



### Lebenslauf

Anfangen hat meine Computerbegeisterung mit etwa zehn Jahren. Damals schenkte meine Großmutter mir einige Spiele von Atari. Diese Spiele waren aber nur kurzzeitig von Interesse.

Ich kaufte mir fast alle Computer-Zeitschriften und konnte in den Kaufhäusern nicht an den Computern vorbeigehen, die dort ausgestellt waren.

Als mein Vater sich für seine Handelsvertretung einen Commodore-Computer zulegte, hatte ich Gelegenheit, durch meine Vorkenntnisse, die Gespräche zwischen einer Leasingfirma und meinem Vater zu beeinflussen.

Meine Großmutter unterstützte mein Computerhobby auch finanziell und ich war mit 15 Jahren in der Lage, mir meinen eigenen C 64 mit Floppy zu kaufen. Nach den großen Ferien komme ich in die gymnasiale Oberstufe. Als Abiturfächer werde ich Physik, Mathematik und Englisch wählen, da ich Informatik studieren möchte. Diesen Beruf würde ich gern in der Forschung ausüben.

Zur Zeit überarbeite ich das EDV-Programm der Firma meines Vaters, um es bedienungsfreundlicher zu gestalten und dieses auch für einen Tennisclub von etwa 500 Mitgliedern (Verwaltung, Mitteilungen, Beitrags-einzug) zu verwenden.

(Stefan Becker)

# »Dokumentationshilfe«: Cross-Referenz-Liste C 64

**Die Dokumentation von Basic-Programmen ist manchmal eine mühevoll Aufgabe. Trotzdem ist sie sinnvoll oder sogar notwendig. Aus diesem Grund schlugen wir einen Programmierwettbewerb zur Dokumentationshilfe vor, dessen Ergebnis hier vor Ihnen liegt.**

Jeder Basic-Programmierer kennt das Problem: Man schreibt lange und auch komplizierte Programme und merkt, daß langsam aber sicher der Überblick verloren geht. Oder auch, wenn versucht wird, ein fremdes, nicht selbst erstelltes Programm zu verstehen: Was bedeutet diese Variable, und wo taucht sie im Programm noch auf? Von wo und wie oft wird diese Programmzeile angesprungen?

Bei gut strukturierten Programmen findet man relativ schnell durch, aber die sind selten zu finden. Doch selbst dann ist eine gute Beschreibung eine hilfreiche Sache und erleichtert den späteren Wiedereinstieg.

Eine Crossreferenzliste unterstützt bei der Erstellung einer Dokumentation. Gerade bei kommerziellen Software-Projekten wird nicht auf sie verzichtet. Sie enthält dabei nicht nur eine Aufzählung aller Sprünge (sofern bei höheren Programmiersprachen überhaupt vorhanden), sondern vor allem eine komplette Variablenliste. Und diese Funktionen hat auch unser Programm. Im einzelnen enthält sie:

1. Liste aller Zeilen, in denen Sprungbefehle enthalten sind. Angegeben wird die Zeilennummer und die angesprungene Zeile.
2. Liste aller Zeilen, die angesprungen werden. Angegeben sind die Zeilennummer und alle Zeilen, von denen aus diese Zeile angesprungen wird. Eine sehr nützliche Information.
3. Eine Liste aller im Programm vorkommenden Variablen. Angegeben wird, alphabetisch sortiert, der

Variablenname und die Zeilen, in denen sie vorkommt. Zusätzlich hat man die Möglichkeit, zu jeder Variablen einen kurzen Kommentar mit einzugeben.

4. Liste aller Variablen mit dem vorhin genannten Kommentar, aber ohne Hinweise auf die Zeilen, in denen sie vorkommen.

Mit diesen Informationen kann man schon etwas anfangen. Doch jetzt etwas für Programmierer selbst:

### Wie funktioniert das?

Wie sucht man nach Variablen? Wie nach Sprungbefehlen? Wie so oft, ist die Lösung ganz einfach.

