

GENIE
TRS-80
USER CLUB
" BREMERHAVEN "

C L U B I N F O C L U B I N F O

Bitte an alle HRG'ler : Bitte schickt mir Ausdrücke, die ich als Titelbild verwenden kann (sonst ist diese Seite bald kahl)!!!

Das Titelbild stammt diesmal aus dem INFO des Münchener TRS-80 User Clubs.



4. JAHRGANG 07. AUSGABE JULI 1986

REDAKTION: RALF FOLKERTS, NUTTHORNER STR. 9, 2075 BUCKHOLMBERG
PRINTED 1986 BY PETER SPIESS, RENNERTSHOFEN
AUFLAGE: 095 EXEMPLARE

INHALT

CLUB INFO

IN DIESEM HEFT FINDET IHR FOLGENDE ARTIKEL:

- 2 Inhaltsverzeichnis
- 2 Fragebögen
- 3 Internes vom Betreuer
- 4 - 6 Spooler von Paul - Jürgen Schmitz
- 4 Wichtige Frage an alle !! (MS-DOS Ecke)
- 7 - 9 Frame von Kajot Mühlenbein
- 9 Mailbox Corner: Neuer User in C.I.A. Mailbox !! (Jubel)
- 10 - 14 Multidos Testbericht von: Mir
- 15 - 22 Handler von Paul Kröher
- 16 Hinweis zum Thema Anzeigen (und Portoersparnis)
- 23 - 24 Grafik JKL für das Genie IIIs von Arnulf Sopp
- 25 'Ganz schön verschaukelt'
- 26 - 28 Psychologische Aspekte des Programmierens
- 29 - 30 Mehr Fernesehen für's Geld von Arnulf Sopp
- 31 - 32 Personelles im INFO

Ich danke allen 'Autoren' und denjenigen Mitgliedern, die mir Artikel zugeschickt haben recht herzlich.

- Fragebögen

Also wirklich,

ich habe bis heute (13.07.1986) erst 34 Fragebögen zurückerhalten. Das ist noch nicht einmal die Hälfte. Unter diesen Umständen habe ich bisher von der Auswertung der Bögen abgesehen - und werde auch nicht eher mit der Auswertung anfangen, bis ich mindestens 40 Stück (das ist so ziemlich die Hälfte) hier liegen habe. Ich fordere daher noch einmal alle auf, die es bisher noch nicht getan haben, doch bitte die Bögen jetzt auszufüllen und zurückzusenden.

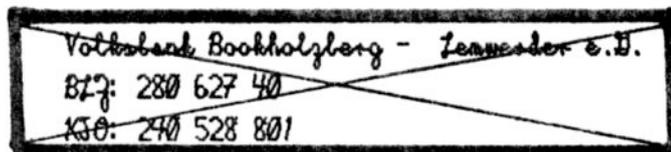
INTERNES VON BETREUER

Bankverbindung des Clubs:

Postgiroamt Hannover

BIZ: 250 100 30

KSO: 5342 84 - 308



Bitte, wenn möglich, das Postgirokonto benutzen (ich danke allen, die dies bereits getan haben).

Hallo Leute,

entl.

diese Ausgabe erscheint leider ein paar Tage später als erwartet. Der Grund ist die Versendungsart. Peter hat die INFOS ja immer als Büchersendung verschickt. Da meiner Meinung nach keine Gründe dagegen sprachen, führte ich diese Maßnahme fort. Nachdem ich jedoch die Hefte am 27.06. in den Briefkasten geworfen hatte, erhielt ich einen Anruf vom Postamt. Der Grund: Die INFOS könnten nicht als Büchersendung verschickt werden; ausnahmsweise seien die Hefte aber noch weggeschickt worden. Als Folge davon ging ich zuerst einmal zur Post, um mich zu erkundigen, was gegen den Versand als Büchersendung spräche. Der Grund war die Werbung (private Anzeigen und die Anzeigen von Peter), die nicht in einer Büchersendung erscheinen dürfe. Da aber selbst ohne Werbung nicht sicher war, ob es eine Büchersendung ist, brachte ich ein Exemplar des INFOS zur Post, wo es zur Oberpostdirektion eingeschickt wurde. Das genaue Ergebnis dieser Prüfung werde ich dann morgen erfahren; dann gehe ich wieder zur Post. Ihr könnt dann selber sehen, ob ich diese Ausgabe als Büchersendung verschicken kann, oder ob es eine Drucksache ist (DM 0,20 teurer).

Während des Einbindens der Hefte unterlaufen mir immer wieder Pannen. In der Mai Ausgabe waren 43 Hefte statt links leider rechts eingeleimt; das Lesen wurde dadurch erschwert. In der Juni Ausgabe hatte ich den Leim zu stark verdünnt, weshalb er weiter in das Papier eindrang als geplant. Hier war das Umblättern dann eine Kunst für sich. Ich hoffe nur, daß mir in diesem Heft nicht noch so ein Fehler unterläuft.

Die letzte Ausgabe ist leider wegen der Banksache etwas aus den Fugen geraten; ich hatte das Heft gerade komplett hier liegen, als mir der Brief von der Bank ins Haus flatterte. Da ich nicht alles neu drucken wollte, habe ich versucht, alles 'so' wieder hinzubiegen. Der Multidos Testbericht ist daher auch in diesem Heft; ich habe den Update vor zwei Wochen erhalten, kann also einiges von seinen Fähigkeiten berichten.

Ach ja, da Peter im September in seinen Urlaub geht, kann ich nicht genau sagen, wann die September - Ausgabe kommt.

Spooler für das Genie - eine kurze Lösung

Zum Thema Spooler ist ja schon mehrfach etwas geschrieben worden. Bekanntlich sind dies Programme, die auszudruckende Zeichen solange irgendwo im RAM Zwischenspeichern, bis das Ausgabegerät diese Daten weiter verarbeiten kann.

Das folgende Programm soll diese Aufgabe mit minimalem Aufwand realisieren, d.h. kurz und knapp, selbst wenn dies bedeutet, daß auf ergänzende Eigenschaften verzichtet werden muß. (Dazu gibt's ja z.B. schon den vorzüglichen Spooler, den früher einmal Arnulf Sopp im Info veröffentlicht hat). Das Programm kann wohl noch gut mit dem ASM eingetippt werden.

Zum Ablauf: Das Programm wird zunächst in den Speicher geladen. Die Adresse bei E000 kann gegebenenfalls auch woanders hin verlegt werden. Da das Programm auf das EG6A ausgelegt ist, wird dieser Banker zunächst geschaltet und danach der eigentliche Spooler in den RAM Bereich parallel des Level A ROM kopiert. Ein Aufruf von system /123A5 ist danach natürlich nicht mehr möglich. Vier möchte, kann die Adresse (Zeile 7) auch verändern, z.B. um nur den oft überflüssigen Monitor (system /13 5 7 9) zu überschreiben .

Zeile 21 bis 27 dient der Verbiegung der Adressen, so daß danach alle Ausgaben auf den Spooler umgeleitet und eine Abfrage der Tastatur (bis zum Spoolerüberlauf) stattfindet.

Der Spooler selbst benutzt den Speicherbereich 0000 bis 2FFF, er erlaubt demnach 12287 Bytes für den Drucker bereit zu halten.

Ich habe meinen Spooler in das Auto-Kommando des DOS eingebunden, so daß er nach dem Booten gleich zur Verfügung steht.

Viele Grüße
Paul-Jürgen Schmitz

GESUCHT GEFUNDEN FRAGEN

FRAGEN VON MITGLIEDERN AN MITGLIEDER FUER MITGLIEDER

Große Frage, den Fortschritt betreffend:

In der letzten Zeit fragen immer wieder Mitglieder bei mir an, ob wir nicht eine MS-DOS - Ecke im INFO einrichten sollten, da unsere 'guten alten' 8bitter ja doch im Aussterben begriffen sind, und die MS-DOS Geräte 'groß kommen'. Im Prinzip bin ich durchaus für eine solche MS-DOS Ecke. Ich meine jedoch, daß sie sich auf die Geräte beschränken sollte, die entweder 'GENIE' oder 'Tandy' im Typenschild stehen haben. Meiner Meinung nach sollten wir jetzt nämlich nicht vom Genie/TRS-80 Club zu einem IBM, Commodore, Plantron, Tandberg, ... Club werden.

Ich bitte Euch um eine Stellungnahme zu diesem Thema. Falls die Resonanz gering bleibt gehe ich davon aus, daß Ihr mit einer MS-DOS Ecke einverstanden seid. In dem Fall werde ich, sobald ich entsprechende Artikel habe, diese in's INFO bringen. Voraussetzung ist dafür natürlich auch, daß diese Artikel wirklich nur so um die 1 - 2 Seiten je Heft lang sind. Wie gesagt: Schreibt mir bitte, was Ihr davon haltet.

```

00001 ;*****
00002 ;* DRUCKERSPOOLER FÜR EG64 UND GENIE I/II *
00003 ;* Version von Jürgen Schmitz, Eschborn *
00004 ;* Mai 1986 *
00005 ;* Filename ist: SPOOLER/asm ==) SPOOLER/cmd *
00006 ;*****
3010 00007 ADR EQU 03010H ;Adresse Anfang L-4 RAM
00008 ;kann geändert werden!
E000 00009 ORG 0E000H ;Neuer START
00010 ;
E000 3E0A 00011 START LD A,0AH ;read Level 4 RAM
E002 D3DF 00012 OUT (ODFH),A ; und schalten!
E004 3C 00013 INC A ;write Level 4 RAM
E005 D3DF 00014 OUT (ODFH),A ; und schalten!
00015 ;----- Kopiere Spooler nach L 4 RAM -----
E007 111030 00016 LD DE,ADR ;Zieladresse wie festgel.
E00A 2125E0 00017 LD HL,PRINT ;Anfangadresse
E00D 017F00 00018 LD BC,MAXL-PRINT ;Anzahl Verschiebebytes
E010 EDB0 00019 LDIR ; * Verschieben *
00020 ;----- Verschiebelogik für Spooler -----
E012 211030 00021 ANEG LD HL,PRINT-OFFSET
E015 222640 00022 LD (4026H),HL ;DRUCKER DCB
E018 2A1640 00023 LD HL,(4016H) ;TASTATUR DCB
E01B 228930 00024 LD (OLDDRV-OFFSET),HL
E01E 214730 00025 LD HL,TASTA-OFFSET
E021 221640 00026 LD (4016H),HL
E024 C9 00027 RET ;zurueck zum Aufruf!
00028 ;----- Unterprogramm Ausdruck -----
B015 00029 OFFSET EQU $-ADR ;Verschiebebytes
E025 79 00030 PRINT LD A,C
E026 B7 00031 OR A
E027 C8 00032 RET Z ;Statusabfrage
00033 ;----- RAM schalten -----
E028 F3 00034 DI ;Interrupts aus!
E029 3E09 00035 LD A,09H ;RAM 0000 - 2FFFschreiben
E02B D3DF 00036 OUT (ODFH),A
00037 ;----- Speichern -----
E02D 2A8B30 00038 LD HL,(FREI-OFFSET)
E030 71 00039 LD (HL),C ; SPEICHERN
E031 23 00040 INC HL ;Speicheradresse erhöhen
00041 ;----- Testen -----
E032 7C 00042 LD A,H
E033 FE30 00043 CP 30H ;BUFFERUEBERLAUF 3000H ??
E035 201C 00044 JR NZ,RET1 ;BUFFER NOCH NICHT VOLL
00045 ;----- Drucken -----
E037 2A8D30 00046 LD HL,(ZEIG-OFFSET)
E03A 7C 00047 DRLOOP LD A,H
E03B FE30 00048 CP 30H ;BUFFERUEBERLAUF 3000H ??
E03D 280E 00049 JR Z,RET2 ;ALLES GEDRUCKT
E03F DBFD 00050 WAIT IN A,(OFDH) ;STATUS LESEN
E041 E6F0 00051 AND OFOH
E043 FE30 00052 CP 30H
E045 20F8 00053 JR NZ,WAIT ;BIS DRUCKER BEREIT
E047 7E 00054 LD A,(HL) ;ZEICHEN HOLEN
E048 D3FD 00055 OUT (OFDH),A ; UND DRUCKEN
E04A 23 00056 INC HL ;NAECHSTES BYTE
E04B 18ED 00057 JR DRLOOP
00058 ;----- Unterprogramm Speichern/Drucken -----
E04D 210000 00059 RET2 LD HL,0000
E050 228D30 00060 LD (ZEIG-OFFSET),HL
E053 3E01 00061 RET1 LD A,01H ;ROM einblenden schreiben
E055 D3DF 00062 OUT (ODFH),A
E057 228B30 00063 LD (FREI-OFFSET),HL
E05A FB 00064 EI ;Interrupts wieder ein!
E05B C9 00065 RET ; Ende Unterprogramm
00066 ;-----
E05C C5 00067 TASTA PUSH BC
E05D D5 00068 PUSH DE
E05E F5 00069 PUSH AF

