

Die Nummer 1 für C64 und C128

64'er

64'er

DAS MAGAZIN FÜR COMPUTERSPIELER

Das beste C-128-Kopierprogramm: **HEXER V1.4**

C64 inside

So entsteht ein Betriebssystem

- **Auf Speichersafari:** So nutzt man Systemroutinen!
- **Alternative Systeme:** CP/M, CS-DOS, Cobol

Software-Tests

- **Laser Genius:** traumhaftes Programmier-Tool
- **Laser Compiler:** Basic – schnell wie Assembler!

Programme auf Disk

- **Robox:** rasantes Actiongame
- **Tools** zu VisAss und Basic-Assembler
- **Packer-Pack:** Daten in der "Schnellpresse"
- **1541-DOS:** PC-Disketten mit dem C64 lesen

Neue Super-GAMES: **Motley Tetris** und **Jetliner**



BITTE NICHT KNICKEN - DATENTRÄGER IM HEFT!

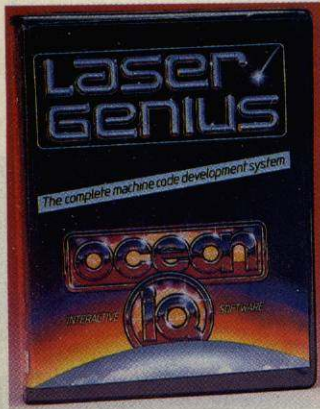
SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

INHALT 9/94



45 Die Alternative für alle, die von Basic 2.0 die Nase voll haben: Mit "Laser Genius Assembler" wird Maschinensprache zum Spaziergang!



Ihren Haushalt über den C 64 zu steuern, ist jetzt mit dem neuen "Tele-Commander" von EHA-Elektronik möglich. Medium: das Stromnetz!

51

Aktuell

News & Facts	4
Zehn Jahre Plus/4	5

Betriebssysteme

C-64-Speichersafari:	
Betriebssysteme im Überblick	6
Programmieren mit Basic	10
Mehrsprachig auf vielen Plattformen:	
Betriebssystem-Entwicklung	13
Ins rechte Licht gerückt:	
CP/M – Control Program for Microcomputers	14
Das Multitalent:	
CS-DOS für den C 128	17

Tips & Tricks

Schlagwörter zum Nachschlagen (Folge 4):	
Computer-Lexikon	23

Programmieren

Gamers-Programming-Guide (Folge 5)	26
Power für den Boss: Effizientes Programmieren mit dem Basic-Boss	29
Computer auf der Schulbank (Folge 3):	
Künstliche Intelligenz	30

DFÜ

Programme à la carte:	
Internet: Dateien per Post	34

Software

Hexbytes zum Schleuderpreis:	
Assembler/Monitor 64	36
Programmier-Tools:	
U.a. Basic-Assembler-Converter, VIS-Ass-Notizbuch, VIS-Ass und SMON	38
Utilities:	
Packer-Pack	39
Heiße Schlacht:	
Action-Game Robox	40
Programmieren in Umgangssprache?	
Interpreter: Cobol 64/128	42
C-128-Kopier-Tool:	
Neues vom Hexer II	43
Treffpunkt Btx:	
Btx-Anbieter mit dem C 64	44
Heißer Draht zur Maschinensprache:	
Laser Genius Assembler	45
Serie (Folge 1):	
How to GoDot	48

Hardware

Butler im Lehnstuhl:	
Der Tele-Commander	51

Spiele

Diverse, raffinierte Tips zu:	
Harakiri, Dallas Quest und Turn'n Burn u.a.	52
Longplay (Folge 1):	
Spirit of Adventure	58
Im Test:	
Motley Tetris und Jetliner	61

Rubriken

Inserentenverzeichnis	16
Diskettenseiten	20
Impressum	22
Kleinanzeigenauftrag	22
Leserforum	25
Vorschau	62



Seite 13

Seite 45

Seite 38

Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Zusatzsoftware zum Drews-Decoder

“Btx-Extra” heißt die neue Zusatzsoftware zum 64er-Btx-Decoder 1.6 der Firma Drews. Es wurden folgende neue Features integriert (und damit erstmals ähnlicher Komfort erreicht, wie ihn das Btx-Modul schon länger bietet):

- vollautomatische Anwahl inkl. Eingabe von Anschlußkennung und Paßwort
- Texterstellung offline, d.h. Sie sparen Zeit und Geld
- Makros (automatische Befehlsfolgen) in beliebiger Länge und Anzahl
- Komfortabler Makroeditor

(schreibt Ihre Aktionen mit)

- booten von jedem Laufwerk
 - Laufwerkswechsel on-/offline
- Im 128er-Modus wird zusätzlich geboten:

- Nutzung des Zehner-Blocks
- schnelles Booten
- Makroaufruf durch Sonderfunktionstasten

Die Software von Programmierer Wolfgang Grimm (für die der Drews-Btx-Manager Voraussetzung ist) können Sie für 9 Mark direkt in Btx laden, und zwar unter:

- *732#*92250222# oder
- *719#*94191930#

Maverick in Deutschland



Das Hauptmenü des Maverick

Das Utility-Paket “Maverick” V 5.1 wird es in Kürze auch in Deutschland geben. Die Software, die einerseits aus einer Vielzahl von Kopierprogrammen für jeden Zweck, andererseits aus interessanten Disketten-Utilities (Dir-Editor, Textanzeiger usw.) besteht, gibt’s bei Performance Peripherals, der Preis steht allerdings noch nicht fest. Besonders interessant ist das Paket sicher für Besitzer von 1581-Laufwerken, denn die meisten

Tools (u.a. auch Single- und Double-Drive-Kopierer) arbeiten auch mit dem 3,5-Zoll-Laufwerk. Sehr komfortabel: Ram-Erweiterungen und 64-KByte-VDC des C 128 werden unterstützt. Übrigens soll auch die Zusatz-Hardware “Speed-Control-Box” und “RamBoard” in Deutschland verkauft werden.

Das 64-KByte-VDC-Upgrade für den C 128 wird jetzt kinderleicht: Performance Peripherals bietet in Kürze ein kleines Modul an, das ohne jegliche Lötarbeiten auskommt. Unterstützt wird es u.a. auch vom Maverick.

Performance Peripherals, M. Renz, 53332 Bornheim

Neue Version des C64S

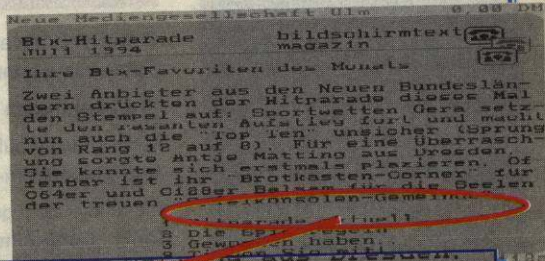
Der C-64-Emulator “C64S” von Miha Peternel ist soeben in der endgültigen Version 1.0 erschienen. Als Hardware-Minimum wird zwar ein 386-PC empfohlen, doch damit schleicht der Emulator nur so dahin. Ein 486 DX sollte es schon sein, damit echtes C-64-Feeling aufkommt. Im Vergleich zur Version 0.9 wurde vor allem an der Kompatibilität gefeilt. Der volle 6510-Befehlssatz wird nun unterstützt, VIC-Interrupts (außer Lightpen) sind zugelassen, bei CIA 1 wird nun auch Timer B emuliert und bei CIA 2 beide Timer und der RS-232-Port. Besonders sinnvoll: Ein Utility “COM1541”

zum Direkt-Transfer von 1541-Daten zum PC wird nun (in der registrierten Version) mitgeliefert. Die 1541 auch unter dem Emulator am PC laufen zu lassen, ist allerdings erst für die nächsten Versionen vorgesehen. Zur Zeit wird die registrierte Version nur in den USA für 69 Dollar vertrieben, europäische Interessenten wenden sich an den Programmierer selbst.

Miha Peternel, Pot v Bitnje 66, SI-6400 Kranj, Slovenia, oder per EMail: miha@hermes.si

Aus zwei mach eins

Zwar ging es nicht durch die Weltpresse, aber auch in der C-64-Welt gibt’s manchmal “geschäftliche” Veränderungen. Seit 1.8. nämlich hat die nicht unbekannt Shareware-Versandfirma MasterMMSoft den Besitzer gewechselt. Aber damit soll sich für den User nichts ändern. Ein weiteres interessantes Detail: Mit *geos# und *matting# waren im Juli erst-



Soviel Ehre für einen Spielcomputer ...

mals seit längerer Zeit wieder zwei Btx-Angebote für den C 64 in den Btx-Top-20 zu finden. Der Kommentar der Redaktion vom “bildschirmtext magazin”, in dem der C 64 als “Spielkonsole” bezeichnet wurde (nachzulesen auf Seite *34344112#), zeugte allerdings nicht von allzuviel Sachkenntnis (aber das ist der C-64-User ja schon fast gewöhnt).

Flash-8-News

Das endgültige neue “Konfigurieren”-File für die Flash 8 unter Geos ist inzwischen fertig. Damit ist es möglich, auch den internen Speicher der 1-MByte-Variante als RAM-Floppy anzusprechen. Das neue File gibt’s sogar in zwei Varianten: eine von Discount 2000, dem Flash-8-Hersteller, und eine weitere von Performance Peripherals

(PP), die zusätzlich noch eine über Winkeladapter angeschlossene GeoRAM oder BBG unterstützt.

Bei Neubestellungen gib’s die Software automatisch zur Flash 8 dazu, nachträglich sind zumindest bei PP 13 Mark inkl. Porto zu bezahlen. Der Winkeladapter für die externe Ram-Erweiterung kostet bei PP genau 24 Mark.

RamLink-Commander

Vorbild “Norton” läßt grüßen: Mit dem “RamLink-Commander” steht eine weitere Software für “Power-User” mit C 128, 80-Zeichen-Farbmonitor und beliebiger CMD-Hardware zur Verfügung (40 Mark). In der “Public Diamonds”-Reihe erscheint in Kürze das Shareware-Strategie-Spiel “Starfleet Command” für den C 128 in bewährter Ausstattung (gedruckte Anleitung, 19,80 Mark). Eine Ankündigung für Adventure-Freaks: Im Oktober wird ein neues Adventure-Construction-Set veröffentlicht, mit dem sich Grafik-Text-Adventures mit Joystick- und Mausbedienung zusammensetzen lassen (49 Mark).

Die GUSSen kommen

Auch von “Geos User Software Sachsen” gibt’s Neuheiten zu melden: GeoCom wird jetzt endgültig ausgeliefert. Ab August wird die erste “Tips&Tricks”-Diskette zu GeoCom vertrieben (20 Mark, mit vielen interessanten Quelltexten). “The Best of” wird mit dem zweiten Teil fortgesetzt, der außer Utilities vor allem einen komfortablen Dialogbox-Editor für MegaAss-Programmierer enthält. Als zusätzlicher Service bietet GUSS jetzt an, Software auch auf 3,5-Zoll-Disketten zu verschicken. Und last but not least: Direktbestellung und Kataloganforderung kann nun direkt über Btx erfolgen: *732#*922502#.

Independent Softworks, 79199 Burg a.W.

GUSS, Denis Döhler, 04347 Leipzig

Zehn Jahre Plus/4

Der Plus/4 lebt!

1994 wurde der Plus/4 zehn Jahre alt. Grund genug, seine Geschichte noch einmal Revue passieren zu lassen und über den "Stand der Technik" zu berichten.

Im Jahre 1984 wurde der Commodore 264 auf der Hannover Messe vorgestellt, um später unter dem Namen Plus/4 auf dem deutschen Markt zu erscheinen. Den Namen bekam der Plus/4 von der eingebauten 3+1-Software, die sich jederzeit abrufbereit in einem 32-KByte-ROM befindet. Diese

den Breakout-Clones), "Botticelli" oder "Hungaroring" geschrieben. Später wurde der Plus/4 revolutioniert, als die ersten Programm-konvertierungen vom C 64 auf-tauchten. Code wie Pigmy oder Mucsi/Muffbusters stellten 1:1-Konvertierungen von Spielen, u.a. "Revs", "Bard's Tale III", "Bor-

lösenden Grafik 320*200 Punkte mit voller Basic-Unterstützung und es stehen 121 Farben zur Verfügung. Der Sound be-schränkt sich auf zwei Tongenera-toren. Leider

wurden die Schnittstellen des Plus/4 gegenüber dem C 64 grundlegend geändert. Ein weiter-er Nachteil gegenüber dem C 64 sind die fehlenden Hardware-Sprites, währenddessen der Plus/4 mit einigen Verbesserungen ausge-stattet wurde: Er bekam einen Re-set-Schalter, einen Erweiterungs-port zum Anschluß einer paralle-len Floppy (der 1551), vier ge-trennte Cursor-Tasten, vier Tasten mit acht belegten und änderbaren Funktionen, eine richtige RS232-Schnittstelle und die Möglichkeit, die interne 3+1-Software durch ei-gene EPROMs zu ersetzen.

256 KByte Standard

Die aktuellen Verbesserungen dieser Hardware lassen sich eben-falls sehen. Zuerst wurde der Speicher standardisiert, indem Hans Lux und Christian Schäf-fner eine 256-KByte-Erweiterung entwickelten, die von Zoltan Csordas überarbeitet wurde. Als zweites Meisterstück gilt die SYNERGY-SID-Karte von Cristi-an Schäfner, der den SID aus dem C 64 auf eine Steckkarte für den Erweiterungsport des Plus/4 gebannt hat. Da die SID-Karte auch noch einen Joystickport be-sitzt, kann man neben Joysticks und Mäusen auch einen PC-Ana-log-Joystick über einen Adapter von Hans Lux anschließen. Als Erweiterung für die SID-Karte hat Christian Schäfner auch den Di-giblastler entwickelt. Diese Zu-satzkarte enthält einen 8-Bit-Sampler und ist fähig, 8-Bit-Digis und MODs auszugeben. Durch den 6551-Baustein kann der Plus/4 mit dem gerade veröffent-lichten Btx-Soft-ware-Decoder 2400 bps errei-chen. In der näch-ten Zeit wird es auch ein Termi-nal-Programm mit 80-Zeichen-Darstellung ge-ben.

Neben der SID-Karte und dem Digiblastler gibt es noch wei-tere Hardware für

Hallo Plus/4-Fans

Mit dieser Seite wollen wir eine re-gelmäßige Plus/4-Ecke starten. Einerseits werden wir News vorstel-len, andererseits gibt's auch Tips & Tricks zu Hard- und Software. Wir hoffen, daß Nur-C-64-Besitzer das "Einsprengsel" akzeptieren, vielleicht erhält ja auch die C-64-Szene ein paar Anregungen von den Plus/4-Bastlern?

Musiker. Mit dem MIDI-Inter-face wird ein Synthesizer am Erweiterungs-port des Plus/4 angeschlossen. An der MIDI-Software wird gerade geschrie-ben, während

die SID-Karte voll unterstützt wird. Eine weitere Neuentwick-lung für die 256-KByte-Erweite-rung ist "Page-Setter-Quer". Mit Page-Setter ist man in der Lage, Text und Grafiken z.B. für Brief-köpfe oder Werbeplakate zu ver-binden.

Geos auf Plus/4

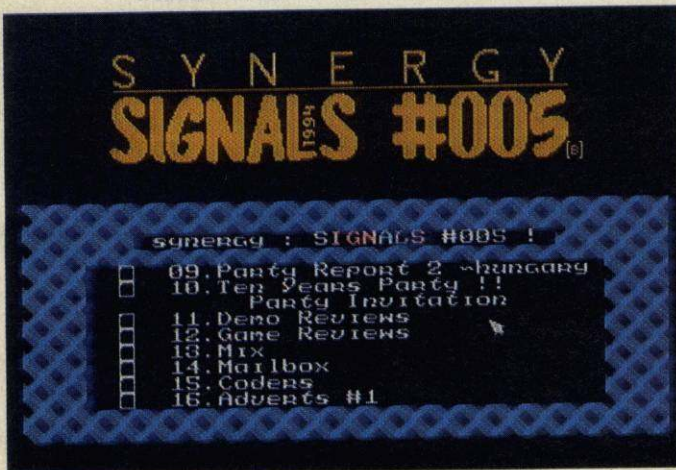
Ein weiteres interessantes Pro-gramm ist "GEOS V3.5". Diese C-64-Konvertierung läuft nur mit der Floppy 1551. Die Standard-Applikationen sind kompatibel zur C-64-Version und auch um-gekehrt laufen viele Programme auf dem Plus/4. Leider ist auch bei diesem Programm die rech-tliche Seite umstritten.

Für Grafik-Fans wird gerade "GoDot" von Arndt Dettke für den Plus/4 gecodet. GoDot, das Bildverarbeitungsprogramm Nr. 1, ist die ideale Ergänzung zu "Graudruck" von W. Hai. Dieses Programm erlaubt den Ausdruck farbiger Grafiken in Graustufen.

Als herausragendes Programm für die Komposition, Erzeugung und Verarbeitung von 4- und 8-Bit-Digis gilt das "Digital Com-posing Kit". Es wurde von Chri-stian Prieth programmiert, der auch für die ersten MODs auf dem Plus/4 verantwortlich war.

Die vorgestellte Hard- und Software ist nur noch bei Privat-personen erhältlich, da Firmen kaum noch Interesse am Plus/4 haben. Da der Plus/4 die gleiche Peripherie wie der C 64 benutzt, ist er bestimmt auch für C-64-User interessant. Vielleicht wird er ja mehr als ein Zweitcomputer?

Andreas Friedemann / ma



Hauptmenü des eben erschienenen Diskmags "Signals" Nr. 5

wirklich schlechten Programme und der Einführungspreis von 1298 Mark waren wohl verant-wortlich dafür, daß sich der Plus/4 nie richtig durchgesetzt hat. Auch die Verkaufskampagne von Com-modore, die im Jahre 1986 den Plus/4 über Kaufhausketten "ver-scherbelte", konnte nichts am frühen Ende des Plus/4 ändern. Die Produktion wurde gestoppt, die Restposten hat man verkauft und 1989 wurde die letzte Spezial-Zeitschrift abgesetzt.

War dies das Ende des Plus/4? Nein, denn gerade 1989 kamen die ersten Freeware-Programme, vor allem aus Ungarn. Dort wurde der Plus/4 in den Schulen einge-setzt, was zur Folge hatte, daß er zum ersten Mal wirklich ausge-nutzt wurde. Während man in Deutschland nur "Mercenary" oder "ACE" kannte, wurden in Ungarn Programme wie "Digital Ball" (immer noch führend unter

rowed Times", "Tass Times", "Barbarian" oder "Elite" her. Lei-der sind die Spiele nie offiziell von den Softwarefirmen freigege-ben worden bzw. die wußten gar nichts von diesen Programmen.

Hardware weiterentwickelt

Die Hardware des Plus/4 hat sich ab 1990 gegenüber dem Plus/4 von 1984 stark verbessert. Der Plus/4 hat offiziell einen Spei-cher von 64 KByte, davon waren 60 KByte voll vom Basic V3.5 nutzbar. Das komfortable Basic befindet sich, wie das Betriebs-system und der residente Maschi-ne-Sprache-Monitor, in einem 32-KByte-ROM. Die CPU (MOS 7501) ist softwarekompatibel zum MOS 6502 und kann bis 1,76 MHz getaktet werden. Die Grafik und den Sound verwaltet der TED (8360R2). Dieser Multifunktions-baustein erlaubt bei der hochauf-

PARTY-EINLADUNG

Am 2. Oktober 1994 findet die alljährliche PLUS/4-Par-ty statt. Zum 10. Geburtstag des PLUS/4 trifft man sich von 10 bis 22 Uhr im Jugendclubhaus "Trarbacher 23", Trarbacher-Str. 23, 13088 Berlin-Weißensee. Die Ver-anstaltung bietet 150 Usern Platz und ist mit der Straßenbahn-Linie 21 oder der Bus-Linie 155 errei-chbar. Der Eintritt kostet 10 Mark, wovon ein Teil für die Programmier-, Grafik- und Sound-Wettbewerbe ver-wendet wird. Die Competition findet von 17 bis 19 Uhr statt. Weitere Infos in dem aktuellen Plus/4-Disk-Ma-gazin "Signals 5" oder über die SYNERGY-Hotline (08161-63791, Andreas Friedemann).

Chip-Rundreise

C64 - SPEICHERSAFARI!

Um den C 64 zum Leben zu erwecken, benötigt er ein "Gehirn", das ihn steuert. Das Betriebssystem übernimmt in der Praxis diese Aufgabe. Wir machen einen Streifzug durch den Speicher und zeigen Ihnen, was das Betriebssystem zu bieten hat.

Haben Sie sich schon mal gefragt, was da passiert, wenn man im Direktmodus dem C 64 Kommandos übergibt und er sie auch brav ausführt? Da muß ein guter Geist im Hintergrund arbeiten und die Bausteine des Computers kontrollieren. Es ist das Betriebssystem des C 64, das ins ROM eingebaut ist. Das ROM (Read only Memory) ist ein Chip, der zwar aus seinem Speicher le-

Zero-Page kopieren

Wer oft Manipulationen der Zero-Page (Page 0) oder folgenden drei Seiten vornimmt, sollte sich diese Speicherstellen in einen nicht benutzten RAM-Bereich kopieren. Bei Bedarf kann man so schnell den ursprünglichen Zustand wieder herstellen, indem man aus der "Kopie" die Werte wieder zurückschreibt.

sen, aber dem User das Schreiben auf die Speicherstellen verweigert. Das Gegenteil ist das RAM (Random Access Memory), das beliebiges Schreiben und Lesen aller Speicherstellen zuläßt.

Das Betriebssystem des C 64 wird auch Kernall genannt. Es besteht aus zahlreichen Unterrou-tinen und ist ein mächtiges Programm-paket, daß sich der User in vielen Fällen zunutze machen kann.

Was, wie und wo?

Um das Betriebssystem des C64 zu nutzen, muß man natürlich wissen, an welcher Stelle in der Speicherlandschaft sich der Kernall befindet. In Bild 1 finden

Sie ein Schema, was die Standard-belegung des C 64 verdeutlicht. Dabei muß man sich den Speicher des C 64 wie eine lange Straße vorstellen, an der jede Speicher-stelle wie ein Haus fungiert und einem Byte entspricht. Tabelle 1 führt noch einmal in dezimaler und hexadezimaler Schreibweise die einzelnen Bereiche auf.

Wenn wir uns die Speicherland-schaft vom Anfang aus betrach-ten, finden wir am Start die Pages (Seiten) 0 bis 3, die je 256 Bytes lang sind. In ihnen stehen wichtige Zeiger (Pointer), die Informatio-nen zur Steuerung des C 64 ent-halten. Sie werden intensiv durch das Betriebssystem genutzt und können durch den User auch ver-ändert werden. Einige Bereiche bleiben durch den C 64 unange-tastet und können zur Zwi-schenspeicherung genutzt werden. Die Belegung der Pointer findet man im Handbuch des C 64 im Anhang.

Die wichtigste Speicherstelle in diesem Bereich ist die Nummer 1. Hier werden die Bänke des C 64 umgeschaltet. Dazu ist zu bemer-ken, daß die Entwickler des C 64 verschiedene Bereiche des Spei-chers doppelt oder sogar dreifach belegt haben. Deshalb liegen an manchen Stellen ROM- und RAM-Bereiche übereinander und können je nach Bedarf genutzt oder deaktiviert werden. Wer das Handling mit den Banken beherrscht, kann seinem Lieblings-computer 64 KByte RAM ent-locken und zusätzlich das Be-triebssystem nutzen. Viele Ma-schinensprache-Programmierer (Assembler) schalten deshalb das Basic aus und nutzen dafür das darunter befindliche RAM für ei-

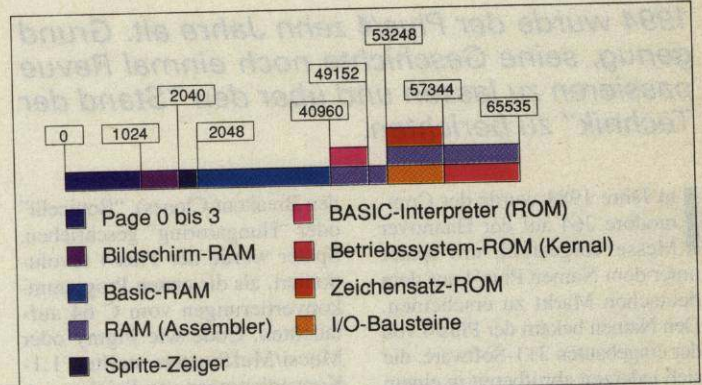


Bild 1: Der komplette Speicher des C 64 mit seinen Memory-Banks auf einen Blick, alle Adressen noch einmal in Tabelle 1

So werden Kernall-Routinen genutzt

Um eine Routine des Kernall in eigenen Programmen zu nutzen, müssen als erstes alle relevanten Parameter übergeben werden. Dann wird die Routine per JSR-Befehl (in Basic SYS) angesprochen. Nach Erledigung der Aufgabe übergibt der Kernall wieder die Kontrolle an das eigene Maschinensprache-Programm. Nun muß die Auswertung eines eventuell aufgetretenen Fehlers und seine Behandlung folgen. Gibt's keinen Error, kann das Ergebnis der Kernall-routine weiterverarbeitet werden. Kommt es zu einem Fehler während der Abarbeitung einer Kernall-Routine, wird das Übertragungs-Bit des Akkumulators gesetzt und eine Fehler-Kennziffer im Akku übergeben. Einige Routinen arbeiten nicht nach diesem Verfahren und statt dessen wird die Betriebs-system-Routine READST (s.Tabelle 4) benutzt. Zum Abschluß noch die Fehlercodes:

Nummer	Bedeutung
0	Routine durch STOP-Taste beendet
1	zu viele Dateien offen
2	Datei bereits offen
3	Datei nicht offen
4	Datei nicht gefunden
5	Gerät nicht vorhanden
6	keine Eingabe-Datei
7	keine Ausgabe-Datei
8	Dateiname fehlt
9	unzulässige Gerätenummer
140	Speicherende verändert (RS232C)

gene Zwecke. Es ist auch möglich, für eine bestimmte Zeit ROM-Bereiche auszublenden und

das RAM im selben Abschnitt als Puffer zu nutzen.

Mit dem Befehl POKE 1,X kann die Speicherstelle leicht manipuliert werden und gewünschte Bereiche ein- bzw. ausgeblendet werden. Vorsicht ist aber geboten, denn wenn man den I/O-Bereich ausblendet, nimmt der C 64 keine Eingabe mehr an und es kann zu einem fatalen Absturz des Systems führen. Wollen Sie den genannten Bereich ausblenden, sollten Sie alle Eingaben per Tastatur sperren. Maschinensprache-Programmierer unter-

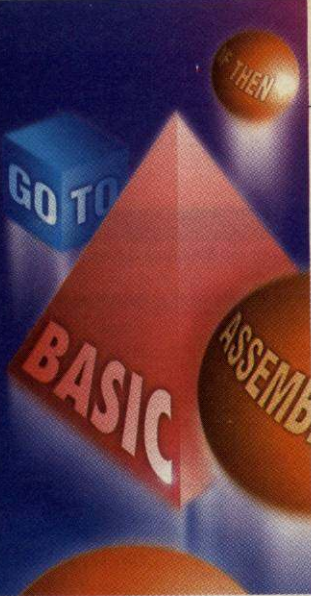
brechen alle Interrupts mit dem Befehl SEI und dürften, so lange sie mit den Bänken des C 64 spielen, ihre Ruhe vor Abstürzen haben. Nach den Operationen und Restaurierung des bekannten Zustands, kann man mit CLI Interrupts wieder zulassen. Wie man die Bänke des C 64 aktiviert, können sie aus dem Schema in Bild 2 entnehmen.

Einige frei Speicherstellen, die man als Programmierer für eigene Zwecke benutzen kann, finden Sie in Tabelle 2.

Soweit erst einmal zu den ersten vier Pages. Genauere Informationen finden Sie wie gesagt im

Tabelle 1: Der Speicheraufbau des C 64

Speicherbereich		Bedeutung
dezimal	hexadezimal	
0-1023	\$0000-\$03ff	Page 0 bis 3
1024-2039	\$0400-\$07ff	Bildschirm-Speicher
2040-2047	\$07ff-\$07ff	Sprite-Pointer
2048-40959	\$0800-\$9fff	Basic-Speicher
40960-49151	\$a000-\$bfff	Bank 1: RAM
		Bank 2: Basic-Interpreter
		RAM
49152-53247	\$c000-\$cfff	Bank 1: I/O-Bausteine
53248-57343	\$d000-\$dfff	Bank 2: RAM
		Bank 3: Zeichensatz-ROM
		Bank 1: Kernall
		Bank 2: RAM
57344-65535	\$e000-\$ffff	



Packer, die diese Bereiche zum Entpacken nutzen. Das Ausweichen auf Bildschirm- bzw. Color-RAM wird erst interessant, wenn man den kompletten Speicher des C64 ausgenutzt hat. Erst dann sollte man zu dieser Variante greifen. Wer dennoch Teile des Bildschirms für eigene Programme nutzen

wender bekannt sein dürfte. Ab hier werden die meisten Programme abgelegt. Maschinensprache-Programmierer könne diesen Bereich selbstverständlich für ihre Zwecke nutzen. Der Start des Basic-Speicher kann durch Manipulation von Zeigern in der Zero-Page verändert werden. Näheres zu diesem Thema finden Sie in der Basic-Corner-Ausgabe 12/92. In den Bereich ab 49152 (hex. \$c000) können ohne weiteres für Maschinensprache-Programme

Handbuch des C 64. Ab und an werden wir im Verlauf diese Artikels, auf diesen Bereich zurückkommen und genauer auf einige spezielle Vektoren (Pointer) eingehen.

Das RAM im Detail

Als nächstes wollen wir das RAM ein wenig genauer unter die Lupe nehmen. Ab Speicherstelle

will, sollte das Farb-RAM, entsprechend der gewählten Bereiche, gleich der Hintergrundfarbe setzen (s. Listing). In diesem Falle läuft das Programm quasi unsichtbar für den Nutzer ab, da Zeichen die im Bildschirm stehen, die gleiche Farbe wie der Hintergrund haben. Im Anschluß an den Bildschirmspeicher folgen 8 Bytes, in denen die Form der 8 Sprites be-

Tabelle 2: Freie Speicherstellen der Pages 0 bis 3

Speicherstelle	Besonderheit
02/\$02	problemlos verwendbar
158-159/\$9e-\$9f	nicht während Datasetten-Betrieb
165-167/\$a5-\$a7	nicht während Datasetten-Betrieb
169-171/\$a9-\$ab	nicht während Datasetten-/RS232-Betrieb
176-182/\$b0-\$b6	nicht während Datasetten-/RS232-Betrieb
247-250/\$f7-\$fa	nicht während RS232-Betrieb
195-196/\$c3-\$c4	wird durch RUN/STOP-RESTORE beeinflusst
251-254/\$fb-\$fe	problemlos verwendbar
820-827/\$334-\$33b	nicht benutzt
828-1019/\$33c-\$3fb	Kassettenpuffer, nicht während Datasetten-Betrieb
1020-1023/\$3fc-\$3f	nicht benutzt

1024 (hex.\$0400) beginnt der Bildschirmspeicher, in den im Standardmodus alle Zeichen geschrieben werden. Äquivalent dazu existiert ab 55296 (hex. \$d800) der Farbspeicher. So lange beide Bereiche nicht für Bildschirmausgaben benutzt werden, kann man sie für eigene Programme hernehmen. Ein Beispiel hierfür sind

Basic-Compiler

Compiler übersetzen Hochsprachen in Maschinensprache und machen Programme um ein Vielfaches schneller als interpretierte Programme. Der bekannteste BASIC-Compiler ist der "Basic-Boss" und zugleich der schnellste (Test 8/94). Compiler für andere Hochsprachen (Pascal oder C) existieren ebenfalls für den C 64.

Literatur-Hinweise

Als wichtigste Quellen für "Schnüffler" im Betriebssystem des C 64 haben sich drei Bücher etabliert: *Alles über den C 64* - das Programmierhandbuch C 64 - *Tips, Tricks und Tools*, Florian Müller, C 64 für Insider, Florian Müller. Alle Bücher: Markt & Technik Verlag. Vor allem das letzte Werk beschäftigt sich ausgiebig mit dem Betriebssystem des C 64 und beinhaltet eine komplette Memory-Map.

stimmt wird. Der User kann sie als Pointer benutzen, solange keine Sprites im Spiel sind, sonst gibt's Grafikmüll bei den kleinen Kobolden.

Ab 2048 (hex.\$0800) beginnt der Basic-Speicher, der jedem An-

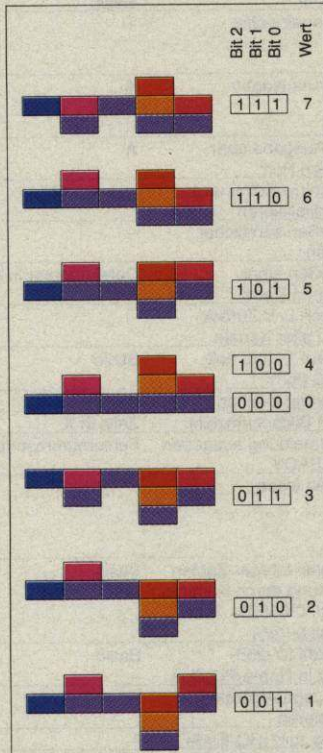


Bild 2: Das Bank-Management des C 64 per Speicherstelle 1

abgelegt werden. Hierzu müssen keine Veränderungen bei der Bank-Struktur (s.o.) vorgenommen werden. Alle anderen Bereiche des RAM müssen durch Umschalten per Speicherstelle 1 aktiviert werden. Näheres in Bild 2.

Der Basic-Interpreter

Dieser Teil des C 64 übersetzt (interpretiert) Basic-Programme in Maschinensprache, die der C 64 versteht. Jeder Befehl wird während des Programmablaufs übersetzt, was natürlich die Geschwindigkeit des Computers stark herabbremst. Wer seine Maschine schneller machen will, kann das Basic-Programm mit einem Compiled (s. Kasten) in ein lauffähiges Maschinenspracheprogramm übersetzen. Compilierte (übersetzte) Programme brauchen während der Arbeit des Computers nicht mehr mit dem Interpreter bearbei-

Wichtige Werkzeuge

Nach der Lektüre des Artikels wird sich so mancher Leser fragen, wie man am besten ans Betriebssystem des C 64 rankommt. Eine Wanderung durch den Speicher läßt sich am besten via Maschinensprache-monitor realisieren. Diese Tools sind meist Bestandteil eines Multifunktionsmoduls (z.B. Magic Formel) und verbrauchen keinen Speicherplatz. Wer nicht soviel Geld ausgeben will, kann auf einen Monitor auf Software-Basis zurückgreifen. Auf der Diskette zu diesem Heft finden sie den Monitor SMON. Er wird mit:

```
LOAD"SMON*",8,1
geladen und mit SYS9*4096 gestartet, der Prozessors-Status angezeigt und der Cursor erscheint.
Komfortabler lassen sich Programme aber mit einem Assembler schreiben. Auf der Diskette finden Sie den Assembler "VIS-Ass", mit dem sie Maschinensprache-Programme schreiben können.
```

Listing: Color-RAM ausschalten

```
10 REM FARBEN FUER BILDSCHIRM
UND ZEICHEN GLEICH
20 B=PEEK(53281): REM HINTER-
GRUNDFARBE ERMITTELN
30 FOR I=0 TO 999
40 : POKE 55296+I,B: REM IN
FARBSPEICHER SCHREIBEN
50 NEXT I
60 END
```

tet zu werden. Ein anderer Weg, die Bremse Basic zu umgehen, ist die Erstellung eine Programms komplett oder teilweise in Maschinensprache.

Der I/O-Bereich

Der Basic-Interpreter ist aber nicht nur für Basic-Freaks interessant. Assembler-Programmierer können sich diesen Teil des C 64 auch zunutze machen und in das ROM gebrannte Routinen nutzen. Mehr im Kasten "So werden Kern-Routinengenutzt"

Dieser Abschnitt des C-64-Speicher kontrolliert alle Ein- und Ausgaben (Input und Output) des Computer. Der I/O-Baustein teilt sich in fünf Abschnitte auf. Tabelle 3 erklärt die einzelnen Teile und Bild 3 zeigt die Anordnung. Genauere Ausführungen zur Programmierung dieser Bausteine würde den Rahmen des Artikels sprengen. Deshalb können wir nur auf einige Spezialartikel verweisen: Kasten "Literaturhinweise".

Das Betriebssystem

Der Kernal des C 64 wurde durch die Entwickler offen gestaltet und ermöglicht die Nutzung der vielen Subroutinen durch Programmierer. Was macht der Kernal eigentlich, wenn wir unseren Computer aktivieren?

Tabelle 3: Der I/O-Baustein

Abschnitt	Bedeutung
VIC - Video Interface Controlle	steuert die Grafik
SID - Sound Interface Device	steuert Sound und Musik
Bildschirmfarbspeicher	Festlegung der Farben passend zum Bildschirmspeicher (ab \$0400)
CIA 1 - Complex Interface Adapter	Steuerung und Ein- bzw. Ausgabe
CIA 2 - Complex Interface Adapter	Steuerung und Ein- bzw. Ausgabe

Tabelle 4: Wichtige Betriebssystem-Routinen im Überblick

Die an dieser Stelle beschriebenen Routinen sind nicht alle Funktionen, die der C 64 zu bieten hat. Die Tabelle kann nur ein Bruchteil davon aufzeigen. Weitere Funktionen finden Sie in einschlägiger Literatur (s. Kaste "Literaturhinweise").

