

```

1095 DATA 9660:REM PRUEFSUMME BLOCK 12 <067>
1100 DATA 173,32,247,183,169,251,166,20,164,21,
134,253,132,254,76,95 <019>
1110 DATA 225,76,89,225,169,0,133,253,169,199,
133,254,169,0,141,250 <244>
1120 DATA 198,141,251,198,141,252,198,141,253,
198,169,147,32,210,255,32 <197>
1130 DATA 205,192,160,3,132,251,32,165,255,133,
252,164,144,208,117,32 <073>
1140 DATA 165,255,164,144,208,110,164,251,136,
208,233,166,252,32,205,189 <250>
1145 DATA 12998:REM PRUEFSUMME BLOCK 13 <174>
1150 DATA 169,32,32,210,255,32,165,255,166,144,
208,88,170,240,9,32 <219>
1160 DATA 210,255,32,123,196,76,21,196,169,13,
32,210,255,152,160,0 <212>
1170 DATA 145,253,76,158,196,169,18,32,210,255,
32,180,197,169,0,174 <052>
1180 DATA 250,198,232,142,250,198,32,205,189,
169,146,32,210,255,169,32 <201>
1190 DATA 32,210,255,169,45,32,210,255,169,32,
32,210,255,165,253,24 <039>
1195 DATA 11628:REM PRUEFSUMME BLOCK 14 <214>
1200 DATA 105,17,144,5,164,254,200,132,254,133,
253,169,0,141,252,198 <098>
1210 DATA 160,2,208,128,32,11,197,32,66,246,96,
201,34,208,16,174 <174>
1220 DATA 253,198,240,5,162,0,76,139,196,162,1,
142,253,198,96,174 <001>
1230 DATA 253,198,240,250,172,252,198,200,140,
252,198,145,253,96,173,250 <094>
1240 DATA 198,201,20,240,3,76,53,196,169,37,162,
203,160,197,32,197 <062>
1245 DATA 11941:REM PRUEFSUMME BLOCK 15 <007>
1250 DATA 196,32,228,255,240,251,201,133,240,30,
201,134,240,71,201,135 <224>
1260 DATA 240,85,76,177,196,133,60,134,61,132,
62,160,0,177,61,32 <226>
1270 DATA 210,255,200,196,60,208,246,96,169,147,
32,210,255,169,0,141 <179>
1280 DATA 250,198,141,251,198,141,252,198,133,
253,169,199,133,254,160,0 <099>
1290 DATA 169,32,153,0,199,200,208,250,160,0,
153,0,200,200,192,90 <019>
1295 DATA 12304:REM PRUEFSUMME BLOCK 16 <053>
1300 DATA 208,248,76,53,196,32,119,196,76,174,
167,169,23,162,240,160 <239>
1310 DATA 197,32,197,196,76,168,196,169,22,162,
7,160,198,32,197,196 <224>
1320 DATA 32,119,196,169,0,133,251,133,252,133,
253,32,207,255,56,233 <225>
1330 DATA 48,133,251,32,207,255,201,38,208,4,
133,253,169,13,201,13 <123>
1340 DATA 208,7,169,0,133,252,76,98,197,56,233,
48,133,252,32,207 <069>
1345 DATA 11218:REM PRUEFSUMME BLOCK 17 <107>
1350 DATA 255,133,253,169,0,160,10,24,101,
251,136,208,250,24,101,252 <227>
1360 DATA 133,251,169,0,133,252,164,251,24,105,
17,144,5,166,252,232 <205>
1370 DATA 134,252,133,251,136,208,241,165,252,
24,105,199,133,252,177,251 <220>
1380 DATA 166,251,232,208,5,164,252,200,132,252,
164,252,32,189,255,165 <131>
1390 DATA 253,201,38,240,17,174,160,192,160,1,
32,186,255,32,87,195 <202>
1395 DATA 12680:REM PRUEFSUMME BLOCK 18 <162>
1400 DATA 32,114,195,76,174,167,174,160,192,160,
0,32,186,255,32,144 <009>
1410 DATA 195,76,174,167,173,250,198,201,9,208,
5,169,1,141,251,198 <244>
1420 DATA 173,251,198,208,5,169,48,32,210,255,
96,17,18,60,70,49 <102>
1430 DATA 62,32,87,69,73,84,69,82,32,60,70,51,
62,32,69,78 <079>
1440 DATA 68,69,32,60,70,53,62,32,76,65,68,69,
78,146,17,13 <139>
1445 DATA 8358:REM PRUEFSUMME BLOCK 19 <172>
1450 DATA 145,157,157,157,157,157,32,32,32,32,
157,157,157,157,17,32 <070>
1460 DATA 32,32,32,157,157,157,157,78,85,77,77,
69,82,32,68,69 <070>
1470 DATA 83,32,80,82,79,71,82,65,77,77,83,63,
32,13,-1 <212>
1475 DATA 4015:REM PRUEFSUMME BLOCK 20 <180>

```

Fileprotect 64

Sicher kennen auch Sie das Problem: »OPEN1,8,15,"S:xyz*"« und schon ist es passiert! Wie leicht löscht man unbeabsichtigt ein Programm oder eine Datei von der Diskette.

Das Programm Fileproject 64 sorgt dafür, daß Ihnen so etwas nicht mehr passieren kann. Mit ihm lassen sich alle Filetypen der Floppy 1541 vorm Scratching schützen. Natürlich kann der Schutz auch wieder entfernt werden.