```

E05F	3A4038	00070	LD	A,(3840H)	;TASTATUR	
E062	CB57	00071	BIT	2,A		
E064	2027	00072	JR	NZ,RESET	;DRUCK ABBRECHEN ?	
E066	2A8D30	00073	LD	HL,(ZEIG-OFFSET)		
E069	ED5B8B30	00074	LD	DE,(FREI-OFFSET)		
E06D	B7	00075	OR	A		
E06E	ED52	00076	SBC	HL,DE		
E070	281B	00077	JR	Z,RESET	;DRUCKERBREICH LEER ?	
E072	DBFD	00078	IN	A,(OFDH)		
E074	E6F0	00079	AND	OFOH	;STATUS LESEN	
E076	FE30	00080	CP	30H		
E078	201C	00081	JR	NZ,WEITER	;DRUCKER BUSY	
E07A	2A8D30	00082	LD	HL,(ZEIG-OFFSET)		
E07D	F3	00083	DI		;Interrupts aus!	
E07E	3E08	00084	LD	A,08H	;BASIC-RAM einblenden	
E080	D3DF	00085	OUT	(ODFH),A	; lesen	
E082	7E	00086	LD	A,(HL)	;ZEICHEN HOLEN	
E083	D3FD	00087	OUT	(ODFH),A	; UND DRUCKEN	
E085	23	00088	INC	HL		
E086	228D30	00089	LD	(ZEIG-OFFSET),HL	;ZEIGER AUF NAECHSTES Z.	
E089	ED52	00090	SBC	HL,DE		
E08B	2009	00091	JR	NZ,WEITER		
E08D	210000	00092	RESET	LD	HL,0000H	
E090	228D30	00093	LD	(ZEIG-OFFSET),HL	;Pointer zurück	
E093	228B30	00094	LD	(FREI-OFFSET),HL		
E096	AF	00095	WEITER	XOR	A	;ROM EIN für lesen!
E097	D3DF	00096	OUT	(ODFH),A		
E099	FB	00097	EI		;Interrupts wieder ein!	
E09A	F1	00098	POP	AF		
E09B	D1	00099	POP	DE		
E09C	C1	00100	POP	BC		
E09D	C30000	00101	JP	0000H	;TASTATUR ABFRAGEN	
E09E		00102	OLDDRV	EQU	\$-2	
EOA0	0000	00103	FREI	DEFW	0000H	;Zeiger freier Bereich
EOA2	0000	00104	ZEIG	DEFW	0000H	;Zeiger nächstes Byte
EOA4		00105	MAXL	EQU	\$;ENDE DES SPOOLERS**
E000		00106	END	START		

00000 Fehler
31821 Zeichen verfügbar

ADR	3010	00007	00016	00029	
ANFG	E012	00021			
DRLOOP	E03A	00047	00057		
FREI	EOA0	00103	00038	00063	00074 00094
MAXL	EOA4	00105	00018		
OFFSET	B015	00029	00021	00024	00025 00038 00046 00060 00063 00073 00074 00082 00089 00093 00094
OLDDRV	E09E	00102	00024		
PRINT	E025	00030	00017	00018	00021
RESET	E08D	00092	00072	00077	
RET1	E053	00061	00044		
RET2	E04D	00059	00049		
START	E000	00011	00106		
TASTA	E05C	00067	00025		
WAIT	E03F	00050	00053		
WEITER	E096	00095	00081	00091	
ZEIG	EOA2	00104	00046	00060	00073 00082 00089 00093

Dies ist keine traurige Nachricht,
sondern eine erfreuliche !

Es ist der Ausdruck eines neuen BASIC-Befehls :

* * * ' F R A M E ' * * *

* * * **Hallo Clubfreunde !** * * *
* * * **Seid Ihr im Bilde ?** * * *

In meinem Bemühen, mich mit der Z80-Maschinensprache anzufreunden (warum eigentlich?) habe ich einen ganzen Stoß mit Assembler-Listings durchgeblättert, die ich einmal aus Computer-Zeitschriften entnommen und aufgehoben habe in der halbweisen Voraussicht, daß ich mich eines Tages damit befassen würde. Dieser (diese) Tag(e) ist (sind) offenbar gekommen. Dem Rate weiser Kollegen folgend, versuche ich - nachdem der Vokabelschatz einigermaßen beherrscht wird - mich da hineinzudenken und Z80's Schleichwege nachzuvollziehen... (Übrigens: mit wachsendem Erfolg.)

Dabei stieß ich auf eine "Utility" (aus Unsicherheit über deren Nutzen sagen wir ja nicht "Nützlichkeit" oder gar "Einrichtung"...), wie man den BASIC-Wortschatz erweitern kann. Das ist sicher für Phantasie-Begabte, denen 123 TOKENs zuwenig sind, von Interesse.

Dargestellt wird das Vorgehen dort (in der BOMICRO 9/83) an einem Beispiel, wie man schlagartig einen Rahmen entlang der Begrenzung des Bildschirms erzeugen und somit das auf diesem Produzierte würdigen und ehren kann - ohne Rücksicht auf dessen wahren Wert; denn unser Sklave ist Gottlob noch nicht kritisch (der Tag kommt noch, auch hier - wartet's ab!) Dieser Rahmen kann im BASIC-Programm durch ein einziges neues Befehlswort aufgerufen werden, nachdem man diesem einen Namen gegeben; hier sinnigerweise: "FRAME".

Euch wird weniger dieses "Rahmen-Programm" interessieren (Bleistift und Lineal kosten zwar Zeit und Mühe, sparen aber HIMEM) als die Methode, BASIC einen Befehl hinzuzufügen. Darum dachte ich mir: Tippe das Source-Listing ab, assembleiere es und drucke es mal für den Club aus!

Dieser Einfall ist für wilde Programmierer sehr bequem.
Falls Ihr aber längst "im Bilde" -
bitte, fallt nicht aus dem FRAME!

Gedacht - getan; und mit "Grüß Gott"
verbleib' ich Euer stets ==> KaJot

1D78		00100	EXIT	EQU	1D78H
7100		00110		ORG	7100H
7100	2198FF	00120	LOAD	LD	HL,FRAME
7103	220440	00130		LD	(4004H),HL
7106	C32D40	00140		JP	402DH
FF98		00150		ORG	0FF98H
FF98	E3	00160	FRAME	EX	(SP),HL
FF99	7D	00170		LD	A,L
FF9A	FE5B	00180		CP	5BH
FF9C	2003	00190		JR	NZ,NO
FF9E	7C	00200		LD	A,H
FF9F	FE1D	00210		CP	1DH
FFA1	E3	00220	NO	EX	(SP),HL
FFA2	C2781D	00230		JP	NZ,EXIT
FFA5	CD781D	00240		CALL	1D78H
FFA8	FE46	00250		CP	'F'
FFAA	2804	00260		JR	Z,YES
FFAC	2B	00270	NO2	DEC	HL
FFAD	C3781D	00280		JP	EXIT
FFB0	E5	00290	YES	PUSH	HL
FFB1	CD781D	00300	YES10	CALL	1D78H
FFB4	FE3A	00310		CP	3AH
FFB6	280B	00320		JR	Z,YES20
FFB8	FE00	00330		CP	00H
FFBA	2807	00340		JR	Z,YES20
FFBC	FED5	00350		CP	0D5H
FFBE	20F1	00360		JR	NZ,YES10
FFC0	E1	00370		POP	HL
FFC1	18E9	00380		JR	NO2
FFC3	E1	00390	YES20	POP	HL
FFC4	23	00400		INC	HL
FFC5	CF	00410		RST	0BH
FFC6	52	00420		DEFB	'R'
FFC7	CF	00430		RST	0BH
FFC8	41	00440		DEFB	'A'
FFC9	CF	00450		RST	0BH
FFCA	4D	00460		DEFB	'M'
FFCB	CF	00470		RST	0BH
FFCC	45	00480		DEFB	'E'
FFCD	E5	00490		PUSH	HL
FFCE	21003C	00500		LD	HL,3C00H
FFD1	CDEBFF	00510		CALL	HORIZ
FFD4	21C03F	00512		LD	HL,3FC0H
FFD7	CDEBFF	00514		CALL	HORIZ
FFDA	21003C	00520		LD	HL,3C00H
FFDD	CD44FF	00530		CALL	VERT
FFE0	213F3C	00540		LD	HL,3C3FH
FFE3	CD44FF	00550		CALL	VERT
FFE6	E1	00560		POP	HL
FFE7	2B	00570		DEC	HL
FFE8	C3781D	00580		JP	EXIT
FFEB	3EBF	00590	HORIZ	LD	A,0BFH
FFED	0640	00600		LD	B,40H
FFEF	77	00610	HOR10	LD	(HL),A
FFF0	23	00620		INC	HL
FFF1	10FC	00630		DJNZ	HOR10
FFF3	C9	00640		RET	
FFF4	3EBF	00650	VERT	LD	A,0BFH
FFF6	0610	00660		LD	B,10H
FFF8	114000	00670		LD	DE,40H
FFFB	77	00680	VE10	LD	(HL),A
FFFC	19	00690		ADD	HL,DE
FFFD	10FC	00700		DJNZ	VE10
FFFF	C9	00710		RET	
7100		00720		END	LOAD

00000 TOTAL ERRORS
34150 TEXT AREA BYTES LEFT

Symbol Table

EXIT	1D78	00100	00230	00280	00580
FRAME	FF98	00160	00120		
HDR10	FFEF	00610	00630		
HORIZ	FFEB	00590	00510	00514	
LOAD	7100	00120	00720		
NO	FFA1	00220	00190		
NO2	FFAC	00270	00380		
VE10	FFFB	00680	00700		
VERT	FFF4	00650	00530	00550	
YES	FFB0	00290	00260		
YES10	FFB1	00300	00360		
YES20	FFC3	00390	00320	00340	

Ja ja - ich weiß: Es ist eine große Sünde, ein Assembler-Listing ohne Kommentare abzudrucken!

Ich bitte 10hoch5-mal um Nachsicht! Als ich den Source-Code vor Jahrenden abschrieb, hatte ich noch keinen Sinn dafür.

Und heute bin ich zu faul, das Ganze noch einmal zu schreiben.

*** Denkt mal selber nach und seid dankbar für meine kleine Gabe... *** KJ

MAILBOX - CORNER

News von der C.L.A. Mailbox in HB und anderen

Telefon: 0421 / 59 21 64, 24h, 300/300, 8/2/no

Was positives kann ich von der Clubmailbox berichten. Dort sind wir inzwischen schon zu dritt. Heinz - Gerd Küster hat sich nämlich mit seinem neuen Terminalprogramm in FORTH eingeloggt. Ihr könnt ihn - wenn Ihr euch mal in der Box meldet - unter dem Usernamen HEINZ GERD erreichen (siehe hierzu auch die Mitgliederliste). Ich hoffe nur, daß sich so nach und nach immer mehr Mitglieder dort melden. Ich selbst habe meine Aktivitäten dort auf einmal je Woche 'verkürzt'. Statt, wie bisher, jeweils Mittwochs und Sonntags melde ich mich jetzt nur noch Sonntags oder Montags. Wie geasgt, ich hoffe auf eine rege Beteiligung an der Box.

Multidos Testbericht

von Ralf Folkerts

Als ich im Jahre 1983 ein DOS suchte, das ich als Ersatz für G-DOS verwenden wollte, fiel mir eine Anzeige der Firma Computer Service in Bürstadt auf. Dort gab es Multidos, Version 1.6 zusammen mit EBASIC zum Preis von DM 165,-. Da habe ich dann natürlich zugeschlagen. Die Diskette war aber leider in SD und ließ sich nicht auf DD kopieren. Nachdem ich mir diese 'Diskettenplatzverschwendung' eine Zeit angeschaut habe (da hat man ein FC Laufwerk und benutzt SD für das Betriebssystem!), habe ich mir dann einen D-Change auf die DD Version bei C.E.C. bestellt. Als diese kam war inzwischen die Version 1.6a 'aktuell'; diese bekam ich auch. Ich habe dann nichts mehr von C.E.C. gehört, bis ich erfuhr, daß AlphaBit den Vertrieb von Multidos übernommen hat. Das war Ende 1985. Da erfuhr ich auch, daß es inzwischen eine Version 1.7 gibt. Diese habe ich jedoch nicht bestellt, da ich mit meiner 1.6a keine Probleme hatte. Als ich dann aber einen Multidos Newsletter erhielt, mit dem die Version 2.0 angekündigt wurde, da griff ich wieder zu. Nachdem ich diese in Händen hielt hatte ich aber zuerst man eine harte Nuß zu knacken. Es funktionierten nämlich weder der BOOT noch der DEAD Befehl (bei DEAD wird zuerst der Speicher ab 4303h gelöscht und dann gebooted). Ich habe mir daher den gesamten Kommandointerpreter disassembliert und dann den Code gesucht (und geändert).

In dem nachfolgenden Testbericht werde ich zuerst einmal über die allgemeinen Vorteile von MD, auch in Version 1.6, berichten und dann die speziellen Vorzüge der 2.0 hervorheben.

Als Vorteile von MD sind zuerst einmal seine Geschwindigkeit und seine 'Format - Erkennung' zu nennen (es stellt sich selbsttätig auf das jeweilige Format ein (SD, DD oder PD)). Die DOS Funktionen laufen schneller ab, als z.B. im G-DOS (ich kann hier aber auch nur von der 2.1 G-DOS Version berichten). Wenn man ein einfaches Directory haben will, drückt man einfach eine Zifferntaste von 0 - 3; das Directory erscheint dann sofort. Beim Backup (gesamte Diskette kopieren) läuft alles wahlweise über ein Menü ab oder es werden die Parameter übernommen, die in der Befehlszeile angegeben werden. Zuerst wird die Queldiskette untersucht (welche Density, wieviel Spuren,...). Wenn die Diskette in G- oder NEWDOS formatiert ist, muß noch die Anzahl der Spuren eingegeben werden. Wenn man diese eingegeben hat, geht das Formatieren los. Am Bildschirm werden die laufenden Funktionen angezeigt. Wenn eine Diskette bereits formatiert ist, kann man, wie bei GDOS die Funktion nach einem Hinweis mit Angabe des Namens und Datums abbrechen. Fährt man fort, werden alle Schritte am Bildschirm angezeigt (die Funktion und die aktuelle Spur mit Sektor). Wenn was 'hakt' kann man also sofort sehen, wo. Wird eine Spur 'ausgelockt' (ich habe, um dies zu testen, das Laufwerk beim 'Verify' kurz geöffnet) erscheint der Hinweis, das in der Spur X ein Granule 'locked out' ist. Beim anschließenden Kopieren werden nur die Granules kopiert, die auch tatsächlich Daten enthalten; leere werden nicht benutzt. Hierdurch läuft der Kopiervorgang bei nicht ganz vollen Disketten noch schneller ab.

Zum Zappen steht ein menügesteuerter Zapper bereit, der (in der Version 2.0) vom Sektorkopieren und einzelne Spuren formatieren bis zum Directory Repair alles kann, was man braucht. Diese Repair Option war auch schon in Version 1.6 implementiert. Ich habe damit eine Newdos Diskette, die selbst SU aufgegeben hatte ohne Probleme repariert. Danach war zwar ein File weg, aber besser, als die ganze Diskette neu zu formatieren. Mit dem Zapper kann man 'File Sektoren', 'Disk Sektoren' und 'Memory' Zappen. Bei der Eingabe der Option 'File Sektoren' erscheint dann die Frage nach dem Filenamen. Ist dieser eingegeben, wird angezeigt, wieviel Sektoren die Datei hat.