Wie ein Basic-Programm im Speicher steht, wurde ausführlich im 64'er, Ausgabe 2/85, Seite 87, beschrieben. Fast in der gleichen Form wird ein Programm auch auf Diskette gespeichert. Lediglich zwei Byte für die Ladeadresse des Programms kommen dazu. Beim absoluten Laden eines Programms (LOAD"NAME",8,1) holt sich der Basic-Interpreter diese zwei Byte, um festzustellen, an welcher Stelle im Speicher das Programm beginnt. Beim normalen LOAD-Befehl, also ohne die ",1" am Ende, wird jedes Programm ab Basic-Anfang gesetzt, also ab \$801 (= dezimal 2049).

An einem kleinen Beispiel soll gezeigt werden, wie man sich alle Zeilen anzeigen lassen kann, die den Befehl GOTO enthalten. Sehen Sie sich dazu das Flußdiagramm (Bild 1) an.

Als erstes muß eine »Programmdatei« geöffnet werden. Das geht zum Beispiel mit OPEN2,8,2,"NAME,P,R". »P« steht dabei für Programm und »R« für READ, also lesen.

Dann folgt das Lesen der Byte. Dazu nimmt man den GET#-Befehl. Der INPUT#-Befehl ist zwar wesentlich schneller, »schafft« aber nur maximal 88 Zeichen, benötigt ein Carriage Return (CHR\$(13)) als Endekennzeichen (bei Programmen sind es Null-Byte) und verträgt keine Kommata (in Basic-Programmen oft vorhanden). Außerdem erlaubt uns der GET#-Befehl die Überprüfung einzelner Zeichen direkt nach dem Laden.

Lesen wir jetzt also die ersten Byte eines Basic-Programms. Am Anfang, das sind die ersten zwei Bytes, steht die Startadresse. Bei Basic-Programmen ist sie 2049, hexadezimal ausgedrückt \$0801. Darauf folgen zwei Byte für die Anfangsadresse der nächsten Basic-Zeile (gemeint ist hier die Adresse im RAM, die natürlich auch gespeichert wird). Erst jetzt beginnt die wirkliche Basic-Zeile: Zwei Byte für die Zeilennummer, aufgetrennt in Low- und High-Byte, gefolgt vom Basic-Text. Das Ende einer Basic-Zeile erkennen Sie (und der Basic-Interpreter) an einem Null-Byte. Danach folgt wieder die Anfangsadresse der nächsten Basic-Zeile (Low- und High-Byte) und der nächsten Zeilennummer (auch Low- und High-Byte), danach der Basic-Text, etc., etc.. Das Ende eines Basic-Programms ist gekennzeichnet durch drei aufeinanderfolgende Null-Byte, und auch



führungszeichen überlesen (Zeile 750-790).

In diesem Stadium kommt Arbeit auf Sie zu. Jetzt haben Sie die Möglichkeit, die Variablen mit einem Text zu versehen. Dabei müssen Sie sich allerdings auf 25 Zeichen beschränken.

Antworten Sie auf die Frage »WOLLEN SIE ZU DEN VARIABLEN BEMERKUNGEN EINGEBEN« mit »J«, so können sie mit Cursor UP und DOWN durch die Liste wandern. Die momentane Variable wird jeweils angezeigt. Haben Sie die richtige Stelle gefunden, so drücken Sie »RETURN« und können Ihren Text eingeben. Durch nochmaliges Drücken der »RETURN«-Taste wird der Text gespeichert.

Haben Sie Ihre Texte verteilt, so beginnt das Programm nach Drücken der

Befehl	Token		Befehl	Token		Befehl	Token	
	DEZ	HEX		DEZ	HEX		DEZ	HEX
END	128	80	CONT	154	9A	SGN	180	B4
FOR	129	81	LIST	155	9B	INT	181	B5
NEXT	130	82	CLR	156	9C	ABS	182	B6
DATA	131	83	CMD	157	9D	USR	183	B7
INPUT #	132	84	SYS	158	9E	FRE	184	B8
INPUT	133	85	OPEN	159	9F	POS	185	B9
DIM	134	86	CLOSE	160	A0	SQR	186	BA
READ	135	87	GET	161	A1	RND	187	BB
LET	136	88	NEW	162	A2	LOG	188	BC
GOTO	137	89	TAB	163	A3	EXP	189	BD
RUN	138	8A	TO	164	A4	COS	190	BE
IF	139	8B	FN	165	A5	SIN	191	BF
REST.	140	8C	SPC	166	A6	TAN	192	C0
GOSUB	141	8D	THEN	167	A7	ATN	193	C1
RETURN	142	8E	NOT	168	A8	PEEK	194	C2
REM	143	8F	STEP	169	A9	LEN	195	C3
STOP	144	90	+	170	AA	STR\$	196	C4
ON	145	91	-	171	AB	VAL	197	C5
WAIT	146	92	*	172	AC	ASC	198	C6
LOAD	147	93	/	173	AD	CHR\$	199	C7
SAVE	148	94		174	AE	LEFT\$	200	C8
VERIFY	149	95	AND	175	AF	RIGHT\$	201	C9
DEF	150	96	OR	176	B0	MID\$	202	CA
POKE	151	97	>	177	B1	GO	203	CB
PRINT #	152	98	=	178	B2			
PRINT	153	99	<	179	B3			

