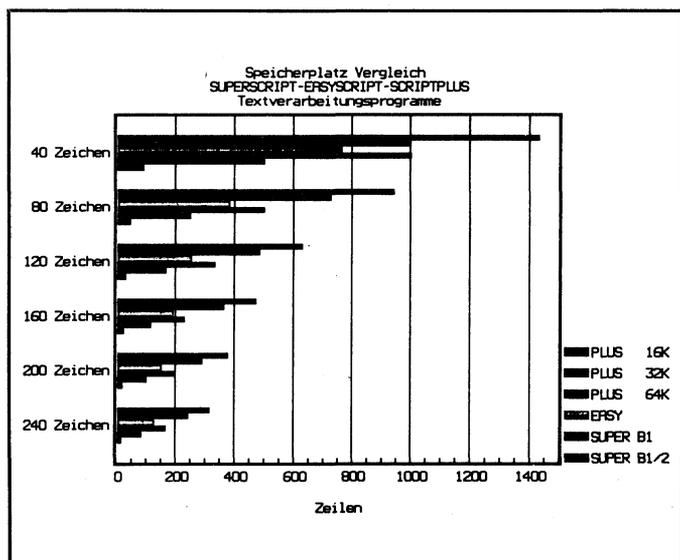
Befehl: Reset
Syntax: SYS 1646
Wirkung: Löst einen Reset aus

Befehl: Cursor
Syntax: SYS 1650,X,Y
Wirkung: Positioniert den Cursor an der Position X, Y
Variablen: X=Spalte (0-39)
 Y=Zeile (0-24)

Befehl: Window
Syntax: SYS 1674,L,O,U,R
Wirkung: Definiert ein Bildschirmfenster mit den Koordinaten L, O, U, R
Variablen: L=linker Rand
 O=oberer Rand
 U=unterer Rand
 R=rechter Rand

Befehl: Musik
Syntax: SYS 1716, L, TG, T, OK, TL
Wirkung: Spielt den Ton T der Oktave OK mit der Lautstärke L des Tongenerators TG TL-Takteinheiten lang
Variablen: L=Lautstärke (0-8)
 TG=Tongenerator (0-2)
 T=Tonhöhe (0-255)
 OK=Oktave (0-3)
 TL=Tonlänge (0-255)

Befehl: Merge
Syntax: SYS 4099/Load/SYS4126
Wirkung: Lädt ein Programm nach, ohne das alte, im Speicher stehende Programm zu überschreiben. Nach dem Mergen mit SYS 4126 empfiehlt sich, zu renumbern



Auf der mitgelieferten Diskette werden einige Druckerdefinitionstabellen bereitgestellt, aber diese genügen nicht, die Druckervielfalt, die an den Commodore-Rechnern angeschlossen werden, abzudecken. Die hier abgedruckte Defaultliste ist für den Mannesmann Tally 85 mit Centronics-Interface über Userportansteuerung geeignet. Wenn Sie einen Drucker besitzen, der in der folgenden Liste nicht enthalten ist, müssen Sie eine entsprechende Defaultliste selbst anfertigen. Mit dem Druckerhandbuch und einer Defaultliste, die Ihrem Drucker am nächsten kommt, haben Sie die besten Voraussetzungen. □
 (Fortsetzung folgt)

Robert Wagner

SERVICE

Befehl: BG-Color
Syntax: SYS4137 (ein)/SYS4158 (aus)
Wirkung: Bei jedem Interrupt wird die Rahmenfarbe um eins erhöht

Befehl: HG-Color
Syntax: SYS4171 (ein)/SYS4158 (aus)
Wirkung: Siehe BG-Color, nur daß anstelle der Rahmenfarbe die Hintergrundfarbe verändert wird

Befehl: Beep
Syntax: SYS4191 (ein)/SYS4158 (aus)
Wirkung: Bei jedem Tastendruck wird ein Pieps verursacht. Dadurch ist Blindschreiben möglich

Befehl: B.T.A.
Syntax: SYS4232
Wirkung: BASIC-Tool wird nach einem Reset wieder aktiviert und der BASIC-Start verschoben (auf \$1400)

Befehl: Turbo-Mode
Syntax: SYS4294 (ein)/SYS4303 (aus)
Wirkung: Schaltet den Bildschirm ab. Dadurch wird der Computer um etwa 30 Prozent schneller. Dies ist besonders bei zeitraubenden Berechnungen sehr nützlich

Befehl: E-C-Mode
Syntax: SYS4312 (ein)/SYS4321 (aus)
Wirkung: Dieser Befehl erlaubt, jedem Zeichen vier Hintergrundfarben zu geben. Es können allerdings nur noch die Bildschirmcodes bis einschließlich 63 dargestellt werden:
 A=A mit Hintergrundfarbe 0
 S+A=A mit Hintergrundfarbe 1
 R+A=A mit Hintergrundfarbe 2
 R+S+A=A mit Hintergrundfarbe 3

Befehl: 38 Spalten
Syntax: SYS4330
Wirkung: Der Bildschirm wird auf 38 Spalten verkleinert. Mit diesem Befehl lassen sich schöne Smooth-Scrolling-Effekte erzielen

Befehl: 40 Spalten
Syntax: SYS4339
Wirkung: Der Bildschirm wird wieder auf den Normalzustand gestellt

Befehl: 24 Zeilen
Syntax: SYS4348
Wirkung: Der Bildschirm wird auf 24 Zeilen verkleinert. Für diesen Befehl gilt das für den Befehl 38 Spalten gesagte

Befehl: 25 Zeilen
Syntax: SYS4357
Wirkung: Der Bildschirm wird wieder auf Normalzustand geschaltet

Befehl: Space-Zeile
Syntax: SYS4366,Y
Wirkung: Löscht die Zeile Y.
Variable: Y=Zeile (0-24)

Befehl: Clear Window
Syntax: SYS4375
Wirkung: Hebt ein zuvor definiertes Bildschirmfenster auf

Befehl: List Error
Syntax: SYS4383 (ein)/SYS4394 (aus)
Wirkung: Bei Eingabe von List+Return wird ein Syntax-Error ausgegeben. Das im Speicher vorhandene Programm läßt sich somit nicht mehr listen. Dies ist besonders für Adventures geeignet

Befehl: M-C-T-Mode
Syntax: SYS4405 (ein)/SYS4414 (aus)
Wirkung: Jedes Zeichen erhält drei Zeichenfarben, dadurch wird es (manchmal) sehr entfremdet. Es bekommt aber eine 3-D-Ansicht

Achtung: Es funktioniert nur, wenn die Zeichenfarben 9 bis 16 gewählt werden

Befehl: Save Error
Syntax: SYS4423 (ein)/SYS4434 (aus)
Wirkung: Save+Return ergibt einen Syntax-Error, so daß das Programm nicht gespeichert werden kann

Befehl: Swap HC/RC
Syntax: SYS4445
Wirkung: Vertauscht Hintergrund- und Rahmenfarbe

Befehl: Out
Syntax: SYS4462
Wirkung: BASIC-Tool wird ausgeschaltet und total gelöscht. Vorsicht!

Befehl: Softrevers
Syntax: SYS4510
Wirkung: Die Reverszeichen werden softwaremäßig erzeugt. Dadurch können 256 Zeichen neu definiert werden

Befehl: Hardrevers
Syntax: SYS4519
Wirkung: Die Reverszeichen werden hardwaremäßig erzeugt. Mit diesem Befehl wird Softrevers aufgehoben

Befehl: Kill
Syntax: SYS4528
Wirkung: Ein BASIC-Programm wird total gelöscht. Selbst Old kann hier nicht mehr helfen

Befehl: Dreieck (HGR)
Syntax: SYS4545, F, X1, Y1, X2, Y2, X3, Y3
Wirkung: Zeichnet ein Dreieck in hochauflösender Grafik

Variablen: X1, Y1 = Eckpunkt 1
 X2, Y2 = Eckpunkt 2
 X3, Y3 = Eckpunkt 3
 F=Farbe -1 + Helligkeit * 16

Befehl: M-C-T-Color
Syntax: SYS4672, MF1, MF2
Wirkung: Stellt die Zusatzfarben 1 + 2 für den M-C-T-Mode. Berechnet wird die Farbe wie beim Befehl Dreieck

SERVICE

Befehl: E-C-Color
Syntax: SYS4701, H1, H2, H3
Wirkung: Bestimmt die Hintergrundfarben 1 bis 3 für den E-C-Mode. Die Berechnung der Farben erfolgt nach dem Schema der Dreieck-Farbe

Befehl: Savehgr
Syntax: SYS4743
Wirkung: Die hochauflösende Grafik wird auf Kasette abgespeichert und kann durch Loadhgr wieder geladen werden

Befehl: Loadhgr
Syntax: SYS4778
Wirkung: Eine auf Kasette abgespeicherte Grafik wird geladen

Befehl: Soff
Syntax: SYS4803, ST
Wirkung: Schaltet die Stimme ST ab, ohne die anderen Stimmen zu beeinflussen
Variable: ST=Stimme (1 bis 3)

Befehl: Son
Syntax: SYS4842, ST
Wirkung: Schaltet die Stimme ST an
Variable: ST=Stimme (1 bis 3)

Befehl: Zcopy
Syntax: SYS4881
Wirkung: Kopiert den Zeichensatz (\$D000-\$D7FF) ins RAM (ab \$3800). Außerdem wird das BASIC-Ende auf \$37FF herabgesetzt, und somit der neue Zeichensatz vor dem Überschreiben durch Variablen geschützt

Befehl: Zon
Syntax: SYS4945
Wirkung: Schaltet den neuen Zeichensatz an, der ab \$3800 liegt

Befehl: D.Z.
Syntax: SYS4956/SYS5080
Wirkung: Der deutsche Zeichensatz mit den Zeichen Ä, Ö, Ü, ß wird erzeugt
Tasten: SHIFT+A = Ä
 SHIFT+O = Ö
 SHIFT+U = Ü
 SHIFT+S = ß

Befehl: Z.D.
Syntax: SYS4473, AD, A, B, C, D, E, F, G, H
Wirkung: Mit diesem Befehl können eigene Zeichen definiert werden
Variablen: AD = Adresse = 8*[BC]+14336
 A-H = Bytes (0-255)
 [BC] = Pokecode

Befehl: B.V.
Syntax: SYS4501,X
Wirkung: Dieser Befehl verknüpft den Bildschirminhalt mit X(EOR)
Variable: X (0-255)

TIPS ZUM UMGANG MIT BASIC-TOOL

1. Programmteil 1 (BASIC-Tool-Lader) wird je nach

Speichermedium mit (D)SAVE "BASIC-TOOLLOADER" abgespeichert. Nach dem Start mit Run lädt dieses automatisch den Hauptteil nach und startet ihn.

2. Die Programme sind vor dem ersten Laden unbedingt zu save. Damit der Programmname des Hauptteiles mit der im BASIC-Tool-Lader angegebenen Ladeanweisung übereinstimmt, ist dieser mit (D)SAVE "BASIC-TOOL" abzuspeichern.

3. Für Diskettenbetrieb ändern Sie bitte folgende Zeilen ab:

```
654 DATA A9,02,A2,08,A0,02,20,BA
676 DATA A9,02,A2,08,A0,00,20,BA
```

4. Probieren Sie das Demoprogramm BASIC-Tool Demo aus, nachdem Sie vorher natürlich BASIC-Tool geladen und gestartet haben. □

BASIC-TOOL DEMO

```
10 rem basic-tool demo=====c16 <ol>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by michael inden <gh>
50 rem <pd>
60 rem <ah>
70 rem basic v3.5 + basic-team <md>
80 rem c16/116/plus4 <ki>
90 rem ===== <jg>
100 clr:gosub430:printc1$c4$c4$"
    demo-menue" <kp>
110 printc4$c4$" 1 = dreiecke zeichnen" <hk>
120 printc4$" 2 = neue zeichen definieren" <mm>
130 printc4$" 3 = deutscher Zeichensatz" <fd>
140 printc4$" 4 = bildverknuepfung + window" <lo>
150 printc4$" 5 = swap" <in>
160 printc4$" 6 = ende" <pe>
170 geta$:a=val(a$):ifa<1ora>6then goto170 <ih>
180 onagoto190,230,310,340,400,420 <oa>
190 rem dreiecke <mk>
200 graphic1,1:forx=100to170step2 <on>
210 sys4545,1,100,10,x,x,x+20,x:next <hd>
220 getkeyk$:graphicclr:goto100 <fe>
230 rem n. zeichen <pa>
240 printc1$:sys4881:sys4945 <dk>
250 sys4473,14344,4,14,14,14,31,159,159,255 <fl>
260 sys4473,14352,0,0,0,0,0,2,134,255 <fh>
270 sys4473,14360,255,159,159,31,14,14,14,4 <ec>
280 sys4473,14368,255,134,2,0,0,0,0,0 <gj>
```

LISTING

```

290 print"ab"c4$;c1$c1$"cd" <pb>
300 getkeyk$:sys5080:getkeyk$:goto
100 <ip>
310 rem deut. zeichensatz <ng>
320 sys4956:printcl$c4$c4$" eine A
hnliche UberprUfung muS erOrtert
werden !" <aa>
330 getkeyk$:sys5080:getkeyk$:goto
100 <mn>
340 rem verknuepf. + window <kd>
350 printcl$:sys1674,S,S,20,20 <el>
360 fory=5to20:forx=5to20 <oh>
370 char1,x,y,"*":nextx,y:sys4501,
54 <ce>
380 getkeyk$:sys4501,54 <mk>
390 getkeyk$:goto100 <kj>
400 rem swap <pf>
410 sys4445:getkeyk$:sys4445:getke
yk$:goto100 <ni>
420 end <ml>
430 rem nachspann ===== <nm>
440 rem * farbcodes/steuer codes * <mb>
450 c4$=chr$(017):cl$=chr$(147) <ll>
460 c1$=chr$(157) <ho>
470 return <dn>
480 rem =====e=n=d=e===== <fc>

```

BASIC-TOOL-LOADER

```

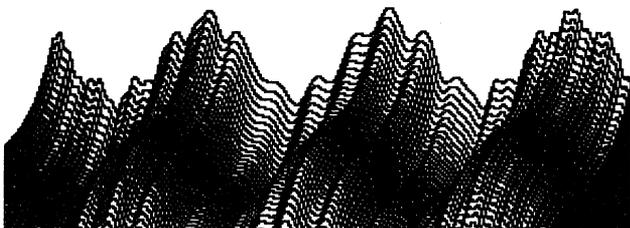
100 rem basic-toolloader=====c16 <eb>
110 rem by michael inden <mm>
120 rem ===== <gn>
130 gosub200 <on>
140 printcl$dn$dn$"poke44,20:poke4
3,1:poke5120,0:new"dn$dn$ <ej>
150 print"load"+chr$(34)+"basic-to
ol"+chr$(34)+"", "peek(174)dn$dn$dn$
dn$dn$ <gh>
160 print"run"ho$; <la>
170 fort=0to2:poke1319+t,13:next:p
oke239,3 <ef>
180 end <oj>
190 : <lh>
200 rem nachspann <ig>
210 rem * steuer codes * <ai>
220 cl$=chr$(147):ho$=chr$(019) <me>
230 dn$=chr$(017) <od>
240 return <gp>
250 rem =====e=n=d=e===== <be>

```

```

100 rem basic-tool=====c16 <on>
102 rem (p) commodore-welt-team <ji>
104 rem ===== <of>
106 rem (c) by michael inden <eo>
108 rem <gh>
110 rem <gl>
112 rem basic v3.5+mc <cf>
114 rem c16/c116/plus 4 <lj>
116 rem ===== <al>
118 : <cg>
120 gosub60000 <nf>
122 color1,2:color0,1:printh$h$c1$ <bj>
124 : <dc>
126 rem datas fuer befehle <pg>
128 : <dl>
130 fort=1630to1771:readba$ <eg>
132 b=dec(ba$):poket,b:next <lm>
134 : <eh>
136 dataa9,14,85,2c,ea,20,18,88 <gd>
138 data4c,4c,88,ea,4c,19,80,ea <ma>
140 data4c,f9,ff,ea,20,91,94,20 <lk>
142 data84,9d,8a,48,20,91,94,20 <kp>
144 data84,9d,68,a8,18,4c,f0,ff <en>
146 dataea,ea,ea,ea,ea,20,91,94 <np>
148 data20,84,9d,8e,e6,07,20,91 <kh>
150 data94,20,84,9d,8e,e7,07,20 <aj>
152 data91,94,20,84,9d,8e,e5,07 <ph>
154 data20,91,94,20,84,9d,8e,e8 <fe>
156 data07,20,67,c5,60,ea,20,91 <fg>
158 data94,20,84,9d,20,c0,b8,ea <io>
160 data20,91,94,20,84,9d,86,80 <kb>
162 dataea,20,91,94,20,84,9d,86 <fg>
164 data7e,20,91,94,20,84,9d,86 <ci>
166 data7f,ea,20,91,94,20,84,9d <fe>
168 data8a,a8,a9,00,4c,61,b8,ea <gi>
170 dataea,ea,ea,ea,ea,ea <mo>
172 : <jd>
174 rem datas fuer merge <hl>
176 : <jl>
178 fort=4099to4136:readda$ <ik>
180 da=dec(da$):poket,da:next <in>
182 : <kh>
184 dataa5,2b,8d,1b,10,a5,2c,8d <fk>
186 data1c,10,38,a5,2d,e9,02,85 <gd>
188 data2b,a5,2e,e9,00,85,2c,60 <ci>
190 dataea,ea,ea,ad,1b,10,85,2b <ff>
192 dataad,1c,10,85,2c,60 <ek>
194 : <lp>
196 rem datas fuer bg-colour <no>
198 : <mh>
200 fort=4137to4170:readda$ <ne>
202 da=dec(da$):poket,da:next <ci>
204 : <nd>
206 data78,a9,36,a2,10,8d,14,03 <ji>
208 data8e,15,03,58,60,ee,19,ff <nc>
210 dataea,4c,0e,ce,00,78,a9,0e <nc>
212 dataa2,ce,8d,14,03,8e,15,03 <lm>
214 data58,60 <gg>

```



20/64/128

Das unabhängige Commodore-Magazin

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre COMMODORE WELT nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Ihr“ Kiosk nicht beliefert wurde? Kein Problem! Für ganze 60 DM liefern wir Ihnen per Post zwölf Hefte ins Haus (Ausland 80 DM). Einfach den Bestellschein ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). COMMODORE WELT kommt dann pünktlich ins Haus.



**WICHTIGE RECHTLICHE
GARANTIE!**

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen – Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für zwölf Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

DAS SONDERANGEBOT: PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen COMMODORE WELT: KLEINANZEIGEN SIND KOSTENLOSE FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift.

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS

Wir veröffentlichen nur Kleinanzeigen privater In-

serenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 5.00 plus Mehrwertsteuer.

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet. Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen.

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe.

Die Insertion ist nicht vom Kauf des Heftes abhängig. Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes – Computer – gehören, nicht abzurufen oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeigenteils zulässt.

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen die Listings dieses Heftes auf

Kassette **2/88** (20 DM) Diskette **2/88** (25 DM)

Ich zahle:

Zutreffendes bitte ankreuzen!

per beigefügtem Scheck () Bar ()

Gegen Bankabbuchung am Versandtag ()

Meine Bank (mit Ortsname) _____ **2/88**

Meine Kontonummer _____

Meine Bankleitzahl _____ (steht auf jedem Bankauszug) _____

Vorname _____ Nachname _____

Str./Nr. _____ Plz./Ort _____

Verkaufsbedingungen: Lieferung nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme. Umtausch bei Nichtfunktionieren.

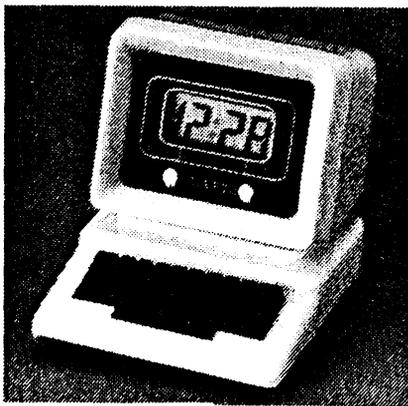
Unterschrift _____

Bitte ausschneiden und einsenden an

**COMMODORE WELT
KASSETTENSERVICE 2/88
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM**

LESER WERBEN LESER

GEWINNEN SIE EINE COMPUTER-UHR! Und zusätzlich eventuell noch ein großes Commodore-Buch. Oder ein Paket Disketten. ODER AUCH EINEN COMMODORE-DRUCKER — ODER EINE DISKETTENSTATION! Wie? Sie werben einen Abonnenten. Dann haben Sie auf jeden Fall schon die Computer-Uhr gewonnen. Zusätzlich verlosen wir unter allen, die mitmachen, jeden Monat vier weitere wertvolle Preise. Und alle sechs Monate gibt es einen Hauptpreis unter allen Abo-Werbern zu gewinnen. Also: Mitmachen. Mitgewinnen.



Herrn/Frau _____

Straße/Hausnr. _____

Plz./Ort _____

Ja, ich mache mit beim Abo-Wettbewerb. Ich habe

als neuen Abonnenten der COMMODORE WELT geworben.

Der neue Abonnent war bisher noch nicht Bezieher dieser Zeitschrift.

Als Prämie erhalte ich nach Eingang des Abo-Entgeltes auf jeden Fall eine Computer-Uhr, wie abgebildet, und nehme zusätzlich noch an der Verlosung des Monats sowie der halbjährlichen Hauptpreise teil. Mir ist bekannt, daß der Rechtsweg bei den Verlosungen ausgeschlossen ist.

Meinen Preis senden Sie an

Name _____

Straße/Hsnr. _____

Plz./Ort _____

(Bitte ausschneiden und zusammen mit der Abo-Bestellkarte links einsenden!) **2/88**

VERDIENEN SIE GELD MIT IHREM COMPUTER!

Haben Sie einen Commodore VC 20 oder C 64? Einen 16/116, Plus 4? Oder einen 128? Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet COMMODORE WELT Ihnen die Möglichkeit, mit diesem Hobby Geld zu verdienen!

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware – eventuelle Erweiterungen – benutzte Peripherie – hervorgehen muß (Schauen Sie sich dazu den Kopf unserer Programmlistings an.)

Benötigt werden: Zwei Listings des Programms sowie eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,-!

Sollten Sie keinen Drucker haben, genügt der Datenträger.

Sie erhalten Ihre Kassette/Diskette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleinige Inhaber der Urheberrechte sind! Benutzen Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Redaktion amerikanische und englische Fachzeitschriften liest und „umgestaltete“ Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

Name des Einsenders: _____
Straße/Hausnr./Tel.: _____
Plz/Ort: _____

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

Benötigte Geräte: _____

Beigefügt () Listings () Kassette () Diskette

Ich versichere, der alleinige Urheber des Programmes zu sein!

Hiermit ermächtige ich die Redaktion, dieses Programm abzdrukken und wirtschaftlich zu verwerten. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung, das Copyright geht insoweit auf den Verlag über.

Rechtsverbindliche Unterschrift

COMMODORE WELT
PROGRAMM-REDAKTION
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

```

216 : <ol>
218 rem datas fuer hg-colour <ch>
220 : <pd>
222 fort=4171to4190:readda$ <jc>
224 da=dec(da$):poket,da:next <gg>
226 : <pp>
228 data78,a9,59,a2,10,8d,14,03 <ff>
230 data8e,15,03,58,60,00,ee,15 <fj>
232 dataff,4c,0e,ce <oh>
234 : <ap>
236 rem datas fuer beep <cf>
238 : <bh>
240 fort=4191to4231:readd$ <me>
242 d=dec(d$):poket,d:next <ll>
244 : <cd>
246 data78,a9,6c,a0,10,8d,14,03 <bj>
248 data8c,15,03,58,60,a5,c6,c9 <eo>
250 data40,d0,0b,ad,11,ff,29,ef <hj>
252 data8d,11,ff,4c,0e,ce,ad,11 <dn>
254 dataff,09,10,8d,11,ff,4c,0e,ce <ck>
256 : <dm>
258 rem datas fuer basic-toolaktiv
ierer <bp>
260 : <ee>
262 fort=4232to4293:readda$ <am>
264 da=dec(da$):poket,da:next <bc>
266 : <fa>
268 dataa2,01,a0,14,86,2b,84,2c <kk>
270 dataa9,93,20,d2,ff,a2,00,bd <ne>
272 dataaf,10,9d,00,0c,e8,e0,17 <ll>
274 datad0,f5,a2,02,86,cd,a2,00 <dg>
276 data86,ca,4c,5e,de,00,00,20 <db>
278 dataaa,aa,a0,82,81,93,89,83 <ef>
280 dataad,94,8f,8f,8c,a0,96,b1 <hd>
282 dataae,b0,a0,aa,aa,20 <cn>
284 : <he>
286 rem datas fuer turbo-mode <lb>
288 : <hm>
290 fort=4294to4311:readda$ <oo>
292 da=dec(da$):poket,da:next <om>
294 : <ii>
296 dataad,06,ff,29,ef,8d,06,ff <cg>
298 data60,ad,06,ff,09,10,8d,06 <mn>
300 dataff,60 <ip>
302 : <ji>
304 rem datas fuer extended-colour
-mode <cd>
306 : <ka>
308 fort=4312to4329:readda$ <gg>
310 da=dec(da$):poket,da:next <ia>
312 : <km>
314 dataad,06,ff,09,40,8d,06,ff <ka>
316 data60,ad,06,ff,29,1b,8d,06 <ec>
318 dataff,60 <ad>
320 : <lm>
322 rem datas fuer 38 / 40 spalten <ig>
324 : <me>
326 fort=4330to4347:readda$ <ej>
328 da=dec(da$):poket,da:next <je>
330 : <na>
332 dataad,07,ff,29,f0,8d,07,ff <ok>
334 data60,ad,07,ff,09,08,8d,07 <pn>
336 dataff,60 <ig>
338 : <oa>
340 rem datas fuer 24 / 25 zeilen <pl>
342 : <oi>
344 fort=4348to4365:readda$ <oa>
346 da=dec(da$):poket,da:next <ag>
348 : <pe>
350 dataad,06,ff,29,f7,8d,06,ff <bd>
352 data60,ad,06,ff,09,08,8d,06 <bo>
354 dataff,60 <ai>
356 : <ae>
358 rem datas fuer spacezeile <oi>
360 : <am>
362 fort=4366to4374:readda$ <ka>
364 da=dec(da$):poket,da:next <gf>
366 : <bi>
368 data20,91,94,20,84,9d,4c,f7 <ll>
370 datada <gn>
372 : <ce>
374 rem datas fuer clearwindow <kd>
376 : <cm>
378 fort=4375to4382:readda$ <mk>
380 da=dec(da$):poket,da:next <ef>
382 : <di>
384 dataa9,13,20,d2,ff,4c,d2,ff <pf>
386 : <eb>
388 rem datas fuer listerror <di>
390 : <ej>
392 fort=4383to4404:readda$ <kg>
394 da=dec(da$):poket,da:next <kd>
396 : <ff>
398 dataa9,a1,a2,94,8d,06,03,8e <mc>
400 data07,03,60,a9,6e,a2,8b,8d <ac>
402 data06,03,8e,07,03,60 <pc>
404 : <gf>
406 rem datas fuer multi-colour-te
xt-mode <mc>
408 : <gn>
410 fort=4405to4422:readda$ <ok>
412 da=dec(da$):poket,da:next <ge>
414 : <hj>
416 dataad,07,ff,09,10,8d,07,ff <dp>
418 data60,ad,07,ff,29,ef,8d,07 <dc>
420 dataff,60 <lf>
422 : <ij>
424 rem datas fuer saveerror <lm>
426 : <jb>
428 fort=4423to4444:readda$ <pl>
430 da=dec(da$):poket,da:next <dk>
432 : <jn>
434 dataa9,a1,8d,30,03,a0,94,8c <le>
436 data31,03,60 <md>
438 dataa9,a4,8d,30,03,a9,f1,8d <fn>
440 data31,03,60 <kl>