Multidos Testbericht

von Ralf Folkerts

Man muß dann den Sektor eingeben, den man haben will. Mit den Pfeiltasten kann man Sektorweise vor- und zurückblättern. Wählt man 'Disk Sektoren' gibt man zuerst das Laufwerk ein. Dann folgt die Frage nach Spur und Sektor (in dieser Spur). Hier kann man mit den Pfeiltasten Spur- oder Sektorweise vorwärts- rückwärts blättern. Nach Eingabe von 'T' kann man Spur und Sektor neu eingeben, nach S kann ein neuer Sektor angegeben werden. Bei der Speicher - Option gibt man die Adresse ein und kann den Bildschirm mit den Pfeiltasten weiterschalten. Hier kann man wählen zwischen Byteweise, 16 Bytes und 256 Bytes (eine Bildschirmseite). Um in einer der genannten Optionen zu modifizieren drückt man einfach 'M'. Es erscheinen dann zwei Cursor. Einer im Hex, der andere im ASCII Feld. Zum Ändern kann man den Cursor dahin bewegen, wo man ihn braucht und dann die neuen Hex Ziffern eingeben. Will man in ASCII ändern drückt man das Affena. Dann kann in ASCII gezapt werden. Durch nochmaliges Drücken vom Affena kommt man dann wieder in den HEX - Modus.

Wenn man viel Dateien zu kopieren hat ist die VFU sehr hilfreich. Sie ersetzt die Purge und die Copy 'FRG' Optionen anderer DOSse. Nach dem Aufruf muß man wählen, ob man kopieren, ausführen, verschieben, löschen oder sich ein Directory drucken will. Bei kopieren und verschieben erscheint dann eine Zeile mit Optionen. Man kann hier selektieren, wann kopiert werden soll (Anmerkung: Multidos datiert alle Dateien im Directory. Beim Anzeigen mit des Directory mit der 'A' Option erscheint dann das Erstellungsdatum auch auf dem Bildschirm). Hier kann dann gewählt werden ob alle Dateien kopiert werden sollen, oder nur dann wenn die Datei auf der Zieldiskette noch nicht existiert, oder nur wenn die Datei auf der Zieldiskette ein älteres Datum hat als die Quelle, oder wenn Quell- und Zielfile in der Größe differieren, oder... Nach Eingabe der Option wählt man dann, ob alle Dateien genommen werden oder ob selektiv gearbeitet werden soll. Schließlich folgen Fragen, ob auch Invisible Dateien beachtet werden sollen, ob System Dateien beachtet werden sollen und zum Schluß die Frage nach Quell- und Ziellaufwerk. Hat man alles eingegeben, erscheint auf dem Bildschirm das Directory. Mit den Pfeiltasten oder der Leertaste kann man dann zwischen den Dateien vor und zurückspringen. Mit der 'Y' Taste werden alle Dateien, die man bearbeiten (kopieren, löschen, verschieben, ausführen) will markiert. Vor ihnen erscheint ein '+'. Hat man eine Datei markiert die man gar nicht verwenden will, kann man dies mit der 'N' Taste wieder rückgängig machen. Wenn man die Option gewählt hat, das alle Dateien beachtet werden sollen, erscheint vor allen das '+' Zeichen. Wenn alles markiert ist drückt man <NL>. Man kann dann wählen, ob man diese Funktion ausführen oder abbrechen will. Drückt man ausführen, werden die Dateien entweder kopiert oder gelöscht. Bei der Funktion 'verschieben' werden die Dateien zuerst kopiert und dann gelöscht. Bei der Funktion ausführen kann man nur eine Datei wählen, die dann ausgeführt wird (Maschinenprogramme werden so geladen und gestartet, bei BASIC Programmen wird erst BASIC aktiviert und das Programm dann gestartet). Während der Arbeit zeigt das Programm an, welche Datei es gerade bearbeitet (mit der Funktion (z.B. 'Killing file xxxxxx/yyy:n')).

So, das war so ziemlich das wichtigste, was im großen und ganzen auch bei der 1.6 läuft. Ach ja, gut ist auch die 'R' Option beim DIR Befehl (zeigt alle gelöschten Dateien) und der RESTOR Befehl, der diese Datei wieder herstellt. Auf die Sachen, die nur in der 2.0 'drin sind will ich jetzt eingehen. Die Library Befehle sind übrigens ziemlich die gleichen. In der 2.0 sind jedoch einige 'rausgeflogen (z.B. wird die Help Funktion jetzt von einem extra Programm ausgeführt).

Multidos Testbericht

von Ralf Folkerts

Wenn man sich das Directory der Systemdiskette ansieht, fällt als erstes auf, daß es keine 'DOS0/SYS', 'SYS0/SYS',... mehr gibt. Die Systemdateien (auch die BASIC Overlays) haben alle 'passende' Namen bekommen (siehe auch Ausdruck). Der alte File 'DOS/SYS' heißt jetzt 'SYSRES/SYS', der Kommandointerpreter heißt nicht mehr 'DOS1/SYS' sondern 'Command/DOL'. Die Extension 'DOL' bedeutet 'DOSOverLay', 'BOL' bedeutet 'BASICOverLay'. Hierdurch kann man sehen, wozu ein File dient, bevor man im Handbuch nachgeschlagen hat.

0 MULTIDOS 01/01/86 40 log 40 phy cyls 43 grans 64.50 K

Allocate/DOL S	BACKUP/CMD I	BASIC/CMD I	BBASIC/CMD I
CAT/CMD P	CDIR/CMD	COPY/CMD I	CREF/BOL S
Close/DOL S	Command/DOL S	DBLFIX/CMD	DDT/CMD P
DIR/SYS S	Debug/DOL S	EDIT/BOL S	ERROR/BOL S
Error/DOL S	FMAP/CMD	FORMAT/CMD I	GR/CMD
HELP/CMD	LO/CMD	Library1/EXT S	Library2/EXT S
MEM/CMD P	MEMDISK/CMD	Minidos/DOL S	Open/DOL S
PACK/BOL S	PRT/CMD	RENUM/BOL S	RS/CMD P
SPOOL/CMD	SYSGEN/CMD	SYSRES/SYS S	TAPE/CMD
UNPACK/BOL S	UTIL/BOL S	VFU/CMD P	ZAP/CMD P

Die erste Neuerung ist das Konzept. Dies ist bei der 2.0 mehr nach LDOS gegangen. So kann man mit CONFIG wählen, ob ein Laufwerk aktiv ist oder nicht, kann einzelne Laufwerke schreibschützen (sie verhalten sich dann, als wenn der Aufkleber auf der Kerbe wäre), kann den physikalischen Laufwerken andere logische Laufwerke zuordnen,... . Bei doppelseitigen Laufwerken kann man wählen, ob sie wie ein Laufwerk (ein Directory) oder wie zwei behandelt werden sollen (zwei Directories). Wenn man die Option wählt daß beide getrennt behandelt werden sollen, erreicht man die Rückseite mit einem "" nach der Nummer (bzw. durch drücken von <SHIFT> <ZIFFERTASTE>, wenn man das Kurzmenü wählt). Was gerade aktiv ist, kann man sich im Klartext anzeigen lassen (siehe Ausdruck).

config

```
:0, Phy = 0, 5" Floppy, double density,
    one sided, one volume, step rate = 30 mS.
:1, Phy = 1, 5" Floppy, double density,
    one sided, one volume, step rate = 12 mS.
:2, Phy = 2, 5" Floppy, double density,
    one sided, one volume, step rate = 06 mS.
:3, MEMORY
:4, Nil
:5, Nil
:6, Nil
:7, Nil
```

Auch neu ist ein SYSGEN, mit dem man Multidos auf eine andere Dichte oder auch auf eine Harddisk kopieren kann. Hier läuft alles per Menü ab. Ferner wurde ein neues FORMS eingeführt. Die neue Version benötigt jetzt ein Programm 'PRT/CMD' in TOPMEM. Dieses Programm kann auch einzelne Bytes ausfiltern; die ist besonders hilfreich, wenn man z.B. die Bildschirmausgabe auf den Drucker leitet, da beim NEC die Cursor - Ein Funktion, die ja allen Systemausgaben vorangestellt wird, die Großschrift selektiert. Daher sind dann alle Kommentare des Gerätes in Großschrift. Mit PRT kann man dann alle Codes die stören ausfiltern.

Multidos Testbericht

von Ralf Folkerts

Bei Forms kann man, wie in der alten Version auch, die Zeilenlänge, die Zeilen/Seite und die zu druckenden Zeilen/Seite anwählen.

Wenn die Anzahl von zu druckenden Zeilen/Seite erreicht ist, wird ein Form Feed ausgeführt. Neu ist in der 2.0 die Möglichkeit, auch einen linken Rand zu setzen. Zusammen mit den anderen Optionen kann man hiermit jetzt Listings drucken, die weder den Lochrand noch die Perforation zwischen den Seiten 'verwenden'.

Der ROUTE Befehl wurde erweitert. Man kann jetzt auch auf Dateien umleiten. Mit dem DEVICE Befehl, die die Treiberadresse der Devices anzeigt, kann man sich ansehen, welche Device gelinkt oder gerouted ist und wohin es läuft (linken bedeutet, daß die Ein- und Ausgaben sowohl zu der einen als auch zu der anderen Device gehen, routen ist eine Umleitung). Im Beispiel habe ich die Bildschirmausgabe auf den Drucker gelinkt, den ich auf einen Disk File gerouted habe.

device

KI = I at X'4C20'

DO = 0 at X'4CD8', linked to PR

PR = 0 at X'FDFD', routed to :1

Ebenfalls neu ist eine komplette WILDCARD- und MASK Filterung beim Directory. Es stehen hierzu 2 Zeichen zur Verfügung: '*' und '?'. Ein '*' ersetzt eine ganze Gruppe von Buchstaben, das '?' ersetzt einen einzigen Buchstaben. Wenn man z.B. alle Dateien haben will, die als ersten Buchstaben im Namen ein 'B' haben, gibt man ein 'DIR B*/*'. Sollen alle Dateien genommen werden, deren Extension CMD lautet gibt man 'DIR */CMD' ein. Um alle Dateien zu bekommen, die als letzten Buchstaben in der Extension ein 'G' und als erstes Zeichen in der Extension ein 'F' haben gibt man 'DIR */F?G' ein. Man kann hiermit (nach kurzer Gewöhnung) sehr gut arbeiten. Dies wird dadurch unterstützt, daß Multidos das Directory alphabetisch sortiert (immer, also auch bei der 'Eintasten - Version').

Mit das beste ist jedoch eine MEMDISK. Bei der Initialisierung kann man angeben, wie groß diese sein soll (von 2 - 16 Track; jede Track entspricht 2 K). Die MEMDISK wird dann wie ein normales Laufwerk geführt. Sie wird hierbei als erstes freies Laufwerk eingesetzt (s. auch den CONFIG Ausdruck weiter oben). Von dieser Disk kann man sich ein Directory ziehen, sich eine Free Map ausdrucken lassen (hierbei wird angezeigt, welche Granules belegt sind oder welche Files wo liegen), die Dateien kopieren, löschen, Nur FORMATieren und BACKUPpen kann man die Memdisk nicht (hier erscheint der Hinweis 'Can't FORMAT a MEMDISK'). Durch die Möglichkeit, physikalische und logische Laufwerke zu 'tauschen' kann man die MEMDISK auch als Systemdisk verwenden.

Im technischen Anhang des Buches stehen wieder alle Systemadressen und deren Bedeutung (11 Seiten voll; nicht so ein kurzer Abriß wie im NEWDOS Handbuch). Danach folgt ein Bereich mit ZAPs. Hier steht, wie man das System z.T. noch besser an seine eigenen Bedürfnisse anpassen kann (z.B. kann man Sonderzeichen und / oder Kleinbuchstaben in Filenamen als 'gültig' deklarieren, wodurch z.B. Filenamen wie 'Test' und 'TEST' sich unterscheiden). Hier stehen auch noch ein paar andere ZAPs. Der meiner Meinung nach interessanteste geht darauf ein, wie man die Lesegeschwindigkeit durch Ändern des 'Interleave' beschleunigen kann. Bei der Formatierung werden die Sektoren auf der Disk ja nicht nacheinander angeordnet, sondern versetzt. Die Anzahl von Sektoren, die zwischen den einzelnen stehen, nennt man den Interleave. Bei DD Disketten beträgt dieser z.B. i.a. drei. Das heißt, die Sektoren stehen in der Reihenfolge '0, 6, 12, 1, 7, 13, ...' auf der Scheibe.

Multidos Testbericht

von Ralf Folkerts

Um eine ganze Spur zu lesen, sind also drei Umdrehungen (mindestens) der Diskette notwendig. Unter Multidos kann dieser Interleave unter DD jedoch ab 2,5MHZ auf zwei gesenkt werden, da es die Sektoren so flink verarbeitet. Je zu lesender Spur spart man hier 200 ms.

So, ich meine, daß es jetzt vom DOS her reicht. Ich will jetzt noch kurz auf das Disk BASIC eingehen.