Name	Adresse (hex/dez)	Funktion	Eingabe	Ausgabe	Vorbereitung	Meldung/Besonderheit
				Byte in A	TALK, TKSA	Fehler durch READST lesen
ACPTR	\$fa5/65445	Daten vom seriellen Bus lesen				
BLTUC	\$a3bf/41919	Verschiebt 256-Byte-Speicherblöcke	Block-Start: \$5f/\$60, Block ,ende+1: \$5a/5b Ziel-Ende+1: \$58/\$59			
CHKCLS	\$aef7/44791	Prüft auf "j"	Basic			bei Fehlen SYNTAX-ERROR
CHKCOM	\$aefd/44797	Prüft auf " "	Basic			bei Fehlen SYNTAX ERROR
CHKIN	\$ffc6/65478	Kanal für Eingabe öffnen	Nummer der logischen Datei in X			OPEN Fehler in A
CHKOUT	\$ffc9/65481	Kanal für Ausgabe öffnen	Nummer der logischen Datei in X			OPEN Fehler in A
CHKOPN	\$aefa/44794	Prüft auf "("	Basic		OPEN/CHKIN	bei Fehlen SYNTAX-ERROR
CHRIN	\$ffc/65487	Zeicheneingabe		A		Ist kein Kanal durch Vorbereitung definiert, wird Tastatur angenommen.
CHROUT	\$ffd2/65490	Zeichenausgabe	A	Bildschirm	OPEN/CHKOUT	Ist kein Kanal durch Vorbereitung definiert, wird der Bildschirm angenommen.
CIOUT	\$fa8/65448	Byte-Ausgabe über seriellen Port	A		LISTEN, SECOND	Fehler durch READST
CINT	\$f81/65409	Bildschirm-Editor und VIC initialisieren				
CLALL	\$fe7/65511	Schließen sämtlicher Dateien				
CLOSE	\$ffc3/65475	Schließen einer logischen Datei	Dateinummer in A			
CLRCHN	\$fcc/65484	Löschen und Zurückstellen aller Kanäle				
COMBYT	\$e200/5785	Prüft auf " " und holt Zahl (0-255)	Basic	X-Reg.		
CLEAR	\$e544/58692	Bildschirm löschen				
CLRLN	\$e9ff/59903	Löscht Bildschirmzeile	Zeile in X			Fehlernummern in Tabelle 5
ERROR	\$a437/42039	Fehlermeldung ausgeben und READY	Fehlernummer in X			
GETIN	\$ffe4/65508	Zeichen lesen		A	CHKIN, OPEN	Fehler über READST, ist kein Kanal durch Vorbereitung definiert, wird der Bildschirm angenommen.
GETNUM	\$b7eb/40939	Holt zwei Integer-Zahlen (Trennung durch Komma, 1. Zahl: 0-65535, 2. Zahl: 0-255)	Basic	1. Zahl: \$14/\$15, 2. Zahl: X		1. Zahl als 2-Byte-Int. 2. Zahl als 1-Byte-Int.
GTBYTC	\$b79b/47003	Holt Zahl (0-255)	Basic	X-Reg.		
HOME	\$e566/58726	Cursor in Home-Position				
IOINIT	\$ff84/65412	Ein-/Ausgabegeräte initialisieren				
IRQ	\$ea31/59953	Sprung zum IRQ-Ende				
LINPRT	\$bdcd/48589	Ausgabe einer 2-Byte-Integer-Zahl	X/Lo- und A/Hi-Byte	Bildschirm		
LISTEN	\$ff84/65412	LISTEN-Befehl für Gerät	am seriellen Bus Geräte Nummer (0-31) in A			Fehler über READST
LOAD	\$ffd5/65493	RAM laden von Peripherie	X/Y Startadresse, A=0 (Load)		SETLFS, SETNAM	Fehler über A und READST
LOADT	\$e168/57704	Load File	Parameter aus Basic-Text			Fehler über ST
MEMBOT	\$ffc/65436	Lesen/Schreiben des	X/Y	X/Y		vor Start ÜF gelöscht = Schreiben,
MEMTOP	\$ff99/65433	RAM-Anfangs Lesen/Schreiben des RAM-Endes	X/Y	X/Y		ÜF gesetzt = Lesen vor Start ÜF gelöscht = Schreiben, ÜF gesetzt = Lesen
OPEN	\$ffc0/65472	Öffnen einer logischen Datei			SETLFS, SETNAM	Fehler über A und READST
OVERR	\$b97e/47486	Ausgabe von OVERFLOW ERROR		Bildschirm		
PLOT	\$fff0/65520	Setzt/Liest Cursor-Position	X/Y	X/Y		vor Start ÜF gelöscht = Setzen, ÜF gesetzt = Lesen
PLOTK	\$e50a/5863	Setzt Cursor-Position in	X/Y			
PRT	\$e716/59158	Gibt Zeichen im Akku auf Bildschirm aus	Zeichen als ASCII in A	Bildschirm		
RAMTAS	\$ff87/65415	RAM-Test				
RDTIM	\$ffde/65502	Systemtaktgeber lesen		A=Time, X=Time+1, Y=Time+2		
READST	\$ffb7/65463	Statuswort ST lesen		A		Bedeutung in der Tabelle 5
READY	\$a474/42823	erzeugt READY-Status				READY
RESTOR	\$ff8a/65418	Normalzustand des Systems herstellen				
SAVE	\$ffd8/65496	Speichern von Daten	A=Start-Pointer, X/Y-Endadresse		SETLFS, SETNAM	Fehler in A oder über READST
SAVET	\$e156/57686	Save File	Parameter aus Basic-Text			
SCNKEY	\$ff9f/65439	Abfrage der Tastatur		A		evtl. Vorbereitung durch IOINIT

Fortsetzung Tabelle 4 von Seite 8

Name	Adresse (hex/dez)	Funktion	Eingabe	Ausgabe	Vorbereitung	Meldung/Besonderheit
SCREEN	\$fed/65517	Ermitteln des Bildschirmformats	-	X/Y	-	-
SCROLL	\$e8ea/59626	Scrollt Bildschirm um eine Zeile nach oben	-	-	-	-
SCRLIN	\$e9c8/59848	Scrollt eine Zeile nach oben	\$ac/\$ad Quelle BS, \$ae/4af Quelle FS, \$d1/\$d2 Ziel BS, \$f3/\$f4	Bildschirm	-	-
SECOND	\$ff93/65427	Sekundäradresse für LISTEN	Sekundäradressen in A	-	LISTEN	Fehler über READST
SETLFS	\$ffbba/65466	Einrichten einer logischen Datei	Dateinummer in A, Gerätenummer in X, Befehl in Y	-	-	Adressen für die Geräte sind in Tabelle 6 zu finden
SETMSG	\$ff90/65424	Ausgabe von Systemmeldungen	A (Nummer der Meldung 1-30)	Bildschirm	-	-
SETNAM	\$ffbd/65469	Dateiname festlegen	A-Namenslänge, X/Y Ort	-	-	-
SETTIM	\$ffdb/65499	Systemtaktgeber setzen	X,Y,A	-	-	-
STOP	\$ffe1/65505	Warten auf STOP-Taste	-	-	-	-
STROUT	\$ab1e/43806	Gibt String am Bildschirm aus (Ende 0)	A/Y Adresse für String	Bildschirm	-	-
SYNERR	\$af08/44808	Ausgabe von SYNTAX ERROR	-	Bildschirm	-	-
TKSA	\$ff96/65430	Sekundäradresse an Gerät	-	-	TALK	Fehler über READST
UDTIM	\$ffea/65514	Aktualisierung des Systemtaktgebers	-	-	-	-
UNTLK	\$ffba/65466	Beenden der Datenübertragung am seriellen Port	-	-	-	Fehler über READST
VERIFY	\$e165/57701	Verify File	Parameter aus Basic-Text	-	-	-

A = Akkumulator, X = X-Register, Y = Y-Register, Ü F = Übertrags-Flag, BS = Bildschirmspeicher, FS = Farbspeicher

Tabelle 5: Fehlerauswertung mit READST

Bedeutung der Systemvariablen ST:		
Drucker, Datasette, Floppy		
Bit	Bedeutung	
0	Fehler beim Schreiben Drucker/Floppy	
1	Fehler beim Lesen Drucker/Floppy	
2	Kurzer Block Datasette	
3	Langer Block Datasette	
4	unkorrigierbarer Lesefehler Datasette	
5	Prüfsummenfehler Datasette	
6	Ende der Datei (End of File)	Datasette/Floppy
7	Bandende/Gerät nicht verfügbar	Datasette/Floppy
RS232-Schnittstelle		
0	Fehler bei Paritäts-Test	
1	Fehler bei Bit-Folge	
2	Überlauf des Eingabepuffers	
3	leerer Eingabepuffer	
4	kein CTS-Signal (Handshake)	
5	unbenutzt	
6	kein DSR-Signal (Handshake)	
7	Übertragung unterbrochen	

1. Zurücksetzen des Stapelspeichers und löschen des Dezimalmodus (entspricht dem Maschinensprache-Befehl CLD)

2. Prüfen der Adresse \$8000 (dez. 32768) nach einem ROM (Cartridges o.ä.) mit Automatikstart. Ist diese vorhanden, wird die normale Initialisierung unterbrochen und die weitere Steuerung dem Code des ROMs übertragen. Wenn nicht, geht es wie in Punkt 3 beschrieben weiter.

3. Initialisierung der I/O-Bausteine durch den Kernall und des seriellen Busses. Die beiden Chips der CIA 6526 werden für die Tastaturabfrage auf die richtigen Werte gesetzt und der 60-Hz-Timer aktiviert. Alle Register des

Soundchips (SID) werden gelöscht. Wahl der richtigen Basic-Speicher-Konfiguration und Abschalten des Kassettenmotors.

4. Als nächstes werden ein RAM-Test durch den Kernall ausgeführt und die oberen und unteren Speicherzeiger gesetzt. Der Kassettenpuffer wird eingerichtet und die Zero-Page beschrieben.

Der RAM-Test ist eine nichtlöschende Routine, die bei \$0300

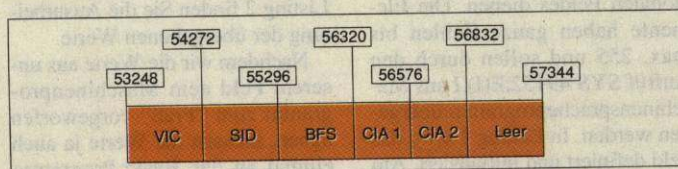


Bild 3: Die Aufteilung des I/O-Bausteins des C 64

Tabelle 6: Codes der Peripherie-Geräte

Adresse	Gerät
0	Tastatur
1	Datasette
2	RS232C
3	Bildschirm
4	serieller Drucker
5	Floppy

(dez.768) beginnt und in aufsteigender Folge arbeitet. Trifft die Routine auf die erste Nicht-RAM-Adresse, setzt sie die Speicher-Obergrenze. Der untere Zeiger wird stets auf \$0800 (dez. 2048) gesetzt und der Bildschirm auf \$0400 (dez.1024).

5. Ein- und Ausgabe-Vektoren werden auf Standardwerte verbogen, Bildschirm wird gelöscht und alle Bildschirm-Editor-Variablen zurückgesetzt. Dann springt der C 64 nach \$a000 (dez. 40960) indirekt und löst das Basic aus.

Der Kernall stellt eigentlich eine Sprungtabelle dar, die mit lauter JMP-Anweisungen gespickt ist. Das hat den Vorteil, daß bei Änderungen im Kernall nur diese Tabelle geändert werden und so der Programmierer nicht umlernen muß. Außerdem müssen Label in

Assembler-Quelltexten oder Variablen in Programmen nicht verändert werden. Der grundlegende Einsatz einer Kernall-Routine wird im Kasten "So werden Kernall-Routinen genutzt" beschrieben.

Neben den Kernall-Routinen, kann man, wie schon erwähnt, auch Teile des Basic-ROMs nutzen. Natürlich muß das Basic aktiviert sein und darf nicht über die Speicherstelle 1 ausgeblendet sein.

Wenn Sie jetzt ein wenig im Betriebssystem stöbern, wird sicher schnell klar, daß einige Routinen überflüssig und andere wieder nicht optimal sind. Besitzer einer Floppy können getrost die Teile des Kernalls vergessen, die für das Handling von Kassetten verantwortlich sind. Theoretisch könnte man sie entfernen und durch andere Programme nutzen. Dazu muß man aber ein neues ROM herstellen. Ein anderer Weg ist das Kopieren des ROM-Bereichs ins RAM darunter und Umschalten der Bänke. Die Routinen im RAM lassen sich jetzt patchen.

An dieser Stelle möchte ich unseren Ausflug in den Dschungel des C-64-Speichers beenden. Vielleicht helfen Ihnen die neu kennengelernten Routinen beim Nutzen Ihres C 64 weiter. Über Wünsche, Meinungen, Anregungen und Beiträge zu diesem Thema würden wir uns sehr freuen. Viel Spaß bei der nächsten ganz privaten Speichersafari - Ihr 64er-Team!

Jörn-Erik Burkert



Gleich vorab, nicht jede Betriebssystem-Routine ist zur Beschleunigung beim Programmieren mit Basic geeignet. Müssen erst zahlreiche Werte durch POKE-Befehle in den Speicher geschrieben werden, leidet die Geschwindigkeit und man sollte dann auf die Nutzung der Routinen verzichten.

Basic und Betriebssystem

Sie möchten ganz wieselflink Ihren Cursor setzen und dann Bildschirmausgaben realisieren? Kein Problem mit einer ganzen Latte an PRINT-Befehlen, die nicht gerade die schnellsten sind und dazu noch eine ganze Menge Speicher schlucken. Besser man nutzt die Kernal-Routine PLOT. Sie hat die Startadresse 65520 und verlangt die Koordinaten des Cursors im X- und Y-Register der CPU. Mit zwei POKEs und einem SYS-Befehl hat man den Cursor, wohin man ihn haben will. Ein Beispiel:

```
10 REM DEMO FÜR PLOT UNTER BASIC
20 X=5:Y=20:REM KOORDINTEN FÜR CURSOR
30 GOSUB 10000
40 REM IM PROGRAMM WEITER
50 END
100 REM PLOT-ROUTINE
110 POKE 781,X:REM X-REGISTER BESCHREIBEN
120 POKE 782,Y:REM Y-REGISTER BESCHREIBEN
130 SYS 65520
140 RETURN
```

Ebenso wie die Routine PLOT, lassen sich SCROLL, CLRLN und SCRLIN nutzen. Letztere Funktion ist eine flinke Möglichkeit, Zeilen auch in Basic auf dem Bildschirm zu verschieben. Wer seine Daten auf 40 Zeichen pro Eintrag formatiert, kann die Linie-Scroll-Funktion auch zum Umschaukeln von Datensätzen "mißbrauchen".

Von Basic zu Assembler

Die Arbeit mit der Maschinensprache bringt neben Geschwin-

Programmieren

Der große Sprung

Suchen Sie schon immer eine Möglichkeit, Ihren Basic-Programmen ein wenig Beine zu machen. Nehmen Sie die Unterstützung durchs Betriebssystem an und geben Sie auch in Basic Gas!

Listing 1: Das Basic-Grundprogramm

```
10 DIM E(10): REM DIMENSIONIERUNG
20 FOR I=0 to 9: REM LESEN DER DATEN
30 : READ E(I) : REM UND ZUWEISUNG
40 : : REM DER FELDELEMENTE
50 : SYS 49152,E(I),I
60 NEXT
70 END
80 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

digkeitsgewinn auch die Möglichkeit, in verschiedene Prozesse (z.B. Interrupts) des C 64 einzugreifen. Wer langsam, aber sicher umsteigen will, sollte zu Beginn noch das Rahmenprogramm in Basic und kleinere Hilfsroutinen in Assembler schreiben. Um Werte aus einem Basic-Programm auf Maschinenebene zu übernehmen, gibt es mehrere Möglichkeiten:

1. Man POKet die Werte in einen freien Speicherbereich.

Listing 3: Variablen aus dem Speicher holen

```
10 DIM E(10): REM FELD DEFINIEREN
20 FOR I=1 to 10
30 : SYS 49152,I: REM WERT ÜBERGEBEN
40 : E(I)=AA: REM ELEMENT SETZEN
50 NEXT
```

2. Man übergibt die Werte als Parameter innerhalb des SYS-Befehls.

Punkt 1 kann unter Umständen sehr langsam sein und sollte möglichst vermieden werden. Die zweite Variante ist wesentlich eleganter. Deshalb wollen wir diesen Weg einmal genauer untersuchen.

Back to Basic

Als Grundlage soll die Übergabe von Elementen eines eindimensionalen Feldes dienen. Die Elemente haben ganze Zahlen bis max. 255 und sollen durch den Aufruf SYS 49152,E(I),I ans Maschinenspracheprogramm übergeben werden. In Listing 1 wird das Feld definiert und initialisiert. Am Ende findet die Übergabe statt. In

um könnte man die Werte aus dem Speicher mit dem PEEK-Befehl lesen, was nicht besonders schnell ist. Wir wollen aber einer Variablen im Speicher einen Wert zuweisen und in einer Schleife den jeweils aktuellen Wert an das aktuelle Feldelement übergeben. Dazu werden wir uns im Maschinenprogramm eine Hilfsvariable anlegen. In Listing 3 sehen Sie das Basic-Rahmenprogramm und in Listing 4 die Assemblerroutine. Bei der Auswertung von I im Maschinenprogramm muß die Routine für Integerzahlen genommen werden, da I als Variable nicht als Byte definiert ist. Die Übernahme des Lo-Bytes ins X-Register zum Auslesen der Tabelle reicht vollkommen aus, da wir uns ja auf Zahlen bis 255 beschränkt hatten. Mit einem Maschinensprache-Monitor können Sie den Aufbau der Variable AA im Speicher untersuchen. Wer Strings oder Integer verarbeiten möchte, sollte sich selbige in Basic definieren und

mit dem Monitor untersuchen. Es dürfte kein größeres Problem sein, das Wissen auf die beiden Variablentypen zu übertragen. Sie sehen, daß die Kommunikation zwischen Basic und Assembler nicht das größte Problem ist. Das dargestellte Beispiel ist sicher noch nicht die optimale Lösung. Es wäre auch denkbar, eine Tabelle aus Variablen ab 49152 (\$c000) anzulegen und sie durch ein Maschinenprogramm zu verwalten. So wird wertvoller Basic-

Listing 2: Die Maschinen-Routine zu Listing 1

```
*=$c000 ; programmstart
jsr $aefd ;pruefe auf komma
jsr $b7eb;hole 2-byte
;integer und 1-
;byte-integer
lda $14 ;e(i)
sta puffer,x ;schreibe wert in
;Tabelle
rts
puffer: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Listing 4: Die Definition einer Variable zu Listing 3

```
*=$c000
sta $7a ;neue Variable
lda #>text ;setzen
sta $7b
jsr $a9a5 ;LET-Routine
pla
sta $7b ;Vektor zurück-
pla ;schreiben nach
sta $7a ;$7a/$7b
rts
text: .byte $41,$41,$b2,$0,$0
```

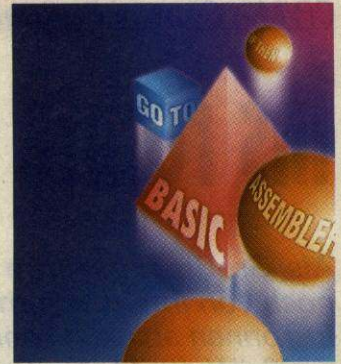
Listing 2 finden Sie die Ausarbeitung der übergebenen Werte.

Nachdem wir die Werte aus unserem Feld dem Maschinenprogramm zum "Fraß" vorgeworfen haben, müssen die Werte ja auch einmal an das Basic-Programm zurückgegeben werden. Wieder-

Speicher gespart. Mit einigem Erfindergeist lassen sich Programme optimieren und der Speicherplatz des C 64 ausnutzen. Wenn Sie unsere kleinen Anregungen aufgreifen und ausbauen, verstehen Sie auch Ihre Programme demnächst mit mehr Pep und Geschwindigkeit. Jörn-Erik Burkert

Betriebssystem-Entwicklung

Mehrsprachig auf vielen Plattformen



Heutige Betriebssysteme (BS) haben mit denen von vor 20 oder 15 Jahren nichts mehr gemein. CP/M (Control Program for Microprocessors, entwickelt 1974) auf einem Z80-Prozessor sind Museumsstücke, die heute kaum mehr einer kennt. Heute wird bereits bei und während der Hardware-Entwicklung entschieden, was und welche Teilbereiche in Software zu realisieren sind. Hard- und Software-Entwickler sind viel enger zusammengedrückt, und das Schlagwort vom Hard- und Software-Co-Design ist bei vielen modernen Elektronik-Entwicklern bereits längst Realität (Bild 1). Wie aber geht sowas vorstatten?

Das BS besteht immer aus einem Kern und den dazugehörigen Dienstprogrammen. Teile dieser BS werden in einem System fest eingebunden (z.B. BIOS = Basic Input Output System in einem ROM, Systemeinstellungen oder mathematische Routinen in EPROM, etc.), ein weiterer Teil wird in einer sogenannten Maschinensprache geschrieben. Sie dient der Verständigung der Hardwarekomponenten (Speicherkarte, Ein- Ausgabeschnittstellen, Disketten- und Festplattenlaufwerke, Monitore, Maus etc.) sowohl mit der Zentraleinheit (CPU = Central Processor Unit) als auch untereinander. Wiederum sind Teile dieser Maschinensprache unveränderbar, andere lassen sich je nach Hardwarekomponenten konfigurieren. Bestes Beispiel dafür ist das von den meisten DOS-Rechnern her bekannte CMOS-Setup, ein Menü, in dem der Anwender z.B. Datum und Zeit festlegen, die Festplatte konfigurieren, den an-

Jedes elektronische Gerät mit einem Mikroprozessor benötigt ein Betriebssystem, um seine Aufgaben zu verrichten. Unser kleiner Exkurs zeigt Betriebssysteme einmal aus ganz anderen Blickwinkeln.

geschlossenen Bildschirmtyp oder die Speicherverwaltung bestimmen kann. Dazu werden Teilbereiche des BS aus dem Festwertspeicher (ROM) gelesen und in den Arbeitsspeicher geladen, wo ihre Variablen dann verändert werden können. Interessant dabei ist, daß jede Hardwarekomponente die Befehle der Maschinensprache erst einmal verstehen muß, denn jede periphere Hardware hat normalerweise ihre eigene Sprache, wurde also von den entsprechenden Entwicklern auf deren Art und Weise programmiert. Ohne Normierungen und Standards würde das zu einem babylonischem Sprachgewirr führen, weshalb Richtlinien und Interpretationen (Compiler) hier sehr hilfreich sind. Doch nun zurück zu den neuen BS, die als sogenannte Loadware auf die Geräte, die sie später einmal »betreiben« sollen, abgestimmt sind. Am anschaulichsten werden derartige BS an Handys oder modernen Display-Telefonen. Bleiben wir zunächst einmal beim Handy, hier ist GSM (Global System for Mobile Communications) das international genormte BS, das in seiner Struktur dem bekannten BS Unix gleicht (Bild 2). In einer umfangreichen Dokumentation stellt GSM si-

das bereits angesprochene Hard- und Software-Co-Design, das bereits bei der Spezifikation eines neuen Geräts beginnt. Je nach Bauelement verteilen die Systementwickler die Design-Arbeiten in Software für den statischen Teil des BS und den flexiblen Bereich,

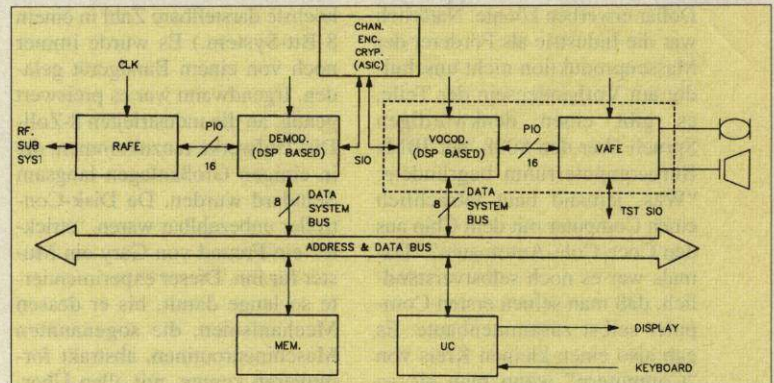


Bild 2: Blockschaubild einer GSM-Mobilstation, das zeigt, welche Hardwarekomponenten mit Software versorgt werden müssen

cher, daß die Hersteller von Mobilfunktelefonen die Normen einhalten, wollen sie mit ihren Geräten auf dem internationalen Markt zugelassen werden. Gleichermaßen aber ist GSM so offen, daß die Funknetzbetreiber, hier zunächst die der D1- und D2-Netze, vielfältige Dienste wie Mailboxen, Anrufbeantworter, Notizbuchfunktionen etc. miteinbinden können. Zunächst aber regelt GSM das Zusammenspiel der einzelnen Hardwarekomponenten wie Sender und Empfänger, Display, Tastatur, Karten-Handling (D1- oder D2-Karte), Schnittstellen, Stromversorgung und Anschluß externer Peripherie wie z.B. Mikrofon und Lautsprecher als Freisprecheinrichtung in einem Kraftfahrzeug. Viele Bereiche dieser BS-Software verrichten ihre Arbeit, ohne daß der Anwender etwas davon merkt, andere wiederum benötigen die zwingende aktive Mitarbeit des Besitzers und rufen z.B. zur Eingabe für die Identifikation (PIN-Code, Diebstahlsicherungscode) auf. Gemeinsam für diese Bereiche gilt

das ist zum Beispiel die landessprachliche Konfiguration der Bedieneroberfläche.

Deshalb ist es auch möglich, so ein Handy z.B. nach Schweden oder Italien mitzunehmen und es auch dort in der gewohnten Weise in Betrieb zu nehmen. Auf dem Display erscheint automatisch die Kennung des jeweiligen Netzbetreibers.

Der hat nämlich aufgrund der offenen Spezifikation des BS genau die Software-Schnittstelle konfiguriert, die nun die schwedische oder italienische Kennung auf dem Display erscheinen läßt. Umgekehrt ist es natürlich genauso, italienische oder schwedische Handys funktionieren bei uns ebenso, wie die eigenen Geräte, die z.B. vom selben Gerätehersteller gefertigt und geliefert werden. Was also macht der Hersteller? Er versieht seine Systeme mit den entsprechenden BS erst kurz vor der Auslieferung, Teilbereiche werden sogar erst bei der Erstinbetriebnahme (automatisch) nachgeladen, bzw. freigeschaltet.

Claus Baumann/lb

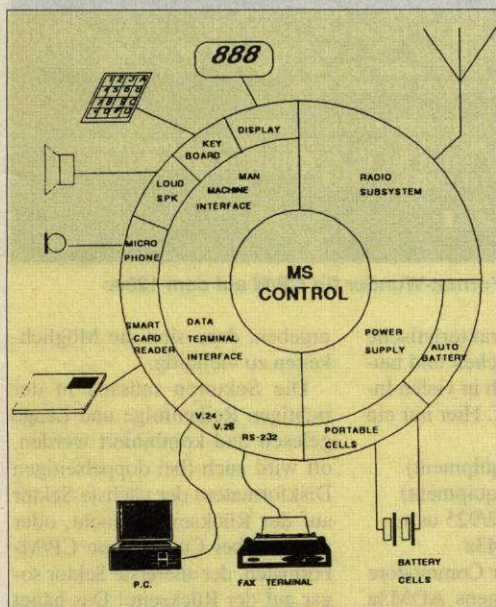
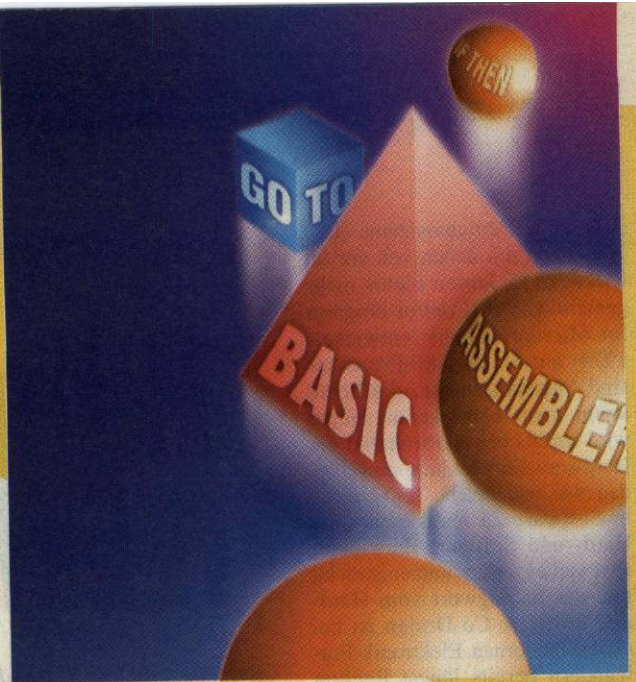


Bild 1: Funktionale Gliederung einer Mobilstation

CP/M – Control Program for Microcomputers

Ins rechte Licht gerückt

Als Großrechner ihre Namen hauptsächlich noch von ihrem Volumen und Gesamtgewicht in Tonnen ableiten konnten, war das (Schwarzweiß-)Weltbild der Computerwelt noch vollkommen in Ordnung – aber heute ...



Computer waren etwas Teures und nur Firmen konnten sich Erwerb und Service leisten. Wer konnte auch schon erwarten, daß man bald ein kleines Lötstellenmonster für schlappe 1000 Dollar erwerben könnte. Natürlich war die Industrie als Förderer der Massenproduktion nicht unschuldig am Vorhandensein der Teile, es gibt einen denkwürdigen Spruch über den 8088, der IBMs Bürocomputerruhm begründete: "Was, jemand baut tatsächlich einen Computer mit dem Chip aus den Coca-Cola-Automaten?" Damals war es noch selbstverständlich, daß man seinen ersten Computer selbst zusammenbaute. Es gab also einen kleinen Kreis von "Computern" wenn man sie so nennen will, die mit einem Minimum an Speicher (meist etwa 2 KByte) nur Maschinencode verstanden. Dafür reichte es meist, oft gab es noch ein Bandgerät zum Speichern/Laden der Daten.

Die Anfänge

Irgendwann begannen einige Leute, den Unterschied zu einem praktisch einsetzbaren Computer zu bemerken. Irgendwie konnte man die Daten nicht richtig verwalten und schon gar nicht weitergeben. Also beschlossen verschiedene Leute, den Datenzugriff zu steuern und zu normieren. Der erfolgreichste war damals Gary Kildall. Er benutzte wiederholbare Funktionen, indem er sie festschrieb und begann, sie in der Reihenfolge der Notwendigkeit zu nummerieren. Er begann, seine Sammlung als eine Art System zu sehen und nannte sie demzufolge "system calls – Systemaufrufe".

Natürlich wurde immer mehr Speicher notwendig, um alle Systemaufrufe im Speicher zu halten, bis man schließlich auf eine gigantische Größe von 26 KByte (!) kam. Da man aber noch einen Arbeitsspeicher brauchte, um

überhaupt Programme ausführen zu können, mußte erweitert werden. Die natürliche Grenze von 64 KByte erwies sich als geeignet. (64 KByte entsprechen 65 535 Byte, also FFFF HEX. Das ist die höchste darstellbare Zahl in einem 8-Bit-System.) Es wurde immer noch von einem Bandgerät geladen. Irgendwann war es preiswert genug, an die industriellen 8-Zoll-Disklaufwerke ranzukommen, die in einigen Großanlagen langsam Standard wurden. Da Disk-Controller unbezahlbar waren, "strickte" ein Freund von Gary ein Muster für ihn. Dieser experimentierte so lange damit, bis er dessen Mechanismen, die sogenannten Maschinenroutinen, abstrakt formulieren konnte, mit allen Übergabeparametern und den zu erwartenden Ergebnissen. Jetzt war es ihm möglich, seine Liste der Funktionen um die Ein-/Ausgaberroutinen der Disk-Controller zu erweitern. Jetzt mußte er sein System dazu überreden, auf das Diskettenlaufwerk zu wandern und es anschließend von diesem zu starten. Da er dazu irgendwo ja ein Programm abrufen und ablaufen lassen mußte, entschied er sich, einen kleinen Teil in ein EPROM zu brennen, von dem aus die Operationen durchgeführt werden konnten.

Anschließend ergänzte er das EPROM um einen sogenannten Urlader, das "Bootstrap-Programm", das den ersten Teil seines Systems in den Speicher lud und weitere Operationen möglich machte. Er erfand die berühmte Dreiteilung, die auch heute noch Leute wie Bill Gates erschauern macht. Ein maschinenabhängiges Segment, das auf jedem Rechner anders aussieht, das sogenannte BIOS (Basic Input/Output System), das BDOS (Basic Disk Input/Output System) und als Krönung die Kommandozentrale, der CCP (Console Command Processor). Die Bezeichnung "Basic"

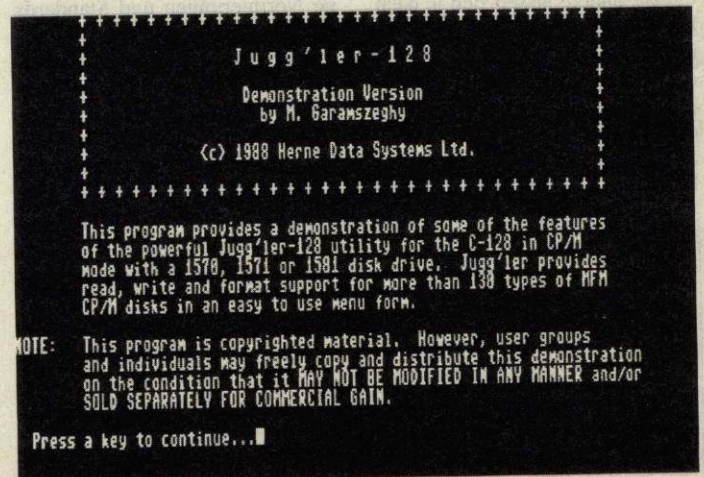
steht für grundlegend, nicht zu verwechseln mit BASIC (Beginner's All-Purpose Symbolic Instruction Code), das ja eine Abkürzung darstellt.

Der Endspurt

Da er damit erstmals einen "Microcomputer" wirklich kontrollieren konnte, nannte er sein System CP/M, Control Program for Microcomputers. Was die meisten heute nicht mehr wissen, ist, daß die Computer außer etwas klobigen Abmessungen auch noch ein separates Terminal besaßen, eine Einheit aus Bildschirm und Tastatur, die über die serielle Schnittstelle die Ein- und Ausgabe von Daten ermöglichte. Alle diese Ter-

Theorie und Praxis

Grundsätzlich könnte man also sagen, daß jede Standardsoftware auf jedem CP/M-Rechner laufen müßte. Theoretisch. Es gibt aber zig Hindernisse, denn irgendwie mußte die Software ja auf den Rechner kommen. Das erste war das Diskettenformat. Dieses unterteilt sich wiederum in das physikalische und das logische Diskettenformat. Wenn das physikalische Diskettenformat stimmt, paßt erstens die Diskette von den Abmessungen ins Laufwerk, aber zweitens kann auch jede Spur der Diskette mit einem Disk-Editor "angefahren" und gelesen werden. Einen Sinn müssen diese Daten jedoch nicht sofort zwangsläufig



Juggler, das Multi-Format-Wunder für CP/M auf dem 128er

minals hatten charakteristische Bildschirmsteuerzeichen und tauchen auch heute noch in vielen Installationsmenüs auf. Hier nur ein paar Beispiele:

- VT52 (Digital Equipment)
 - VT100 (Digital Equipment)
 - Televideo 910/912/925 usw.
 - Lear/Siegler ADM3a
- und viele andere. Der Commodore C128 emuliert übrigens ADM3a und Televideo Terminals.

ergeben, dazu sind die Möglichkeiten zu vielfältig.

Die Sektoren müssen in der richtigen Reihenfolge und Länge gelesen und kombiniert werden, oft wird auch (bei doppelseitigen Diskformaten) der nächste Sektor auf der Rückseite gesucht, oder gar, wie bei Commodore CP/M-Formaten, der allererste Sektor sogar auf der Rückseite! Das hängt im letzten Fall mit "Sparsamkeit"

zusammen, als man sich ein Inverter-Glied schenkte bei der Konstruktion der Commodore-Laufwerke. Dadurch bedingt werden die Kopfseiten-Bits, die "Head-IDs" invers ausgewertet, also 0 wird 1 und umgekehrt. Derartige Dinge haben in der Vergangenheit oft wirksam den ursprünglich geplanten Austausch mit anderen Rechnern verhindert.

Berüchtigtes Beispiel ist das Format KAYPRO 4, das der C128 problemlos lesen kann. Aber: erzeugt man das KAYPRO-4-Format auf einem C128, wird aus den ursprünglichen Kopf-IDs 0/0 der Wert 0/1! Resultat: weder ein echter KAYPRO 4 noch ein anderer CP/M-Computer kann dieses Format erkennen. Wenn man's weiß, ist es oft einfach mit einer kleinen Änderung getan. So geschehen in der Definition für das Formatprogramm 22DISK unter DOS. Seit einem kleinen "Eingriff" kann man damit jetzt auf DOS-Rechnern jederzeit das KAYPRO-4/10-(KAYPRO-IV-)Format des C128 lesen, schreiben und formatieren.

Der Start

Wenn also unser Bootstrap-Loader die Instruktionen auf der Diskette ausgeführt hat und das BIOS, das BDOS und der CCP geladen sind, kommt der damals sensationelle, heute eher ernüchternde Augenblick, da auf dem Bildschirm ein A> auftaucht. So mancher hat hier schon begeistert immer wieder auf die ENTER-Taste gehämmert, ohne Erfolg. Das Ergebnis war und ist A>. Damit teilt uns nämlich das Betriebssystem einiges mit.

Erstens, daß es geladen ist; zweitens, daß kein Fehler vorliegt; und drittens, daß es sich um Laufwerk A: handelt, wo am sogenannten "Prompt" (hier das Größer-Zeichen) Befehle eingetippt werden können. Jedes Kommando wird mit der ENTER-(bzw. RETURN-)Taste abgeschickt und gelangt an den CCP, den Console Command Processor. Dieser prüft, ob ein interner Befehl gemeint sein könnte, wie DIR, ERA und dergleichen, andernfalls sucht er auf dem aktuellen Laufwerk (und im Userbereich) nach einem externen Kommando mit dem angegebenen Namen.

Feinheiten

Grundsätzlich muß man davon ausgehen, daß das ursprüngliche CP/M 1.14 quasi nicht mehr existent ist, da es zu viele Unzulänglichkeiten

enthielt. Der Nachfolger, das CP/M 2.2, ist dagegen der häufigste Vertreter des CP/M-80. Grundsätzlich läuft es auf allen 8080, Z80 und kompatiblen Prozessoren, wodurch es sich auch für Emulatoren auf anderen Betriebssystemen eignet. Auch verwandte Prozessoren wie 8045 und andere Exoten können dieses CP/M "fahren". Aufgrund dessen, daß bei einem Diskwechsel aus Zeitgründen nicht alle Verzeichniseinträge geprüft werden, setzt das CP/M 2.2 eine gewechselte Diskette erst einmal auf READ ONLY (Nur-Lesen). Beim ersten Versuch, auf diese Diskette zu schreiben, wird man mit dem Ärgernis Nummer eins des CP/M-80 (CP/M 2.2) konfrontiert: BDOS ERR ON A: SELECT ...

Hinter dieser kryptischen Botschaft versteckt sich die klare Mitteilung des CP/M, daß ein grundsätzlicher Fehler der Disk Ein-/Ausgabe vorliegt. Es wurde ein Laufwerk selektiert, das eine schreibgeschützte Diskette enthielt und verbotenerweise versucht, darauf zu schreiben. Dabei wäre alles so einfach gewesen. Man braucht nur auf jeder Diskette eine Kopie des Betriebssystems (in den Systemspuren) und muß nach dem Wechsel CTRL-C drücken, also Control-Taste und C gleichzeitig. Nun wird das BDOS zurückgesetzt, das Laufwerk wird angemeldet und das BDOS, sowie der CCP werden neu geladen. Jetzt ist Schreiben ohne Ärger möglich.

Es gibt Public Domain Tools, wie etwa NSWEEP, die eine Diskette "einloggen" können, ohne auf die Kommando-Ebene zu gehen. Damit braucht man natürlich auch nicht ein CP/M auf jeder Diskette und kann leichter wechseln, vorausgesetzt, man verläßt das NSWEEP nicht vorzeitig.

Hier setzten in der Vergangenheit viele Programmierer an, um das CP/M kompatibel zu verbessern. Z80DOS, SUPERDOS25, P2DOS und NOVADOS waren die ersten "Rewrites" des BDOS, in denen Dinge wie automatischer Login, Datum/Uhrzeit-Stempel, Suchpfad und standardisierte Systemerweiterungen verwirklicht wurden.

Diese Unzulänglichkeiten waren natürlich auch den Entwicklern bei Digital Research bekannt, und so wurden ins gerade entstehende, neue, CP/M einige dieser Features dauerhaft implementiert. In der Tat wurden außer dem automatischen Disk-Login so viele Neuerungen eingebaut, daß man gleich ein Zeichen setzte mit einer Versionsnummer 3.xx, zu-

nächst mal 3.0, das man wegen des großen Leistungsumfangs CP/M Plus nannte.

Digital Research im Zugzwang

Was war neu? Mal abgesehen von AMSTRAD und ein paar anderen Lizenznehmern, war der CCP.COM von CP/M Plus ein separates Programm (ähnlich wie der COMMAND.COM von DOS) ohne die Größenbeschränkung einer Systemspur mit vielen, erweiterten Befehlen. Es gab natürlich ein eingebautes DIR-Kommando, aber es war auch möglich, ein externes DIR.COM-Kommando zu laden, dessen Möglichkeiten sehr viel mächtiger waren. Es ist vielleicht nicht jedem klar, welche immense Neuerung das war, daß ein gleichlautendes Kommando einmal intern und einmal extern gesucht wurde, aber das schafft noch nicht einmal der Standard-COMMAND.COM von MS-DOS!

Aber nicht nur der CCP war neuer und größer geworden. Ein komplettes, neues BDOS wurde geschrieben, das viele, interessante Neuerungen enthielt. So hatte man z. B. Zugriff auf die exakte Anzahl der Bytes einer Datei (über den File Control Block FCB), statt wie bisher nur auf Einheiten von 128 Byte Länge (128

Byte = 1 Record). Es gab jetzt eine Art Cache-Funktion, die mit internen Puffern arbeitete. Bestes Beispiel ist das "Merken" des Verzeichniseintrages einer Diskette, wenn man zum Beispiel mehrmals hintereinander DIR aufruft, erfolgt nur beim ersten Mal ein wirklicher Diskettenzugriff. Jeder C128-Besitzer kennt den Geschwindigkeitsvorteil bestimmt. Damit aber nicht genug. Datum und Uhrzeit waren bei CP/M Plus eine der wichtigsten Neuerungen. Hierbei konnte man zwei von drei Möglichkeiten für seine Dateien wählen: Erzeugungsdatum (CREATE), Veränderungsdatum (MODIFY) oder Zugriffsdatum (ACCESS). Vom Verzeichnis jeder Diskette wurde ein Viertel der Eintragsplätze geopfert, um für diese Verwaltung Platz zu machen. Wird der Diskettenplatz benötigt, kann man nachträglich die Datums- und Uhrzeit-Stempel entfernen und so wieder mehr Einträge verwalten.

Genial ist auch die Leistung, das leidige Problem der Adressierung der Betriebssystem-Geräte, der DEVICES, durch vollständige Einbindung als logische Devices zu überwinden. Nun konnte man erstmals Programme schreiben, die auf Standard-Geräte zugreifen, das bekannteste darunter ist (nach der CONSOLE) das AUX-Device, also die serielle Schnittstelle. Die Implementierung ist so, wie man es auch von DOS her kennt, welches Zufall: wird beim Start eine serielle Schnittstelle gefunden, wird sie mit Standardwerten initialisiert, wenn nicht, wird einfach weitergemacht. Dadurch konnte man unabhängig von der physikalischen Ausstattung dasselbe CP/M Plus nehmen. Ausnahmen bilden hier nur maschinenspezifische "Abnormalitäten".

Neu war auch, daß man mit einem externen Kommando DEVICE.COM die Parameter und Zuweisungen der Devices steuern konnte, anders als bei CP/M 2.2, wo man nur über physikalische Adressen zugreifen konnte. Es gab sogar schon sogenannte I/O-Redirection, also "Ein-/Ausgabenumlenkung", mit deren Hilfe man die CONSOLE-Aktion in ein Device (zum Beispiel eine Datei) oder aus einem Device (zum Beispiel einer SUBMIT-Stackeldatei) umlenken konnte.

Die Syntax war recht einfach:

PUT CON(sole) FILE datei.out schrieb alles, was auf dem Bildschirm erschien, auch eine Datei namens DATEI.OUT. Mit PUT CON(sole) CON(sole) wurde die Umleitung beendet. GET CON

Bezugsquellen für CP/M-Software

Helmut Jungkunz, Zacherlstr. 14, 85737 Ismaning, neue Tel.-Nr. 089/96146 33 (24 Stunden am Tag Bestellung möglich, Achtung! Anrufbeantworter!)
Mailbox CP/M Support 18:00 h bis 6:00 h 300 - 14400 bps
089/9614575. Keine Gebühren.
Wer nicht selbst DFÜ macht, kann ja einen Freund fragen.
Katalogdiskette KAYPRO-IV-Format (oder 2x KAYPRO II) gegen 5 Mark in Briefmarken und Leerdiskette. Rücketikett und Rückporto bitte beilegen! Es können nur MFM-CP/M-Formate bearbeitet werden, kein 1570/1571 GCR!
Komplette CP/M P.D.-Software aller Vereine, Z-P.D. für Z80 M&T Restposten günstig auf Anfrage und NZ-COM bzw. Z3PLUS Autoinstaller für ZCP/R (CP/M Betriebssystemaufsatz, 70 Mark), ZMAC Assembler, und vieles mehr.
Juggler-128 (170 Formate, 50 Mark), darunter auch PCW 3.5" (1581 für 3.5" Formate notwendig).
Auch gebrauchte Software und Bücher werden vermittelt.
Fa. Wiedmann Unternehmensberatung, Korbinianplatz 2/6, 85737 Ismaning, Tel. 089/96 50 29
M&T Software dBASE II, Multiplan, WordStar. CP/M Plus Ersatzdisketten für C128 und AMSTRAD. Kann auch im 1570- bzw 1571-Format liefern.