**Bild 2. Basic-Befehlswörter und deren Token.**

```

Programmname: test
sprungtabelle
-----
zeile :   sprung auf zeile
      140: 1000 2000 3000

zeile :   wird angesprungen von
      1000: 140
      2000: 140
      3000: 140
    
```

**Bild 3. Diese Sprungtabelle wird ausgegeben, wenn das Programm »Listing 1« untersucht wird.**

```

Variablenliste ohne Zeilennummer
-----
a$( :      eingegebene Werte
aw  :      Parameter ON-Anweisung
aw$ :      siehe aw
b(  :      keine Ahnung
c%( :      wie b(
i   :
j   :
na$ :      Dateiname
x   :      Hilfsvariable
    
```

**Bild 4. Diese Variablenliste erstellt »Cross-Ref 64« nach Untersuchung von »Listing 1«. Die Kommentare müssen natürlich zusätzlich eingegeben werden.**

```

liste der variablen :
-----
a$( :      10   40   50   100
aw  :      130  140 1000 2000 3000
aw$ :      130
b(  :      10   20
c%( :      10
i   :      20  100
j   :      30   40   50   60   90
na$ :      70   80
x   :      90  1000 2000 3000
    
```

**Bild 5. Auch eine Variablenliste ohne Zeilennummern ist möglich.**

»N«-Taste zu drucken (Zeilen 1180 bis 1280). Auch hier können Sie den Druckvorgang wiederholen (Bild 4 und 5).

**Umstellungshinweise**

Dieses Programm ist auf allen Computern lauffähig, die ihren Basic-Text nach Commodoreart speichern. Potentielle »Umschreiber« können sich daher auf die Zeile 330 beschränken. Auch die Druckausgabe ist leicht modifizierbar. Die Sekundäradresse der OPEN-Befehle

in den Zeilen 580 und 1180 stellen den Epson MX-80 über das Print-64-Interface auf Klein/Großschrift um. Dies kann auch durch normale »OPENS« mit nachfolgenden Steuer-codes erfolgen.

Sogar bei einem Kassettenlaufwerk ist das Programm verwendbar. Auch hier ändern Sie in den Zeilen 290 und 730 die OPEN-Befehle (Listing 4).

(Stefan Becker/gk)

```

5 REM X=2
10 DIMA$(100),B(20),C%(20)
20 FORI=1TO20:B(I)=I:NEXT
30 FORJ=1TO20
40 PRINT" WERT NR. "J;:INPUTA$(J)
50 IF A$(J)="X" THEN J=20
60 NEXT J
70 INPUT"PRG.NAME = ";NA$
80 OPEN2,8,2,NA$+"P,W"
90 FORX=1TOJ
100 PRINT#2,A$(I)
110 NEXT
120 CLOSE2
130 INPUT"AUSWAHL = ";AW$:AW=VAL(AW$)
140 ON AW GOSUB 1000,2000,3000
150 END
1000 PRINTAW:X=1:RETURN
2000 PRINTAW:X=2:RETURN
3000 PRINTAW:X=3:RETURN
    
```