Zuerst einmal: Das Super BASIC genannte BASIC von Multidos ist recht schnell: Für das folgende Programm benötigte das NEWDOS BASIC 2:47 Sekunden, Super BASIC schaffte es in 1:42 Sekunden.

```
10 FOR A=1 TO 100
20 NEXT A
30 FOR B=&H1 TO &H100
40 NEXT B
50 FOR A=1 TO 10
60 CMD"T"
70 CMD"R"
80 NEXT
90 END
```

Das Super BASIC ist jedoch nicht ganz so umfangreich wie das vom NEWDOS. So fehlen die speziellen Dateibefehle oder Optionen wie dynamisches Löschen von Zeilen,... . Für allgemeine Anwendungen ist es jedoch mehr als ausreichend. Als Ergänzung zum SuperBASIC ist noch ein BOSS genannter Debugger auf der Diskette. Bei diesem kann man ein Programm auf Knopfdruck in HIMEM speichern und ein zweites laufen lassen, die Werte von Variablen während des Programmlaufes ansehen und ändern (danach wird der Bildschirminhalt wieder hergestellt), Andere Optionen sind Single Step bis Zeilenende, Single Step je Befehl, Trace on (die Zeilennummern versauen nicht mehr das ganze Bild; es werden vier Zeilen in der oberen rechten Ecke angezeigt (mit Symbol, welche gerade ausgeführt wird)) usw. Wenn man mal echt 'nen Käfer sucht ist dies wirklich eine saubere Lösung. In beiden BASICs steht einem eine Pack und Unpack Routine zur Verfügung (die eine packt mehrere Programmzeilen zu einer zusammen und entfernt, auf Wunsch, auch Spaces, die andere setzt zwischen alle Befehle Leerzeichen und nimmt je Befehl eine Zeile; ferner wird das Programm dann noch in 10ner Schritten renummeriert). Man kann auch REMs entfernen (CMD'X') oder das Programm im Speicher in Level II transferieren. Auch gut ist im BASIC ein Globaler Editor. Mit diesem kann man im Programm Befehle/Zeichen löschen oder ersetzen. Ferner kann man CHR\$(XX) - Ketten durch gepackte Strings ersetzen lassen, wodurch Platz gespart und das Programm schneller wird.

Das war eigentlich alles, was ich für erwähnenswert halte. Auch hier stehe ich für Rückfragen gerne bereit.

P.S. Die eingeschobenen Ausdrücke habe ich erzeugt, indem ich den Bildschirm jeweils auf einen Disk - File gerOUTEd habe, und beim Formatieren und Drucken des Berichtes diese Dateien in den Text gechaint habe.

HANDLER/TXT

HANDLER/CMD ist ein Programm das von BASIC aus aufgerufen wird mit USR um komfortablen Input zu ermöglichen.

Es lädt ab FE00 hex. Aus diesem Grunde muß Mem-Size beim BASIC-Aufruf auf 65024 gesetzt werden, damit es nicht zerstört wird. Wer den Banker hat kann HANDLER/CMD natürlich im "sicheren Bereich" fahren. Es ist dann nur im BASIC-Programm DEFUSR entsprechend der Adresse festzulegen.

Im normalen BASIC INPUT kann man folgendes nicht:

1. Keine maximale Inputlänge festlegen
2. keine Angabe über die Länge der eingegebenen Zeichen (nur über Umwege via LEN)
3. keine Festlegung für nur numerische Eingabe oder nur Großbuchstaben Eingabe.
(numerische Variable zwar möglich doch ERROR wenn Eingabe nicht numerisch)
4. Der Cursor kann nicht zurückbewegt werden ohne die bisherigen Eingaben zu löschen.
5. Einfügen von Zeichen mitten in die bisherige Eingabe ist nicht möglich
6. Durch den Gebrauch von CLEAR, Hochpfeil und Shift Hochpfeil kann der Bildschirminhalt zerstört werden.
7. Benutzung von Zeichen wie " : ; , ist nicht ohne weiteres möglich (Dieses geht zwar unter DISK-BASIC via LINEINPUT doch die anderen Nachteile bleiben).

HANDLER/CMD gibt BASIC-Programmen beinahe kommerziellen Charakter. Der Cursor wird in das Inputfeld, welches durch zwei CHR\$(128) -grafische Blanks- begrenzt wird, gesetzt. In dem Inputfeld ist der Cursor frei beweglich z.B. durch Rechts- und Linkspfeil.

Mit dem Hochpfeil wird ein Leerzeichen eingefügt z.B. in schon vorhandenen Text.

Mit CLEAR wird das Zeichen unter dem Cursor gelöscht.

Mit NEW LINE geht es zur nächsten BASIC-Funktion. Die wiederum kann auf das nächste Eingabefeld verzweigen.

Falls die automatische Funktion (SKIP) vorgewählt ist, geht es am Ende des Eingabefeldes ebenfalls zur nächsten BASIC-Funktion.

Die Steuercodes im Einzelnen:

Rechtspfeil	:	eine Position weiter
Linkspfeil	:	eine Position zurück
Hochpfeil	:	Leerzeichen an der Cursorposition einfügen
Shift Rechtspfeil	:	Mit +1 als Argument zurück zum BASIC
Shift Linkspfeil	:	Wenn nicht am Beginn des Eingabefeldes, dann zum Beginn springen sonst zum BASIC mit -1 als Argument
CLEAR	:	Zeichen unter dem Cursor löschen
Shift CLEAR	:	Vom Cursor ab alles mit Leerzeichen füllen
NEW LINE	:	Zurück zum BASIC mit 0 als Argument
BREAK	:	Zurück zum BASIC mit 2 als Argument

Die Argumente können im BASIC-Programm zur weiteren Behandlung benutzt werden, z.B. IF Q =...THEN (das Argument befindet sich in der Variablen Q)

Bei Aufruf des Maschinenprogramms mittels USR Funktion können folgende Optionen vorgewählt werden:

0	:	alphanumerisches Feld
1	:	zurück zum BASIC am Ende des Feldes
2	:	Stop am Ende des Feldes (zum BASIC mittels NEW LINE)
3	:	Fülle Eingabefeld mit CHR\$(127)
7	:	nur die Kontrolltasten als Eingabe erlaubt
8	:	nur Großbuchstaben (Uppercase lock)
9	:	nur numerische Eingabe
> 255	:	Blinkgeschwindigkeit des Cursors

Nach Beendigung der Eingabe und Rücksprung zum BASIC befinden sich die eingegebenen Zeichen in der Variablen SC\$ und müssen dann den Wünschen des Programmierers entsprechend an andere Variable übergeben werden.

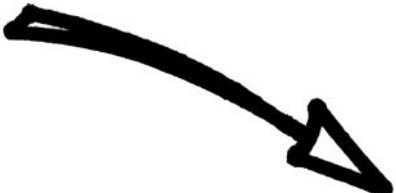
Die Länge der eingegebenen Zeichen befindet sich in der Variablen SC%. Das Eingabefeld wird auf dem mittels PRINT\$,CHR\$(128) und PRINT\$,...,CHR\$(128) begrenzt. Es kann also nur so lang sein, wie Zeichen zwischen den beiden CHR\$(128) passen.

mit PRINT\$,...; wird der Cursor an die erste Stelle des Eingabefeldes gesetzt.

Das BASIC-Programm HANDLER/DEM demonstriert einige der Möglichkeiten von HANDLER/CMD.

Viel Spaß mit dieser Routine wünscht Paul Kröher

Es folgen auf den weiteren Seiten
das Assemblerlisting
und
ein BASIC-Demonstrationsprogramm



A n z e i g e n



Können in den INFOs leider bis auf weiteres nicht mehr erscheinen. In einer Büchersendung dürfen Anpreisungen nämlich nur

- a) auf dem Umschlag oder
- b) an zwei aufeinanderfolgenden Seiten am Anfang und Ende des Heftes

stehen. Da dies dann aber z.Zt. etwas viel Anzeigen wären, verzichte ich erstmal auf alles. Sollte sich jedoch eine entsprechende Anzahl von Anpreisungen finden, könnte ich evtl. die vier Seiten damit drucken. Ich hoffe auf Euer Verständnis..

```

00100 ; -----
00110 ; SCREEN HANDLER 2.0
00120 ; AUTOR: RUUD BROERS IN NIEUWEGEIN (NL)
00130 ; AUS DER CLUBZEITSCHRIFT DER
00140 ; TRS-80 GEBRUIKERS VERENIGING AUSGABE 6/86
00150 ; UEBERSETZT VON PAUL KROEHER
00160 ; -----

```

```

00170 ; DAS INPUT-FELD BEGINNT UND ENDET MIT CHR$(128)
00180 ; HANDLER WIRD ALS USR-FUNKTION AUFGERUFEN
00190 ; FUNKTION UND BEDIENUNG SIEHE HANDLER/TXT
00200 ;

```

```

FE01 00210      ORG      OFE01H
402D 00220 DOS      EQU      402DH
06CC 00230 BASIC    EQU      06CCH
0A7F 00240 GETHL    EQU      0A7FH      ; HOLE USR-ARGUMENT
0A9A 00250 PUTHL    EQU      0A9AH      ; GEBE USR-ARGUMENT
4020 00260 CURSOR   EQU      4020H      ; CURSOR POSITION
002B 00270 GETKEY   EQU      002BH      ; TASTATURABFRAGE
260D 00280 LOCATE   EQU      260DH      ; SUCHE VARIABLE
001F 00290 CLEAR    EQU      31
000D 00300 ENTER    EQU      13
0001 00310 BREAK    EQU      1
0200 00320 RATE      EQU      0200H
008F 00330 CRS      EQU      143      ; CURSOR ZEICHEN
007F 00340 FCHR     EQU      127      ; EINGABEFELD MARKIEREN
FE01 CD7F0A 00350 MAIN    CALL     GETHL
FE04 7C     00360      LD      A,H
FE05 B7     00370      OR      A
FE06 C2BBFF 00380      JP      NZ,SRATE      ; BLINKFREQUENZ CURSOR
FE09 7D     00390      LD      A,L      ; HOLE USR-ARGUMENT
FE0A 32C3FF 00400      LD      (OPTIE),A      ; UND SICHERN
FE0D FE01   00410      CP      1
FE0F CAA1FF 00420      JP      Z,ENDSTF
FE12 FE02   00430      CP      2
FE14 CA9DFF 00440      JP      Z,ENDSTF
FE17 FE03   00450      CP      3
FE19 CAABFF 00460      JP      Z,UNDERL
FE1C 210002 00470      LD      HL,RATE      ; BLINKZAEHLER
FE1D      00480 SRATE1  EQU      $-2
FE1F 22C4FF 00490      LD      (CNT),HL      ; AUF STARTWERT
FE22 CD8DFF 00500 LOOP1  CALL     CRSBEG      ; CURSOR AUF BEGINN
FE25 7E     00510 LOOP2  LD      A,(HL)      ; HOLE CHR VOM SCHIRM
FE26 FE80   00520      CP      80H      ; UND IN BUFFER ABLEGEN
FE28 3003   00530      JR      NC,LOOP3      ; WENN ES NICHT
FE2A 32C6FF 00540      LD      (CHRBUF),A      ; DER CURSOR IST
FE2D CD66FF 00550 LOOP3  CALL     BLINK
FE30 CD2B00 00560      CALL     GETKEY      ; TASTATURABFRAGE
FE33 B7     00570      OR      A      ; TASTE GEDRUECKT?
FE34 28F7   00580      JR      Z,LOOP3      ; NEIN, NEU ABFRAGEN
FE36 32C2FF 00590      LD      (KEY),A      ; TASTATUR SICHERN
FE39 CD80FF 00600      CALL     CRSCHR      ; CURSOR ZUM CHARACTER
FE3C 3AC2FF 00610      LD      A,(KEY)
FE3F FE20   00620      CP      20H      ; FUNKTION ODER BUCHSTABE
FE41 3844   00630      JR      C,FUNCT
FE43 FE5B   00640      CP      5BH      ; HOCHPFEIL ?
FE45 2840   00650      JR      Z,FUNCT
FE47 FE7B   00660      CP      7BH      ; SHIFT HOCHPFEIL ?
FE49 283C   00670      JR      Z,FUNCT
FE4B 3AC3FF 00680      LD      A,(OPTIE)      ; HOLE ARGUMENT
FE4E FE07   00690      CP      7
FE50 2835   00700      JR      Z,FUNCT      ; NUR FUNKTIONEN
FE52 FE09   00710      CP      9      ; NUR ZIFFERN
FE54 2818   00720      JR      Z,NEGENV      ; NUR NUMMERISCH
FE56 FE08   00730      CP      8      ; UPPERCASE LOCK ?

```

FE5E	2802	00740	JR	Z,UCL	
FE5A	181D	00750	JR	VALID	; HIER WEITERE OPTIONEN
FE5C	3AC2FF	00760 UCL	LD	A, (KEY)	
FE5F	FE61	00770	CP	'a'	; CONVERT
FE61	3809	00780	JR	C,UCL2	; TD
FE63	FE7B	00790	CP	'z'+1	; UPPERCASE
FE65	3005	00800	JR	NC,UCL2	
FE67	E65F	00810	AND	5FH	
FE69	32C2FF	00820	LD	(KEY),A	; UND WIEDER IN BUFFER
FE6C	180B	00830 UCL2	JR	VALID	
FE6E	3AC2FF	00840 NEGENV	LD	A, (KEY)	; JA, DANN DARFS ALLEIN
FE71	FE30	00850	CP	'0'	; 0 - 9 SEIN
FE73	38B0	00860	JR	C,LOOP2	; SONST NOCHMALS
FE75	FE3A	00870	CP	'9'+1	
FE77	30AC	00880	JR	NC,LOOP2	
FE79	3AC2FF	00890 VALID	LD	A, (KEY)	; HOLE ZEICHEN
FE7C	77	00900	LD	(HL),A	; UND AUF DEM SCHIRM
		00910 ;	LD	(CHRBUF),A	; UND IN DEN BUFFER
		00911 ;			VORSTEHENDES ERGIBT EINE SCHLECHTE BILDSCHIRMANZEIGE
		00912 ;			DA DAS LETZTE ZEICHEN NOCHMALS AUF DIE NEUE POSITION
		00913 ;			AUSGEGEBEN WIRD. DESHALB FEHLEN LASSEN !!!
FE7D	23	00920	INC	HL	; NAECHSTE SCHIRMSTELLE
FE7E	7E	00930	LD	A, (HL)	
FE7F	FE80	00940	CP	80H	; ENDE DES INPUTFELDES
FE81	20A2	00950	JR	NZ,LOOP2	; NEIN NOCHMALS
FE83	2B	00960	DEC	HL	
FE84	C31DFF	00970	JP	TABF	; STOP ODER SKIP
FE85		00980 STPSKP	EQU	\$-2	
FE87	3AC2FF	00990 FUNCT	LD	A, (KEY)	
FE8A	FE09	01000	CP	9	; RECHTSPFEIL
FE8C	281F	01010	JR	Z, INC	
FE8E	FE08	01020	CP	8	; LINKSPFEIL
FE90	2826	01030	JR	Z, DEC	
FE92	FE5B	01040	CP	5BH	; HOCHPFEIL
FE94	2855	01050	JR	Z, INSERT	
FE96	FE1F	01060	CP	CLEAR	
FE98	2829	01070	JR	Z, DCLEAR	
FE9A	FE0D	01080	CP	ENTER	
FE9C	287B	01090	JR	Z, DENTER	
FE9E	FE01	01100	CP	BREAK	
FEA0	2873	01110	JR	Z, DBREAK	
FEA2	FE19	01120	CP	25	; SHIFT RECHTSPFEIL
FEA4	2877	01130	JR	Z, TABF	
FEA6	FE18	01140	CP	24	; SHIFT LINKSPFEIL
FEA8	2860	01150	JR	Z, TABM	
FEAA	C325FE	01160	JP	LOOP2	
		01170 ;			-----
FEAD	23	01180 INC	INC	HL	
FEAE	7E	01190	LD	A, (HL)	
FEAF	FE80	01200	CP	80H	; ENDE INPUTFELD?
FEB1	C225FE	01210	JP	NZ,LOOP2	; NEIN NOCHMALS
FEB4	2B	01220	DEC	HL	; JA, STOP ODER SKIP
FEB5	C325FE	01230	JP	LOOP2	
FEB6		01240 STPSK2	EQU	\$-2	
		01250 ;			-----
FEB8	2B	01260 DEC	DEC	HL	
FEB9	7E	01270	LD	A, (HL)	
FEBA	FE80	01280	CP	80H	; BEGINN INPUTFELD?
FEBC	C225FE	01290	JP	NZ,LOOP2	; NEIN NOCHMALS
FEBF	23	01300	INC	HL	; JA, STOP
FEC0	C325FE	01310	JP	LOOP2	
		01320 ;			-----
FEC3	3A8038	01330 DCLEAR	LD	A, (3880H)	; SHIFTTASTEN
FEC6	B7	01340	OR	A	
FEC7	280D	01350	JR	Z, DEL	; NEIN