FILE haurein.txt simulierte aus HAUREIN.TXT eine Tastatureingabe. Das Ganze kennt man in UNIX mit der Syntax:

[Aktion] >datei.out bzw. [Zutun] < haurein.txt

Hierbei ist die Tatsache interessant, daß auch die feste Zuordnung des temporären Laufwerks für Zwischendateien entfällt und man mit SET TEMPORARY=M die RAM-Disk als "Müllhalde" deklarieren kann. Außerdem kann eine Suchreihenfolge (erst .SUB, dann .COM) bestimmt werden, CP/M Plus kennt auch einen Suchpfad mit bis zu vier Laufwerken, in dessen Reihenfolge die bezeichneten Laufwerke nach einem Befehl abgesucht werden. Ein "*" kennzeichnet dabei das aktuelle Laufwerk, das DEFAULT DRIVE. Einziger Nachteil immer noch: quer über Userbereiche konnte immer noch nicht gesucht werden. Ein kleiner Trick half hier aber das Schlimmste zu verschmerzen. Alle Dateien auf A: User 0, deren Attribut mit SET auf SYSTEM und READONLY gesetzt worden war, waren aus anderen Userbereichen aufrufbar und wurden gefunden. Die Möglichkeit des Paßwortschutzes war

ebenfalls eine entscheidende Neuerung, wenngleich sie nie die große Bedeutung erlangte, die sich DRI erhoffte, nämlich bei kommerzieller Nutzung. Dazu kam CP/M Plus einfach viel zu spät. Ein paar kleine Macken waren auch noch drin und so besserte man schnell nach mit Version 3.1, die man zum Teil selber generieren konnte. Dazu gab es ein standardisiertes PATCH.COM-Programm, das darüber hinaus auch noch die Möglichkeit bot, Systemerweiterungen, sogenannte RSXe, automatisch zu laden.

Einige wenige Programme, wie das Multi-Format-Wunder "Juggler-128" nutzen diese Technik für sich aus, um den CCP zu zwingen, beim Systemstart bestimmte Aktionen vorzunehmen. Per Default ist eine solche Möglichkeit aber auch schon "eingebaut": die Datei PROFILE.SUB wird beim Booten gelesen und dem Stapelprozessor SUBMIT.COM zur Verarbeitung gegeben. Damit hatte man eine Möglichkeit geschaffen, wie beim AUTOEXEC.BAT des DOS, automatisch komplexe Operationen (mit I/O-Redirection!) ablaufen zu lassen. Inzwischen gibt es ein tolles

Tool aus der PUBLIC DOMAIN, das in keinem CP/M-Plus "Waffenschränk" fehlen sollte: CHN31.COM. So kann man eine CP/M-Plus-Befehlszeile in eine .COM-Datei umwandeln und spart sich a) die Datei SUBMIT.COM und kann b) die DISKETTE schreibgeschützt lassen, da keine temporäre Datei angelegt wird. Eine CP/M-Plus-Kommandozeile kann übrigens maximal 128 Byte betragen und kennt mehrere Befehle hintereinander, separiert durch ein (!) Ausrufezeichen.

Beispiel:

```
CHN31 START
>DATE SET!PIP
M:=S*.*[vo]!PIP
M:=HELP*.*[vo]!M:!HELP PIP
START.COM written
```

Alles hinter dem Eingabepfeil ist eine gültige CP/M Plus Kommandozeile! Das [vo] unmittelbar hinter der PIP-Anweisung stellt sicher, daß die Datei vollständig kopiert wird, auch wenn es eine Objekt-Datei ist (Ein binäres Programm, eventuell mit CTRL-Z als Inhalt), mit VERIFY. Trifft PIP nämlich beim Kopieren auf ein CTRL-Z, so denkt es "Aha, CTRL-Z, also Dateiende" und hört auf. Böser Fehler! Also, am

besten immer angeben, schaden tut es nie! Man kann ja auch ein COPY.COM erzeugen:

```
CHN31 COPY
>PIP $2=$1[vo]
COPY.COM written
```

Jetzt kann man mit COPY S*.* M: problemlos à la DOS kopieren. Die Platzhalter "\$" werden in der entsprechenden Reihenfolge des Auftretens beziffert. Dadurch kann das zweite und erste Aufrufargument vertauscht werden. Das [vo] sollte hier allerdings seinen festen Platz intern haben.

Es ist natürlich nicht möglich, eine umfassende Einführung in einem einzigen Artikel unterzubringen, deswegen soll diese Reihe ja auch fortgesetzt werden. Aber vielleicht hat ja der eine oder andere plötzlich Interesse an der Sache gefunden und möchte sich etwas mehr damit befassen. Dann wäre ja der Zweck schon erreicht.

Ach ja, was aus Gary Kildall geworden ist? Gary (der erst kürzlich verstorben ist) lebte lange zurückgezogen auf einer Farm in Texas mit seiner Frau und hatte dort eine Firma mit kleinen Projekten laufen.

Helmut Jungkuntz / ma

SORRY, WERBUNG GESPERRT!

64ER ONLINE



WWW.64ER-ONLINE.DE



CS-DOS für den C 128

Das Multitalent

Das "CS-DOS" hat nicht grundlos einen Namen, der an Microsoftsche Vorbilder erinnert. Was Sie bei diesem Multitalent erwartet und wie Sie seiner Herr werden, haben wir kurz & knapp zusammengefaßt.

```

text          $0b00-0b19
time         $0b00-0b1f
type        $0b00-0b5f
val         $0b00-0b09
vdel       $0b00-0c29
vdump     $0b00-0b1f
changeunit $0b00-2115
rcv       $3000-3763
#         (batch)
cd        (batch)
nd        (batch)
cls       (type)
colors   (type)
esc       (type)
arc       $1c01-2c3a
lha       $1c01-2e08
lhd       $1c01-4e0c
lhx       $1c01-520e

75,182 bytes used in 58 File(s). 186,962
bytes free.
(RAM disk in 1750 banks 0-3.)
ready.
    
```

CS-DOS initialisiert beim Booten eine RAM-Disk und speichert wichtige Befehle darin

Das Grundprinzip ist mittlerweile in der ganzen Computer-Welt verbreitet: Ein Programm (hier die "SHELL") überwacht Ihre Eingaben und führt ganz nach Wunsch bestimmte Aktionen aus. Zunächst prüft CS-DOS, ob an erster Stelle ein "*" eingegeben wurde – in diesem Falle werden alle Ausgaben bis zum nächsten READY an den Drucker geschickt. Falls CS-DOS am Anfang der Eingabezeile jedoch ein > oder >> findet, ist das ein Signal dafür, daß die Ausgabe auf Diskette umgeleitet werden soll. Last but not least muß CS-DOS auch noch auf "Schalter" testen, die durch "/" gekennzeichnet sind.

Wie es dann weitergeht, hängt davon ab, ob Sie der Zeile einen Laufwerksbuchstaben vorangestellt haben (Laufwerk 8 ist A:, Laufwerk 9 C:, das Boot-Laufwerk kann mit Z: angesprochen werden). Wenn ja, wird auf Diskette ein entsprechendes Programm gesucht, wenn nein, sucht CS-DOS nach einem Kommando, und wenn alles fehlschlägt, wird die Eingabe an den Basic-Interpreter weitergegeben.

Installation

Starten Sie im 128er-Modus die "SHELL". Das Programm sucht dann auf dem Bootlaufwerk eine sequentielle Datei "Autoexec", die aus CS-DOS-Kommandos im Klartext besteht. Sinnvoll ist es z.B., eine RAM-Disk anzulegen, Laufwerke zu konfigurieren, häufig benötigte Kommandos im Ram zu installieren. "Autoexec" können Sie mit jedem Programm editieren, das sequentielle Dateien bearbeiten und ohne Steuerzeichen speichern kann. Natürlich empfiehlt sich für den C 128 der Klassiker "ZED".

Die mitgelieferte Autoexec installiert automatisch bereits die wichtigsten Befehle in einer 16 KByte großen RAM-Disk (die allerdings nicht wie ein normales Diskettenlaufwerk angesprochen werden kann). Weitere Kommandos können Sie mit "INSTALL" hinzufügen, bis die 16 KByte voll sind. Platz schaffen können Sie dort wieder mit "REMOVE", z.B. REMOVE ECHO.

Ein Tip: Wenn Sie Ihre RAM-Disk so umgestellt haben, wie Sie es sich vorstellen, sollten Sie den Inhalt komplett unter dem Titel "Meinram" auf Diskette sichern: PUTRAM MEINRAM

Beim nächsten Mal sparen Sie sich dann einige Arbeitsschritte, indem Sie GETRAM MEINRAM

eingeben und damit alles wieder in den gewünschten Zustand versetzen. Der Getram-Befehl kann natürlich auch gleich in die Autoexec eingefügt werden.

Übrigens: Wenn Ihnen die korrekte Syntax eines Befehls entfallen ist, legen Sie einfach die Diskette mit den DOC-Files ein und lassen sich die entsprechende Hilfedatei mit TYPE (Filename) ausgeben.

Batch mal wieder

Eines der mächtigsten Features von CS-DOS ist die (von MS-DOS übernommene) Fähigkeit, ganze Befehlsfolgen ("Batches") hintereinander abzarbeiten. Ein Batch-File benötigt keine spezielle Dateinamens-Endung, es muß nur sequentiell aufgebaut sein. Sobald Sie den Namen der Datei eingeben, werden alle darin stehenden Kommandos schrittweise abgearbeitet, und zwar solange, bis die Batchdatei

- a) zu Ende ist
- b) auf den Befehl POKE 7162,0 trifft (weiter geht's mit POKE 7162,255)
- c) eine andere Batchdatei aufgerufen wird (nachfolgende Befehle werden ignoriert!)
- d) der User CBM und CTRL gleichzeitig drückt

Zusätzlich gibt's Sprung- und Schleifenbefehle, die eine richtige Programmierung ermöglichen. BEGIN ... LOOP stellt etwa eine grundlegende Schleifenkonstruktion dar (die allerdings nur funk-

tioniert, wenn die Batch-Datei aus der RAM-Disk aufgerufen wird). IF hingegen ermöglicht Verzweigungen, und GOTO springt zu einem (mit Doppelpunkt gekennzeichneten) Label. Den MS-DOS-Befehl ECHO OFF (Inhalt der Batch-Datei soll nicht auf dem Bildschirm erscheinen) können Sie durch POKE 7147,255 simulieren (durch POKE 7147,0 rückgängig zu machen). Was Batch-Dateien erst wirklich flexibel macht, ist, daß Sie beim Aufruf auch Parameter übergeben können! Wenn Die Batchdatei "Meinbatch" z.B. die Zeile ECHO %0 %1 %2 %3 enthält, und Ihre Eingabe lautet MEINBATCH A B C D E F G erhalten Sie MEINBATCH A B C auf dem Bildschirm. Der mit "%0" übergebene Parameter enthält also stets den Namen des Batchfiles selbst, und die anderen (maximal neun) entsprechend die folgenden Parameter.

Packen und Entpacken

Die bei CS-DOS mitgelieferten Packer waren bereits Thema eines 64er-Artikels. Es kann nur nochmals unterstrichen werden: Es gibt zumindest für den C 128 nichts Gleichwertiges. Um mit CS-LHARC zu packen, benötigen Sie allerdings eine Commodore-RAM-Erweiterung! Das Resultat ist entsprechend professionell: gute Packrate, Entpacken der ebenfalls zu erzeugenden selbstextrahieren-

den Archive auch im 64er-Modus und auf Laufwerke 8/9 möglich, "Begrüßungstext", der beim Entpacken gezeigt wird usw.

Noch mehr Nützliches

Es gibt eine Vielzahl weiterer Befehle, deren komplette Erläuterung ein ganzes Heft füllen würde.

Deshalb wollen wir hier nur die wichtigsten Kommandos in Kürze erläutern:

ASSIGN: ändert die Zuordnung der Laufwerke

CHANGE: ändert Texte oder Basic-Programme im Speicher

CHECK: stark erweitertes Directory-Kommando, es wird aber die komplette Diskette blockweise überprüft

CHANGEUNIT: ändert Laufwerksadressen softwaremäßig

COMP: vergleicht zwei Files

COPY: kopiert Dateien zwischen zwei Laufwerken

DEL: löscht eine Datei

DIR: Directory

EXEC: führt sequentielle Datei aus

FIND: durchsucht Texte bzw. Basic-Programme nach Strings

GETKEY: wartet auf Tastendruck

GETRAM: lädt RAM-Disk

GOTO: Batch-Sprungbefehl

INSTALL: lädt Befehle in die RAM-Disk

MERGE: verbindet Basic-Programme

OLD: restauriert Basic-Programm

PUT: speichert RAM-Inhalt

PUTRAM: speichert RAM-Disk

RDIR: listet Befehle der RAM-Disk

RDOFF: RAM-Disk wird gelöscht

REMOVE: entfernt Kommandos aus der RAM-Disk

RSAVE: kopiert Programm aus RAM-Disk auf Floppy

SETRAM: hält Speicherplatz für RAM-Disk frei

TYPE: zeigt Dateiinhalte

VAL: verwandelt numerischen Ausdruck in Zahl

Wir hoffen, daß wir Ihnen einen kleinen Überblick verschaffen konnten. CS-DOS ist tatsächlich fast ein Muß für jeden C-128-Besitzer. Übrigens gibt's die deutsche Anleitung inzwischen auch im Geos-Format als Shareware bei jedem Sharewarehändler und ausgedruckt für 19,80 Mark bei Independent Softworks. CS-DOS selbst ist nach wie vor Shareware, allerdings verzichtet der Inhaber der Rechte, Parsec Inc., momentan auf Gebühren. Die in der zu CS-DOS gelieferten Anleitung genannte Adresse ist nicht mehr gültig!
Matthias Matting

Gedruckte Anleitung inkl. Disk:
Independent Softworks, 79199 Burg am Wald
CS-DOS als Telesoftware in Btx:
*732#*922502#

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

0	"	"	USR	5	"SIR-HUFPACK"	PRG	0	"	"	USR	
0	"	"	USR	13	"SIR-INSTALL"	PRG	0	"	"	USR	Seite 43
0	"	1541-DOS	USR	14	"SIR-COMPACT III"	PRG	0	"	"	USR	
0	"	"	USR	10	"BITSTREAMER 1.01"	PRG	0	"	"	USR	
172	"	1541-DOS.DOC"	PRG	0	"	USR	0	"	"	USR	
81	"	1541-DOS.EXE"	PRG	0	"	USR	40	"	"	USR	
9	"	1541-DOS.REA"	PRG	0	"	USR	40	"	"	USR	
110	"	1541-DOS.SFX"	PRG	0	"	USR	40	"	"	USR	
0	"	"	USR	13	"TIER RATEN.KURS3"	PRG	38	"	"	USR	
0	"	"	USR				79	BLOCKS FREE.			
0	"	PACKER-PACK	USR								

DISKETTE SEITE 2

0	"	"	DEL<	7	"DEMO.BOMBS"	PRG	96	"IMP.SOURCE #4"	SEQ	
0	"	NOTEPAD FUER	DEL<	0	"	DEL<	0	"	DEL<	Seite 40
0	"	VISASS	DEL<	0	"	DEL<	0	"	DEL<	
0	"	"	DEL<	0	"	DEL<	0	"	DEL<	
9	"	INSTALL.NB"	PRG	0	"	DEL<	98	"ROBOX"	PRG	
67	"	VIS-ASS+NB-DEMO"	PRG	3	"VISMON"	PRG	0	"	DEL<	
0	"	"	DEL<	15	"VIS-SMON \$9000"	PRG	0	"	DEL<	
0	"	CONVERTER FUER	DEL<	59	"VIS-ASS V6"	PRG	18	"N<OITTL<	USR	
0	"	BASIC-ASS	DEL<	2	"VISMON CONVERTER"	PRG	20	"S<AIL<	USR	
0	"	"	DEL<	20	"NOTE"	PRG	28	"N<OITTL< DR<	USR	
6	"	BA-CONVERTER"	PRG	0	"	DEL<	15	"S<ILT<	USR	
41	"	BASIC-ASSEMBLER"	PRG	0	"	DEL<	27	"ITPCX"	USR	
9	"	DEMO.ALLGEMEIN"	PRG	0	"	DEL<	14	"ITPCX T<	USR	
4	"	DEMO.PRINT"	PRG	0	"	DEL<	52	BLOCKS FREE.		
3	"	DEMO.ON X GOSUB"	PRG	42	"IMPEROID V4"	PRG				
3	"	DEMO.DATA"	PRG							

WICHTIGE HINWEISE zur beiliegenden Diskette:

1

Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.

2

Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatzproblemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit dem File-Copy-Programm auf eine leere formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.

3

Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.

ALLE PROGRAMME aus diesem Heft



HIER

Kleinanzeigen-Auftrag für den **COMPUTER-MARKT**

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe von 64'er den folgenden Kleinanzeigentext unter der Rubrik
(z.B. C 64, C 128, VC 20, Software, Zubehör, Verschiedenes).

Meine Anzeige ist eine **private Kleinanzeige** (5 Zeilen mit je 32 Buchstaben, maximal 160 Zeichen)
 DM 5,- liegen bar als Scheck bei. **Bitte keine Briefmarken!**
Bezahlung über Postscheckkonto nicht mehr möglich.

Meine Anzeige ist eine **gewerbliche Kleinanzeige für DM 12,-** (zzgl.MwSt.) je Druckzeile

Anschrift:

64'er
Computermarkt
MagnaMedia Verlag AG
Postfach 1304
85531 Haar

Absender:

Name/Vorname

Straße

PLZ/Ort

Telefon

Bei Angeboten: Ich bestätige, daß ich alle Rechte
 an den angebotenen Sachen besitze

Datum:

Unterschrift:



Impressum

Chefredakteur: Georg Klinge (gk)
Stellv. Chefredakteur: Harald Beiler (bl)
 verantwortlich für den redaktionellen Teil
Chef vom Dienst: Uschi Anders (ua)
Textchef: Jens Maasberg
Redaktion: Jörn-Erik Burkert (lb), Matthias Matting (ma)
Redaktionsassistentz: Helga Dietz

So erreichen Sie die Redaktion:
 Tel. 089/4613-202, Fax 089/4613-5001, Btx *640644#

Manuskripteinsendungen: Manuskripte und Programm listings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der MagnaMedia Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programm listings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von MagnaMedia Verlag AG verlegten Publikationen und dazu, daß die MagnaMedia Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und verteilt und durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Uschi Böcker, Erich Schulze, Dagmar Beminger
DTP-Operator: Dorothea Voss
Titellayout: Wolfgang Berns
Computergrafik: Alexander Gerhardt
Fotografie: Roland Müller

So erreichen Sie die Anzeigenabteilung
 Tel. 089/4613-962, Telefax 089/4613-394

Anzeigenleitung: Peter Kusterer
Anzeigenverwaltung und Disposition: Anja Böhl (233)
Anzeigenpreise: Es gilt die Preisliste Nr. 2 vom 01. 01. 1994
Erscheinungsweise: monatlich (zwei Ausgaben im Jahr)
Leitung Herstellung: Klaus Buck (180)

Technik: Sycam Druckvorstufen GmbH, Hans-Pinsel-Str. 2, 85540 Haar

Druck: Druckerei E. Schwend GmbH & Co. KG, Schmollerstr. 31, 74523 Schwäbisch Hall

Urheberrecht: Alle im 64'er erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen und Zweitverwertung, vorbehalten. Reproduktionen, gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebene Lösung oder verwendete Bezeichnung frei von gewerblichen Schutzrechten sind.

Haftung: Für den Fall, daß im 64'er unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.

Vertriebsleitung: Benno Gaab

Abonnement-Service:

64'er Aboservice
 74168 Neckarsulm, Tel.: 07132/959-242,
 Fax: 07132/959-244
Einzelheft: DM 9,80
Jahresabonnement Inland
 (12 Ausgaben): DM 99,-
 (inkl. MwSt., Versand und Zustellgebühr)
Jahresabonnement Ausland: DM 123,-
 (Luftpost auf Anfrage)
Österreich: DSB-Aboservice GmbH,
 Arenbergstr. 33, A-5020 Salzburg, Tel.: 0662/643866,
 Jahresabonnementpreis: öS 816
Schweiz: Aboverwaltungs AG, Sägestr. 14,
 CH-5600 Lenzburg, Tel.: 064/519131,
 Jahresabonnementpreis: sfr. 99,-
Nachbestellung Einzelhefte:
 64'er Leserservice
 Heiner-Fleischmann-Str. 2, 74172 Neckarsulm
 Tel. 07132/969-181 – Fax 07132/969-190
 (Heftpreis + DM 6,- Versandpauschale)

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge können für Werbezwecke in Form von Sonderdrucken hergestellt werden. Anfragen an Thomas Lux, Tel. 089/4613-5039, Telefax: 089/4613-5041

Anzeigen-Auslandsvertretungen:

USA: M&T International Marketing, Telefon: 001-415-358-9500
 Fax: 001-415-358-9739
Großbritannien: Smyth International, London, Tel. 0044-83140-5058, Fax 0044-81-341-9602
Frankreich: Adresse International, Neuilly, Tel. 0033-1-46378717, Fax 0033-1-46371946
Taiwan: Acer TWP Co., Taipei, Tel. 008862-713-6959,
 Fax 008862-715-1950
Japan: Media Sales Japan, Tokyo, Tel. 0081-33504-1925,
 Fax 0081-33595-1709
Italien: Media International, Mariano, Tel. 0039-31-751494,
 Fax 0039-31-751482
Holland: Insight Media, Laren, Tel. 0031-2153-12042,
 Fax 0031-2153-10572
Israel: Baruch Schaefer, Holon, Tel. 00972-3-556-2256,
 Fax 00972-3-556-6944
Korea: Young Media Inc, Seoul, Tel. 00822-765-4819,
 Fax 00822-757-5789
Hongkong: The Third Wave (H.K.) Ltd., Tel 00952-7640989,
 Fax 00852-7643857

1994 MagnaMedia Verlag Aktiengesellschaft

Vorstand: Carl-Franz von Quad (Vors.), Kenneth Clifford

Verlagsleiter: Wolfram Höfler

Produktionschef: Michael Koeppel

Direktor Zeitschriften: Michael M. Pauly

Anschrift des Verlages:

MagnaMedia Verlag Aktiengesellschaft,
 Postfach 1304, 85531 Haar bei München,
 Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Diese Zeitschrift ist auf chlorfreiem Papier mit einem Altpapieranteil von 30% gedruckt. Die Druckfarben sind schwermetallfrei.

Schlagwörter zum Nachschlagen!

Folge 4

Weiter geht's mit unserem Computer-Lexikon. Es soll Sie in vielen Fortsetzungen nicht nur allein über die Welt der Commodore-Computer C64/C128, sondern auch bewußt über andere Geräte und deren Systeme informieren (MS-DOS, Windows, RISC OS usw.). Denn: mitreden kann nur, wer Bescheid weiß!

G

Garbage Collection: Routine im Basic-Interpreter, die quasi als Müllabfuhr den Variablenspeicher eines Programms nach Werten und Zeichenketten durchsucht, die man nicht mehr benötigt und diesen Platz "besenrein" wieder freigibt.

Gateway: spezielle Verbindungsrechner, die mehrere Netze zu einem Verbund koppeln (Voraussetzung: geeignete Schnittstellen).

Gatter: (Gate). Elektronisches Schaltglied, das eine Boolesche Funktion realisiert. Gatter haben normalerweise einen bzw. zwei Eingänge und nur einen Ausgang. Die dort anliegende Spannung ist eine Funktion der Eingangsspannung(en). Von den 16 zweistelligen Boole'schen Funktionen sind in der Elektronik nur fünf relevant: AND, OR, NAND, NOR und XOR. Als einstellige Funktion existiert NOT (logisches NICHT).

GDT: (Global Descriptor Table). Tabelle, in der die einzelnen

Speichersegmente im "Protected Mode" des Prozessors (z.B. beim AT 486) enthalten sind.

Gerätetreiber: (Device Driver, Interrupt Handler). Spezielles Programm zur Nutzung bestimmter Peripherie-Geräte (z.B. Drucker). Gerätetreiber ergänzen das jeweilige DOS um zusätzliche Eigenschaften zur komfortablen Verwaltung solcher Geräte.

Gerätename: Logische Bezeichnung, mit der man DOS anweist, auf dieses Gerät zuzugreifen. Bei den C 64/C 128 sind es stets Nummern (z.B. "4" oder "5" für Drucker, "8" bis "11" für Laufwerke), bei MS-DOS normalerweise Buchstaben (Drive A bis E, CON für Tastatur/Bildschirm, PRN für Drucker usw.).

Gibson-Mix: Mix-Tests sind Verfahren, um die Leistungsfähigkeit eines Computers zu bewerten. Man stellt eine Mischung von Einzeloperationen zusammen. Dieser Mix wird dann an die jeweilige Maschinsprache des Rechners angepaßt, aktiviert und die Bearbeitungszeit gemessen.

Der Gibson-Mix enthält repräsentative mathematisch-technische Problemstellungen.

Gleitpunktzahlen: (Floating Point, Fließkommazahlen). Bei der Gleitpunktarithmetik wird das Dezimalkomma vom Computer automatisch berücksichtigt (im Gegensatz zur sonst üblichen Festpunkt-Arithmetik = Integerzahlen). Damit kann man mit Zahlenbereichen rechnen, die weit über die Möglichkeiten von Integerzahlen hinausgehen (die dürfen nämlich nicht höher sein als die höchste, vom Computer adressierbare Speicherstelle).

Die Formel:

$M \times B^E$

M ist die Mantisse, B die Basis 10 und E der Exponent.

Die Mantisse enthält den eigentlichen Zahlenwert (z.B. 0.45689) und den Dezimalpunkt (steht normalerweise vor der ersten signifikanten Ziffer).

Als interne Basis sind 2, 10 und 16 üblich. Die Kombination aus Basic und Exponent gibt Auskunft über die Größe der Zahl (= tatsächliche Position des Dezimalpunkts). Beispiel:
0.45689E2

... entspricht dem reellen Wert "45.689". Die meisten Basic-Interpreter arbeiten mit der bereits im System integrierten Basis 10 (bei der Bildschirmausgabe durch den Buchstaben E gekennzeichnet).

Grafikauflösung: Der C 64 arbeitet mit der Standardauflösung 320 x 200 Bildpunkte, der 16-KByte-VDC des C 128 bewältigt in der Normalkonfiguration 640 x 200 Pixel (höhere Auflösungen sind beim 64-KByte-VDC auf Grund des zusätzlichen RAM möglich, z.B. 640 x 720, 720 x 600 usw.).

Der PC/AT schafft z.B. Grafikauflösungen von 1024 x 768 oder 1280 x 1024 Pixeln (abhängig vom verwendeten Monitor und Grafikkartentyp).

Grafikkarte: (Graphics Board). Erweiterungsplatine, mit der man Mikrocomputer (vorwiegend PCs) in die Lage versetzt, Bildschirme

Video-Controller-Bausteine (VIC und VDC), die sich über bestimmte Register direkt programmieren lassen.

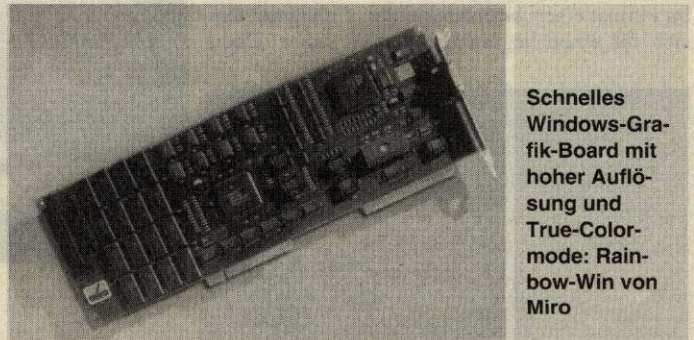
Grafikmodus: Betriebsart des Video-Chips, in der sich alle Pixel des Bildschirms adressieren lassen. Nicht nur Hires-Bilder, sondern auch Textpassagen werden als Hires-Grafik interpretiert. Geos 2.0 z.B. arbeitet ausschließlich im Grafikmodus, ebenso Windows 3.1 beim PC/AT oder DTP-Programme mit den entsprechenden Computern.

Grafikprozessor: Spezialspeicherbaustein, der im Computersystem in Verbindung mit dem Bildwiederholungspeicher (Video-RAM) Grafiken erzeugt oder verändert. Meist besitzen Grafikprozessoren Intelligenz und können grafische Grundelemente selbst kreieren – den Startbefehl gibt natürlich die CPU (z.B. sind beim Amiga "Copper" und "Blitter" Grafikprozessoren, die von der 68xxx-CPU quasi eine Checkliste bekommen, die dann abgearbeitet wird).

Der VIC- bzw. VDC-Chip der C 64/C 128 sind allerdings "ahnungslos" Prozessoren, die ohne entsprechende CPU-Befehle zum Einschalten oder Löschen von Bildpunkten (in Basic oder Assembler) keine Grafik abliefern können.

Grafiktablett: elektronisches Zeichenbrett, das Zeichenbewegungen per Hand (Linien, Kreise usw.) in digitale, vom Computer lesbare Daten umwandelt.

Das Gerät merkt sich entweder die Bewegungen eines elektronisch angeschlossenen Stifts, der über die Unterlage geführt wird oder registriert die Koordinaten eines kontaktempfindlichen, elektrischen Rasters der Tablettoberfläche. Zur Weiterverarbeitung der



Schnelles Windows-Grafik-Board mit hoher Auflösung und True-Color-mode: Rainbow-Win von Miro

mit höherer Auflösung oder noch mehr Farben zu betreiben. Manche Grafikkarten sind intelligent (besitzen also einen eigenen Prozessor).

Bei den C 64/C 128 ist der Einbau solcher Grafikkarten nicht vorgesehen: Sie haben separate

Werte ist allerdings noch ergänzende Software nötig (z.B. Mal- und Zeichenprogramme). Bekanntestes Beispiel aus der C 64-Szene: "Koala-Pad", auf dem man freihändig zeichnen konnte (wird heute nicht mehr hergestellt oder vertrieben). Die Daten wurden

vom Malprogramm "Koala Painter" weiterverarbeitet.

GRP: Dateinamen-Erweiterung (Suffix) für Gruppdateien des Programm-Managers von Microsoft Windows.

H

Halbbyte: (Nibble). Besteht aus 4 Bit. Hat ein Byte z.B. den Dezimalwert 162, enthält es als binäres Bitmuster

10100010

und läßt sich in die Halbbytes 1010 und 0010 zerlegen (hexadezimal: \$A0 und 02). Das gesamte Bitmuster lautet dann hexadezimal: \$A2.

Halbduplex: Bei der seriellen Datenübertragung (z.B. DFÜ) werden Daten zum empfangenden Computer gesendet und nur dort angezeigt (im Gegensatz zu "voll-duplex", wo die Bytes vom Empfänger zur Kontrolle nochmals zurückkommen). Diese Methode verwendet man nur bei Übertragungsmodi, bei denen Überprüfung unwichtig ist.

Halbleiter: Material, dessen Leitfähigkeit zwischen der von Metall und eines Isolators liegt. Durch absichtliche Verunreinigung (Dotierung) des Trägermaterials (monokristalline Germanium- und Silizium-Scheiben, Spezialausdruck: wafers) mit bestimmten chemischen Elementen (z.B. Arsen, Indium) entstehen negativ und positiv leitende Grundmaterialien, aus denen sich die in der Elektronik verwendeten Dioden, Transistoren und Bauelemente in integrierten Schaltungen zusammensetzen.

Handheld-Computer: extrem flacher Personal Computer, etwa im Format einer Aktentasche, der sich für einen begrenzten Zeit-



Noch winziger, dafür aber nur begrenztes Sichtfeld: Notepads

raum netzunabhängig betreiben läßt. Besitzt anstatt eines externen Monitors einen integrierten LCD-Screen und läßt sich mit diverser Peripherie verbinden (Printer, Laufwerke, externer Bildschirm etc.). Typische Vertreter der Handhelds sind **Laptops** und **Notebooks** der PC/AT-Szene, noch kleiner und handlicher sind **Note-pads**. Der Begriff Handheld wird heute kaum noch benutzt.

Handle: numerischer Wert, der als Schlüssel für den Zugriff auf Dateien oder Geräte dient. Er wird vom DOS an ein Programm übergeben, das Funktionen zum Öffnen oder Erzeugen einer Datei aktiviert bzw. ein anderes Gerät aufruft.

Hardcopy: Bezeichnung für die Druckausgabe des Inhalts eines Text- oder Grafikbildschirms, die man per Tastendruck einleitet. Im Gegensatz zu anderen Computern (z.B. PCs) kennt der C 64 in seiner Normalkonfiguration diese Funktion nicht; allerdings existieren für ihn viele Betriebssystem-Erweiterungen, spezielle Hardcopy-Module oder -Routinen in Maschinensprache.

Harddisk: (Festplatte). Massenspeicher für Direktzugriff, das aus einer harten (nicht flexiblen) Kunststoff-scheibe mit magnetisierbarer Ober-

fläche besteht. Die Harddisk befindet sich im Winchester-Laufwerk. Beim Betrieb dreht sich die Platte mit hoher Geschwindigkeit und wird abgetastet. Da es auf hohe Positionierungsgenauigkeit des Schreib-/Lesekopfs ankommt, müssen Festplatten exakt zentriert sein.

Hardware-Interrupt: Signal, mit dem ein Gerät eine Dienstleistung oder Unterstützung vom Computer anfordert.

Hauptplatine: (Motherboard). Grundausrüstung, die zum Betrieb eines Rechners notwendig ist: CPU, Arbeitsspeicher, Adreß- und Datenbus sowie Schnittstellen für Peripherie. Bei manchen Computertypen läßt sich z.B. der Arbeitsspeicher (RAM) durch zusätzliche Speicher-Chips erhöhen, andere Rechner brauchen separate Erweiterungsplatinen.

Head-Crash: Die Oberfläche einer Diskette oder Festplatte wird durch den Schreib-/Lesekopf (Head) des Laufwerks magnetisiert (Schreiben) oder ein magnetisierter Zustand festgestellt (Lesen). Dabei steht der Kopf nicht in direktem Kontakt zur Diskette, sondern schwebt in geringem Abstand auf einem Luftkissen über der Scheibe (ca. 0,001 mm Zwischenraum). Berührt der Schreib-/Lesekopf aber die Disk, wird der Datenträger zerstört: dann entsteht bewußter "Head Crash".

Heap-Sort: Sortierverfahren, das 1964 von J. Williams entwickelt wurde. Damit wird sim-

Hexadezimalsystem: Zahlensystem mit 16 Ziffern – zunächst von 0 bis 9, die letzten sechs werden durch die Buchstaben A bis F repräsentiert. Diese Art der Zahlendarstellung eignet sich ideal zur Anzeige der 16 x 16 = 256 maximalen Werte, die in eine Speicherstelle passen.

Hidden Files: Dateien, deren Namen nicht im Directory auftauchen. Beim PC/AT sind das z.B. die Systemfiles im Hauptverzeichnis: IBMBIO.COM, bzw. IO.SYS, IBMDOS.SYS oder MSDOS.SYS. Sie besitzen das spezielle Dateiattribut "hidden" (verborgen).

High-Density-Disk: Disketten mit hoher Speicherkapazität (z.B. 1,44 MByte bei 3,5-Zoll- und 1,2 MByte bei 5,25-Zoll-Scheiben).

Die Laufwerke des C 64/C 128 (1541, 1571, 1581) eignen sich nicht für solche qualitativ hochwertigen Diskettentypen.

Hilfe-Datei: Textfile (überwiegend mit der Endung .HLP), das nützliche Erläuterungen zu Befehlen, Modi und Infos enthält.

Hi-RAM: RAM-Speicher des AT (ab 286), der hinter der 1-MByte-Grenze beginnt und sich nur innerhalb des "Protected Mode" ansprechen läßt.

Hires: (High Resolution) – Alle Bildpunkte (Pixel) eines hochauflösenden Bildschirms (beim C 64: 320 x 200 = 64 000 Bildpunkte) lassen sich einzeln ansprechen, allerdings nur mit zwei Farben (Hintergrund- und Zeichenfarbe).

Hochformat: Standardformat bei der Druckausgabe (z.B. Brief): Um das Blatt zu lesen, hält man es hochkant. Beim breiteren Querformat erscheint die Druckausgabe seitlich gekippt.

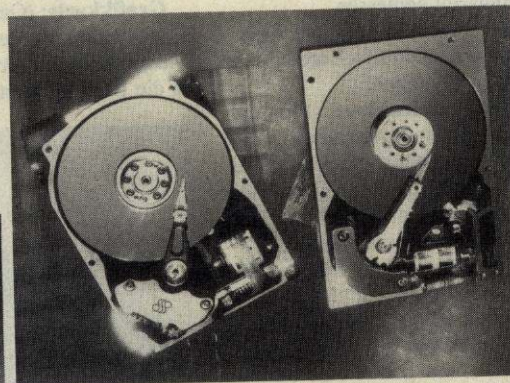
Host-Rechner: Zentralcomputer eines Netzwerks. Er enthält das primäre Netzwerk-Betriebssystem und die Gerätetreiber. Außerdem kontrolliert der "Host" (Wirt) den korrekten Systemstart der Workstation, die Kommunikationsparameter für die übrigen Rechner im Netz, sowie die Gate-

way-Kommunikation mit anderen Netzwerken.

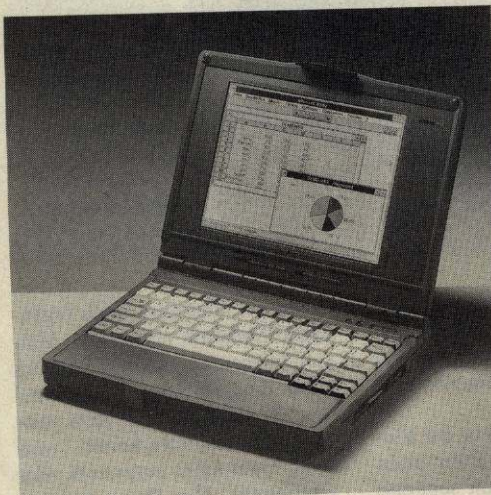
Hot Key: Tastenkombination, die einem residenten Programm das Signal gibt, seine Arbeit zu beginnen.

HLP: Namenserverweiterung der Hilfe-Dateien (z.B. bei CP/M, MS-DOS, Windows), um Infos an den User weiterzugeben.

(wird fortgesetzt)



Zwei 3,5-Zoll-Festplatten-Laufwerke von Seagate (Kapazität: 260 und 340 MByte)



Notebooks – typische Vertreter der Handheld-Computer

ple Auswahlortierung optimiert, indem zu sortierende Datenelemente schrittweise in eine Haufenstruktur (= Heap) übertragen werden. Man entfernt das oberste (größte) Element vom Heap und überträgt es in die neue Reihung. Das zuvor zweitgrößte Element rückt nun nach und kommt als nächstes in die Haufenstruktur, bis der Heap leer ist.

way-Kommunikation mit anderen Netzwerken.

Hot Key: Tastenkombination, die einem residenten Programm das Signal gibt, seine Arbeit zu beginnen.

HLP: Namenserverweiterung der Hilfe-Dateien (z.B. bei CP/M, MS-DOS, Windows), um Infos an den User weiterzugeben.

(wird fortgesetzt)



Copper im C 64

Frage von Philipp Kaiser in der 64'er 6/94: Ich möchte einen Co-Prozessor in meinen C 64 einbauen. Was muß ich dabei beachten?

Zunächst brauchen Sie einige Hardware-Kenntnisse zum C 64. Dann nehmen Sie einen Prozessor gleichen Typs (CPU 6510) und bringen ihn auf einer Platine an – genauso wie ein 512-Byte-RAM, das den Stack, die x- und y-Register sowie den Akkumulator und einige Pseudo-Register enthält, mit denen man "mechanisch" zwischen Intern und Extern (DMA high, DMA low) umschaltet.

Um die Software-Kompatibilität steht's aber schlecht: Man braucht ein weiteres Pseudo-Register, um wieder zurückzuschalten – also in der Praxis recht aufwendig. *R. Schindler, Völklinge*

Vielleicht ist das ein Denkanstoß für die Hardware-Spezialisten unter unseren Lesern?

Doppellaufwerk

Kürzlich bekam ich ein älteres Commodore-Doppellaufwerk 8250 geschenkt, das ich am C 64 anschließen und vor allem mit Geos nutzen möchte – leider gibt's keine Unterlagen zu diesem Gerät. Irgendwo habe ich gelesen, daß man es per Modul am Expansionsport anschließt – das würde zusätzlich für einen erheblichen Geschwindigkeitszuwachs sorgen! Wer hat Infos zu den Doppellaufwerken 8050 und 8250, kennt die Kontaktbelegung der 24poligen Anschlußbuchse und hat Infos zum Datenformat und Zeitdiagramm? Optimal wären natürlich Schaltpläne!

Danny Heinz, Carlsfeld

Magische Stimme

Auf einem Computer-Flohmarkt habe ich das Magic-Voice-Modul von Commodore sehr preisgünstig bekommen – allerdings ohne Bedienungsanleitung. Bisher konnte ich nur den SAY-

Befehl nutzen, z.B. SAY "Hallo" bzw. SAY A\$. Dann plappert der C 64 das Wort nach (stets nur eines), wenn es im Wortschatz vorhanden ist. Es gibt numerierte Wörter von 0 bis 234. Man kann sie auch mit SAY <Nummer> aufrufen.