**Listing 1. Dieses Programm liefert die Listen Bild 3 bis 5.**

```

10 REM***** <003>
20 REM* PROGRAMNAME : XREF * <236>
30 REM* C-64 * <219>
40 REM* FLOPPY 1541 O. AEHNLICHE * <174>
50 REM* DRUCKER (Z.B. MPS 801) * <247>
60 REM* VON STEFAN BECKER * <127>
70 REM***** <063>
80 CLR:GOTO 150 <069>
90 GET#1,A$:X=ASC(A$+N$):IF(64 AND ST)=0 T
HEN RETURN <169>
100 CLOSE 1:IF PA THEN PA=0:GOTO 520 <255>
110 GOTO 960 <150>
120 GOSUB 90:X1=X:GOSUB 90:X=X1+256*X:RETU
RN <075>
130 IF SP<OB AND,VA<OB AND PS<OB THEN RETU
RN:REM*** GRENZEN ERREICHT ? *** <188>
140 PRINT:PRINT"BITTE DIE VARIABLE OB IN Z
EILE 150 VERGROESSERN.":GOTO 1310 <222>
150 OB=500:DIM SP$(OB),PS$(OB),VA$(OB) <117>
160 REM*** OB IST OBERGRENZE DER ANZAHL DE
R SPRUENGE UND VARIABLEN *** <247>
170 POKE 53280,6:POKE 53281,6 <219>
180 N$=CHR$(0) <032>
190 LE$="(50SPACE)" <105>
200 PRINT CHR$(147)CHR$(9)CHR$(14)CHR$(8)C
HR$(144); <031>
210 PRINT" (RVSON,10SPACE)GROSS-REFERENZ-LI
STER(9SPACE)" <089>
220 PRINT" (2DOWN)AUSGABE AUF ":PRINT" (DOWN
,RVSON) (RVOFF)ILDSCHIRM ODER (SPACE,RV
SON) (RVOFF)RUCKER (3UP)" <109>
230 PRINT SPC(12);:POKE 204,0 <175>
240 GET A$:IF A$<>"B"AND A$<>"D"GOTO 240 <041>
250 IF A$="B"THEN PRINT"ILDSCHIRM.":DV=3:
GOTO 270 <074>
260 PRINT"DRUCKER.":DV=4 <090>
270 PRINT" (DOWN,23SPACE,UP)" <142>
280 OPEN 1,0:PRINT"PROGRAMMNAME: ";:INPUT#
1,NA$:CLOSE 1:PRINT <131>
290 OPEN 2,8,15:OPEN 1,8,2,NA$+" ,P,R":GOSU
B 1290:GOSUB 120:IF X=2049 THEN 310 <067>
300 PRINT:PRINT"DAS PROGRAMM MUSS AB 2049
LIEGEN.":GOTO 1310 <030>
302 : <105>
304 : <107>
305 REM***** <001>
306 REM PASS 1 SPRUNGTABELLE <002>
308 REM***** <004>
309 : <112>
310 PRINT" (CLR)PASS 1 (SUCHEN DER SPRUENGE
) (DOWN)":PA=1 <169>
320 GOSUB 120:GOSUB 120:ZE$=RIGHT$(" (5SPAC
E)" +STR$(X),5):PRINT" (HOME,2DOWN)"ZE$
<079>
330 GOSUB 90:IF X=137 OR X=138 OR X=141 OR
X=167 THEN S1$="":GOTO 360 <074>
340 IF X=0 GOTO 320 <056>
350 GOTO 330 <126>
360 GOSUB 90:IF X=32 GOTO 360 <181>
370 IF X>=48 AND X<=57 THEN S1$=S1$+A$:GOT
O 360 <017>
380 IF S1$=""GOTO 470 <202>
390 S1$=RIGHT$(" (6SPACE)" +S1$,6) <243>
400 IF LEFT$(SP$(SP),5)<>ZE$THEN SP=SP+1:G
OSUB 130:SP$(SP)=ZE$+" : <223>
410 IF LEN(SP$(SP))>70 THEN SP=SP+1:GOSUB
130:SP$(SP)=ZE$+" : <116>
420 IF RIGHT$(SP$(SP),6)<>S1$THEN SP$(SP)=
SP$(SP)+S1$ <148>
430 FOR I=1 TO PS <163>
440 IF LEFT$(PS$(I),6)<>S1$THEN NEXT:PS=I:
GOSUB 130:PS$(I)=S1$+" : <165>
450 IF LEN(PS$(I))>70 THEN NEXT:PS=PS+1:I=
PS:GOSUB 130:PS$(I)=S1$+" : <097>
460 IF RIGHT$(PS$(I),5)<>ZE$THEN PS$(I)=PS
$(I)+" "+ZE$ <209>
470 IF X=0 GOTO 320 <186>
480 IF X=44 THEN S1$="":GOTO 360 <127>
490 IF X=58 OR X>=127 OR(X)=65 AND X<=90)G
OTO 330 <013>
500 PRINT:PRINT" (DOWN)FEHLER IM QUELLPROGR
AMM. ZEILE:"ZE$:GOTO 1310 <150>
502 : <049>
504 : <051>
506 REM ----- <253>
510 REM*** SORTIEREN DER SPRUNGZIELE (FELD
PS$ ( ** <170>
515 REM ----- <007>
516 : <064>
520 FOR I=1 TO PS <254>