```

```

442 :
444 rem datas fuer swap hc/rc
446 :
448 fort=4445to4461:readda$
450 da=dec(da$):poket,da:next
452 :
454 dataad,19,ff,85,d1,ad,15,ff
456 data8d,19,ff,a5,d1,8d,15,ff
458 data60
460 :
462 rem datas fuer out
464 :
466 fort=4462to4472:readda$
468 da=dec(da$):poket,da:next
470 :
472 dataa9,10,85,2c,a9,01,85,2b
474 data4c,f9,ff
476 :
478 rem datas fuer zeichendefiniti
on
480 :
482 fort=4473to4500:readda$
484 da=dec(da$):poket,da:next
486 :
488 data20,de,9d,85,d3,84,d2,a0
490 data00,84,d0,20,91,94,20,84
492 data9d,a4,d0,8a,91,d2,c8,c0
494 data08,d0,ee,60
496 :
498 rem datas fuer bildverknuepfun
g
500 :
502 fort=4501to4508:readda$
504 da=dec(da$):poket,da:next
506 :
508 data20,81,9d,86,d2,4c,e3,13
510 :
512 fort=5091to5117:readda$
514 da=dec(da$):poket,da:next
516 :
518 dataa9,0c,85,d1,a9,00,85,d0
520 dataa2,04,a0,00,b1,d0,45,d2
522 data91,d0,88,d0,f7,e6,d1,ca
524 datad0,f2,60
526 :
528 rem datas fuer soft-/hardrever
s
530 :
532 fort=4510to4527:readda$
534 da=dec(da$):poket,da:next
536 :
538 dataad,07,ff,09,80,8d,07,ff
540 data60
542 dataad,07,ff,29,7f,8d,07,ff
544 data60
546 :
548 rem datas fuer kill
550 :
442 :
444 rem datas fuer swap hc/rc
446 :
448 fort=4445to4461:readda$
450 da=dec(da$):poket,da:next
452 :
454 dataad,19,ff,85,d1,ad,15,ff
456 data8d,19,ff,a5,d1,8d,15,ff
458 data60
460 :
462 rem datas fuer out
464 :
466 fort=4462to4472:readda$
468 da=dec(da$):poket,da:next
470 :
472 dataa9,10,85,2c,a9,01,85,2b
474 data4c,f9,ff
476 :
478 rem datas fuer zeichendefiniti
on
480 :
482 fort=4473to4500:readda$
484 da=dec(da$):poket,da:next
486 :
488 data20,de,9d,85,d3,84,d2,a0
490 data00,84,d0,20,91,94,20,84
492 data9d,a4,d0,8a,91,d2,c8,c0
494 data08,d0,ee,60
496 :
498 rem datas fuer bildverknuepfun
g
500 :
502 fort=4501to4508:readda$
504 da=dec(da$):poket,da:next
506 :
508 data20,81,9d,86,d2,4c,e3,13
510 :
512 fort=5091to5117:readda$
514 da=dec(da$):poket,da:next
516 :
518 dataa9,0c,85,d1,a9,00,85,d0
520 dataa2,04,a0,00,b1,d0,45,d2
522 data91,d0,88,d0,f7,e6,d1,ca
524 datad0,f2,60
526 :
528 rem datas fuer soft-/hardrever
s
530 :
532 fort=4510to4527:readda$
534 da=dec(da$):poket,da:next
536 :
538 dataad,07,ff,09,80,8d,07,ff
540 data60
542 dataad,07,ff,29,7f,8d,07,ff
544 data60
546 :
548 rem datas fuer kill
550 :
552 fort=4528to4544:readda$
554 da=dec(da$):poket,da:next
556 :
558 dataa9,01,85,5f,a9,15,85,60
560 dataa9,ff,85,14,85,15,4c,65
562 dataae
564 :
566 rem datas fuer dreieck
568 :
570 fort=4545to4671:readda$
572 da=dec(da$):poket,da:next
574 :
576 data20,91,94,20,84,9d,86,84
578 data20,91,94,20,84,9d,86,d0
580 data20,91,94,20,84,9d,86,d1
582 data20,91,94,20,84,9d,86,d2
584 data20,91,94,20,84,9d,86,d3
586 data20,91,94,20,84,9d,86,d4
588 data20,91,94,20,84,9d,86,d5
590 dataa2,00,86,d6,ea,a5,d6,8d
592 dataae,02,8d,b0,02,8d,b2,02
594 data8d,b4,02,ea,a5,d0,8d,ad
596 data02,a5,d1,8d,af,02,a5,d2
598 data8d,b1,02,a5,d3,8d,b3,02
600 data20,da,c0,ea,a5,d4,8d,b1
602 data02,a5,d5,8d,b3,02,20,da
604 datac0,ea,a5,d0,8d,ad,02,a5
606 datad1,8d,af,02,4c,da,c0
608 :
610 rem datas fuer m-c-t-colours
612 :
614 fort=4672to4700:readda$
616 da=dec(da$):poket,da:next
618 :
620 data20,91,94,20,84,9d,86,d1
622 data20,91,94,20,84,9d,86,d2
624 dataea,a5,d1,8d,16,ff,a5,d2
626 data8d,17,ff,ea,60
628 :
630 rem datas fuer e-c-colours
632 :
634 fort=4701to4742:readda$
636 da=dec(da$):poket,da:next
638 :
640 data20,91,94,20,84,9d,86,d1
642 data20,91,94,20,84,9d,86,d2
644 data20,91,94,20,84,9d,86,d3
646 dataea,a5,d1,8d,16,ff,a5,d2
648 data8d,17,ff,a5,d3,8d,18,ff
650 dataea,60
652 :
654 rem datas fuer savehgr
656 :
658 fort=4743to4777:readda$
660 da=dec(da$):poket,da:next
662 :
664 dataa9,02,a2,01,a0,01,20,ba
666 dataff,a9,00,a2,00,a0,00,20

```

```

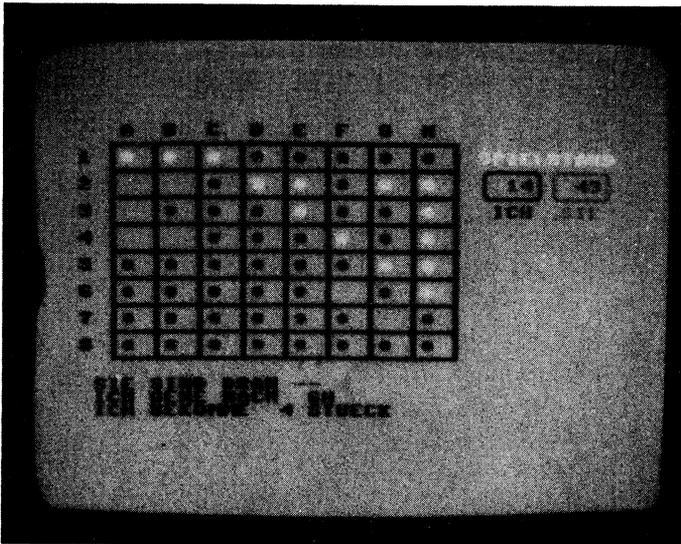
668 databd,ff,a9,00,85,d2,a9,20 <fo>
670 data85,d3,a9,d2,a2,00,a0,40 <mo>
672 data4c,d8,ff <ma>
674 : <id>
676 rem datas fuer loadhgr <fa>
678 : <il>
680 fort=4778to4802:readda$ <ao>
682 da=dec(da$):poket,da:next <nn>
684 : <jh>
686 dataa9,02,a2,01,a0,00,20,ba <mc>
688 dataff,a9,00,a2,00,a0,00,20 <lf>
690 databd,ff,a2,00,a0,20,4c,d5 <dl>
692 dataff <je>
694 : <kl>
696 rem datas fuer soff <fi>
698 : <ld>
700 fort=4803to4841:readda$ <hd>
702 da=dec(da$):poket,da:next <kf>
704 : <lp>
706 data20,91,94,20,84,9d,86,d0 <ke>
708 dataad,11,ff,e0,01,d0,05,29 <nm>
710 dataef,4c,e6,12,e0,02,d0,05 <ap>
712 data29,df,4c,e6,12,e0,03,d0 <ij>
714 data05,29,bf,8d,11,ff,60 <mh>
716 : <nh>
718 rem datas fuer son <nf>
720 : <np>
722 fort=4842to4880:readda$ <eo>
724 da=dec(da$):poket,da:next <mb>
726 : <ol>
728 data20,91,94,20,84,9d,86,d0 <bd>
730 dataad,11,ff,e0,01,d0,05,09 <lh>
732 data10,4c,0d,13,e0,02,d0,05 <eh>
734 data09,20,4c,0d,13,e0,03,d0 <fp>
736 data05,09,40,8d,11,ff,60 <jj>
738 : <ad>
740 rem datas fuer zcopy <ge>
742 : <al>
744 fort=4881to4944:readda$ <eo>
746 da=dec(da$):poket,da:next <fo>
748 : <bh>
750 dataa2,00,bd,00,d0,9d,00,38 <ba>
752 databd,00,d1,9d,00,39,bd,00 <nh>
754 datad2,9d,00,3a,bd,00,d3,9d <lk>
756 data00,3b,bd,00,d4,9d,00,3c <dj>
758 databd,00,d5,9d,00,3d,bd,00 <mf>
760 datad6,9d,00,3e,bd,00,d7,9d <bo>
762 data00,3f,e8,e0,ff,d0,cb,a0 <lm>
764 data37,a9,ff,85,37,84,38,60 <hl>
766 : <dk>
768 rem datas fuer zon <pe>
770 : <ee>
772 fort=4945to4955:readda$ <po>
774 da=dec(da$):poket,da:next <bc>
776 : <fa>
778 dataa9,00,8d,12,ff,a9,38,8d <np>
780 data13,ff,60 <bp>
782 : <fm>
784 rem datas fuer d.z. <bk>
786 : <ge>
788 fort=4956to5090:readda$ <eh>
790 da=dec(da$):poket,da:next <gn>
792 : <ha>
794 dataa2,00,bd,00,d0,9d,00,3c <bb>
796 databd,00,d1,9d,00,3d,bd,00 <mi>
798 datad2,9d,00,3e,bd,00,d3,9d <oa>
800 data00,3f,ca,d0,e5,a9,00,8d <kb>
802 data12,ff,a9,3c,8d,13,ff,a2 <nk>
804 data00,bd,b8,13,9d,08,3e,e8 <lk>
806 datae0,08,d0,f5,a2,00,bd,c0 <bi>
808 data13,9d,78,3e,e8,e0,08,d0 <op>
810 dataf5,a2,00,bd,c8,13,9d,a8 <km>
812 data3e,e8,e0,08,d0,f5,a2,00 <oj>
814 databd,d0,13,9d,98,3e,e8,e0 <bo>
816 data0d,d0,f5,60 <mp>
818 data66,18,3c,66,7e,66,66,00 <al>
820 data66,3c,66,66,66,66,3c,00 <al>
822 data66,00,66,66,66,66,3c,00 <ge>
824 data78,66,66,78,66,66,6c,60 <dp>
826 dataa9,c4,a2,d0,8d,12,ff,8e <ke>
828 data13,ff,60 <lp>
830 : <lm>
832 rem basic-tool <ag>
834 : <me>
836 sys4232 <mf>
838 : <mm>
840 rem datas fuer (c) <nn>
842 : <ne>
844 fort=1015to1063:readb$ <fb>
846 b=dec(b$):poket,b:next <dd>
848 : <oa>
850 dataa9,ff,a0,03,20,88,90,60 <od>
852 data2a,2a,2a,2a,2a,20,28,43 <dg>
854 data29,20,42,59,20,4d,49,43 <jk>
856 data48,41,45,4c,20,49,4e,44 <od>
858 data45,4e,20,49,4e,20,31,39 <jn>
860 data38,37,20,2a,2a,2a,2a,2a <op>
862 data00 <kf>
864 : <aa>
866 char1,0,2,"":sys1015 <pi>
868 : <ai>
870 sys1674,4,0,24,39 <de>
872 sys1716,8,1,234,2,25 <fg>
874 end <fk>
876 : <bi>
60000 rem nachspann <ge>
60010 rem * steuercodes * <ko>
60020 h$=chr$(019):cl$=chr$(147) <fa>
60030 return <fo>

```

COMMODORE-WELT
Ehrlich und
unabhängig

Retour - Brettspiel

Retour ist eine Version des bekannten Brettspiels Reversi für den C16/116/Plus4. Sie spielen gegen den Computer. Nach dem Starten des Programms können Sie wählen, ob Sie die Spielanleitung sehen wollen.



Dort finden Sie weitere Erklärungen. Danach wählen Sie einen der beiden Schwierigkeitsgrade. Ob Sie gerne die roten oder gelben Steine hätten, ob Sie zuerst beginnen wollen, oder der Computer, können Sie selbst bestimmen. □

```

10 rem retour-brettspiel=====c16 <la>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by detlef lokay <hg>
50 rem <pd>
60 rem <ah>
70 rem basic v3.5 <nl>
80 rem c16/116/plus4 <ki>
90 rem ===== <jg>
100 gosub 3290 <fi>
110 printcl$:color0,2,5:color4,2,5
:color1,10,3 <eb>
120 print" "z5$z5$z5$z5$z5$" "z5$z
5$z5$z5$z5$z5$" "z5$z5$z5$z5$z5$; <dj>
130 print" "z5$z5$z5$z5$z5$" "z5$b
3$z5$" "z5$z5$z5$z5$z5$ <gk>
140 print" "z5$b3$z5$" "z5$b3$b4$z
5$b3$z5$b3$z5$" "z5$b3$z5$" "z5$b3
$z5$ <gb>
150 print" "z5$z5$z5$z5$z5$" "z5$z
5$z5$z5$b4$z5$b3$z5$b3$z5$; <ja>
160 print" "z5$b3$z5$" "z5$z5$z5$z
5$z5$ <gj>
170 print" "z5$b2$z5$b2$z5$b4$b3$z
5$b3$z5$b3$z5$s2$z5$b3$z5$s2$z5$b3
$z5$s2$ <hl>
180 print" "z5$b3$z5$s2$z5$b4$b3$z
5$b3$z5$b3$z5$s2$z5$b3$z5$s2$z5$b3
$z5$ <og>
190 print" "z5$b3$z5$" "z5$z5$z5$z
5$z5$b3$z5$b3$z5$z5$z5$z5$z5$; <aj>
200 print" "z5$z5$z5$z5$z5$" "z5$b
3$z5$ <he>
210 color1,7,1 <gm>
220 aa$="UCCCCCCCCCCCCCCCCCCCCCI <kl>
230 ab$="B"+b6$+"lokay"+b4$+b4$+"B
" <jp>
240 ac$="B martin-bihn-str.3 B" <co>
250 ad$="B 6054"+b5$+"rodgau 5 B" <ok>
260 af$="JCCCCCCCCCCCCCCCCCCCCCK" <ea>
270 char1,8,9,aa$ <hg>
280 char1,8,10,ab$ <an>
290 char1,8,11,ac$ <jd>
300 char1,8,12,ad$ <fa>
310 char1,8,13,af$:color1,6,2 <oo>
320 char1,2,20,"c-16 version eines
strategiespieles." <gi>
330 forx=1to2000:next <md>
340 char1,2,20,b$+b$+b$+b6$:color1
,1 <hk>
350 char1,6,20,fl$+ye$+"zum start
taste druecken."+lb$+fo$ <df>
360 getkeyy$ <fe>
370 printcl$:printlb$ <hi>
380 printcl$bkl$left$(qd$,5)b3$"ret
our -- ein spiel fuer knobler":pri
nt <am>
390 printb3$"sie muessen hoellisch

```

```

aufpassen," <fh>
400 printb3$"sonst gewinne ich,den <fn>
n ich bin " <da>
410 printb3$"schlauer als mancher <ed>
schlauberger" <af>
420 foru=1to30:uy$=uy$+chr$(29):ne <gh>
xt <kk>
430 forz=1to35:uz$=uz$+chr$(29):ne <ja>
xt <cj>
440 dn$=he$+left$(qd$,20) <ep>
450 fori=1to39:e$=e$+" ":next <gn>
460 fori=1to39:e$=e$+c1$:next <on>
470 printchr$(144) <hd>
480 fori=1to39:r$=r$+chr$(29):next <fh>
490 dima(9,9),i4(8),j4(8),d$(2) <kp>
500 printchr$(147) <ad>
510 printc4$"brauchen sie anweisun <io>
gen?"; <fi>
520 getkeyx$ <ge>
530 ifx$="j"thenprint" ja" <np>
540 ifx$<>"j"thenprint" nein" <fn>
550 ifleft$(x$,1)<>"j"then890 <np>
560 fori=1to500:next <ki>
570 printc1$ "gespielt wird auf <bg>
einem 8 x 8 brett," <cc>
580 print"reihe "rn$"1"rf$" bis "r <cc>
n$"8"rf$" und spalte "rn$"a"rf$" b <cc>
is "rn$"h"rf$" <cc>
590 print"am anfang ist alles offe <dj>
n.nur in der" <dj>
600 print"mitte liegen vier steine <ck>
.versuchen" <ck>
610 print"sie ihren stein so zu le <lh>
gen,dass er" <lh>
620 print"meine steine flankiert.h <pg>
orizontal," <pg>
630 print"vertikal oder diagonal.b <la>
is zum ende." <la>
640 print"sie wenden ihre steine,w <jb>
enn" <jb>
650 printb3$"CCC"ym$"CCC"ym$"CCC"y <ia>
m$"CCC"ym$"CCC"ym$"CCC" <ia>
660 printb6$"B "re$"Q"bk$" B "re$" <ki>
Q"bk$" B "re$"Q"bk$" B "re$"Q"bk$" <ki>
B "ye$"Q"bk$" <ki>
670 printb3$"CCC"ym$"CCC"ym$"CCC"y <kc>
m$"CCC"ym$"CCC"ym$"CCC" <kc>
680 print:print"sie,wie hier im be <kl>
ispiel,die " <kl>
690 print"steine von"re$" Q"bk$" i <pg>
n"ye$" Q"bk$" verwandeln" <pg>
700 i=ti:j=1:printhe$left$(qd$,9)l <pb>
eft$(qr$,4); <pb>
710 if(ti-i)>1200then750 <gd>
720 ifj=1thenprintye$+"Q"+bk$+c1$; <gl>
: goto740 <gl>
730 print" "c1$; <fl>
740 j=-1*j:fork=1to150:next:goto71
0 <fn>
750 printhe$left$(qd$,9)c3$c3$c3$y <fn>
e$" Q"bk$" B"ye$" Q"bk$; <ed>
760 print" B"ye$" Q"bk$" B"ye$" Q" <ed>
bk$" B"ye$" Q"bk$left$(qd$,4) <gh>
770 print"durch eingabe einer zahl <ja>
fuer die" <ja>
780 print"reihe und ein buchstabe <ep>
fuer die spalte." <ep>
790 print"achtung! es muss mindest <ak>
ens 1 stein " <ak>
800 print"gedreht werden.wenn dies <pf>
nicht moeglich" <pf>
810 print"ist,muss gepasst werden, <jh>
indem sie" <jh>
820 print"fuer reihenangabe einfac <ab>
h 'a 0' ein" <ab>
830 print"geben." <im>
840 print"wenn sie bereit sind,bit <fc>
te 'j' druecken." <fc>
850 getkeyx$ <bc>
860 ifx$<>"j"then850 <od>
870 printc1$ <ca>
880 f2=0 <lp>
890 print"soll ich mich anstrengen <gj>
?"; <gj>
900 s2=0 <gp>
910 getkeyx$ <no>
920 ifx$="j"thenprint" ja" <oo>
930 ifx$<>"j"thenprint" nein" <jm>
940 ifleft$(x$,1)="j"then960 <ke>
950 s2=4 <jg>
960 b=-1:w=1 <hj>
970 d$(b+1)=re$+"Q"+bk$ <ad>
980 d$(0+1)=" " <on>
990 d$(w+1)=ye$+"Q"+bk$ <kd>
1000 fork=1to8 <jd>
1010 readi4(k),j4(k) <fh>
1020 next <ib>
1030 fori=0to9:forj=0to9 <jk>
1040 a(i,j)=0 <be>
1050 nextj,i <ma>
1060 a(4,4)=w <bo>
1070 a(5,5)=w <cm>
1080 a(4,5)=b <bf>
1090 a(5,4)=b <ec>
1100 c1=2:h1=c1:n1=4:z=0 <jf>
1110 print" spielen sie"re$" rot"b <oo>
k$" oder"ye$" gelb"bk$" ?":print <oo>
1120 c=w:h=b <bo>
1130 getkeyx$ <kc>
1140 ifx$="r"thenprintre$+"rote st <mh>
eine"+bk$:print <mh>
1150 ifx$<>"r"thenprintye$"gelbe s <dm>
teine"bk$:print <dm>
1160 if left$(x$,1)="r"then1180 <ed>
1170 c=b:h=w <bl>
1180 print"wollen sie anfangen"; <pk>

```

1190	getkeyx\$	<go>	1720	goto1270	<fb>
1200	ifx\$= "n"thengosub3050	<gl>	1730	ifa(i,j)=0then 1760	<ak>
1210	ifx\$<>"n"thengosub3090	<ob>	1740	printdn\$c4\$e\$b2\$"leider beset	
1220	fori=1to2000:next	<ig>	zt"		<ca>
1230	printc1\$;	<lc>	1750	goto1840	<cg>
1240	gosub2660	<gl>	1760	gosub2330	<jg>
1250	ifleft\$(x\$,1)<>"n"then1620	<lh>	1770	iff1=1then1800	<ko>
1260	printdn\$e\$b2\$"ich denke nach		1780	printdn\$c4\$e\$b2\$"das ist fals	
!"		<pd>	ch"		<jl>
1270	b1=-1:i3=0:j3=0:t1=c:t2=h	<lc>	1790	goto1840	<bl>
1280	fori=1to8:forj=1to8	<pn>	1800	u=-1	<ok>
1290	ifa(i,j)<>0then1430		1810	gosub2410	<pa>
1300	gosub2330	<pa>	1820	ifs1>0then1860	<oc>
1310	iff1=0then1430	<bh>	1830	printdn\$c4\$e\$b2\$"nicht in der	
1320	u=-1	<mm>	reihe"		<bn>
1330	gosub 2410	<km>	1840	printe\$	<mk>
1340	ifs1=0then1430	<jd>	1850	goto1630	<kj>
1350	if(i-1)*(i-8)<>0then1370	<j>	1860	z=0	<fh>
1360	s1=s1+s2	<hp>	1870	printdn\$c4\$e\$b2\$"sie erhalten	
1370	if(j-1)*(j-8)<>0then1390	<ag>	;"		<od>
1380	s1=s1+s2	<fd>	1880	prints1;	<kb>
1390	ifs1<b1then1430	<ed>	1890	print"von mir":printe\$	<ba>
1400	ifs1>b1then1420		1900	u=1	<ib>
1410	ifrnd(1)>0.5then1430	<lf>	1910	gosub2410	<gn>
1420	b1=s1:i3=i:j3=j	<hf>	1920	h1=h1+s1+1	<dp>
1430	nextj,i	<ji>	1930	c1=c1-s1	<pf>
1440	ifb1>0then1490	<ob>	1940	n1=n1+1	<fo>
1450	printdn\$c4\$e\$b2\$"ich passe"	<aj>	1950	gosub2660	<jc>
1460	ifz=1then1980	<lh>	1960	ifc1=0orn1=64then1980	<kn>
1470	z=1	<bl>	1970	goto1270	
1480	goto1620	<df>	1980	printdn\$;e\$:printe\$:printe\$:g	
1490	z=0	<lj>	osub3230		<eo>
1500	printdn\$c4\$e\$b2\$"ich gehe nac		1990	printdn\$e\$b2\$"sie haben "h1"s	
h "right\$(str\$(i3),1)chr\$(j3+64)"	<nb>	teine"			<dm>
1510	remgosub2450	<bg>	2000	printb2\$"ich habe "c1"steine"	<ad>
1520	i=i3:j=j3:u=1	<eo>	2010	ifh1=c1then2050	<cb>
1530	gosub2410	<ln>	2020	ifh1>c1then2070	<pd>
1540	c1=c1+s1+1	<on>	2030	printb2\$"ich habe gewonnen."	<fn>
1550	h1=h1-s1	<cg>	2040	goto2080	<hl>
1560	n1=n1+1	<jp>	2050	print"hart !!"	<gb>
1570	printdn\$c4\$c4\$e\$b2\$"ich bekom		2060	goto2260	<bb>
me ";	<fi>	2070	printb2\$"ihr sieg !!"		<kd>
1580	prints1;	<ng>	2080	c1=c1-h1	<gg>
1590	print"stueck":gosub3230	<ch>	2090	ifc1>0then2110	<dg>
1600	gosub2660	<jg>	2100	c1=-c1	<ho>
1610	ifh1=0orn1=64then1980	<ej>	2110	c1=(64*c1)/n1	<pn>
1620	t1=h:t2=c	<ml>	2120	printb2\$"das war ein ";	<dp>
1630	printdn\$e\$b2\$"sie sind dran -		2130	ifc1<11 then2250	<mp>
- ";	<nj>	2140	ifc1<25 then2230		<hk>
1640	gosub2820	<jf>	2150	ifc1<39 then2210	<oh>
1650	ifi>8orj>8 ora(i,j)>8then163		2160	ifc1<53 then2190	<gc>
0	<nf>	2170	print"tolles spiel."		<in>
1660	ifi<>0then 1730	<jn>	2180	goto 2260	<bm>
1670	printdn\$;e\$"passen?(j/n)";	<dk>	2190	print"ausflug."	
1680	getx\$:ifx\$=""then1680	<na>	2200	goto2260	<kd>
1690	ifleft\$(x\$,1)<>"j"then1630	<ao>	2210	print"kampf."	<kb>
1700	ifz=1then1980	<mj>	2220	goto2260	<bn>
1710	z=1	<kd>	2230	print"heisses spiel."	<lc>

2240 goto2260	<jg>	2740 nextj	<ej>
2250 print"eine blamage fuer mich."		2750 print	<ck>
"	<do>	2760 ifi<>8thenprintb3\$za\$"CC"ym\$"	
2260 print	<fc>	CC"ym\$"CC"ym\$"CC"ym\$"CC"ym\$"CC"ym\$	
2270 printb2\$"noch ein spiel?"	<eb>	"CC"ym\$"CC"zi\$	<ej>
2280 getx\$:ifx\$=""then2280	<lh>	2770 nexti	<lp>
2290 ifleft\$(x\$,1)="j"thenrun	<jj>	2780 printb3\$zc\$"CC"zg\$"CC"zg\$"CC"	
2300 printcl\$:print:print:print:pr		zg\$"CC"zg\$"CC"zg\$"CC"zg\$"CC"	
int:print:printb6\$"danke fuer das		"zs\$	<lp>
spiel."	<od>	2790 return	<gp>
2310 print:printb5\$"mir hat's spas		2800 end	<hf>
s gemacht!"	<da>	2810 printdn\$c4\$e\$b2\$"falsch ! noc	
2320 forx=1to1500:next:printcl\$:en		hmal !":goto1630	<dd>
d	<ko>	2820 i=-1:j=-1:k=1	<pk>
2330 f1=0	<lm>	2830 getx\$:ifx\$<>"then2910	<fo>
2340 fori1=-1to1	<nl>	2840 ifk=1thenprintz5\$+" "+c1\$+c1\$	
2350 forj1=-1to1	<cn>	;:goto2860	<fo>
2360 ifa(i+i1,j+j1)=t2then2390	<gd>	2850 print" c1\$;	<kf>
2370 nextj1,i1	<jl>	2860 k=-1*k	<be>
2380 return	<di>	2870 forl=1to200:next	<dp>
2390 f1=1	<hj>	2880 ifi>8 orj>8 thengoto2810	<jj>
2400 return	<ga>	2890 goto2830	<dk>
2410 s1=0	<cd>	2900 ifj>8 ori>8 thengoto2810	<il>
2420 fork=1to8	<oh>	2910 g=asc(x\$)	<am>
2430 i5=i4(k)	<dg>	2920 if47<gandg<58theni=g-48:print	
2440 j5=j4(k)		rn\$x\$r\$f\$;	<hj>
2450 i6=i+i5	<lo>	2930 if64<gandg<74thenj=g-64:print	
2460 j6=j+j5	<ab>	rn\$x\$r\$f\$;	<hl>
2470 s3=0	<ph>	2940 ifi<>-1andj<>-1thenreturn	<al>
2480 ifa(i6,j6)<>t2then2640	<dh>	2950 goto2830	<nd>
2490 s3=s3+1	<cd>	2960 printleft\$(dn\$,2*(i3-1)+5);le	
2500 i6=i6+i5	<oo>	ft\$(r\$,4*(j3-1)+5);	<cg>
2510 j6=j6+j5	<ef>	2970 k=1:m=0:x\$=d\$(c+1)	<nh>
2520 ifa(i6,j6)=t1then2550	<id>	2980 ifk=1thenprintx\$c1\$;:goto3000	<lb>
2530 ifa(i6,j6)=0 then2640	<hl>	2990 printb2\$c1\$;	<hh>
2540 goto2490	<ec>	3000 k=-1*k	<db>
2550 s1=s1+s3	<nk>	3010 forl=1to200:next	<ag>
2560 ifu<>1then2640	<bn>	3020 ifm>9thenreturn	<gd>
2570 i6=i	<ao>	3030 m=m+1:goto2980	<pk>
2580 j6=j	<lk>	3040 data0,1,-1,1,-1,0,-1,-1,0,-1,	
2590 fork1=0tos3	<ng>	1,-1,1,0,1,1	<nj>
2600 a(i6,j6)=t1	<nl>	3050 printchr\$(147):print:print:pr	
2610 i6=i6+i5	<fh>	int	<lg>
2620 j6=j6+j5	<of>	3060 printb3\$"sie schlauberger,":p	
2630 nextk1	<fa>	rint:print	<mg>
2640 nextk	<lh>	3070 printb3\$"gleich werden sie si	
2650 gosub3230: return	<dd>	ch wundern."	<il>
2660 ifh1=2andc1=2thengosub3130	<mk>	3080 fori=1to500:next: return	<hk>
2670 printhe\$tab(0)" ";	<il>	3090 printcl\$:print:print:print:pr	
2680 printc4\$b3\$"a b c d e f		int	<hp>
g h "	<ga>	3100 printb4\$"so koennen sie nie g	
2690 printb3\$zf\$"CC"zh\$"CC"zh\$"CC"		ewinnen.":print:print	<ed>
zh\$"CC"zh\$"CC"zh\$"CC"zh\$"CC"zh\$"CC"		3110 printb4\$"na,dann fangen sie m	
"zd\$	<al>	al an."	<cl>
2700 fori=1to8	<ea>	3120 fori=1to500:next: return	<ai>
2710 printi"B";	<hp>	3130 ua\$=ye\$+"spielstand"+lb\$	<hk>
2720 forj=1to8	<fb>	3140 ub\$=gr\$+"UCCCI"+pu\$+"UCCCI"+1	
2730 printd\$(a(i,j)+1)" B";	<jo>	b\$	<km>

RETOUR-BRETTSPIEL

```

3150 uc$=gr$+"B"+b3$+"B"+pu$+"B"+b
3$+"B"+lb$ <ga>
3160 ud$=gr$+"JCCCK"+pu$+"JCCCK"+1
b$ <fk>
3170 char1,29,3,ua$ <ok>
3180 char1,29,4,ub$ <en>
3190 char1,29,5,uc$ <ie>
3200 char1,29,6,ud$ <ih>
3210 char1,30,7,gr$+"ich"+lb$+b2$+
pu$+"sie"+bk$ <kf>
3220 return <mp>
3230 char1,30,5,b3$ <ie>
3240 color1,4,3 <nk>
3250 printhe$left$(qd$,5);uy$;c1 <pc>
3260 char1,35,5,b3$ <pc>
3270 printhe$left$(qd$,5);uz$;h1 :
color1,1 <ob>
3280 return <eh>
3290 b$=chr$(32):b2$=b$+b$ <fm>
3300 b3$=b2$+b$:b4$=b3$+b$ <cd>
3310 b5$=b4$+b$:b6$=b5$+b$ <be>
3320 b$=b5$+b5$ <op>
3330 rem nachspann ===== <ah>
3340 rem * farbcodes/steuer codes * <jl>
3350 c4$=chr$(017):rn$=chr$(018) <ae>
3360 he$=chr$(019):re$=chr$(028) <nn>
3370 c3$=chr$(029):gr$=chr$(030) <gl>
3380 fl$=chr$(130):fo$=chr$(132) <dk>
3390 bk$=chr$(144):rf$=chr$(146) <il>
3400 cl$=chr$(147):lb$=chr$(154) <jj>
3410 pu$=chr$(156):c1$=chr$(157) <ga>
3420 ye$=chr$(158) <gl>
3430 rem *** zeichensatz/graphik * <lp>
3440 s2$=chr$(160):z5$=chr$(166) <ah>
3450 za$=chr$(171):zc$=chr$(173) <jn>
3460 zd$=chr$(174):zf$=chr$(176) <fg>
3470 zg$=chr$(177):zh$=chr$(178) <hp>
3480 zi$=chr$(179):zs$=chr$(189) <cn>
3490 ym$=chr$(219) <mk>
3500 rem ***** zeichenfolgen * <am>
3510 for q=1 to 40 <pd>
3520 qd$=qd$+c4$:qr$=qr$+c3$ <pn>
3530 next q <ld>
3540 return <fb>
3550 rem ===== <np>
3560 rem 12277 bytes memory <hb>
3570 rem 00336 bytes program <ac>
3580 rem 00567 bytes variables <ob>
3590 rem 00629 bytes arrays <fd>
3600 rem 00922 bytes strings <ph>
3610 rem 01823 bytes free (0) <jc>
3620 rem ===== <jo>

```

Rainbow

Rainbow ist ein unterhaltsames Strategiespiel, an dem vor allem Schnelldenker ihre Freude haben werden. Es kann sowohl allein gegen den Highscore, als auch im Wettkampf zu zweit gespielt werden. In beiden Fällen wird man feststellen, daß es bei diesem Spiel weniger auf Glück, sondern auf Training und Denkvermögen ankommt.