Das Modul hat einen Audio-Ein- und Ausgang (Cinch-Buchsen). Der Anschluß AUDIO IN hat meine Phantasie angeregt: Ist es möglich, per Mikrofon eigene Sätze einzugeben, die dann digital gespeichert werden? Wer weiß darüber Bescheid?

Dr. Schulte-Holthausen, Lingen

Folie genügt

Frage von Timo Weinand in der 64'er 7/94: Wenn ich länger als fünf Minuten mit meiner Floppy 1541 arbeite, tauchen plötzlich I/O-Fehler auf. Woran liegt's?

Mit meiner Uralt-Floppy hatte ich ein ähnliches Problem: die Diskettenverriegelung drückte die Scheibe nicht mehr richtig an. Manchmal drehte sich die Disk nur ruckweise oder blieb ganz stehen.

Kleben Sie eine dünne Folie unter den Andruckhebel – dann müßte es wieder klappen (falls kein Hardware-Fehler vorliegt!).

Manfred Ciupke, Köln

Bunte Geos-Texte

Frage von Ralph Grund in der 64'er 2/94: Wie lassen sich Geos-Texte farbig ausgeben?

Die Frage wurde bereits in der 64'er 5/94 von Hagen Edlich be-

antwortet: Es gibt aber noch eine andere Methode: Man erreicht das auch durch die Manipulation der NLQ-Druckertreiber. Durch im Text verankerte Steuerzeichen lassen sich praktisch alle Druckbefehle zum Printer schicken. Die notwendigen Tools zum Ändern der Treiber findet man auf den Zubehör-Disketten des Geos-Systems (2.0 bzw. 2.5, also "Printer Edit" und die genannten NLQ-Treiber). Eine Disk mit zahlreichen Druckertreibern und eine Anleitung zum "Selbermachen" ist erhältlich gegen Einsendung von 13 Mark (inkl. Versandkosten) bei: Harald Gräfling, Dreyers Kamp 1, 28865 Lilienthal.

Printer Edit kann nicht alles ...

Ich habe festgestellt, daß man einige Geos-Druckertreiber mit "Printer Edit" modifizieren kann, aber noch lange nicht jeden!

Bei der Datei "!!LQ-1054/4" gibt's keine Probleme, aber "MPS-1550" läßt sich auf Biegen und Brechen nicht ändern! Dabei brauche ich dringend eine angepaßte Version für meinen Farbdrucker MPS 1550 C. Wie macht man aus dem Schwarzweiß-Treiber einen, der Farbe berücksichtigt? *Friedrich-W. Riechers, Eppstein*

Laserdrucker und Geos

Kürzlich habe ich den Oki-Laser OL 400 sehr preisgünstig bekommen. Mit Geos druckt er am Userport hervorragend, allerdings

nur in einer Schriftart. Um einen residenten Drucker-Font zu nutzen, der maximale Qualität bietet, muß ich in COMMODORE_GE schreiben und beim Ausdruck NLQ wählen. Attribute wie schräg, fett oder unterstrichen sind nicht möglich. Bei hochauflösendem Druck mit anderen Schriftarten erscheinen die Buchstaben treppenförmig. "GeoLaser" wäre ein Ausweg, aber da klappt's nur mit seriell angeschlossenen Druckern. Gibt es im PD-, Shareware- oder kommerziellen Software-Bereich ein Programm, das wie GeoLaser mehrere Drucker-Fonts aktiviert, aber aufs Parallelkabel reagiert?

Andreas Neef, Dresden

Spinnt die 1541?

Jedesmal, wenn ich meinen C 64 anwerfe und einen Brief mit Startexter schreiben will, stürzt das Programm nach ca. 30 Minuten ab: ein knackendes Geräusch – und vorbei ist's mit der Herrlichkeit. Ist die Floppy 1541 daran schuld? Hat jemand mit Startexter bereits ähnliche Probleme gehabt? *Hugo Hochstein, Essen*

C 128 und Monitor 1084

Der 1084 ist zwar ein Super-Monitor, leider nützt das dazugehörige Amiga-Anschlußkabel meinem C128D (Blech) überhaupt nichts. Der Monitor ist mit einem 8poligen TTL-RGB-Rundstecker, links mit einem sechspoligen RGB-Rundstecker und drei CVBS/Audio-Cinchstecker-Anschlüssen ausgerüstet.

Wie muß die Steckerbelegung fürs passende Kabel aussehen?

Alfons Sintke, Triftern

Die korrekten DIPs

Frage von Hans-Ulrich Dirlam in der 64'er 7/94: Kennt ein Leser die genaue DIP-Schalterstellung des Druckers Präsident 6320?

Mit dieser Stellung der DIP-Schalter (s. Kasten) arbeite ich im Textmodus (NLQ, nur mit Commodore-Schrift), bei anderen Fonts oder Grafik im Grafikmodus. Ich besitze das COM-Interface und benutze unter Geos 2.5 den Druckertreiber "FX 80/100".

Gerhard Bermig, Halle/Saale

Die Fragen unserer Leser werden selbstverständlich sofort nach Eingang einer Lösung brieflich vorab beantwortet – die Veröffentlichung dient dem allgemeinen Interesse.

Präsident 6320 (DIP-Schalterstellung)			
DIP-Nr.	Stellung	Funktion	Wirkung
6.1	on	Papierendekontakt	nicht wirksam
6.2	off	Summer	wirksam
7.1	on	automatische Zeilenschaltung	wirksam
7.2	off	automatischer Wagenrücklauf	wirksam
8.1	on	Zeichensatz deutsch	
8.2	off	Zeichensatz deutsch	
9.1	on	Papiervorschub (1 Zoll)	nicht wirksam
9.2	off	NLQ	wirksam
10.1	off	Schriftart PICA	
10.2	off	Schriftart PICA	
11.1	off	Nulldarstellung	ohne Schrägstrich
11.2	off	Zeilenabstand	1/6 Zoll
12.1	off	Formatlänge	12 Zoll
12.2	off	Formatlänge	12 Zoll
13.1	off	Steuercode EPSON	
13.2	on	Steuercode EPSON	
14.1	off	Steuercode EPSON	
14.2	off	nicht benutzt	
15.1	off	nicht benutzt	
15.2	off	nicht benutzt	
16.1	off	nicht benutzt	
16.2	off	nicht benutzt	
17.1	off	nicht benutzt	
17.2	off	nicht benutzt	
18.1	off	Geräteadresse	4
18.2	on	Befehlsatz	EPSON

Der Gamers-Programming-Guide

Unsere heutige Herausforderung hat einen kompliziert klingenden Namen: "Objekt-Formation". Dieses Thema der Struktur-Programmierung bleibt meist nur den Kreativgenies der Spieleabteilung überlassen, da ein normalsterblicher Hochsprachensympathisant doch eher selten über das vorherberechnete Bewegen und Animieren einzelner Grafik-Elemente sich seinen Kopf zerbricht. Ähnlich sieht es hier auch in der Literatur aus: Suchwort "Objekt-Formation" – glatte Fehlanzeige!

Hausaufgaben-Lösung

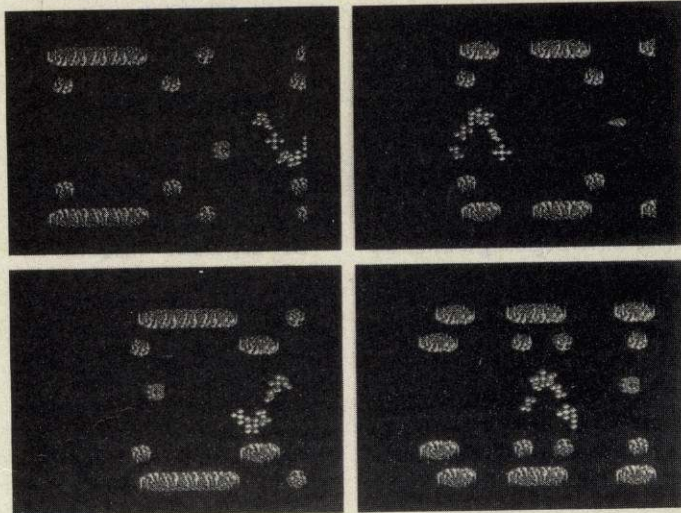
Bevor es aber so richtig an den Speck geht, die Auflösung zur letzten Ausgabe: Die Routine mit der Bezeichnung -???- diente dazu, die ersten Sprites (max. acht) bereits im unsichtbaren Bereich zu setzen, damit später nur noch die Sprites aufgerufen werden, die wirklich gemultiplext werden. Nur wenn die Routine erkennt, daß neun oder mehr Sprites aktiviert sind, löst sie einen IRQ zum Multiplexen aus. Sicher, wenn ich gleich losstarte, um meine dokumentierte Lösung zu erläutern, wird es sehr C-64-spezifisch ablaufen. Doch sind die Prinzipien und Grundmuster einer solchen Routine immer dieselben. Das heißt, diese Idee läßt sich auch ohne erhebliche Änderungen auf andere CPUs übertragen. Aus der Praxis weiß ich, daß auch Amiga oder PC sehr gut auf diese Art zu programmieren sind. Sollte jemand sogar in der Lage sein, auf Konsolen zu entwickeln und schon Erfahrung haben, so ist folgendes Schema der Objekt-Verwaltung geradezu optimal ...

Die Vorbereitung

Zu Beginn stellen wir uns wie immer die Frage: Was muß die Routine können? Angeführte Aufgabenstellung resultiert aus jahrelanger Beschäftigung mit Grafik-Animation:

1. Wie schnell muß die Routine sein?
2. Wieviel Sprites müssen in wieviel Rasterzeit bewegt werden?
3. Welcher zeit- und programmtechnische Aufwand darf der Routine zugemessen werden?

Nachdem unsere Sprites in Bewegung gekommen sind, wollen wir den kleinen Kobolden ein wenig Freiheit verschaffen und den Formationen Heimtücke einhauchen. Alien-Horden des Feind mit Pep werden unserem Imperiod-Gleiter einige Probleme bereiten.



Sinus-Tabellen sind bei Text-Scrollern angebracht, bei einem Action-Spiel aber müssen ausgefeilte Bewegungs-Routinen her, um den Sprites heimtückisches Leben einzuhauchen

4. Wie steht es mit der Kompatibilität zum Rest des Hauptprogramms?

In unserem Projekt "IMPERIOD" ist die Sache mit der Rasterzeit nicht so eng. Wir arbeiten mit sieben Sprites (Gegnern). Spiele, in welchen Horden mit bis zu 20 oder mehr Sprites erscheinen, sind in der Entwicklung bei der Rasterzeit viel genauer und besser durchdacht zu gestalten. Wir haben es uns schon ein wenig leichter gemacht, um nicht zu sagen einfach, denn auch sieben Sprites bringen durch ihre Formation und Animation eine Reihe von Begleit-Abfragen und Checkpoints mit sich, die heute unsere Aufmerksamkeit in Anspruch nehmen werden. Eins sei gleich vorweggenommen: wollen wir unsere Aliens in Formationen packen, welche dem Spieler einiges abverlangen und ihn nicht schon beim zweiten Versuch ein Gähnen entlocken, dann brauchen wir eines auf keinen Fall: jegliche Art der sogenannten "Sinusmaker".

Stupide Sinus-Tabellen sind vielleicht in gewissen Demo-

oder Slide-Shows recht nützlich, in Spielen der Action-Sorte haben sie nichts verloren. Wir müssen das schon mehr bieten, um mehr zu bieten.

Flexible Formationen

Unser Ziel ist es, sämtliche feindlichen Sprites in möglichst flexiblen, abwechslungsreichen Formationen, in welchen die einzelnen Bewegungen wieder sehr flüssig (s.Sinusmaker) wirken, über den Screen zu dirigieren. Daß wir dabei nicht jeden Bewegungsablauf getrennt in Form von Tabellen, deren Byte-Werte einfach nacheinander als X/Y-Positionen aufgerufen werden, im Speicher verwalten dürfen, ist logisch.

Angenommen wir lassen ein Alien fünf Sekunden von rechts nach links in gewisser Sinus-Manner wandern, haben wir bereits eine Page (255 Bytes) an RAM verbraucht. Wollen wir die Show auch in Y-Richtung starten, kostet dies die nächste Page, usw. Man sieht: gehen wir in diese Richtung, werden wir starr und unflexibel.

Zusätzlich gehen Massen an Speicher sprichwörtlich für nichts verloren (um den Sinusmaker-Freunden den letzten Kick zu verpassen ...). Gut, wir denken um und greifen zu folgender Lösung: Wir basteln eine Routine, welche X- und Y-Bewegungen getrennt verwaltet und Wiederholungen jeglicher Art erkennen soll. Alle Bewegungen laufen im überdimensionalen Bereich, also mehr als 320 x 200 Pixel, damit Sprites auch vom Rand ins Sichtbare bewegt werden können.

Clevere Routine im Einsatz

Da sich Tabellen bis jetzt immer als sehr hilfreich und Opcode-sparend erwiesen haben, legen wir sämtliche Bewegungs-Abläufe in geordneter Reihenfolge ab. Die Routine muß allerdings erkennen, wann sich eine Bewegung wiederholen soll, und wie oft. Um nochmals kostbaren Speicherplatz zu sparen, fügen wir ein Erkennungsbyte (on/off) ein, um Bewegungen spiegeln zu können. Um aber die Aliens überhaupt auf den Bildschirm zu setzen und ihnen zu sagen, welchen Weg sie einschlagen sollen, benötigen wir eine eigenständige Routine, die nur diese Aufgabe erledigt. Wir nennen diese Routine "GetEnemies".

Neue Tabellen, die Auskunft über Ort und Zeitpunkt der Angriffs-Formation geben sowie deren Bewegungs-Ablauf initialisieren, dienen dazu, um die Verwaltung möglichst übersichtlich zu halten. Wenn wir Formations-Studien bestimmter Animationen nachträglich verändern wollen, sparen wir so viel Zeit und vor allem Nerven! In "GetEnemies" lassen wir einen Timer laufen, der immer dann, wenn er auf 0 ist, eine neue Byte-Serie einliest, um ein Objekt neu zu starten, bzw. ins Spielgeschehen zu schicken. Dazu notwendig ist die genaue Start-Position (2 Bytes, X/Y), Block & Farbe, das Info-Byte für die Bewegung (wo soll die Formation in der Datenliste zu finden sein?), und zum Schluß noch der neue Wert für den Timer, der den Abstand zum nächsten Sprite, das erscheinen soll, fixiert. Ist ein Objekt einmal gestartet, wird es in unserer ursprünglichen und eigentlichen Bewegungs-Routine

über den Bildschirm geschoben. Sie hat nur die Aufgabe, von der initialisierten Startposition aus das Sprite (nach Tabelle) in alle Richtungen zu setzen. Bewegungs-Spiegelungen in X- und Y-Richtung werden durch unser Programm erkannt, genauso wie die Y-Koordinaten der Sprites, die den sichtbaren Bereich des Bildschirms verlassen oder vom Spieler vernichtet wurden.

Diese Objekte müssen vom Bildschirm verschwinden und werden auf ZERO gestellt. Der Zustand der Sprites wird auf OFF geschaltet und für neue Aktionen freigegeben. Gut, das klingt sehr theoretisch. Der dokumentierte Assembler-Quelltext (Listing 2) sollte hier Durchblick schaffen und helfen, die Materie besser zu verstehen. Vielleicht nicht beim ersten Mal, aber spätestens beim zweiten Anlauf.

Die GetEnemies-Routine

Auf geht's: Zuallererst wollen wir die Objekt-Bewegungsroutine "Move Enemies" detailliert durchleuchten.

Gleich zu Beginn, bei Label "m1" wird die Y-Position getestet. Ist diese gleich 0, gilt das Sprite als tot und wird die Animation übersprungen. Bei "m2" wird ein Count-Byte (Zähler) geladen, welches darüber Auskunft gibt, wie oft ein Bewegungs-Step noch durchgeführt werden muß. Dazu ist wichtig, daß sich diese Schrittgrößen in "horiz" und "verti" befinden und in X und Y jeweils in beide Richtungen durch positive oder negative Werte richten können. Ist der "Count"(er) abgelaufen, werden ab "mm2" neue Steps (Schritte) geladen. Mit dem Index "point" werden aus der "movedats"-Tabelle drei Wer-

te geladen. Der erste ist im Normalfall der Y-Step, dann folgt der X-Step und schließlich noch der Wert für den Counter, der signalisiert, wie oft dieser Schritt vollzogen werden soll.

Im "Normalfall" deshalb: Ist nämlich im ersten Byte nur das siebente Bit und das erste Bit und (oder) das zweite Bit gesetzt (es werden Werte von #128 bis #131 getestet), wird gesprungen und (oder) gespiegelt. Das heißt, wenn unsere Routine einen Wert von #128 findet, eine ganze Reihenfolge von Steps wiederholt wird, solange unser "wieder"-Byte noch nicht den Wert des zweiten Tabellen-Bytes (movedats+1) erreicht hat.

Wiederholt wird ab der Adresse, die im nächsten Byte (movedats+2) zu finden ist. Dabei wird unser "wieder"-Byte erhöht, damit sich der Rechner merkt, wie

oft bereits zurückgesprungen wurde (siehe Label "m5"). Ist das siebente Bit in Verbindung mit den ersten beiden Bits gesetzt, also #129-#131, hat dies Auswirkung auf unsere "mirrx"- und "mirry"-Flags, die zur Spiegelung dienen. Bei positivem Test werden diese Flags einfach invertiert, d.h. egal, ob die Bewegung bis jetzt invertiert war oder nicht.

Bei der nächsten Wiederholung wird sie genau umgekehrt ausgeführt. Bei "m6" wird noch unser "point"-Index entsprechend erhöht, um auf die nächsten Adressen zu zeigen, um schließlich die eigentlichen Bewegungs-Werte zu erfassen. "m7" ist die Station, die von der CPU im Normalfall angesprungen wird, wenn keine Sonderwerte (7. Bit) bearbeitet werden; hier werden die Steps gesetzt. Je nachdem, ob eine Spiegelung in den mirrx- und mirry-Flags zu

Listing 1: Die Objekt-Bewegungs-Routine

```

;-----
;
; Objektverwaltung fuer Imperoid-Projekt by H.Sommer 1994
;-----
m1  ldx #7          ;7 Objekte
    lda y,x        ;Y-Koord.laden
    beq tm20       ; 0 dann m2
    jmp m20        ;sonst Ende
m2  lda count,x    ;Zähler laden
    beq mm2        ;0 dann neue Werte
    jmp m10        ;sonst Formation
mm2 ldy point,x    ;Pointer ins Y-Reg
    lda movedats,y ;Erster Wert
    cmp #128       ;7.Bit gesetzt ?
    beq m5         ;ja, neue Addy
    cmp #129       ;Bit 7+0 gesetzt ?
    beq m4         ;dann Y-Spiegeln
    cmp #130       ;Bit 7+1 gesetzt ?
    bne m3         ;dann X-Spiegeln
    lda mirrx,x    ;Mirrx-Byte
    eor #1         ;mit #1
    sta mirrx,x    ;verknüpfen
    jmp m5         ;weiter...
m3  cmp #131       ;Bit 7+0+1 gesetzt ?
    bne m7         ;nein, dann m7
    lda mirrx,x    ;ja, dann
    eor #1         ;X & Y-Spiegeln
    sta mirrx,x    ;...
m4  lda mirry,x    ;hier wird
    eor #1         ;mirry
    sta mirry,x    ;invertiert
m5  inc wieder,x   ;Move-Wiederholungen
    lda wieder,x   ;zählen und mit
    cmp movedats+1,y ;Wert aus movedats
    beq m6         ;vergleichen
    lda movedats+2,y ;neuen Wert für
    sta point,x    ;pointer schreiben
    tay           ;und ins Y-Reg.
    jmp m7         ;dann nach m7
m6  lda point,x    ;pointer um #3
    clc           ;erhöhen, da immer
    adc #3        ;3 Werte eingelesen
    sta point,x   ;werden
    tay           ;wieder ins Y-Reg.
m7  lda mirry,x   ;Test auf Spiegeln
    bne mm7       ;Ymirror gesetzt ?
    lda movedats,x ;Wert normal
    sta verti,x   ;setzen
    jmp m8        ;weiter zum X-Wert
mm7 lda #0        ;Y-Wert von #0
    sec          ;subtrahieren, um
    sbc movedats,y ;Umkehrwert zu
    sta verti,x   ;erhalten
m8  lda mirrx,x   ;Test auf X-spiegeln
    bne mm8       ;ja, dann mm8
    lda movedats+1,y ;Horizontal-Bewegung
    sta horiz,x   ;normal
    jmp m9        ;weiter nach m9
mm8 lda #0        ;Umkehrwert wie
    sec          ;gehabt nun auch
    sbc movedats+1,y ;für X-Wert
    sta horiz,x   ;errechnen
m9  lda movedats+2,y ;Zähler-Wert aus
    sta count,x   ;movedats+2 in count
    lda point,x   ;point-byte
    clc          ;wieder um
    adc #3       ;#3
    sta point,x   ;erhöhen
;----- eigentliche Bewegung ab hier: -----
m10 dec count,x   ;Zähler erniedrigen
    lda y,x       ;Y-Koordinate
    clc          ;plus
    adc verti,x   ;verti-byte
    sta y,x       ;und schreiben
    lda horiz,x   ;Horiz-Bewegung:
    bpl m11       ;positiv: nach m11
    clc          ;sonst
    adc x,x       ;einfach x-Kord.
    sta x,x       ;dazuzählen
    bcs m20       ;bei Übertrag: m20
    bcc m12       ;sonst Xhi-Wert...
m11 clc          ;wieder
    adc x,x       ;zu X-Koord.
    sta x,x       ;addieren
    bcc m20       ;kein Übertrag: m20
m12 lda xh,x     ;entsprechendes
    eor #1        ;Xh-Byte mit #1
    sta xh,x     ;setzen/löschen
m20 dex         ;Sprite-Nr. weniger
    beq m21       ;0 dann fertig
    jmp m1        ;zurück nach m1
m21

```



Vier Schappschüsse zum aktuellen Stand unseres Spiele-Programmier-Projekt "Imperio" – die Sprites bewegen sich in zahlreichen Varianten geschmeidig über den Bildschirm

berücksichtigen ist oder nicht, werden die Tabellen-Bytes aus "movedats" in die "verti"- und "horiz"-Labels geschrieben. Wie zuvor erwähnt, können diese Werte positiv oder negativ sein.

Nun sind wir bei "m10" angelangt, kommen wir endlich zum Bewegen der Sprite-Koordinaten. In Y-Richtung ist dies recht simpel, da der 1-Byte-Step einfach addiert werden kann. Wir erreichen eine Bewegung in die gewünschte Richtung. Überträge bei der Addition sind uninteressant. In horizontaler Richtung sieht's da anders aus, denn wir arbeiten auf einer X-Achse, die mindestens

320 Pixel breit ist und da hier Über- und Unterläufe registriert werden müssen, und zwar im *xh*-Byte. Prüfen wir zuerst, ob der "horiz"-step positiv oder negativ ist und ob er nach rechts oder links schreitet. Dies erledigt der "bpl m11"-Befehl. Danach wird addiert und das *xh*-Byte dementsprechend behandelt.

Bei "m20" angelangt, sind wir auch schon am Ende der Bewegungs-Routine. Auch das geübte Coder-Auge wird diese Art der Animation, Bewegung und Spiegelung mit scharfem Blick betrachten müssen, um alles zu erfassen ...

Sprites-Verwaltung

Nun noch die Erläuterung der "GetEnemies-Objects"-Routine in das heutige Deutsch:

Alles läuft im 1/3-Action-Mode ab, d.h. der "t51"+1-Counter wird jeden dritten Raster-Durchlauf erhöht. Ist dieser auf 0, werden aus der Tabelle "enemies" neue Start-Indizes gelesen und verarbeitet. Wird ein #255-Wert registriert, setzt unsere Routine die Zeropage-Adressen "elo" und "ehi" wieder an ihren Startwert zurück und die Formationen werden von Beginn an wiederholt.

Jeder andere Wert wird bei "t51" ins X-Register geschoben und dient ab jetzt als Sprite-Index. Also gibt der erste Wert aus der Tabelle an, welches Sprite (0-7) gestartet werden soll. Jetzt werden alle Info-Bytes gelöscht (*xh*, *mirrx* usw.) und die Werte 2 und 3 als X- und Y-Position geladen und verwertet. Die X-Position ist als Loes-Wert zu sehen, da sie direkt in das entsprechende *ix*-Byte geschrieben und danach erst verdoppelt wird.

Wir erreichen somit die gesamte X-Breite mit nur einem Byte. Wert Nr. 4 dient als Block-Start-Adresse. Im Verlauf des nächsten Kursteils werden wir noch genauer auf Sprite-Block-Animationen eingehen. Byte Nr. 5 wird sowohl als Farbwert als auch als Flag für Spiegelungen gewertet. Da ein Farbwert nur vier Bits benötigt,

also ein Lo-Nibble, spricht nichts gegen eine Benützung des sechsten und siebenten Bits als Mirror-Flags. In der Gegend des Labels "t53" sehen wir, daß sich ein gesetztes 7. Bit auf die X-Spiegelung auswirkt, genauso wie ein sechstes Bit das "mirry"-Flag beeinflusst. Nach "t54" wird als Byte Nr. 6 die Bewegung initialisiert und zum guten Ende noch ein Timer-Wert für "t51" gesetzt.

Phuuu! Die letzten beiden Kursteile waren bestimmt das Härteste am gesamten Kurs. Wenn wir uns nächstes Mal mit Sprite-Block-Animationen beschäftigen, wird's sicher nicht noch mal ganz so heiß. Eine Hausaufgabe gibt's diesmal nicht, da das Studium des Programms genügend Zeit in Anspruch nimmt. Hannes Sommer/lb

Kursübersicht

- Folge 1**
Anfang/Speicheraufteilung/
Modi-Wahl
- Folge 2**
Einführung Grafik-Effekte
- Folge 3**
Scrolling/Spiel-Verwaltung
- Folge 4**
Sprites/Multiplexen
- Folge 5**
Objekt-Animation/Formation
- Folge 6**
Kollisions-Abfrage/Verwaltung
- Folge 7**
Sound und Musik
- Folge 8**
Level-Design und letzter Schliff

Listing 2: Die GetEnemies-Routine

```

t51      dec t51+1          ;Counter vermindern
        lda #20           ;und laden
        bne t55           ; 0 dann t55
        ldy #0           ;Y-Reg. als Index
        lda (elo),y      ;y=0: Sprite-Nr.
        cmp #255         ;compare auf #255
        bne tt51         ;Nein, normal weiter
        lda #enemies     ;Adresse (lo+hi)
        sta elo          ;neu initialisieren
        lda #enemies     ;und nach elo und
        sta ehi          ;ehi schreiben
        lda (elo),y      ;Inhalt laden
        tax              ;Sprite-Nr.ins X-Reg
        iny              ;nächstes Byte
        lda #0           ;Initialisierung
        sta xh,x         ;sämtlicher
        sta mirrx,x      ;Sprite-
        sta mirry,x      ;Info-
        sta wieder,x     ;Bytes
        sta count,x      ;...
        lda (elo),y      ;Y=1: X-Pos
        sta ix,x         ;nach ix (lo-res)
        asl a             ;verdoppeln und
        sta x,x          ;ins x schreiben
        bcc t52          ;Überlauf ?
        inc xh,x         ;ja, dann Xhibyte
        iny              ;nächstes Byte
t52      lda (elo),y      ;Y=2: Y-Pos
        clc              ;dazu als
        adc #110         ;Fixwert #110
        sta y,y          ;addieren
        iny              ;nächstes Byte
        lda (elo),y      ;Y=3: Block
        sta b,x          ;ins b-byte
        iny              ;nächstes Byte
        lda (elo),y      ;Y=4: Colour
        sta c,x          ;ins c-byte
        and #128         ;Bit 7 testen
        beq t53          ;nicht gesetzt: t53
        inc mirrx,x      ;Xmirror on!
        lda (elo),y      ;nochmals lesen
        and #64          ;Bit 6 testen
        beq t54          ;0, dann nach t54
        inc mirry,x      ;Ymirror on!
        iny              ;nächstes Byte
t54      lda (elo),y      ;Y=5: Movement
        sta point,x      ;nach pointer
        iny              ;letztes Byte
        lda (elo),y      ;Y=6: Delay
        sta t51+1        ;nach t51-Timer
        lda elo          ;elo-
        clc              ;Adressen
        adc #7           ;um #7
        sta elo          ;erhöhen
        bcc t55          ;Überlauf ?
        inc ehi          ;Hi-Byte erhöhen
t55      ;Fine !!!
    
```

Effizientes Programmieren mit dem Basic-Boss



Power für den Boss

Wer mit dem Basic-Boss arbeitet, hat grundsätzlich zwei Möglichkeiten:

1. Man schreibt seine Programme wie gewohnt und kompiliert sie dann. Dies bringt im Vergleich zum normalen BASIC-Programm schon einen deutlichen Geschwindigkeitsgewinn.

2. Grundsätzlich werden bei der Quelltext-Programmierung die Compilierungsmöglichkeiten voll ausgenutzt. Das bedeutet eine erheblich schnellere Codeausführung und -übersetzung sowie einen kürzeren Objektcode. Doch gleichzeitig wird dabei in den meisten Fällen die Lauffähigkeit unter dem Interpreter verlorengelassen. Nachteil: für jeden Testlauf muß der sehr lange Compiler erst geladen werden.

Da wir in diesem Workshop den Basic-Boss mal so richtig ausreizen wollen, ist nur die 2. Möglichkeit von Interesse.

Die Grundregeln

Vergessen Sie alles, was Sie über speicherplatzsparende Programmierung oder Beschleunigung des BASIC unter dem Interpreter gelesen haben! Der Basic-Boss stellt andere Anforderungen.

Zunächst ist es ratsam, so strukturiert und übersichtlich wie möglich zu programmieren. Falls Ihnen PASCAL ein Begriff ist, sei gesagt, daß die dortigen Strukturen auch für den Basic-Boss sehr ratsam sind: Definitionsteil für Variablen und Konstanten, dann alle Prozeduren (Unterroutinen) und Funktionen und schließlich ein kurzes Hauptprogramm, das eigentlich nur Steuerungsfunktion haben sollte und die entsprechenden Module per GOSUB aufruft. Vermeiden Sie unnötige Sprünge mit GOTO sowohl im Hauptprogramm als auch in den Modulen. Wer sorgfältig arbeitet, wird sie kaum brauchen.

Planen Sie genau, welche Variablen wofür sind und welche Wertebereiche sie annehmen können. Am effektivsten arbeiten

Im Vergleichstest mit dem Compiler "Basic-64" hat sich der "Basic Boss" behaupten können. Grund genug, unserem Test in Heft 8 heute ein paar Tricks zur optimalen Nutzung folgen zu lassen.

Byte und Word, notfalls auch Integer (die Unterschiede sind im Handbuch erklärt).

Keine unnötigen Typwandlungen durch entsprechende Vorausplanungen! Und legen Sie sich für häufig wiederkehrende Rechnungen Tabellen an. Wie das am sinnvollsten geht, steht in der Anleitung sehr gut beschrieben.

Statt mehrdimensionaler Arrays sollten Sie eindimensionale Datenreihen einsetzen und die Positionen entsprechend berechnen oder zusätzlich Indextabellen anlegen. Gerade in diesem Bereich sind deutliche Geschwindigkeitsgewinne möglich, wenn man sich vor dem Programmieren genauere Gedanken macht.

Wenn Sie sich an all diese Regeln halten, dürfte es keine Schwierigkeiten mehr machen, die Direktiven `£fastfor`, `£fastarray` und `£shortif` einzusetzen.

Ein Beispiel

Folgendes kurze Programm

```
10 OPEN1,8,0,"$":
POKE781,1;SYS65478:GETAS,AS
20 GETAS,AS:IFST=64 THEN-
SYS65484;CLOSE1:END
30 GETAS,BS:PRINT"{left}"ASC(AS
+CHR$(0))+256*ASC(BS+CHR$(0));
40 GETAS:PRINTAS;:IFA$""THEN40
50 PRINT:GOTO20
```

ist eine Directory-Anzeigeroutine. In BASIC läuft sie ziemlich langsam, kompiliert ist ein Geschwindigkeitsgewinn festzustellen. Der Basic-Boss verfügt über neue Ein- und Ausgabebefehle, so daß es noch besser geht, was aber den Verlust der Lauffähigkeit des Quelltextes unter dem Interpreter zur Folge hat. ("`<`" steht für Linkspfeil)

```
10 OPEN1,8,0,"$"
20 <-INI
25 X=@WORD
30 X=@WORD
```

Da das Floppy-DOS bei einem geöffneten Kanal (hier NR. 1) nur Lese- oder Schreibzugriffe erlaubt, müssen wir dem Compiler mitteilen, was wir wollen (Lesen durch Kanal 1, also `<-in1`).

Die zweifache Ausführung von `x=@word` liest jeweils einen Word-Wert (also zwei Byte) aus der geöffneten Datei, dies entspricht den GET-Befehlen der Zeilen 10 und 20 aus dem ersten Listing und überspringt dadurch die für uns uninteressante Startadresse und den Zeilenlink. Wenn damit das Dateiende erreicht ist, erfolgt ein Abbruch,

```
(35 IF ST THEN 70)
ansonsten wird die nächste Blocklänge gelesen, der der weitere Dateinamentext folgt:
40 PRINT @WORD
50 A=@BYTE:<-BYTE A:IF A THEN
50
60 PRINT:GOTO 30
```

Dieser Abschnitt erfüllt die gleiche Funktion wie schon die Zeilen 30-50 in Listing 1.

Nun reicht es nicht, die Datei einfach zu schließen, es muß erst noch auf Bildschirm- und Tastaturkanal, also die Standardeinstellung, umgeschaltet werden.

```
70 <-RESET
80 CLOSE 1
```

Von der Funktionsweise her sind die Programme gleich, die Übersetzung kann jedoch vom Basic-Boss verschieden bearbeitet werden und führt im zweiten Fall zu besseren Ergebnissen.

Overlay-Technik

Durch den Befehl `fallram` wird der Bereich von \$0801 bis \$ffff für den Objektcode benutzt – also stolze 62 KByte. Für große Programme ist dies sicher sehr nützlich, jedoch auch mit einigen Problemen verbunden: Meist wird der Speicher nach \$d000 für Grafik-,

Sprite-, Musik-, Zeichensatz- oder andere Arten von Daten verwendet. Was ist nun, wenn wir davon ausgehen, daß unser Programm sehr lang ist und der Bereich ab \$d000 für andere Sachen als Programmcode gebraucht wird?

Die Lösung: Module. Die Programmmodule legen wir in einzelne Dateien ab. Dann ist noch ein Steuerungsprogramm erforderlich, das jeweils das Modul lädt, welches gerade gebraucht wird. Dieses kurze Menü legen wir an den Basic-Start und lassen es auch unkompiliert. Die Module werden dann, je nach Länge der Menüs, z.B. nach \$1000 oder \$2000 gelegt. Dazu reicht die Anweisung `£routstart (adresse)` meist aus, allerdings muß der Routinenteil (wird bei der Compilierung angezeigt) bei \$d000 beendet sein, für den Programmteil – also den eigentlichen Quellcode – ist das, zumindest im `fallram`-Modus, egal.

Diese Module werden nun absolut vom Menü nachgeladen und mit SYS (adresse) gestartet. Nach der Beendigung kehrt die so aufgerufene Routine an die Stelle nach dem SYS-Befehl zurück.

Datenaustausch der Module

Was ist nun, wenn diese Module untereinander oder mit dem Steuer(menü)programm kommunizieren wollen, also Datenaustausch nötig ist?

Mit Variablen geht es nicht, da die Programme alle ihren eigenen Variablenspeicher verwalten. Also suchen wir uns einige selten gebrauchte Speicherstellen, z.B. den Cassettenpuffer oder die letzten Bytes des Bildschirmspeichers und lassen die Programme durch PEEK und POKE darauf zugreifen. Strings müssen natürlich via Schleife als ASCII-Werte abgelegt werden, um sie dann von dem aufzurufenden Programm erneut zusammensetzen zu lassen.

Olaf Dzwoja / ma



Warnung

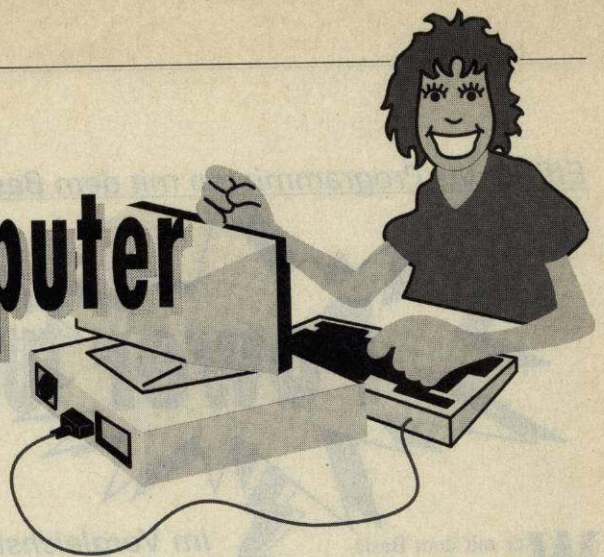
Für die Arbeiten mit dem Compiler sollten Sie sich je eine separate Compilierungs- und Quellcodediskette anlegen, da der Compiler von Zeit zu Zeit mal den Quellcode zerstören kann. Darauf wird zwar in einer Infodatei auf der Programm diskette hingewiesen, doch ist die Erklärung des Autors schwer nachzuvollziehen. Der Fehler tritt sporadisch und vom Quellcode unabhängig auf, ist nicht rekonstruierbar und somit schwer, wie vom Autor angegeben, bei der Hardware zu suchen. Sicherheitskopien der Programme lösen dieses Problem jedoch auf einfache Weise.

Künstliche Intelligenz

Folge 3

auf der Schulbank

Computer



Wir haben dem C 64 beigebracht, bis zu 200 verschiedene Tiernamen zu erkennen und deren Unterscheidungsmerkmale einzuordnen. Im dritten und letzten Teil unseres Kurses geht's darum, wie sich der Computer die Daten auf Dauer merkt.

In den bisherigen beiden Kursfolgen wurde gezeigt, wie der C 64 lernt: leider ist sein mühsam angeeignetes Wissen bei Ausschalten des Computers futsch – die Daten existieren bislang nur im Arbeitsspeicher, also im Kurzzeitgedächtnis des C 64. Bei jedem Neustart des Programms gehen alle Informationen verloren.

Das Langzeitgedächtnis unseres Computers ist die Diskette. Wir kennen vier Methoden, Daten auf solchen Scheiben abzulegen:

- als Teil des Basic-Programms,
- sequentielle,
- relative oder
- Direktzugriffsdatei (Random Access).

Unser Problem ist nämlich, daß der Datenbestand laufend größer wird – bei jedem Programmdurchlauf, bei dem neue Tiere und deren Unterscheidungsmerkmale dazukommen.

Alle genannten Speichermethoden haben Vor- und Nachteile. Wir wählen die einfachste: das Sichern in einer sequentiellen Datei (SEQ). Die anderen Versionen böten genug Haken und Ösen für einen separaten Kurs.

Der Übersichtlichkeit zuliebe: sequentielle Dateien

Bei SEQ-Dateien kann man lediglich in fixierter Reihenfolge auf die Daten zugreifen, ihr Vorteil

liegt allerdings in der Übersichtlichkeit. Außerdem dürfen die einzelnen Daten (= Variablenstrings) unterschiedlich lang sein.

Daran sieht man, daß unsere geplante sequentielle Datei große Ähnlichkeit mit unserem Variablenfeld (String-Array) hat, das ebenfalls aus verschiedenen langen Zeichenketten besteht. Nachteil: Lange Wartezeiten, wenn man innerhalb umfangreicher Dateien nach bestimmten Informationen sucht. Auch lassen sich nur so viele Daten speichern, wie in den Arbeitsspeicher des C 64 passen.

OPEN-Anweisung

- öffnet eine Datei zum Schreiben oder Lesen.
- Hinter der OPEN-Anweisung stehen bis zu sechs weitere Angaben, die durch Kommata voneinander getrennt sein müssen:

OPEN Dateinummer (1 bis 255),
Gerätenummer, Sekundäradresse
(1 bis 15), "Dateiname, Dateityp, Modus"

- bis zu zehn Dateien mit unterschiedlichen Dateinummern dürfen gleichzeitig geöffnet sein,
- Diese **Gerätenummern** werden akzeptiert:

- 0 = Tastatur,
- 1 = Datensette,
- 3 = Bildschirm,
- 4/5 = Drucker,
- 8 = Floppylaufwerk 1,
- 9 = ... Nr. 2,
- 10 = ... Nr. 3,
- 11 = ... Nr. 4.

- Die **Sekundäradressen** 0 und 1 sind bei Diskettenstationen vom Betriebssystem reserviert. Die Werte 2 bis 14 belegen je einen numerierten Datenkanal (Achtung: es dürfen maximal drei Datenkanäle aktiviert sein!), Nr. 15 ist für den Befehlskanal eingerichtet (Diskettenoperationen wie Formatieren, Namen ändern, Dateien löschen usw.).
- Das Floppy-DOS verwendet vier **Dateitypen**:

- S = sequentielle Datei (SEQ),
- U = User (USR),
- P = Programm (PRG),
- R = relative Datei (REL).
- Mit **Modus** legt man fest, wie der Datenkanal benutzt wird:
W = Schreiben (WRITE),
R = Lesen (READ),
A = Anfügen (APPEND),
M = Lesen nicht geschlossener Dateien.