```

```

530 FOR J=I TO PS <033>
540 IF LEFT$(PS$(I),5)<LEFT$(PS$(J),5)GOTO
560 <055>
550 PS$(0)=PS$(I):PS$(I)=PS$(J):PS$(J)=PS$
(0) <200>
560 NEXT J <254>
570 NEXT I <007>
574 : <122>
575 REM ----- <112>
576 REM AUSGABE SPRUENGE + SPRUNGZIELE <181>
577 REM ----- <114>
578 : <126>
580 OPEN 1,DV,7:PRINT#1,"PROGRAMMNAME: "NA
$:PRINT#1 <168>
590 PRINT#1,"SPRUNGTABELLE (11SPACE)" <097>
600 PRINT#1,"-----" <037>
605 PRINT#1,"ZEILE : (3SPACE)SPRUNG AUF ZEI
LE (6SPACE)":PRINT#1 <130>
610 FOR I=1 TO SP <088>
620 IF LEFT$(SP$(I-1),5)=LEFT$(SP$(I),5)TH
EN PRINT#1,SPC(7)MID$(SP$(I),7):GOTO 6
40 <027>
630 PRINT#1," SP$(I) <018>
640 NEXT I:PRINT#1 <080>
645 PRINT#1,"ZEILE : (6SPACE)WIRD ANGESPRUN
GEN VON":PRINT#1 <044>
650 PS$(0)="" :FOR I=2 TO PS <249>
660 IF LEFT$(PS$(I-1),6)<>LEFT$(PS$(I),6)T
HEN PRINT#1,PS$(I):GOTO 680 <119>
670 PRINT#1,SPC(6);MID$(PS$(I),7) <180>
680 NEXT I:CLOSE 1 <128>
690 PRINT" (DOWN,RVSON) (RVOFF)OCHMAL/ (RVS
ON) (RVOFF)EITER" <062>
700 GET A$:IF A$<>"N"AND A$<>"W"GOTO 700 <023>
710 IF A$="N"GOTO 580 <034>
712 : <004>
714 : <006>
715 REM***** <156>
716 REM PASS 2 VARIABLE SUCHEN <210>
718 REM***** <159>
719 : <011>
720 PRINT" (CLR)PASS 2 (SUCHEN DER VARIABLE
N) (DOWN)" <194>
730 OPEN 1,8,2,NA$+" ,P,R":GOSUB 1290:GOSUB
120 <034>
740 GOSUB 120:GOSUB 120:ZE$=RIGHT$(" (5SPAC
E)" +STR$(X),5):PRINT" (HOME,2DOWN)"ZE$:
Y=0 <105>
750 GOSUB 90 <230>
760 IF X=0 GOTO 740 <227>
770 IF X=34 OR X=131 OR X=143 GOTO 800 <195>
780 IF X>64 AND X<91 GOTO 850 <087>
790 GOTO 750 <062>
800 IF X=131 THEN Y=1 <016>
810 GOSUB 90:IF X=0 GOTO 740 <070>
820 IF X=34 GOTO 750 <088>
830 IF X=58 AND Y=1 GOTO 750 <083>
840 GOTO 810 <109>
850 V1$=A$ <023>
860 GOSUB 90 <085>
870 IF X=36 OR X=37 OR(X>47 AND X<58)OR(X)
64 AND X<91)THEN V1$=V1$+A$:GOTO 860 <222>
880 IF X=40 THEN V1$=V1$+A$ <042>
890 V1$=LEFT$(V1$+" (4SPACE)",4) <236>
900 FOR I=1 TO VA <111>
910 IF LEFT$(VA$(I),4)<>V1$THEN NEXT:VA=I:
GOSUB 130:VA$(I)=V1$+" (2SPACE)": <093>
920 IF LEN(VA$(I))>50 THEN NEXT:VA=VA+1:I=
VA:GOSUB 130:VA$(I)=VA$+" (2SPACE)": <014>
930 IF RIGHT$(VA$(I),5)<>ZE$THEN VA$(I)=VA
$(I)+" "+ZE$ <133>
940 IF X=0 GOTO 740 <152>
950 GOTO 750 <222>
952 : <245>
954 : <247>
956 REM ----- <193>
960 REM *** SORTIEREN DER VARIABLEN (FELD
VA$ ( ** <158>
965 REM ----- <202>
968 : <005>
970 FOR I=1 TO VA <181>
980 FOR J=I TO VA <216>
990 IF LEFT$(VA$(I),4)>LEFT$(VA$(J),4)GOTO
1010 <005>
1000 VA$(0)=VA$(I):VA$(I)=VA$(J):VA$(J)=VA
$(0) <067>
1010 NEXT J <193>
1020 NEXT I:VA=VA-1:I=VA <140>
1025 : <063>