Das vorliegende Programm wurde in einem einfachen BASIC geschrieben. Anfänger können hierdurch hinsichtlich einiger Routinen (zum Beispiel Zeitabfrage) sicher Anregungen für eigene Programme entnehmen. Viel Wert wurde auf eine möglichst knappe Programmierung gelegt. Eine weitere Verkürzung läßt sich durch Weglassen der Zeile zehn zusammen mit dem Anleitungsteil (Zeile 1140 bis 1430) erreichen; die Bedienungsanleitung kann schnell dem Listing entnommen werden. □

```

10 rem rainbow=====c16 <ep>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by peter bergen <pk>
50 rem <pd>
60 rem <ah>
70 rem basic v3.5 <n1>
80 rem c16/116/plus4 <ki>
90 rem ===== <jg>
100 gosub1540:vol6:gosub1240 <jc>
110 vol6:dimf%(4,24):dms1%(27):di
ms2%(27):dms5%(16):dimco%(24):hs=
0:a=rnd(1)-ti <dj>
120 restore:fora=3to24step3:readb:
co%(a)=b:s5%(b)=a:next <mf>
130 data2,3,5,8,9,13,15,16 <kk>
140 color4,1:color0,1:scnclr:p1=0:
p2=0:hb=1 <pb>
150 fora=3to24step3:s1%(a)=0:s2%(a
)=0:next <en>
160 scnclr:char1,5,10,wh$+"wieviel
spieler (1 oder 2) ?" <lc>
170 poke239,0:getkeyz$ <gc>
180 ifz$="1"thensa=1:goto240 <da>
190 ifz$="2"thensa=2:goto240 <no>
200 goto170 <dc>
210 rem <de>
220 rem hauptprogramm <nf>
230 rem <fm>
240 scnclr:gosub1080:forhb=3to24st
ep3:vb=5:gosub1030:next:h1=3:gosub
420 <lb>
250 ifsa=2thenforhb=3to24step3:vb=
20:gosub1030:next:h2=3:gosub430 <hf>
260 forhb=3to24step3:gosub1040:nex
t:ti$="000000":tz=34:color1,7,4 <ga>
270 ifsa=1thengosub1110:gosub1130:
gosub740:gosub760 <pa>
280 ifsa=2thengosub1110:gosub1120:
gosub740:gosub750 <hb>
290 rem <nf>
300 rem joystickabfrage <ho>
310 rem <pn>
320 ifjoy(1)=3andh1<24thengosub440
:h1=h1+3:gosub420:gosub490:hb=h1:g
osub670 <jc>
330 ifjoy(2)=3andh2<24thengosub450
:h2=h2+3:gosub430:gosub580:hb=h2:g
osub690 <gk>
340 ifjoy(1)=7andh1>3thengosub440:
h1=h1-3:gosub420:gosub490:hb=h1:go
sub670 <fj>
350 ifjoy(2)=7andh2>3thengosub450:
h2=h2-3:gosub430:gosub580:hb=h2:go
sub690 <im>
360 ifti$>"000010"thenti$="000000"
:elsegoto320 <kp>
370 tz=tz-1:char1,tz,2,bk$+rn$+" " <mc>
380 iftz=3then880:elsegoto320 <el>
390 rem <jo>
400 rem zeiger <io>
410 rem <mg>
420 color1,2,7:char1,h1,9,rf$+yq$+
rn$+"1"+rf$+z8$+rn$:return <cm>
430 color1,2,7:char1,h2,18,rn$+z8$
+"2"+yq$:return <bd>
440 color1,1:char1,h1,9,b3$:return <ib>
450 color1,1:char1,h2,18,b3$:retur
n <ae>
460 rem <ck>
470 rem leistenabfrage 1.spieler <ke>
480 rem <fc>
490 sf=f%(1,h1):hb=s5%(sf):s1%(hb)
=s1%(hb)+1 <no>
500 ifs1%(hb)=4thenvb=5:sound3,500
,10:gosub1030:s1%(hb)=0:hb=h1:gosu
b1040:return <ol>
510 vb=8-s1%(hb):gosub1060 <ga>
520 b=0:fora=3to24step3:ifs1%(a)>0
thenb=b+1 <eh>
530 next:ifb<8thenreturn <nd>
540 forhb=3to24step3:gosub770:vb=5
:gosub1030:s1%(hb)=0:next:return <gk>
550 rem <np>
560 rem leistenabfrage 2.spieler <fb>
570 rem <ah>
580 sf=f%(4,h2):hb=s5%(sf):s2%(hb)
=s2%(hb)+1 <jh>
590 ifs2%(hb)=4thenvb=20:sound3,60
0,20:gosub1030:s2%(hb)=0:hb=h2:gos
ub1040:return <lb>
600 vb=19+s2%(hb):gosub1060 <kh>
610 b=0:fora=3to24step3:ifs2%(a)>0
thenb=b+1 <kh>
620 next:ifb<8thenreturn <he>
630 forhb=3to24step3:gosub810:vb=2
0:gosub1030:s2%(hb)=0:next:return <fl>
640 rem <je>
650 rem spielfeldveraenderung <cm>
660 rem <lm>
670 r1=f%(1,hb):r2=f%(2,hb):r3=f%(
3,hb):r4=f%(4,hb) <np>
680 f%(1,hb)=r2:f%(2,hb)=r3:f%(3,h
b)=r4:f%(4,hb)=r1:gosub1070:return <no>
690 r1=f%(1,hb):r2=f%(2,hb):r3=f%(
3,hb):r4=f%(4,hb) <nn>
700 f%(1,hb)=r4:f%(2,hb)=r1:f%(3,h
b)=r2:f%(4,hb)=r3:gosub1070:return <bh>
710 rem <ca>
720 rem punktestand <gk>
730 rem <ei>
740 char1,30,6,rn$+wh$:printusing"
####";p1:sound1,s1%(hb)*100,3:retu
rn <lb>
750 char1,30,21,rn$+wh$:printusing
"####";p2:sound1,s2%(hb)*100,3:ret

```

```

urn
760 char1,30,14,rn$+wh$:printusing
"####";hs:return
770 ifs1%(hb)=1thenp1=p1+1
780 ifs1%(hb)=2thenp1=p1+3
790 ifs1%(hb)=3thenp1=p1+10
800 gosub740:return
810 ifs2%(hb)=1thenp2=p2+1
820 ifs2%(hb)=2thenp2=p2+3
830 ifs2%(hb)=3thenp2=p2+10
840 gosub750:return
850 rem
860 rem ende
870 rem
880 ifsa=2then900
890 ifp1>hsthenhs=p1:gosub760
900 char1,3,12,b$+b$+b6$:char1,3,1
5,b$+b$+b6$
910 char1,3,14,wh$+rn$+" noch ein
spiel? (j/n) ":e$="ende-ende-ende
-ende-ende-ende":a=1
920 restore1170
930 getz$
940 char1,3,13,bl$:printmid$(e$,a,
24):a=a-1:ifa=0thena=4
950 readt1,t2,t3:sound1,t1,t2:soun
d2,t1+3,t2:c=c+1:fort4=1tot3:next
960 ifc=25thenc=1:restore1170
970 ifz$="j"then140
980 ifz$="n"thenscnclr:stop
990 goto930
1000 rem
1010 rem routinen
1020 rem
1030 color1,co%(hb),3:fora=vbtovb+
2:char1,hb,a,rn$+"0"+zm$+"P":next:
return
1040 b=10:fora=1to4:x=int(rnd(1)*8
+1)*3:f%(a,hb)=co%(x):gosub1050:ne
xt:return
1050 color1,co%(x),3:char1,hb,b,rn
$+"0"+zm$+"P":char1,hb,b+1,"L"+ze$
+zp$:b=b+2:return
1060 color1,co%(hb),7:char1,hb,vb,
rn$+"0"+zm$+"P":return
1070 b=10:fora=1to4:x=s5%(f%(a,hb)
):gosub1050:next:return
1080 color1,2:fora=3to33:char1,a,1
,rf$+ze$:char1,a,3,zm$:next
1090 char1,2,2,z9$:char1,34,2,zj$
1100 color1,8,7:fora=3to33:char1,a
,2,rn$+" ":next:return
1110 char1,30,5,"1.sp":return
1120 char1,30,20,"2.sp":return
1130 char1,30,13,"high":return
1140 rem
1150 rem sound
1160 rem
1170 data798,5,30,798,5,60,739,10,
60,685,10,60,739,7,120,739,20,200
1180 data798,5,30,798,5,60,739,10,
60,685,10,60,739,7,120,739,20,200
1190 data685,5,30,685,5,60,643,10,
60,685,10,60,643,7,120,643,20,200
1200 data798,5,30,798,5,60,739,10,
60,685,10,60,739,7,120,739,20,200
1210 rem
1220 rem anleitung
1230 rem
1240 color0,1:color1,7,4:scnclr:ch
ar1,0,2,""
1250 print"versuchen sie innerhalb
einer vorgege-"
1260 print"benen zeit (gelber balk
en) soviel punkte"
1270 print"wie moeglich zu erreich
en. punkte werden"
1280 print"vergeben, sobald jede f
arbe auf ihrer"
1290 print"leiste (schmale felder)
mit mind. einem"
1300 print"leuchtfeld vertreten is
t."
1310 print"danach wird wie folgt a
bgerechnet:"
1320 print"1 feld"b3$"einer farbe=
1 punkt"
1330 print"2 felder einer farbe= 3
punkte"
1340 print"3 felder einer farbe=10
punkte"
1350 print"die leistenfelder werde
n gemaess den"
1360 print"breiten feldern zum leu
chten gebracht,"
1370 print"indem der zeiger des sp
ielers (weiss)"
1380 print"mit dem joystick versch
oben wird. es"
1390 print"zaehlt das feld, welche
s dem zeiger am"
1400 print"naechsten steht. danach
findet eine ver-"
1410 print"schiebung der breiten f
elder innerhalb"
1420 print"der gewaehlten spalte s
tatt."
1430 print"ist eine farbe mit 3 le
uchtfeldern in"
1440 print"der leiste vertreten, b
ewirkt ein vier-"
1450 print"ter abruf eine loeschun
g und eine zu-"
1460 print"fallsaenderung der brei
ten felder."
1470 print"(weiter mit leertaste)"

```

Titelbild - Designer

Mit Hilfe eines Screen-Editors ein Titelbild zu erstellen, ist wesentlich komfortabler, als mit Print-anweisungen zu arbeiten. Das Problem ist nur, wie kann das mit einem solchen Editor erstellte Bild in das eigene BASIC-Programm eingebunden werden?

Text- und Farb-RAM zusammen ergeben 2048 Byte, worunter sich auch viele nicht ausdrückbare Codes befinden. Wenn hierfür Stringvariablen definiert würden, wird so ein Bild geradezu unerträglich lang. Jedoch kann für eine Datenkomprimierung gesorgt werden. Sie sorgt sogar dafür, daß dabei noch eine Umwandlung in darstellbare Codes stattfindet. Probieren Sie hierzu doch einmal das Programm Bildgenerator aus.

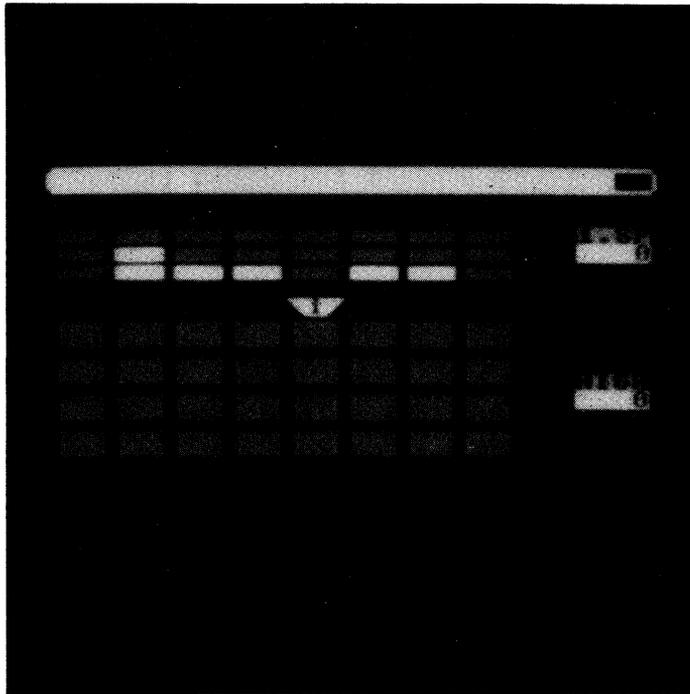
BILDER AUS DATAZEILEN

Die Datazeilen im Programm enthalten Codes, die, in geeigneter Weise interpretiert, die Bildschirmdarstellung eines Hauses bewirken. Darstellbar werden die Codes durch die Beschränkung auf eine geeignete Auswahl. Anstatt Codes von 0 bis 255 zu benutzen, werden nun 32 Buchstaben von chr\$(64) bis chr\$(95) und acht Zahlen von chr\$(48) bis chr\$(55) verwendet. 8*32 ergibt ebenfalls 256, also die Gesamtzahl der möglichen Zeichen. Es braucht nun nicht jedesmal eine Zahl und ein Buchstaben verwendet zu werden, um einen beliebigen Code darzustellen. Eine neue Zahlenangabe ist nur nötig, sofern ein neuer 32-Zeichen-Codebereich angewählt werden soll. Sollen mehr als zwei gleiche Codes hintereinander ausgegeben werden, so braucht nicht das zu wiederholende Zeichen mehrmals erfaßt zu sein, sondern lediglich die Anzahl der Wiederholungen. „+“, gefolgt von einem Buchstaben, kann bis zu 32 Wiederholungen ausdrücken, „*“, gefolgt von zwei Buchstaben, sogar 1024 Wiederholungen des zuletzt ausgegebenen Buchstabens. Hierbei nehmen Leerstellen eine Sonderstellung ein.

Blanks kommen oft vor, das macht eine gesonderte Behandlung wünschenswert. Ein Leerzeichen soll nämlich den gespeicherten Code des letzten Zeichens in keiner Weise beeinflussen. Damit das Leerzeichen sichtbar ist, wird ein Rufzeichen verwendet. Wiederholungen des Leerzeichens erfolgen mit „-“, gefolgt von einem Buchstaben oder mit „/“, dem zwei Buchstaben angehängt werden. Das Dollarzeichen, an das sich eine vierstellige Hexzahl anschließt, stellt den Schreibzeiger auf die entsprechende Adresse um. Das Rautezeichen „#“ kennzeichnet das Ende des zu generierenden Codes. Eine schnelle Maschinenroutine anstelle des langsamen BASIC-Programmes würde Bilder mit geradezu atemberaubender Geschwindigkeit auf den Bildschirm zaubern.

BILDER IN DATAS WANDELN

Was uns jetzt noch fehlt, ist die Erzeugung der Bildschirmcodes. Zuerst benötigen wir hierzu ein Bild, welches in die entsprechenden Datas umzuwandeln ist. Mit dem Programmteil Miniscreen, einem sehr einfachen Screen-Editor, dürfte uns die Erstellung nicht allzu schwer fallen. Wir müssen das BASIC-Ende herabsetzen, damit unsere Daten, die wir zuerst in den



RAINBOW

```

c2$                                <ff>
1480 restore1170:e$="rainbow-rainb
ow-rainbow-rainbow-rainbow-rainbow
-":b=2:a=8:c=1                    <nj>
1490 getz$                          <bj>
1500 color1,b,6:char1,0,0,rn$:prin
tmid$(e$,a,39):a=a-1:ifa=0thena=8:
b=int(rnd(1)*14)+2                <bc>
1510 readt1,t2,t3:sound1,t1,t2:sou
nd2,t1+3,t2:c=c+1:fort4=1tot3:next <pi>
1520 ifc=25thenc=1:restore1170    <pj>
1530 ifz$=" "thenreturn:elsegoto14
90                                  <bk>
1540 b$=chr$(32):b3$=b$+b$+b$     <mm>
1550 b6$=b3$+b3$:b$=b$+b6$+b3$    <ag>
1560 rem nachspann =====<jk>
1570 rem * farbcodes/steuercodes * <kc>
1580 wh$=chr$(005):rn$=chr$(018)   <bn>
1590 bl$=chr$(031):bk$=chr$(144)   <eh>
1600 c2$=chr$(145):rf$=chr$(146)   <gi>
1610 rem *** zeichensatz/graphik * <kc>
1620 z8$=chr$(169):z9$=chr$(170)   <ni>
1630 ze$=chr$(175):zj$=chr$(180)   <mo>
1640 zm$=chr$(183):zp$=chr$(186)   <de>
1650 yq$=chr$(223)                 <md>
1660 return                          <jd>
1670 rem =====<jd>
1680 rem 12277 bytes memory         <nn>
1690 rem 05366 bytes program        <oe>
1700 rem 00252 bytes variables      <ni>
1710 rem 00483 bytes arrays         <gh>
1720 rem 00432 bytes strings        <jf>
1730 rem 05744 bytes free (0)       <bk>
1740 rem =====<ai>

```

LISTING

BILDGENERATOR

```

10 rem bildgenerator=====c16 <fa>
20 rem by a.mittelmeyer <gm>
30 rem ----- <lj>
40 rem bilddatas <jc>
50 rem ----- <ll>
60 data $0c00/et2o3wt/aet!t-]2n3w+
m2m-wn-pm-un-rm-s <pp>
70 data n-tm-qn-vm-ono3w+u2pm-o3t!
!o+d-h+e!!j-pt <dg>
80 data !t!2]!3j!!2o3ww2p!!3t!2]!
3j!!j-pt!!t!2] <kj>
90 data !3j!!t!!j!!t!2]!3j!!j-pt!!
t!2]!3j!!t!!j! <co>
100 data !t!2]!3j!!j-pt!!t!2]!3g!!
t2o!3j!!t!2]!3j <kp>
110 data !!j-pt!!w+d!!t!!j!!w+d!!j
-pt-it!!j-ij-p2l3o+h2l3oozo+hz/fp# <kp>
120 rem ----- <po>
130 rem bildgenerator <mf>
140 rem ----- <fi>
150 scnclr:ar=0:xx=0:goto170 <ac>
160 ad=ad+1 <od>
170 gosub330 <hi>
180 rem ----- <mp>
190 rem interpreterlogik <fa>
200 rem ----- <bf>
210 ifx=33thenpokead,32:goto160 <ce>
220 ifx>63thenx0=(xand31)orz:pokea
d,x0:goto160 <ge>
230 ifx>47thenz=(xand7)*32:goto170 <bc>
240 ifx=36thenx$="":fori=1to4:gosu
b330:x$=x$+chr$(x):next:ad=dec(x$)
:z=32:x0=0:goto170 <hl>
250 ifx=35thenend <hp>
260 ifx=45orx=47thenzx=32:elsezx=x
0 <bl>
270 ifx=47orx=42thengosub330:w=(xa
nd31)*32:elsew=0 <fh>
280 gosub330:w=(xand31)-1+w:i=ad <oj>
290 forad=itoi+w:pokead,zx:next:go
to170 <cp>
300 rem ----- <jl>
310 rem leseroutine <ig>
320 rem ----- <kp>
330 ifar=xxthenar=1:readxx$:xx=len
(xx$):elsear=ar+1 <op>
340 x=asc(mid$(xx$,ar,1)):return <im>
350 rem =====e=n=d=e===== <mb>

```

```

10 rem titelbild-designer=====c16 <ap>
20 rem (p) commodore-welt-team <ie>
30 rem ===== <ng>
40 rem (c) by alfons mittelmeyer <cg>
50 rem <pd>
60 rem <ah>
70 rem basic v3.5 <nl>
80 rem c16/116/plus4 <ki>
90 rem ===== <jg>
100 rem ----- <op>
110 rem auswahlmenue <en>
120 rem ----- <po>
130 poke55,0:poke56,48:clr:scnclr:
printchr$(144); <km>
140 bl$=" ":fori=1to3:print:bl$=bl
$+bl$:next <lf>
150 fori=1to7:print:readx$:printbl
$x$:next <mh>
160 getx:ifx=0then160 <gh>
170 scnclr:printchr$(144):onxgoto3
10,270,260 <ke>
180 end <oj>
190 data titelbild-designer <ge>
200 data ----- <lp>
210 data 1=neuanlage <lo>
220 data 2=bild ansehen <ip>
230 data 3=bild editieren <fb>
240 data ----- <ac>
250 data ihre wahl? <ac>
260 gosub740:goto310 <ff>
270 gosub740:end <dp>
280 rem ----- <io>
290 rem mini-screeneditor <og>
300 rem ----- <jl>
310 open1,0:printchr$(19)chr$(27)"
m"; <nh>
320 trap340:input#1,x$:trap <ol>
330 close1:printchr$(27)"l";:goto3
90 <le>
340 resume <ia>
350 rem ----- <ie>
360 rem code-wandler <ll>
370 rem ----- <ad>
380 data 65305,65305,65301,65301,3
072,4071,2048,3047,-1 <pn>
390 ar=dec("3000"):restore380 <jp>
400 readaa:ifaa=-1thenpokear,35:go
to610 <kf>
410 readee:pokear,36:xx$=hex$(aa):
fori=1to4:pokear+i,asc(mid$(xx$,i,
1)):next:ar=ar+5 <ph>
420 gosub430:goto400 <pg>
430 x0=0:z0=1:b=0:j=0 <jk>
440 for i=aa to ee <dk>
450 x=peek(i):ifx=32thengosub500:b
=b+1:goto490 <kd>
460 gosub540:ifx=x0thenj=j+1:goto4
90 <hm>

```

Hauptspeicher poken, geschützt sind. An das Ende des Programms fügen wir die Umsetzroutine und den Datawandler an. Der Bildgenerator kann auch ohne die anderen Programmteile existieren. Wenn wir alles bis auf diesen Teil nach Erstellung des Bildes löschen und den Rest renumbern, so haben wir das Bild für die Verwendung in BASIC-Programmen zur Verfügung.