Vier markante Basic-Anweisungen unterstützen die Arbeit mit SEQ-Dateien:

- OPEN und CLOSE (Datei öffnen und schließen),
- PRINT#: überträgt Daten auf Disk,
- INPUT#: ... holt sie aus der Diskettendatei und legt sie wieder im Computerspeicher ab.

Speichert man das erworbene Wissen unseres C 64 als SEQ-Datei, muß stets der gesamte Array-Inhalt W\$(200) gesichert werden – der alte Datenbestand auf Disk wird dabei überschrieben.

Wer aber die im Textkasten zur OPEN-Anweisung genannten Modi genauer untersucht, findet den APPEND-Befehl, um bestehende Dateien zu vergrößern.

Da sich jetzt sicher mancher beschwert, es sei doch viel zu umständlich, die gesamte Datei zu überschreiben, wenn man doch den APPEND-Befehl habe, wollen wir uns dem Thema ausführlich widmen.

Geben Sie unser Beispiellisting ein:

```
2000 OPEN 1,8,2,"TEST,S,W
2010 FOR Z=0 TO 3
2020 PRINT#1,Z
2030 NEXT Z
2040 CLOSE 1
```

Nach dem Startbefehl RUN 2000 (achten Sie darauf, daß eine Disk im Laufwerk liegt!), transferiert der C 64 die Zahlen 0 bis 3 in die Datei "Test".

Ergänzen Sie das Listing:

```
2050 OPEN 1,8,2,"TEST,S,A"
2060 PRINT#1,25
2070 CLOSE 1
```

Mit RUN 2050 hängt man nun den numerischen Wert "25" an die Datei "Test".

Und das ist allerdings der Knackpunkt: Die APPEND-Anweisung schreibt stets ans Ende einer SEQ-Datei, wir hängen zwar ebenfalls zwei neue Einträge an unser Datenfile (W\$(W) und W\$(W+1)), ersetzen aber auch gleichzeitig den Inhalt des bereits reservierten Speicherplatzes W\$(M) durch neue Daten.

CLOSE-Anweisung

- ... benötigt als Parameter die Dateinummer des OPEN-Befehls und schließt die Datei,
- Man muß jede Datei unbedingt schließen, sonst wird hinter den letzten Eintrag keine Endmarkierung gesetzt – die letzten Daten gehen dadurch verloren oder die Gesamtdatei läßt sich nicht mehr laden!
- Im Directory sind nicht geschlossene Dateien mit dem Asterisk (= <*>) gekennzeichnet.

Das kann APPEND jedoch nicht: Daten innerhalb eines Files austauschen. Das klappert nur bei REL-Dateien, denn sie sind anders strukturiert.

Sie sehen, es bleibt uns gar nichts anderes übrig, als die alte Datei auf Disk mit der neuen Tierliste komplett zu überschreiben.

Routinen zum Sichern

In unserem (in der letzten Kursfolge vollendeten) Demo-Programm gibt's zwei Stellen, wo Datenspeicherung Sinn macht:

- nach dem Lernvorgang (also nach Zeile 970),
- vor dem Abschluß eines Ratespiels (nach Zeile 1120).

Wir empfehlen Vorschlag 1 – aus Sicherheitsgründen. Es wäre schade, nach einem langen Spiel mit vielen Neueingaben bei einer verunzten Eingabe auszusteigen (Vorsicht: INPUT-Befehl) und dann nach einem Neustart mit RUN alle Daten zu verlieren!

Für die Datensicherung ist der Zeilenblock ab 1000 in unserem Basic-Programm vorgesehen:

```
1010 OPEN
1,8,2,"@:TIERLISTE,S,W"
1020 FOR Z=0 TO VAL(W$(0))
1030 PRINT#1,W$(Z)
1040 NEXT Z
1050 CLOSE 1
```

Per Schleife (Zeilen 1020 bis 1040) wird einfach von 0 bis zum letzten belegten Speicherplatz hochgezählt und jeder Variableninhalt einzeln auf Disk übertragen. Der aktuelle Wert des letzten Speicherplatzes steht als String in

W\$(0), per VAL-Funktion wird er in eine Zahl umgewandelt. Achten Sie dabei auf die strikte Anzahl der öffnenden und schließenden Klammern!

Den Dateinamen bestimmt man in Zeile 1010 (z.B. "Tierliste"); ergänzt durch die Parameter S (SEQ-Datei) und W (= Daten schreiben). Vor dem Dateinamen finden Sie das At-Sign-Zeichen (Klammeraffe) – das ist das REPLACE-Kommando fürs Floppy-DOS, eine bereits bestehende, gleichnamige Datei zu überschreiben.

Bei älteren Floppy-Versionen kann der REPLACE-Befehl Sorgen bereiten (Diskettenblöcke werden nicht mehr korrekt an die alte Stelle geschrieben) – das umgeht man am besten mit der SCRATCH-Anweisung (alte Datei auf Disk löschen), die man in Zeile 1000 verankert (dann sieht auch Zeile 1010 ein bißchen anders aus):
 1000 OPEN 2,8,15,"S:TIERLISTE"
 1010 OPEN 1,8,2,"TIERLISTE,S,W"

Auch Zeile 1050 ändert sich: jetzt muß man zwei geöffnete Datenpfade wieder schließen:
 1050 CLOSE 1: CLOSE 2

Datenzugriff bei Programmstart

Bleibt noch die Aufgabe, die in "Tierliste" gesicherten Daten vor Beginn des Ratespiels wieder in den Arbeitsspeicher des C 64 zu laden.

Damit fällt unsere Variablendefinition ab Zeile 210 weg – dieser Bereich wird durch die Laderoutine ersetzt:

```
200 OPEN 1,8,2,"TIERLISTE,S,R"  
210 INPUT#1,W$(0)  
220 FOR Z=1 TO VAL(W$(0))-1  
230 INPUT#1,W$(Z)  
240 NEXT Z  
250 CLOSE 1
```

Zeile 200 entspricht Nr. 1020, als Modus wurde jetzt aber R (= Lesen) eingestellt. Auch die Zählschleife ab Zeile 220 ähnelt der in 1020 bis 1040.

Allerdings: bei Programmbeginn, also noch vor dem Laden, wissen wir noch nicht, wieviele Daten das File enthält und bis zu welcher Zahl die Leseschleife laufen muß.

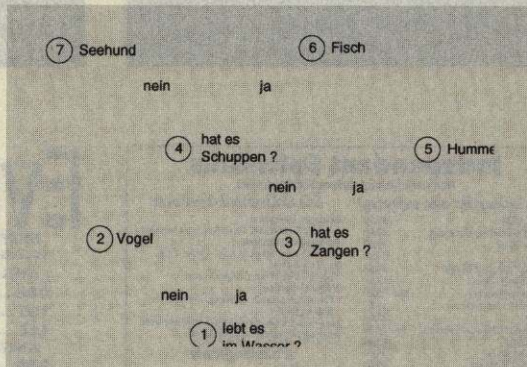
Das ist nämlich die erste Info unserer Diskettendatei (Zeile 210 liest den Wert), sie wird in der Variablen W\$(0) abgelegt. Deshalb macht die Zählschleife in Zeile 220 ab Z - 1 weiter und läuft nur bis VAL(W\$(0))-1.

Eigentlich steht einem Probeauf unseres (fast vollständigen) KI-Programms nichts mehr im Weg – wäre da nicht das Problem,

daß der C 64 – nach dem Einschalten ahnungslos wie ein Säugling – natürlich auch nicht wissen kann, ob auf der Disk im Laufwerk die Datei "Tierliste" auch tatsächlich vorhanden ist.

Wenn nicht, provoziert die Laderoutine ab Zeile 200 das unvermeidliche Blinken der Floppy-LED – als Kennzeichen, daß sich ein Disk-Error eingeschlichen hat.

Wenigstens stürzt unser Programm nicht ab, sondern macht unbeeinträchtigt weiter – es läuft über Programmblock 400 hinaus bis in die Eingabeschleife ab Zeile 550. Allerdings ist die anschließende Unterschiedsfrage (ja/nein) dann sinnlos – das Programm hat ja keine Daten!



Wie ein Baum mit vielen Ästen: Das Wissen des Computers wächst mit jeder Frage!

INPUT#-Anweisung

- ... liest Daten einer mit OPEN geöffneten Datei von externen Eingabegeräten (Datasette, Floppy, CMD-Harddisks).
- Einzulesende Zeichenketten dürfen maximal 88 Zeichen lang sein.
- Es gelten die gleichen syntaktischen Vorschriften wie beim PRINT#-Befehl.

Sind schon Daten im Speicher?

Wir sollten also verhindern, daß diese Situation bei Programmstart überhaupt entsteht. Die Routine baut man am besten beim Startblock ab Zeile 310 ein:

```
310 PRINT "DENKE DIR BITTE EIN TIER AUS, DAS ICH ERRATEN MUSS."  
320 INPUT "BIST DU BEREIT (JA/NEIN) ";A$  
330 IF LEFT$(A$,1) <> "J" THEN  
320
```

Aufpassen: diese drei Zeilen werden bei jedem Start aktiviert, unabhängig davon, ob eine Tierliste geladen wurde oder nicht. Trifft das zu, steht in W\$(0) jeder andere Wert, nur keine Null (andernfalls dient 0 als Kennzeichen, daß sich noch keine Datei im Computer befindet!). Dieser Ursprungszustand wurde in Zeile

110 dokumentiert: W\$(0) = "0".

Es läßt sich leicht abfragen:

```
340 IF W$(0) <> "0" THEN 400
```

Wenn also unser KI-Programm nicht nach Zeile 400 (Unterschied U/R) springt, weil in W\$(0) noch kein Wert steht, muß der C 64 bereits seine erste Lektion über sich ergehen lassen:

```
350 PRINT "ICH KENNE NOCH KEIN TIER, AN WELCHES HAST DU GEDACHT"  
360 INPUT T$  
370 W$(0) = "2"  
380 W$(1) = "R" + T$  
390 GOTO 970
```

Lesern, die unseren Kurs aufmerksam verfolgt haben, ist bestimmt aufgefallen, daß die Frage in Zeile 350 schon einmal in unse-

Tiere raten (Variablenliste)

Name	Funktion
W\$(0)	Bezeichnung des Arrays (Datenfelds), in dem das erlernte Wissen gespeichert ist
AS	Antwort auf die Frage JA/NEIN
BS	Gegenteil von AS
T\$	Tiername, den sich der Spieler ausgedacht hat
FS	Gegenteil von T\$, die den Unterschied zwischen falschergerätem und gedachtem Tier beschreibt
US()	Ersatzvariable für W\$(M) zur professionelleren Bearbeitung der Unterschiedsfrage
M	Markierungszahl, die hinter J und N steht
W	Anzahl der belegten Speicherplätze, steht in W\$(0)
Z	Zählvariable bei Schleifen

motzte Textausgabe usw. Der Autor hat z.B. absichtlich jeden Ausgabertext auf dem Bildschirm stehen lassen, damit man den Programmablauf Schritt für Schritt nachvollziehen kann.

KI – Realität oder Utopie?

Unzählige Science-fiction-Stories erzählen von Computern, die intelligent sind, sprechen und vor allem lernen können. Bis heute ist die Diskussion darüber nicht verstummt, ob's in naher Zukunft tatsächlich intelligente Maschinen geben wird, die nicht nur stur ein Programm abspulen, sondern Schritt für Schritt neues Wissen anhäufen, einordnen und vor allem auf Dauer behalten können. Die Verfechter dieser Theorie verweisen stolz auf erste bescheidene Erfolge, die man bei Versuchen in den USA erzielt hat. Von den Skeptikern wird "AI" (Artificial Intelligence) jedoch als unwichtige Spielerei abgetan.

Wir haben in unseren drei Kursfolgen ein Programm entwickelt, das man getrost in die Rubrik "Selbstlernende Software" einstufen darf. Schon heute gibt's (z.B. bei Datenbanken) ausgefeilte Suchsysteme und Auskunftsroutinen, die exakt diesem Lernprinzip folgen (man nennt sie deshalb "Expertensysteme").

Lassen Sie sich nicht daran hindern, unser Demo-Programm so umzugestalten, daß man es beispielsweise zur Ausgabe technischer oder medizinischer Diagnosen verwenden kann – Ihrer Phantasie sind keine Grenzen gesetzt! Denken Sie daran, daß man nur durch Fragen lernt – es gibt also keine dummen Fragen, sondern nur Dumme, die niemals fragen!

Dr. Ing. Helmuth Hauck/bl

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.64ER-ONLINE.DE

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.64ER-ONLINE.DE

lt.
er-
ld-
rt-
Ort
en.
en-

Man monatlich 30 Mark muß man für einen "richtigen" Internet-Zugang, der auch "FTP" und "Telnet" erlaubt (s. Ausgabe 8), schon rechnen. Glücklicherweise gibt es Mail-Server, die Ihre elektronische Post (EMail) bearbeiten und analysieren und anschließend bestimmte Kommandos ausführen, sprich: Ihnen ein Programm per Mail zuschicken.

Das Grundprinzip

Nehmen wir an, Sie benötigen die neueste Demoversion von "64NET", die (das wissen Sie vom deutschen Vertreter) auf einem Rechner in Finnland gespeichert ist. Die Internet-Adresse dieses Rechners kennen Sie auch: "ftp.funet.fi". Wenn Sie nun einen "echten" Internet-Zugang hätten, bräuchten Sie sich nur in Ihre Mailbox einzuloggen und mit dem Kommando "ftp ftp.funet.fi" eine Verbindung zum finnischen Rechner aufzubauen. Doch auch per normaler EMail funktioniert das Verfahren: Sie schicken einem bestimmten Rechner ("Mailserver") einen Brief, dessen Text aus Kommandos besteht, die der Server interpretiert und ausführt. Es handelt sich dabei (in der Regel) um genau dieselben Kommandos, die Sie online eingeben müßten, wenn Sie sich direkt in den FTP-Rechner einloggen würden.

So geht's

Nehmen wir an, Sie wüßten nur, daß das Programm "64net" heißt und irgendwo auf einem Internet-Rechner auf Sie wartet.

Erster Schritt:

Post an "Archie" – er muß Ihnen helfen, das File ausfindig zu machen. Wenn Sie zunächst eine Anleitung benötigen, brauchen Sie nur das Wort "HELP" im Text der Nachricht unterzubringen.

Um dann z.B. das File "64net" suchen zu lassen, müßte folgender Text an Archie geschickt werden: set search sub
prog 64net

Voraussetzungen für FTP per EMail

- Das Mailboxnetz, in dem Sie User sind, muß:
- einen Übergang zum Internet besitzen (das ist fast immer gegeben)
 - dieser Übergang muß in beiden Richtungen funktionieren
 - das Mailboxnetz muß den Versand von Programmen als Mail ("uucode") unterstützen, anderenfalls können Sie nur Textfiles (Quellcodes usw.) anfordern

Internet: Dateien per Post

Programme à la carte

Nicht jeder C-64-Fan kann oder will sich einen Internet-Vollzugang leisten. Briefe ins Internet zu verschicken, ist jedoch auch von vielen Hobbynetzen aus möglich. Wir zeigen, wie Sie sich Programme per elektronischer Post angeln können.

Die erste Zeile stellt den Suchmodus so ein, daß auch "Unterstrings" gesucht werden (falls Sie den Programm-Namen nicht exakt kennen).

Als Antwort erhalten Sie (wieder per Mail) den Namen des FTP-Rechners (hier ftp.funet.fi), das Unterverzeichnis und den korrekten Namen der Datei.

Zweiter Schritt:

Mit den kompletten Angaben können Sie jetzt den Mailserver

z.B. maximal 8 KByte große Nachrichten erlaubt (wenn auch technisch mehr machbar wäre – man bedenke aber, daß der Mailversand in einem Hobbynet auf Sysop-Kosten geht!).

Interessante Texte

Nicht nur zur Software-Beschaffung ist Internet-EMail geeignet. Genauso gut ist es möglich, sich zur Teilnahme an News-

```

= Window Dial Terminal Edit View Back Macro Options [01:02] 19:32:14am
lruxrux 1 managers funstaff 45 Apr 26 15:40 c64s09k.readme -> /pub/msd
sdos/interfaces/emulator/c64s09k.readme
lruxrux 1 news ftp 42 Apr 28 17:45 c64s09c.exe -> /pub/msd
s/interfaces/emulator/c64s09c.exe
lruxr 1 albert ftp 21738 Mar 10 09:37 c64trans.zoo
lruxrux 1 managers funstaff 42 Apr 26 15:40 c64ut15.zip -> /pub/msd
s/interfaces/emulator/c64ut15.zip
lruxrux 1 managers funstaff 42 Apr 26 15:40 copy2d64.arj -> /pub/msd
s/utilities/diskutil/copy2d64.arj
lruxrux 1 news ftp 42 May 12 13:23 copy2d64.readme -> /pub/
msdos/utilities/diskutil/copy2d64.dsc
lruxrux 1 albert ftp 113240 Mar 10 08:35 d64.lha
lruxrux 1 albert ftp 16079 Mar 10 08:41 dip.d64.zip
lruxrux 1 news ftp 26 May 12 20:29 mac64-04.hqx -> /pub/mac
/app1/mac64 04.hqx
lruxrux 1 managers funstaff 37 Apr 26 15:40 x64 0.2.1.readme -> ...
./X11/contrib/x64-0.2.1.readme
lruxrux 1 managers funstaff 37 Apr 26 15:40 x64 0.2.1.tar.gz -> ...
./X11/contrib/x64-0.2.1.tar.gz
lruxrux 1 managers funstaff 41 Apr 26 15:40 zip2d64.arj -> /pub/msd
s/utilities/diskutil/zip2d64.arj
nie.funet.fi:/pub/cbm/c64/emulation
ncftp)
134.1p ANSI 38400N1.2 /cubenet

```

Wir haben uns über die Münchner "Cubenet" in den FTP-Server ftp.funet.fi eingeloggt.

überraschen. So könnte der Text der Message aussehen:

```

connect ftp.funet.fi
cd /cbm/c64/util
chunksize 8000
binary
uucode
get 64net

```

Der Server wird zunächst eine Verbindung aufbauen, dann ins entsprechende Unterverzeichnis wechseln, in den Binär-(Programm-)Modus wechseln und die Datei "64net" codiert in reinem Text an Sie schicken. Beachten Sie bitte, daß einige Hobby-Netze das Versenden codierter Binärfiles nicht erlauben! In diesem Fall müssen Sie sich bei FTP via EMail auf echte Textfiles beschränken. Mit "chunksize" wird übrigens die Größe eines Datenpakets eingestellt, im Fido-Netz

groups oder Newsletters anzumelden. Man sendet dazu eine Nachricht an einen entsprechenden Server mit dem Stichwort SUBSCRIBE und bekommt dann regelmäßig die Beiträge zugesandt. Die elektronische Zeitschrift "Commodore Hacking" (eine Fundgrube für Programmierer) erscheint

Für C-64-/C-128-Besitzer interessante FTP-Sites

FTP (Dateitransfer):

- ftp.funet.fi (alles)
- cnga.waterloo.ca (Demos, Spiele, Betriebssysteme u.a.)
- uceng.uc.edu (CP/M für C64 und C128)
- wilbur.stanford.edu (Mid-Software C64)
- bert.psyc.uepi.ca (Demos)
- ftp.hrz.uni-kassel.de (VC-20-Software)

z.B. in dieser Weise, außerdem gibt's interessante Newsletter u.a. für C-65-Besitzer usw.

Und die Kosten?

Die hängen völlig von der Mailbox ab, in der Sie vertreten sind. Oft wird nach übertragenen KByte abgerechnet, in anderen Boxen ist EMail bis zu einem Maximalbetrag (z.B. 1 MByte / Monat) kostenlos.

Für die Benutzung der FTP-Mailserver werden natürlich keine Kosten berechnet, aber selbst ein durchschnittliches Directory kann schon auf einige MByte Dateigröße kommen.

Compuserve-User müssen allerdings für jede aus dem Internet kommende Nachricht selbst zahlen (an Stelle des Absenders!).

Matthias Matting

Einige Mail- und Archie-Server

FTP-Mailserver:

ftpmail decwrl.dec.com (ftp)
ftpmail grasping.insa-lyon.fr (ftp)
BITFTP vm.gmd.de (ftp)
mserv x400.ieunet.ie (ftp)

Archie-Server:

archie archie.funet.fi (archie)
archie doc.ic.ac.uk (archie)
archie archie.th-darmstadt.de (archie)

Beachten Sie, daß fast immer Hochschulrechner genutzt werden, auf denen natürlich tagsüber auch andere Arbeiten laufen.

Mail-Adressierung ins Internet

Falls Sie nicht direkt ins Internet schreiben können, müssen Sie folgende Adressierung verwenden:

Fido:

Adresse: 2:242/6.1
Username: UUCP
1. Zeile der Nachricht:
Internet-Adresse
(max. 8 KByte, nicht codiert)

Compuserve:

der Internet-Adresse muß das Schlüsselwort INTERNET: vorangestellt werden (auch binäre Nachrichten problemlos)

Mausnetz:

Adresse wie im Internet (max. 16 KByte, aus Internet ins Mausnetz max. 48 KByte, binär problematisch und nicht erlaubt)

Z-Netz:

Adresse: Internet-Adresse (% statt @!), gefolgt von @UUCP.ZER
Bsp.: mmatting@cube.net@UUCP.ZER (max. 10 KByte, binär möglich, aber offiziell nicht erlaubt)

Btx:

verschiedene kommerzielle Anbieter (u.a. Rehbein) ermöglichen Nachrichtenversand und -empfang, jedoch nicht für FTP geeignet, da empfangene Mails in Btx-Seiten eingespiegelt werden. Die Preise sind z.Zt. noch hoch, sinnvolle Nachrichtenlänge unter 1 KByte.

SORRY, WERBUNG GESPERRT!

64ER ONLINE



WWW.64ER-ONLINE.DE

Assembler/Monitor 64

Hexbytes zum Schleuderpreis

Kennen Sie eines der am meisten verbreiteten Assembler-Tools des Commodore 64: "Profimat"? Unter anderer Bezeichnung hat es ein Software-Händler in England wieder aufgespürt!

**64'er
TEST**

1984 wurde das Maschinensprache-Entwicklungspaket von Data Becker, Düsseldorf, heraus-

gegeben. Es bestand aus einem Assembler- und einem Monitorprogramm. Damals mußte man 99 Mark dafür anlegen.

Inzwischen gibt's gute C-64-Assembler wie Sand am Meer – es war vor allem der Dumpingpreis von 9,90 Mark, der uns heiß machte, das Maschinensprache-System anzusehen.

Auf den ersten Blick nicht wegzuleugnen: modernere Assembler haben wesentlich mehr Funktionen und Pseudo-Opcodes als Assembler/Monitor 64. Doch gerade für Einsteiger in die Maschinensprache-Welt der Commodore-Processoren eignet sich das Programm optimal: die Bedienungsmodalitäten begreift man nämlich im Handumdrehen.

Neben den Mnemonics gibt's 17 Pseudo-Opcodes (sie werden

durch einen führenden Punkt eingeleitet), die den Quelltext bedeutend verständlicher gestalten. Dazu gehören so nützliche Funktionen wie .FILE- (lädt Quellcode-Dateien von Floppy oder Data-Sette), oder das Äquivalent der Basic-Anweisung IF (der Reaktionsbefehl THEN entfällt!).

OPTIONS ist der ausgeprägteste Pseudo-Opcode: er beeinflusst Ein-/Ausgabefunktionen, Makros usw.

Gemeinsam geht's besser!

Assembler und Monitor können gleichzeitig im Speicher und abwechselnd benutzt werden – der Assembler belegt das RAM von \$8000 (32768) bis \$9FFF (40959), der Maschinensprache-Monitor dagegen den Bereich zwischen \$C000 (49152) und \$CBFF (52223).

Der Quelltext (Source-Code) kann im Basic-Speicher Ausmaße bis zu maximal 30 KByte annehmen.

Im Monitorprogramm erleichtern 15 Kommandos (es sind die von anderen Monitoren bekannten einbuchstabigen Befehle) die Arbeit mit den Mnemonics. Mit BREAKPOINT (U) und QUICKTRACE (Q) lassen sich z.B. Teilbereiche im Source-Code testen, das WALK-Kommando (W) aktiviert den Einzelschrittbetrieb (Single-Step-Mode) des geplanten Pro-

grammprojekts. Die Anzeige der Register, der Flags und des aktuellen Befehls erfolgt dabei nach jedem Step.

Die BANK-Anweisung (B) wechselt die Speicherkonfiguration des C64 und kann auch den gesamten Speicher als RAM freigeben.

Das Handbuch gibt's nur in Englisch, ist dafür aber recht übersichtlich aufgebaut und mit vielen Beispielprogrammen (= Quelltexte) gespickt. Allein der Abschnitt über Makroprogrammierung beansprucht sechs Seiten.

Das Handbuch beschreibt lediglich die Bedienung der beiden Maschinensprache-Entwicklungs-Tools. Zum Lernen von Assembler reicht es aber nicht aus; die im Vorwort genannte "einschlägige Literatur" gibt's leider nur noch auf Computer-Flohmärkten oder wird in privaten Kleinanzeigen angepriesen.

Toller Preissturz

Der damals recht teure Profimat (99 Mark) kostet in der englischen Fassung gerade noch den "Zehnt": 9,90 Mark.

Die Anschaffung lohnt sich deshalb für jeden, der einmal in die Welt der Maschinensprache hineinschnuppern will.

Andree Herman/bl

Assembler/Monitor 64 (Monitor)	
Befehl	Funktion
R	Anzeige des Statusregisters
M	Hexdump des Speicherinhalts
G	Maschinenprogramm starten
L	laden
S	speichern
D	Assembler-Listing auf Screen ausgeben
C	Speicherbereiche vergleichen
T	... verschieben
H	Byte-Folge im Programm suchen
F	Bereich mit angegebenem Wert füllen
B	Speicherkonfiguration wählen (Adresse 1)
W	Einzelschrittmodus
Q	Trace mit Breakpoint
U	Breakpoint setzen
X	Exit, zurück zum Basic 2.0

```
(C) 1984 FIRST PUBLISHING
10000-0000
NO ERRORS
READY.
NEW
READY.
10 SWS 32768
20 LDA #500
30 STA $D020
40 LDX #501
50 STX $D021
60 STX $D286
70 LDA $01
80 PHA $40
90 ORA $40
RUN
*** ASSEMBLER 64 V2.0 ***
(C) 1984 FIRST PUBLISHING
20000-C012
NO ERRORS
READY.
```

Die Umwandlung in den Objektcode leitet man per RUN ein

Assembler/Monitor 64 (Pseudo-Opcodes)

Befehl	Funktion
.BYTE	8-Bit-Hexzahl positioniert einen Bytewert an der vorgesehenen Stelle, z.B. .BYTE \$2C
.WORD	16-Bit-Hexzahl belegt einen 16-Bit-Wert, z.B. WORD \$C000
.FILE	Laufwerksnr., "Name" lädt den Quelltext "Name" von Disk oder Tape
.IF Ausdruck..	GOTO Zeilennummer verzweigt, wenn der Ausdruck wahr ist
.GOTO	Zeilennummer unmittelbarer Sprung
.GTB	Abkürzung für Go To Basic: Rückkehr zum Basic 2.0 des C 64
.ASC "text"	verankert die ASCII-Werte der Zeichenkette "Text" an entsprechender Stelle im Objektcode
.SYS Ausdruck	ruft andere Maschinensprache-Programme während der Assemblierung auf
.STM Ausdruck	schafft mehr Platz im Speicher für die Symboltabelle
.SST	Gerätenr., Sekundäradresse, "Filename" speichert die Symboltabelle aufs angegebene Gerät (normalerweise Disklaufwerk)
.LST	Gerätenr., Sekundäradresse, "Filename" Symboltabelle laden
.FLP	Ausdruck integriert Fließkomma-Konstanten im Objektcode
.END (device, "filename")	Kennzeichen für den Quelltext (unbedingt angeben!)
.SYM	bringt eine Liste aller verwendeten Symbole und ihrer Werte
.PAGE	Länge, linker Rand formatiert das Assembler-Listing
.TITLE "text"	Standardtext einfügen
.OPT Optionen, ...	diverse Arbeitsanweisungen an den Assembler (Ausgabe auf Drucker oder Disk, Makros usw.)

64'er-Wertung:
Assembler/Monitor 64

Einsteiger-Modell eines Maschinensprache-Entwicklungspakets, das früher unter der Bezeichnung "Profimat" vertrieben wurde (Data Becker). Unkomplizierte Programmbedienung.

Positiv

- günstiger Preis
- simples Handling
- unterstützt Floppy und Data-Sette
- Programme nicht kopiergeschützt (eigene Sicherheitskopien möglich!)

Negativ

- Handbuch in Englisch
- spärliche Unterstützung von Disk- und Druckoperationen
- Kopierschutz (keine eigene Sicherheitskopie möglich!)

Wichtige Daten

Testkonfiguration: C 64 (Brotkasten), Floppy 1541-II
Bezugsquelle: T.S. Datensysteme, Denisstr. 45, 90255 Nürnberg, Tel. 09 11/28 82 86, Fax 09 11/26 89 73
Preis: 9,90 Mark

Beurteilung

Funktionen: ++
Bedienung: +++
Dokumentation: +
Preis/Leistung: +++

64'er COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64'er« bietet allen Computerfans die Gelegenheit, für nur 5,- DM eine private Kleinanzeige mit bis zu 4 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **November-Ausgabe** (erscheint am 21.10.94): Schicken Sie Ihren Anzeigentext bis 13. September (Eingangsdatum beim Verlag) an »64'er«. Später eingehende Aufträge werden in der **Dezember-Ausgabe** (erscheint am 25.11.94) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte auf dem Mittelhefter.

Bitte beachten Sie: Ihr Anzeigentext darf maximal 4 Zeilen mit je 40 Buchstaben betragen.

Schicken Sie uns DM 5,- als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« z. Preis von DM 12,- je Zeile Text veröffentlicht.

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

SORRY, WERBUNG GESPERRT!

64ER ONLINE



WWW.64ER-ONLINE.DE

Programmier-Tools

Helfer im Triple-Pack

Programmierer, die den Basic-Assembler (10/93) nutzen oder mit dem VIS-ASS (3/92) in Maschinensprache arbeiten, bekommen in dieser Ausgabe drei tolle Tools, die ihre Arbeit unterstützen.

Basic-Assembler-Converter

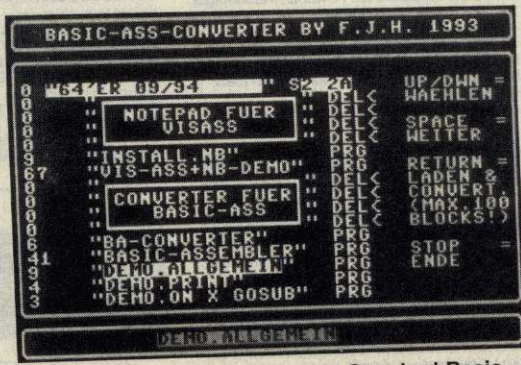
Mit diesem Konverter kann man Standard-Basic-Texte in das Format des Basic-Assemblers (10/93) übersetzen und dort weiterverarbeiten. Nach dem Laden mit:

```
LOAD"BA-CONVERTER",8,1
```

und dem Start mit dem RUN-Befehl erscheint ein einfaches Menü. Im unteren Teil befindet sich eine Statuszeile, die angibt, was zu tun ist.

Die Diskette mit dem zu konvertierenden Programm wird ins Laufwerk 8 eingelegt und per Tastendruck das Directory ins Menü geholt. Mit den Cursor-Tasten für Auf und Ab kann das gewünschte Programm gewählt werden. Bestätigt wird mit RETURN. Nun wird das Basic-Listing geladen und konvertiert. Dabei sollte man darauf achten, daß das zu wandelnde Programm nicht länger als 100 Block ist, da es sonst zum Konflikt mit dem Konverter bzw. dem Basic-Assembler kommt. Außerdem prüft das Tool nicht, ob es sich wirklich um ein Basic-Listing handelt! Mit der STOP-Taste wird die Directory-Anzeige bzw. der Konverter verlassen.

Nachdem das Programm geladen ist, erfolgt die Konvertierung und die Speicherung auf Disk mit dem Prefix "BA.". Das konvertierte Listing kann jetzt direkt in den Basic-Assembler (Programm plus Demos befinden sich ebenfalls auf Disk), bearbeitet und "assembliert" werden. Probleme kann es bei einigen PRINT-Anweisungen geben. So z.B. wenn Leerzeichen bei "HALLO" ausgegeben werden sollen.

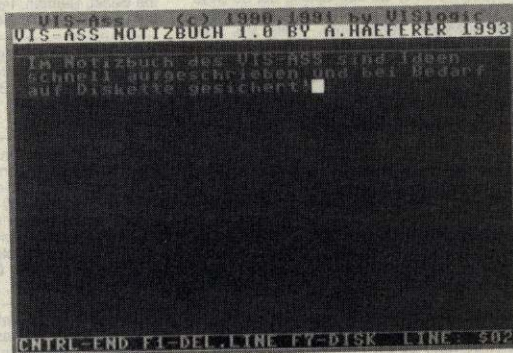


BA-Converter: Komfortabel vom Standard-Basic zum Basic-Assembler

len. Steht das Anführungszeichen nicht in derselben Zeile, wie das erste, werden die Leerzeichen zwischen dem rechten Rand und dem letzten Zeichen nicht gespeichert! In solchen Fällen überstehende SPACE-Zeichen mit SHIFT + SPACE ersetzen oder die Zeile auf mehrere Anweisungen verteilen. Außerdem müssen die Anführungszeichen immer paarweise vorhanden sein, da es sonst Probleme beim "Assemblieren" gibt! Franz-Johann Heuwing/lb

VIS-Ass-Notizbuch

Bei größeren Projekten sind kleinere Notizen sehr wichtig. Man kann sich seine Ideen auf ein Blatt Papier notieren oder ein Notepad auf dem Computerbildschirm nutzen. Mit dem Tool VIS-Ass-Notizbuch ist das kein Problem, denn mit der zusätzlichen Funktion kann man bis zu 120 Zeilen notieren und auf Diskette sichern. Natürlich kann auch wieder laden und auf dem Bildschirm betrachten. Auf unserer Programmservice-Diskette befindet

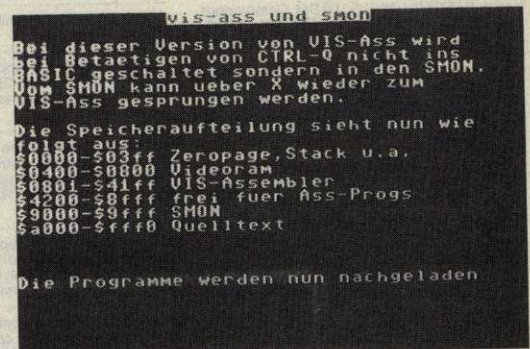


VIS-Ass-Notizbuch: Wichtige Anmerkungen auf dem Bildschirm eingetippt

sich eine fertig insallierte VIS-Ass-Version inklusive Notepad. Außerdem existiert der Installer auf Disk. Um den VIS-Ass um das Notizbuch zu erweitern, muß die Grundversion nur geladen werden. Anschließend kann man das Installprogramm mit "LOAD"INSTALL.NB",8,1 direkt in den Speicher holen. Nun muß man eine Diskette zum Sichern des neuen VIS-Ass einlegen und kann die Installation mit SYS 49152 starten. Der Assembler wird nun unter dem Namen "VIS-NB" auf Diskette gesichert. Er steht nun zur Arbeit zur Verfügung und kann wie gewohnt geladen und gestartet werden. Es ist jederzeit möglich, das Notizbuch mit COMMODORE/CTRL, mit Ausnahme bei den Diskettenoperationen, aufzurufen. Verlassen wird das Tool mit der CONTROL-Taste. Mit der Kombination SHIFT/CLR erfolgt nach einer Sicherheitsabfrage ein Löschen des Bildschirms. Mit der Funktionstaste F7 gelangt man ins Disketten-Menü

Hinweis: Durch den neuen Screen-RAM-Start muß beim Laden von altem Quellcode (Startadresse \$4200) unbedingt mit "SCR-RAM laden" gearbeitet werden, da sonst ein Absturz vorprogrammiert ist. Außerdem kann das Notizbuch nach dem Laden "Zeichenmüll" enthalten. Einfach nur den Bildschirm löschen oder eine Notiz laden.

Außerdem ist im Notizbuch ein Screen-Blanker integriert, der sich nach einer Minute aktiviert, wenn keine Taste betätigt wurde. Der Bildschirm wird abgeschaltet und wird erst nach einem Tastendruck wieder aktiviert. Programmierer, die VIS-Ass noch weiter ergänzen wollen, können frei auf das Programm zugreifen, da keinerlei Routinen des Assemblers benutzt



VIS-Ass-Notizbuch: Wichtige Anmerkungen auf dem Bildschirm eingetippt

wurden. Nur die IRQ-Routine wurde lediglich um die Tastatur-Abfrage zur Aktivierung des Notizbuches erweitert.

Andreas Häferer/lb

VIS-Ass und SMON

Um Maschinenprogramme näher unter die Lupe zu nehmen, nutzt man in den meisten Fällen einen Monitor.

Wer Besitzer eines Multifunktionsmoduls ist, kann den dort integrierten Monitor nutzen, um auf Speicherjagd zu gehen. Sonst muß ein Software-Monitor her. Hier bietet sich der Maschinensprache-Monitor SMON an. Er kann leicht modifiziert und an die unterschiedlichsten Speicherbereiche angepaßt werden. Wer aber den Assembler VIS-Ass und einen Monitor nutzen will, muß beide immer wieder wechselnd starten. Unser drittes Programmier-Tool verbindet den Assembler mit dem Maschinensprache-Monitor SMON. Per Patch werden der Monitor und der Assembler modifiziert und man kann zwischen beiden Tools hin- und herschalten. Auf unserer Diskette finden Sie den SMON in der Version ab

und kann Texte sichern bzw. laden. Die Notizen werden unter dem Namen "NOTIZ" auf Diskette abgelegt und sind 20 Block lang. Zum Abschluß noch ein

Die Speicherbelegung:	
\$4200-\$49ff	Notizbuch
\$4a00-\$dfff	neues VIS-Ass-Screen-RAM
\$e000-\$f300	Textspeicher
\$f400-\$ffff	Puffer für Zero-Page, Screen etc.

1541 liest und schreibt DOS-Disketten

"1541-Laufwerke können DOS-Disketten weder lesen noch schreiben", war lange Zeit unsere Standard-Antwort auf entsprechende Leserfragen. Um so überraschter waren wir, auf einem Internet-FTP-Server eine Freeware-Datei "1541DOS" zu finden, in deren Dokumentation eine kleine Hardwareänderung beschrieben wird, mit der auch die weit verbreitete 1541 Zugang zu MFM-formatierten Disketten findet. Die Beschreibung des Umbaus und die nötige Software wollten wir Ihnen auf keinen Fall vorenthalten, deshalb haben wir Platz auf der Programmdiskette dafür geopfert. Allein die Dokumentation ist 176 Blöcke groß!

Die Erfinder des kleinen Umbaus warnen in der Anleitung ausdrücklich, daß er nur von jemand mit Löterfahrung vorgenommen werden sollte. Zwar ist nicht viel zu löten, aber an diffizilen Stellen.

Das kann die Floppy nach dem Umbau: Lesen von einseitigen DOS-Disketten mit doppelter Dichte (180 KByte). Dieses Format sollte jeder PC mit 5,25-Zoll-Laufwerk beherrschen. Beim Schreiben von einseitigen Disketten mit einfacher Dichte (90 KByte) wird's komplizierter, da ein spezielles MFM-Format verwendet wird, das nur mit dem (ebenfalls mitgelieferten) "READ90K" vom PC gelesen wird.

Was ist zu verändern: 1. Möglichkeit: ungenutzten I/O-Port der VIA 6522 mit zwei ungenutzten Eingängen des Zählerbausteins 74LS193 verbinden. Nachteil: Fastload mit Parallelkabel nicht mehr möglich.

2. Möglichkeit: Es wird Pin CA1 der ersten VIA genutzt, dieser muß allerdings noch (mit einem ebenfalls ungenutzten Inverter eines 7406) invertiert werden.

3. Möglichkeit: Es wird ein Umschalter zwischen "normale 1541-Diskette" und "MFM-Diskette" genutzt.

In dem (gepackten) Programmfile auf unserer Diskette finden Sie auch ein Schema als GIF-Bild. Die Anleitung selbst ist allerdings komplett in Englisch!

Das Paket ist eine echte Gemeinschaftsleistung – Hardware von Daniel Fleming, Software von Richard Hable und DOS-Programm von Marko Makela. Anfragen können Sie an die im Programm genannte Internet-Adresse richten.

Utilities

Packer-Pack

Um den knappen Platz auf einer Diskette richtig auszunutzen, werden Programme oft komprimiert. Um die Files zu "quetschen", benutzt man Pack-Programme. Vier Tools dieser Art bietet unser "Packer-Pack".

Die vier Programme unseres Packs arbeiten nach verschiedenen Schemen.