Die Sprungadressen sind so angelegt, daß Sie beim Abtippen die REM-Zeichen nicht mit eingeben müssen.

Nach dem Starten des Programms erscheint das Titelbild mit dem Menü (Bild 1), von dem aus Sie in drei Unterprogramme verzweigen, oder das Programm beenden können. Drücken Sie hier die »F8-Taste«, so erzeugen Sie einen System-Reset (SYS 64738). Haben Sie Fileprotect 64 noch nicht abgespeichert, müssen Sie es mit der »STOP-Taste« beenden, sonst war die Arbeit des Abtippens umsonst!

Nehmen wir an, Sie wollen das Directory einer Diskette einlesen. Betätigen Sie dazu bitte die »F1-Taste«. Der Bildschirm wird gelöscht, es erscheint die Überschrift »DIRECTORY« und Sie werden aufgefordert eine Diskette einzulegen und eine Taste zu drücken. Sollten Sie versehentlich eine Taste drücken und es befindet sich noch keine Disk im Laufwerk, so wird die Fehlermeldung »DISK FEHLER« ausgegeben (Bild 2) und ein erneuter Tastendruck erwartet. Nach Auflisten der Einträge müssen Sie wieder eine Taste drücken (Bild 3), um zum Menü zurückzukehren. Hat das Directory mehr als 15 Einträge, so erscheint die Meldung »WEITERE EINTRAEGE BITTE TASTE DRUECKEN« und es wird eine neue Seite angelegt und angezeigt.

Möchten Sie ein Programm schützen, betätigen Sie die »F3-Taste«. Jetzt werden Sie erneut aufgefordert, eine Diskette einzulegen und eine Taste zu drücken. Haben Sie dies getan, prüft Fileprotect 64, ob eine Disk eingelegt ist und ob sie mit einem Schreibschutz versehen ist! Ist dies der Fall, wird die Fehlermeldung »SCHREIBSCHUTZ ENTFERNEN« ausgegeben (Bild 4). Haben Sie eine korrekte Disk eingelegt, durchsucht das Programm die Directory nach ungeschützten Files; wurde ein solcher Eintrag gefunden, so wird dessen Filetyp und Name ausgegeben. Danach erscheint »PROTECT (J/N)« (Bild 5) und Sie können entscheiden, ob Sie das File schützen wollen oder nicht. Sind keine weiteren ungeschützten Files auf der Diskette, kehrt das Programm zum Menü zurück. Drücken Sie jetzt die »F1-Taste« und lesen das Directory ein, so erkennen Sie geschützte Files an dem >-Zeichen hinter dem Filetyp. Wählen Sie erneut »Protect« an, werden Sie feststellen, daß die eben geschützten Files nicht mehr abgefragt werden! Möchten Sie den Scratching wieder entfernen, wählen Sie »F5 UNPROTECT«. Alles weitere läuft dann so ab, wie Sie es von »Protect« kennen.

Nun möchte ich noch einige Worte über die Funktionsweise von »Fileprotect 64« sagen.

Das Betriebssystem der Floppy 1541 kennt, wie Sie sicher wissen, fünf verschiedene Filetypen. Diese können geöffnet, geschlossen oder geschützt sein. Somit sind dann 15 verschiedene Zustände möglich. Der Filetyp wird in der Directory durch das erste Byte eines Fileeintrages gekennzeichnet. Wie

Sie in Tabelle 1 sehen, unterscheidet sich ein geschützter Filetyp von einem ungeschützten durch das gesetzte sechste Bit. Das Setzen des Bits erreicht man, indem man den ungeschützten Filetyp durch die »OR«-Funktion mit der Zahl 64 verknüpft. In Tabelle 2 sehen Sie an dem Beispiel »PRG«, wie die OR-Verknüpfung arbeitet. Die beiden Operanden werden Bit für Bit verglichen. Ist in einem der Faktoren ein Bit gesetzt, also 1, wird es auch im Ergebnisbyte gesetzt. Um das sechste Bit wieder auf 0 zu setzen, wählen wir die Verknüpfung »AND«. Jetzt ergibt sich eine 1 im Ergebnisbyte nur dann, wenn beide Faktoren den Wert 1 haben. Diese Eigenschaft können wir nutzen, um eine bestimmte Bitstelle auf 0 zu setzen. Wir verknüpfen das geschützte Byte mit dem Faktor 191 (dual 1011 1111) und erhalten wieder den ungeschützten Filetyp 130 (Tabelle 3).

Genug der grauen Theorie! Wie kann man diese Änderungen nun auf der Diskette vornehmen? Dazu müssen wir den Filetyp (1 Byte) von der Disk lesen, ihn ändern und dann wieder zurückschreiben. Das Betriebssystem der 1541 sieht hierzu die Befehle »U1« (Block lesen) und »U2« (Block schreiben) vor. Mit diesen Befehlen können Sie aber nur ganze Blöcke, also 256 Byte in den Puffer einer Direktzugriffsdatei einlesen oder aus ihm auf die Diskette schreiben. Da wir nur ein bestimmtes Byte benötigen, nutzen wir die Möglichkeit, den Zeiger, der die augenblickliche Lese- oder Schreibposition im Puffer angibt, auf einzelne Bytes setzen zu können. Der Befehl hierfür lautet »B-P« (Buffer Pointer). Alle Befehle, die für die Floppy bestimmt sind, werden über den Befehlskanal gesendet, der im »OPEN«-Befehl (Zeile 640) durch die Sekundäradresse 15 angesprochen wird. Mit dem PRINT #-Befehl werden jetzt die Kommandos übermittelt. Zum Beispiel lesen eines Sektors von Spur 18 (Zeile 2160). Daten können nicht direkt von der Diskette in den Computer eingelesen werden, sondern müssen in einem Datenpuffer zwischengespeichert werden. Dazu eröffnen wir eine Direktzugriffsdatei (Zeile 650), aus der wir dann mit dem GET #-Befehl die Bytes einzeln herauslesen.