FEC9	E5	01360		PUSH	HL	
FECA	3620	01370	FILL	LD	(HL), ' '	; SHIFT CLEAR
FECB	23	01380		INC	HL	; FELD AUFFUELLEN
FECD	7E	01390		LD	A, (HL)	; MIT LEERZEICHEN
FECE	FE80	01400		CP	80H	
FEDO	20F8	01410		JR	NZ, FILL	
FED2	E1	01420		POP	HL	
FED3	C325FE	01430		JP	LOOP2	; NOCHMALS
		01440				-----
FED6	E5	01450	DEL	PUSH	HL	; LOESCHE LETZTES ZEICHEN
FED7	44	01460		LD	B, H	
FED8	4D	01470		LD	C, L	
FED9	23	01480		INC	HL	
FEDA	7E	01490	DEL1	LD	A, (HL)	; SCHIEBE EINE POSITION
FEDB	FE80	01500		CP	80H	; ZUM ENDE DES
FEDD	2805	01510		JR	Z, DEL2	; INPUTFELDES
FEDF	02	01520		LD	(BC), A	
FEE0	23	01530		INC	HL	
FEE1	03	01540		INC	BC	
FEE2	18F6	01550		JR	DEL1	
FEE4	3E20	01560	DEL2	LD	A, ' '	
FEE6	02	01570		LD	(BC), A	
FEE7	E1	01580		POP	HL	
FEE8	C325FE	01590		JP	LOOP2	; NOCHMALS
		01600				-----
FEEB	E5	01610	INSERT	PUSH	HL	; EINFUEGEN
FEED	7E	01620	INS1	LD	A, (HL)	; LETZTES ZEICHEN
FEED	FE80	01630		CP	80H	; SCHON ENDE INPUTFELD
FEED	2803	01640		JR	Z, INS2	
FEF1	23	01650		INC	HL	
FEF2	18F8	01660		JR	INS1	
FEF4	2B	01670	INS2	DEC	HL	
FEF5	54	01680		LD	D, H	; SCHIEBE ALLES
FEF6	5D	01690		LD	E, L	
FEF7	44	01700		LD	B, H	; EINE POSITION WEITER
FEF8	4D	01710		LD	C, L	
FEF9	E1	01720		POP	HL	
FEFA	DF	01730		RST	18H	; ALLES KLAR?
FEFB	2808	01740		JR	Z, INS4	
FEFD	0B	01750		DEC	BC	
FEFE	0A	01760	INS3	LD	A, (BC)	
FEFF	12	01770		LD	(DE), A	
FF00	1B	01780		DEC	DE	
FF01	0B	01790		DEC	BC	
FF02	DF	01800		RST	18H	; CP DE, HL
FF03	20F9	01810		JR	NZ, INS3	
FF05	3620	01820	INS4	LD	(HL), ' '	; MIT LEERZEICHEN FUELLEN
FF07	C325FE	01830		JP	LOOP2	
		01840				-----
FF0A	2B	01850	TABM	DEC	HL	
FF0B	7E	01860		LD	A, (HL)	
FF0C	FE80	01870		CP	80H	
FF0E	C222FE	01880		JP	NZ, LOOP1	
FF11	21FFFF	01890		LD	HL, OFFFFH	
FF14	FD	01900		DEFB	0FDH	
		01910				-----
FF15	210200	01920	DBREAK	LD	HL, 0002	; RETURN CODE
FF18	FD	01930		DEFB	0FDH	
		01940				-----
FF19	210000	01950	DENTER	LD	HL, 0000	; RETURN CODE
FF1C	FD	01960		DEFB	0FDH	
		01970				-----
FF1D	210100	01980	TABP	LD	HL, 0001	; RETURN CODE
		01990				-----
FF20	E5	02000	EXIT	PUSH	HL	; RETTE RETURN CODE

FF21	CD8DFF	02010		CALL	CRSBEG	; HL -> FELDBEGINN
FF24	010100	02020		LD	BC,0001	; LAENGE 1
FF27	7E	02030	EXIT1	LD	A,(HL)	; SUCHE FELDENDE
FF28	FE80	02040		CP	80H	
FF2A	280A	02050		JR	Z,EXIT2	
FF2C	FE7F	02060		CP	FCHR	
FF2E	2002	02070		JR	NZ,EXIT11	
FF30	3620	02080		LD	(HL),20H	
FF32	03	02090	EXIT11	INC	BC	
FF33	23	02100		INC	HL	
FF34	18F1	02110		JR	EXIT1	
FF36	2B	02120	EXIT2	DEC	HL	; KOREKTUR HL,BC
FF37	0B	02130		DEC	BC	
FF38	7E	02140		LD	A,(HL)	; SUCHE CHR <> ' '
FF39	FE20	02150		CP	' '	
FF3B	28F9	02160		JR	Z,EXIT2	; NOCH NICHT? SCHLEIFE
FF3D	79	02170		LD	A,C	
FF3E	32C7FF	02180		LD	(LEN),A	; FELDLAENGE SICHERN
FF41	21C9FF	02190		LD	HL,SC	; SUCHE VARIABLE
FF44	CD0D26	02200		CALL	LOCATE	; SC%
FF47	21C7FF	02210		LD	HL,LEN	; 'DE' ZEIGER AUF
FF4A	010200	02220		LD	BC,2	; VARIABLE
FF4D	EDB0	02230		LDIR		
FF4F	21CDFF	02240		LD	HL,SCSTR	; SUCHE VARIABLE
FF52	CD0D26	02250		CALL	LOCATE	; SC\$
FF55	3AC7FF	02260		LD	A,(LEN)	; STRINGLAENGE
FF58	12	02270		LD	(DE),A	
FF59	CD8DFF	02280		CALL	CRSBEG	; SETZE STRINGPOINTER
FF5C	13	02290		INC	DE	
FF5D	7D	02300		LD	A,L	; LSB
FF5E	12	02310		LD	(DE),A	
FF5F	13	02320		INC	DE	
FF60	7C	02330		LD	A,H	; MSB
FF61	12	02340		LD	(DE),A	
FF62	E1	02350		POP	HL	; HOLE RETURNCODE
FF63	C39A0A	02360		JP	PUTHL	; ZUM BASIC
		02370				
FF66	E5	02380	BLINK	PUSH	HL	; CURSORPOSITION RETTEN
FF67	2AC4FF	02390		LD	HL,(CNT)	; ZAEHLER SCHON AUF
FF6A	2B	02400		DEC	HL	; NULL ?
FF6B	7D	02410		LD	A,L	
FF6C	B4	02420		OR	H	
FF6D	2805	02430		JR	Z,BLINK2	; JA
FF6F	22C4FF	02440		LD	(CNT),HL	; NEIN ABSPEICHERN
FF72	E1	02450		POP	HL	
FF73	C9	02460		RET		
FF74	210002	02470	BLINK2	LD	HL,RATE	; SETZE ZAEHLER AUF
FF75		02480	SRATE2	EQU	\$-2	
FF77	22C4FF	02490		LD	(CNT),HL	; BEGINNWERT
FF7A	E1	02500		POP	HL	; HOLE CURSORPOSITION
FF7B	7E	02510		LD	A,(HL)	
FF7C	FE80	02520		CP	80H	; CURSOR AN?
FF7E	3805	02530		JR	C,CHRCRS	; NEIN CHR -> CURSOR
		02540	; ODER	JR	NZ,CHRCRS	; INVERSER CURSOR
FF80	3AC6FF	02550	CRSCHR	LD	A,(CHRBUF)	; CURS -> CHR
FF83	77	02560		LD	(HL),A	; HOLE CHR AUS BUFFER
FF84	C9	02570		RET		; ZUM BILDSCHIRM
FF85	7E	02580	CHRCRS	LD	A,(HL)	; HOLE CHR VOM SCHIRM
FF86	32C6FF	02590		LD	(CHRBUF),A	; IN DEN BUFFER
FF89	3E8F	02600		LD	A,CRS	; CURSOR AUS
		02610		CP	'a'	
		02620		JR	C,UC	
		02630		JP	'z'+1	; INVERSE
		02640		JR	NC,UC	
		02650		AND	5FH	

```

02660 ;UC      ADD      A,160      ; CONVERT TO INVERS
FF8B 77        02670      LD      (HL),A      ; CURSOR AUF SCHIRM
FF8C C9        02680      RET
02690 ;-----
FF8D 2A2040    02700 CRSBEG  LD      HL,(CURSOR) ; SETZE CURSOR
FF90 7E        02710 CRS1   LD      A,(HL)   ; ZUM BEGINN
FF91 FE80      02720      CP      80H      ; DES FELDES
FF93 2803      02730      JR      Z,CRS2
FF95 2B        02740      DEC     HL
FF96 18F8      02750      JR      CRS1
FF98 23        02760 CRS2   INC     HL
FF99 222040    02770      LD      (CURSOR),HL
FF9C C9        02780      RET
02790 ;-----
FF9D 2125FE    02800 ENDSTP  LD      HL,LOOP2
FFA0 FD        02810      DEFB   0FDH
FFA1 211DFF    02820 ENDSKP  LD      HL,TABP
FFA4 2285FE    02830 END2   LD      (STPSKP),HL
FFA7 22B6FE    02840      LD      (STPSK2),HL
FFAA C9        02850      RET
02860 ;-----
FFAB CD8DFF    02870 UNDERL CALL   CRSBEG      ; HL -> BEGINN
FFAE 7E        02880 UNDER2 LD      A,(HL)
FFAF FE80      02890      CP      80H      ; SCHON ENDE
FFB1 C8        02900      RET      Z
FFB2 FE20      02910      CP      20H
FFB4 2002      02920      JR      NZ,UNDER3
FFB6 367F      02930      LD      (HL),FCHR
FFB8 23        02940 UNDER3  INC     HL
FFB9 18F3      02950      JR      UNDER2
02960 ;-----
FFBB 221DFE    02970 SRATE  LD      (SRATE1),HL
FFBE 2275FF    02980      LD      (SRATE2),HL
FFC1 C9        02990      RET
03000 ;-----
FFC2 00        03010 KEY   DEFB   00
FFC3 00        03020 OPTIE  DEFB   00
FFC4 0002      03030 CNT   DEFW   RATE
FFC6 20        03040 CHRBUF  DEFB   20H
FFC7 0000      03050 LEN   DEFW   00
FFC9 53        03060 SC    DEFM   'SC%'
FFCC 00        03070      DEFB   0
FFCD 53        03080 SCSTR  DEFM   'SC#'
FFD0 00        03090      DEFB   0
03100 ;-----
402D          03110 ;      END     BASIC      ; FUER LEVEL II
          03120      END     DOS

```

00000 mal gepennt
27063 Zeichen verfügbar

```

10 *HANDLER/DEM
20 CLEAR5000:CLS
30 DEFUSR0=&HFE01
40 GOSUB360: * BAUE BILDSCHIRM AUF
50 Q=USR0(2): * STOP AM ENDE DES FELDES
60 Q=USR0(512) * SETZE BLINKFREQUENZ
70 * FUELLE FELD
80 PRINT$12+4*64,,:Q=USR0(3):Q=USR0(0)' hole Name
90 GOSUB480: * ZEIGE RESULTAT hier kann Resultat aus SC$ in andere Va
riable uebernommen werden zur weiteren Verarbeitung
100 IFQ=-1THEN50: * BEI SHIFT LINKSPFEIL NEU STARTEN
110 IFQ=2THEN320
120 * FULLE FELD
130 PRINT$12+5*64,,:Q=USR0(3):Q=USR0(0)' hole Strasse
140 IFQ=-1THEN50 * BEI SHIFT LINKSPFEIL 1 FELD ZURUECK
150 IFQ=2THEN320
160 GOSUB480
170 PRINT$47+5*64,,:Q=USR0(3):Q=USR0(9):'hole Hausnummer
180 GOSUB480
190 IFQ=-1THEN130
200 IFQ=2THEN320
210 * SKIP AM ENDE , UPPERCASE LOCK
220 PRINT$14+6*64,,:Q=USR0(1)+USR0(3):Q=USR0(8)' hole PLZ
230 GOSUB480
240 IFQ=-1THEN170
250 IFQ=2THEN320
260 Q=USR0(2) *STOP AM ENDE
270 PRINT$35+6*64,,:Q=USR0(3):Q=USR0(0)'hole Ort
280 GOSUB480
290 IFQ=-1THEN220
300 IFQ=2THEN320
310 GOTD50 * hier nur zur Demonstration Sprung wieder zur Eingabe
320 PRINT$15*64,"DRUECKE <BREAK> UM ZU STOPPEN";
330 PRINTCHR$(31)CHR$(128)" "CHR$(128)CHR$(24)CHR$(24);
340 Q=USR0(256):Q=USR0(7) * NUR KONTROLTASTEN
350 IFQ=2THENEDELSERUN
360 CLS:PRINT"+STRING$(62,"-")"+";
370 PRINT"+ NAMENS EINGABESCHIRM"CHR$(215)"DATUM: 06.06.1986 +";
380 PRINT"+STRING$(62,"-")"+";
390 PRINT
400 PRINT"IHR NAME : "CHR$(128)STRING$(30," ")CHR$(128)
410 PRINT"STRASSE : "CHR$(128)STRING$(30," ")CHR$(128);
420 PRINT"NR.: "CHR$(128)" "CHR$(128)
430 PRINT"POSTLEITZAHL: "CHR$(128)" "CHR$(128)"ORT :";
440 PRINTCHR$(128)STRING$(30," ")CHR$(128)
450 PRINT
460 PRINT"+STRING$(62,"-")"+";
470 RETURN
480 PRINT$640,"FELDLAENGE : ";SC%
490 PRINT"FELDSTRING : ";SC%;CHR$(31)
500 PRINT"RETURN CODE : ";Q;" "
510 RETURN