Sir Compact III

Mit "Sir Compact III" können Sie Basic- und Assembler-Pro-

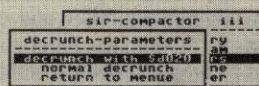
gramme bis um mehr als die Hälfte verdichtet werden. Das Tool gehört zu den Bit-Kompaktoren und wird mit:

```
LOAD"SIR-COMPACT III",8
```

geladen und mit dem RUN-Befehl gestartet. Im nun erscheinenden Menü kann mit den Cursor-Tasten

Tabelle 1: Sir Compact III

Befehl	Wirkung
SHOW DIRECTORY	Diskettenverzeichnis seitenweise anzeigen, <i>SPACE</i> blättert im Directory und <i>RETURN</i> bricht ab
CRUNCH PROGRAM	Packen eines Programms. Alle relevanten Parameter für das Packen finden Sie in Tabelle 2. Nach der Eingabe der Parameter folgt die Eingabe des Wertes für Speicherstelle 1. In den meisten Fällen beträgt der Wert 55 (hex.\$37). Nähere Informationen zur Belegung der Speicherstelle 1 und ihrer Funktion finden Sie im Artikel "C 64 inside" in dieser Ausgabe. Der <i>Override</i> -Wert legt fest, wo das Programm nach dem Entpacken im Speicher steht. Sind alle Werte korrekt eingegeben, kann man den Packvorgang mit <i>Weiter</i> starten. Das zu packende Programm wird geladen, gepackt und der Vorgang kann durch die Statuszeile beobachtet werden. Das gepackte File bekommt den Nachsatz <i>SIR</i> . Nach dem Packen kann man das Tool verlassen oder einen neuen Komprimierungsvorgang starten.
SET PARAMETERS	Einstellung des Bildschirmeffekts (Voreinstellung \$d020). Soll der Bildschirm beim Entpacken ohne Effekt erscheinen, muß die Option "normal decrunch" ausgewählt werden.
SET BASIC-LINE	Dient zum Anhängen eines Kommentars (max. 15 Zeichen) in der Basic-Zeile des gepackten Programms.
LEAVE CRUNCHER	Nach Sicherheitsabfrage verlassen des Programms mit einem Reset.



Basic-/Assembler-Programme lassen sich mit Sir Compact zusammenquetschen

auf- und abwärts gescrollt und mit der RETURN-Taste ausgewählt werden. In Tabelle 1 finden Sie alle Menüpunkte erklärt.

Christian Dombacher/lb

Bitstreamer

Der "Bitstreamer 1.01" ist ein Cruncher, er wird mit:

```
LOAD"BITSTREAM*",8,1
```

geladen und gestartet.

Tabelle 2: Pack-Parameter Sir Compact III

start with abs.sys: Programmstart auf absolute Adresse einstellen
start with imm.sys: Programmstart im indirekten Sprung
Fuse run statement: Wichtig zum Packen von Basic-Programmen, wobei das zu packende Basic-File nicht länger als 70 Block sein darf. Eventuell mit einem anderen Packer vorpacken. Hilfreich bei kritischen Files ist die Anwendung der absoluten Einsprungadresse der Basic-Kopfzeile (z.B. SYS 2061, hex.\$0811), anstelle der *use run statement*-Option bzw. der Jump in die Basic-Schleife (hex.\$a7ae).
expert/ger.version: Programm setzt die richtige Adresse eines mit der Expert-Cartridge (Multifunktionsmodul von Jahn-Technik) eingefrorenen Files ein.
expert/eng.version: Wie oben, aber für den Mode von Trilogic.
don't start program: File wird nur entpackt und nicht gestartet.

\$9000 und eine geänderte Version des VIS-Ass. Wenn Sie das Ladeprogramm:

```
LOAD"VISMON",8,1
```

im Speicher installieren, werden beide Tools nachgeladen. Nun kann man mit der Tastenkombination *CONTROL+Q* vom VIS-Ass in den SMON springen und nach Lust und Laune den Speicher untersuchen.

Wird der Monitor mit dem Befehl X wieder verlassen, gelangt man in den Text-Editor des VIS-Ass zurück.

Unbedingt zu beachten beim Einsatz beider Programme ist nur der verminderte Speicher, da der SMON den Bereich ab \$9000 bis \$a000 belegt. lb

Nun folgen nacheinander einige Abfragen:

Source: Name Original-Programm

Target: Name des gecrunchten Programms

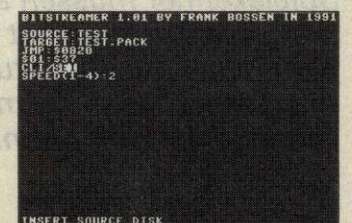
JMP:\$ Startadresse des zu packenden Programms

\$01: Wert der Adresse (s.o.) beim Starten

CLI/SEI: Wert des Interruptflag (S bzw. C drücken)

SPEED (1-4): Geschwindigkeit des Packvorgangs, wobei Qualität und Geschwindigkeit zueinander umgekehrt proportional sind

INSERT SOURCE DISK: Einlegen der Quelldiskette



Bitstreamer – klein und effektiv

Alle Eingaben erfolgen Hexadezimal. Nun wird das zu packende File geladen und bearbeitet. Falls ein Fehler entdeckt wird, kann mit der Taste *RUN/STOP* den Vorgang abgebrochen werden. Der Arbeitsablauf wird durch einen Zähler dokumentiert, der mit weiterem Programmablauf gegen Null geht. Beim Crunchen gibt's einen weiteren Zähler, der angibt, wie viele Bytes gewonnen wurden. Die Anzeige hat hexadezimale Schreibweise und ein vorgestelltes F zeigt negative Zahlen an (z.B. \$fff=-1). Nach dem Packvorgang erscheint die Meldung *INSERT TARGET DISK*, die auffordert die Zieldiskette zum Sichern des gecrunchten Files einzulegen. Frank Bossen/lb

SIR-Hufpack

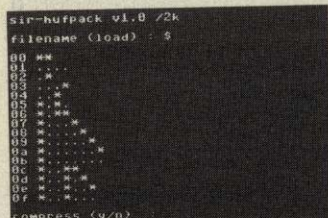
Das Tool entzieht Programmen die Luft per Huffman-Algorithmus. Es wird mit:

LOAD "SIR-HUFFPACK", 8, 1

geladen und mit dem RUN-Befehl gestartet. Das File kann den Adreßbereich von hex. \$0200 bis \$ffff einnehmen. Als erstes scannt der Packer das File und ermittelt die optimale Datenzusammensetzung. Die ermittelten Werte werden in Form einer Tabelle auf dem Bildschirm angezeigt. Häufige Nibbles (Halbytes) werden durch längere Bitwörter ersetzt. Je stärker die Längen der Bit-Wörter differieren, desto besser das Packergebnis. Im schlechtesten Fall gibt's nur Vierer-Kombinationen und der Packvorgang kann getrost vergessen werden. Stehen die Zeichen günstig für den Packvorgang, kann mit der Y-Taste bestätigt werden. Jetzt wird nur noch die Startadresse und die Prozessorstatus-Register eingegeben, dann

wird direkt von Disk gepackt. Nach Abschluß der Packerei wird nur noch der Name des neuen Files eingegeben und es auf Disk gesichert.

Christian Dombacher/lb



Huffmann-Verfahren in Action

SIR-Install

Packer auf PC und Amiga beherrschen das Verfahren des AR-CEN, das es ermöglicht, mehrere kleine Files zu einem großen Programm zusammenzufassen und zu packen. So lassen sich Programme mit mehreren Teilen gut archivieren und diskettensparend zusammenquetschen.

"Sir Install" erledigt diese Aufgabe auch auf dem C 64 und verwendet einen Sequenzpacker. Nach dem Laden mit LOAD "SIR-INSTALL", 8, 1 und dem Programmstart mit RUN, meldet sich das Tool mit "INSERT DISK!".

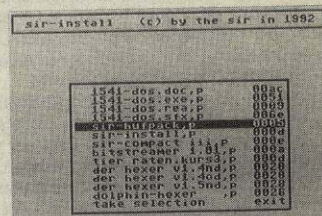
Nun muß nur die Diskette mit den zum Packen vorgesehenen Files eingelegt und der Einlesevorgang kann mit einer beliebigen Taste gestartet werden. Im nun erscheinenden Window lassen sich die gewünschten Programme mit den Cursor-Tasten wählen und mit RETURN bestätigen.

Die Wahl des Punktes "take selection exit" leitet den eigentlichen Vorgang ein. In der folgenden Eingabezeile kann noch eine kleine Notiz über die Programme verfaßt werden, die beim Entpacken gezeigt wird. Nun linkt das Tool alle Files und beginnt mit dem Packen.

Der Vorgang kann je nach Anzahl und Länge der Files einige Minuten bis zu einer halben Stun-

de dauern. Ist der C 64 fertig, kann der Filename des neuen Files in einer Inputzeile eingegeben werden und die Daten werden auf Disk geschrieben.

Zum Entpacken der Einzelteile wird das Pack-File einfach geladen, eine Diskette mit entsper-



Files gearct und gepackt ergeben ein File zur Ablage auf der Diskette

chend viel Speicherplatz ins Diskettenlaufwerk eingelegt und der Vorgang gestartet.

Der Rest des Pack-Arc-Verfahrens geschieht dann schließlich automatisch.

Christian Dombacher/lb

Action-Game "Robox"

Heiße Schlacht

Spielerkenner müssen aufpassen, unser Game in dieser Ausgabe hat nichts mit dem gleichnamigen Adventure zu tun. Es geht vielmehr mit einer Laser-Wumme bewaffnet gegen scheußliche Kreaturen.

An der Schwelle des 31. Jahrhunderts werden die Bewohner eines kleinen Außenpostens im Beta-System von einem Kampfraumschiff überfallen.

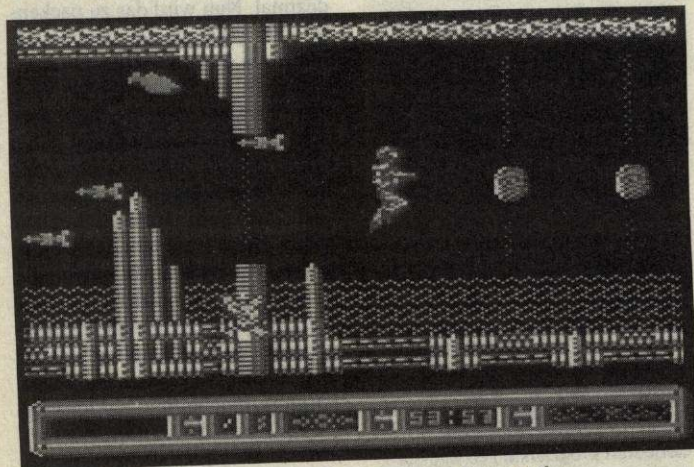
Die Plasma-Fresser betreten die ferne Welt und saugen die komplette Atmosphäre und das Wasser des Planeten ab. Der Große Rat entscheidet, daß man den Super-

kämpfer Robox wieder aktiviert und ihn gegen die Eindringlinge schickt. Der Roboter ist eine Maschine der Klasse C und schlechthin das Beste vom Besten.

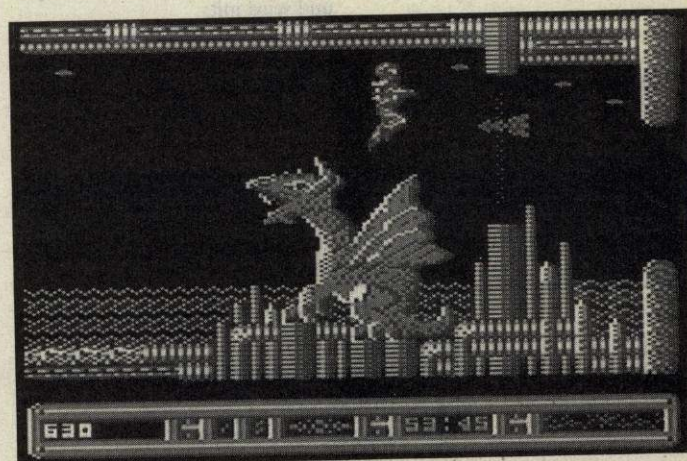
Also macht sich der Held auf und beginnt sein Abenteuer durch die Gänge des Planeten und muß dabei sehr vorsichtig sein. Die Gegner lauern an jeder Ecke und

Equipment mit Extras aufbessern kann.

Nach dem Laden mit "LOAD "ROBOX", 8, 1 und dem Start mit RUN erscheint das Titelbild, das mit der SPACE-Taste verlassen wird. Nun folgt man den Bildschirmanweisungen. Mit einem Joystick in Port #2 wird Robox gesteuert. Der Blechkamerad bal-



Unzählige Fallen und Hindernisse erwarten Robox, den Elitekämpfer mit dem Stahlanzug



Auf High-Tech und fiese Monster setzen die Plasma-Fresser, aber sie haben nicht mit Robox gerechnet

haben viele Fallen integriert. Kollisionen mit den Feinden und den Wänden führen zu Energie-Verlust. Berührt Robox eine Falle, folgt ein Game Over.

Außerdem stehen dem Blechkamerad für die einzelnen Level Zeitlimits zur Verfügung. Am Ende einer Stage folgt ein kleiner Bonuslevel, in dem der Held sein

lert, sobald der Feuerbutton betätigt wird. Verliert Robox sein Leben, erscheint "Game Over" auf dem Schirm.

Ein erneuter Start des Spiels erfolgt durch den Druck der Taste S. Also – das Lasergewehr scharf gemacht und den Aliens eins aufs Fell gebrannt! Viel Spaß dabei!

H. W. Müller/lb

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

Programmieren in Umgangssprache?

Interpreter: Cobol 64/128

Die Programmiersprache "Cobol" soll sich besonders gut für Computer-Laien eignen, da sich diese Programme durch die dem Englischen sehr ähnliche Syntax fast von selbst dokumentieren. Wir haben die Probe aufs Exempel gemacht.

64'er TEST

Cobol, die "Common Business Oriented Language", ist für geschäftliche Zwecke noch immer

eine der meistgenutzten Sprachen. Von der Vax bis zum PC gibt's Implementationen, und auch am C 64 bzw. C 128 kann man in Cobol programmieren. Seit der "Erfindung" 1959 hat sich natürlich einiges verändert, doch das Hauptmerkmal ist erhalten: Die Sprachsyntax ähnelt der der englischen Sprache sehr. Für die Basic-Zeile

A=B+C

würde man in Cobol schreiben
ADD A B GIVING C.

Die Ausstattung

Die uns zum Test vorliegende Software ist eigentlich schon (für C-64-Verhältnisse) uralte, denn sie stammt aus dem Jahre 1984 (Cobol 64) bzw. 1986 (Cobol 128). Trotzdem hat sie erst jetzt wieder den Sprung über den "großen Teich" geschafft. Dementsprechend gibt's auch kein deutsches, sondern ein englisches Handbuch, das aber die Grundlagen von Cobol (wenn auch ein bißchen zu theoretisch) sehr gut erklärt.

Das System besteht aus Editor, Interpreter und Debugger – das heißt auch, daß Sie Cobol-Programme nicht weitergeben können, denn zum Ausführen Ihrer Werke wird stets der Interpreter benötigt.

Die Cobol-128-Version hat einen Vorteil: Es müssen keine Programmteile nachgeladen werden, was die Arbeit mit dem Interpreter

```
READY.
SYS 2051

COBOL 64 VERSION 2.5
(C)1984 K.A.ALEXANDER/ABACUS
ENTER: 1=EDIT, 2=RUN, 3=DEBUG, 4=SAVE, 5=GET
6=NEW-PROG/EDIT, 7=CRUNCH, 8=PRINT-ON
9=PRINT-OFF, 10=NEW-NAME
6
**START NEW-PROG
**ENTER FILE NAME IN QUOTES
"MYTEST"

SEARCHING FOR COEDIT
**START EDIT8 $5C10
```

Nach dem Laden und Starten (per Sys-Befehl!) präsentiert sich das Hauptmenü mit Interpreter, Editor usw.

sehr beschleunigt. Trotzdem kann man jederzeit C-64- und C-128-Programme ineinander konvertieren. Es ist jedoch nicht möglich, mit Cobol 64 erzeugte Sourcecodes direkt unter Cobol 128 ablaufen zu lassen und umgekehrt.

Struktur von Cobol

Der C-64-Implementation (die dem ANSI-Standard von 1974 weitgehend entspricht) merkt man den Entstehungszeitpunkt von Cobol noch deutlich an. Fortran-Programmierer kennen das ebenfalls: Es ist **nicht** egal, auf welcher Position in einer Zeile ein Befehl steht. (Lochkarten lassen grüßen!) Deshalb sind dem "Editing Format" im Handbuch auch immerhin fünf Seiten gewidmet. In den Spalten 1 bis 6 findet man z.B. die "Sequence Area", sprich: die Zeilennummer (grundsätzlich sechs Ziffern). Daran schließt sich in Spalte 7 die "Indicator Area" an, die Symbole für Kommentare oder fortgesetzte Zeilen enthält. Unterprogramme, Programmteile usw. müssen in Spalte 8 bis 11 eingeleitet werden. Erst ab Spalte 12 darf schließlich der eigentliche Programmcode beginnen.

Die "Längsrichtung" eines Cobol-Programms ist genauso strikt gegliedert wie die "Querrichtung" (die Spalten): Es sind unbedingt die vier "Abteilungen" (Divisions) "Identification", "Environment", "Data" und "Procedure" zu unter-

scheiden. Jede dieser Abteilungen wiederum enthält Sektionen (Sections), die z.T. zwingend, z.T. freiwillig einzutragen sind.

In der "Identification"-Abteilung wird z.B. der Programmname festgelegt, der Programmierer genannt usw.

Die für die "Umgebung" (Environment) zuständige Abteilung enthält u.a. den Typ des Computers (Quelle und Ziel, hier jeweils C 64 oder C 128), diverse Ausnahmeregeln (hier können Sie z.B.

das Währungssymbol auswechseln) und Angaben über die benötigten Daten-Files (Laufwerk, Dateiname).

Die Daten-Abteilung ist logischerweise für die verwendeten Daten zuständig: Wie bei vielen anderen Programmiersprachen müssen die einzelnen Typen recht ausgiebig deklariert werden, in Cobol ist allerdings auch noch anzugeben, an welcher Stelle der Dezimalpunkt sitzen soll und wie genau die Zahl gerundet wird.

Last but not least beginnt in der "Procedure"-Abteilung das eigentliche Programm, das solange abgearbeitet wird, bis es auf eine STOP RUN-Anweisung trifft. Es stehen die wichtigsten Vergleichs- und Schleifenkonstruktionen zur Verfügung, jeweils in einem Exemplar (IF und LOOP), was die Auswahl erleichtert. Unterprogramme ("Paragraphs") können Sie mit PERFORM anspringen. Der größte Vorteil von Cobol, die Orientierung an der menschlichen Sprache, sollte sich eigentlich in diesem Programmteil offenbaren, aber der "Wortschatz" von Cobol ist so begrenzt, daß man sich die wenigen Befehle oftmals auch im Oktadezimalsystem merken könnte. Über arithmetische Grundbefehle (ADD, DIVIDE, MULTIPLY, SUBTRACT) kommt man nämlich kaum hinaus, und Möglichkeiten, Routinen aus dem Basic-Kernel anzuspringen, gibt's überhaupt nicht.

Zum Reinschnuppern

Auf Grund des doch sehr beschränkten Funktionsumfangs kann Cobol 64/128 wirklich nur echten Programmier-Freaks und künftigen Wirtschafts-Informatik-Studenten empfohlen werden, die einfach aus Interesse lernen wollen, wie Cobol funktioniert. Für ernsthafte Anwendungen ist selbst Basic V2 besser geeignet, zumal das Weitergeben von Cobol-Programmen nicht möglich ist. Es kommt hinzu, daß der Preis zwar im Vergleich zu PC-Cobol-Systemen sehr niedrig ist, aber für ein C-64-Programm mit 70 Mark inzwischen eher hoch gegriffen.

Matthias Matting



64'er-Wertung:

Cobol 64/128

Programmiersprache für C 64 bzw. C 128

Positiv

- 64er- und 128er-Version
- zum Reinschnuppern gut geeignet

Negativ

- ziemlich teuer
- komplett in englisch
- geringer Sprachumfang

Wichtige Daten

Bezugsquelle: MasterMMSoft, 79199 Burg am Wald
Preis: 69 Mark (C 64), 79 Mark (C 128)
Testkonfiguration: C 128D, 1581, 1571

Beurteilung

Funktionen: ++
Bedienung: ++
Dokumentation: +
Preis/Leistung: 0

befriedigend



Neues C-128-Kopier-Tool vom Hexer

Viele Leser haben uns in unzähligen Briefen und Anrufen gebeten, das beliebteste Kopierprogramm für den C 128 nochmals auf Diskette anzubieten. Voilà – auf unserer Diskette zum Heft finden Sie vier unterschiedliche Fassungen:

- **Hexer V1.4nd:** für C 128DCR (Blechgehäuse) mit integrierter Floppystation (DOS 3.1),
- **Hexer V1.4od:** für C 128D (Plastik, DOS 3.0),
- **Dolphin Hexer:** für C-128-User, dessen Floppy-DOS durch "Dolphin DOS" ersetzt wurde,
- **Hexer V1.5nd:** für Besitzer eines Single-C-128 und der Floppy 1571-II.

Alle Versionen erkennen und unterstützen ein RAM-Erweiterungssteckmodul im Expansionport: REU 1750 oder REU 1764.

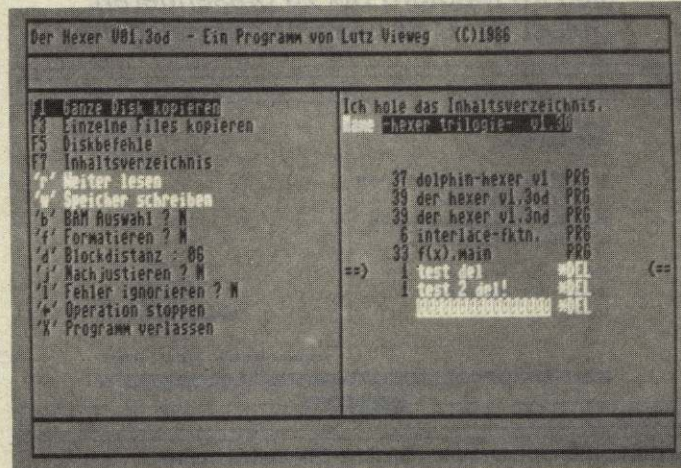
Nach dem Laden und Starten: RUN "DER HEXER V1.4ND" meldet sich das Kopier-Tool mit dem übersichtlichen Hauptmenü. Verwenden Sie zum Aufruf der Funktionen die angegebene Taste, per <SPACE> aktiviert man die Option.

F1 – Ganze Disk kopieren: ... überträgt den gesamten Inhalt einer Diskseite auf die andere. Das Programm erkennt automatisch, ob es sich um eine ein- oder doppelseitig beschriebene Disk handelt. Der Dolphin-Hexer testet zusätzlich, ob die Spuren 36 bis 40 beschrieben sind. Ohne REU sind bei einseitig formatierten Scheiben zwei Diskettenwechsel pro Diskseite nötig, bei einer doppelseitigen also vier. Anschließend springt der Auswahlbalken des Hauptmenüs automatisch zur Option "W - Speicher schreiben" (s. Beschreibung).

F3 – Einzelne Files kopieren: ... funktioniert wie jedes andere File-Copy: damit transferiert man gewünschte Dateien auf eine oder mehrere andere Disketten. Wenn das Inhaltsverzeichnis erscheint, wählt man die Files per <CRSR auf/ab> und markiert sie per Tipp auf <RETURN>. <SHIFT CLR/HOME> macht die Auswahl rückgängig, per <Pfeil links> bricht man die Funktion ab.

Die Dateien werden beim Markieren nummeriert und in dieser Reihenfolge auf der Zieldisk abgelegt. Sogar offiziell gelöschte Files lassen sich damit übertragen: Sie sind im Directory zwar weiß (= Dateityp DEL) gekennzeichnet, werden aber in der Zieldisk als normale PRG-Dateien eingetragen. Je nach Menge der ausgewählten Files können ohne REU natürlich auch dann mehrere Dis-

Das beste Kopierprogramm für den C 128 feiert Wiederauferstehung – für alle, die's noch nicht haben: "Der Hexer II", ergänzt um eine geänderte Version für die Floppy 1571-II.



Übersichtliche Menüauswahl per Tastendruck: Hexer V1.4

kettenwechsel notwendig sein, <SPACE> leitet den Kopiervorgang ein.

F5 – Diskbefehle: Man hat die Wahl zwischen *Formatieren* (ein- oder doppelseitig), *Files löschen* und *Files umbenennen*. Entsprechende Eingaben zu diesen Optionen sind im oberen Fenster zu erledigen (z.B. Diskname, ID, neue File-Bezeichnung usw.). Achtung: der Dolphin-Hexer erwartet die Befehle im DOS-Format der Floppy 1541 ohne OPEN- und CLOSE-Anweisungen (s. Floppy-Handbuch, z.B. "N: DISKNAME, S1" beim Formatieren oder "S: NAME" beim Löschen).

F7 – Inhaltsverzeichnis: ... entspricht der normalen Directory-Funktion des C 128. Den raschen Durchlauf der Liste stoppt man per <NO SCROLL>. Erfreulich: der Kopierspeicher wird davon nicht beeinflusst oder gar zerstört.

R – Weiter lesen: Läßt sich eine Diskseite oder eine umfangreiche File-Auswahl nicht auf einmal lesen und kopieren, steigt man auf diese Option um und macht jetzt mit dem Rest weiter.

W – Speicher schreiben: Befinden sich bereits die gesamte Diskseite oder ausgewählte Einzel-Files im Speicher (noch besser: in der REU), lassen sich diese Daten auf beliebig viele Zieldisketten übertragen. Erst erneutes

Einlesen einer Quelldisk (per <F1> oder <F3> überschreibt) diesen Datenpuffer wieder.

B – BAM-Auswahl: ... macht nur Sinn, wenn man eine ganze Diskseite per <F1> übertragen will. Oft befinden sich darauf noch jede Menge freier, unbelegter Sektoren, die dann beim Kopieren nicht berücksichtigt werden. Kritisch wird's allerdings bei Disketten, die viele "versteckte" Direktzugriffsblöcke enthalten, die in der BAM nicht eingetragen sind: dann sollte man auf diese Menüoption verzichten!

F – Formatieren: ... dient als Schalter zwischen "J" und "N" und teilt dem Programm mit, daß die jeweiligen Spuren und Sektoren vor dem Kopieren zusätzlich formatiert werden sollen (nur relevant bei Funktion F1, also "ganze Diskseite"). Wir empfehlen diese Funktion auf jeden Fall.

D – Blockdistanz: Damit legt man fest, welchen Abstand die einzelnen Sektoren beim Schreiben der Files bekommen sollen: die Floppy 1541 braucht stets den Interleave-Faktor (Sektorenversatz) "10", die Floppy 1571 arbeitet korrekt mit dem Wert "06".

J – nachjustieren: ... legt fest, ob ein zunächst fehlerhaft gelesener Block, jeweils um eine Halbspur nach oben oder unten versetzt, nochmals gelesen werden

soll. Das hat den Vorteil, daß sich auch Disketten kopieren lassen, die mit dejustierten und verstellten Schreib-Lese-Köpfen alter Laufwerke erzeugt wurden.

L – Fehler ignorieren: Wenn Sie diese Option aktivieren (J), macht der Hexer dennoch weiter, obwohl er einen Prüfsummenfehler gefunden hat. Lassen Sie sich nicht verunsichern – die Error-Meldung erscheint auf jeden Fall, aber der Sektor wird trotz Prüfsummenfehler unbeirrt übertragen! So hat der Hexer kein Problem mit mancher kopiergeschützter Software ...

<Pfeil links> – Operation stoppen: Nahezu jeden aktivierten Menüpunkt kann man damit verlassen, allerdings sind alle Daten im Pufferspeicher weg.

SHIFT X – Programm verlassen: ... bricht das Programm ab, Tipp auf <SPACE> = Reset.

Programmhinweise

... zur Version "Hexer 1.5": Dies ist eine gepatchte Fassung von Nr. V1.4, die speziell für den Single-C-128 mit externer Floppy 1571 gilt. Etwa im Sommer 1990 brachte Commodore eine neue 1571-Diskstation heraus, bei der etliche Adreßeinsparungen im Floppy-DOS geändert wurden. Herkömmliche Hexer-Versionen haben also keine Chance mit dieser Diskettenstation. Nicht vergessen: Verwenden Sie die Fassung 1.5 nur mit solchen Laufwerken!

Bei den neuen Hexer-Fassungen "1.4od" und "1.4nd" wurden die Farben für die Bildschirmausgabe manipuliert (schwarze Schrift, hellgrüner Hintergrund).

Möchten Sie das Programm wieder in den Einschaltzustand wie unmittelbar nach dem Laden versetzen, genügt ein Tipp auf <RESTORE>: Damit springt der Computer zur CBM-Kennung in Bank 1 und löst quasi einen Reset aus, der aber abgefangen wird. Der C 128 verzweigt zur Startadresse des Hexers und installiert ihn erneut. Das ist vor allem dann wichtig, wenn Sie die Floppystation zwischendurch aus- und wieder eingeschaltet haben.

Das Kopierprogramm verwendet den Burst-Modus der Floppy 1571. Die 1541-Laufwerke am Single-C-128 haben also keine Chance, mit dem Hexer in Kontakt zu treten. Dafür ist es dem Kopierprogramm aber vollkommen egal, ob man waschechte doppelseitige 1571-Scheiben mit C-128-Programmen oder einseitige 1541-Disketten kopiert.

Lutz Vieweg/bl

Auf den ersten Blick scheint es wirklich nicht einfach, in die exklusiven Reihen der Btx-Anbieter vorzustoßen. Während in der Mailbox-Szene immer ein freundlicher Sysop bereit ist, Neulingen die Arbeit mit und an einer Mailbox zu erklären, gähnt im Btx-Bereich die große Leere. Wer mit einem Programm ökonomischen Erfolg hat, gibt Geschäfts-Details nicht so gern weiter.

So gehen Sie vor

Wie stets, ist der Antrag aller Taten Anfang. Es ist ein "Auftrag im Bildschirmtext" zu stellen, der an den Geschäftskundenvertrieb Ihres örtlichen Fernmeldeamts (dort gibt's auch normalerweise die Formulare) oder direkt an das Fernmeldeamt 3 Berlin, Dienststelle TBtx, gerichtet wird. Eine regionale Leitseite genügt völlig, diese kostet wesentlich weniger als eine bundesweite und ist trotzdem von überall her zu erreichen. Ungefähr vierzehn Tage später ist Ihr Auftrag realisiert: Sie können Ihr Kürzel eingeben und landen daraufhin auf Ihrer Leitseite, die aber noch vollkommen leer und ohne jede Funktion ist. Bevor Sie jetzt an die Arbeit gehen, müssen Sie sich über die Struktur Ihres Programms klar werden. In maximal drei Auswahlritten sollte ein User die ihn interessierende Seite erreichen können. Als Btx-Nutzer wissen Sie ja, wie andere dies realisiert haben.

Der Dialogeditor

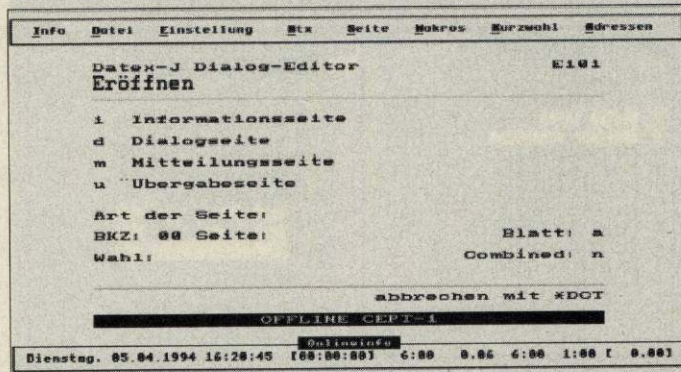
Auf Seite *910# finden Sie den Dialogeditor, mit dem Sie Btx-Seiten gestalten können. Sie müssen zunächst wählen, ob Sie Seiten eröffnen, ändern, löschen usw. wollen. Anschließend müssen Sie alle Angaben zu Seitentyp, Verknüpfungen, Gebühren usw. eigenhändig eingeben und schließlich auch die Seite editieren. Selbst wenn Sie rege von der Kopierfunktion Gebrauch machen, ist die Gestaltung eines kompletten Programms allerdings langwierig. An dieser Stelle kommt eine Software fürs Commodore-Btx-Modul zum Zuge. Der Anbieter Daniel A. Rehbein hat eine Sammlung leistungsfähiger Tools für den C 64 programmiert, die wir hier kurz beschreiben wollen.

Btx-Input: Man lädt dieses Programm als Zusatzsoftware ins Modul. Es ändert die Belegung der Tastenkombination F7 + Q (Abschalten des Moduls und Sprung ins Basic). Ruft man diese Funktion jetzt auf, muß man den Namen

Btx-Anbieter mit dem C 64

Treffpunkt Btx

Zwar steigen die Btx-Nutzerzahlen ständig, doch zugleich bröckelt die Zahl der Anbieter — obwohl es weder teuer noch kompliziert ist, sich im Medium Btx zu präsentieren. Selbst der C 64 kann da mitmischen!



Im Dialog-Editor bearbeiten Sie Ihre Btx-Seiten online

einer Datei eingeben, die dann 1:1 in den Btx-Dialogeditor übertragen wird. Sie können also beispielsweise eine Seite, die Sie offline im Capturepuffer editiert haben, senden und so Kosten sparen.

Btx-Edit: Die Weiterentwicklung von Btx-Input. Nun läßt sich auch eine neue Tastaturbelegung aktivieren, auf der die wichtigsten Ceptcodes gespeichert sind. Man kann die verschiedenen Zeichensätze und -größen, Blinkmodi etc. setzen und damit sein Programm besser gestalten.

Btx-Tas-Rehbein: Später entschloß sich Rehbein, seine Programme an Btx-Comfort, die Supersoftware für das Modul, anzupassen. Dieses File erzeugt einen Tastaturtreiber, der mit der Belegung bei Btx-Edit identisch ist.

Send: Dies ist ein Basicprogramm, das man lädt, nachdem man aus dem Modul in den 64er Modus gesprungen ist (F7 + 6). Mit diesem Programm kann man Dateien in den Dialogeditor einspielen. Die Programme Btx-Input bzw. Btx-Edit werden also nicht gebraucht.

Wichtige Anbieterseiten

- *92# Abrechnungsdaten
- *910# Anbieterbereich
- *104910# Informationen für Anbieter
- *10492# Inkassodaten
- *2000034345# Gebührenverzeichnis
- *2000054652431000# Anbieter-Handbuch

Wichtige Gebühren

- Anbieterberechtigung einmalig 50 Mark
- Leitseite regional inkl. einem Kürzel monatlich 57,50 Mark
- Einarbeiten je Seite 0,115 Mark
- Speichern je Seite und Tag regional 0,023 Mark
- Seitenabrufstatistik je Seite und Tag 0,575 Mark
- Inkasso der Vergütungen 23 Mark + 2 %

Kleines Glossar

- CEPT** Kurz für den Btx-Grafikstandard
- DRCS** Dynamically Redefinable Character Set (vom Anbieter definierbarer Zeichensatz)
- Bulk-Update** Einspielen der Btx-Seiten im Transparent-Modus (mit C 64 nicht möglich)

Select: Dieses Tool erzeugt eine Kopie eines anderen Files. Man bestimmt aber selbst, von welchem Byte bis zu welchem Byte kopiert werden soll.

Count: Damit kann man sich anzeigen lassen, wieviel Byte eine Datei hat. Das ist nützlich, um herauszufinden, ob die Datei nicht zu lang ist, da eine Btx-Seite nur ca. 1900 Bytes aufnehmen kann.

Show: Zeigt eine Datei Byte für Byte an. Man bekommt die Dezimalzahl des Bytes sowie das eigentliche ASCII-Zeichen angezeigt.

Get DRCS: Damit kann man aus einer Capturedfile die DRCS-Daten rippen. Sie werden als separates File gespeichert, das man auch einfach in den Dialogeditor einspielen kann.

DRCS-Split: Trennt ein Capturefile von den DRCS-Daten und speichert die DRCS-Daten in ein File und den Rest der Seite in ein anderes.

Grafik: Ein überdurchschnittlich leistungsfähiges Tool. Hiermit kann man einen Teil einer Bitmap in DRCS umrechnen und dann in dem Dialogeditor einspielen. Man gestaltet also die Grafik als Bitmap, rechnet sie dann in DRCS um und kann sie im Btx-Programm verwenden.

Telesoft: Hiermit kann man Programme in Btx-Seiten einspielen, die später als Telesoftware geladen werden sollen. Man kann selbst bestimmen, wieviel Bytes in der ersten TSW-Seite und in allen folgenden enthalten sein sollen.

Wenn man weiß, daß ein komplettes Editier-System mit Grafik-Editor, Telesoftware-Konverter usw. für PC mindestens 1500 Mark kostet, lernt man die Leistungsfähigkeit eines C 64 gleich doppelt zu schätzen.

Stefan Eberspächer / ma

Literatur:
Milczewsky/Pielhauer, Datex-J:
Das Praktiker-Handbuch,
Neue Mediengesellschaft Ulm, 1993

Vergleich Mailbox vs. Btx-Programm

Faktor	Mailbox	Btx-Programm
Hardware	C 64, Festplatte / RAM-Erweiterung, Modem	C 64, Modem
Kosten – Betreiber	Telefonanschlüsse	Leitseite, Speicher- und Überarbeitungsgebühren
Kosten – Anrufer	Ferntarif	Ortstarif + 2 Pf. / Seite + 2 Pf. /min.
Zeitaufwand	ungefähr gleich	(min. 30 min / Tag)
Wartung	ortsgebunden bzw. Fernwartung	nicht ortsgebunden
Online-Unterhaltung	möglich	nicht möglich
Software-Upload	jeder User	nur Berechtigte
Software-Download	jeder User	jeder User
Geschwindigkeit	modemabhängig	z. Zt. max. 2400 bps
Grafik	ANSI / CBM	CEPT

Laser Genius Assembler

Heißer Draht zur Maschinensprache

Irgendwann reicht's! Das karge und langsame Basic 2.0 des C 64 nervt. Gott sei Dank gibt es Alternativen.

**64'er
TEST**

Pascal, Cobol, Fortran, Prolog, C – die Computerwelt kennt viele ausgezeichnete

Programmiersprachen. Doch mit dem C 64 sind sie meist zu langsam oder der jeweilige Compiler hat sich vom Markt gestohlen. Was bleibt, ist die (zu Unrecht) gefürchtete Assembler-Programmierung. Doch mit dem richtigen Werkzeug ist Maschinensprache effektiver und hundertmal schneller als jede andere Programmiersprache. Zum Rüstzeug gehören ein Assemblerlehrbuch, Unterlagen über die Systemroutinen des

Assembler-Einsteiger ist, sondern zusätzliche Befehle beschrieben werden, die den Umgang mit Quell-Codes erleichtern sollen.

Die Programme des Entwicklungssystems (wahlweise auf Disk oder Tape) teilen sich die Benutzeroberfläche auf dem Screen. Der Editor arrangiert sich mit dem Assembler, das Maschinensprache-Monitorprogramm benutzt die Bildschirmmaske gemeinsam mit dem Analyser. Außerdem gibt's auf der Systemdisk ein Utility, das den Objekt-Code in ein ausführbares Programm verwandelt.

Editor und Assembler

Das Tool zur Eingabe des Quell-Codes gibt's in zwei Versionen: als "Low Memory"- (\$0800-\$47FF) und "High Memory"-Fassung (\$9000-\$CFFF). Beide nutzen auch die Zero-Page. Allein 26 Pseudo-Opcodes, zusätzlich zu den normalen Mnemonics, und die Verwendung von Labels erleichtern die Programmierarbeit enorm. Frei definierbare Labels dürfen bis zu 14 Zeichen lang sein.

Der Assembler kennt zwei Spielarten:

- **Resident-Modus** (für kleinere Source-Texte bis zu 2500 Listingzeilen);
- **Disk-Modus** (den nutzen umfangreiche, mehrteilige Programmprojekte).

Der letztgenannte Modus ist eher für fortgeschrittene Programmierer geeignet, denn z.B. Nachladen von zusätzlichen Programmteilen und Verwendung von Makros in mehrteiligen Source-Listings ist nicht so einfach zu kopieren, wie's auf den ersten Blick aussieht.

Makros sind selbstdefinierte Unterprogramme, für strukturierte Programmierung unentbehrlich. Mit zwei kurzen Befehlen sind solche Subroutinen rasch definiert. Leider sagt das Handbuch nicht, wieviele Makros maximal einsetzbar sind und welche Größe sie be-

```

$0000: BAD $2F
LASER GENIUS
STATUS: NO BDIZC
          10110001
(C) 1986
OASIS SOFTWARE
S=$FF      $0
X=$0       $0
Y=$0       $0
A=$0       $0
ROM OFF    $0
          $FF
          $FF

$FFE8 2C 03 4C 9B F6 4C 05 E5 L L L
$FFF0 4C 0A E5 4C 00 E5 52 L L L
$FFF8 42 59 43 FE E2 FC 48 BVC H
$0000 >2F< 36 00 AA B1 91 B3 /6
$0008 00 00 00 00 FF 00 00 00
$0010 00 00 00 00 22 0B 19 16
$0018 00 0C 08 08 00 00 00 00

GENIUS MONITOR/ANALYSER.....BY M.LEWIS

```

Bildschirmoberfläche des Monitor- und Analyseprogramms

legen dürfen. Verschiedene Kommandos, bekannt aus Basic-Dialekten (z.B. IF...THEN...ELSE) stehen zur Verfügung. Textausgabe auf dem Bildschirm aktiviert man z.B. per PRINT. Für den Umgang mit Druckern oder Diskettenlaufwerken kommt man ohne langwierige Parameterübergabe und Unterprogrammaufrufe aus, dafür existieren Befehle wie SKIP, HEIGHT oder FILE.

Der wie eine Befehlsweiterung aufgebaute **Editor** besitzt eine Reihe von Anweisungen, die ebenfalls sehr an Basic erinnern. Für die teilweise recht langen Befehlsfolgen lassen sich auch Abkürzungen verwenden, die man dann z.B. auf die Funktionstasten legt. Die sind nach dem Aktivieren des Programms zwar schon vorbesetzt, lassen sich aber jederzeit ändern.

Bei der Eingabe eines Assembler-Listings (= Source-Code) erfolgt ein automatischer Syntax-Check. Man kann die Funktion natürlich auch abschalten, um den Quell-Code in aller Ruhe einzugeben. Die Syntaxprüfung läßt sich auch später noch realisieren.