Im Bild 7 sehen Sie ein kleines Beispiel für die Anwendung der genannten Befehle. Mit diesem Unterprogramm können Sie den Namen einer Diskette einlesen.

Zeile	Kommentar
100-350	Titel des PGMs, Adresse des Autors
360-450	Haupt-Programm, von hier aus werden die einzelnen Unterprogramme angesprochen. Aus welchen Zeilen die Unterprogramme angesprochen werden, entnehmen Sie bitte Tabelle 4.
460-530	Setzen der Bildschirmfarben und festlegen der Konstanten »BLS«
540-600	Unter-PGM warten auf Tastendruck und ein Zeichen aus dem Tastaturpuffer holen.
610-660	Unter-PGM: Befehlskanal und Direktzugriffsdatei auf der Floppy öffnen.
670-730	Unter-PGM. Zeichen aus Floppy-Puffer einlesen und sicherstellen, daß kein Leerstring weitergegeben wird.
740-960	Ausgabe des Titelbildes und Menüs
970-1250	In diesem Unterprogramm wird geprüft, ob eine Disk eingelegt ist oder ob bei Protect und Unprotect ein Schreibschutz auf der Disk vorhanden ist. Außerdem wird wenn kein Fehler auftrat, bei Protect und Unprotect eine Tabellenüberschrift ausgegeben.
1260-1430	Unter-PGM. Filetyp einlesen und auswerten.
1440-2040	Unter-PGM Directory einlesen.
2050-2510	Unter-PGM: Protect oder Unprotect. Ist im Haupt-PGM P=1 dann wird die Routine Protect, ist P=0 die Routine Unprotect durchgeführt

Die Funktionen der einzelnen Variablen entnehmen Sie bitte Tabelle 5. Eine wichtige Anmerkung: Damit Fileprotect 64 einwandfrei arbeitet, muß in Zeile 2499 hinter dem »(FT)« unbe-

dingt ein Semikolon (;) gesetzt sein! Ist dies nicht der Fall, können Sie geschützte Programme nicht mehr laden. Ohne Semikolon wird ein CHR\$(13) an den Wert angehängt. Es würden also zwei Byte und damit die Startadresse des Programms überschrieben!

(Jochen Fette / rg)

ABB. : 1

FILEPROTECT 64

1984 BY JOCHEN FETTE

DIRECTORY
 PROTECT
 UNPROTECT
 END

IHRE WAHL ?

ABB. : 2

DISK EINLEGEN UND TASTE DRUECKEN

DISK FEHLER !

ABB. : 3

DISK EINLEGEN UND TASTE DRUECKEN . . .

DISK EINLEGEN UND TASTE DRUECKEN .

```

2 TEST 1 PRG
1 DISK ERROR PRG
5 DRUCK MASKE2 PRG
1 DRUCK$ MASKE PRG
5 DRUCK TEST1 PRG
3 GUNIS KARTEI-KAR PRG
115 ↑ ART. SEQ
9 TEST 1 PRG
33 3-D 1G PRG
33 3-D 2G PRG<
4 TEST PRG<
12 FILEPROTECT 64 PRG
3 BALKEN PRG
    
```

438 FREIE BLOECKE

TASTE DRUECKEN . . .

ABB. : 4

DISK EINLEGEN UND TASTE DRUECKEN .

DISK EINLEGEN UND TASTE DRUECKEN .

SCHREIBSCHUTZ ENTFERNEN !

ABB. : 5

DISK EINLEGEN UND TASTE DRUECKEN . . .

DISK EINLEGEN UND TASTE DRUECKEN . . .

TYP NAME

```

PRG TEST 1 PROTECT (J/N)N
PRG DISK ERROR PROTECT (J/N)N
PRG DRUCK MASKE2 PROTECT (J/N)J
PRG DRUCK$ MASKE PROTECT (J/N)J
PRG DRUCK TEST1 PROTECT (J/N)N
PRG GUNIS KARTEI-KAR PROTECT (J/N)N
SEQ ↑ ART. PROTECT (J/N)N
PRG TEST 1 PROTECT (J/N)J
PRG 3-D 1G PROTECT (J/N)J
PRG FILEPROTECT 64 PROTECT (J/N)J
PRG BALKEN PROTECT (J/N)N
    
```

ABB. : 6

DISK EINLEGEN UND TASTE DRUECKEN . . .

DISK EINLEGEN UND TASTE DRUECKEN . . .

TYP NAME

```

PRG< DRUCK MASKE2 UNPROTECT (J/N)N
PRG< DRUCK$ MASKE UNPROTECT (J/N)N
PRG< TEST 1 UNPROTECT (J/N)J
PRG< 3-D 1G UNPROTECT (J/N)J
PRG< 3-D 2G UNPROTECT (J/N)J
PRG< TEST UNPROTECT (J/N)J
PRG< FILEPROTECT 64 UNPROTECT (J/N)J
    
```

ABB. : 7

```

100 OPEN15,8,15,"I0"
110 OPEN2,8,2,"#"
120 PRINT#15,"B-R";2;0;18;0
130 PRINT#15,"B-P";2;144
140 N$=""
150 FOR I=0 TO 15
160 GET#2,X$:IF X$=CHR$(160) THEN I=15:GOTO 180
170 N$=N$+X$
180 NEXT I
190 PRINT "DISK NAME: ";N$
200 CLOSE 2
210 CLOSE 15
220 END
    
```

READY.