LISTING

```

470 gosub500:z=int(x/32):ifz<>z0then
enz0=z:pokear,z+48:ar=ar+1 <kk>
480 y=x-32*z:pokear,y+64:ar=ar+1:x
0=x:goto490 <pa>
490 next:gosub500:goto540 <oh>
500 ifj=0thenreturn <im>
510 ifj=1thenpokear,y+64:ar=ar+1:j
=0:return <nc>
520 ifj<32thenpokear,43:pokear+1,j
+64:ar=ar+2:j=0:return <mo>
530 pokear,42:j1=int(j/32):j=j-32*
j1:pokear+1,j1+64:pokear+2,j+64:ar
=ar+3:j=0:return <dk>
540 ifb=0thenreturn <eg>
550 ifb<3thenfork=0tob-1:pokear+k,
33:next:ar=ar+b:b=0:return <bc>
560 ifb<32thenpokear,45:pokear+1,b
+64:ar=ar+2:b=0:return <jp>
570 pokear,47:b1=int(b/32):b=b-32*
b1:pokear+1,b1+64:pokear+2,b+64:ar
=ar+3:b=0:return <hn>
580 rem ----- <ha>
590 rem datawandler <ic>
600 rem ----- <hm>
610 ax=dec("3000"):gosub680 <ah>
620 gosub690:i=0:x$="" <mi>
630 x=peek(ax+i):i=i+1:x$=x$+chr$(
x) <mm>
640 if x=35theny$=chr$(0):goto660:
elseifi<25then630 <gd>
650 y$="run620"+chr$(13) <ic>
660 key1,str$(ax)+" data "+x$+chr$(
13)+y$ <bp>
670 ax=ax+i:gosub680:poke2035,0:sy
s56364:scnclr:end <gj>
680 ah=int(ax/256):al=ax-256*ah:po
ke216,al:poke217,ah:return <ac>
690 ax=peek(217)*256+peek(216):ret
urn <mf>
700 rem ----- <ca>
710 rem bildgenerator <cm>
720 rem ----- <da>
730 gosub740:end <jd>
740 scnclr:ar=0:xx=0:restore940:go
to760 <bg>
750 ad=ad+1 <n1>
760 gosub920 <in>
770 rem ----- <ob>
780 rem interpreterlogik <n1>
790 rem ----- <io>
800 ifx=33thenpokead,32:goto750 <mm>
810 ifx>63thenx0=(xand31)orz:pokea
d,x0:goto750 <dc>
820 ifx>47thenz=(xand7)*32:goto760 <fp>
830 ifx=36thenx$="":fori=1to4:gosu
b920:x$=x$+chr$(x):next:ad=dec(x$)
:z=32:x0=0:goto760 <ed>
840 ifx=35thenreturn <ma>

```

```

850 ifx=45orx=47thenz=32:elsez=x
0 <oj>
860 ifx=47orx=42thengosub920:w=(x
and31)*32:elsew=0 <dk>
870 gosub920:w=(xand31)-1+w:i=ad <mf>
880 forad=itoi+w:pokead,zx:next:go
to760 <hj>
890 rem ----- <kn>
900 rem leseroutine <po>
910 rem ----- <fc>
920 ifar=xxthenar=1:readxx$:xx=len
(xx$):elsear=ar+1 <gf>
930 x=asc(mid$(xx$,ar,1)):return <pj>
940 x=benoetigter schrott <la>
950 rem ----- <kf>
960 rem bilddatas (programmende) <pg>
970 rem ----- <jk>

```

Erläuterungen zum Listing:

```

130: BASIC-Ende wird herabgesetzt
140-150: Menüerstellung
160-180: Verteiler
190-250: Menüdatas
310: Tastatur als Eingabegerät
320: Eingabe mit Fehlerbehandlung
330: Eingabekanal schließen
380: Zu erfassende Bereiche:
65305 = Rahmenfarbe
65301 = Hintergrundfarbe
3072-4071 = Textspeicher
2048-3047 = Farbspeicher
390: Anfangswerte Codewandlung
ar = Adresse für gewandelten Code
400-420: Abarbeitung der Codebereiche
430: Anfangswerte:
x0 = voriges Zeichen (0-31)
z0 = aktueller Bereich (0-7)
b = Anzahl der Blanks
j = Anzahl der Zeichenwiederholung
440-490: Abarbeitungsschleife
450: Prüfung auf Blank
460: Prüfung auf vorangegangene Blanks
und Behandlung von Zeichenwieder-
holung
470: Prüfung auf Zeichenwiederholung und
Codebereich
480: Code schreiben
490: Schleifenende mit Prüfung auf voran-
gegangene Blanks oder Zeichenwieder-
holung
500-530: Abschluß von Zeichenwiederholung
540-570: Abschluß von Blankwiederholung
610: Codeanfang savein
620: Codeanfang holen, Anfangswerte
Code lesen und an String anfügen
640: Prüfung auf Codenode und Stringlänge
650: Wiedereintritt in Datawandlung
660: Data-Zeile + Wiedereintrittsbefehl
auf Funktionstaste legen
670: Leseadresse aktualisieren und savein,
Funktionstatendruck simulieren,
BASIC-Ende
680: Adress-Save-Routine
690: Adress-Lese-Routine
740: Anfangswerte, Read-Zeiger auf Bild-
datas (keine Rem-Zeile)

```

SERVICE

- 750: Erhöhung der Schreibadresse
- 760: Aufruf der Dataleseroutine
- 800: Prüfung auf Leerzeichen
- 810: Prüfung auf zu schreibenden Code
- 820: Prüfung auf Änderung des Codebereichs
- 830: Prüfung auf Änderung des Adressbereichs
- 840: Prüfung auf Codeende
- 850: Prüfung, ob Blank- oder Codewiederholung
- 860: Prüfung, ob kurze oder lange Wiederholung
- 870: Bestimmung der Wiederholzahl
- 880: Wiederholroutine
- 920-930: Data-Lese-Routine
- 940: Zeile, auf welche an Stelle der erst zu erzeugenden Bilddatas verwiesen wird

MASCHINENSPRACHE-KURS FOLGT

Da BASIC sich nicht gerade durch eine übertriebene Schnelligkeit auszeichnet, kann es schon eine Minute dauern, bis so ein Bild umgesetzt ist. Mit Maschinensprache dagegen ist das etwas ganz anderes. Im nächsten Teil unserer Mikrocomputer-Grundlagen-Serie werden wir uns daher intensiv mit Maschinensprache befassen.

A. Mittelmeyer □

Das war 1987

JAHRESINHALTSVERZEICHNIS 1987

Test & Technik

Ausgabe-Seite	Stichwort:
1/87 / 4-5	Werner: Mach hin, du Dussell! Fünf Superspiele
1/87 / 8	Ledergöttin: Nur für Erwachsene
1/87 / 11	Rossmöller: Wie aus TurboTrans Turboaccess wurde
1/87 / 74-75	Epson EX 800: Sehr zufrieden
1/87 / 76-79	Sidecar: Macht aus dem Amiga einen kompatiblen PC
2/87 / 8-9	Howard The Duck: Er kämpft gegen alle und gegen die Uhr
2/87 / 57-58	Super Cycle: Ein Motorradrennen erster Klasse
2/87 / 12-13	C16/Plus4: Programmgesteuerter Lichtgriffel
2/87 / 59	Buch: Jubel-Trubel-Heiterkeit für alle Spielefreaks
2/87 / 60-63	Okimate 20: Der kleine fleißige Farbdrucker
3/87 / 4-7	128PC/D: Drei 128er im Redaktions-Langzeittest
3/87 / 57	Page-Setter: Der Amiga macht jetzt auch Zeitungen
3/87 / 62-63	Starcomm: DFÜ-Programme ohne Fehl und Tadel
4/87 / 8-10	Labyrinth: Das Spiel verlangt dem Spieler viel Hirnschmalz ab
4/87 / 14-15	Wintergames: Neue interessante Spielfür den Amiga
4/87 / 11	Text 16: Eine Textverarbeitung in der 16er Reihe
4/87 / 59	The Final Cartridge II: Erheblich verbessert
4/87 / 57-58	Quadjet: Verfarb-Tintenstrahldrucker für rund 1000 Mark
5/87 / 4	Nemesis: Das bislang ungelöste Rätsel
5/87 / 5-6	Shogun: Werden Sie Held in Japan
5/87 / 13	Portal: Mal wieder die Erde retten per Computer
5/87 / 14-15	Geos: Macht Ihren C64 leistungsfähiger
5/87 / 48	Starcomm: Ein DFÜ-Programm für Ihren C64
5/87 / 57-59	Calc Plus: Preiswert und dennoch ein gutes Kalkulationsprogramm
5/87 / 62	Netzwerk: Wieviel 64er wollen Sie zusammenschließen?
5/87 / 60-61	Digiview: Friert Bilder für den Amiga ein
5/87 / 8	Superpic: Ein Hardcopymodul für den C64
6/87 / 4-6	Amiga-Spiele: Kingsofts neue Software zum Taschengeldpreis
6/87 / 7	Digidrumm: Superinterface am C64
6/87 / 8	Der neue 128er: Was hat sich geändert?
6/87 / 14	Für Falschbrenner: Ein Profi-Epromlöcher im Test
6/87 / 15	C128 COM: DFÜ betreiben unter CP/M

6/87 / 20-23	Wiesmann Interface 98064: Was es kann und wie man es benutzt
6/87 / 76-78	Spiellecke C64: In Test, Aliens, Star Raider II und Leviathan
7/87 / 4-8	Amiga 500 oder 2000: Die Qual der Wahl für jeden
7/87 / 8-10	Profi-Grafik für den C64: Er sorgt immer wieder für Aufsehen
7/87 / 14-15	Startexter 64: Ein Klassiker im neuen Gewand
7/87 / 24-25	Speicher-Riese C64: Ein Traum wird wahr - 256 KByte für den 64er
7/87 / 28-29	Grafik-Künstler: Starpainter jetzt auch für den Bruder 128er
7/87 / 73	Sindbad: Träumen Sie auch von schönen Frauen und viel Action?
7/87 / 74-75	Arkanoid: Vorsicht, dieses Spiel macht süchtig!
7/87 / 76-77	Murder On The Atlantic: Selbst Agatha Christie hätte Probleme
8/87 / 8-9	GoAmiga: Das erste Dateiverwaltungsprogramm aus der Schweiz
8/87 / 19	Maus: Ein Tierchen aus München für den C64
8/87 / 22-23	IC-Tester: Was tun, wenn man den Baustein nicht kennt?
8/87 / 60-61	Speichererweiterung: Mehr RAM für den 128er
8/87 / 73	CSJ-Games: Drei auf einen Streich
8/87 / 74	Battle: Bumm Bumm, Schiff versenkt
8/87 / 75	Flugschulung: Zu zweit den Tornado steuern oder gegeneinander
8/87 / 76	Spy Versus Spy: Die dritte Version
9/87 / 4-7	Game Fever: TopGun, The Last Ninja und Quartett
9/87 / 7	Saracen: Der Kampf des Kreuzritters gegen die Ungläubigen
9/87 / 8-9	Barbarian: Unglaublich, aber wahr
9/87 / 10	The Sydney Affair: Wer einmal den Mörder kennt...
9/87 / 11	Battle Of Antietam: Der Amerikanische Bürgerkrieg als Spiel
9/87 / 12-13	OCP Artstudio II: Fantastisch, was das Softwarepaket leistet
9/87 / 16-17	Textomat Amiga: Da wird die Konkurrenz ganz schön zu beißen haben
9/87 / 65	Die Urkunde: Dann erbt mal schön
9/87 / 66	The Last Warrior: Muß es sowas noch geben?
9/87 / 71	Dolmetscher: Was ist, wenn Sie Ihr Textprogramm ändern?
9/87 / 72-75	Pulsar: Eprom-Brennen kinderleicht, mit allem drum und dran
9/87 / 77-78	Amiga 500: 512 zusätzliche KByte und etliche Zugaben mehr
9/87 / 68-70	RKT Printerface: Macht endlich Schluß mit dem Druckerwirrwarr
10/87 / 4-5	Floppy 1581: Speicherriese im Miniformat
10/87 / 8-11	Vermer und Cholo: Games der Spitzenklasse
10/87 / 73	Zing: CLI-Erweiterung macht Mäuse nötig
10/87 / 79	Abenteurer: Schatzsucher für Adventure-Einsteiger
10/87 / 74-75	DFÜ DeLuxe: Tron Modem am Amiga
11/87 / 8	Stationfall: Spaß mit dem trottelligen Roboter
11/87 / 9	California Games: Ein neuer Joystick-Killer
11/87 / 20	INAD: Wie intelligent ist intelligent?
11/87 / 21-22	Druckerinterface: Wundertier
11/87 / 23-24	Seikosha SL 80 V: Ein Drucker als Tiefstapler
11/87 / 62-63	Computerfibel: Neue Möglichkeiten für den C64
11/87 / 73	Tronmodem: Auf ein neues
12/87 / 4	Super-Sprint: Autorenrennen einmal anders
12/87 / 5	Zynaps: Und noch ein Weltraumspiel, doch langweilig ist es nicht
12/87 / 6	Bubble-Bubble: Es geht auch friedlicher - ein Automaten-Spiel
12/87 / 7	Jack the Nipper II: Das böartige Baby geht wieder auf Jagd
12/87 / 18-23	RAM-Floppy auf dem Plus4: Tips zum Selbstbau usw.
12/87 / 54-57	Centronics am Plus4: Anleitung zum Selbstbau
12/87 / 49	DOS Modul: Ohne viel Umbau die 1541 beschleunigen
12/87 / 64	Bit Profi: Mit einem simplen Rechenschieber endlich durchblicken

Serie - Service & Report

Ausgabe-Seite:	Stichwort:
1/87 / 6-7	News: Software-Error beim Amiga, Neues von Activision usw.
1/87 / 9	Polen: Warten auf die Botschaft
1/87 / 10	RS 232: Die richtigen Schaltbilder
1/87 / 66	Bücher: Neuerscheinungen, für Sie gelesen
1/87 / 71-72	Kartei: Wissenswertes für die Commodores
1/87 / 73	Kunst & Computer: Kein Gegensatz!
1/87 / 75	Chips: Wie Sie Bezeichnungen entziffern

SERVICE

- 1/87 / 67-70 Börse: Gesucht und gefunden
 2/87 / 6-7 C64: Ein Millionär läßt grüßen
 2/87 / 10-11 Commodore: 1987 - ein Commodore-Festival
 2/87 / 14-15 Umbau: Script Plus als ROM im Plus4!
 2/87 / 16 Die Seite 16: Tips und Tricks auf der Sechzehnerseite
 2/87 / 19-21 Plotter 1520: Und er kann doch drucken
 2/87 / 50 Service: Drei Schablonen für die 16er Funktionstasten
 2/87 / 51-52 Karteikarten: Tips und Tricks zum Ausschneiden
 3/87 / 8 C16/Plus4: 128 Byte-Tastaturpuffer
 3/87 / 9-10 1541: Wie man eine Doppelfloppy baut
 3/87 / 12-15 RS 232: Wissen Sie, was in dieser Schnittstelle steckt?
 3/87 / 16-18 Die Seite 16: Natürlich wieder mit kleinen Beispielprogrammen
 3/87 / 19-20 Serie: Forth-Sprache und System gleichzeitig für den C64
 3/87 / 53-54 Kartei: Tips und Tricks zum Sammeln
 3/87 / 55-56 Börse: Gesucht und gefunden
 3/87 / 58-59 Neu auf dem Markt: Commodore läßt die Amigas von der Kette
 3/87 / 60-61 Turbo: Wissen Sie, wie Sie den Amiga schneller machen?
 4/87 / 4-7 Cebit 87: Die wichtigsten Neuheiten dieser Messe
 4/87 / 12-13 BASIC 7.0: Ein wirklich starkes Stück
 4/87 / 16-19 Die Seite 16: Wieder viel Wissenswertes und kleine Listings
 4/87 / 50-51 Börse: Gesucht und gefunden
 4/87 / 52-54 FORTH: Die neue Programmiersprache für den C64
 4/87 / 55-56 Kartei: Unser kleiner Aktenordner zum Ausschneiden
 5/87 / 16-19 Die Seite 16: Tips und Tricks für C16 und Plus4
 5/87 / 20-21 Serie: Forth - ein Grafikkünstler am C64?
 5/87 / 50-51 Bücher: Für Sie gelesen und bewertet
 5/87 / 53-54 Kartei: Hinweise zu allen Commodore-Computern
 5/87 / 55-56 Börse: Gesucht und gefunden
 6/87 / 9-10 Geos: Diesmal sehen wir dem Betriebssystem „unters Hemd“
 6/87 / 16-18 Die Seite 16: Neue Tricks rund um den C16 und vieles mehr
 6/87 / 24 FORTH-Serie: Die FORTH-Gesellschaft und deren Arbeit
 6/87 / 55 Daumenkino: Bilder vom Siegerlisting
 6/87 / 62 BASIC-Kurs: Disketten und Kassettenhandling beim 128er
 6/87 / 64-66 IEEE 488 Interface: Was er kann und wie er funktioniert
 6/87 / 67 Müllschlucker: Über die Garbage Collection auf Computern
 6/87 / 68-70 Börse: Gesucht und gefunden
 6/87 / 71-72 Kartei: Diesmal über Vektoren und Paßwörter
 6/87 / 74-75 BASIC-Token 128: Eine Beschreibung der 128er BASIC-Token
 7/87 / 16-19 Die Seite 16: Weiter geht es mit Listsschutz-Methoden usw.
 7/87 / 21 BASIC-Wissen: So programmieren Sie optimal
 7/87 / 22-23 DFÜ: Wissen Sie was eine NUI ist?
 7/87 / 26-27 Amiga: Zweidisk ohne Probleme - mit 5 DM sind Sie dabei
 8/87 / 4-7 Computer nach Polen: Ein Hilferuf ist angekommen
 8/87 / 10-12 Datenübertragung: So funktioniert die CA-Mailbox
 8/87 / 13-14 Geos: Im zweiten Teil geht es um die grafischen Möglichkeiten
 8/87 / 15 News & Facts: Kleine Meldung rund um die Commodore-Typen
 9/87 / 14-15 Glosse: Endlich enttarnt - die Fehlermeldungen Ihres Computers
 9/87 / 18-21 Die Seite 16: Rettungsanker für DELETE, Einblicke ins Disk-BAM
 9/87 / 22 Druckeroutine: Druckeranpassung für Datei 128
 9/87 / 22 Directory: Unser kleines Utility löscht den Speicher nicht
 9/87 / 23-27 Copy Pr. 1541: Eine randvolle Diskette in nur drei Zügen kopiert
 9/87 / 48-50 Serie: Was Sie mit dem Videochip des 128 alles anstellen können
 9/87 / 51-53 Error: Was sie bedeuten und wie man mit ihnen umgeht
 9/87 / 54 News: Rund um Commodore-Neuheiten und Wissenswertes
 9/87 / 55-59 Computerwissen: Ist er intelligenter oder nicht?
 10/87 / 15-17 Sprites im Einbau: Mehr über Programmierung der kleinen Grafiken
 10/87 / 21 Schnittstellen: Umschalten leicht gemacht
 10/87 / 22-23 Übersicht: Die verschiedenen Druckertypen im großen Vergleich
 10/87 / 24 Maschine: Wir machen es leicht, in Maschine zu programmieren
 10/87 / 25-26 Immer besser: Der Checksummer der Commodore Welt neu überarbeitet
 10/87 / 27-28 Anpassungsschwierigkeiten: Nochmals Hilfe für C16er-Besitzer
 10/87 / 35 News: Alles Neue für die Commodore-Homecomputer
 10/87 / 62-64 Für Durchblicker: Unser 128er BASIC-Kurs über Grafikbefehle
 10/87 / 65 Fundgrube: Das aktuelle Bezugsquellenverzeichnis
 10/87 / 66 Invasion: Die Mäuse kommen
 10/87 / 67 Userport durchschaut: Wie er funktioniert
 10/87 / 76-78 Quo vadis: Der Weg von Commodore auf dem deutschen Markt
 10/87 / 16 Die Seite 16: 128 KByte Erweiterung und Tastaturhilfen
 11/87 / 4-7 C64/III: Der C64 ist tot - es lebe der C64/III!
 11/87 / 12-13 News & Facts: Neues rund um die Commodore-Computer
 11/87 / 19 BASIC-Code: Byte aus dem Äther
 11/87 / 64-65 Arbeitswelt: Jobkiller Computer?
 12/87 / 9-10 BASIC 7.0: Neuer Teil des BASIC-Kurses, rund um die Floppy
 12/87 / 12-13 News von der PCW; Wir berichten von den interessantesten Spielen
 12/87 / 16-17 Schwerpunktthema Arbeitsspeicher
 12/87 / 24-28 Midi: Wie die Musikwelt mit dem C64 verbunden wird
 12/87 / 30 News & Facts: Interessantes aus der Computerwelt
 12/87 / 60-62 System: Ein Bericht über die Münchner Computermesse
 12/87 / 71 Noch lange nicht tot: Warum der C64 auch weiterhin existiert
 12/87 / 74-78 Computer-Fieber: Unsere neue Serie über die Computerfreaks

Tips & Tricks

- Ausgabe-Seite: Stichwort:
 1/87 / 59-61 Nadel/Matrixdrucker: Das gute Stück
 1/87 / 62-63 Tintenstrahldrucker: Jet Dimension für das Computerzeitalter
 1/87 / 64-65 Typenraddrucker: Wenn Schönheit über alles geht
 2/87 / 53 Von der PC-Meßbrücke bis zu Fehlern im Commodore-Handbuch
 7/87 / 30-34 Laufschrift auf dem C64: Hexa oder Dezimal
 7/87 / 40-46 Sprites: Wie Sie besser Sprites auf dem C64 programmieren
 7/87 / 78-79 VC 20: Eine Legende lebt
 8/87 / 16-18 Die Seite 16
 8/87 / 45-47 Die Sache mit dem Ü und Ä auf C16 und Plus4-Reihe
 8/87 / 54 Bücher: Für Sie gelesen und bewertet
 8/87 / 62-63 BASIC 7.0: Kennen Sie alle grafischen Befehle?
 8/87 / 65-66 Load Error: Warum plötzlich Programme nicht mehr laufen?
 8/87 / 67-70 Börse: Gesucht und gefunden

Listings

- Ausgabe-Seite: Stichwort:
 1/87 / 16-17 Inputmask: Schneller und sicherer Daten eingeben
 1/87 / 22-27 Lawine: Ein beliebtes Kugelspiel
 1/87 / 28-30 Handelskontor: Kalkulieren, Gewinn und Verlust für Kaufleute
 1/87 / 31-32 Kontenführung: Mit Geld umgehen
 1/87 / 33-38 Mafia: Schaffen Sie es, aus 200 DM Spesen 2000 zu machen?
 1/87 / 45-48 Funktionen für Mathefuchse
 1/87 / 49-49 Diskmonitor: Daten lesen, verändern und speichern
 1/87 / 55-58 Schönschrift: Unterlängen und Umlaute auf billige Drucker
 2/87 / 22-25 Galgenraten: Das bekannte Spiel für den C64
 2/87 / 26-30 Bingo: Lassen Sie mal die Kugel auf Ihrem Plus4 rollen
 2/87 / 36-39 Data Generator: Umwandeln von Maschinen-Sprachen in Data-Zeilen
 2/87 / 40-45 Lotterie: Wer hat die höchsten Karten?
 2/87 / 46-48 Kommunikation: So einfach ist es, C= Prg. in einen PC zu laden
 3/87 / 21-26 Mathehilfe: Die wichtigsten Formeln des Lebens
 3/87 / 27-29 Horse: Locken Sie Ihren Gegner in die Falle
 3/87 / 35-40 Sprite-Editor: Im Handumdrehen Sprites selbst entwerfen
 3/87 / 41-46 Stundenplan: Ob Lehrer oder Schüler
 3/87 / 47-48 Arrow: Robin Hood und seine Bogenkünste
 3/87 / 49-52 Kalah: Ein interessantes Brettspiel
 4/87 / 20-24 Rollo Buff: Das absolute Spiel, der reine Wahnsinn
 4/87 / 25-30 Loisada: Retten Sie Schiffbrüchige im Weltall
 4/87 / 38-40 Supermind: Mastermind, einmal anders und besser

SERVICE

COMMODORE SPECIAL
C64/ C128
TEST-JAHRBUCH 1988

Alles über Ihren C16-128 PC

Tips Tests Tricks

Rund 150 Seiten Einkaufsführer

Hand- und Software im Test! - Kaufberatung

COMMODORE SPECIAL für Ihren Rechner

TEST-JAHRBUCH 1988 C16

COMMODORE SPECIAL
P/4 C16 116
TEST-JAHRBUCH 1988

Tests Tips Tricks Kaufberatung

Alles über Ihren C16/P4

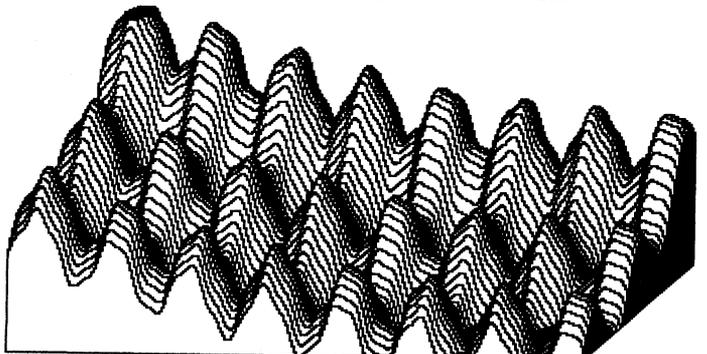
Das Handbuch für C16/P4

DAS BESTE AUS CW
 Sammelband Nr. 3/88
 DM 19,80 - ÖS 158,- SFR 19,80

- 4/87 / 42-49 *Merkut: Jackpot a la Las Vegas für zuhause*
 5/87 / 22-23 *Boolsche Wahrheit: Warum 254 plus 2 nur 255 ist*
 5/87 / 24-26 *Diskeditor: Wissen Sie, was auf Ihrer Diskette ist?*
 5/87 / 29-36 *Autokosten: Was der fahrbare Untersatz wirklich kostet*
 5/87 / 37-40 *Grafik 128: Nutzen Sie die starken BASIC-Befehle des 128*
 5/87 / 41-46 *Paradies: Schafe fressen Gras, Wölfe fressen Schafe und und und*
 6/87 / 19 *Grafik 128: Berichtigung des Grafikprogrammes (5/87)*
 6/87 / 25-29 *Music Master: Unglaublich, was der 16er für Sounds beherrscht*
 6/87 / 30-35 *Ausbrecher: Sie wollen nicht länger im Gefängnis sitzen*
 6/87 / 36-38 *Quickmac: Pacman kehrt mit neuen Gegnern zurück*
 6/87 / 46-48 *Diskmonitor: Byte für Byte kontrollieren und manipulieren*
 6/87 / 49-51 *Taxi: Bis zu vier Mitspieler kämpfen um den Sieg*
 6/87 / 52-53 *Deutscher Zeichensatz: Auf Bildschirm und Drucker*
 6/87 / 56-59 *Dia-Show: Treiberprogramm für die Diaprojektorensteuerung*
 6/87 / 59 *Fang die Maus: Ein witziges Geschicklichkeitsspiel*
 6/87 / 60-61 *Datagenerator 16: Problemstellung und Lösung*
 6/87 / 43-45 *Videodatei 128: Ordnung bringen in das Filme-Chaos*
 7/87 / 47-51 *Fenster 64: Bis zu drei Bildschirmfenster auf dem C64*
 7/87 / 52-64 *BASIC: 52 neue Befehle für Ihren C64-Supererweiterung*
 7/87 / 66-67 *Checksummer: Das gab's noch nie!*
 8/87 / 26-31 *Black Jack: 17 und 4 mal anders*
 8/87 / 24-25 *Berichtigung von SCS BASIC (Heft 7/87)*
 8/87 / 33-34 *Starcrump: Ein Weltraumspiel fordert die Geschicklichkeit*
 9/87 / 28-33 *Texter 128: Ein Programm, das mehr kann als manches Profipaket*
 9/87 / 34-38 *Telefonbuch: Dateiverwaltung für den Amiga in Amiga-BASIC*
 9/87 / 43-45 *Sensor: Wie gut ist Ihr Gedächtnis? Der Computer prüft's*
 9/87 / 46-47 *Skeet: Tontaubenschießen hört sich einfach an, aber...*
 10/87 / 29-32 *Der Drucker lernt malen: Grafik-Editor für den C16/P4-Drucker*
 10/87 / 33-35 *Schneller geht's kaum: C64-Disk in 20 Sekunden formatiert*
 10/87 / 46-54 *Unterwelt: Ein Gangsterspiel für den 128er*
 10/87 / 58-60 *Überflieger: C16/Plus4 Hubschraubersimulation*
 10/87 / 61 *Astronauten: Mit dem 16er Raketen starten*
 11/87 / 43-45 *Memory: Finden Sie die passenden Paare*
 11/87 / 46-49 *Roter Baron: Sie steuern den legendären Einsitzer*
 11/87 / 50-52 *Spider: Hundert Punkte in kürzester Zeit oder Sie sind out*
 11/87 / 53-56 *Chardesigner: Komfortabler Zeichensatzeditor*
 11/87 / 57-61 *Sprite-Scanner: Zieht Sprites aus jedem (!) Programm*
 11/87 / 61 *Checksummer: Paßt für alle Commodore-Modelle!*
 12/87 / 31-32 *Money: Ein Musikprogramm für den 128er*
 12/87 / 33-34 *Print Sprite: Sprites editieren - einmal anders*
 12/87 / 36-43 *Stradenk: Zwei Spieler stellen ihre Strategie unter Beweis*
 12/87 / 44-47 *Commando Max: Science-fiction mit dem C16 durch dieses Spiel*

Dialog

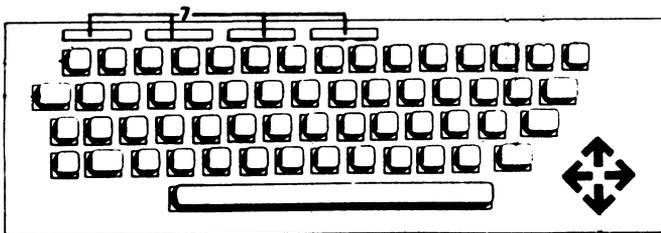
- Ausgabe-Seite: Stichwort:
- 5/87 / 9-12 *Kontenjournal auch mit 40 Zeichen, Plus4 auch MIDI-kompatibel?*
 6/87 / 11-13 *Briefe usw.*
 7/87 / 11-13 *Mit 61 zu alt zum Computern?*
 10/87 / 12-14 *Werner-Probleme, Programmierer gesucht, Scharfes Adventure usw.*
 11/87 / 10-11 *Probleme mit Textomat und DOS-Shell?*
 12/87 / 10-11 *Scrollen auf dem 128, geschlossene Fenster*



AUF DER SUCHE NACH VERLORENEN TEXTEN

Pfadfinder- Motto: Jeden Tag eine gute Tat!

Wem das Mißgeschick wiederfuhr, eingetippte Textseiten mit unvorsichtigem Hantieren von Textverarbeitungsprogrammen zu vernichten, bekommt hier einige Tips und Tricks in die Hand, das noch zu retten, was zu retten ist.



Jeder war schon mal auf der Suche, nicht nach dem verlorenen Schatz, sondern nach verloren gegangenen Texten. Wohl jeder Anwender, der häufig mit Textverarbeitungsprogrammen arbeitet, hat schon einmal einen Text versehentlich gelöscht. Zum Beispiel kann es beim Arbeiten mit Viza-write sehr leicht passieren,

WER SUCHET DATAS DER FINDET DATAS

daß aus einem sehr langen Text eine kleine Passage übernommen werden soll. Speichert der Anwender diesen jedoch ab, nachdem er den nicht benötigten Teil des Textes gelöscht hat, ohne den Namen des Files zu ändern, so steht nachher nur noch der Textausschnitt auf Diskette und der Rest des mühsam eingetippten Originaltextes ist verloren.