```

Graphik-JKL für das Genie III s

Kürzlich schrieb ich eine Routine, die mit Shift-JKL die hochauflösende Graphik des G3s zu Papier bringt. Darüber habe ich auch einen Artikel verzapft, der aber bisher (heute ist der 21. 4. 86) noch nicht erschienen ist. Mit dieser Routine konnten die gewöhnlichen ASCII- bzw. die alternativen Zeichen nicht ausgedruckt werden. Der Grund ist folgender: Die CPU kann in den Zeichengenerator zwar die Bitmuster für einen Zeichensatz schreiben, das Character-RAM aber nicht wieder auslesen. Das schafft nur der Bildschirm. Da nun sehr unterschiedliche Zeichensätze geladen werden können, ist es nicht sinnvoll, ganz einfach die Zeichen des Druckers zu diesem Zweck zu mißbrauchen.

Inzwischen ist eine neue Routine fertig, die das packt. Obgleich der Zeichengenerator aus der Sicht der CPU ein WOM (write only memory - nein, das ist hier ausnahmsweise kein Witz) ist, können die Bitmuster der Zeichen dennoch gelesen werden. Sie stehen nämlich zusätzlich im Hauptspeicher in Bank 1 von 0100 bis 10FF. Das Hardcopy-Programm liest sie von dort aus und verknüpft sie XOR mit der Graphik, genau wie es der Bildschirm tut. So entsteht auf dem Papier eine punktgenaue Kopie des Bildschirms. Ein bißchen ist das allerdings gelogen, weil wegen gefälligerer und natürlicherer Proportionen des Ausdrucks jede Dotspalte abwechselnd ein- oder zweimal gedruckt wird.

Die drei Hardcopies sind alle vom selben Bildschirm entstanden. Die am Fuß dieser Seite ist mit dem gewohnten JKL ohne Shift erstellt. Es werden wie üblich nur druckbare Zeichen ausgegeben. Die obere Hardcopy auf der nächsten Seite gibt dieselbe Anzeige wieder, aber mit Shift-JKL. Der schwarze Fladen in der linken Hälfte ist nicht gerade klassische Kunst, aber er ist mit der HRG entstanden, um die Wirkung des Programms zu zeigen. Vor dem unteren Ausdruck wurde die HRG ausgeblendet, aber nicht gelöscht. Das Programm "merkt" das und druckt sie nicht mit. Auf diese Weise kann man z. B. invers drucken, ohne deshalb gleich den Einschaltmüll des Graphikspeichers mit aufs Papier zu kriegen.

Diese Routine liegt in SYS27/SYS. Wer Interesse hat, sie in sein DOS einzubauen (klappt nur mit dem G3s), kann mir im frankierten und adressierten Rückumschlag eine Systemdiskette (G-DOS 2.4) schicken.

Annulf Sopp

```
----- GENIE-DOS INFO -----
Tastatur: 4516h           Speicherende (HIMEM): FFFFh
Monitor:  4505h           Kopfzeilen: 00h, Fußzeilen: 00h
Drucker:  058Dh          Druckzeile: 0Eh von 48h
----- ZEICHENSATZ -----
!"#$%&'()*+,-./0123456789:;<=>?
$ABCDEFGHIJKLMNPOQRSTUVWXYZÄÜ^_`abcdefghijklmnopqrstuvwxyzäöüß
```

```
----- AKTIVE PORTS -----
D2=44 D3=44 D4=00 D5=00 E0=9F E1=9F E2=9F E3=9F E8=3F
E9=3F EA=3F EB=3F EC=80 ED=29 EE=03 F1=AA F7=00 F9=40
FA=84 FD=3F
```

Hau rein, is Tango!

```

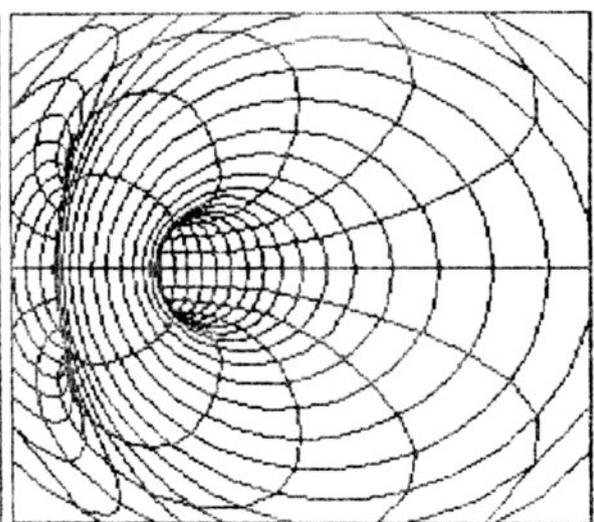
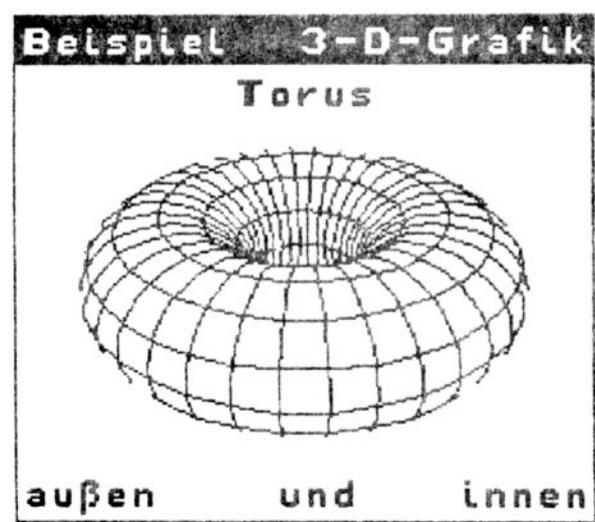
----- GENIE-DOS INFO -----
Tastatur: 4516h                               Speicherende (HIMEM): FFFFh
Monitor: 4505h                                Kopfzeilen: 00h, Fußzeilen: 00h
Drucker: 058Dh                               Druckzeile: 38h von 48h
----- ZEICHENSATZ -----
@ABCDEFGHIJKLMNÖPQRSTUVWXYZ↑↓↔° !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNÖPQRSTUVWXYZÄÜÛ^π$abcdefghijklmnopqrstuövwxyzäöüß#
@ABCDEFGHIJKLMNÖPQRSTUVWXYZ↑↓↔° !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNÖPQRSTUVWXYZÄÜÛ^π$abcdefghijklmnopqrstuövwxyzäöüß#
----- AKTIVE PORTS -----
D2=44 D3=44 D4=00 D5=00 E0=9F E1=9F E2=9F E3=9F E8=3F
E9=3F EA=3F EB=3F EC=80 ED=2E EE=10 EF=45 F1=AA F7=C0
F9=40 FA=86 FD=3F
Hau rein, is Tango!

```

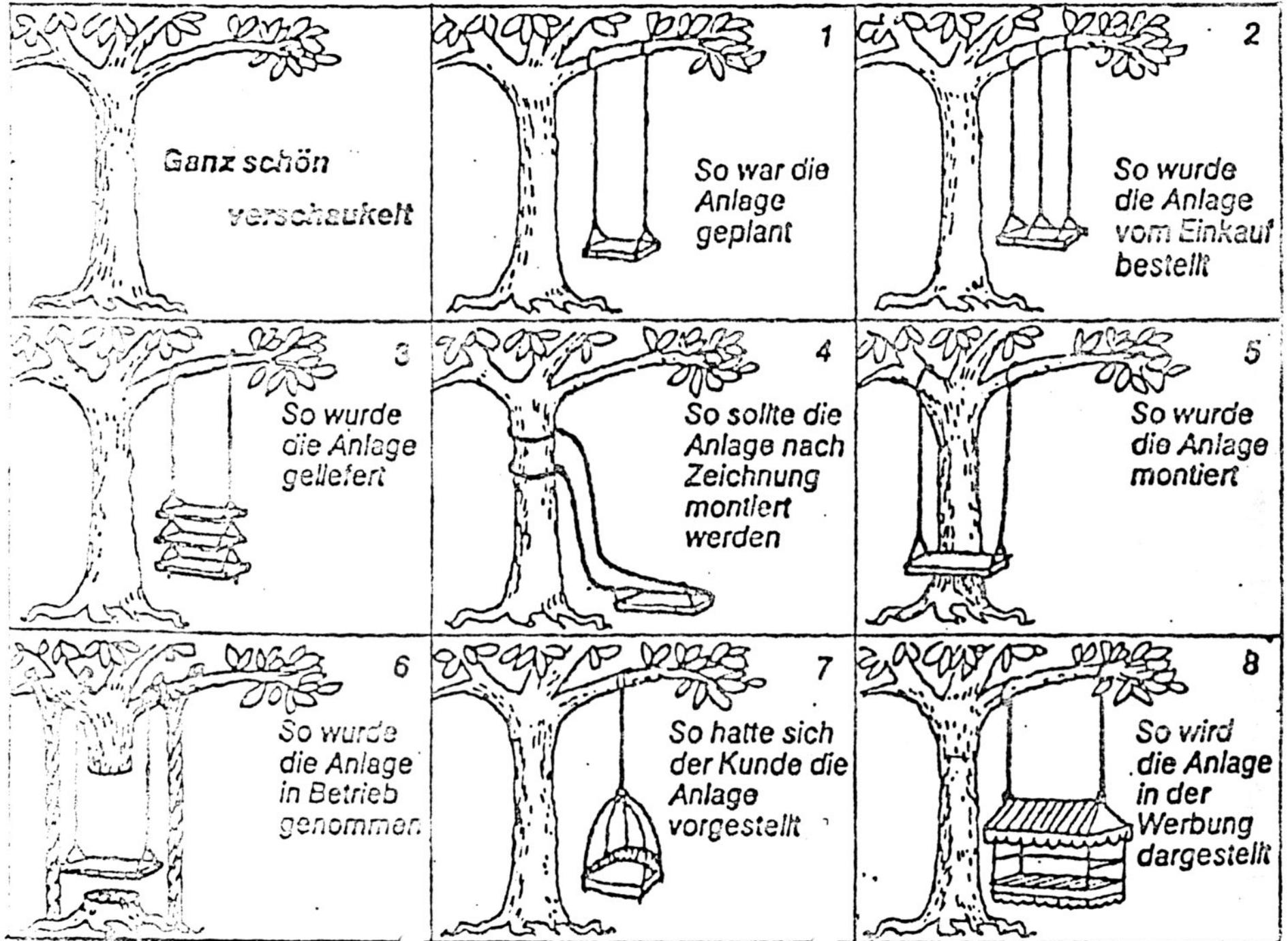
```

----- GENIE-DOS INFO -----
Tastatur: 4516h                               Speicherende (HIMEM): FFFFh
Monitor: 4505h                                Kopfzeilen: 00h, Fußzeilen: 00h
Drucker: 058Dh                               Druckzeile: 38h von 48h
----- ZEICHENSATZ -----
@ABCDEFGHIJKLMNÖPQRSTUVWXYZ↑↓↔° !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNÖPQRSTUVWXYZÄÜÛ^π$abcdefghijklmnopqrstuövwxyzäöüß#
@ABCDEFGHIJKLMNÖPQRSTUVWXYZ↑↓↔° !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNÖPQRSTUVWXYZÄÜÛ^π$abcdefghijklmnopqrstuövwxyzäöüß#
----- AKTIVE PORTS -----
D2=44 D3=44 D4=00 D5=00 E0=9F E1=9F E2=9F E3=9F E8=3F
E9=3F EA=3F EB=3F EC=80 ED=2E EE=10 EF=45 F1=AA F7=C0
F9=40 FA=84 FD=3F
Hau rein, is Tango!

```



(C) Torus by TCS, Hardcopies by Sopp



Psychologische Aspekte des Programmierens

Ruedi Baumann *
H. R. Schärer, Zeichnungen

Software ist, vergleichbar einem Kind, das grossgezogen wird, ein weitgehend exaktes Spiegelbild des psychischen Zustandes des einzelnen, des Teams und der gesamten Organisation, für die sie erstellt wird.

1. Einleitung

Die Aufgabe des Programmierers besteht darin, Problemstellungen auf eindeutige, handfeste Strukturen zurückzuführen. Er muss die zu lösende Aufgabe «berechenbar» machen. Vieldeutigkeiten, Erwägungen, Gefühle, Stimmungen, Ängste, Motivationen und Überzeugungen sind dem Programmierer suspekt, weil er sie in seiner Modellwelt nicht erfassen kann. Die Abneigung gegen alle diese «unberechenbaren» Dinge und ihre Verbannung aus der eigenen Gedankenwelt führt leicht zu einer arroganten Haltung denen gegenüber, deren Wissen weniger greifbar und eindeutig ist. Die Beziehung eines Programmier-

ers zu seinen Programmen hat Konsequenzen auf sein Selbstbewusstsein. Wenn der Computer einen Fehler in seinem Programm findet, kann das für ihn folgendes bedeuten: Mein Programm ist fehlerhaft. Das Programm ist ein Teil von mir, ein Spiegelbild, es trägt meinen Namen. Somit bin ich fehlerhaft! Umworben und gefürchtet, hochgepriesen und scharf kritisiert hat der Programmierer selber Mühe, sich in der wirtschaftlichen Umgebung zurechtzufinden. Programmieren gilt, wie Schachspielen oder Beschäftigung mit reiner Mathematik, allgemein als eine extrem rationale Tätigkeit. Anhand von Aussagen verschiedener Programmierer versuchen die folgenden «Porträts» den Nachweis zu erbringen, dass diese Auffassung nicht unbedingt richtig ist.