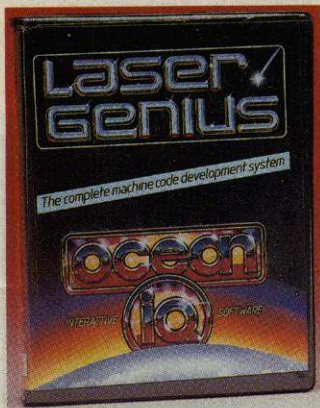
Vor jeder Löschkaktion (NEW oder DELETE) erscheint eine Sicherheitsabfrage, die man bestätigen muß.

Der Umgang mit Diskettenlaufwerk oder Drucker ist komfortabel gelöst. Die unvermeidliche Endung ".8" entfällt, Programmnamen muß man nicht mehr in Anführungszeichen setzen.

Fürs Speichern sind vier unterschiedliche SAVE-Methoden vorgesehen: Quelltexte haben ein an-

deres Dateiformat als Objekt-Codes oder ausführbare Maschinensprache.

Den Programmnamen darf man schon in der ersten Listingzeile verewigen. Nach der entsprechenden SAVE-Anweisung erwartet das Programm lediglich ein Synonym – und der Filename wird aus dem Source-Code geholt. FSAVE sichert den Source-Text in formatierter Form. Damit läßt



Der Laser Genius Assembler "live" in Buchform

C 64 und ein Entwicklungssystem, wie z.B. das "Laser Genius Machinecode Development System" von Ocean-Software.

Das Programmpaket besteht aus Editor, Zwei-Wege-Makroassembler, einem Monitorprogramm und einem Analyser.

Die Seiten der englischen (!) Bedienungsanleitung nerven beim Lesen feuerrot: Das mag zwar Leute abschrecken, die Bedienungsanleitungen kopieren, aber ehrliche Käufer ruinieren sich damit die Augen. Auch die vielen Querverweise verwirren rasch.

Schon im Einführungstext der Anleitung wird jedem klar, daß dies kein Unterrichtsmaterial für

64'er-Wertung: Laser Genius

Umfangreiches und komfortables Assembler-Entwicklungspaket, das aber mehr auf fortgeschrittene Programmierer zugeschnitten ist (Grundkenntnisse der 6502/6510-Maschinensprache sind Voraussetzung). Der Editor besitzt einen umfangreichen Befehlssatz, der leicht zu erlernen ist.

Positiv

- günstiger Preis
- Programme nicht kopiergeschützt (eigene Sicherheitskopien möglich)

Negativ

- Handbuch (nur englisch, schwer lesbar durch Rotfärbung der Seiten)

Wichtige Daten

Testkonfiguration: C 64 (Brotkasten), Floppy 1541-II
Bezugsquelle: T.S. Datensysteme, Denisstr. 45, 90255 Nürnberg, Tel.: 0911/28 82 86, Fax: 0911/26 89 73

Beurteilung

Funktionen: +++
Bedienung: ++
Dokumentation: +
Preis/Leistung: ++

gut

er sich in Textprogrammen verarbeiten oder direkt zum Drucker schicken. Apropos Drucker: Man kann Druckerdaten und -kommandos direkt zum User-Port umleiten.

Auch alle CTRL-Codes werden damit direkt an die Centronics-Schnittstelle gesandt. Das OC-Kommando aktiviert den Datenpacker für alle Objekt-Codes. Damit werden superlange Objekt-Files aufs Minimum gestaucht. Weitere Anweisungen: RENUMBER (Zeilen neu nummerieren), AUTO (Zeilennummern automatisch vorgeben), FIND (Zeichen oder Texte im Source-Code suchen und finden) oder MOVE (Quelltextblöcke verschieben).

Per eingebauter Taschenrechnerfunktion lassen sich Werte arithmetisch oder logisch verknüpfen. Die Ausgabe erscheint binär, oktal, dezimal, hexadezimal oder als ASCII-Zeichen. In Assembler-Listings kann man Zah-

len dezimal oder hexadezimal eingeben. Zum besseren Überblick wurde im Editor/Assembler ein abgespecktes Monitorprogramm integriert.

Neben den üblichen Funktionen wie Disassemblieren und Hexdump (Ausgabe der Speicherinhalte) gibt's noch die Trace-Funktion zur Fehlersuche. Sehr interessant präsentiert sich die Methode zum Verschieben von Speicherbereichen: alle Sprungadressen von JMP, JSR, usw. werden an den neuen Speicherbereich sinnvoll angepaßt.

Komfortabler Monitor mit Analyse-Programm

Die Oberfläche des Monitors und Analysers sieht dagegen völlig anders aus: Hier gibt's lediglich eine Kommandozeile für Befehls- und Dateneingaben. Die untere Screen-Hälfte zeigt den Memorydump und die Komman-

dozeile, im oberen Bereich erkennt man die disassemblierten Mnemonics und den aktuellen Befehl sowie alle Statusregister inkl. Flags (unverzichtbar für erfolgreiche Fehlersuche!).

Der Einzelschrittbetrieb (Single Step) arbeitet Befehl für Befehl ab, dabei aktualisieren sich die Registerinhalte. Bei längeren Programmen kann der Einzelschrittmodus allerdings zur Qual werden: dann aktiviert man den automatischen Schrittbetrieb mit acht unterschiedlichen Optionen (sie manipulieren Arbeitsgeschwindigkeit und die jeweilige Aktualisierung der Registerinhalte).

Wer Breakpoints benutzt, erleichtert sich die Fehlersuche. So lassen sich in selbstentwickelten Programmen bestimmte Bereiche eingrenzen, um sie anschließend genauer zu untersuchen.

Der **Analyser** hat ein paar Funktionen in petto, mit denen man dem Fehlerteufel endgültig

den Garaus macht. Dieser Programmteil des Entwicklungssystems kennt Befehle, die in etwa der Programmiersprache FORTH entsprechen. Doch daran sollten sich nur fortgeschrittene Programmierer halten, denn schon allein die strengen Vorschriften der Befehlseingabe bringen Einsteiger in arge Verlegenheit.

Kurz und bündig

Man darf das Laser-Genius-Entwicklungspaket zur kleinen Gruppe leistungsfähiger C-64-Assembler rechnen. Schwach ist das Handbuch (englisch, schlecht lesbar!). Der günstige Preis von 19,90 Mark macht diesen Nachteil aber wieder wett.

Das Handbuch zu Laser Genius bietet außerdem noch eine Liste aller Assembler-Befehle und die Beschreibung der CPU-6502/6510-Adressierungsarten.

Andree Herman/bl

Laser Basic Compiler

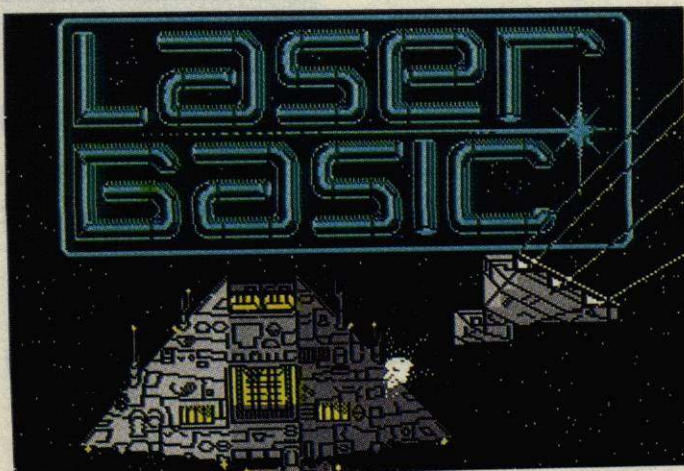
High Speed für Basic-Programme

Compiler machen Basic-Listings hoffähig und sind für jeden Basic-Programmierer ein unentbehrliches Werkzeug.

**64'er
TEST**

Der "Laser Basic Compiler" ist außerdem ein Multitalent: er verarbeitet auch das Standard-Basic 2.0 des C 64. Das Paket wurde von Ocean-Software

entwickelt und bezeichnet sich vollmundig als "Geheimnis fortschrittlicher Spielentwicklung". Wer nun glaubt, ein Game-Construction-Set vor sich zu haben, wird enttäuscht: Mit Spieleprogrammierung hat die Software so



Laser Basic: erweitert den Basic-Interpreter um 256 neue Befehle

*** C64 LASER BASIC COMPILER ***

(C) 1985 D. HUNTER

READY.

Keine Einschaltmeldung, sondern das Ergebnis eines LIST-Befehls bei kompilierter Software

viel (oder so wenig) zu tun wie die schrille Pop-Madonna mit einer Nonne.

Man kann lediglich vermuten, was die Hersteller damit andeuten wollten: Programme (z.B. Spiele), die mit der komfortablen Basic-Erweiterung "Laser Basic" (das dritte Kind aus der Laser-Software-Reihe) entwickelt wurden, lassen sich mit dem 3-Paß-Compiler nahezu vollständig ins Maschinensprache-Format übersetzen –

womit solche Programme mit Geschwindigkeit und Professionalität veredelt werden.

Nützliche Utilities

Die Systemdisk zum Compiler-Paket bietet zusätzlich nützliche Utilities:

- der **Bootgenerator** erzeugt ein Ladeprogramm fürs Kompilat,
- **Turbo** nutzt nur Datensette-Besitzern etwas – damit beschleunigt

sich die Datenaufzeichnung auf Band ums Siebenfache der Normalgeschwindigkeit. Der Compiler macht's möglich, die bearbeiteten Programme direkt im Turbo-tapeformat zu speichern.

Ein umfangreiches Demonstrationsprogramm findet man ebenfalls auf der Disk. Es soll beweisen, wie eindrucksvoll sich das Tempo von Grafik, Sprites und Textausgabe in umgewandelten Programmen präsentiert. Dazu gibt's natürlich die unentbehrlichen Unterprogramm-bibliotheken (LIB1 bis LIB6).

Der Compiler läßt sich sehr einfach bedienen und kann auf alle angeschlossenen Floppies und (weniger interessant) auf Datensette zugreifen. Man muß nach dem

und entschärft wurden. Allerdings erhöht sich dabei der vom Kompi- lat benötigte Speicherplatz – genauso wie bei der Verwendung von Fließkommazahlen.

Kurz wird lang, lang wird kurz

Trotz des enormen Geschwindigkeitszuwachs sollte man nicht vergessen, daß kurze Basic-Programme anschließend um etliches länger sind als im Urzustand (ca. 30 Blocks) – bei umfangreichen ist's genau umgekehrt: Ein 13 KByte großes Basic-Listing belegt nach dem Kompilieren immerhin 16,5 KByte im Speicher. Ein Programm mit 33 KByte dagegen

braucht 5,5 Minuten zur Umwandlung, ist aber dann 4 KByte kürzer als vorher. Der Grund: in jedes kompilierte Listing wird automatisch eine Unterprogramm-bibliothek (Run-Time-Library) eingebaut, deren Umfang stets gleich bleibt. Diese Bibliothek ist auch dafür zuständig, daß Programme, die mit der Befehlerweiterung Laser-Basic entwickelt wurden, ohne dieses Tool auskommen (im Gegen-

satz dazu sind z.B. Simons-Basic-Listings nie ohne vorher aktivierte Basic-Erweiterung lauffähig!).

Achtung, wichtige Info für Programmierer! Solche Kompilate



Der Laser Compiler wurde als Erweiterung zu Laser Basic entwickelt

Start lediglich das gewünschte Programm laden (auch der Joker <*> ist als Abkürzung erlaubt) und festlegen, ob es sich um ein Commodore-Basic-2.0-Programm oder einen Laser-Basic-Quelltext handelt.

Bei allen Basic-Compilern ein kritischer Punkt: die Wahl des Zahlentyps – Ganzzahlen-Arithmetik (Integer) oder Fließkomma (= Exponentialzahlen). Den integrierten Error-Check sollte man auf alle Fälle aktivieren. Damit ist sichergestellt, daß für den späteren Programmstart alle Fehler vorher entdeckt

Laser Compiler (Testreihe)

Zum Test wurden drei unterschiedlich lange Basic-2.0-Listings mit der Option "C" (Commodore-Basic) bearbeitet (Länge in Blocks auf Diskette). Unsere Tabelle zeigt die Ergebnisse.

Original	Kompilat	Zeitaufwand (ohne Laden, Speichern und Diskettenwechsel)
6	36	16 Sekunden
50	70	33 Sekunden
107	116	4 Minuten, 9 Sekunden

Hinweis: Ein Basic-Programm mit 135 Blocks auf Diskette (= 34 290 Byte) ließ sich auch nach fünf Versuchen nicht kompilieren (Abbruch mit der Meldung "Fatal Error"). Achten Sie also darauf, daß zu bearbeitende Listings nie größer als maximal 32 KByte sind (= 129 Blocks auf Diskette).



Sprites und Windows: Softscrolling in jeder gewünschten Richtung

unterliegen keinerlei Copyrightbestimmungen und dürfen deshalb auch frei vertrieben und angeboten werden.

Es scheint eine Marotte von Ocean-Software zu sein, alle Handbücher auf dunkelrotes Papier mit schwarzer Schrift zu drucken – auch eine Art von Kopierschutz, der Kopiergeräte des heutigen Standards kaum in Verlegenheit bringt. Wenigstens ist die Anleitung in Deutsch und sehr ausführlich. Neben der üblichen Befehlerläuterung enthält es nützliche Programmertips und die Beschreibung der Fehlermeldungen (15 verschiedene). Allerdings – ein bißchen Kritik muß schon sein: So befindet sich z.B. ein Absatz zur Ganzzahlen-Arithmetik im Kapitel über den Bootgenerator. Oder andersrum: der logischerweise fehlende Absatz über den Generator steckt nun im Kapitel über die "Transzendenten Funktionen". Ein wenig mehr Sorgfalt beim Zusammenstellen des Handbuchs hätte den guten Eindruck der Software in vollem Umfang gerettet ...

Auf einen Blick

Der Laser-Compiler übersetzt flott und fehlerfrei selbst überdimensionale Listings des Entwicklungssystems Laser Basic, auch das normale Basic 2.0 des C 64 wird akzeptiert (statt der Runtime-Bibliothek LIB1 benutzt der Compiler dann LIB4).

Achten Sie beim Kompilieren darauf, daß auf der Arbeitsdisk noch mindestens doppelt so viel Platz frei ist, wie das Originalprogramm verbraucht. Während des Kompilierens schreibt Laser Compiler temporäre Dateien auf Disk, die anschließend wieder gelöscht werden.

Hinweis: Selbstverständlich lassen sich lediglich waschechte Ba-

sic-Programme kompilieren, also keine Maschinensprache-Files mit Basic-Startzeile, die z.B. nach dem LIST-Befehl "10 SYS 2061" bringen!

Wer Laser Basic noch nicht besitzt (die verbesserte Version von "Basic Lightning"), sollte sich an die Bezugsadresse im Bewertungskasten wenden.

Andree Herman/bl

64'er-Wertung: Laser Compiler

wurde entwickelt als Ergänzung des Programm-Entwicklungspakets "Laser Basic" (Betriebssystem-Erweiterung) desselben Herstellers. Da der Befehlsatz auch die Basic-2.0-Befehle enthält, läßt sich jedes normale C-64-Basic-Programm ebenfalls kompilieren.

Positiv

- günstiger Preis
- deutsche Anleitung
- Programme nicht kopiergeschützt (eigene Sicherheitskopien möglich)
- ausführliche Demo- und Hilfsprogramme
- schnelle Datenumwandlung
- akzeptiert alle angeschlossenen Diskettenstationen

Negativ

- Handbuch (schwer lesbar durch Rotfärbung der Seiten und mangelnde Sorgfalt bei der Textzusammenstellung)
- Kopierschutz auf Original-Disk (keine Sicherheitskopien möglich!)

Wichtige Daten

Testkonfiguration: C 64 (Brotkasten)/C 128-Single, Floppy 1541-II
Bezugsquelle: T.S. Datensysteme, Denisstr. 45, 90255 Nürnberg, Tel.: 0911/28 82 86, Fax: 0911/26 89 73
Preis: 19,90 Mark

Beurteilung

Funktionen: +++
Bedienung: +++
Dokumentation: ++
Preis/Leistung: ++

sehr gut

SORRY, WERBUNG GESPERRT!

64ER C

WWW.64ER-ONLINE.DE

How to GoDot

Folge 1

Unsere Sammelserie soll die Arbeit mit "GoDot" grundlegend erleichtern und beleuchtet dann in Schlagwörtern die Bedienungsweise der über 100 Module dieses Allround-Grafiktalents. Diesmal beginnen wir mit den grundlegenden Handhabungshinweisen. Stichwörter, auf die später eingegangen wird, sind gekennzeichnet. Am Ende bekommen Sie zusätzlich ein Schlagwortregister zum schnellen Auffinden.

Die vielen Funktionen des Image-Prozessors "GoDot" (unser Programm des Monats in 64er-Ausgabe 7/93, inzwischen durch viele Zusätze ergänzt) können den User schnell erschlagen. Im wesentlichen hat sich die Bedienung von GoDot nicht geändert, aber einige Verbesserungen sind seit der damaligen Veröffentlichung des Programms vorgenommen worden.

Die Anleitung bezieht sich auf die Update-Version von GoDot in Heft 6/94, die ja jeder 64er-Leser besitzen müßte. Außerdem werden alle Module, die während der nächsten Ausgaben für GoDot entstehen, automatisch in die Anleitung einfließen.

Bevor Sie mit der Nutzung des Programms beginnen, sollten Sie (wie bei jeder anderen Software auch) unbedingt eine Arbeitskopie mit Hilfe eines üblichen Backup-Programms anfertigen. Auf der Arbeits-Diskette können eventuell einige Files gelöscht werden, die man für die eigene Arbeit nicht braucht (das Angebot an Savern ist ja kaum zu überschauen, nicht jeden werden Sie benötigen). Der freie Speicherplatz kann sehr gut zum Sichern von Daten dienen. Diese Vorgehensweise ist vor allem für Besitzer von nur einem Disketten-Laufwerk interessant, da so lästiges Wechseln von Floppy-Discs entfällt.

Grundsätzlich läuft GoDot auf jedem C 64/C 128 mit einem Diskettenlaufwerk. Besitzer mehrerer

Laufwerke, Festplatten oder einer RAM-Erweiterung o.ä. kommen in den Genuß komfortableren Arbeitens. Gängige Speeder (z.B. Dolphin-DOS, Speed-DOS) arbeiten mit GoDot zusammen. Bei Exoten kann es Probleme geben. In solch einem Falle muß die Erweiterung abgeschaltet oder entfernt werden. Multifunktions-Module (u.a. Magic Formel oder Action Replay Cartridge) werden von GoDot akzeptiert. Das Tool ist modular aufgebaut und hält für jeden Arbeitsschritt den entsprechenden Programmteil im Speicher oder lädt ihn nach.

Die Steuerung erfolgt über Maus, Tastatur oder mit einem Joystick. Das Programm reagiert jederzeit auf Eingaben durch alle drei genannte Geräte.

Um mit GoDot zu arbeiten, laden Sie das Programm mit **LOAD"GODOT",8,1** und starten es mit dem RUN-Befehl. Bei der Nutzung unserer Kurzanleitung sollten sie auf die Querverweise achten, die mit ► Stichwort gekennzeichnet sind.

Wolfgang Kling und
Arndt Detke/lb

How to GoDot: So läuft's ab!

Um die Anleitung zu GoDot immer zur Hand zu haben, werden alle Teile der Beschreibung zum Ausschneiden gestaltet. Einfach die Seite aus dem Heft getrennt und in einem Ordner abgelegt. Nun kann man jederzeit in der Anleitung nachschlagen, ohne in zahlreichen 64er-Magazinen blättern zu müssen.

Grundlagen

► **Mauszeiger, Zeiger, Pointer**
Der Standardmauszeiger besteht aus einem kurzen, zweifarbigen Balken. Er ist immer sichtbar, außer es findet ein Zugriff auf eine Diskette statt oder man hat ihn außerhalb des Bildrands bewegt. Gesteuert wird der Zeiger mit einer Analogmaus (z.B. 1351), mit einem Joystick oder per Cursor-Tasten. Man muß dazu keine besondere Einstellung vornehmen.

Alle Eingabegeräte können zu gleicher Zeit verwendet werden. Mauszeiger lassen sich durch andere ersetzen (► **mod.NewPointer**) bzw. in ► **INI-Files**, wobei auch der "Hotspot" (der Punkt, mit dem gezeigt wird – normalerweise: links oben) angepaßt werden kann (Beispiel: ptr.Cross). Wem die Cursor-Tasten zu langsam sind:

- C=h: rechts
- C=f: links
- C=s: abwärts
- C=F3: aufwärts

Die rechte Maustaste ist unbenutzt, die linke hat Auslösefunktion (wie Feuer und Return)

► **Booten, GoDot starten**

Um GoDot zu starten, ist das File "godot" unerlässlich. Es gibt keine Möglichkeit, das Programm mit SYS z.B. nach einem Reset wieder zu aktivieren. Die-

ses Startfile erledigt wichtige Aufgaben:

● alle angeschlossenen Floppylaufwerke werden getestet, die 1571 dabei in den doppelseitigen Modus geschaltet;

● alle möglichen ► **RAM-Erweiterungen** werden getestet, eingebunden (► **Devices**) und gemäß dem aktuellen ► **INI-File** mit Inhalt versorgt (Melung: "Installing...").

● zum Schluß nimmt "godot" alle Grundeinstellungen vor (Farben usw.) und startet GoDot mit dem Hauptbildschirm (s. Bild 1). Von welchem Laufwerk gebootet wird, ist dabei unerheblich. Beim Starten werden insgesamt mindestens vier Dateien geladen: "godot", "god.set" (der Zeichensatz), "god.main" (der GoDot-Kernel) und "god.upmem" (mit den Routinen zum ► **Rendern** eines Bildes). Fehlt eines dieser Files, ist GoDot nicht lauffähig und endet mit einem Reset!

► **INI-File, Voreinstellungen, Konfiguration**

GoDots Bildschirmfarben, (Zeiger, Standardzugriffslaufwerke und einige, die RAM-Erweiterungen betreffende Einstellungen erfolgen bereits beim "Booten. GoDot lädt zu diesem Zweck das File "god.ini" (oder ein anderes "ini.*") und führt die dort abgelegten Befehle aus. "INI-Files erstellt man wie ein

Basic-Programm. Jede Zeile beginnt allerdings mit einem Ausführungszeichen und enthält nur eine einzige Anweisung (s. Tabelle). Fehlerhafte Zeilen werden bei der Ausführung ignoriert. Solch ein File wird mit dem Kürzel "ini." am Anfang des Namens gespeichert. Damit GoDot Kenntnis von einem INI-File bekommt, hängt man sie der Ladeanweisung im Direktmodus als Parameter an, z.B.:

```
LOAD "GODOT",8:"SCAN"
```

Die Befehle der Script-Sprache finden Sie in Tabelle 1.

► **Gadgets, Funktionsflächen, Fenster, Requester**

Schaut man sich den Hauptbildschirm von GoDot an, fällt auf, daß er weder Pulldown-Menüs (wie GEOS), noch Textmenüs (wie viele Basic-Programme) zeigt. Vielmehr ist sein Bildschirm in sechs umrahmte Bereiche gegliedert, in denen beschriftete Kästen zu finden sind.

Die Graustufen des C 64

Diese Bereiche sind fest auf dem Screen verankert und nicht verschiebbar, sie heißen "Windows" oder "Fenster". Sie stellen zusammengehörige Funktionen zur Verfügung, die über die Kästen – "Gadgets" oder "Funktionsflächen" – aktiviert werden.

Dazu werden die Gadgets "angeklickt", d.h. man zeigt darauf und betätigt die Auslösetaste (► **Zeiger**) des Eingabegeräts. Das Gadget blinkt kurz und reagiert auf drei verschiedene Weisen:

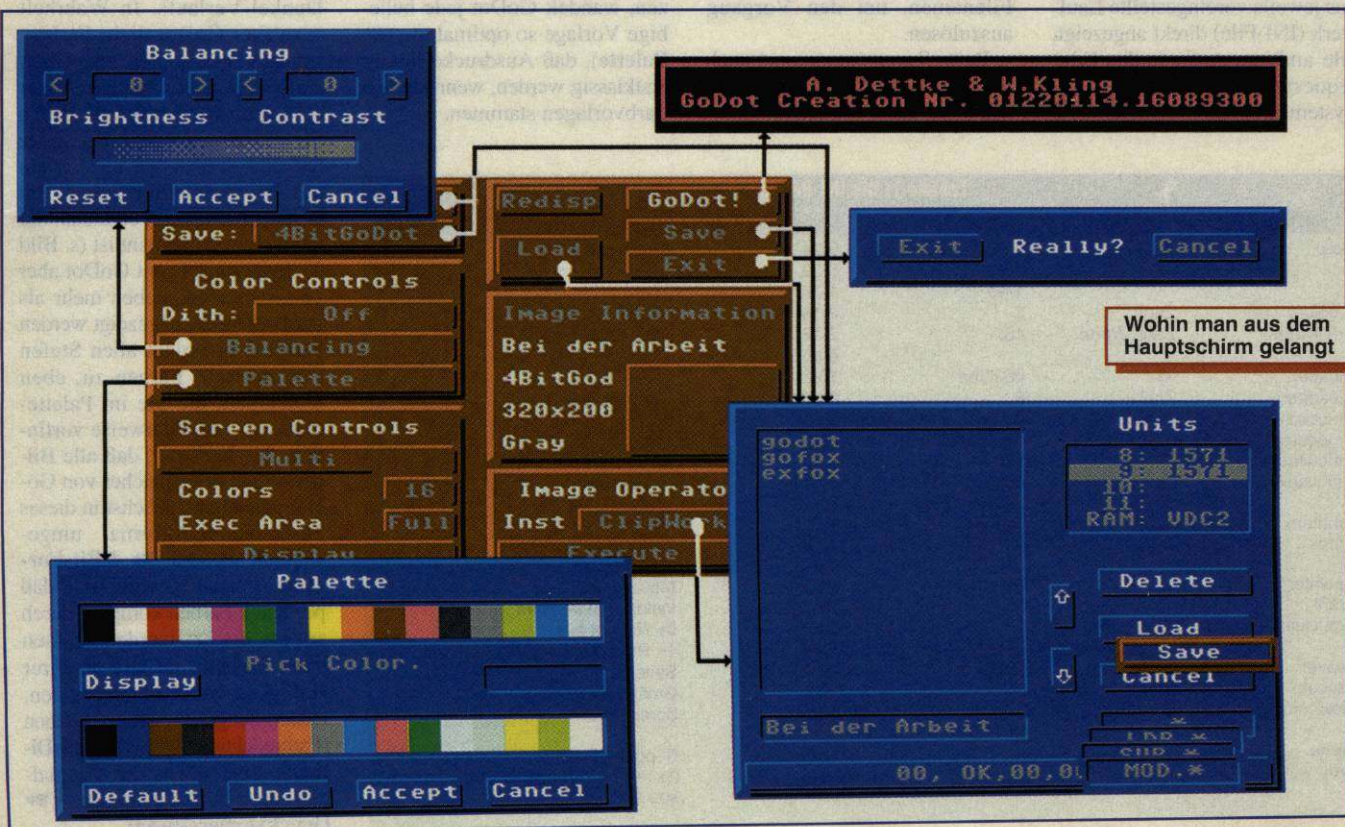
1. Man hat ein ► **Bildbearbeitungsmodul** in Gang gesetzt, das die Grafikdaten GoDots verändert. Das Gadget bleibt wie es war, aber im Fenster ► **"Image Information"** bewegt sich ein feiner weißer Strich abwärts. Er zeigt, daß GoDot beschäftigt ist.

2. Man hat eine Einstellung vorgenommen. Jetzt verändert sich die Beschriftung des Gadgets. Das Gadget "Multi" im Fenster "Screen Controls" wechselt z.B. auf "Hires". Die gültige Einstellung ist die jeweils sichtbare!

3. Es öffnet sich ein Window, das den bisherigen Bildschirm teilweise überlagert und neue Funktionen zur Verfügung stellt, ein "Requester". Beispiel: Klicken Sie auf das Gadget "Palette" im Fenster "Color Controls". Requester bieten immer eine Abbruchfunktion ("Cancel" oder "Leave"), falls man versehentlich hineingeraten ist.

► **Filerequester**

Der Filerequester hat eine zentrale Stellung innerhalb GoDots. Über ihn sind alle Arten von Speichermedien ansprechbar (alle Floppylaufwerke und alle unterstützten ► **RAM-Erweite-**



rungen). Aus dem Hauptbildschirm führen allein fünf Gadgets direkt hierhin: alle mit der Beschriftung "Load" bzw. "Save" und das "Inst"-Gadget im Fenster "Image Operators" (s. Bild 1). Jedes dieser Gadgets nutzt den Filerequester auf seine Weise, indem nämlich im Directory-Fenster ausschließlich solche Files angezeigt und bearbeitet werden, die zum aufrufenden Gadget passen. Alle anderen werden weggefiltert. Welche Files aus dem Aufruf stammen, erkennt man am Fenster ganz rechts unten. In der Abbildung haben wir die häufigsten Filterangaben einmal übereinandergeblendet. Es können noch andere erscheinen:

"*" – es werden Files angezeigt, die Grafiken sein könnten, der Aufruf erfolgte vom besonders großen "Load"-Gadget bzw. vom "Save"-Gadget im selben Fenster oben rechts

"LDR.*" und "SVR.*" – GoDot zeigt alle **Lader-** und **Saver-**Module, die sich auf dem ausgewählten Laufwerk befinden, der Aufruf kam von ganz links oben, aus dem I/O-Fenster (beschriftet mit "Load:" und "Save:")

"MOD.*" – nur **Bildbearbeitungsmodule** oder (Systemtools werden ausgegeben, aufgerufen vom "Inst"-Gadget in "Image Operators".

Bei "Load" und "Save" wird das jeweils voreingestellte Laufwerk (INI-File) direkt angezeigt, alle anderen Aufrufe des Filerequesters geben zunächst das Systemdrive aus. Normalerweise

das **Boot-Drive**; **RAM-Erweiterungen**. Durch Klicken ins "Units"-Gadget läßt sich ein anderer Drive ansprechen.

Im "Units"-Gadget werden die Drives (mit Typenangabe) angezeigt, die momentan zur Verfügung stehen. In der Abbildung sehen Sie die Angaben für einen C128DB mit zwei Laufwerken. Fehlt die Typenbezeichnung, gibt es unter dieser Adresse keinen Drive. Klickt man die Drivenummer trotzdem an, erfolgt im Statusgadget ganz unten die Meldung "Off." und GoDot wählt selbständig die Einstellung "RAM", denn die stimmt in jedem Fall.

Sollte das Directoryfenster nicht alle vorhandenen Files zeigen können (16 passen hinein), so blättern Sie mit Hilfe der beiden kleinen, mit einem Pfeil beschrifteten Gadgets in die entsprechende Richtung. Eine Auswahl treffen Sie schließlich, indem Sie das gewünschte File einmal anklicken. Der Name erscheint daraufhin im Input-Gadget unterhalb des Directories. Dann wird mit dem "Load"-Gadget des Filerequesters betätigt. Es wird daraufhin geladen (was GoDot mit einem Anzeigebalken im Statusgadget begleitet) bzw. installiert (ohne Anzeige). Sollten Ihnen zwei örtlich voneinander getrennte Klicks zu umständlich erscheinen, genügt auch ein doppelter Klick auf den Filenamen, um den Vorgang auszulösen.

Beim Saven (manchmal auch beim Laden) ist es sinnvoll, einen Filenamen von Hand vorzu-

geben. Das erreichen Sie, wenn Sie ins Input-Gadget klicken. Namen können hier, wie auch sonst üblich, beim Laden mit einem Stern abgekürzt werden.

Wenn einmal die Floppy-LED blinkt, und Sie befinden sich im Filerequester, holen Sie sich die Fehlermeldung mit einem Klick ins Statusgadget ab.

Mit dem Filerequester können Sie auch bequem Dateien löschen. Sie klicken den Namen an und lösen dann das "Delete"-Gadget aus. Auch hierbei funktioniert die Sache mit dem Sternchen. Sollten Sie aus Versehen einen einzelnen Stern eingeben und danach auf "Delete" drücken, werden Ihnen alle Files des angegebenen Filters gelöscht!

Übrigens kann GoDot nichts mit den Filenamen von GEOS anfangen, wenn sie als Kleinbuchstaben (in GEOS) eingetippt wurden. GEOS benutzt für Buchstaben die Standard-ASCII-Codierung, die GoDot nicht beherrscht. Mit GEOS-Großbuchstaben funktioniert's hingegen.

Druckvorbereitung, Balancing

GoDot kann (noch) nicht selber ausdrucken. Dafür jedoch unterstützt es Fileformate, die zu Programmen mit hervorragenden Druckertreibern gehören, allen voran Print- bzw. Pagefox (**svr.PFoxGB**; **svr.Pagefox.90**; **mod.4Bit2PFox**). Um dem Ganzen die Krone aufzusetzen, wandelt GoDot jede beliebige Vorlage so optimal um (**Palette**), daß Drucke immer erstklassig werden, wenn sie von Farbvorlagen stammen.

Dazu bedient man sich vor allem des "Color Controls"-Fensters. Zunächst stellt man den **Grafikmodus** auf "Hires" (in "Screen Controls"), die Anzahl der zu **rendernden** Farben auf 2 ("Colors" im gleichen Fenster) und ändert im **Palettenrequester** die Grafikfarben auf "Default". Dann wählt man ein geeignetes **Raster** im "Dith"-Gadget aus und **rendert** die Grafik zur Probe ("Display"). Wenn Helligkeit und Pixelanordnung nicht den Erwartungen entsprechen, bessert man nach, indem man im "Balancing"-Requester Kontrast- und Helligkeitswerte verstellt oder ein neues **Raster** auswählt.

Die oben genannten Saver **rendern** übrigens die Vorlage vierfach feiner als auf einem C-64-Bildschirm dargestellt werden kann (nämlich in 640x400 Pixeln). Wenn man einen C128 besitzt, sollte man sich einen echten Eindruck mit den Modulen **mod.4Bit2VDC*** und **mod.VDCEd** verschaffen, die auf einem zusätzlich angeschlossenen 80-Zeichen-Bildschirm ausgegeben werden. Weitere Druckvorbereitungsmodule sind die Saver **svr.PCXmono** (MS-DOS-Rechner) und **svr.Pagesetter** (Commodore-Plus4). **Palette, Palettenrequester**

GoDots zugrundeliegende Anordnung der Grafikfarben folgt dem Prinzip eines Hell-Dunkel-Verlaufs. In Wahrheit verwaltet GoDot nicht Farben, sondern Helligkeiten, also Graustufen. Hardwaremäßig gibt der C64 neun Graustufen ab: Rufen Sie einmal den Palette-Requester auf und drehen Sie die Farben aus Ihrem Monitor heraus. Im unteren Farbgadget erkennen Sie deutlich, was gemeint ist (s. Bild 2). Insgesamt kennt GoDot aber 16 Graustufen, sieben mehr als hardwareseitig angezeigt werden können. Er ordnet allen Stufen (rechnerisch) Farben zu, eben diejenigen, die Sie im Palette-requester normalerweise vorfinden. Das bedeutet, daß alle Bilder, die in den Speicher von GoDot gelangen, zunächst in dieses 16-Graustufen-Format umgewandelt werden (**4-Bit-Format**). Vorteil ist einerseits, daß per Klick Farben komplett durch andere ersetzt werden können bzw. Farben, die sich in ihrer Helligkeit kaum unterscheiden, zusammengefaßt werden können (Farbreduktion; **mod.ApplyDither**); andererseits die originalgetreue Druckwiedergabe (**Druckvorbereitung**).

Tabelle 1: Die INI-Befehle

Befehl	Parameter	Default	Bereich	Zweck
clip	zl, sp, br, ho	0, 0, 40,25	0-24 0-39	Clipvorgabe
colors	Zahl	1-40	1-25	Anzahl, Farben
defcolors	Zahl, Werte	2 16,...	2-16 2-16, 0-15	Defaultpaletten
loader	ldr.*	4BitGod	16 Z.	Lader
loadfrom	Zahl	8	8-11	Grafikdrive (Load)
mode	Zahl	0	0;>0	0=Hires; >0=Multi
module	mod.*	ClipWorks	16Z.	Modul
mousecolors	Zahl, Zahl	1, 0	0-15	Mausfarben
mousexy	Zahl, Zahl	80, 80	0-511	Mausposition
pattern	pat.*	0-255 System	16Z.	Raster
pfox	Zahl	0	0;>0	0=Pagefox f.Module >0= Pagefox f. Grafik
pointer	ptr.*	Standard	16 Z.	Mauszeiger
ram	*..*		16 Z.	Installieren im RAM
reuclear	Zahl	1	0;>0	0= Reboot aus 1= Reboot wieder an
saver	svr.*	4BitGod	16 Z.	Saver
saveto	Zahl	8	8-11	Grafikdrive (Save)
textcolors	s, l, t, h, m	0, 14, 12,15,60	0-15	Screenfarben
ticks	Zahl	13	0-255	Doppelklickzähler
vdc	Zahl	1	0;>0	0= VDC für Module; >0= VDC für Grafik

Das ganze Haus reagiert auf die Befehle Ihres C 64! Eine verführerische Vorstellung, die sich mit dem "Tele-Commander" zumindest teilweise umsetzen läßt. Das Übertragungsmedium steht in jedem Raum mehrfach zur Verfügung.

64'er TEST

Der "automatisierte Haushalt" wurde schon oft in Science-fiction-Romanen dargestellt. Viele Elektronik-Versender haben Schaltuhren im Programm, die auf Wochen voraus zu programmieren sind. Die "Intelligenz" eines Computers fehlt ihnen jedoch in der Regel, und auch bei der Zusammenarbeit mit Computer-Software sieht es schlecht aus.

EHA-Elektronik in Köln hat jetzt eine Schaltung aus der Zeitschrift "Elektor" in funktionsfähige Hardware "materialisiert", die (das ist das Besondere) sich vom Computer aus fernsteuern läßt. Die Software dafür liegt für PC (DOS und Windows), Amiga, Atari, Mac und eben auch für den C 64 vor, so daß Ihrem Automatisierungsdrang nur noch ca. 300 Mark für den Sender und je 175 Mark pro Empfänger im Wege stehen.

So funktioniert's

Statt infrarot (wie oft bei derartigen Fernsteuerungen üblich) geht das Schaltsignal einen sehr praktischen Weg, über den sowieso jedes Haushaltgerät erreichbar ist: das Stromnetz. Der Transmitter sendet ein 110-kHz-Signal ins

Das Geister-Haus

Der Tele-Commander Butler im Lehnstuhl

Netz, das von allen angeschlossenen Empfängern geprüft wird. Nur die Schalteinheit reagiert jedoch, deren Adresse mit der im Signal verschlüsselten übereinstimmt. Anschließend meldet die Schalteinheit ihren neuen Zustand brav zurück. Sie können bis zu 8 A mit einer Schalteinheit betrei-



Der Empfänger: wird für jedes zu schaltende Gerät benötigt



Der Sender: drei Statusanzeigen geben Auskunft über den aktuellen Zustand

ben, insgesamt stehen 128 verschiedene Adressen zur Verfügung, d.h. es können bis zu 128 Geräte angesprochen werden. Die Verbindung zwischen C 64 und Transmitter wird über ein Userport-Kabel realisiert, das (endlich mal ein Vorteil des C 64) nicht

einmal einen Pegelwandler benötigt (wie in diesem Fall PC und Amiga, die ja 12 V an der seriellen Schnittstelle liefern).

Die Software

Im Prospekt als "komfortabel" angekündigt, erweist sie sich als Schwachstelle des Pakets. Obwohl sie immerhin 70 Mark kostet, geht sie über ein Basic-Programm nicht hinaus, mit dem die einzelnen (im Programm als Relais bezeichneten) Schalteinheiten angesteuert werden. Der Quelltext ist gut strukturiert und ausführlich kommentiert, so daß einer Umsetzung nicht allzuviel im Wege steht. Interessant wäre z.B. eine Umsetzung für GeoTec oder als simples Steuerprogramm unter Geos.

Wünsche offen

Billig ist die Hardware nicht, wenn auch der Bastler einige Mark einsparen kann. Wenn es bereits eine wirklich komfortable Software gäbe, könnte das Netzmodem tatsächlich ernsthaft angewendet werden. Im jetzigen Lieferumfang sind dazu allerdings noch einige Programmierarbeiten seitens des Anwenders nötig, so daß wir das Paket eher dem Profi als dem Einsteiger empfehlen können

Matthias Matting

Preise für Sender und Empfänger	
Schalteinheit (Empfänger)	
Bausatz	100,50 Mark
Fertiggerät	175,50 Mark
Transmitter (Sender)	
Bausatz	175,50 Mark
Bausatz bestückt	225,50 Mark
Fertiggerät	275,50 Mark
Software	
C 64 inkl. Kabel	69,75 Mark
Amiga inkl. Kabel	72,25 Mark

TESTPROGRAMM FUER EHA-NETZMODEM

STATUS RELAIS : STATUSTEXT RELAIS-NR : 127

[A] - AENDERN RELAIS-ADRESSE

[T] - TOGGLE RELAIS

[C] - CLEAR STATUS

[R] - RESET NETZMODEM

[Q] - QUIT

Die Steuersoftware: erfüllt zwar ihre Aufgabe, einzelne Relais anzusteuern, ist aber wenig komfortabel

64'er-Wertung: Tele-Commander

Fernsteuerung für elektrische Geräte über Stromnetz

Positiv

- interessante Idee
- Hardware professionell und technisch sicher umgesetzt
- Rückmeldung durch Schalteinheit)

Negativ

- relativ teuer
- Software nicht sehr komfortabel

Wichtige Daten

Bezugsquelle: EHA-Elektronik Walter Hackländer, 50735 Köln
Preis: ab 340 Mark (Sender und Empfänger als Bausatz, Software)
Testkonfiguration: C 128D, 1581, 1571

Beurteilung

Funktionen: +++
Bedienung: +
Dokumentation: +
Preis/Leistung: ++

gut

Hallo Fans!