Auch bei einem Stromausfall während des Abspeicherns kann der Text aus dem Directory verschwunden sein. Ebenso kann es passieren, daß sich das Programm durch einen Bedienungsfehler „aufhängt“, was natürlich bei einem guten Programm nicht vorkommen sollte. Manchmal denkt man jedoch, das Programm wäre abgestürzt, weil der Computer keine Reaktion mehr zeigt, obwohl das Programm nur darauf wartet, daß man die Diskettenstation wieder einschaltet. Dies kann vorkommen, wenn man beim Eingeben schnell mal (zum Beispiel bei Viza-write) das Directory ansieht und dann die Diskettenstation abschaltet, bevor der Vorgang abgeschlossen ist. In einem solchen Fall genügt es

meist, die Diskettenstation noch einmal kurz einzuschalten.

Besonders ärgerlich ist bei einem solchen Textverlust die Tatsache, daß in den meisten Fällen der größte Teil immer noch auf der Diskette steht, aber man kann nicht mehr auf normalem Weg darauf zugreifen. Wurde der Directoryeintrag gelöscht, so kann man dies, wenn kein Schreibzugriff auf die Diskette erfolgte, mit einem Disketten-Monitor retten, indem den Filetyp (Nulltes Byte eines Directoryeintrags, siehe Floppyhandbuch) wieder restauriert (mit \$80 oder verknöpfen, zum Beispiel aus \$01 wieder \$81 für SEQ-Files).

Leider geht dies bei vielen Textprogrammen auf Grund sinnloser Schutzabfragen nicht. Einige Textverarbeitungsprogramme löschen ein File so gründlich, daß sogar die in der BAM markierten Blöcke wieder freigegeben werden. Jetzt ist die Suche nach dem verschwundenen Text natürlich um einiges schwieriger. Dem Autor dieses Pfadfinder-Programmes ist genau das Gleiche passiert. Ein Text von etwa 100 Blöcken steht irgendwo auf der Diskette und ist doch scheinbar so unerreichbar. Der Pfadfinder sucht diesen Text. Wer weiß, wie die Diskettenstation abspeichert, kommt leicht auf die nötigen Tricks, die die Daten wieder zurückholt (vgl. Handbuch zu den

DIRECTORY SPUR 18 DER SCHLÜSSEL

Diskettenstationen). Im Directory, welches sich auf der Spur 18 befindet, steht außer dem Filenamen, dessen Länge und dem Filetyp, auch die Startadresse (in Spur- und Sektornummer). In diesem Block (mit 256 Byte) steht nun am Anfang die Adresse des nächsten Blocks, der sich im Prinzip irgendwo auf der Dis-

kette befinden kann. Auf diese Weise „hangelt“ sich die Floppystation von Block zu Block, von Spur zu Spur.

Fehlt nun der Start des Files durch einen Bedienungsfehler, so kann der geübte Anwender wenigstens die restlichen Blöcke retten, wenn sie nicht bereits mit anderen Files überschrieben sind. Meist ist dies nicht der Fall, denn ein intelligentes Textverarbeitungsprogramm speichert zuerst den neuen Text vollständig ab, bevor es den alten löscht. Wenn Sie in einem solchen Notfall den Pfadfinder zur Hand haben, können Sie also fast den ganzen Text retten.

AUF DER SUCHE NACH BLÖCKEN

Der Sinn des Programms ist also folgender: Es sucht Ketten von zusammenhängenden Blöcken auf der Diskette, die noch Teile eines verschwundenen Programms enthalten. Bei der Programmstellung wurde weitgehend auf Peek und Poke verzichtet, damit es auf allen Commodore-Computern läuft. Lediglich die Abfrage der Commodore-Taste (C=) erfolgt mit Peek. Die entsprechende Adresse steht in der Variablen CBM und wird automatisch auf den Rechner eingest. (bei C128 den C64-Modus verwenden!). Mit dieser Taste kann man die gerade Untersuchung abbrechen und zum nächsten Teilschritt übergehen, was besonders in der Testphase hilfreich ist. Das Programm geht wie folgt vor:

Schritt 1:

Die Einträge im Directory werden mit Filetyp, Länge und Startadresse angezeigt. Dann werden alle im Directory angezeigten (auch gelöschte, wenn noch kein Validate durchgeführt wurde) Blöcke untersucht und als belegt gekennzeichnet. Die *Lesen Sie bitte weiter a. S. 63*

LISTING

DISK TO TEXT

```

100 rem disk to text ===== <go>
110 rem (p) commodore welt team = <eh>
120 rem ===== <gn>
130 rem (c) rudolf schmid-fabian = <cb>
140 rem heidelberg = <jh>
150 rem version 9.12 = <oo>
160 rem c64 & c16 + drucker = <bd>
170 rem ===== <cc>
180 poke56,19:clr:n=20*256:h$=chr$
(019) <ol>
190 printchr$(147):printtab(5)"ASC
II-File (1)" <lo>
200 printtab(5)"VIZA-File (2)" <ip>
210 geta$:ifa$=""then210 <nd>
220 ft=val(a$):onft+1goto190,260,2
30 <he>
230 input"Filename eines intakten
VIZA-Files";p$ <bb>
240 open2,8,2,p$+",p,r":fori=1to20
0:get#2,a$:a=asc(a$+chr$(000)):pok
en,a:n=n+1:next <nf>
250 close2:vz=1 <ph>
260 input"track,sector";tr,se <nb>
270 gosub440 <mg>
280 get#2,t$,s$:tr=asc(t$+chr$(000
)):se=asc(s$+chr$(000)):print"trac
k,sector"tr,se <op>
290 fori=3to256:get#2,a$:bc=asc(a$
+chr$(000)):poken,bc:n=n+1 <in>
300 ifvz=0then330 <ba>
310 ifbc>128thena$="" :goto330 <eg>
320 a=bc-64*(bc<32orbc>95)-32*(bc>
63andbc<96):a$=chr$(a) <fe>
330 printa$;:nexti:print <ag>
340 gosub490 <cm>
350 iftr>0thengoto270 <ah>
360 hlt=n:goto370 <bd>
370 input"neue Diskette eingelegt
(j)";d$:ifd$<>"j"then370 <mf>
380 n=20*256 <ke>
390 input"neuer Filename";p$ <gi>
400 input"Filetyp (s)eq (p)rg";ft$ <ko>
410 open2,8,2,p$+", "+ft$+",p,w":fo
ri=ntohlt:a$=chr$(peek(i)):print#2
,a$;:next <lf>
420 end <ml>
430 rem ***** unterprogramme * <ml>
440 open15,8,15,"i":open2,8,2,"#" <ho>
450 rem ***** block read * <gg>
460 print#15,"u1:"2;0;tr;se <ba>
470 rem ***** bufferpointer * <eo>
480 print#15,"b-p:"2;0:return <bc>
490 close2:close15:return <ki>
500 rem disk to text ===== <gp>

```

```

100 rem pfindfinder ===== <gn>
110 rem (p) commodore welt team = <eh>
120 rem ===== <gn>
130 rem (c) rudolf schmid-fabian = <cb>
140 rem heidelberg = <jh>
150 rem version 9.12 = <oo>
160 rem c64 & c16 + drucker = <bd>
170 rem ===== <cc>
180 rem *** track & sektor aller * <ej>
190 rem ***** programme suchen * <cm>
200 rem start 1. block directory * <ea>
210 tr=18:se=1 <ff>
220 rem *** c64 oder c16 auswahl * <ln>
230 cbm=653:ifpeek(44)>8thencbm=13
47 <pp>
240 dim na$(100),tr$(100),se$(100)
,b1$(100),ts$(35,20) <lb>
250 le$=chr$(157)+chr$(157):cl$=ch
r$(147):printchr$(014) <gm>
260 printcl$:input"Directory auf D
rucker ausgeben";q$:ifq$="j"thend=
1:open4,4 <fh>
270 printcl$"Blockadresse (track,s
ektor)"tr;se:print <ib>
280 gosub1080:ts$(18,1)=1:ts$(18,0
)=1 <fh>
290 get#2,t$,s$:tr=asc(t$+chr$(000
)):se=asc(s$+chr$(000)) <go>
300 iftr>0thents%(tr,se)=1 <bp>
310 ifdthenprint#4,"track,sector:
";tr,se <po>
320 fori=1to8:get#2,pa$,tr$,se$:e=
e+1 <dh>
330 pa=asc(pa$+chr$(0)):tr%(e)=asc
(tr$+chr$(000)):se%(e)=asc(se$+chr
$(000)) <fc>
340 iftr%(e)=0thene=e-1:gosub1130:
goto480 <lo>
350 printpa;tr%(e);se%(e); <cc>
360 n$="" :for n=3to27:get#2,a$:ifa$
=chr$(160)thena$="" <og>
370 n$=n$a$:nextn <ka>
380 na$(e)=left$(n$,16) <dl>
390 print,le$chr$(018)na$(e)chr$(1
46); <ga>
400 ifpa=0thenna$(e)=na$(e)+"* " <af>
410 get#2,b1$,bh$:b1%(e)=asc(b1$+c
hr$(0))+256*asc(bh$+chr$(0)):print
b1%(e) <ob>
420 ifdthenprint#4,pa,tr%(e),se%(e
),na$(e),b1%(e) <db>
430 get#2,w$,w$:iftr%(e)>0thennext <lm>
440 print:print"blockadresse (trac
k,sector)"tr;se:print <pp>
450 ifpeek(cbm)=2thengosub1130:got
o480 <ob>
460 gosub1130 <gh>
470 iftr>0thengoto280 <cl>

```

LISTING

```

480 close4:print:print"Taste...." <id>
490 getq$:ifq$=""then490 <bd>
500 rem spur der programme jagen * <cm>
510 printcl$ <im>
520 input"Programmspuren auf Druck
er ausgeben";q$:ifq$="j"thend=1:op
en4,4 <hi>
530 fori=1to35:cd$=cd$+chr$(017):c
r$=cr$+chr$(029):z$=z$+"0":next <kk>
540 p1$=chr$(019)+left$(cd$,23):p2
$=p1$+chr$(017) <ga>
550 printcl$;:fori=0to20:printz$+c
hr$(018)+right$(" "+str$(i),2):nex
t <kj>
560 printchr$(019)left$(cd$,21)chr
$(018)"123456789012345678901234567
89012345" <il>
570 tr=18:forse=0to20:onts%(18,se)
gosub1050:next <hn>
580 forr=1toe:tr=tr%(r):se=se%(r):
n$=na$(r) <kf>
590 printp1$chr$(018)n$chr$(146);s
tr$(tr)+str$(se)+" " <bk>
600 ifdthenprint#4,n$;tr;se <eo>
610 gosub1080:ts%(tr,se)=1:gosub10
50 <in>
620 get#2,t$,s$:tr=asc(t$+chr$(000
)):se=asc(s$+chr$(000)) <ae>
630 printp2$" "chr$(145):p
rinttr;se;:ifdthenprint#4,tr;se <ll>
640 ifpeek(cbm)=2thengosub1130:got
o680 <bn>
650 gosub1130 <el>
660 iftr>0thengoto610 <co>
670 nextr <nj>
680 printchr$(145):input"Blockbele
gung auf Drucker ausgeben";q$:ifq$
<>"j"then710 <nm>
690 fors=0to20:fort=1to35:print#4,
mid$(ts%(t,s),2);:nextt <ao>
700 print#4,right$(" "+str$(s),4
):nexts <gi>
710 close4 <jg>
720 rem spuren suchen, die nicht * <kj>
730 rem **** im directory stehen * <ng>
740 printcl$:print:print"Jetzt wir
d auf zusammenhaengende Blocks" <ie>
750 print"untersucht, die nicht im
Directory" <hc>
760 print"stehen.":print:print"Pru
efmoeglichkeiten:":print" alle B
locks (a)" <fd>
770 print" nur belegte Blocks
(b)" <cc>
780 getq$:ifq$<"a"orq$>"b"then780 <me>
790 ifq$="b"thenb=1:gosub1150 <aj>
800 printcl$:input"auf Drucker aus
geben";q$:ifq$="j"thend=1:open4,4 <nb>
810 printcl$:input"Start-Track (1-
35)";q$:ta=val(q$):if(ta<1)or(ta>3
5)then810 <bj>
820 fort=tato35 <hm>
830 m=20:ift>17thenm=18:ift>24then
m=17:ift>30thenm=16 <nd>
840 fors=0tom:p=1:printcl$"t,s:"t,
s <da>
850 ifpeek(cbm)=2thent=35:s=m:goto
1000 <dk>
860 ifts%(t,s)=1then1000 <pi>
870 ifbthenifb%(t,s)=0then1000 <fd>
880 ts%(t,s)=1:tr=t:se=s <pe>
890 gosub1080 <gd>
900 get#2,t$,s$:tr=asc(t$+chr$(000
)):se=asc(s$+chr$(000)) <df>
910 iftr=0ortr>35thengosub1130:got
o1000 <ia>
920 ifse>20thengosub1130:goto1000 <ad>
930 ifts%(tr,se)=1thengosub1130:go
to1000 <io>
940 ts%(tr,se)=1 <jb>
950 ifpthenp=0:printt;s,tr;se:ifdt
henprint#4,t;s;tr;se:goto980 <fg>
960 print,tr;se:ifdthenprint#4,spc
(7)tr;se <om>
970 ifpeek(cbm)=2thengosub1130:got
o1010 <km>
980 gosub1130 <ki>
990 iftr>0then890 <jm>
1000 nexts <af>
1010 nextt:close4 <mp>
1020 printcl$:input"Einzelspuren v
erfolgen";q$:ifq$="j"then1440 <md>
1030 end <je>
1040 rem ***** cursorposition * <fj>
1050 p$=chr$(19)+left$(cr$,tr)+rig
ht$(cd$,se) <pl>
1060 printp$"1":return <hl>
1070 rem *** gem. unterprogramme * <kl>
1080 open15,8,15,"i":open2,8,2,"#" <nk>
1090 rem ***** block read * <bl>
1100 print#15,"u1:"2;0;tr;se <lc>
1110 rem ***** bufferpointer * <pf>
1120 print#15,"b-p:"2;0:return <nh>
1130 close2:close15:return <dm>
1140 rem bam einlesen & anzeigen * <jd>
1150 fori=1to40:cr$=cr$+chr$(029):
cd$=cd$+chr$(017):next:ho$=chr$(01
9) <gc>
1160 dim a%(143),b%(35,20) <de>
1170 tr=18:se=0:gosub1080 <al>
1180 fori=0to143:get#2,a$:a%(i)=as
c(a$+chr$(000)):next:gosub1130:a=4 <nm>
1190 printchr$(147)left$(cd$,22)"0
0000000011111111112222222222333333
" <gd>
1200 print"12345678901234567890123

```

LISTING

```

456789012345" <bd>
1210 printho$left$(cd$,6):print"
      BBB      A      MM      MM <mc>
1220 print"      B B      A A <am>
      M M M M
1230 print"      B B      A A <oe>
      M M M
1240 print"      BBB      AAAAA <fh>
      M M M
1250 print"      B B      A A <ih>
      M M
1260 print"      B B      A A <gn>
      M M
1270 print"      BBB      A A <ne>
      M M
1280 print" <gj>
1290 fort=1to35:z=1:x=0 <pc>
1300 m=20:ift>17thenm=18:ift>24the
nm=17:ift>30thenm=16 <gp>
1310 fors=0tom <ao>
1320 w=-((a%(a+z)and2^x)=0):w$=mid
$(str$(w),2):b%(t,s)=w <ij>
1330 printho$+left$(cr$,t-1)+left$
(cd$,s)+w$ <mg>
1340 x=x+1:ifx<8then1360 <dc>
1350 x=0:z=z+1 <de>
1360 nexts:a=a+4:nextt <cc>
1370 getq$:ifq$=""then1370 <kd>
1380 printcl$:input"BAM auf Drucke
r ausgeben";q$:ifq$<>"j"then1420 <mk>
1390 open4,4:fors=0to20:fort=1to35
:print#4,mid$(b%(t,s),2);:nextt <lj>
1400 print#4,right$(" " +str$(s),
4):nexts <hf>
1410 print#4:print#4"1234567890123
4567890123456789012345":close4 <di>
1420 return <lb>
1430 rem ** einzelne spuren jagen
* <ni>
1440 printchr$(147) <ig>
1450 input"Start (Track,sector)";t
r,se <jp>
1460 iftr<fortr>35then1440 <lf>
1470 ifse>20then1440 <hl>
1480 gosub1080:get#2,t$,s$:tr=asc(
t$+chr$(000)):se=asc(s$+chr$(000)) <pp>
1490 printtr;se <kp>
1500 ifpeek(cbm)=2thengosub1130:go
to1450 <ec>
1510 gosub1130 <cg>
1520 iftr>0thengoto1480 <ip>
1530 goto1450 <hn>
1540 rem pfinder ===== <eo>

```

Achtung in Zeile 1320 bedeutet das Zeichen ^ = Pfeil nach Oben.

BLOCKREAD ALL

```

100 rem blockread ===== <eb>
110 rem (p) commodore welt team = <eh>
120 rem ===== <gn>
130 rem (c) rudolf schmid-fabian = <cb>
140 rem heidelberg = <jh>
150 rem version 1.7 = <kp>
160 rem c64 & c16 + drucker = <bd>
170 rem ===== <cc>
180 rem * c64 oder c16 (c=taste) * <lm>
190 cbm=653:ifpeek(44)>8thencbm=13
47 <pa>
200 printchr$(147):printtab(5)"ASC
II-File (1)" <cd>
210 printtab(5)"VIZA-File (2)" <dd>
220 geta$:ifa$=""then220 <gp>
230 ft=val(a$):onft+1goto200,260,2
50 <fm>
240 rem ** flagge fuer vizafiles * <de>
250 vc=1 <pl>
260 input"start-track,start-sector
";ta,sa <ie>
270 input"zeichen pro track (max.
256)";zt <eh>
280 input"Druckerausgabe (j)";q$
:ifq$="j"thend=1:open4,4 <go>
290 fortr=tato35:forse=sato20 <hg>
300 gosub440:z$="" <gc>
310 get#2,t$,s$:print"track,sector
"tr,se <ko>
320 fori=3tozt:get#2,a$:a=asc(a$+c
hr$(000)) <ia>
330 ifvz=0then360 <nk>
340 ifbc>128thena$="" :goto360 <jn>
350 a=bc-64*(a<32ora>95)-32*(a>63a
nda<96):a$=chr$(a) <ao>
360 if(aand127)<32thena$="*" <ci>
370 z$=z$+a$:nexti <fb>
380 printz$:ifdtheniflen(z$)>1then
print#4,tr,se:print#4,z$ <im>
390 gosub490 <fg>
400 ifpeek(cbm)=2thentr=35:se=20 <op>
410 nextse:nexttr:close4 <fi>
420 end <ml>
430 rem ***** unterprogramme * <ml>
440 open15,8,15,"i":open2,8,2,"#" <ho>
450 rem ***** block read * <gg>
460 print#15,"u1:"2;0;tr;se <ba>
470 rem ***** bufferpointer * <eo>
480 print#15,"b-p:"2;0:return <bc>
490 close2:close15:return <ki>
500 rem blockread ===== <jl>

```

Fortsetzung von Seite 59

Blockfolgen und alles andere kann auf Wunsch auf einem Commodore-Drukker (#4) ausgegeben werden. Befindet sich das gesuchte File darunter, so kann man sich die Startadresse (Spur, Sektor) notieren und die weiteren Punkte überspringen.

Schritt 2:

Alle Blöcke, die noch nicht mit Directory-Programmen belegt sind, werden auf Zusammenhang geprüft. Hierbei kann man wählen, ob man alle Blöcke, oder nur solche, die im BAM noch als belegt gekennzeichnet sind, untersuchen will und ab welchem Track man suchen will. Hierbei ist es wichtig, zu wissen, daß die Diskette immer von innen nach außen beschrieben wird, also beginnend bei Spur 17 zur kleineren Spurnummer oder bei Spur 19 zur größeren Spurnummer. Hat man also eine relativ neue Diskette, auf der nur wenige Files stehen, ist es sinnvoll, zum Beispiel bei Spur 16 zu beginnen, bei sehr vollen Disketten bei Spur 1.

Schritt 3:

Jetzt befinden sich mehrere Blockketten, die zusammenhängen, als Ausdruck auf dem Druckerpapier. Der Anwender kann bei beliebigen Blocks anfangen und sich anzeigen lassen, wie weit diese Kette geht. Dabei ist meist der Block der eigentliche Anfang, der am nächsten an Track 18 (Inhaltsverzeichnis) liegt.

Schritt 4:

Mit dem Hilfsprogramm Blockread kann man dann noch die Blöcke lesen (Bildschirmcode wie Vizawrite oder ASCII-Code wie Easyscript/Superscript/Scriptplus). Zudem besteht die Wahlmöglichkeit, wieviele Zeichen pro Block angezeigt werden sollen. Auf diese Weise kann bei den zusammenhängenden Blöcken der eigentliche Textanfang gefunden werden.

Schritt 5:

Dieser fünfte Schritt ist natürlich der wichtigste. Nachdem der Anwender sich versichert hat, bei welcher Adresse auf der Diskette der Text anfängt, kann er mit dem Programm Disk To Text den Text von der Diskette in den Speicher einlesen. Hierbei wird bei Vizawrite-Files, bzw. Textfiles von anderen Textverarbeitungsprogrammen, die mit Bildschirm-Code arbeiten, in ASCII-Code umgewandelt.

Dies hat eine praktische Vereinfachung zur Folge, wenn das Text-File dann wieder weiterverarbeitet wird. Fast jedes moderne Textverarbeitungssystem hat die Möglichkeit, ASCII-Files einzulesen, auch dann, wenn es normalerweise mit Bildschirm-Code arbeitet.

Beim Textverarbeitungsprogramm Vizawrite wird dieses mit dem Merge-Kommando bewerkstelligt, wobei anstelle der Seitenzahlen ein 'S' eingesetzt werden muß. Andernfalls wird es bei Vizawrite nämlich etwas schwierig, die Textfiles wieder zu restaurieren, da in den ersten 200 Byte wichtige Informationen, wie Seitenzahl und Textlänge stehen und diese Defaults meist verloren sind. Ohne diese Informationen ist jedoch, ein normales Einlesen von Vizawrite-Files nicht mehr möglich.

Viel Spaß beim Experimentieren, aber mit der nötigen Sorgfalt, am Anfang nur mit den Sicherheitskopien oder mit Testfiles rumprobieren, es könnte auch einmal etwas schief gehen beim Manipulieren von Disketten-Einträgen.

SICHERHEITSKOPIE EIN MUSS

Also, die erste und wichtigste Regel beim Arbeiten mit Textverarbeitungsprogrammen lautet: Sicherheitskopien, Sicherheitskopien, Sicherheitskopien...

RSF/CW-Team

FIND – AUF VARIABLENSUCHE

Minen - Hund

Ein Minenhund soll Minen finden. Das ist auch die Aufgabe von FIND beim Suchen von Variablen (Numerisch oder Strings). Eine Doppelvergabe von Variablennamen soll endgültig der Vergangenheit angehören und den Programmablauf nicht mehr stören.

Das Interessanteste, was Sie mit Ihrem Computer anstellen können, ist ohne Zweifel, eigene Programme mit ihm zu basteln.

In den meisten Fällen wird das in BASIC geschehen. Allerdings hat das BASIC 2.0 des C64 einen nicht zu unterschätzenden Nachteil im Vergleich zu anderen BASIC-Dialekten. Die im selbstgebauten Programm verwendeten Variablennamen (numerisch oder Strings) dürfen nach ihrer Definition nur einmal verwendet werden. Sonst verlieren sie innerhalb des Programmes ihren ursprünglich zgedachten Wert. Vor allem bei numerischen Variablen kann das verheerende Folgen für den Programmablauf haben, wenn etwa der Name ,A'

in einem Unterprogramm mit einer ganz anderen Funktion und einem völlig neuen Wert eingesetzt wird.

NOTIZEN SIND GUT – FIND IST BESSER

Spätestens bei Zeile 100 der komfortablen Dateiverwaltung, die Sie gerade programmieren wollen, wissen Sie nicht mehr genau, ob Sie die neu zu definierende Variable 'AS', die Sie für Ihr Unterprogramm zum Sortieren brauchen, nicht vielleicht doch schon vorher in den bisher vorliegenden Programmzeilen gebraucht haben, stimmt's? Sie haben jetzt drei Möglichkeiten:

1. Sie durchsuchen sämtliche Zeilen Ihres halbfertigen Programms. Das ist mühsam.

FIND . 64

```

10 rem find.64                                <ni>
11 rem by commodore welt/h.b.                <ag>
12 dimhx(75):fori=0to9:hx(48+i)=i:           <bi>
   hx(65+i)=i+10:next
13 fori=49152to49494:readmc$:hi=asc(         <ld>
   left$(mc$,1)):lo=asc(right$(mc$,
   1))
14 dz=16*hx(hi)+hx(lo):pokei,dz:ne          <ib>
   xt
15 sys49152:new                               <ck>
16 data 78,a2,00,86,62,a2,bf,86             <be>
17 data 63,a0,ff,a2,37,86,01,b1             <ki>
18 data 62,a2,34,86,01,91,62,88             <ma>
19 data c0,ff,d0,ef,c6,63,a5,63             <po>
20 data c9,9f,d0,e5,a2,ff,86,63             <ai>
21 data a0,ff,a2,37,86,01,b1,62             <dk>
22 data a2,34,86,01,91,62,88,c0             <ec>
23 data ff,d0,ef,c6,63,a5,63,c9             <fj>
24 data df,d0,e5,a2,37,86,01,a9             <pg>
25 data 60,8d,12,a7,a9,58,8d,08             <hg>

```

TIPS & TRICKS

```

26 data 03,a9,c0,8d,09,03,58,60 <hf>
27 data 20,73,00,c9,46,f0,06,20 <id>
28 data 79,00,4c,e7,a7,20,73,00 <ka>
29 data c9,49,f0,03,4c,5f,c0,20 <fo>
30 data 73,00,c9,4e,f0,03,4c,5f <fb>
31 data c0,20,73,00,c9,44,f0,03 <jj>
32 data 4c,5f,c0,20,73,00,20,8d <jd>
33 data c0,20,74,a4,60,c9,22,f0 <gc>
34 data 12,a0,00,84,f7,99,58,c1 <op>
35 data c8,20,73,00,d0,f7,99,58 <ii>
36 data c1,f0,1c,a9,ff,85,f7,a0 <pe>
37 data 01,b1,7a,c9,22,f0,06,99 <gg>
38 data 57,c1,c8,d0,f4,a9,00,99 <nb>
39 data 57,c1,20,73,00,d0,fb,45 <ja>
40 data 2b,85,5f,a5,2c,85,60,a0 <km>
41 data 01,b1,5f,d0,01,60,a2,00 <of>
42 data 86,f8,a0,04,b1,5f,f0,28 <ko>
43 data c9,22,d0,06,a9,ff,45,f8 <ib>
44 data 85,f8,a5,f7,c5,f8,d0,13 <om>
45 data b1,5f,dd,58,c1,d0,0c,e8 <lh>
46 data bd,58,c1,d0,08,20,0f,c1 <nd>
47 data 4c,c7,c0,a2,00,c8,d0,d4 <mo>
48 data a0,00,b1,5f,aa,c8,b1,5f <bk>
49 data 86,5f,85,60,4c,c7,c0,a9 <lm>
50 data 8d,20,d2,ff,a0,02,b1,5f <jg>
51 data aa,c8,84,49,b1,5f,20,cd <kh>
52 data bd,a9,20,20,d2,ff,a4,49 <mh>
53 data c8,84,49,84,0f,a9,36,85 <dg>
54 data 01,20,03,a7,a9,37,85,01 <ca>
55 data 20,87,ea,a5,cb,c9,39,f0 <fi>
56 data 04,20,2c,a8,60,20,87,ea <nm>
57 data a5,cb,c9,40,f0,f7,c9,39 <pm>
58 data f0,f3,a9,00,85,c6,60 <bh>

```

2. Sie machen sich jedesmal auf einem Zettel handschriftlich eine Notiz, wenn Sie einen neuen Variablennamen für Ihr Programm verwenden (sehr lobenswerte Methode, doch im Programmierfeifer hat das schon so mancher vergessen und ist dann ebenfalls gezwungen, zu Punkt 1 zurückzukehren).
3. Die nervenschonendste Methode: Sie tippen sich das Listing 'FIND.64' ab und starten es nach dem Speichern.