* GD SBB, Abt. Kraftwerke, 3001 Bern.



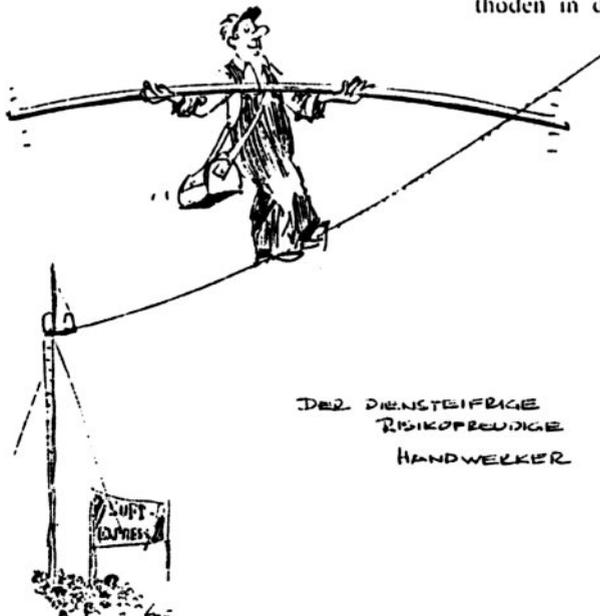
DIE UNENTBEHRLICHE SCHLÜSSELFIGUR

2. Porträts

2.1 Die unentbehrliche Schlüsselfigur

Sie ist nicht nur in der Programmierergemeinde zu finden, tritt hier aber besonders schmerzhaft in Erscheinung. Der Grund: Die üblichen Dokumentationsmethoden in der Software haben

noch nicht den Stand erreicht, der nötig wäre, um sie personenunabhängig zu machen. Hat der Programmierer in einem Projekt durch besondere Leistungen den «Star»-Status erreicht, geniesst er eine für das Unternehmen gefährlich starke Stellung. Nützt er sie aus, kann das ein Unternehmen teuer zu stehen kommen.



DER DIENSTEIFRIGE
RISIKOFREUDIGE
HANDWERKER

2.2 Der dienstefrige, risikofreudige Handwerker

Durch seine optimistische Initiative und seine energische Hilfsbereitschaft kann er grosses Unheil anrichten. Keine Aufgabe ist ihm zu schwierig, kein Problem zu kompliziert und kein Termin zu kurz. Die Überlebenschancen mit einer solchen dienstefrigen, risikofreudigen Einstellung sind gering.

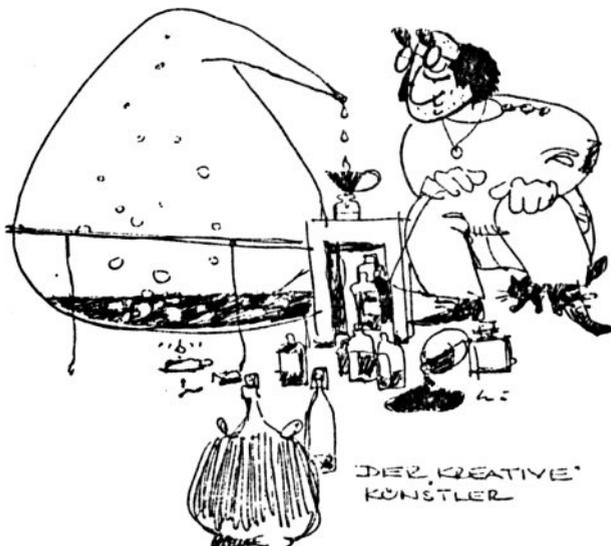


DER GENIALE FACHIDIOT

2.3 Der geniale Fachidiot

Er programmiert in mehreren Hochsprachen wie Pascal, Fortran, Cobol oder Basic und beherrscht meist noch mehrere Assemblerdialekte. Die eigentliche Problemlösung interessiert ihn

eigentlich nur sekundär. Viel wichtiger sind für ihn «Overlays», «Exception-Handlung», Mehrfachverschachtelungen oder indexsequentielle Datenkonstruktionen. Echte Kundenprobleme sind einem genialen Fachidioten unwürdig.



DER 'KREATIVE' KÜNSTLER

2.4 Der kreative Künstler

Er fühlt sich selber als Rechenkern und entwirft das Programm vollständig in seinem Geist. Er läuft durch das Programm,

durch Schleifen, Sprünge usw. Am Schluss muss das Programm ein ästhetisches Bild ergeben. Trifft dies nicht zu, weiss der kreative Künstler, dass das Programm nicht laufen wird.



DIE SCHWEIGSAMEN PRIESTER

2.5 Der schweigsame Priester
Dieser Typ geht beim Programmieren in einen anderen Bewusstseinszustand über, zum Beispiel so etwas wie Trance. Das Gefühl ist ausgeschaltet. Das Programm wird nach einer

langen Periode abgeschirmter Tätigkeit wie durch Zauberei plötzlich verfügbar. Das Image des schweigsamen Priesters wird geschützt durch die weitverbreitete Ehrfurcht vor der «Software».



DER FORSCHENDE SCHWÄRMER

2.6 Der forschende Schwärmer

Nur das Unbekannte ist ihm gut genug, um morgen die Herausforderungen von heute zu meistern. Er fühlt sich in der Lage, jede beliebige moderne Softwaresprache auf einer fremden

Hardware zu implementieren und zu beherrschen. Der forschende Schwärmer findet die Fehler im Schlaf. Er weiss, wo der Punkt ist, eine Sache abzubauen. Wenn er danach schlafen geht, hat er nach dem Aufwachen den Fehler gefunden.



DIE GESTRESSTESTE NACHTEULE

2.7 Die gestresste Nachteule
 Sie hat in einem geregelten Arbeitszeitsystem nichts zu suchen, denn sie ist meist zu den unkonventionellsten Zeiten anzutreffen. Gearbeitet wird bis zum

Morgengrauen, inmitten dicker Rauchschwaden und umgeben von starkem Kaffeedunst. Rückt der Liefertermin näher, sollte die Nacht einer gestressten Nachteule 24 Stunden dauern.



DER BITSTRICKER/ANBETER

2.8 Der Bit-Anbeter
 Masken und Menüs, bestehend aus Bits, Bytes und Interrupts werden vom Bit-Anbeter gestaltet. Er besitzt einen sechsten Sinn, um zu entscheiden, ob ein schwieriges Einzelproblem

«geht» oder «nicht geht». Nachdem das Gefühl auf «es geht» steht, versucht er eine Lösung für das Problem zu finden. Dabei missachtet der Bit-Anbeter meist alle komplizierten Randbedingungen und kämpft sich nach «Bit-Hackerart» durch.



2.9 Der Chip-Anbeter

Für ihn ist es Ehrensache, nur Software für Chips zu schreiben, die auf dem Markt erst angekündigt, aber noch nicht erhältlich sind. Nur der allerletzte Schrei aus dem Silicon Valley ermöglicht eine effiziente, raffinierte

Problemlösung. Der Chip-Anbeter besitzt ein uneingeschränktes Vertrauen in den technologischen Fortschritt.



DER VERBISSENE, WELTFREMD-
 EINZELKÄMPFER

2.10 Der verbissene, weltfremde Einzelkämpfer
 Entgegen der Theorie der «Kreativität in der stillen Kammer» braucht der Einzelkämpfer ein Terminal. Er arbeitet wenn er codiert, und das ausschliess-

lich am Terminal. Der verbissene, weltfremde Einzelkämpfer schreibt seine Prozeduren von «ausen nach innen» und riskiert dabei, durch Interpretationsfehler auf den Holzweg abzugleiten.

3. Fazit

Aus diesen Beobachtungen erkennen wir, dass die Probleme, die wir mit der Software haben, in unserer eigenen Psyche beheimatet sind und dass wir sie lediglich auf unsere Maschinen projiziert haben. Wollen wir von unseren Programmierern erwarten, dass sie vollkommener werden als wir selber sind? Haben Programmierleistungen nicht

viel mehr mit Kreativität und mit Kunst zu tun, als wir gemeinhin wahrhaben wollen?

Literatur:

Brooks P., The Mythical Man-Month, Addison-Wesley Publishing Company, London 1975. Weinberg G., The Psychology of Computer Programming, Van Nostrand Reinhold Company, London 1971.

Mehr Fernsehen für's Geld

Das Genie III s (und vermutlich einige andere Genies "darunter") arbeitet im Gegensatz zum Genie I/II mit einem Video-Controller (CRTC, cathode ray tube controller). Der hat eine Reihe von Registern, mit denen gewisse Bildschirmzustände programmiert werden. G-DOS 2.4 unterstützt lt. Handbuch diese Möglichkeiten. In einem Falle ist das aber pure Hochstapelei: Der Befehl ## soll angeblich in der Lage sein, u. a. die beiden Bildschirmseiten des Formats 16*64 Zeichen untereinander auszutauschen. Er hält sich aber vornehm zurück. Jedenfalls ist mir dergleichen noch nie gelungen. Bis heute.

Dazu ein wenig Theorie: In den Registern 12 und 13 des CRTC ist die relative Startadresse des angezeigten Bildschirms im Verhältnis zur Startadresse des Video-RAMs abgelegt. Dieses beginnt bei 3800h. Beim Kleinbildschirm 16*64 liegt die erste angezeigte Adresse bei 3C00h, also 0400h Bytes hoch im Video-RAM. Daher wird beim Umschalten auf diesen "Kleinbildschirm" 0400h in die Register 12 und 13 des Controllers eingeschrieben. Dazu wird zunächst 0Ch (12d) auf den Port F6h ausgegeben, um das Register Nr. 12 zu adressieren. Der Port F7h kriegt dann das MSB dieses Anzeige-Offsets mitgeteilt. Anschließend wird über F6h das Register 13 aufgeweckt, um über F7h das LSB ausgegeben zu können.

Um nun eine Bildschirmseite in die andere zu kopieren, beide auszutauschen, die eine oder andere vollzumalen usw., muß demnach das MSB des Anzeige-Offsets entweder 00h oder 04h lauten. Auf diese Weise wird die obere oder die untere Bildschirmseite adressiert. Den CRTC interessiert nämlich nicht die Bohne, wo die Designer des G-DOS oder des Genie den Bildschirm gerne memory mapped haben möchten. Er schreibt stur dorthin, wo es ihm seine Register 12 und 13 vorschlagen. Schreibt man nun nach Ausgabe des Offsets 0000h einen Text ab 3800h (Großbildschirm) und gibt dann den Offset 0400h aus (Kleinbildschirm), dann ist der Text verschwunden, weil nicht dort gelesen wird, wo er steht.

Das ist das Funktionsprinzip der beiden Bildschirmseiten, nach dem der ##-Befehl - angeblich, leider nicht wirklich - arbeitet. Um in einem Assembler- oder BASIC-Programm dennoch dieses interessante Feature des CRTC nutzen zu können, möchte ich hier ein Strickmuster vorstellen, nach dem der Bildschirm dem User aus der Hand frißt. Weil sie einfacher zu verstehen ist, zunächst die BASIC-Variante:

In der Zeile 10 wird mit dem Library-Befehl ##,V der Bildschirm auf 32*64 Zeichen eingestellt. Er liegt nun im Bereich 3800-3FFFh. CLS löscht ihn. Links oben wird der Text mit dem Hinweis auf die Seite 0 angezeigt. Mit dem Library-Befehl 64 wird auf den Bildschirm mit 16*64 Zeichen umgeschaltet (3C00-3FFFh). Wieder links oben, aber ein ganzes Kilobyte höher wird der Hinweis auf die Seite 1 geprintet. Danach wird das Offset-Register 12 adressiert. Es genügt nämlich, das MSB des Offsets zu verändern. Das LSB ist immer 00h. In der Zeile 20 wird eben dieser Offset abwechselnd zu 0000h oder zu 0400h definiert. Dazwischen liegt jeweils eine Verzögerungsschleife, um den Effekt beobachten zu können: Beide zuvor in den Bildschirm geladenen Texte kommen abwechselnd zur Anzeige.

Das darüber gelistete Maschinenprogramm tut genau dasselbe. Die ausführlichen Kommentare erklären gewiß, was sich da tut. Ich möchte darüber hinaus lediglich noch auf die Unterprogramme eingehen, die gerollt werden:

Das UP an 3497h schaltet auf die maximale Auflösung mit 32*64 Zeichen. Dazu holt es die entsprechenden CRTC-Parameter aus 3790h ff. und gibt sie an den Controller aus. 34A1h schaltet auf die gleiche Weise auf 16*64 Zeichen, die kleinste Auflösung. Der Vollständigkeit halber seien noch die beiden Unterprogramme für die zwei verbleibenden Bildschirmfor-

mate genannt: 34A6h schaltet auf 24*64, 349Ch auf 25*80 Zeichen. Das UP an 01C9h löscht den Bildschirm bei jedem Format. Mit der Routine an 4467h wird ein Text, auf den HL zeigt, auf den Bildschirm übertragen. Er muß mit 03h oder 0Dh abgeschlossen sein.

Beide Programmvarianten lassen sich mit den von Fall zu Fall erforderlichen Modifikationen in eigene Programme einbauen. Eine denkbare Anwendung: Auf Knopfdruck soll der User ein Menü angezeigt bekommen. Dazu wird nach Umschalten auf den Großbildschirm das Menü einfach mit PRINT in den Bildschirm geladen. Nach Rückkehr zum Kleinbildschirm bleibt dieser Text erhalten. Er wartet sozusagen im Keller. Bei Bedarf braucht nur das Register 12 des CRTC über Port F6h adressiert zu werden, damit man sodann 00 auf Port F7h ausgeben kann. Damit ist der Bildschirm ab 3800h definiert. Der versteckte Text erscheint, der zuletzt angezeigte verschwindet. Nach Ausgabe von 04h auf Port F7h werden beide Bildschirmseiten wieder zurückgetauscht. Es ist, als wäre nichts geschehen.