TABELLE 1

Dual	Dez	Hex	Filetyp
0000	0000	..0	\$00 *DEL
0000	0001	..1	\$01 *SEQ
0000	0010	..2	\$02 *PRG
0000	0011	..3	\$03 *USR
0000	0100	..4	\$04 *REL
1000	0000	128	\$80 DEL
1000	0001	129	\$81 SEQ
1000	0010	130	\$82 PRG
1000	0011	131	\$83 USR
1000	0100	132	\$84 REL
1100	0000	192	\$C0 DEL<
1100	0001	193	\$C1 SEQ<
1100	0010	194	\$C2 PRG<
1100	0011	195	\$C3 USR<
1100	0100	196	\$C4 REL<

TABELLE 2

Die 'OR - Verknuepfung'

Dual	Dez	Typ
1000	0010	130 PRG
0100	0000	.64 Verknuepfungsfaktor
1100	0010	194 PRG<

TABELLE 3

Die AND - Verknuepfung

Dual	Dez	Typ
1100 0010	194	PRG
1011 1111	191	Verknuepfungsfaktor
1000 0010	130	PRG

TABELLE 5

A#	->	Get von Tastatur
BB	->	Belegte Blocks
BL#	->	Formatstring fuer Bildschirmausgaben
FB	->	Freie Blocks
FT#	->	Filetyp als Wort
FT	->	Filetyp als Zahl
HB	->	High Byte der belegten Blocks
I	->	Laufvariable
LB	->	Low Byte der belegten Blocks
N#	->	Name des aktuellen Directoryeintrages
P	->	Schalter fuer Protect oder UnProtect
RM	->	Rueckmeldung von Unterprogrammen
S	->	Aktueller Sektor
SA	->	Alter Sektor
T	->	Track
UE	->	Schalter fuer Ueberschrift ja/nein
X#	->	Get von Diskette
Y	->	Laufvariable
ZE	->	Zeilenzaehler bei Unter-PGM Directory

TABELLE 4

SPRUNGTABELLE

Zeile	Wird angesprungen von Zeile				
370	440*				
380	400?				
490	360@				
570	380@	1020@	1890@	2030@	2420@
640	1070@	1530@	2140@		
700	1290@	1590@	1610@	1680@	1700@
	1780@	2170@	2190@	2330@	2360@
770	370@				
1000	1510@	2120@			
1140	1060?				
1180	1140?				
1250	1180?	1210?*			
1290	1650@	2240@			
1470	430@	1520?			
1570	1940?				
1810	1790?*				
1930	1660?	1840?			
2080	430@	2130?			
2160	2290?				
2330	2250?@	2260?@			
2420	2440?				
2510	2430?*				

* = GOTO
 @ = GOSUB
 ? = IF THEN
 ?* = IF THEN GOTO
 ?@ = IF THEN GOSUB

100 REM	-----	<110>
110 :		<168>
120 REM FILE PROTECT 6 4		<178>
130 :		<188>
140 REM SCRATCH-SCHUTZ VON FILES		<151>
150 :		<208>
160 REM FUER		<097>
170 :		<228>
180 REM C-64 UND 1541		<207>
190 :		<248>
200 REM		<025>
210 :		<012>
220 REM JOCHEN FETTE		<154>
230 :		<032>
240 REM GASSELSTIEGE 14		<100>
250 :		<052>
260 REM 4400 MUENSTER		<207>

```

270 : <073>
280 : <083>
290 REM ----- <045>
300 : <103>
310 : <113>
320 : <123>
330 : <133>
340 : <143>
350 : <153>
360 GOSUB 490:REM INITIALISIEREN <117>
370 GOSUB 770:REM TITELBILD <004>
380 GOSUB 570:REM WARTEN AUF TASTE <157>
390 IF ASC(A$)=140 THEN SYS 64738:REM ENDE? -> RESET!
                                     <089>
400 IF ASC(A$)>135 OR ASC(A$)<133 THEN 380:REM UNGUELTIGE TASTE
                                     <221>
410 IF ASC(A$)=134 THEN P=1:REM PROTECT <176>
420 IF ASC(A$)=135 THEN P=0:REM UNPROTECT <093>
430 ON ASC(A$)-132 GOSUB 1470,2080,2080 <066>
440 GOTO 370 <220>
450 END <067>
460 REM ----- <170>
470 REM INITIALISIEREN <127>
480 REM ----- <190>
490 BL$="[SPACE20]":REM 20 - SPACE <087>
500 POKE 53280,9:REM FARBE FUER RAND AUF 'BRAUN' <021>
510 POKE 53281,9:REM FARBE FUER HINTERGRUND AUF 'BRAUN' <069>
520 PRINT CHR$(5);:REM FARBE FUER SCHRIFT AUF 'WEISS' <078>
530 RETURN <162>
540 REM ----- <251>
550 REM WARTEN AUF BELIEBIGE TASTE UND EIN ZEICHEN VON TASTATUR HOLEN <002>
560 REM ----- <015>
570 A$="" <151>
580 POKE 198,0:WAIT 198,255:REM WARTEN AUF BELIEBIGE TASTE <128>
590 GET A$ <086>
600 RETURN <232>
610 REM ----- <065>
620 REM DISK OEFFNEN <041>
630 REM ----- <085>
640 OPEN 15,8,15,"I0":REM BEFEHLSKANAL 15 ,UND 'I0' FUER INITIALISIEREN <026>
650 OPEN 2,8,2,"#":REM DATEN PUFFER 2 ,UND '#' FUER DIRECKTZUGRIFF <186>
660 RETURN <036>
670 REM ----- <125>
680 REM 1 ZEICHEN AUS DISK LESEN <251>
690 REM ----- <145>
700 X$="" <048>
710 GET#2,X$:REM 1 ZEICHEN LESEN <221>
720 IF X$="" THEN X$=CHR$(0):REM SONST 'ILLEGAL QUANTITY ERROR' <157>
730 RETURN <106>
740 REM ----- <195>
750 REM TITELBILD AUSGEBEN <102>
760 REM ----- <215>
770 PRINT CHR$(147);:REM 'CLR' <133>
780 PRINT SPC(13);"FILEPROTECT[SPACE]64" <005>
790 PRINT SPC(12);"EEEEEEEEEEEEEEEE":REM 'SCHIFT + E' <251>
800 PRINT <156>
810 PRINT SPC(6);CHR$(18);"SCRATCH-SCHUTZ[SPACE]VON[SPACE]FILES[SPACE]FUER";CHR$(146)
:REM INVERS <066>
820 PRINT SPC(13);CHR$(18);"C-64[SPACE]UND[SPACE2]1541";CHR$(146)
:REM INVERS <088>
830 PRINT <218>
840 PRINT CHR$(144);"[SPACE8](C)[SPACE]1984[SPACE2]BY[SPACE]JOCHEN[SPACE]FETTE";CHR$(5)
:REM SCHWARZ <187>