NEUE BASIC-ANWEISUNG: FIND

Obwohl Sie einige Sekunden warten müssen, bis die Daten eingelesen und an den vorgesehenen Speicherplatz gePOKEd wor-

den sind (von 49152 bis 49494) lohnt sich die Warterei. Ihr C64 kennt jetzt nämlich auch den Befehl Find (den gibt's nicht einmal im Super BASIC 7.0 des C128!). Wozu der gut sein soll? Probieren geht über studieren. Wollen Sie jetzt zum Beispiel wissen, ob Sie den Variablennamen 'A\$' in Ihrem Dateiprogramm, das Sie grad bearbeiten, schon verwendet haben, so genügt die Eingabe:

FIND A\$

(die Leerstelle nach Find ist nicht unbedingt nötig, sie kann auch weggelassen werden). In Windeseile listet Ihr C64 auf dem Bildschirm alle Programmzeilen auf, in denen 'A\$' (oder jeder von Ihnen gewünschte Suchbegriff vorkommt. Eins ist allerdings wichtig:

Forschen Sie nach ausgeschriebenem Zeichenketten, die in Anführungszeichen stehen müssen, so sind diese „Gänsefüßchen“ auch beim Find-Befehl zwingend, also zum Beispiel

FIND "HALLO!"

Suchen Sie nach Anweisungen oder Befehlen des C64, so brauchen Sie Anführungszeichen nicht zu setzen.

FIND POKE listet alle Zeilen, in denen eine POKE-Anweisung enthalten ist.

Falls sich nach einer Find-Anweisung der Computer wieder mit 'READY' meldet, ohne irgendwelche Zeilen aufgelistet zu haben, so ist entweder der gesuchte Begriff in keiner der Programmzeilen enthalten (darauf können Sie sich hundertprozentig verlassen) oder Sie

haben den Suchbegriff nicht ordnungsgemäß angegeben (etwa keine Anführungszeichen gemacht, obwohl der Begriff im Programm darin eingekleidet ist, oder umgekehrt). Auch hier kommt's eben auch auf Genauigkeit an, das ist im Umgang mit Computern leider unerlässlich.

RESET-GESCHÜTZT

Die Tastenkombination STOP/RESTORE kann dieser Mini-BASIC-Erweiterung nichts anhaben, auch prinzipiell kein 'RESET' (software- oder hardwaremäßig). Sie müssen dann lediglich durch die Direkteingabe von 'SYS 49152' diese Routine wieder initialisieren. Bestimmt wird Ihnen 'FIND.64' gerade als Programmierhilfe noch gute Dienste leisten. *hb*

CLINCH – Club Of Intercommunication For Computer- Hard- and Software

Wir sind 15 Mitglieder, die sich mit folgenden Rechnern befassen: Atari 400–130 XE Atari 260–1040 ST Commodore C128 MS-DOS-kompatiblen Rechnern.

Unser Ziel und Zweck ist die Hilfestellung bei der Benutzung der Ataris und anderen Rechnern, Kontakt mit anderen User-Clubs im In- und Ausland, Anstauschen von Tips und Tricks für den Computer und Game Helps.

Unser Club trifft sich monatlich im Vereinslokal. Clubsatzungen und Clubstatuten werden jedem Mitglied ausgehändigt. Eine Clubzeitung erscheint zweimonatlich mit den Themenkreisen Tips und Tricks, Hard- und Software, Buchbesprechungen, Gebrauchtwaren An/Verkauf. An Hardware-Erweiterungen halten wir ab 03/88 bereit:

- Eine I/O-Karte mit acht Ein- und Ausgängen zum Steuern von kleinen Anlagen, zum Beispiel einer Eisenbahn. Diese Karte wird in den Modul-Schacht gesteckt, so daß die Joystick-Eingänge frei bleiben;
 - Brennen von Micro-Chips (acht Bit)
- An Software halten wir bereit:
- Eine Programmbibliothek aus 750 Public-Domain-Programmen für unsere Atarianer;
 - etwa 50 Programme für unsere ST-Freaks;
 - etwa 30 Disk für unsere C128-Freaks;
 - und einige PD-Soft für MS-DOS-Rechner.
- Weitere Informationen erhalten Sie durch den Clinch Computer Club Am Eichelgärtchen 31 D-5401 Halsenbach/Hunsrück oder Clinch Computer Club c/o Josef Schumacher Hauptstraße 62 D-5592 Klotten/Cochem
- P.S. Der Clubbeitrag beträgt 5,- Mark im Monat. Rückporto nicht vergessen. □

BAD CAT

OLYMPIADE DER KATZE

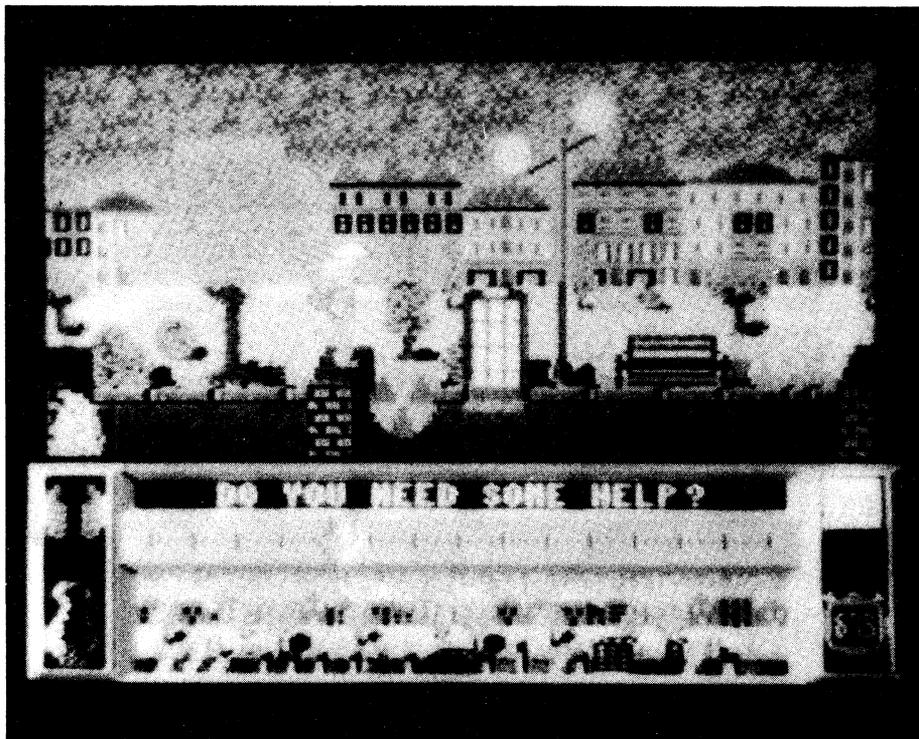
Warum immer nur eine Olympiade für Menschen? Auch die lieben kleinen Kätzchen wollen mal Champion werden.

Aufgrund der etwas eigenwilligen Teilnehmer sind auch die Disziplinen sehr exotisch. Veranstaltet wird die ganze Katzenmeisterschaft nicht in einem Stadion, sondern an verschiedenen Plätzen einer Großstadt. So heißt es zum Beispiel in einer Disziplin, schlicht und einfach den Stadtpark zu durchqueren, während die andere einen Kampf gegen den Erzfeind aller Katzen, den Hund, fordert. Das ist aber nicht etwa eine Schlägerei, sondern ein Kegelmassaker in einer Kneipe. Dazu jedoch später.

EXOTISCHE DISZIPLINEN

Insgesamt sind es vier Orte, in denen die recht fremdländisch anmutenden Wettkämpfe ausgetragen werden. Nebenbei gibt es auch noch ein kleines Zwischenspiel der Spieler, in dem der Spieler seinen Kater auf einem Motorrad durch die Stadt lotsen muß. Das hätte man sich aber ruhig sparen können.

Hier gilt es nur, zahlreichen Hindernissen auszuweichen, um so schnell wie möglich zu einem bestimmten Punkt zu gelangen. Betrachten wir die Disziplinen mal genauer: Im Stadtpark geht es darum, zuerst eine Reihe von kleinen Teichen zu überwinden, bevor man sämtliche Mauern hinter sich bringen kann. Erst dann wird's wirklich schwierig. Jetzt heißt es nämlich, den Kater auf einem Zirkusball durch die Gegend zu balancieren. Wie das zu schaffen ist, wissen bislang nur die Autoren des Spiels. Hat der Spieler das geschafft – was unserem Tester nie gelang und ihn fast zur Verzweiflung trieb – so stellt sich dem Kätzchen ein niedriges Gerüst in den Weg. Wie das aussieht, konnten wir ebenfalls leider nie erfahren. Anschließend kämen noch eine Reihe weiterer Hindernisse, unserer Meinung nach wird der Spieler aber schon Probleme haben, allein das Gerüst zu erreichen. In der Arena, oder sagen wir besser: im Schwimmbad, hat der Kater die Aufgabe, von einem auf dem Wasser schwimmenden „Bumper“ zum anderen zu springen und, wenn mög-



lich, noch Figuren von einer Sensorleiste mitzunehmen. Das ist zum Glück nicht ganz so schwierig wie der Balanceakt im Park und somit durchaus zu schaffen.

Vom Swimmingpool in den Untergrund, auch Kanalisation genannt: Hier hat der Bildschirmheld eine kleine Expedition durch die Kloake der Stadt hinter sich zu bringen. Ratten, Fledermäuse und vor allem auch Wasser sind im Weg. Das Überwinden der lieben Tierchen ist nicht so arg schwierig, das Wasser bildet das Haupthindernis. Denn der Spieler muß es, wie sollte es anders sein, mit Hilfe eines Fasses bewältigen. Die Faßreise endet auch hier wieder mit der Verzweigung des armen Spielers, der ständig, ohne zu wissen warum, ins kalte Naß plumpst. Anschließend gibt es noch eine Hangelparty an den unterirdischen Rohren und dergleichen mehr.

Zur Erholung begibt sich unser Kater jetzt in seine Stammkneipe. Hier erwartet ihn eine kleine Kegelschlacht gegen einen Hund. Ziel dieses Abschnitts ist es, sich gegenseitig umzukegeln. Ist einer der beiden getroffen, so muß er, um den Schock zu überwinden, sofort an der Bar einen runterkippen. Hat das Kätzchen zu viele Schocks erlitten, so kippt es selber um – infolge des vielen

Biers natürlich. Die Kegelparty ist im Grunde genommen ein reines Glücksspiel, auf das der Spieler fast keinen Einfluß hat.

Nun zur Grafik: Sie wäre im Grunde genommen gar nicht so schlecht – wenn man bloß etwas erkennen könnte. Zum Teil ist das Katzen-Sprite etwas plump geraten. Der Hinter- bzw. Vordergrund ist jedoch fast immer dermaßen farbüberladen, daß der Spieler manchmal Mühe hat, seine Spielfigur zu erkennen.

Deren Animation ist recht gut gelungen. Auch die anderen Sprites werden schön bewegt. Viel davon wird aber durch die schlechte Farbgebung wieder zunichte gemacht. Manche Disziplinen stellen an den Spieler recht wenig Anforderungen. Einige Hindernisse sind kein Problem, während man bei anderen nur viel Glück braucht.

FAZIT

Bad Cat ist teilweise unspielbar schwierig. Dem Spieler wird dadurch der ganze Spielspaß verdorben, was sicherlich zur Folge hat, daß das Spiel schnell in der Schublade landen wird. Es ist nur ausgesprochenen Geschicklichkeitsfans zu empfehlen, die sich durch gar nichts aus der Ruhe bringen lassen. (mn)

**COMMODOR
WEIT**

SPECIAL

CW-SPECIAL

**P/4
C16
116**

Sonderheft 5/88

DM 14,80/ÖS 124/SFR 14,80

**Ein
Megabyte
Speicher
im P4 & C16**

**Mehr
Tips &
Tricks**

**JETZT AN
IHRM KIOSK**

**Rund 150 Seiten
für Ihren P4 und C16!**

<p>C16/116 PLUS4 SYS 56364</p> <p>KEY-Taste vom Programm aus aufrufen</p> <p>Die Tastaturpuffermethode ist besonders interessant, wenn man im Programm den MONITOR nutzen will. Meist werden die nötigen Befehle genauso wie auch im Direktmodus auf den Bildschirm geprintet, der Cursor auf den ersten Befehl positioniert und mehrere RETURN-Codes (13) in den Tastaturpuffer geschrieben. Man kann auch die ganze Befehlsfolge auf eine KEY-Taste legen und diese mit 'POKE2035,K:SYS56364' ausführen lassen. K ist dabei die Tastennummer -1.</p>	<p>C16/116 PLUS4 unsichtbare Eingabe</p> <p>Password unsichtbar abfragen</p> <p>Bei der Abfrage eines Password sollte dieses auf dem Schirm nicht sichtbar sein. Dies kann man mit einer GET-Schleife oder dem WAIT-Befehl machen:</p> <pre>10 poke 239,0:wait239,4:print"password:"rem 4 zeichen 20 geta\$,b\$,c\$,d\$:p\$=a\$+b\$=c\$+d\$</pre> <p>Danach wird p\$ mit einem gespeicherten Password verglichen.</p>						
<p>C16/116 PLUS4 PEEK(1347)</p> <p>Flagge für SHIFT, CTRL und C=</p> <p>In der Speicherstelle 1347 wird angezeigt, ob eine oder mehrere dieser Tasten gedrückt ist. Mit PEEK(1347) kann man kontrollieren, welche dieser Tasten gedrückt sind:</p> <pre>PEEK(1347)=1: SHIFT-Taste gedrückt PEEK(1347)=2: C=-Taste gedrückt PEEK(1347)=4: CTRL-Taste gedrückt</pre> <p>Werden zwei oder drei dieser Tasten gedrückt, addieren sich die Werte.</p>	<p>C16/116 PLUS4 Gerätenummer</p> <p>Die Bedeutung der Gerätenummer</p> <p>Die zweite Zahl hinter der Kanalnummer beim OPEN-Befehl ist die Nummer des Geräts, von dem Daten geholt oder zu dem Daten gesendet werden sollen. Die Zahlen entsprechen folgenden Geräten:</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">0: Tastatur</td> <td style="width: 50%;">1: Kasette</td> </tr> <tr> <td>3: Bildschirm</td> <td>4: Drucker (auch 5 mögl.)</td> </tr> <tr> <td>8: Floppy (auch 9 bis 15)</td> <td></td> </tr> </table> <p>Normal für Eingabe mit GET ist 0 und für Ausgabe mit PRINT ist 3. Mit CMD kann die Ausgabe umgelenkt werden (zum Beispiel auf den Drucker).</p>	0: Tastatur	1: Kasette	3: Bildschirm	4: Drucker (auch 5 mögl.)	8: Floppy (auch 9 bis 15)	
0: Tastatur	1: Kasette						
3: Bildschirm	4: Drucker (auch 5 mögl.)						
8: Floppy (auch 9 bis 15)							
<p>C16/116 PLUS4 File-Test</p> <p>Test, ob File auf Diskette vorhanden</p> <p>Werden im Programm Diskettenfiles angesprochen, so muß vorher getestet werden, ob das File vorhanden ist, da das Programm sonst abstürzen kann, wenn es nicht gefunden wird.</p> <p>Beispiel:</p> <pre>10 input"filename,typ (p,s,u)";p\$,t\$:gosub1000:ifft>0then10 20 end 1000 open8,8,8,p\$+",""+t\$=","r":close8 1010 ft=er:print err\$(er),ds\$:return</pre>	<p>C16/116 PLUS4 Bildschirmabschalten</p> <p>Basic schneller durch Bildschirm abschalten</p> <p>Mit POKE 65286,11 kann man den Bildschirm abschalten. Dadurch kann der Computer die für den Bildaufbau nötige Zeit für Berechnungen oder das Erstellen von Grafiken verwenden. Wiedereinschalten mit POKE 65286,27.</p>						
<p>C16/116 PLUS4 Hardcopy</p> <p>Bildschirminhalt auf dem Drucker ausgeben</p> <pre>63200 rem **** hardcopy ***** 63210 open4,4:open3,3:printchr\$(19): 63220 forz=1to25:b\$="" :k=40:fors=1tok:get#3,a\$:b\$=b\$+a\$:next s 63230 ifmid\$(b\$,k,1)="" "think=k-1:b\$=left\$(b\$,k) :ifk>0then63230 63240 print#4,b\$:nextz:close3:close2:return</pre> <p>Die Routine wird im Programm mit GOSUB 63210 gestartet.</p>	<p>C16/116 PLUS4 gerade oder ungerade</p> <p>Paritätsprüfung von Zahlen</p> <p>Vor allem in mathematischen Programmen muß man oft unterscheiden, ob eine Zahl gerade oder ungerade ist. Dies kann auf einfache Weise mit der AND-Funktion geprüft werden. X AND 1 ergibt 0 bei gerade und 1 bei ungerade.</p> <pre>10 def fn odd(x)=xand1 20 input"zahl";z 30 on fnod(z)+1 goto 40,50 40 print"zahl gerade":goto10 50 print"zahl ungerade":goto10</pre>						

Mit INPUT geht es ebenfalls, nur muß man hier die Zeichenfarbe der Hintergrundfarbe anpassen. Dazu wird der Farbcode des Hintergrundes in den Tastaturpuffer gepOKEt. Ein Anführungszeichen (34) erlaubt es, beliebige Zeichen einzugeben.

```
30 t=1319:p=239:printchr$(144)
40 color 0,6
50 poket,31:poket+1,34:pokep,2:rem blue+
60 input"password":p$:printchr$(144)
70 if p$<>"xyz"thenprint"falsch":goto50
80 print"ok"
```

Beispiel (Zeichensatz nach \$3800):

```
100 fori=1 to 8:keyi,"":next:r$=chr$(13)
110 a$="p055,0:p056,56:cL:m0"=r$+"td000 d7ff
3800"=r$+"x"+r$
120 a$=a$+"p065298,192:p065299,56:p0740,56"=r$+"g0
140"=r$:key1,a$:printchr$(147)
130 poke2035,0:sys56364:end:rem key1
140 sys62359:rem key normal
150 key1,"p065298,196:p065299,208:p0740,208"=r$
:rem Zeichensatz normal mit key1
160 printchr$(147)"das programm wird fortgesetzt"
```

Beispiel 1 Floppy = 8 (Startadresse):

```
20 input"programmname":p$:open8,8,8,p$+"p,r":get
#8,a$,b$:close8
30 print"startadresse:"asc(a$+chr$(0))+256*asc(b$+
chr$(0))
```

Beispiel 2 Tastatur als Eingabegerät (RETURN):

```
40 open1,0:input#1,q$:close1
```

Beispiel 3 Bildschirm als Eingabegerät:

```
50 open3,3:printchr$(19)::fors=1 to 40:
60 get#3,a$:b$=b$+a$:next:close3:printb$
```

Beispiel 4 Listing als SEQ-File:

```
open8,8,8,"prgname,s,w":cmd8:list
print#8:close8
```

Anwendung:

Man kann mit diesen Tasten eine einfache Tastenabfrage machen:

```
10 print ch(147)ti$:if peek(1347)=0 then 10
```

Besonders wichtig ist die Speicherstelle, wenn man die Funktionstasten mit mehr als acht Werten belegen will. Man fragt dann zuerst die Speicherstelle 1347 ab, bevor man die Funktionstasten abfragt.

Dieses kleine Programm zeigt den Zeitunterschied für eine Berechnung mit und ohne Bildschirmabschalten. Es geht 33 Prozent schneller.

```
10 poke65286,11:ti$="000000":fori=1 to 200:a=sin
(i*i):next:printi:poke65286,27:
20 ti$="000000":fori=1 to 200:a=sin(i*i):next:
printi
```

Erklärung:

In Zeile 1000 wird das File zum Lesen geöffnet und gleich wieder geschlossen. Da kein Schreibversuch unternommen wurde (PRINT#8,...) blinkt bei nicht vorhandenem File nur die Kontrolllampe an der Floppy. In Zeile 1010 wird der Fehlerkanal ausgelesen. In 'er' steht dann der Fehlertyp (zum Beispiel 62 bei 'FILE NOT FOUND') und in 'errS(er)' der Text.

Beispiel:

Prüfung auf größer, kleiner oder gleich Null:

```
10 input"zahl":z
20 on -(z<0)-(z=0)*2-(z>0)*3 goto 30,40,50
30 print"zahl kleiner 0":goto10
40 print"zahl gleich 0":goto10
50 print"zahl größer 0":goto10
```

Erklärungen:

'OPEN4,4' lenkt die Ausgabe auf den Druckerkanal, 'OPEN3,3' die Eingabe auf den Bildschirm. Jetzt können die Zeichen vom Bildschirm mit GET#3,A\$ gelesen und mit PRINT#,A\$ gedruckt werden. Um unnötige Druckerbewegungen zu ersparen, wird immer zeilenweise gelesen und diese Zeile in 63230 von den Leerstellen am Ende befreit, bevor sie ausgedruckt wird.

BÖRSE

Verk. Software f. Plus4 um die Hälfte, ca. 35 Prog. v. M+T, Kingsoft, Mastertronic u.a. Verk. auch div. Fachlit. v. Data Becker, Liste gg. Porto bei Karl-Heinz Schön, Bergstr. 14, 7015 Korntal

Für Commodore Plus4 Orig.-Software zu verkaufen. Quiwi, Winter Olympiade, Hollywood Poker, Mercenary, Compend auf Disk billig abzugeben. Liste gg. Porto bei Karl-Heinz Schön, Bergstr. 14, 7015 Korntal

Commodore 1280, Farbmon. 1901, Floppy 1571, Textprog. Wordstar, Dateprog. dBase, Grafikprog. u. Spielprogramme. Nur 2.330,- DM. Werner Kiefer, 7701 Aach, Tel. 07774/7299

ACHTUNG, C16/116/+4-USER! Verk. über 20 Spiele wie Winter Events, Musikmaster, Kikstart. Infos f. 80 Pf. bei: Hilmar Demant, Luisenstr. 82, 7528 Karlsdorf-Neuthard 2

Suche f. C16: gebrauchten Farbmon. u. Drucker, Bücher u. Programme (Kass.): Werner Haas, Postf. 1632, 7600 Offenburg, Tel. 07805/3010 nach 17 Uhr

Super-Software f. C16/116/Plus4, C64/128. Kostenlose Infos bei: M. Greifenhagen, Stöcklerstr. 8, CH-8610 Uster, Schweiz

**** C128 **** Suche Kontakt zu anderen C128-Usern im Raum HSK. Suche Prog. f. C64, C128 od. CP/M-Modus. Liste an Herbert Stein, Hauptstr. 26, 5779 Eslohe, Tel. 02973/1779. Habe auch selber Prog.

Wer hat u. sucht C16/116/+4-Programme? Wer hat Interesse an Kontakten, Info u. Tausch m. vielen anderen Freaks? Schickt mir Eure Liste, ich gebe Euch meine! (Disk./Tape) Ingo Sabel, Simmernstr. 38, 5411 Neuhäusel, Gut-Byte!

40 Spiel- u. Anwend.-Programme, Liste m. Startadressen, Liste m. Pokes u. 5 Seiten Anleitung f. 20,- DM auf Disk. Vorkasse. Thorsten Idzellis, Brambusch 23, 2357 Bad Bramstadt

Suche Druckprogramme wie Hi-Eddi, Printfox od. Newsroom f. d. Plus4. Tel. 0203/438266

Suche f. C16: Kopiermodul od. Prog. Tape/Disk., auch 64K-Erw.-Modul. Bitte schreiben Sie an: Kristian Nielsen, Emdrupvej 113, St. Lej 2, 2400 Kopenhagen, Dänemark

ACHTUNG! C16/116/+4!!! Suche Tauschpartner/in f. Spiele u. Anwender, 16-64K, auf Disk. od. Tape. Es sollten sich vor allem Anfänger melden. Tel. 02402/20353, Udo Gath, Buschmühle 14, 5190 Stolberg

C64 - C16 - Plus4 - Lernprg. Techn. Mathe + Schulanwend. + Grafik zu reellem Preis. Cass/Disk z.B. Bruchrechn. Vokab. Geometrie, Zahn. Festigk. Hydr. E-Techn. Katalog 1 DM Briefm.-Comp. Typ angeben! Softvers. A. Ristau Peetzweg 9, 3320 Salzgitter 1

++++ SCANNER & EPROMMER ++++
Verkaufe: Super-Scanner II m. Extension, Merlin PP-64, TOM, PF 1162, 4516 Bissendorf

C16/116+4! Verkaufe Orig.-Spiele: Ace, Battle Star, Big Mac, BMX, Raider, Formula 1, Las Vegas, Shark, Spectiped, Mega zap, Mail Trail, Tutti Frutti, Xargon Wars, Climb, Olympiade u. Games Pack 2, billig. Tel. 02402/20353, Udo

Suche komf. Dia-Verwaltungs- u. Sortierprogramm f. C16 (64K). Erich Rother, Nieberdingstr. 19, 4300 Essen 1

* **ACHTUNG, WICHTIG, EILT** * Suche dringend f. C128: Farb- od. Grünmon. sowie einen NLQ- u. grafikfähigen Drucker. Suche f. C64: Das Modul Freeze Frame MKII. Also: gleich zum Telefon spurten u. anrufen, Tel. 04521/72397

DRUCKER U. LITERATUR: 64er u. Happy-Computer-Hefte ab 1985, 10 St. f. 20,- DM; alle Sonderhefte je 5,- DM; Drucker-Interface BKB-Puffer 70,- DM; Drucker f. C64 preiswert. Tel. 06181/65400

ACHTUNG - ACHTUNG - Wer verkauft mir möglichst billig einen Commodore MPS-1000. Angebote möglichst schnell an: Chris Mühlhausen, Hubenröderstr. 7, 3430 Witzenhausen

VERKAUFE C1280, 1 J. alt, Mouse, Grafikprog., Joycard, Speeder, Freezer, 60 Disk. m. Box, Top-As u. Plus-Assembler, Programmierhandbuch, Floppyhandbuch, CP/M-Disk.-Handbuch, Disk-Locher. Bitte ab 18 Uhr anrufen: Tel. 04131/81993

Suche Handbuch-Bedienungsanl. zum Commodore Drucker MPS 803. Angebote an: Thorsten Boese, Pappelweg 23, 2300 Kiel 1

ACHTUNG! Suche Floppy f. Commodore Plus4. Möglichst billig; bis 150,- DM. Angebote an: Marco Klesper, Kreuzweg 26, 5800 Hagen 1

PLUS4/C16 - VERKAUFE: Winter Olympiade u. Quiwi. Tape je 15,- DM. Suche Drucker f. +4. Harald Pries, Goethestr. 9, 3523 Grebenstein

Suche alles Pascalmaterial f. C64/128. Billig od. am besten kostenlos. Listen od. Angebote an Dominic Birk, Füssenerstr. 11, 8968 Durach. Wer verschenkt Drucker?

Verk. Anwenderprogramm: Hauptnutzungszeit beim Drehen, Fräsen. Für C16 u. 64 KB. Mit Betriebshandbuch. Zum Berechnen bearbeitungsspezifischer Werte (Z.B. Kegeldrehen). Mit grafischer Unterstützung. A. Tape 20,- DM. Selbst programmiert. Lars Lenort, Ramlingerstr. 77, 3167 Burgdorf

C16/PLUS4: Ca. 300 Prog., Public-Domain-Software, zum Tausch anzubieten. Nur Disk. Schickt Eure Angebotslisten u. Wunschlisten an: Uwe Braun, Fasanenring 9, 2822 Schwanewede II. Keine Raubkopien!

VERKAUFE Kass.-Kopiermodul f. C16, 64, 128, m. Netzteil, ca. 20 Spielen f. C16, dt. Anl., Adaptorn f. C16, m. 2. Datasette. Darf nur f. Sicherheitskopien benutzt werden! Kopiert alle Prog.-Kassetten. Tel. 05163/6628

Achtung Commodore 64-Freaks! Suche C64 I m. Floppy od. Datasette. Bezahle gut. 300,- bis 450,- DM. Schreibt mir doch. Ch. Ostermann, Sulenbecker Weg 3a, 5882 Meinerzhagen 1. Es eilt! Kaufe beide Teile auch einzeln

Suche Script Plus m. Anl. Angebote an: Brigitte Nitsch, Steinbruchstr. 17, 4100 Duisburg 1, Tel. 0203/358074. Auch gewerbliche erwünscht!

Drucker SP180VC f. C64 u. 128 zu verkaufen f. DM 400,- (neuwertig). C64, Floppy 1541, Betriebssystem SPEED-DOS PLUS, Textverarb., Giga-CAD Plus u. Software. Tel. 05724/1873

Suche: VC-20 Speicher-Erw. 32K bis 30,- DM. Verkauf DRucker Epson LQ800 (24 Nadeln), m. Traktor f. Endlosblatteinzug u. 2 neuen Farbbändern, 1 Jahr alt, Centr.-Kabel f. Atari od. IBM-Computer, wg. Umstellung auf DIN A3-Format, DM 950,- F. Schuster, Karlsbader Str. 12, 8783 Hammelburg

C16 Turbo Master, alle I/O Anweisungen: SAVE, LOAD, OPEN, CLOSE, CMD, PRINTL, INPUTL, GETL, 10mal schneller. Preis 30,- DM. Info, Best.: Reinhold Krammer, Königsberger Str. 8, 6257 Künfelden 5

* **HELP * HELP * HELP *** Suche dringendst HF-Modul f. VC-20, außerdem preiswerte Lernbücher u. funktionstücht. C16-Platine. Ich warte auf Eure Briefe bis Weihnachten 88. M. Sichel-schmidt, Scheffelstr. 17, 4000 Düsseldorf 30. Danke!