Annulf Sopp

```

5200          00001      ORG      5200h
5200 CD9734  00002 start CALL    3497h      ;auf 32*64 Z. schalten
5203 CDC901  00003      CALL    01c9h      ;Bildsch. ab 3800 löschen
5206 212852  00004      LD      HL,text0    ;Text "Seite 0"
5209 CD6744  00005      CALL    4467h      ;anzeigen
520C CDA134  00006      CALL    34a1h      ;auf 16*64 Z. schalten
520F 214A52  00007      LD      HL,text1    ;Text "Seite 1"
5212 CD6744  00008      CALL    4467h      ;anzeigen
5215 3E0C    00009      LD      A,0ch      ;Anz.-Offset-Reg. d. CRTC
5217 D3F6    00010      OUT    (0f6h),A    ;adressieren
5219 AF      00011      XOR    A           ;A ← 00 (Offset-Byte)
521A F5      00012 loop  PUSH   AF           ;Akku retten
521B 01FFFF  00013      LD      BC,0ffffh  ;Zähler für Warteschleife
521E CD6000  00014      CALL   0060h      ;etwas trödeln
5221 F1      00015      POP    AF           ;Akku zurück
5222 EE04    00016      XOR    04h        ;Offset-Bit umschalten
5224 D3F7    00017      OUT    (0f7h),A    ;neu ausgeben
5226 18F2    00018      JR     loop        ;und so weiter
5228 44      00019 text0  DM     'Dies ist Seite 0 des Bildschirms.',03h
524A 55      00020 text1  DM     'Und dieses hier ist Seite 1.',03h
          00021 endprg
5200          00022      END     start

```

00000 Fehler

Und hier dasselbe in BASIC:

```

10 CMD"##.V":CLS:PRINT"Dies ist Seite 0 des Bildschirms.":CMD"64":
PRINT"Und dieses hier ist Seite 1.":OUT&HF6,12
' Texte schreiben. Anzeige-Offset-Register adressieren

```

```

20 FOR I=0T0500:NEXT:OUT&HF7,A:FOR I=0T0500:NEXT:OUT&HF7,0:GOTO20
' Zwischen den Offsets 0000 und 0400 hin- und herschalten

```

P e r s o n e l l e s i m I N F O

MNUM	NACHNAME	VORNAME	USER-NAME	COMPUTER	LAUFM	KOPP	DRUCKER	BESOND
850264	Albertz	Dieter		GENIE I	2-XX-SSSD		ITOH 8510	
830611	Bach	Siggi		GENIE II	3		NEC 8023	
850768	Berner	Horst		TRS-80 MOD I	1		EPSON FX 80	
851077	Bernhardt	Helmut		GENIE I	2-80-DSDD		EPSON MX 80	KOMTEK, MC CP/M, HRG
860612	Bertschi	Rudolf		GENIE				
840120	Blaschek	Manfred		TRS-80 MOD I	CASSETTE			
850643	Bochtler	Peter		GENIE I	2-40-SSDD		MT 80	TRS MODELL 100
840738	Bornschlegel	Hans		GENIE I	2			
840413	Brake	Thilo		TRS-80 MOD I	2		GEMINI 10 X	
860207	Braun	Harald		GENIE 3003	CASSETTE			EXPANDER
841117	Böckling	Ulrich		TRS-80 MOD I	2-80-DSDD		ITOH	HRG
840646	Dürhammer	Ulf		TRS-80 MOD I	2		STAR DP 8480	I/O IN 19" GENAEUSE
850870	Engel	Jürgen		GENIE I	2-80-DSSD			
840127	Fischbeck	Uwe		COLOUR GENIE				
840110	Folkerts	Ralf	RAFO	GENIE I	4-40-SSDD-53	DATAPH	NEC 8023 BC	TANDY 3
830816	Frey	Wolfgang		GENIE I	1			
851137	Geißler	Hans - Peter		GENIE II	3-40-DSDD	JA	GEMINI 10X	HRG 1B, U-24
841055	Gerblinger	Dieter		COLOUR GENIE	2		ITOH 8510 A	PROMMER
860611	Goltz	Dietmar		TRS-80 MOD I	1-41-SSDD		EPSON FX 80	SPEEDUP, CHARGEN
830507	Grajewski	Werner		GENIE I	2		STAR DP 510	
830815	Grundmann	Waldemar		GES. PRGM.				SO ZIENLICH ALLES
841057	Hanss	Dirk		TRS-80 MOD I	3-40-DSDD			OMIKRON MAPPER
850161	Henne	Wilhelm		TRS-80 MOD I	2		LP ????	
850767	Hinze	Rolf		GENIE I	2		EPSON FX 80+	
851182	Honcamp	Jochen		SPEEDMASTER8	2		GEMINI 10X	
840544	Hose	Rüdiger		TRS-80 MOD I	2		MX 82	TANDY MODEL 100
850162	Jahn	Markus		COLOUR GENIE	CASSETTE		LOGI. FT5001	
850871	Jaschke	Siegfried		GENIE I	4		ITOH 8510 A	
830419	Karnatz	Michael		COLOUR GENIE	1, CASSETTE		GEMINI 10X	PROMMER
860310	Kellermann	Reinhold		GENIE II	4-39-SSDD5		EPSON MX 80	HETRON DOUBLER
840234	Klein	Gerhard		TRS-80 MOD I	1		MX 80 FT	
831023	Kröher	Paul		GENIE I	1-80-DSDD-5			
860209	Krüger	Karl - Herbert		TRS MODELL I	3-80-SSDD-5		MX 80	CP/M2.2,HRG,PROMMER
840336	Kummerow	Jens		GENIE I	2-80-DSDD		DP 510	
840748	Küster	Heinz - Gerd	HEINZ GERD	GENIE I	2		PRAXIS 35	
850142	Langguth	Hans - Otto		GENIE 2 U. 3	6-80-DSDD	TANDY	ITOH 1550	HRG
851080	Liebig	Erich		GENIE I	2-80-SSSD		EPSON MX 80	
851079	Linnenschmidt	Georg		TRS-80 MOD I	2-80-SSSD			
831224	Linneweber	Manfred		TRS-80 MOD 3			MX 80 FT	
850625	Loose	Gerhard	TRS 80	TRS-80 MOD I	2-40-DSDD	JA	TANDY LP VII	
851076	Lutz	Eckhard		TRS-80 MOD I				GRS 80
850765	Mahlert	Herbert		GENIE II	2-40-SSDD		NEC 8023 BC	HRG
860613	Mattern	Peter		TRS-80 MOD 3	2-40-SSDD		EPSON MX82FT	RS232
830508	May	Holger		GENIE I	1		FS TO 100 S	
840126	Meier	Hans-Christian		COLOUR GENIE	1		GEMINI 10X	
850766	Misioch	Waldemar	?????	GENIE I	2		MX 80	NDR KLEINCOMPUTER
850163	Mühlenbein	Klaus - Jürgen		TRS-80 MOD I	3-40-DSDD		RX 80 F/T	HRG; INFOTHEK
851181	Nepp	Alfred		GENIE IIS	2-40-DSDD		EX 80 F/T+	
850614	Obermann	Hartmut		TRS-80 MOD I	3-80-DSDD		RX 80 F/T+	
840954	Offermann	Hartmut		GENIE I	CASSETTE		GP 100 MARK2	
840339	Omasreiter	Inngard		GENIE I,KOMT	2-40-SSDD-5		PC 8023	
860106	Posing	Carlo		TRS-80 M4,4P	2X2	TANDY	LQ 1500	
841056	Punzet	Alfred		GENIE I	3-XX-DSDD		TELETYPE	
840129	Reichelsdorfer	Wolfgang		TRS-80 MOD I	3-80-7S?D		ITOH 8510 A	

P E R S O N E L I E S I M I N F O

MNUM	NACHNAME	VORNAME	USER-NAME	COMPUTER	LAUFW	KOPP	DRUCKER	BESOND
851232	Rubes	Karl		TRS-80 MOD I	3		EPSON FX 80	SPOOLER 48K,HRG
850751	Sandkühler	Peter		GENIE I	2-40-SSDD		EPSON RX 80	
850203	Sanz	Alfonso		GENIE I	X-40-SSDD		ITOH 8510B	HRG
850521	Scharnhölz	Theodor		TRS-80 MOD I	2-XX-XSXD			
830509	Scheidt	Uwe von		TRS-80 MOD I	2		RX 80 FT	
850160	Scheloske	Holger		GENIE II	CASSETTE			
851075	Schiegl	Margit		GENIE I	2-40-SSDD			
850745	Schmid	Alexander		GENIE II, 2S	4		EPSON RX 80	
830302	Schmidt	Horst		GENIE II	CASSETTE			
840235	Schmitz	Paul - Jürgen		GENIE I	2-40-SSDD-5	MC	BROTHER CE60	
850633	Schopen	Peter		TRS-80 MOD I	2		EPSON	COLOR GENIE M. CASS.
841158	Schäfer	Walter		GENIE I	1-80-DSDD		GEMINI 10X	
841259	Seelmann-Eggeber	Jörg		GENIE I	1			
850972	Seibold	Joachim		TRS-80 MOD 3	1-40-DSDD		LINEPR. VII	
850630	Sikora	Ernst		GENIE I	3-80-DSDD		EPSON RX 80	EG64, HRG
8501	Sopp	Arnulf		GENIE I	2-80-DSDD-5	AC III	GEMINI 10X	HRG, EG64MBA
830401	Spieß	Peter		GENIE II	4-80-DSDD-53		NEC 8023	
840340	Stark	Othmar		GENIE I, 2S	4-80-DSDD-53		MX 80 FT	
840128	Thalmeier	Gregor		TRS-80 MOD I	3-70-DSDD	JA	MX 80	
860208	Thum	Helmut		GENIE I	3		ITOH 8510A	HRG 1B
830306	Thönnißen	Heinrich		TRS-80 MOD I	2		MX 80 FT	HRG 1B
840749	Topp	Gerhard		TRS-80 MOD I	2, CASSETTE		RX 80 FT	ECB BUS
850869	Wagner	Günther		TRS-80 MOD 3	3-80-DSDD		LPINEPR. VI	
860105	Wala	Erwin		TRS-80 MOD I	3-80-SSDD		LINEPRINT. I	EPROMMER,RS232,BUFFE
850418	Weidmann	Josef		GENIE I	2-80-DSDD		GEMINI 10X	
850973	Weikamp	Horst		TRS-80 MOD I	2-40-DSDD		MX 80 GRAFTR	HRG
850604	Weiss	Dieter		GENIE I			STAR 510,PRA	
840750	Wittmann	Reinhard		GENIE I	2-40-SSDD		PRINTSTAR101	MICROPROFFESSOR
840852	Wolf	Klaus		TRS-80 MOD I	1, CASSETTE		?	

Im August haben Geburtstag:

Helmut Bernhardt	Waldemar Misioch
Klaus Wolf	Harald Braun
Hans Bornschlegel	Hartmut Obermann
Alfred Punzet	Dirk Hanss
	und Günther Wagner

Meinen herzlichsten Glückwunsch !

Ein neues Mitglied haben wir auch:

Peter Mattern aus Bremervörde. Er besitzt ein TRS-80 Modell III mit 2 Diskettenlaufwerken und einer RS232. Die Druckarbeit, die bei seinem Interessenschwerpunkt 'Software' sicher in Mengen anfällt (er interessiert sich für diverse Programmiersprachen) bewältigt ein Epson MX82F für ihn.

Mehrere Mitglieder haben den Club leider verlassen:

Martin Rüttgers, Thomas Althaus, Olaf Thun (er hat sich einen Atari 520 ST gekauft) und Friedrich Horn - dieser jedoch erst nach der zweiten Mahnung. Ich hatte mich schon gefreut, daß unsere 'Gemeinde' wieder Zuwachs bekommt und stetig wächst - da treten auch schon 'ne Menge wieder aus.

SOFTWARE- WETTBEWERB

Die Ausschreibung erfolgt im Genie/TRS80 User-Club Bremerhaven und im Club-80 gleichzeitig.

Aufgabenstellung:

Zu erstellen ist ein BASIC-Programm mit maximal 33 Zeilen. Die Thematik ist freigestellt. Das Programm muß auf den Rechnern Genie I, II, IIs, (III), oder TRS80 Mod. 1, Mod. 3 lauffähig sein und sollte unter der Herrschaft von NEWDOS80 oder G-DOS stehen. Je mehr "Spezialitäten" eingebaut werden, umso höher fällt die Bewertung aus (z.B. Diskettenroutinen, ansprechen der HRG, Druckerausgabe, Maschinenroutinen, etc.). Das Programm muß natürlich mit einer entsprechenden Betriebsanleitung ausführlich dokumentiert werden. Abtippen aus Zeitschriften (auch aus Clubinfos) oder einsenden von kommerziellen Programmen ist verboten. Im Namen aller Teilnehmer bitte ich um Fairness !!!

Zu gewinnen gibt's:

1. Preis: elektronische Schreibmaschine BROTHER EP-20
2. Preis: Ein 10er Pack Disketten
3. Preis: Buch "Start in die Computergrafik"
- 4.-10. Preis: je eine Leerdiskette

Die Preise wurden gestiftet von der Firma:

Peter Spieß
Offsetdruck + EDV-Zubehör
Trugenhofenerstr. 27
8859 Rennertshofen 1
Tel: 08434/454

Einsendeschluß ist der 30.09.86
(Datum des Poststempels)

Der Rechtsweg ist ausgeschlossen! Die Gewinner werden schriftlich benachrichtigt. Die Bekanntgabe erfolgt dann im jeweiligen Clubinfo. Bitte die Diskettenformatierung angeben! Die Disk's werden nach der Auswertung an den Einsender zurückgeschickt. Einsendeadresse:

Genie/TRS80
User-Club
Peter Spieß
Trugenhofenerstr. 27
8859 Rennertshofen 1