Dallas Quest

Henry Kadow hat "Dallas Quest" gelöst und uns die komplette Reihenfolge der Befehle geschickt. Wer nicht jeden Befehl einzeln eintippen und das Spiel durchspielen will, sollte einfach nur das Cheat-Wort "CLUE" eingeben.

E (East), (E, E, look horse, look Lucy, ride horse, W (West), W, N (North), take sunglasses, N, give sunglasses, go barn, N, drop owl, take shovel, S (South), S, S, take envelope, W (West), W, W, play bugle, dig, read tombstone, E, drop money, north, open desk, take pouch, N, N, W, W, N, look plane, give envelope, open knapsack, put ring in knapsack, put photography in knapsack, look in knapsack, take parachute, close knapsack, jump, open pouch, give pouch, close pouch, S, S, look, tickle anaconda, S, S, go boat, open pouch, give pouch, close close pouch, row boat, play bugle, go post, pull curtain, open pouch, give pouch, close pouch, drop all, take flashing, climb ladder, on flashlight, look sign, off flashlight, drop flashlight, E, go post, take all, drop bugle, drop parachute, open knapsack, put all in knapsack, close knapsack, climb ladder, take flashlight (wenn noch irgendetwas in der Post außer dem Fallschirm und der Trompete zurückgeblieben ist, jeweils holen), take all, W, off flashlight, open knapsack, take photograph, show photograph, close knapsack, take coconuts, read sign:

*Auch in diesem Monat tolle Tips aus dem Fundus unserer Leser, der dank der regen Teilnahme ständig anwächst.
Unser Motto: Leser helfen Lesern!*

Netz (Weg wird frei), W, wirf Beutel (Drache stirbt), schneide Auge, nimm Auge, O, H, W, W, N, N, W, gehe in Hütte, gib Auge (Offel benötigt jetzt Kralle), gehe raus, W, N, N, N, O, vergifte

Fleisch, R, wirf Fleisch (Bestie verschlingt Fleisch und stirbt), schneide Kralle, H, W, S, S, O, gehe in Hütte, gib Kralle (Offel gibt Kristallkugel), nimm Kristallkugel, gehe raus, W, N, N, benutze Zaubertrank, N (kein Schaden durch Nebel dank Zaubertrank), H, benutze Kristallkugel...

Turn'n Burn

Alle Paßwörter hat Henry Kadow auch noch mit seiner Post mitgeschickt.

Die Codes müssen im Titelbild eingegeben werden:

2. COLOROS
3. MITCH
4. VAL
5. LAMB
6. NUT
7. SMALL
8. RATE
9. TREE
10. DELTA

1. Zum Löwen gehen und Spiegel zeigen (show mirror, show lion)
2. Zum Adler gehen und dem Affen Kautabak geben (open pouch, give pouch, give eggs)
3. Zum Elefanten gehen und mit Ring winken (wave ring, show tusk)
4. give mirror, open knapsack, take ring, show ring, warm eggs, on flashlight, drop ring, take map, no, off flashlight, give map

Das Spiel kann nur durchgespielt werden, wenn man den zweiten Weg nimmt.

TIP DES MONATS: HARAKIRI

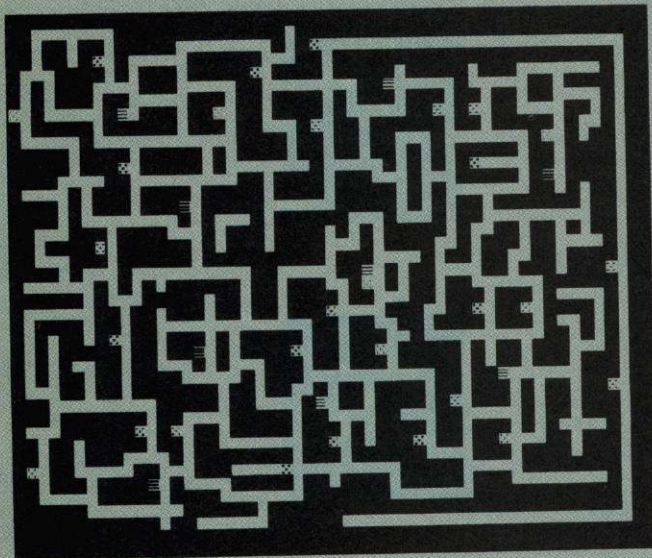
Die Lösung zum Adventure plus Karten ließ uns Matthias Heinz aus Sonneberg zukommen. Dafür erhält er die 100 Mark als Belohnung für den Tip des Monats.

N, N, töte Höhlenmenschen, nimm Dolch, W, N, N, schiebe Wand (Geheimtür öffnet sich), W, nimm Beutel (Drachengift), nimm Gift (gegen "Ungeziefer aller Art"), O, S, S, O, O, N, N, nimm Seil, N, N, N, W, gehe in Hütte (der gute Offel erzählt, daß er dir helfen kann, du sollst ihm dafür aber ein Drachenauge bringen), gehe raus, W, nimm Brecheisen, O, O, N, O, N, nimm Fleisch, S, W, S, S, W, R (runter), R, benutze Brecheisen (öffnet Gitter), befestige Seil (an Leiter), R, nimm Zaubertrank (gegen Nebel des Todes, Tür im W nicht öffnen), H (hoch), H, H, O, O, O, schiebe Stein (ein Loch wird sichtbar), R, zerschneide

Legende zu den Harakiri-Karten:

- 1 - bewegliche Wand
- 2 - Hütte des guten Offel
- 3 - Turm des bösen Offel
- 4 - Spinnennetz
- 5 - Eisengitter
- 6 - Tür zu Ork

Clever & Smart



Die Kanalisation :

Zeichenerklärung: Die Felder die Aussehen wie die fünf auf einem Würfel, sind Stromkästen, die Felder mit den waagerechten Strichen sind Gullys.

Verschollen im Labyrinth der Kanalisation versuchen Clever & Smart, den rettenden Ausgang zu erreichen.

Clever und Smart

Lars Petersen in Dollerup haben es die Abenteuer von "Clever und Smart" angetan. Hier die Karte zur Kanalisation. Die kleinen Würfel sind Stromkästen und die Felder mit den Strichen sind Gullys.

Die dunkle Dimension

Ein wenig Licht ins Spiel von German Design Group hat Aron-D. Gabriel gebracht. Hier seine Tips zu magischen oder verbotenen Waffen, Ausrüstungen und anderen wichtigen Details zum Spiel:

Sicaria Draconen: Fundort: äußerster Rand der Khomwüste (ostrand, L/B D° M, H° G, der Mann in der Schenke von Ackbah verrät den Fundort für 1500 Goldstücke

Ellenbogen: Fundort: Gaht. Der Waffenschmied (ELBE) muß nach dem Bogen gefragt werden, einfach erklären, daß man von Thyra (THYR) geschickt wird. Für 1000 Goldstücke ist der Bogen Euer.

Magische Axt: Fundort: tiefste Stelle im Meer zwischen Mubrak (befindet sich auf einer Insel an der Westküste der Insel) und dem Elfenwald.

HER MIT DEN TIPS

»Hallo Fans!« heißt diese Rubrik und wir wollen Euch darin zum Mitmachen anregen. Wenn Ihr einen Trick kennt, mit dem Ihr in Spielen schummelt oder das Spiel erleichtert, dann schreibt ihn auf und schickt ihn an:

MagnaMedia Verlag AG
Redaktion 64'er
Stichwort: Spieletips
Postfach 1304
85531 Haar b. München

Egal ob POKE, Cheat, Karten (bitte nur auf weißem Papier sauber gezeichnet oder mit Malprogramm) oder Paßwort. Ihr helft anderen Spielern über schwierige Klippen und habt außerdem die Chance, den Spieletip des Monats zu landen und dafür 100 Mark zu kassieren. Also Stifte und Drucker scharf gemacht und Euere heißen Tips und Tricks in den Postkasten.

Euer Jörn-Erik »Leo« Burkert

Cheats, Codes und POKEs

Hier die umfangreiche Mogel-Sammlung von Jörg Dahms, wobei alle undokumentierten POKEs für unendliche Leben stehen:

- Rick Dangerous** POKE 2793,173
- Creatures** POKE 18234,0
- Rolling Ronny** POKE 16515,177:POKE 165230,0
unlim. Energie
POKE 68,42:POKE 69,42
Money (4242 Cent)
POKE 20687,173 unlim. Weapons
POKE 6919,0 unlim. Time
- Test Drive II** POKE 25875,165
- R-Type** POKE 12887,96 keine Kollision
- Cross it** Reset: POKE 18190,12:SYS 19720
- Quest for Tires** RESET: POKE 40464,0:SYS 28684
- Devilrance** POKE 5635,173
- IO** RESET:POKE 25117,189
POKE 27015,10
Kollision aus
POKE 25707,2:POKE 25306,2
zwei Satelliten
POKE POKE 24949,X X=Level (1-4)
SYS 24576
- Rock 'n' Roll** POKE 19167,173
POKE 19300,173
unlim. Fallschirme
POKE 21314,189 jede Menge Schlüssel
POKE 20528,173 viele Werkzeuge
POKE 24464,173 viele Bomben
POKE 18113,173 viel Extra-Power
SYS 16640

Alle POKEs, die ohne SYS-Befehl zum Restart angegeben sind, können nur mit einem Multifunktionsmodul (Action-Replay oder Snapshot) eingegeben und wieder gestartet werden.

Paßwörter für Triget:

Level	Paßwort
5	ARGONIC
10	COLOGNE
15	WOBBLER
20	TRIPLEM



Oil Imperium:

Während des Bohrens einfach RUN/STOP-CTRL+PFEIL NACH LINKS drücken und jedes Gestein wird wie Butter.

Blasrohr: Einfach in Mubrak den Schmied nach BLAS fragen – Kosten: 200 Goldstücke.

Magische Rüstung: In Machaskan den Rüstungsmacher nach MAGI fragen. Dann einfach nur das Wort HELD sagen. Die Lösungen zum folgenden Quiz stehen im Anleitungsbuch zum Spiel (Kapitel 5, Seite 33).

Kompaß: In Mubrak in der Schenke verkauft ein Mann für 200 Goldstücke den Richtungsweiser.

Sextant: In der Kneipe von Thorwal gibt's einen Mann, der für 350 Goldstücke den Sextanten hergibt.

Glasflöte: Für 200 Goldstücke überläßt Euch der Bettler in Mubrak das Ding.

Belohnung: Sprecht den König auf das Wort MUBR an und gebt ihm die Koordinaten B° I, E° J und kassiert dafür 200 Goldstücke.

Der Schatz von UYRP: Die Koordinaten für die Kostbarkeiten gibt der Kartograph im Schloß. er residiert auf einer Insel, die sich an der Verlängerung der Khomküste.

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

Spirit of Adventure

Folge 1

Vor Jahrtausenden verspielten die Bewohner des Inselkontinents Lamarge die Gunst der ihnen bis dahin wohlgesonnenen Götter, indem sie ihnen nicht mehr opferten und zuletzt gar ihre Existenz anzweifelten ...

Zur Strafe brachten die Götter mehrere Katastrophen über das Land. Bei einer davon wurde ein altes Heiligtum zerstört. Seine Überreste sind über das ganze Land verteilt worden. Da diesem Tempel magische Kräfte innewohnten, verfügen auch seine Überreste noch über diese Eigenschaften. Sie wurden fortan als Runensteine benutzt. Langsam kehrte das Leben von Lamarge in geordnete Bahnen zurück. Gelehrte begannen, die Vergangenheit von Lamarge zu ergründen. Sie waren insbesondere an drei magisch verschlossenen Tempeln interessiert, die bis dahin nicht betreten werden konnten. Außerdem hatten sie die verschiedenen Runensteine in 18 Klassen eingeteilt. Später wurde auch das Geheimnis der Tempel gelüftet. Einem Abenteuer gelang es, einen von ihnen zu betreten, als er mit drei Runensteinen zu dem Tempel kam. In den Runentempeln war es möglich, mit Hilfe der Runensteine Zaubersprüche zu kreieren. Leider hatten sich auch böse Mächte Runensteine verschafft. So entstanden zwei rivalisierende Gruppierungen. Der Kult der Wissenden wollte die Kraft der Runen ergründen und diese dann zum Wohle der Bevölkerung einsetzen. In jeder Stadt wurde ein Tempel gegründet, dessen Vorsteher oder Vorsteherin zugleich auch die wichtigste politische Persönlichkeit verkörpert. Als Gegenstück zum Kult der Wissenden entstand die Bruderschaft der Träumer. Sie verbreiteten die schlimme Droge Opitar, die bereits nach einmaliger Einnahme unheilbar süchtig macht. Bereits jeder fünfte Bewohner von Lamarge ist der Dro-



ge verfallen. Die Bruderschaft der Träumer gewann immer mehr Macht in Lamarge, und der Kult der Wissenden konnte ihnen keinen Einhalt gebieten. Deshalb wurde eine Gruppe von sechs zunächst noch unerfahrenen Kämpfern ausgebildet, um die Bruderschaft der Träumer zu vernichten.

Die Abenteuerer

Diese Gruppe besteht aus je einem *Ordinary Warrior*, *Tidician Samurai*, *Ordinary Samurai* und *Allany Priest*. Dazu kommen eine *Allany Banshee* (ein weiblicher *Samurai*) und eine *Goddess* (weibliche Magierin). Diese Party startet im Tempel der Stadt Mooncity.

Das Abenteuer beginnt

Im Kloster von Mooncity spreche ich zunächst mit Rowena, der Stadtvorsteherin. Sie weist mich nochmals in meine Aufgabe ein: Mein Auftrag lautet nun, den Meister der Träumer zu finden und

seine teuflische Herrschaft zu zerstören. Rowena fügt an, daß ich im Falle eines Erfolgs mit einer hohen Belohnung rechnen könne. Dann erhalte ich von ihr die ersten Ausrüstungsgegenstände.

Der Meister der Träumer

Ich ergreife die Waffen und verlasse das Kloster. Meinen ersten Kampf habe ich gegen zwei Zwerge zu bestehen. Nachdem ich sie ziemlich erfolglos mit den Waffen attackiert habe, probiere ich die Angriffe auf geistiger Ebene aus. Jeder meiner Charaktere verfügt über ein gewisses Maß geistiger Energie. Sinkt sie auf Null, ist der Held wie beim Verlust der ganzen "normalen" Körperenergie (Hitpoints) tot. Diese geistige Stärke kann durch gegnerische Geistangriffe dezimiert werden. Genauso kann ich den Gegner durch diese gedanklichen Angriffe überwinden. Auf diesem Sektor sind die Zwerge nicht so stark und müssen die Waffen strecken. Ich erhalte meine ersten

Erfahrungspunkte. Außerdem erbeute ich einen Eisenschild und etwas Gold. In einem Laden erstehe ich für alle Charaktere ein Paar Schuhe. Bei einem weiteren Händler stocke ich meinen Vorrat an Rationen auf und besorge mir einige Fackeln. Auf einem Platz im Norden des Klosters bietet mir eine hübsche Banshee etwas Opitar an. Ich lehne ab, da ich ja gerade den Handel mit den Drogen unterbinden soll. Dann streife ich weiter durch die Stadt. Eine kurze Strecke südwestlich des Klosters wohnt ein weiser alter Mann, *Yakka Deepshaved*. Auf meine Anfrage weicht er mich in ein Gerücht über Rowena ein, demzufolge sie nachts durch die Stadt schleichen solle.

Außerdem verrät mir der Weise die ersten Runenkombinationen für Zaubersprüche. In "Spirit of Adventure" müssen die Sprüche erst in einem Runentempel kreiert werden. Dazu müssen drei Runen in einer bestimmten Kombination hintereinandergelegt werden. Er gibt die Reihenfolge einen Sinn,

erkennt der magiebegabte Charakter den Zauber an seiner Wirkung und trägt ihn in sein Zauberbuch ein, sofern der Spruch aus seiner Klasse stammt und der Charakter die erforderliche Stufe erreicht hat. Fortan kann er ihn jederzeit aussprechen – solange die Magiepunkte reichen. Die ersten sechs Runen habe ich bereits von Rowena erhalten. Nach dieser Audienz stöbere ich noch mehrere Händler mit verschiedenen Warenpaletten und einen Heiler im Südosten auf. Dann verlasse ich die Stadt durch das Nordtor und betrete die Wildnis von Lamarge. Während ich durch das Gelände streife, verbrauche ich regelmäßig Rationen. Doch zum Glück habe ich mir einen großen Vorrat angelegt. Nach einer kurzen Wanderung um einen Bergrücken sehe ich eine Burg vor mir. Neugierig betrete ich sie ...

Rialdos Burg

Die Grafik des Burglabyrinthes ist unter dem heutigen Standard. Die Wand besteht aus einer einfarbig weißen Platte ohne jede Abwechslung und ist in jedem der (insgesamt nur drei!) Dungeons gleich. Dennoch dringe ich tiefer in das Labyrinth vor. Nachdem ich durch eines der Felder, auf denen man gedreht wird, die

davon belebt wird. Verwirrt marschiere ich weiter. Am Ende eines Gangs stoße ich auf eine Diebesbande, die mich sofort angreift. Doch die drei Räuber sind keine ernste Bedrohung für mich, wie die meisten der (gut porträtierten) Monster auf Lamarge ebenfalls. Nachdem ich die Diebe getötet habe, finde ich bei ihren Leichen eine Schatzkarte, die eine Kombination von Farben zeigt. Wozu ich sie benutzen kann, ist mir jedoch noch unbekannt. Nach endloser Wanderung stoße ich auf zehn Skelette – vermutlich frühere Opfer der Diebesbande.

Einige Ecken weiter entdecke ich einige merkwürdige bunte Steinblöcke. Ich betätige sie in der auf der Schatzkarte angegebenen Kombination, und vor mir öffnet sich eine Geheimtür. Ich betrete die Kammer. Es ist das Versteck von Rialdos Schatz. Mehrere Runensteine, 2000 Goldstücke, 5000 Rationen, Rialdos Stab, eine Eislanze, ein Ring, ein Mithrilschild, ein Mystischer Umhang und eine Streitaxt gehen in meinen Besitz

für die Endgegner, absolut tödlich ist. Leider gelingt nicht jeder Angriff mit dem Stab. Auch die Eislanze verfügt über magische Eigenschaften. Mit ihr werden bei einer erfolgreichen Attacke einer ganzen Gruppe von Monstern jeweils rund 20 Hitpoints abgezogen.

Dann suche ich den Ausgang. Beim Verlassen des Gemäuers erhält jeder Charakter 2500 Erfahrungspunkte für den Schatzfund. Schnell kehre ich nach Mooncity zurück und lasse die Party heilen. Dann geht's wieder in die Wildnis zurück.

Energie einbringt. Dann verlasse ich das Schloß und wende mich nach Nordwesten. Ich streife eine Gebirgsflanke. Inmitten einer unüberschaubaren Ebene erblicke ich einen Tempel. Dieses heilige Bauwerk ist einer der insgesamt drei Tempel, in denen ich mit meinen Runen Zaubersprüche kreieren kann. Ich trete ein. Vor mir befindet sich ein Altar mit drei Vertiefungen, in denen ich die Runen plaziere. Nach einigem Probieren lernt Moon, der Kleriker, die beiden Sprüche Körperheilung und Geistheilung. Endlich kann ich auch ein wenig zaubern! Nachdem



Orientierung verloren habe, entdecke ich ein Mosaik, das eine Person zeigt, die aus einem Brunnen eine rote Flüssigkeit trinkt und

über. Mit Rialdos Stab kann ich einem Gegner zwischen 150 und 200 Hitpoints abziehen, was für alle Kreaturen auf Lamarge, auch

Schloß Attic und die ersten Zaubersprüche

Diesmal marschiere ich schnurstracks nach Westen. Bald gelange ich in einen Wald. In ihm steht ein prächtiges Gebäude – Schloß Attic. Ich werde freundlich empfangen. Nach einem zünftigen Mahl werden alle Charaktere gleich zweimal befördert, was ihnen vor allem mehr Hitpoints, geistige und magische

ich noch eine Weile erfolglos herumexperimentiert habe, kehre ich zum Abspeichern nach Mooncity zurück, da man nur in den Klöstern in den Städten den Spielstand sichern kann. Dann mache ich mich wieder nach Norden auf, die Insel zu erkunden. Auf einem Felsen inmitten einer Sandwüste finde ich den zweiten der Runentempel. Noch weiter nördlich, da, wo die Wüste in Wald übergeht, befindet sich ein Dimensionstor,

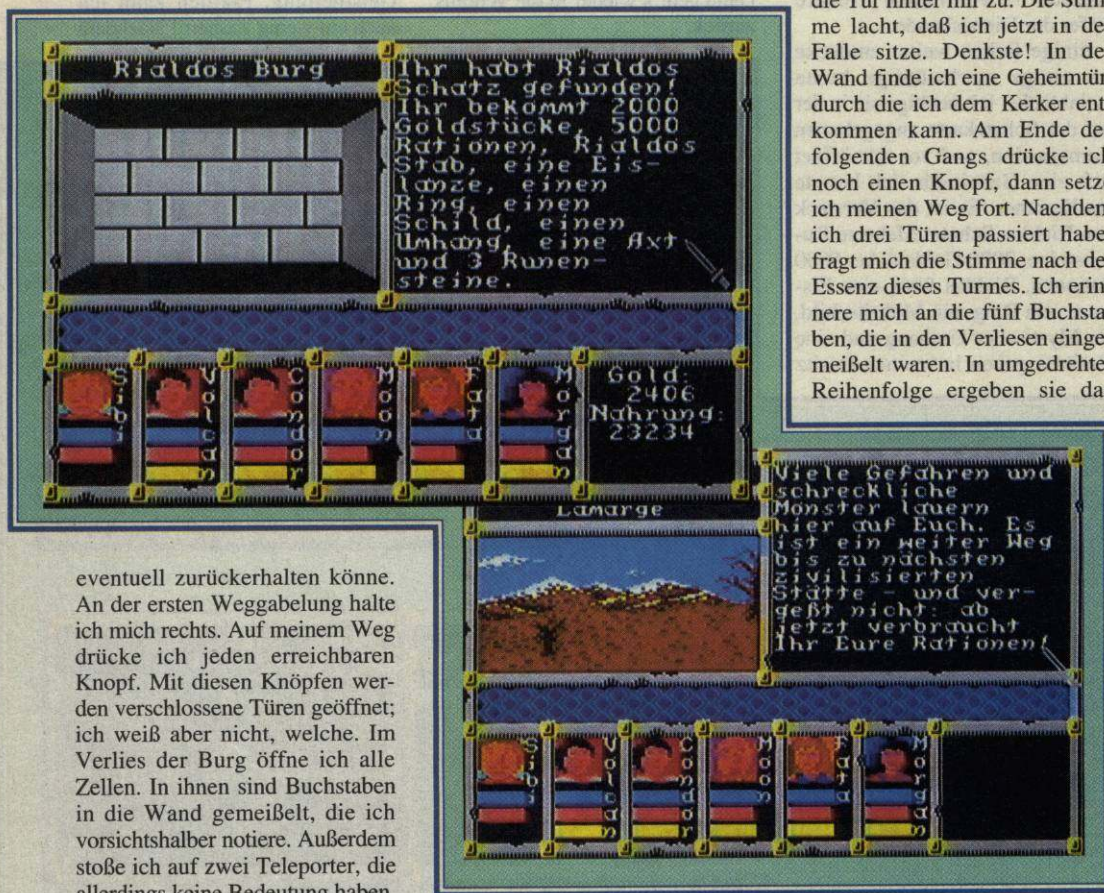
das ich aber noch nicht passieren kann. Dafür entdecke ich einen Pfad, der mich tiefer ins Gebirge im Westen führt. An seinem Ende befindet sich ein Turm.

Selgans Turm

Ich betrete ihn und stelle fest, daß es sich um Selgans Turm handelt. Vor der ersten Tür begrüßt mich eine geheimnisvolle Stimme und nimmt mir quasi als Eintrittsgeld eine Rune ab. Die Stimme meint jedoch, daß ich sie später

nem Schild zu lesen, daß eine kleine Reise lockt. Neugierig betrete ich die Nische – und werde von einem Teleporter in ein mir unbekanntes und ohne den Teleporter nicht erreichbares Gebiet transportiert. Ich beginne sofort mit der Erforschung dieses Gebiets. In einer Ecke stoße ich auf einen Penner. Er lallt, ich solle umkehren und dann immer geradeaus gehen. Obwohl er sich ziemlich merkwürdig verhält, folge ich seinem Rat. Ich durchquere eine Geheimgtür. Hinter einer wei-

storbenen Abenteurers einen Diamanten, eine Rune und ein Basterschwert. Nachdem ich noch einen Knopf gedrückt habe, enttarne ich die Geheimgtür und steige auf der dahinterliegenden Treppe in den dritten Level hinauf. Die Stimme warnt mich und will mich zur Umkehr bewegen – jedoch ohne Erfolg. Ich betrete die erste Abzweigung nach rechts und betätige dort zwei Knöpfe, die anscheinend spätere Türen für mich öffnen. Nachdem ich eine Folterkammer besichtigt habe, schlägt auf einmal die Tür hinter mir zu. Die Stimme lacht, daß ich jetzt in der Falle sitze. Denkste! In der Wand finde ich eine Geheimgtür, durch die ich dem Kerker entkommen kann. Am Ende des folgenden Gangs drücke ich noch einen Knopf, dann setze ich meinen Weg fort. Nachdem ich drei Türen passiert habe, fragt mich die Stimme nach der Essenz dieses Turmes. Ich erinnere mich an die fünf Buchstaben, die in den Verliesen eingemeißelt waren. In umgedrehter Reihenfolge ergeben sie das



eventuell zurückerhalten könne. An der ersten Weggabelung halte ich mich rechts. Auf meinem Weg drücke ich jeden erreichbaren Knopf. Mit diesen Knöpfen werden verschlossene Türen geöffnet; ich weiß aber nicht, welche. Im Verlies der Burg öffne ich alle Zellen. In ihnen sind Buchstaben in die Wand gemeißelt, die ich vorsichtshalber notiere. Außerdem stoße ich auf zwei Teleporter, die allerdings keine Bedeutung haben, da sie mich lediglich innerhalb der unteren Ebene hin- und hertransportieren. Dann kehre ich zum Eingang zurück und wähle nun an der ersten Verzweigung den anderen Gang. Hinter einer Tür liegt ein Zettel, der mich warnt, daß Grubengas mir die Orientierung nehmen kann. Als ich vor einer Wand stehe, fällt mir eine Schrift auf: "Wenn ein Fels kein Fels ist..." – dann ist es eine Geheimgtür! Und richtig, an dieser Stelle kann ich durch die Wand laufen. Am Ende des Schleichwegs steht eine Leiter, die mich in den zweiten Level führt. Dort betätige ich zunächst zwei Knöpfe, von denen sich einer hinter einer unsichtbaren Tür am Ende eines langen Gangs befindet. Daraufhin geben zwei gläserne Wände einen Gang frei. Vor einer Nische ist auf ei-

teren Tür wird eine Falle ausgelöst. Aus den Wänden strömt Giftgas, das der gesamten Party zwanzig Punkte Schaden zufügt. Als Lohn für die überstandene Gefahr entdecke ich hier die Rune, die mir durch die Stimme abgenommen worden war. Diese Stimme meldet sich auch jetzt wieder mit einem hämischen Kommentar. Dann kehre ich wieder um. In einem weiteren Gang entnehme ich einem Schnipsel einer Karte, daß sich die nach oben führende Leiter hinter einer Geheimgtür befindet. Später spricht mich die Stimme erneut an. Sie erinnert mich an die Buchstaben aus dem Gefängnis, teilt mir jedoch nicht mit, wozu ich sie gebrauchen kann. Tiefer in diesem Gang finde ich in den Überresten eines ver-

storbenen Abenteurers einen Diamanten, eine Rune und ein Basterschwert. Nachdem ich noch einen Knopf gedrückt habe, enttarne ich die Geheimgtür und steige auf der dahinterliegenden Treppe in den dritten Level hinauf. Die Stimme warnt mich und will mich zur Umkehr bewegen – jedoch ohne Erfolg. Ich betrete die erste Abzweigung nach rechts und betätige dort zwei Knöpfe, die anscheinend spätere Türen für mich öffnen. Nachdem ich eine Folterkammer besichtigt habe, schlägt auf einmal die Tür hinter mir zu. Die Stimme lacht, daß ich jetzt in der Falle sitze. Denkste! In der Wand finde ich eine Geheimgtür, durch die ich dem Kerker entkommen kann. Am Ende des folgenden Gangs drücke ich noch einen Knopf, dann setze ich meinen Weg fort. Nachdem ich drei Türen passiert habe, fragt mich die Stimme nach der Essenz dieses Turmes. Ich erinnere mich an die fünf Buchstaben, die in den Verliesen eingemeißelt waren. In umgedrehter Reihenfolge ergeben sie das

Volker Siebert, Lutz Nowack/lb

Titel: Spirit of Adventure,

Preis: 39 Mark,

Bezugsquelle: Data House

Dietrich, 34246 Vellmar

Fortsetzung im nächsten Heft

64'er-Longplay

Wer eine Komplettlösung zu einem Spiel sucht, findet in dieser mittlerweile beeindruckenden Liste alle Longplays in der Geschichte des 64'er-Magazins auf einen Blick.

- 4/89: Uridium II
- 5/89: Last Ninja II (Teil 1)
- 6/89: Ghosts'n Goblins
- 7/89: Katakis
- 8/89: Last Ninja II (Teil 2)
- 9/89: Wizball
- 10/89: Grand Monster Slam
- 11/89: Zak McKracken (Teil 1)
- 12/89: Spherical
- 1/90: Zak McKracken (Teil 2)
- 2/90: Oil Imperium
- 3/90: Ultima (Teil 1)
- 4/90: Ultima (Teil 2)
- 5/90: Ultima (Teil 3)
- 6/90: Elite
- 8/90: X-Out
- 11/90: Maniac Mansion
- 12/90: Turrigan
- 1/91: R-Type
- 2/91: Dragon Wars (Teil 1)
- 3/91: Dragon Wars (Teil 2)
- 4/91: Pirates
- 5/91: Bard's Tale (Teil 1)
- 6/91: Bard's Tale (Teil 2)
- 7/91: Turrigan II (Teil 1)
- 8/91: Turrigan II (Teil 2) und Secret Silver Blades
- 9/91: Turrigan II (Teil 3) und The Last Ninja
- 10/91: Bard's Tale 2 (Teil 1)
- 11/91: Bard's Tale 2 (Teil 2) und Saint Dragon
- 12/91: Armalyte (Teil 1)
- 1/92: Bard's Tale 2 (Teil 3)
- 2/92: Bard's Tale 2 (Teil 4) und Armalyte (Teil 2)
- 3/92: Last Ninja 3 (Teil 1)
- 4/92: Defender of the Crown
- 5/92: Buck Rogers
- 6/92: Pool of Radiance Teil 1
- 7/92: Pool of Radiance Teil 2
- 8/92: IO
- 9/92: Dirty
- 10/92: Curse of the Azure Bonds
- 11/92: Ultima 6 (Teil 1)
- 12/92: Ultima 6 (Teil 2)
- 1/93: King's Bounty
- 2/93: Creatures 2
- 3/93: Crime Time
- 4/93: Soul Crystal
- 5/93: Catalypse (Teil 1)
- 6/93: Catalypse (Teil 2)
- 7/93: Elvira 2 (Teil 1)
- 8/93: Elvira 2 (Teil 2)
- 9/93: Times of Lore (Teil 1)
- 10/93: Times of Lore (Teil 2)
- 11/93: First Samurai (Teil 1)
- 12/93: First Samurai (Teil 2)
- 1/94: Elvira – Mistress of the Dark 2 u. 3/94: Centauri Alliance 4 u. 5/94: Rick Dangerous 7/94: Die Prüfung

Spiele-Sonderhefte:

- Top Spiele 2: Bard's Tale 3 Zak McKracken
- Top Spiele 3: Turrigan Death Knights of Krynn
- Top Spiele 4: Maniac Mansion Gateway to the Savage Frontier

Unsere Anschrift:

MagnaMedia Verlag AG
Redaktion 64'er
Stichwort: Longplay
Postfach 1304
85531 Haar bei München

Motley Tetris

Kunterbunte Steine

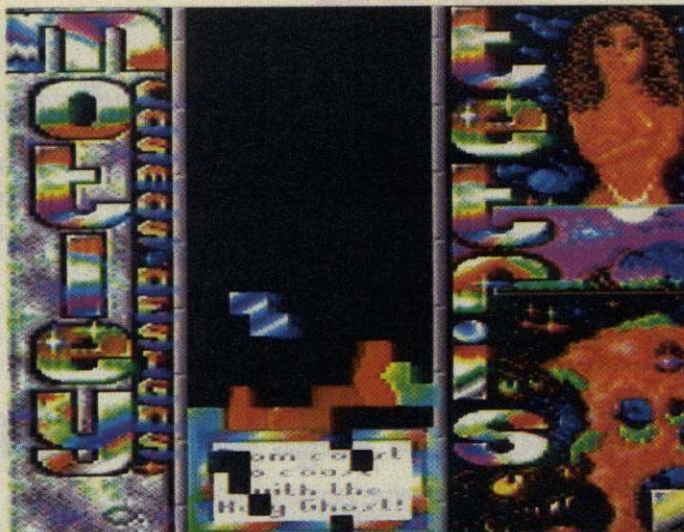
Onach wie vor ist die Spiel-erfindung "Tetris" des Russen Alexej Pajitnow ungeschlagen. Kein Computersystem blieb vom Denkspiel rund um die Klötzchen bisher verschont und die Anzahl der Clones wächst ständig.

So auch auf dem C 64 und man fragt sich, was Cosmos Designs noch in ihrer Version besser machen will. Gleich vorab – das Game bietet keine neuen Features und hält sich an das Vorbild. Dafür wurde beim Outfit kräftig zugelegt und das geniale Spielprinzip des Russen mit einem zeitgemäßen Mantel versehen. Um das Kraut so richtig fett zu machen, wird die Spielfläche in FLI auf den Bildschirm gebracht, was für eine flexiblere Farbverteilung sorgt. Zu allem Überfluß wurde die Re-

gel gebrochen, daß sich FLI-Grafiken und Sprites auf dem Bildschirm zugleich nicht vertragen. Scheinbar geht's doch und das macht den freien Fall bewußter Steinchenperfekt, der durch die knackigen Sounds hervorragend ergänzt wird.

Egal, ob der Dreh mit den Sprites und FLI nun fauler Zauber ist,

Motley Tetris	
64'er	8
WERTUNG	von 10
Spielidee	<input type="checkbox"/>
Grafik	<input type="checkbox"/>
Sound	<input type="checkbox"/>
Schwierigkeit	steigend



Bunt und suchtfördernd: Motley Tetris von Cosmos Designs

einige harte Freaks gleich wieder zu lamentieren beginnen und ihre Einwände vorbringen – das Game macht Spaß und gefällt optisch. Außerdem soll der sagenhafte Preis von knapp acht Mark an dieser Stelle mit einem dicken Lob bedacht werden. Hier sollte jeder interessierte Spieler zuschlagen, denn für dieses Trinkgeld ist das

Spiel wie geschenkt. Vor allem weil das Original-Tetris von Spektrum Holobyte nur noch mit viel Glück zu ergattern ist.

Jörn-Erik Burkert

Titel: Motley Tetris,

Preis: 7,80 Mark,

Bezugsquelle: Talentebank,

A-9020 Klagenfurt



Jetliner glänzt zwar nicht mit Prachtgrafik und Bombensound, dafür bietet die Simulation Spieltiefe und fordert Wissen über die Fliegerei

Wie Piloten ihre Riesenmaschinen in die Luft bekommen, die Passagiere rund um die Welt transportieren und die Metallvögel heil wieder auf den Boden bringen, fasziniert viele. Seit es Computer gibt, sind deshalb Flugsimulationen immer wieder ein Thema für Programmierer. In "Jetliner" ist man Käpt'n einer Verkehrsmaschine und düst durch ganz Europa. Bevor aber der Jet in die Luft geht, heißt es erst einmal den Flug richtig vorzubereiten. An Hand von Karten sucht man sich

den Kurs aus und legt mit Funkfeuern markante Punkte auf der Strecke fest. Außerdem bestimmt man die Art und Menge der Fracht. Hier kann zwischen Waren und Passagieren gewählt werden. Dann folgt die Startvorbereitung, das Auftanken und man geht in die Luft. Am Zielflughafen muß die Maschine logischerweise samt Fracht und Passagiere "am Stück" auf den Boden zurückgebracht werden.

Die Simulation "Jetliner" kann schon jeder Spieler vergessen, der

Jetliner

Einmal Käpt'n sein

auf einen Blick aus dem Cockpit mit sanft scrollenden Landschaften hofft. Das Game zeigt sich spartanisch bei Grafik und Sound. Dafür wurde von den Entwicklern mehr Wert auf flugtechnische Details gelegt. Man muß erst einmal kräftig die umfangreiche Anleitung studieren und sich in die vielen Karten und Tabellen einarbeiten. Außerdem sollte man sein Cockpit aus dem Effeff kennen, damit man alle relevanten Werte und Probleme zur rechten Zeit erkennt und darauf reagieren kann. An Hand des Spiels kann man nur ahnen, wieviele trockene Ausbildungsstunden ein Pilot ertragen muß, ehe er erstmals in einem Cockpit sitzt. Jetliner ist nur für die Spieler geeignet, die sich extrem für das Flugwesen und alles was daran hängt interessieren. Manchmal hat man den Eindruck, daß die Entwickler beim Spieler schon einen Flugschein voraussetzen. Das bemerkt man frühestens, wenn man

die Karten zum Spiel studiert. Hat man sich aber in den Inhalt des Games eingearbeitet und die notwendige Geduld, steigt man schnell in die Luft und dürfte die Grundlagen des Flugwesens beherrschen. Action-Freaks aber sollten sich zunächst nach Alternativen umsehen.

Jörn-Erik Burkert

Titel: Jetliner,

Preis: 39 Mark,

Bezugsquelle: Independent Softworks,

79199 Burg am Wald

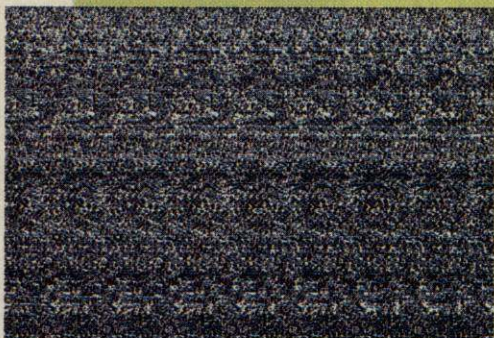
Jetliner	
64'er	7
WERTUNG	von 10
Spielidee	<input type="checkbox"/>
Grafik	<input type="checkbox"/>
Sound	<input type="checkbox"/>
Schwierigkeit	hoch

**DIE NÄCHSTE AUSGABE
ERSCHEINT AM 30.9.94**

Soundgrundlagen

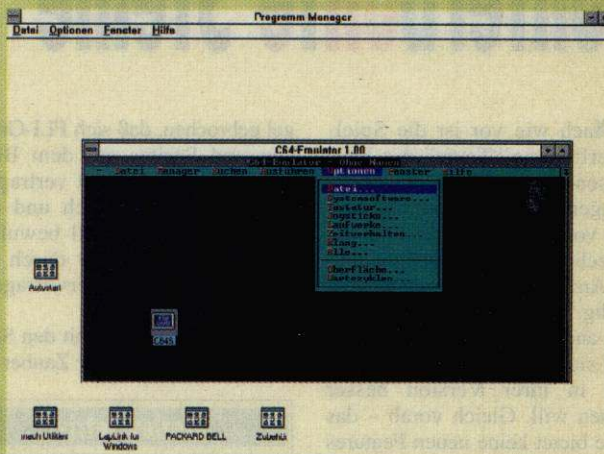
Sie wollten schon immer Musik im Hintergrund Ihres Programms haben? Alle Grundlagen, Tricks und Programmierkniffe in der nächsten Ausgabe des 64'er-Magazins.

Das magische Auge auf dem C64



Faszinierende 3-D-Bilder mit dem C 64

Wildes Chaos auf dem Computerbildschirm! Aber mit einem Trick und Konzentration werden die konfuse Grafiken zu 3-D-Bildern. Mit einem GoDot-Modul kommt die Faszination des Buch-Bestsellers auch auf den C-64-Bildschirm!



Der C 64 ist nicht totzukriegen. PC-Emulatoren bringen den Klassiker auch auf die großen Kisten.

C-64-Emulatoren im Vergleichstest

Die Emulator-Programmierer haben wieder zugeschlagen: Fast gleichzeitig erschienen der "C64S" von Miha Peternel und der "C-64-Emulator" von Wolfgang Lorenz in neuer und endgültiger Version 1.0. Beide Programme werden als Shareware vertrieben – wir haben getestet, ob sie einen C 64 ersetzen können.

SORRY, WERBUNG GESPERRT!

64ER

WWW.64ER-ONLINE.DE



Genloc: drei parallaxscrollende Level voller Action, vielen Gegnern und jeweils einem gefährlichen Obermotzen

Genloc

Endlich ist es soweit: Nach langer Odyssee präsentieren wir im nächsten Monat das Action-Spiel "Genloc" aus der Feder von Michael Strelecki. Dann geht's dem hinterlistigen Doktor Klein an den Kragen, der mit seiner Mutanten-Crew die Welt beherrschen will.

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE

SORRY, WERBUNG GESPERRT!

G4ER ONLINE



WWW.G4ER-ONLINE.DE