SUCHE DRUCKER-HANDBUCH f. Commodore MPS802 4023 zu kaufen od. zu kopieren. A. Stübs, Tel. 02351/6956, 5880 Lüdenscheid, Am Weiten Blick 35, Tel. ab 18 Uhr

*** **C16/116/PLUS4** *** Hast Du Turbo Plus u. suchst gute u. billige Tape-Spiele? Fordere eine Liste an, verk. auch auf Disk. Tel. 0203/358074, Brigitte Nitsch, Steinbruchstr. 17, 4100 Duisburg

HILFE! Wie kann ich die meinem +4 erstellten Grafiken u. Zeichnungen über den MPS 1000 ausdrucken? Suche Scanner u. Lernprogramme a auf Tape. Alex Till, Hermannstr. 37, 7065 Winterbach

SUCHE: C128D, C64, C16, P4, VC20, Commodore-Hardware jeder Art, auch defekte Geräte! Zahle gut! Angebote mit Preisvorst. schriftl. an: Sascha Hren, c/o Liebig, Heckstr. 11, 4800 Bielefeld

C128-User-Clubs aufgepaßt! Ich bastele Eure Clubzeitungen zu voll menügesteuerten Disk.-Magazinen um. Info gg. 80 Pf. Rückporto bei Uwe Ludschowitz, 6090 Rüsselsheim, Hassloch am Wasserkwerk 1

Verkaufe C16, 30 Spiele, 3 Bücher, Basic-Kurs-Kass., Datasette, Fernsehverbindungskabel u. Joystick f. 150,- DM. Tel. 09181/8795

Achtung, C167116/Plus4-User! Endlich: 2. Betriebssystem auf Platine, abstrichfrei schaltbar; versch. Modulsoftware (z.B. Script u. Logo u.a.) a. Umschaltplat. Info gg. Freiumschlag. Netzteile C16/116 1200 mA DM 35,- inkl. Porto u. Verp., nur Vorkasse. Jürgen Braunroth, Moordorfer Str. 30, 3057 Neustadt 1

TYPE TEACHER, Lernprog. f. Schreibmaschinenschreiben auf IBM-Kompatiblen, CGA/Hercules-Karte, 48,- DM. Scharf, Waldspielplatz 12, 8130 Starnberg, Tel. 08151/16410

C16 Suche Programmbe-schr. f. Sports 4 v. Ancosoft, Kan 4, Fußball, Thai B., K. Gotzler, Humperdinckstr. 2, 8000 München 40

Plus4: Suche Tauschpartner (Spiele, Anwender auf Disk.). Wer hat FASTloader/Kopierer f. die 1551? Liste/Disks an Wolfgang Saafar, Tel. ab 15.30 h 0043/222/9592892

Verk. Spiele u. Leerdisketten. Liste gg. 1,30 DM in Briefm. Schreibt an Mario Zimmermann, Schwäblstr. 25, 8070 Ingolstadt! Es lohnt sich!

Kostenlose Freeware bei Fr. Neuper, 8473 Pfreimd, Postfach 72. Gratisliste anfordern.

Verkaufe C16, Datasette, Basic-Kurs (Buch u. Kass.) inkl. ca. 50 Prog. (kaum benutzt). Angebote (mind. 200,- DM) an: Kai Uwe Boß, Ulmenweg 5, 6422 Herbstein, od. Tel. 06643/8286 nach 19 Uhr. Weiterer Prog.-Austausch möglich

Verkaufe guterhaltene Computerzeitschriften z.B. Commodore Welt, Compute mit, ct, Computronic u.v.a.! Ferner Sonderhefte! Till Franzmann, Tel. 06132/8168, Freitag ab 19 Uhr

Suche dt. Anleitung f. Commodore MPS-803 DOT MATRIX PRINTER. Heinz Bredow, Am Rüberg 8, 5940 Lennestadt 1, Tel. 02723/3782

Verkaufe Spiele f. C64 superbillig (1 Spiel f. nur 1,- DM). Liste gg. 2,- DM in Briefm. Schreibt an: Helmut Friedel, Sonnenstr. 17, 8531 Tel. 09161/4497

Suche Tauschpartner f. Plus4 u. C64. WER hat Rubotape f. Plus4? Meldet Euch bei Matthias Schneider, Michaelstr. 3, 2878 Wildeshausen

Mehr als 12 C64-Anwenderprog.: KFZ-Bordbuch, Kurzzeitmessung, Farbwindow, Tagesdifferenz, Säulendiagramm u.a. in Basic u. Masch.-Sprache. Info gg. Freiumschlag v. Thilo v. Stillfried, Meisenweg 13d, 5983 Balve 1

*** SUCHE *** Prog., mit dem ich C64-Prog. auf meinem Plus4 spielen kann. BIETE Softw. z.B. Trivial Pursuit. Schreibt an Sven Kuessner, Brede 6, 5758 Fröndenberg-Ardey. PS: Verkaufe auch Spiele wie Summer Games!!!

SOFTWARE + Für C16 u. Plus4. Selbstentwickelte od. Secondhand-Software. Info gratis! Tips u. Tricks (Pokes u. SYS) nur 1,- DM! Postk. an P. Schäfers, Riekestr. 5, 4402 Grevin 1

BIETE Star VIC-Interface m. Gehäuse f. Star-Drucker (z.B. Gemini-10X) am C64/128 od. VC 20. Anschluß an seriellen Port. Preis vereinb. Thomas Weyrauch, Rheinstr. 20, 5431 Holler, Tel. 02602/5605

SUCHE: gebrauchte Datasette f. C16/Plus4. Schlawinski, Carl-Petersen-Str. 60, 2000 Hamburg 26. Bitte anbieten!

SUCHE Star NG 10 Drucker m. Commodore-Interface, nur dt. Version (FTZ-Nr.!), m. dt. Handbuch, möglichst m. Einzelblatteinzug zu real. Preis. (Neueste NL-Preise beachten!) Tel. 02752/5905

HALLO, C64-Freunde! Gebe Prog. auf Disk. ab. Randvolle Disk. m. Spiel- u. Anwend.-Prog. 20,- DM. Info gg. 1,20 DM Rückporto. System an-geben! Klaus Spengler, Burgunder Str. 9, 4600 Dortmund 30

CBM-600-700er Serie. Wer hat f. diesen REchner Anwendersoftw. zu Hobbypreisen zu verkaufen? Weiter suche ich 64/128er Anwendersoftw. zu kaufen od. zu Tauschen. Tel. 06201/33860, Klaus Günter Derksen, Am Tannenbuckel 18, 6943 Birkenau

+ Plus4/C16/116 + Suche Tauschpartner zwecks Prog.-u. Erfahrungsaustausch. Nur Disk. (Tape im Notfall auch möglich). Habe an die 500 Prog. Schickt Eure Listen an: Arne Woll, Am Stübenhaus 15, 2900 Oldenburg

Verk. Commodore C16 m. 64 KB, Netzteil m. 1,2 A, Tape 1531 u.ca. 400 Prog. f. nur 250,- DM. Kopiermodul f. 1531 nur 40,- DM. Verk. MPS803, 1/2 Jahr alt, f. 350,- DM. Div. Orig.-Prog., Liste gg. Porto bei Karl-Heinz Schön, Bergstr. 14, 7015 Korntal

Verkaufe C-16, 10 Orig.-Spiele u. jede Menge Programme in BASIC auf Tape!!! Die Taste 5 ist defekt. Mit Abdeckhaube! Und Handbuch. Das alles f. 200,- DM u. Porto. Andreas Fehres, Schwalbenweg 15, 6710 Frankenthal. Zupacken, bevor es jemand wegschnappt!

Hallo, C16/Plus4-Freunde! Gebe Prog. auf Disk. ab. Fast 100 Spiele f. 15,- DM. Info gg. 1,20 DM Rückporto. Auch Anwend.-Prog. Klaus Spengler, Burgunder Str. 9, 4600 Dortmund

Suche Anschluß an Commodore Plus4-Besitzer im Raum Aachen u. Umgebung. Alfred Hansen, Dedolphstr. 4, 5100 Aachen

Suche Floppy 1541, SPEED-DOS u. Drucker f. C64 (anschlußbereit). Biete Plus4, Floppy 1551, Drucker VIC 1515 u. ca. 40 randvolle Disk. f. P4. Tausche auch. Tel. 02752/7899 morgens od. nach 21 Uhr

Verkaufe orig. Defender of the Crown, Mania Mansion, Kill until Dead, o. Erw. auf C16/+4, je 20,- DM. Tel. 0231/812114

C16 Turbo Master, alle I/O Anweisungen: SAVE, LOAD, OPEN, CLOSE, CMD, PRINT#, INPUT#, GET#, 10mal schneller. Preis 30,- DM. Info, Best.: Reinhold Krammer, Königsberger Str. 8, 6257 Künfelden 5

Suche Floppy 1541 od. 1551 f. C16. Bitte nur intakte Geräte. Angebote an: A. Hallen, Fürstenwall 24, 4000 Düsseldorf

80-Zeichen-Text-Prog. f. den Plus4 anzubieten m. Anl. Monitor 80Z. erforderlich. Alfred Hansen, Dedolphstr. 4, 5100 Aachen

C64 C128 VC20 User Achtung Lichtgriffel mit Programmen und dt. Anleitung nur DM 49, Versand gegen Scheck/Nachn. Fordern Sie unseren kostenlosen Commodorezubehörsprospekt an. Firma Klaus Schießlbauer, Postfach 1171C, 8458 Sulzbach, Tel.: 09661/6592 bis 21 Uhr. Sonderaktion: Legen Sie Ihrer Prospektanforderung DM 10,- bei. Sie erhalten 10 Superprogramme auf Kass. od. Disk für C64/C128.

Commodore C128 u. 1571 m. Joystick, Disketten u. Anwenderhandbuch. Alle Teile eu. VB DM 950,-. Dirk Geule, Tel. 05674/4321

ACHTUNG! Jrtzt geht's rund! Verkaufe u. tausche Programme f. C16/Plus4 auf Disk. u. Tape. Helmut Plank, Öhlermühle 21, A-3362 Öhling, Österreich

Suche Spiel- u. Anwenderprogramme f. Plus4. Auf Disk., evtl. auch auf Kass. Bin auch am Tausch interessiert! Paul Kasparbauer, Burgweg 1, 8376 Altnußberg

Suche f. C64 Games (Disk.). Verkaufe außerdem noch 28 C64/128-Games (Tape) u. eine G-7000 Kass. Supermampfer um öS 500,-. Schickt Listen an: Hans Hammerschmied, Hauptstr. 14, A-2111 Kleinrotz

Suche 64er Hefte Nr. 4,5,6,7, 12/86 sowie Tips u. Tricks f. Superbase (Plus4). Arno Jaksch, Im Moore 15, 3000 Hannover 1

C16/Plus4;Software, meist Spiele, im Orig., spottbillig. Liste gg. Freiumschlag m. 0,80 DM. Harald Scheel, Kollaustr. 178, 2000 Hamburg 61

C16/+4 -- VERKAUFE: Space Pilot, Bongo, Ace +4, je 10,- DM (Tape). SUCHE: Drucker f. +4. Harald Pries, Goethestr. 9, 3523 Grebenstein

BÖRSE

Wer verschenkt an armen Schüler einen Commodore C64. Wir brauchen ihn zwecks Schule. Wir zahlen höchstens 50,- DM. M. Greifenhagen, Stöcklerstr. 8, CH-8610 Uster, Schweiz

C16/64 KB m. Datasette 1531 u. 50 Spielen, Ace, Ace 2, Summer Event u.a. 200,- DM; C16 m. 1531 f. 150,- DM. Jörg Hildebrandt, Schlenkenbrink 25, 4973 Vlotho-Uffeln

Freesoftw. f. C16/116/Plus4: 3600 Block = 6 Diskseiten nur 20,- DM. Über 100 Programme. Viktor Jürgens, Prinzenstr. 131, 2330 Eckernförde

Ich suche START u. SYS-Adressen f. den Plus4. Suche DINA4 Hardcopy f. Plus4 u. MPS 100. Tel. 04192/4726, Thorsten Idzellis, Brambusch 23, 2357 Bad Bramstadt

Verkaufe 35 Plus4-Spiele od. Anwender auf Disk. f. 20,- DM. Lieferung nur gg. Vorkasse. Frank. Rückumschlag m. 20,- DM an: Stefan Hoffmann, Gasstr. 17, 2308 Preetz. PS: Tausche Programme

C16/116/PLUS4! Wer hat gute Programme (abgetippt, gekauft usw.) u. möchte mit mir tauschen? Jeder Brief wird beantwortet!! Schreibt an: Christian Bartsch, Norkshäuschen 136, 5600 Wuppertal 1

Achtung! Verkaufe Commodore Monitor 1702, techn. wie neu, m. Verp. Anschluß an C64, Preis 500,- DM VB. Tel. 06024/7939. Möglichst im Raum Aschaffenburg!

HILFE HILFE HILFE!
Wer kann mir helfen? Suche jemand, der mir bei der Basic-Sprache f. den C64 weiterhilft. Bitte melden! Norbert Korthols, Bahnhofstr. 185 A, 4100 Duisburg 12, Tel. 0203/439513

Suche f. meinen Neffen (12 J.) Floppy u. Programme f. Plus4. Gerhard Andermann, An der Hofbreite 2, 3412 Nörten-Hardenberg, OT Elvise, Tel. 05503/2590 nach 19 Uhr

Suche C64 od. C128 m. Floppy, Drucker, evtl. auch Monitor. Nehme auch Einzelteile, wenn günstig. Tel. 069/302940.

Verkaufe C16/64K m. Floppy 1551 u. Büchern gg. Höchstgebot. Tel. 02272/7942, Heinz Lurz, Schützen-delle 63, 5012 Bedburg

PLUS4/TEXTADVENTURES: Erfahrungsaustausch u. evtl. gemeinsames Entwickeln v. Programmen wünscht Wolfgang Dethlefs (39 J.), Krokusweg 7, 2382 Kropp, Tel. 04624/1760

Verkaufe Spiele f. C16, Orig. auf Tape zu niedr. Preisen. Ruft an 06233/62937. Andreas Fehres. Jede nur einmal. Grüße M. Deis. Schreibe mal wieder!

60 Spiele f. C64 f. 30,- DM sowie C16/116/Plus4 ebenfalls 30,- DM. Schick die Scheine an: Alexander Käss, Himmelstr. 7, 7987 Weingarten und Ihr bekommt die Software.

C16(64K), Datasette, Floppy 1551, M&T-Sonderhefte, Prog.-Service-Disketten, Spiele auf Disk. u. Kass. (keine abgetippten!), Textverarb. u. Dateiverw. Alles 1A-Zustand, (auch einzeln) billig abzugeben. Tel. 02821/20050 ab 18 Uhr.

Suche Textmanager m. Such- u. Sortier-Routine, Plus4, Kass. Jürgen Schust, Försterweg 40, 2000 Hamburg 54, 040/5407326

C16, 116, Plus/4-Anwender: Der Verein speziell f. uns! Mit Clubzeitschr. u. Software-Bibliothek. Info beim hrc e.V., Bauerland 15, 48 Bielefeld 1

Hallo, Plus/4-C16-Fans! Plus4-User aus der DDR sucht Erfahrungsaustausch u. Briefwechsel. Bin 31 J. (M). würde mich auch sehr über Soft- u. Hardware freuen, da mittellos. Jeder Brief wird beantwort. v. Frank Harnisch, Ringstr. 189, DDR-7060 Leipzig

Verkaufe versch. Zeitschr. (64er, RUN, CINEMA ...) sowie div. Orig.-Prog. (Tape u. Disk.) f. C64. Liste gg. Rückporto. Kaufmann, Burgstr. 61, 7000 Stuttgart 80

Suche intakte Floppy 1551 u. GRAFIKMODUL f. den MPS-802/1526 od. Plotter!! Angebote an: Brigitte Nitsch, Steinbruch Str. 17, 4100 Duisburg 1, T. 0203/358074. PS.: Tausche auch Top-Soft auf Disk.! Beantworte alles!

Für 10,- DM bekommt Ihr die besten Spiele für den C16/+4: 2 Disk. randvoll m. Topsoftware! Suche noch Partner zum Software-Tausch. Nur Disk. Suche noch Superbase (64K). Bitte wenden an: A. Balgar, Fortkamp 7, 4250 Bottrop

Für Plus4, C16(64K): Mathimago 30,- DM; Portfolios 15,- DM. Info bei: Peter Hakenjos, Werderstraße 13, 7500 Karlsruhe

Suche Grundplatte f. C16 (nicht defekt). Angebote an: Lothar Härterich, Hauptstr. 85, 8721 Poppenhausen, Tel. 09725/1434 ab 18 Uhr

Hey, Computer-Freaks! Ich suche div. Data Becker Bücher f. C64/C16/P4. Zahle bis zum halben Neupreis! Schickt Eure Angebote an: Jens Schröder, Osterholzer Möhlendamm 55, 28 Bremen 44, Tel. 0421/421238

Verkaufe meinen C16 + 64 KB. Mit Datasette, Basic-Kurs usw. Ertl. Plus4 m. 256 KB. Preis VB. Verk. auch einige Orig.-Software. Axel Kleinsch, Dümmlinghauserstr. 22, 5270 Gummersbach 1, Tel. 02261/52644 v. 16-17 Uhr

Verkaufe DFU-Prog. u. Taschenrechnerfunktionen. Disk.: 10,- DM. ACHTUNG: 1. Prog. arbeitet nicht mit allen Dataphonen. 2. Keine Texteingabefunktion. Prog. f. C64. Kein Umtausch. Matthias Gutt, Kantstr. 16, 2120 Lüneburg

Suche Footballer of the Year f. Plus4. Der erste erhält 2 Orig.-Spiele (Tape). Markus Ramm, Frankenweg 33, 4630 Bochum 6

Achtung! Suche Drucker f. C64. Da total verarmt, kann ich nur bis zu 150,- DM zahlen! Tel. 02392/71589, Raum Werdohl. Dringend !!!

C16/+4 Gratis-Info anfordern. Kopiermodul, Kopier-Prog., Anwender- u. Spiel-Prog. Jetzt Gratis-Info anfordern!! Th. Görtz, Friedrich-Ebert-Str. 113, 6103 Griesheim, Tel. 06151/719132

ACHTUNG! C16/116/Plus4. Mit POKE4097,1:DELETE1 macht man Reset und new rückgängig. Suche C64 bis 150,- DM, kein Schrott. Tel. 04861/6264. Ich rufe zurück.

IDEAL FÜR EINSTEIGER!
VC20 m.: Datasette, 3K-Erw. 3 Spielemodulen u. 17 Kass. (4 neue, 5 eigene, 7 Orig.-Spiele, Basic-Kurs). M. Porto nur 100,- DM. Köneke, Westendstr. 26, 6072 Dreieich

C128D, neuestes Geräte (Stahlgehäuse), 80-Z.-S/W-Mon., 70 Disk. u. div. Zubehör, alles 6 Mon. alt. VB 900,- DM. Tel. 02234/58964 ab 18 Uhr

Input-Kass. 3/85-8/87 f. 150,- DM; Profimat m. Disk. f. 50,- DM; Roboter-Buch 20,- DM; Lexikon C64 25,- DM; Codier-Blätter Conradbasic 200 St. 6,- DM. Wolfgang Bruns, Graf-Bernhard-Str. 22, 4934 Horn-Mythenberg 01. Tel. 05234/1484 ab 18 Uhr

HEY COMPUTERFREAKS!
Verk. C16 + 64K 150,- DM; Datasette 75,- DM; Floppy 400,- DM; 10 Leerdisk. 40,-; 2 Quickshot II 25,- DM; 2 Adapter 10,- DM; Sonderhefte, VIA, Software (z.B. Winter Olympiade u.v.m.). CDS, 44, rue de Mondercange, 4381 Ehlerange-Luxemburg

Tausche, kaufe, verkaufe Spiele, Anwend.-Prog. u. Musik-Prog. f. C16/+4. Suche 2 Joysticks z. günst. Preis! Bitte um Angebote, Listen u. Ähnliches. Jede Zuschrift wird garantiert beantwortet! Alles an: Jürgen Oschinger, Eschweg 1, 2171 Geversdorf/Oste

*** HALLO C16/P4 FREAKS***
Verkaufe 20 Prog. f. nur 10,- DM. Auf Disk.! Verk. 50 Prog. f. nur 20,- DM. Auf Disk.! Bei Interesse Schein an K. Henk, Brunsauerstr. 4, 518 Eschweiler

An alle C16/Plus4-Fans! Bieten ca. 400 Games ab 5 Pf. pro Spiel, Liste & Lieferbedingungen bei Jordi Rieger, Maiburgerstr. 38, 2950 Leer/Logabirum anfordern od. anrufen 0491/73808, Spider-Crack Computer-Club

Suche: „Mein 1. C64-Programm“ v. G. Lippman, Sybex-Verlag. Riedl, Tel. 0911/440166

Verkaufe div. Input 64, Software-Club-Kassetten u. Commodore Discs (teils orig.-verpackt). Angebot an Till Franzmann, Tel. 06132/8168 (Freitag ab 19 Uhr)

ARBEITEN MIT EINEM FORTRAN COMPILER

Keine Angst vor harten Nüssen

Wenn man eine Problemlösung programmieren will, bedient man sich meist einer höheren Programmiersprache. Allerdings versteht ein Computer auch diese Sprache nicht so, wie man sich dies vielleicht wünscht. Das Programm muß zur Bearbeitung übersetzt werden in eine dem Computer verständliche Form: den Objektcode. Im Gegensatz zu dem in einer Programmiersprache geschriebenen Programm, dem sogenannten Quellprogramm, besteht das Objektprogramm nur aus Ziffern, die in verschlüsselter Form die Informationen des ursprünglichen Programms enthalten. Diese Art von Programm kann vom Computer bearbeitet und verarbeitet werden. Man nennt diese Sorte von Code Maschinensprache.

Es gibt zwei Möglichkeiten, zu dem Objektprogramm zu kommen:

1. Der Interpreter
 2. Der Compiler.
- Der Programmierer, der an einer Anlage arbeitet, die einen Interpreter besitzt, muß kaum Kenntnisse über diesen verfügen. Er erstellt sein Pro-

gramm (Quellprogramm) in einer Programmiersprache und gibt dem Computer anschließend den Befehl, das Programm zu starten. Nun tritt, vom Benutzer unbemerkt, der Interpreter in Aktion. Jeder Befehl, jede Zeile des Programms wird vom Interpreter interpretiert, das heißt, er gibt nicht die Befehle, sondern die

FORTRAN COMPILER LINKAGE EDITOR

den Befehlen entsprechenden Codes weiter. Sie werden dann intern verarbeitet und dem Benutzer wird gemeldet, ob sein Programm einwandfrei bearbeitet werden konnte oder nicht.

Anlagen, die mit Interpretern arbeiten, sind meist auf eine einzige höhere Programmiersprache festgelegt. Verwendet ein Computer jedoch Compiler, kann fast jede beliebige Programmiersprache verarbeitet werden, wenn sie vom Anlagenhersteller angeboten wird.

Compiler sind Übersetzerprogramme, die eine bestimmte höhere Programmiersprache in den oben erwähnten Objekt-

code umwandeln. Dies geschieht nur einmal. Zur Ausführung kommt dann immer nur das übersetzte Programm (Objektprogramm). Der Interpreter muß bei jedem Programmdurchlauf neu in Aktion treten, was sich negativ auf die Ausführungszeit eines Programmes auswirkt.

Mit dem Compiler bearbeitete Programme sind also schneller als interpretierte. Außerdem können Unbefugte in einem kompilierten Programm nicht manipulieren, da es ja nur aus Ziffernkombinationen besteht (Programmschutz).

Die Handhabung eines Compilers ist etwas umständlicher, daher soll hier näher auf sie eingegangen werden.

Als Beispiel dient die Arbeit mit dem Fortran-Compiler für die „2cs“ von Triumph-Adler.

a) Zunächst werden mit Hilfe eines Dienstprogrammes (FUR) vier Dateien angelegt mit den Namen: Quelle, Objekt, Binden, Progrm. In die Datei mit dem (frei gewählten) Namen Quelle wird vom Programmierer das Fortran-Programm geschrieben. Dies geschieht mit Hilfe eines Texteditors. Hier werden noch keinerlei Fehler erkannt, das heißt, der Programmierer hat an dieser Stelle noch keine Kontrolle über die Ausführbarkeit seines Programmes. Die Datei Quelle enthält also das noch unbearbeitete Quellprogramm.

b) Nun wird der Fortran Compiler aufgerufen mit dem Befehl FC. Es muß im einzelnen definiert werden:

Input File:

Hier muß angegeben werden, in welcher Datei das Quellprogramm steht. („Quelle“).

Output File:

In die hier angegebene Datei wird vom Compiler das Objektprogramm

ausgegeben. („Objekt“). List File:

Ein Gerät muß angegeben werden, auf welchem das Übersetzungsprotokoll geschrieben werden soll (Drucker, Bildschirm oder Datei).

Error File:

Auf dem hier eingesetzten Gerät werden die Fehlermeldungen ausgegeben.

Options:

Hier können verschiedene Optionen gewählt oder unterdrückt werden, etwa:

- freies Format;
- Ausdruck einer Querweisliste der verwendeten Variablen,
- Übersetzung mit Ablauf-Protokoll,
- Protokoll des erzeugten Objektcodes,
- bedingte Übersetzung.

Sind alle Angaben vollständig, kann der Compiler gestartet werden. Während der Übersetzung des Quellprogrammes erkennt und meldet der Compiler Anweisungsfehler, Programmfehler und Laufzeitfehler.

Anweisungsfehler verstoßen gegen die syntaktischen oder semantischen Regeln der Sprache (hier Fortran) zum Beispiel: unzulässige Namen, falsche Variablentypen, falsche Dimensionierung, doppelte oder fehlende Anweisungsnummern, falsche Anweisungen.

Programmfehler beziehen sich auf das Programm als Einheit, wie Speicherbereichsfehler, Größe des Programms, Fehler im Betriebssystem, Fehler bei aufgerufenen Peripheriegeräten, unzulässige Compiler-Optionen, falsch aufgebaute D-Schleifen.

Laufzeitfehler können bei der Handhabung der peripheren Geräte auftreten: Ein- und Ausgabefehler, TCA-Fehler (Task Communication Area).