```

Listing 2. Cross-Referenz-Liste 64

```

1030 PRINT{CLR}WOLLEN SIE ZU DEN VARIABLE
N:PRINT BEMERKUNGEN EINGEBEN?(2SPACE
); <071>
1040 GET A$:IF A$="N"GOTO 1180 <219>
1050 IF A$<>"J"GOTO 1040 <078>
1060 PRINT JA(3DOWN) <067>
1070 PRINT{HOME,6DOWN}VARIABLENNAME: ";LE
FT$(VA$(I),4) <000>
1073 PRINT LE$ <131>
1075 IF LEN(VA$(I))>70 THEN PRINT{2UP}"RI
GHT$(VA$(I),25) <235>
1080 GET A$:IF A$<>"{UP}"AND A$<>"{DOWN}"A
ND A$<>CHR$(13)GOTO 1080 <184>
1090 IF A$="{UP}"THEN I=I+1:IF I>VA THEN I
=VA <018>
1100 IF A$="{DOWN}"THEN I=I-1:IF I<1 THEN
I=1 <211>
1110 IF A$<>CHR$(13)GOTO 1070 <123>
1120 VR$=LEFT$(VA$(I),4):OPEN 1,0 <022>
1130 PRINT"TEXT: ";:INPUT#1,TE$:PRINT:CLOSE
1:TE$=LEFT$(TE$,25) <077>
1140 FOR I=VA TO 1 STEP-1 <229>
1150 IF LEFT$(VA$(I),4)<>VR$THEN NEXT:PRIN
T"NICHT VORHANDEN.":GOTO 1070 <159>
1160 VA$(I)=LEFT$(VA$(I)+LE$,55)+RIGHT$(LE
$+TE$,25):GOTO 1030 <240>
1170 VR$=LEFT$(VR$,4) <089>
1174 ; <213>
1175 REM----- <157>
1176 REM      AUSGABE VARIABLE <105>
1177 REM----- <159>
1180 PRINT:OPEN 1,DV,7 <108>
1190 PRINT#1,"{2DOWN}LISTE DER VARIABLEN{5
SPACE}:" <047>
1200 PRINT#1,"-----" <082>
1210 FOR I=VA TO 1 STEP-1 <043>
1220 IF LEFT$(VA$(I+1),4)<>LEFT$(VA$(I),4)
THEN PRINT#1,VA$(I):GOTO 1240 <169>
1230 PRINT#1,SPC(6);MID$(VA$(I),7) <218>
1240 NEXT I <167>
1250 PRINT" {DOWN,RVSON}N{RVOFF}OCHMAL{/ {RV
SON}W{RVOFF}EITER" <112>