```

```

850 PRINT:PRINT:PRINT:PRINT <103>
860 PRINT SPC(10);CHR$(18);"[SPACE]F1[SPACE]";CHR$(146);"[SPACE3]DIRECTORY" <252>
870 PRINT <002>
880 PRINT SPC(10);CHR$(18);"[SPACE]F3[SPACE]";CHR$(146);"[SPACE3]PROTECT" <126>
890 PRINT <022>
900 PRINT SPC(10);CHR$(18);"[SPACE]F5[SPACE]";CHR$(146);"[SPACE3]UNPROTECT" <055>
910 PRINT <042>
920 PRINT SPC(10);CHR$(18);"[SPACE]F8[SPACE]";CHR$(146);"[SPACE3]END" <097>
930 PRINT <062>
940 PRINT <072>
950 PRINT SPC(10);"[SPACE7]IHRE[SPACE]WAHL[SPACE]?";:REM 7 - SPACE <104>
960 RETURN <081>
970 REM ----- <170>
980 REM DISK EINGELEGT ? KEIN SCHREIBSCHUTZ ? <171>
990 REM ----- <190>
1000 PRINT <132>
1010 PRINT"DISK[SPACE]EINLEGEN[SPACE]UND[SPACE]TASTE[SPACE]DRUECKEN[SPACE].[SPACE]. <135>
      [SPACE]. "
1020 GOSUB 570:REM WARTEN AUF TASTE <031>
1030 OPEN 15,8,15,"I0":REM OEFFNEN DES BEFEHLSKANALS UND INITIALISIEREN <071>
1040 INPUT#15,RM:REM FEHLER CODE <099>
1050 CLOSE 15 <036>
1060 IF RM<>0 THEN 1140:REM KEINE DISK <179>
1070 GOSUB 640:REM DISK OEFFNEN <078>
1080 PRINT#15,"B-R";2;0;18;0:REM LESEN EINES BLOCKS <196>
1090 PRINT#15,"B-W";2;0;18;0:REM SCHREIBEN DES BLOCKS <087>
1100 INPUT#15,RM:REM FEHLER CODE <159>
1110 CLOSE 15:REM BEFEHLSKANAL SCHLIESSEN <122>
1120 CLOSE 2:REM DATENPUFFER SCHLIESSEN <036>
1130 PRINT <007>
1140 IF RM<>0 OR UE=0 THEN 1180:REM FEHLER,AUFRUF VON DIRECTORY -> KEINE UEBER. <103>
1150 PRINT"[SPACE]TYP[SPACE2]NAME" <125>
1160 PRINT"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE":REM 39 - SCHIFT 'E' <034>
1170 UE=0 <018>
1180 IF RM=0 THEN 1250:REM KEIN FEHLER -> RETURN <012>
1190 PRINT <067>
1200 IF RM<>26 THEN PRINT CHR$(144);"DISK[SPACE]FEHLER[SPACE]!";CHR$(5) <231>
      :REM SCHWARZE SCHRIFT
1210 IF UE=0 AND RM=26 THEN RM=0:GOTO 1250: <026>
1220 IF RM=26 THEN PRINT CHR$(144);"SCHREIBSCHUTZ[SPACE]ENTFERNEN[SPACE]!";CHR$(5): <229>
1230 FOR W=1 TO 1500:REM WARTESCHLEIFE
      <134>

1240 NEXT W <181>
1250 RETURN <116>
1260 REM ----- <205>
1270 REM FILETYP HOLEN <028>
1280 REM ----- <226>
1290 GOSUB 700:REM 1 ZEICHEN AUS PUFFER <228>
1300 FT=0:FT$="" <131>
1310 IF ASC(X$)=1 THEN FT=1:FT$="*SEQ[SPACE]" <073>
1320 IF ASC(X$)=2 THEN FT=2:FT$="*PRG[SPACE]" <085>
1330 IF ASC(X$)=3 THEN FT=3:FT$="*USR[SPACE]" <114>
1340 IF ASC(X$)=4 THEN FT=4:FT$="*REL[SPACE]" <103>
1350 IF ASC(X$)=129 THEN FT=129:FT$="[SPACE]SEQ[SPACE]" <029>
1360 IF ASC(X$)=130 THEN FT=130:FT$="[SPACE]PRG[SPACE]" <023>
1370 IF ASC(X$)=131 THEN FT=131:FT$="[SPACE]USR[SPACE]" <052>
1380 IF ASC(X$)=132 THEN FT=132:FT$="[SPACE]REL[SPACE]" <041>
1390 IF ASC(X$)=193 THEN FT=193:FT$="[SPACE]SEQ<" <131>
1400 IF ASC(X$)=194 THEN FT=194:FT$="[SPACE]PRG<" <143>
1410 IF ASC(X$)=195 THEN FT=195:FT$="[SPACE]USR<" <172>