COMMODORE WELT/SPECIAL

COMMODORE C64/C128 SOFTWARE JAHRBUCH 1988

DAS BESTE AUS CW
Sammelband Nr. 2/88
DM 14,80-ÖS 124-SFR 14,80

```

100 PRINT"
110 PRINT"THE BEST OF COMMODORE-WELT"
120 PRINT"-----"
130 LOAD "LISTING 1" : RUN
140 LOAD "LISTING 2" : RUN
150 LOAD "LISTING 3" : RUN
160 LOAD "LISTING 4" : RUN
170 REM "
180 REM "IF YOU WILL LOAD AND RUN, "
190 REM "YOU WILL HAVE A LOT OF FUN"
200 REM "
    
```

Die besten Anwender-Programme

Die schönsten Spiele

Hilfreiche Utilities

Das Listing-Heft für Ihren Commodore 128/C64

**Zwei
starke
Helfer
für jeden
128er-
User!**

**Test-
Jahrbuch 88
Software-
Jahrbuch 88**

**Alles über
und für den 128er**

**An ausgewählten
Kiosken und im
Bahnhofs-Buchhandel**

COMMODORE WELT/SPECIAL

COMMODORE C64/C128 TEST-JAHRBUCH 1988

DAS BESTE AUS CW
Sammelband Nr. 4/88
DM 19,80 - ÖS 158 - SFR 19,80

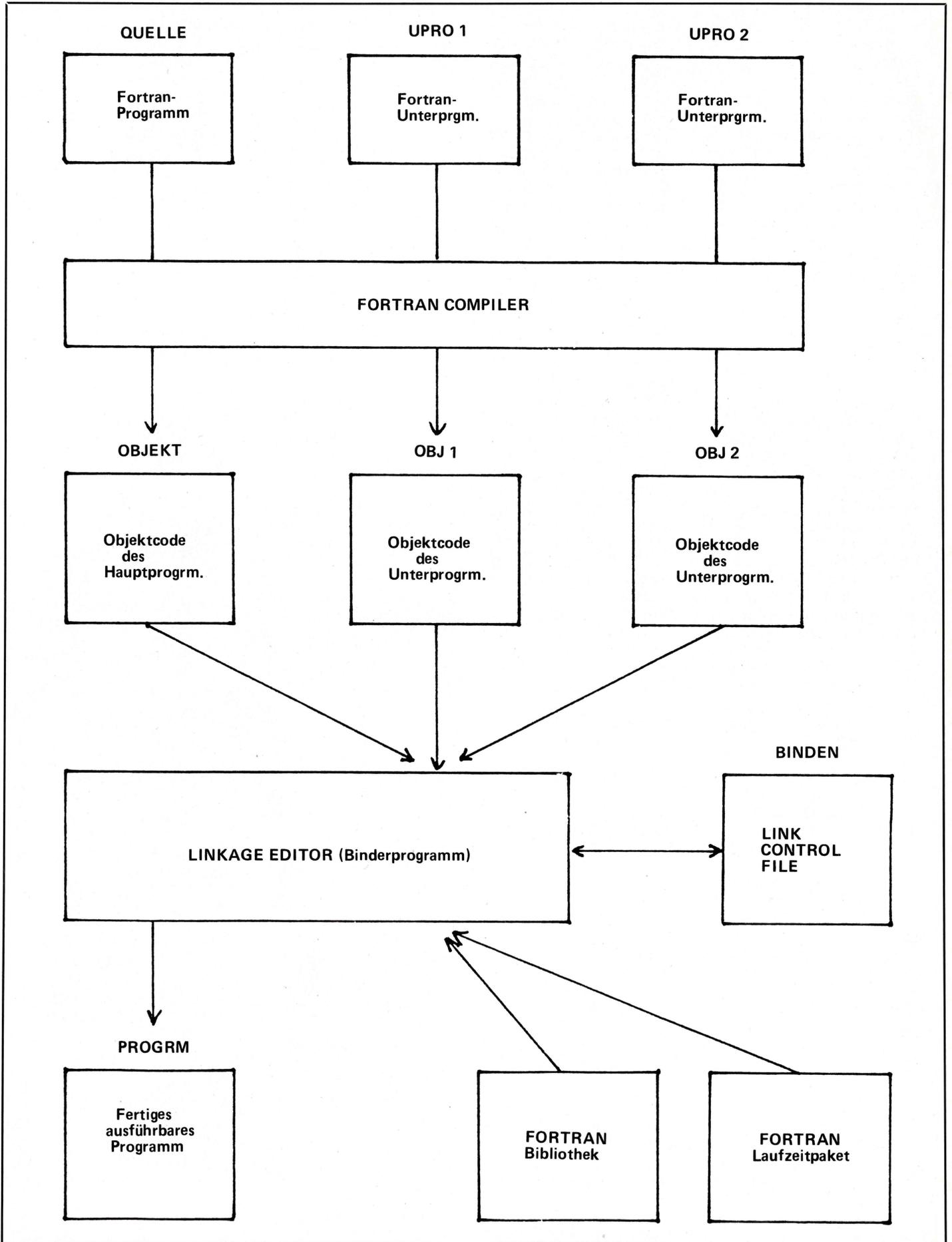
Alles über Ihren C16-128PC

Tips Tests Tricks

Rund 150 Seiten Einkaufsführer

Hard- und Software im Test! - Kaufberatung

SERVICE



Hat der Compiler seinen Durchlauf beendet, so werden Anzahl und Art der Fehler angegeben. Die Aufgabe des Programmierers besteht nun darin, die angezeigten und lokalisierten Fehler zu beseitigen und den Fortran-Compiler erneut aufzurufen. Nur ein fehlerfreier Compiler-Lauf erzeugt auch ein fehlerfreies Objektprogramm, welches in die angegebene Datei geschrieben wird.

c) Um dieses Programm aber ausführen zu können, müssen erst alle im Fortran-Programm aufgerufenen Bibliotheksprogramme und benötigten Laufzeitmodule hinzugebunden werden. Dazu wird in die zuvor angelegte Datei mit dem Namen Binden ein sogenanntes Link Control File geschrieben. Es müssen nur der Name des Programms und die Datei angegeben werden, wohin das fertig übersetzte und gebundene Programm geschrieben werden soll. Welche Laufzeitmoduln und Bibliotheksunterprogramme hinzugebunden werden sollen, wird vom Binderprogramm, dem LED (Linkage Editor) eigenständig erkannt, es müssen vom Programmierer keine weiteren Angaben gemacht werden. Hat man zu seinem Fortran-Programm jedoch eigene Unterprogramme geschrieben, so müssen deren Namen hier angegeben werden und auch, in welcher Datei sie zu finden sind. Beispiel eines Link Control File:

```

PROG PROGRAM

INCL FD2.OBJEKT

END
    
```

Der Name des Programmes lautet Progrm. Das compilierte, noch zu bindende Programm steht in der Datei Objekt auf der Floppy-Diskette im Laufwerk 2. Dieses kleine Programm wird wie das Fortran-Programm mit Hilfe des Text-Editors in die Datei Binden geschrieben.

d) Nun muß die Bindung des Programms erfolgen. Dazu wird der LED aufgerufen.

Es müssen wie beim Fortran Compiler wieder einige Angaben gemacht werden:
Control File:
 Hier wird das zuvor erstellte Link Control File mit seinem Standort angegeben (Datei Binden).
Output File:
 Die vierte Datei (Progrm) wird hier angegeben, das fertige Programm wird auf diese Datei ausgegeben.

List File:
 Wie beim Compiler muß man angeben, wo die Bindung mitprotokolliert werden soll (beispielsweise Drucker).
 Hier erfährt man dann genau, welche Laufzeitmoduln und Bibliotheksunterprogramme zum eigentlichen Programm hinzugebunden werden mußten, um das Programm ausführbar zu machen.

Parameters:
 Ähnlich den Optionen beim Compiler können hier verschiedene List-Variationen gewählt werden:

A=All eine komplette Liste wird abgedruckt,
 S=Short nur die wichtigsten Informationen werden abgedruckt.

e) Das übersetzte und fertig gebundene Programm steht nun in der Datei mit dem Namen Progrm. Unter diesem Namen ist es jederzeit aufrufbar und ausführbar. Der Befehl zum Aus-

führen des Programms lautet in diesem Fall:

EX FD2.LIB.PROGRM

Die Datei Progrm steht als Member in der Library LIB auf der Floppy-Diskette, die sich im Laufwerk 2 befindet (FD2).

EX ist die Abkürzung für Execute (führe aus).

Die hier angewandte Technik ermöglicht jederzeit ein Ändern des Quelltextes, denn das ursprüngliche Fortran-Programm steht auch nach der Compilierung unverändert in der Datei Quelle.

ERLÄUTERUNGEN ZU DEN PROTOKOLLEN

1. Fortran-Compiler

Auf Seite 1 des Compiler-Protokolls wird das Fortran-Programm aufgelistet und die Zeilen werden durchnummeriert. Seite 2 enthält die Adressen, auf denen die im Programm verwendeten Variablen und Konstanten später zu finden sind. Die Adressen sind hexadezimal angegeben. Zum Beispiel beginnt das Feld A bei der hexadezimalen Adresse 0030 und ist 200 Byte lang. Das Feld wurde mit 50 Elementen dimensioniert, und jedes dieser Real-Elemente belegt 4 Byte (4 x 50 = 200). Weiterhin werden auf Seite 3 alle benötigten Unterprogramme aufgeführt, was für die spätere Bindung wichtig ist. In diesem Beispiel werden nur Unterprogramme aus der Laufzeitbibliothek benötigt (Runtime). Eigene Unterprogramme oder Unterprogramme aus der Fortran-Bibliothek werden hier nicht verwendet. Die Fortran-Bibliothek enthält Unterprogramme wie:
 SIN (x) Sinusfunktion von x
 COS (x) Cosinusfunktion von x
 SORT (x) Wurzelfunktion von x
 ABS (x)

Absolutbetrag von x undsoweiter.

Anschließend wird noch aufgelistet, wo die einzelnen Zeilen des Fortran-Programms zu finden sind (hexadezimale Adressen).

Die Startadresse ist in unserem Beispiel 0012, die Länge des Programmes beträgt 138 Byte, der für die Daten reservierte Speicherplatz ist 010A Byte groß.

Die Übersetzung wurde ohne Fehler beendet, das übersetzte Programm kann nun gebunden werden.

2. Linkage Editor

Hier wird das komplette Programm beschrieben. Name des Programmes: Progrm
 Start-Adresse (hex) 0000
 Länge (hex) 2ECO

Anschließend werden alle Moduln und Programmpakete (Unterprogramme) aufgezählt, die zusammengebunden werden.

\$MAIN ist das Hauptprogramm, also das oben aufgelistete Fortran-Programm. Startadresse, Länge und der Tag, an dem es in die Datei geschrieben wurde, werden ebenfalls angegeben.

\$DATA ist der reservierte Speicherplatz für lokale und temporäre Variablen und formale Parameter. Der Speicherbereich von \$DATA folgt immer direkt dem Hauptprogramm \$MAIN.

Die nachfolgende Moduln sind alle aus der Bibliothek Runtime. Sie enthält alle Unterprogramme, die es möglich machen, Fortran-Befehle ausführen zu können. Sie sind zu erkennen an ihrem Namen: Die ersten beiden Zeichen F\$ deuten auf ihre Herkunft hin. Wie beim Fortran-Compiler wird auch hier das Ende der Bindung angezeigt.

*Dipl.-Ing. (FH)
 Oliver Rosenbaum* □

EIN NEUER STREICH VON COMMODORE

Schneller - besser - kleiner - billiger

Es gibt Sachen, die gibt es gar nicht, aber die Floppy Disk 1541-II gibt es. Sie ist das Neueste aus der Commodore-Bauteile-Kiste. Ohne jegliche Ankündigung von Commodore tauchen in den Geschäften die neuen Frühjahrsmodelle des Computerherstellers auf (die alten Modelle 1541, 1541C müssen aber auch noch verkauft werden oder sind gar nicht mehr zu bekommen). Wir konnten zufällig eines dieser noch seltenen Exemplare mit der Serien-No. JA4 000609 ergattern.

Im modernen Design der 1571 zeigt sich nun auch das Floppy 1541-II-Diskettenlaufwerk: Eingriffmulde, Kipphebel für die Verriegelung der Diskette, Gehäusefarbe und Kühlrippen entsprechen genau dem Erscheinungsbild des größeren Bruders. Aber die Maße des neuen Laufwerk-Gehäuses lassen sich sehen: eine Länge von 256 mm, eine Breite von 184 mm und eine Höhe von 77 mm, dies ist weniger als unser Heftformat 210 x 297 mm, das spricht für die neue Bauteile-Integration.

Möglich ist dies durch die schmalen Mitsumi 5.25-Zoll-Laufwerke aus Japan, die nur noch eine Bauhöhe von 40 mm, eine Länge von 190 mm und eine Breite von 145 mm aufweisen; die Diskette selbst ist bekanntlich 133 x 133 mm groß. Nicht zu vergessen die neue Platine des 1541-II-Laufwerks, die eine entsprechende Bauteiledichte auf 180 x 112 mm nachzieht.

EINE PLATINE 180 x 112 mm

In einer Aluwanne von unten und dem Druckgußlaufwerk von oben ist die Platine gegen störende Einflüsse geschützt. Die größten Bauteile auf der Platine sind die zwei Buchsen für den seriellen Anschluß, die Power-Buchse und der Ein-/Aus-schalter, die sich mit dem Geräte-adresseumschalter (Device 8 bis 11), an der Rückseite der Floppy befinden.

Der Betrachter erkennt bei geöffnetem Gehäuse altbekannte Bausteine, wie den MOS 6502 AD-Microprozessor, zwei MOS 6522 VIA (Versatile Interface Adapters), Eingabe-/Ausgabe-Bausteine, einen 6116 RAM (Random Access Memory), 150ns 2KByte Puffer, einen MOS 251968-03 CBM ROM (Read Only Memory), der das 16 K DOS (Disk Operating System) enthält und einige Treiber- und Controller-Bausteine.

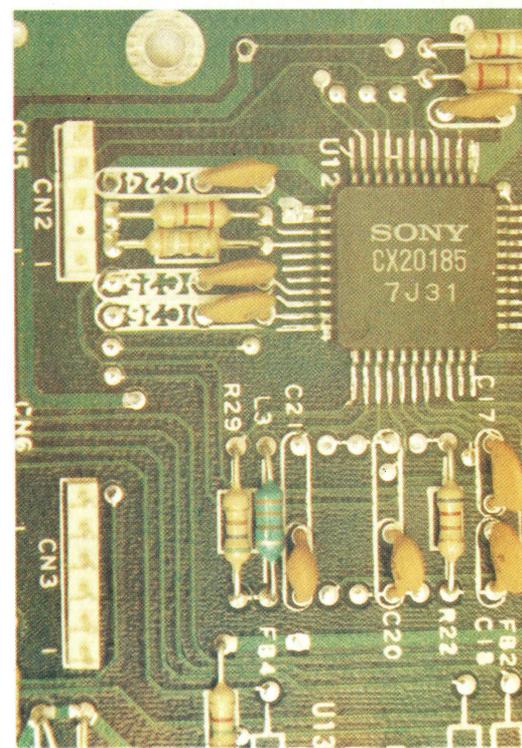
Also 14 Bausteine (20 Bausteine waren es noch beim alten 1541) benötigt das neue Platinenlayout für das 1541-II. Die alten Floppys benötigten zum Betrieb eine Betriebsspannung von fünf und 12 Volt, die neue 1541-II braucht nur noch eine 5-Volt-Spannungsversorgung, die überdies wärmeentwickelnde Spannungsregler überflüssig

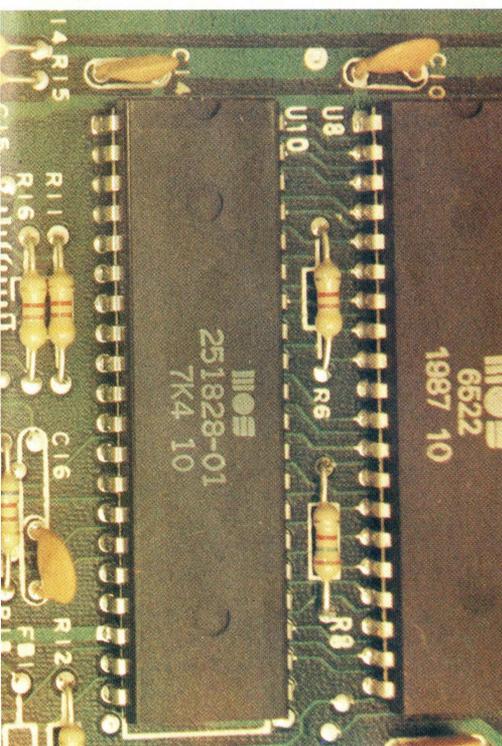
NUR NOCH 5 VOLT SPANNUNGSVERSORGUNG

macht. Die Überhitzungsprobleme des alten 1541 gehören der Vergangenheit an, auch weil das Netzteil in ein eigenes Gehäuse verbannt wurde. Wo hätten die Entwickler es auch noch hineinzwängen sollen bei diesem Floppygehäuse? Das separate Netzteil hat den elektronischen Vorteil, daß so gut wie keine Störstrahlung auf die Floppy einwirkt und ein aufwendiges Abschirmen wie bei der Floppy 1571 entfällt. Der Nachteil: Ein Bügel-eisen (Netzteil) mehr fliegt durch die Gegend, mal abgesehen vom zusätzlichen Kabelsalat. Der Anwender seufzt: „Wo gibt es einen Vorteil, der nicht einen Nachteil nachzieht? Wann gibt es eine zentrale Spannungsversorgung, wie sie zum Beispiel bei den neueren PCs üblich ist?“

WIE KOMPATIBEL IST DIE 1541-II?

Disketten, die von einem 1541-II-Laufwerk formatiert und beschrieben wurden, laufen alle auch auf anderen Commodore-Laufwerken, wie 1540 DOS 2.6, 1541 DOS 2.6, 1541C DOS 2.6, 1551 DOS 2.x, 2031 DOS 1.0, 2040 DOS 1.0, 3040 DOS 1.0, 4040 DOS 2.0/2.1 und 1571 DOS 3.0/3.1, also alle, die zum DOS 2.6 vom 1541-II schreiben und lesekompatibel sind. Mit DOS





1.0 und DOS 2.0 bzw. DOS 1.0 und 2.6 kann es Probleme bei Schreibzugriffen geben, die Disketten können aber wechselseitig gelesen werden, also alle Daten und Programme auf ihr Diskettenformat kopieren und Sie können wieder alle Diskettenbefehle ausführen.

EIN DISKETTENLAUFWERK FÜR ALLE?

Das heißt, das neue 1541-II-Diskettenlaufwerk kann seriell an den VC20, C64, C16, Plus4 und 128PC angeschlossen werden, verhält sich wie das alte 1541, auch der Sound des Stepperanschlags ist zu hören und auch so kritische Software wie Kopierprogramme oder Software-Floppybeschleuniger laufen, die das Floppy-DOS ganz schön durcheinanderwirbeln. Sicherheit hingegen bringt nur der einzelne Versuch. Denn bei diesem Angebot an Software alleine für den C64 gibt es von einzelnen Programmen mindestens drei Versionen, die unsere läuft, die Ihre vielleicht nicht, oder auch umgekehrt. Im großen und ganzen kann aber festgestellt werden, die von uns getestete Software läuft ohne Probleme auch auf dem neuen Commodore 1541-II-Laufwerk.

TIPS UND TRICKS

Die neueren Floppys bieten alle eine Hardware-Methode zur Änderung der Geräteadresse (Default 8), so auch das 1541-II. An der Rückseite des Gerätes befindet sich ein doppelter DIP-Switch, der mit einem kleinen Schraubenzieher auf die gewünschte Geräteadresse fest umgestellt wird. Ein Öffnen des Gehäuses ist dafür nicht nötig.

Links	Rechts	Adresse
Oben	Oben	8
Unten	Oben	9
Oben	Unten	10
Unten	Unten	11

Die Software-Methode, die nur eine vorübergehende Änderung der Geräteadresse bewerkstelligt, ist mit dem folgenden Programm oder im Direktmodus möglich, wobei Werte für GA von 8 bis 30 eingesetzt werden können und nur das zu ändernde Laufwerk eingeschaltet sein darf.

```
10GA=9
20OPEN15,8,15,"MEMORY-
WRITE"CHR$(119)CHR$
(000)CHR$(002)CHR$(GA
+32)CHR$(GA+64)
```

oder

```
OPEN15,8,15,"M-W"CHR$
(119)CHR$(0)CHR$(2)CHR$
(41)CHR$(73)
```

TECHNISCHE DATEN IM VERGLEICH

Da die in der Tabelle 1 genannten Disketten-Laufwerke, alle im Commodore-eigenen GCR-Disketten-Format schreiben oder lesen können (bis auf das Floppy 1571, das zusätzlich auch MFM-Format-Disketten schreiben und lesen kann), wird es bei einem Austausch der Disketten keinerlei größere Probleme geben. Nur die doppelseitig formatierten Disketten der 1571 (die hat ja zwei Schreib-/Leseköpfe), können auf den mit nur einem Schreib-/Lesekopf ausgestatteten 1541/1551-Modellen nicht ordnungsgemäß betrieben werden.

DISKETTENBEFEHLE

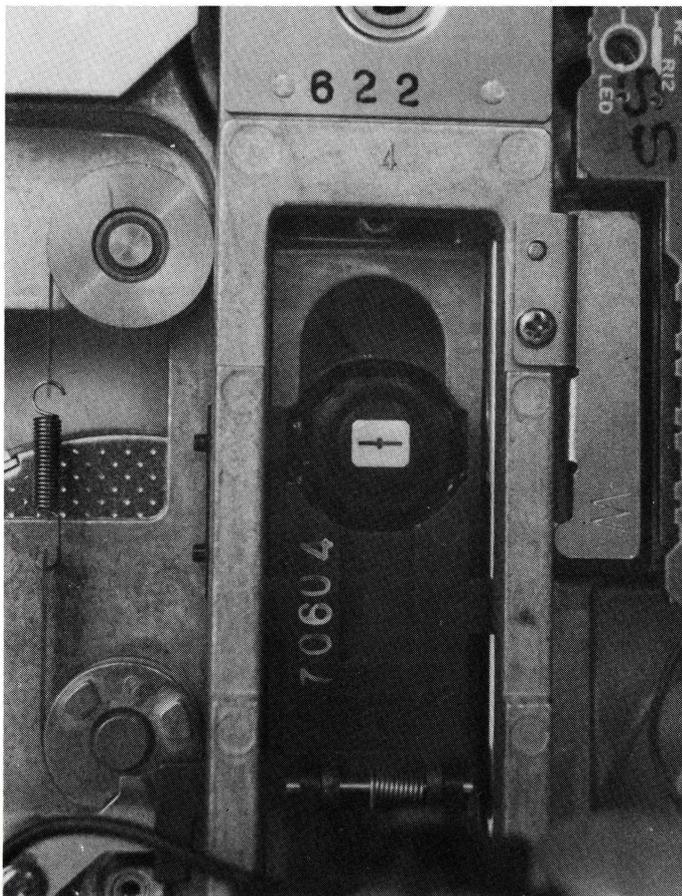
Alle Commodore-Rechner können die im allgemeinen Basic-2.0-Format gehaltenen Diskettenbefehle verarbeiten, ob sie nun VC20 oder 128 PC heißen. Höherwertige Commodore Basic-3.5-, 4.0- oder 7.0-Versionen bereiten dem Floppy keine Probleme, vorausgesetzt, der Rechner hat einen seriellen Anschluß und versteht die entsprechende Basic-Version. So ist es ohne weiteres oder mit Interface möglich, so alles was sich bei Commodore Computer nennt, seriell an das 1541-II-Laufwerk anzuschließen. Sogar ein Amiga mit 64er-Emulator kann angeschlossen werden.

RESÜMEE

Das Laufwerk 1541-II reiht sich in die Serie der neuen Produkte ein, die den Maßstab „immer kleiner, immer leistungsfähiger, immer preisgünstiger“ trägt. Neue Fertigungstechniken und der Dollar-Verfall helfen hier zusammen. Welcher Commodore-Gerätekonfiguration man es auch beistellt, das Laufwerk wird seinen Dienst leisten. Auch unser harter Test konnte dem Gerät nichts anhaben. Noch nachzutragen wäre, daß das Laufwerk 1541-II unter CP/M-Betriebssystem sich wie das alte Modell 1541 verhält. Vobis München stellte uns dankenswerterweise ein Textexemplar des Floppy 1541-II zur Verfügung. Robert Wagner □

SERVICE

Commodore Disk Drive	1541-II	1541A/C	1551	1571
GCR-Format	252019	252019	252019	504038
Kapazität (unformatiert in Bytes)	174848	174848	174848	349696
Kapazität (formatiert in Bytes)	168656	168656	168656	337312
Maximale Größe einer SEQ-Datei	167132	167132	167132	167132
Maximale Größe einer REL-Datei	65535	65535	65535	65535
Files pro Diskette	144	144	144	144
Spuren pro Diskette	35	35	35	70
Sektoren pro Spur	17-21	17-21	17-21	17-21
Sektoren pro Diskette total	683	683	683	1366
frei für Data	644	644	644	1328
Bytes pro Sektor	256	256	256	256
Chips:				
CPU Mikroprozessor MOS	6502AD	6502	6502	6502A
VIA Ein-/Ausgabe Timer Baustein MOS	6522	6522	6522	6522
RAM Bufferspeicher 2K	6116	2114	6116	6116
ROM Betriebssystem DOS	16K	16K	16K	32K
Geräteadresseumschaltung Hardware	8-11	-	-	8-11
Software	8-30	8-30	8-30	8-30
Sekundäradressen	0-15	0-15	0-15	0-15
Schreib-/Leseköpfe	1	1	1	2
Abmessungen:				
Höhe mm	77	97	97	76
Breite mm	184	200	200	216
Länge mm	256	374	374	346
Betriebsdaten:				
Spannung Volt	220-240	220-240	220-240	220-240
Frequenz Hz	50	50	50	50
Leistungsaufnahme Watt	25	25	25	25
Netzteil eingebaut separat	+	+	+	+
Preis in DM	388.-	388.-	279.-	529.-



Befehlsliste der Commodore-Laufwerke

Grundformat BASIC 2.0
 OPEN15,8,15:PRINT#15,Befehl:CLOSE15

Verwaltungsbefehle:

BASIC 2.0
 NEW/COPY/RENAME/SCRATCH/VALIDATE

OPEN15,8,15,"N0:Diskettenname",ID":CLOSE15
 OPEN15,8,15,"C0:Neue Datei",=0:Alte Datei",":CLOSE15
 OPEN15,8,15,"R0:Neuer Name",=Alter Name",":CLOSE15
 OPEN15,8,15,"S0:Dateiname",":CLOSE15
 OPEN15,8,15,"V0:":CLOSE15

BASIC 3.5/7.0
 NEW/COPY/RENAME/SCRATCH/VALIDATE bzw. COLLECT

HEADER"Diskettenname",Ixx,Ux,Dx
 COPY"Alte Datei"TO"Neue Datei"
 RENAME"Alter Name"TO"Neuer Name"
 SCRATCH"Dateiname"
 COLLECT
 BASIC 2.0/3.5/7.0

INITIALIZE

OPEN15,8,15,"I0":CLOSE15

Dateibefehle:

BASIC 2.0
 LOAD/SAVE/VERIFY

LOAD"Dateiname",",8
 SAVE"Dateiname",",8
 VERIFY"Dateiname",",8

BASIC 3.5/7.0
 LOAD/SAVE/VERIFY/DLOAD/DSAVE/DVERIFY

DLOAD"Dateiname","
 DSAVE"Dateiname","
 DVERIFY"Dateiname","

BASIC 7.0
 LOAD/SAVE/VERIFY/DLOAD/DSAVE/DVERIFY/BLOAD/BSAVE/BOOT/DOPEN/DCLOSE/RECORD#

BLOAD"Dateiname",",Dx,Ux,ON Bx,Pxxxxx
 BSAVE"dateiname",",Dx,Ux,ON Bx,Pxxxxx TO Pxxxxx
 BOOT"Dateiname","
 DOPEN# Datei,"Dateiname",",L Eintragslänge,W
 DCLOSE# Datei
 RECORD#,Datei,Eintragsnummer,Offset

BASIC 2.0/3.5/7.0
 OPEN/CLOSE/RECORD#/PRINT#/GET#/INPUT#

OPEN Datei,Gerät,Kanal,"0:Dateiname,Dateityp,Richtung"
 CLOSE Datei
 RECORD# "P"+CHR\$(Kanal)+CHR\$(>Eintrag)+CHR\$(>Eintrag)+CHR\$(Offset)
 PRINT# Datei,Datenliste
 GET# Datei,Variablenliste
 INPUT# Datei,Variablenliste

Direktzugriff Befehle:

BASIC 2.0/3.5/7.0
 BLOCK ALLOCATE/BLOCK-EXECUTE/BLOCK-FREE/BUFFER-POINTER/BLOCK-READ/
 BLOCK-WRITE/MEMORY-EXECUTE/MEMORY-READ/MEMORY-WRITE/USER/UTILITY LOADER

BLOCK-ALLOCATE "B-A";0;Spur#;Sektor#

**Jetzt gibt es
Deutschlands erste
Commodore-Zeitschrift
mit Programm-Diskette
für Ihren 64er und 128er!**

**COMMODORE
DISC
C64/
C128**

**Bis zu 180 kB Programme
ohne Abtippen!**

COMMODORE DISC

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

COMMODORE DISC

550

COMMODORE WELT
SPECIAL

CW-SPECIAL
P/4
C 16
116

SOFT-WARE
JAHR-BUCH
1988

Super-Spiele für Ihren 16/116 und Plus 4!
Die besten Anwender-Programme

DAS BESTE AUS CW
Sammelband Nr. 1/88
DM 14,80-ÖS 124-SFR 14,80



```

100 PRINT"-----"
110 PRINT"THE BEST OF COMMODORE-WELT"
120 PRINT"-----"
130 LOAD "LISTING 1" : RUN
140 LOAD "LISTING 2" : RUN
150 LOAD "LISTING 3" : RUN
160 LOAD "LISTING 4" : RUN
170 REM "-----"
180 REM "IF YOU WILL LOAD AND RUN,"
190 REM "YOU WILL HAVE A LOT OF FUN"
200 REM "-----"
    
```



Zwei starke Helfer für jeden C16/P4-User!

Test-Jahrbuch 88
Software-Jahrbuch 88

Alles über und für den C 16/P 4

An ausgewählten Kiosken und im Bahnhofs-Buchhandel

COMMODORE WELT
SPECIAL

P/4
C 16
116

TEST-JAHR-BUCH 1988

TEST-JAHRBUCH 1988 C 16

DAS BESTE AUS CW
Sammelband Nr. 3/88
DM 19,80 - ÖS 158 - SFR 19,80






Tests
Tips
Tricks
Kaufberatung

Alles über Ihren C16/P4

Das Handbuch für C16/P4