```

```

1260 GET A$:IF A$<>"N"AND A$<>"W"GOTO 1260 <123>
1263 IF A$="N"GOTO 1190 <123>
1266 PRINT"VARIABLENLISTE OHNE ZEILENUMME
R (J/N)" <197>
1267 GET R$:IF R$=""THEN 1267 <182>
1268 IF R$<>"J"THEN PRINT:PRINT"ENDE":GOTO
1310 <230>
1269 PRINT#1:PRINT#1:PRINT#1,"VARIABLENLIS
TE OHNE ZEILENUMMER" <018>
1270 PRINT#1,"-----
----" <166>
1271 FOR I=VA TO 1 STEP-1 <104>
1272 IF LEN(VA$(I))>70 THEN PRINT#1,LEFT$(
VA$(I),8);RIGHT$(VA$(I),25):GOTO 1274 <068>
1273 PRINT#1,LEFT$(VA$(I),8) <196>
1274 NEXT I <201>
1280 GOTO 1310 <083>
1290 INPUT#2,A$,B$:IF A$="00"THEN RETURN <157>
1300 PRINT:PRINT"DISK-ERROR: ";B$ <134>
1310 CLOSE 1:CLOSE 2:END <186>

```

## Listing 2. »Cross-Referenz-Liste 64« (Schluß)

```

10 INPUT"NAME";A$ <148>
20 OPEN 1,8,2,A$+",F,R" <031>
30 GET#1,A$,B$ <054>
40 CLOSE 1 <249>
50 PRINT"ANFANGSADRESSE: ";ASC(A$+CHR$(0))
+256*ASC(B$+CHR$(0)) <089>
60 : <118>
70 REM FUER KASSETTENBETRIEB: <240>
80 REM ZEILE 20: OPEN1,1,0,A$ <161>

```

## Listing 3. Cross-Ref 64 für andere Commodore-Computer

```

290 OPEN 1,1,0,NA$:GOSUB 120:IF X=2049 THE
N 310 <190>
730 OPEN 1,1,0,NA$:GOSUB 120 <158>
999 REM DIE ZEILEN 1290 BIS 1310 ENTFALLEN <026>

```

## Listing 4. Anpassung von Cross-Ref 64 an die Datasette

**Machen Sie Ihr Hobby zum Beruf!**

Wir sind ein moderner Fachverlag mit rund 260 Mitarbeitern in Deutschland mit Niederlassungen in den USA und in der Schweiz. Unser Metier sind Fachzeitschriften und Bücher aus den Bereichen Elektronik und Computer sowie Software für Personal- und Heim-Computer. Zur Verstärkung des Redaktionsteams der Fachzeitschrift »64'er« suchen wir einen

## Commodore 64-Spezialisten als Fachredakteur

Was Sie mitbringen müssen: Erfahrungen im Umgang mit dem Commodore 64 sowie der dazugehörigen Hardware und Software. An Programmiersprachen sollten Sie zumindest Basic und Assembler beherrschen. Vertrautheit mit den gängigsten Standardprogrammen wie Textverarbeitungs-, Datenbank-, Tabellenkalkulations-, Grafik- und Kommunikationssoftware wird vorausgesetzt.

Zu Ihrem Aufgabenbereich gehört das Testen von neuer Hard- und Software, von Listings unserer Leser sowie das Schreiben von fachspezifischen Berichten. Das Aufspüren und Recherchieren von Neuigkeiten in der Branche und die Kontaktpflege zu Herstellern ist ebenso Teil Ihrer Tätigkeit wie der Besuch von Messen im In- und Ausland. Ihre wichtigste Aufgabe ist, für unsere Leser verständliche und informative Artikel zu schreiben.

Diese Stelle ist für Praktiker ebenso geeignet wie für Schulabgänger mit entsprechender Computerefahrung.

Was wir bieten, ist ein ausgezeichnetes Betriebsklima, die Arbeit in einem jungen, dynamischen Team, leistungsgerechte Bezahlung, die üblichen Sozialleistungen, eine betriebliche Altersversorgung sowie den hohen Freizeitwert des Großraums München.

Ihre schriftliche Bewerbung mit den üblichen Unterlagen richten Sie bitte an Michael Scharfenberger, der Ihnen auch für erste Kontaktgespräche telefonisch (089/46 13-122) zur Verfügung steht.

**Markt & Technik**  
Verlag Aktiengesellschaft  
Hans-Pinsel-Str. 2, 8013 Haar bei München