```

```

1420 IF ASC(X$)=196 THEN FT=196:FT$="[SPACE]REL<" <161>
1430 RETURN <041>
1440 REM ----- <130>
1450 REM DIRECTORY EINLESEN <070>
1460 REM ----- <150>
1470 PRINT CHR$(147):REM 'CLR' <008>
1480 PRINT SPC(6);CHR$(18);"D[SPACE]I[SPACE]R[SPACE]E[SPACE]C[SPACE]T[SPACE]O[SPACE]R
[SPACE]Y";CHR$(146):REM REVERS <233>
1490 PRINT <112>
1500 UE=0:REM KEINE UEBERSCHRIFT <024>
1510 GOSUB 1000:REM DISK EINGELEGT? <000>
1520 IF RM<>0 THEN 1470:REM NEUER VERSUCH <142>
1530 GOSUB 640:REM DISK DEFFNEN <027>
1540 T=18:S=1:REM TRACK UND SEKTOR <230>
1550 BB=0:FB=664:REM BELEGTE UND UNBELEGTE BLOECKE <198>
1560 ZE=0:REM ZAEHLER FUER BILDSCHIRMZEILEN <146>
1570 PRINT#15,"B-R";2;0;T;S:REM 1. SEKTOR LESEN <195>
1580 PRINT#15,"B-P";2;0:REM POINTER AUF NAECHSTEN TRACK <011>
1590 GOSUB 700:REM ZEICHEN LESEN <166>
1600 T=ASC(X$):REM NAECHSTER TRACK <186>
1610 GOSUB 700 <116>
1620 S=ASC(X$):REM NAECHSTER SEKTOR <048>
1630 FOR I=0 TO 7:REM 8 DIREKTORY EINTRAEGE PRO SEKTOR
<073>
1640 PRINT#15,"B-P";2;I*32+2:REM POINTER AUF FILETYP <092>
1650 GOSUB 1290:REM FILETYP HOLEN <045>
1660 IF FT=0 THEN 1930:REM NAECHSTER DIRECTORY EINTRAG
<034>
1670 PRINT#15,"B-P";2;I*32+30:REM POINTER AUF ANZAHL DER BLOCKS <229>
1680 GOSUB 700 <186>
1690 LB=ASC(X$):REM LOW BYTE DEZIMAL <104>
1700 GOSUB 700 <206>
1710 HB=ASC(X$)*256:REM HIGH BYTE DEZIMAL <239>
1720 BB=HB+LB:REM BELEGTE BLOCKS <053>
1730 PRINT STR$(BB)+LEFT$(BL$,5-LEN(STR$(BB)));:REM AUSGABE DER BLOCKS <189>
1740 FB=FB-BB:REM FREIE BLOCKS <181>
1750 PRINT#15,"B-P";2;I*32+5:REM POINTER AUF FILE NAME <241>
1760 N$="":REM ALTEN FILENAMEN LOESCHEN <107>
1770 FOR Y=0 TO 15:REM MAXIMAL 16 ZEICHEN
<245>
1780 GOSUB 700 <030>
1790 IF ASC(X$)=160 THEN Y=15:GOTO 1810:REM SCHIFT SPACE NICHT UEBERNEHMEN
<170>
1800 N$=N$+X$ <203>
1810 NEXT Y <244>
1820 PRINT N$+LEFT$(BL$,20-LEN(N$));FT$:REM FILENAMEN UND FILETYP <183>
1830 ZE=ZE+1:REM ZEILENZAEHLER <147>

1840 IF ZE<15 THEN 1930:REM SEITE NOCH NICHT VOLL
<012>
1850 ZE=0 <194>
1860 PRINT <228>
1870 PRINT"WEITERE[SPACE]EINTRAEGE" <219>
1880 PRINT"BITTE[SPACE]TASTE[SPACE]DRUECKEN[SPACE].[SPACE].[SPACE]."; <075>
1890 GOSUB 570:REM WARTEN AUF TASTE <137>
1900 PRINT CHR$(147):REM 'CLR' <184>
1910 PRINT SPC(6);CHR$(18);"D[SPACE]I[SPACE]R[SPACE]E[SPACE]C[SPACE]T[SPACE]O[SPACE]R
[SPACE]Y";CHR$(146):REM INVERS <153>
1920 PRINT:PRINT:PRINT:PRINT <153>
1930 NEXT I:REM NAECHSTER DIRECTORY EINTRAG <129>
1940 IF T<>0 THEN 1570:REM WEITERE EINTRAEGE ?
<051>
1950 CLOSE 2:REM PUFFER SCHLIESSEN <249>

```

```

1960 GLOSE 15:REM                                BEFEHLSKANAL SCHLIESSEN <207>
1970 PRINT                                        <082>
1980 PRINT FB;:REM                                FREIE BLOECKE <072>
1990 IF FB<>1 THEN PRINT"[SPACE3]FREIE[SPACE]BLOECKE" <089>
2000 IF FB=1 THEN PRINT"[SPACE3]FREIER[SPACE]BLOCK" <121>
2010 PRINT                                        <122>
2020 PRINT"TASTE[SPACE]DRUECKEN[SPACE].[SPACE].[SPACE]."; <095>
2030 GOSUB 570:REM                                WARTEN AUF TASTE <021>
2040 RETURN                                       <141>
2050 REM ----- <231>
2060 REM PROTECT ODER UNPROTECT <178>
2070 REM ----- <251>
2080 PRINT CHR$(147):REM                                'CLR' <077>
2090 IF P=1 THEN PRINT SPC(13);CHR$(18);"P[SPACE]R[SPACE]O[SPACE]T[SPACE]E[SPACE]C
[SPACE]T";CHR$(146):REM                                INVERS <077>
2100 IF P=0 THEN PRINT SPC(12);CHR$(18);"U[SPACE]N[SPACE]P[SPACE]R[SPACE]O[SPACE]T
[SPACE]E[SPACE]C[SPACE]T";CHR$(146) <088>
2110 UE=1:REM                                UEBERSCHRIFT <018>
2120 GOSUB 1000:REM                                DISK EINGELEGT? <101>
2130 IF RM<>0 THEN 2080:REM                                WENN FEHLER DANN NEUER VERSUCH
<000>
2140 GOSUB 640:REM                                DISK OEFFNEN <128>
2150 T=18:S=1:REM                                TRACK UND SEKTOR <074>
2160 PRINT#15,"U1[SPACE]2[SPACE]0[SPACE]18",S:REM        SEKTOR 'S' VON TRACK 18 IN PUFFE
R '2' LESEN <031>
2170 GOSUB 700:REM                                ZEICHEN LESEN <236>
2180 T=ASC(X$):REM                                NAECHSTER TRACK <000>
2190 GOSUB 700 <186>
2200 SA=S:REM                                AKTUELLEN SEKTOR MERKEN <065>
2210 S=ASC(X$):REM                                NAECHSTER SEKTOR <128>
2220 FOR I=2 TO 226 STEP 32 <174>
2230 PRINT#15,"B-P";2;I:REM                                POINTER AUF FILETYP <191>
2240 GOSUB 1290:REM                                FILETYPHOLEN <125>
2250 IF FT>128 AND FT<133 AND P=1 THEN GOSUB 2330:REM NUR UNGESCHUETZTE FILES FUER PROT.
<120>
2260 IF FT>192 AND FT<197 AND P=0 THEN GOSUB 2330:REM NUR GESCHUETZTE FILES FUER UNPROT.
<140>
2270 NEXT I:REM                                NAECHSTER DIRECTORY EINTRAG <214>
2280 IF RM=1 THEN RM=0:PRINT#15,"U2[SPACE]2[SPACE]0[SPACE]18";SA
:REM                                KORRIGIRTEN BLOCK SCHREIBEN <005>
2290 IF T<>0 THEN 2160:REM                                WEITERE EINTRAEGE
<079>
2300 CLOSE 2:REM                                DATENPUFFER SCHLIESSEN <196>
2310 CLOSE 15:REM                                BEFEHLSKANAL SCHLIESSEN <047>
2320 RETURN:REM                                DIRECTORY ENDE <065>
2330 GOSUB 700:GOSUB 700:REM                                2BYTE UEBERSPRINGEN <173>
2340 N$="":REM                                ALTEN NAMEN LOESCHEN <146>
2350 FOR Y=0 TO 15:REM                                HOECHSTENS 16 ZEICHEN
<039>
2360 GOSUB 700 <101>
2370 IF ASC(X$)=0 THEN N$=N$+"[SPACE]":REM                                IMMER AUFFUELLEN AUF 16 ZEICH
EN <233>
2380 IF ASC(X$)<>0 THEN N$=N$+X$ <106>
2390 NEXT Y <058>
2400 IF P=1 THEN PRINT FT$+"[SPACE]"+N$+"[SPACE3]PROTECT[SPACE](J/N)";
:REM                                FILETYP UND NAME <127>
2410 IF P=0 THEN PRINT FT$+"[SPACE]"+N$+"[SPACE]UNPROTECT[SPACE](J/N)";
:REM                                FILETYP UND NAME <043>
2420 GOSUB 570:REM                                WARTEN AUF TASTE <157>
2430 IF A$="N"THEN PRINT"N":GOTO 2510:REM                                RETURN
<193>
2440 IF A$<>"J"THEN 2420:REM                                UNGUELTIGE TASTE <037>
2450 PRINT"J" <194>

```

```

2460 PRINT#15,"B-P";2;I:REM
2470 IF P=1 THEN FT=(64 OR FT):REM
2480 IF P=0 THEN FT=(255 AND FT-64):REM
2490 PRINT#2,CHR$(FT);:REM
2500 RM=1:REM
2510 RETURN

```

```

PUFFER POINTER AUF FILETYP <110>
FILETYP SCHUETZEN <004>
SCHUTZ ZURUECKNEHMEN <990>
GEAENDERTER FILETYP <178>
VERAENDERUNGEN DURCHGEFUEHRT <001>
<101>

```

P-Basic-V2: Autostart mit Rück- wärtsgang

Dieser Autostarter erzeugt nicht nur ein einfaches Lade-Programm für Basic und Assembler-Programme. Es wird zusätzlich die RUN/STOP-Taste gesperrt und ein Kopierschutz installiert. Um ein so behandeltes Programm wieder verändern zu können, kann mit einem Befehl der gesamte Vorgang rückgängig gemacht werden.

Das Programm »P-Basic-V2« läuft auf dem Commodore 64 (ohne Basic-Erweiterung) und der Floppy 1541 (nicht auf der Datasette, da das Programm beim Laden den Kassettenpuffer benutzt). Es ist zirka 260 Bytes lang und liegt im Speicher ab 49152 (\$c000).

Nach dem normalen Laden und Starten des Basic-Programmes, ist das Basic-V2 um drei neue Befehle erweitert:

1. !SAVE"Name", Gerätenummer,1
2. !LOAD"Name", Gerätenummer,1
3. !NEW

Der Hauptbefehl ist !SAVE. Mit ihm kann man ein Basic-Programm oder ein Maschinenprogramm (mit Basic-Startzeile) so abspeichern, daß es nach dem Laden von allein startet. Außerdem wird die RUN/STOP/RESTORE-Taste abgestellt. Das Programm ist somit vor Änderungen gesichert. Auch das Kopieren ist nur mit besonderen Kopierprogrammen möglich. Doch durch eine Codewortabfrage in den ersten Zeilen wird es wohl keiner schaffen, das Programm zum Laufen zu bringen. Ein Programm, das mit dieser Basic-Hilfe abgespeichert wurde, muß mit dem Zusatz »,1« absolut, nicht ab Basic-Start (2049), geladen werden. Lädt man ein solches Programm ohne diesen kleinen Zusatz in den Speicher, so läßt es sich weder listen, noch starten. Weil ein normales Programm mit der Zwei-

Byte-Startadresse am Anfang der Datei gespeichert wird, ist es durch ein nochmaliges anderes Abspeichern ganz verloren.

Man kann diese Routine natürlich auch nur zur Bequemlichkeit benutzen. Um die Stopp-Routine wieder zu aktivieren, hilft: POKE808,237 als erster Befehl des Programms.

Ist in dem geschützten Programm nun aber ein Fehler, oder will man es erweitern oder verbessern, so hilft der zweite Befehl. Mit !LOAD kann man alle Programme, die mit Autostart geschützt sind, knacken und nach eigener Lust bearbeiten.

Tritt bei den Arbeiten mit diesen Befehlen ein Fehler auf, den Sie oder das Betriebssystem (Floppy/Computer) verursacht haben, so können sie mit Eingabe von !NEW alle wichtigen Vektoren korrigieren und somit das gerade bearbeitete Programm wieder listen lassen.

Erläuterungen zum Assemblerlisting

In dem Programmabschnitt »Vektorsetzung« wird die Routine zur Erkennung der Befehls-Token (Basic-Befehle) auf die erweiterte Routine (Erkennung) umgelenkt und danach der Name der Basic-Erweiterung ausgegeben.

In der Erkennungsroutine werden die Befehls-Token mit dem neuen Befehl (!) und danach mit den eigentlichen Befehlen (SAVE, LOAD, NEW) verglichen. »SAVE« hat das Basic-Token 148, »LOAD« hat 147 und »NEW« hat 162. Wird keiner dieser Befehle interpretiert, so wird Syntax Error (Jmp \$af08) angezeigt. Die DATAs werden in der Vektorsetzungsroutine ausgegeben.

Der Befehl »LOAD«, der wie der Befehl »SAVE« nur im Direktmodus verwendet werden darf, setzt zuerst das Programmmodusflag, damit es keine Ausgabe wie »searching for ...« und so weiter gibt. Danach werden die LOADparameter geholt (\$e1d4) und nach LOAD verzweigt (\$fd5). Nachdem das Programm geladen ist, werden alle Vektoren mit der NEW-Routine wieder zurückgesetzt und dann zur END-Routine verzweigt (\$a831).

Der Befehl »SAVE« gleicht am Anfang (bis Ende Zeile 166) dem LOAD-Befehl. In Zeile 168 wird die Steuerroutine in den Kassettenpuffer (Sprite 11) geladen. Dann werden die »Input«- und »Output«-Vektoren auf die Steuerroutine gesetzt und das Programm ab Byte 806 gespeichert. Die Vektoren werden dann mit der NEW-Routine normalisiert.

Die Steuerroutine arbeitet ungefähr so wie die NEW-Routine. Nur bei dieser Routine werden zusätzlich die RUN-Zeiger (\$a659) gesetzt, die Run-Stop/Restore-Funktion abgeschaltet und dann zur Interpreterschleife verzweigt.

Bemerkungen zum Programm

Das Befehlszeichen »!« kann natürlich durch fast jedes Zeichen ersetzt werden, nur wenn das Zeichen ein Token ist, muß nicht der ASCII-Wert, sondern der Tokenwert angegeben werden, da sonst der Befehl nicht erkannt wird. Wenn das Zeichen ein Buchstabe ist, kann derselben Variablen kein Wert zugewiesen werden. Denn auch dabei entsteht ein Fehler.

(Jan Kusch/gk)