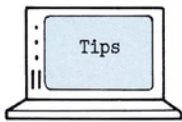


Tips & Tricks für Einsteiger



Diesmal beschäftigen wir uns mit dem kürzesten Sprite-Editor, einem Software-Schreibschutz für Disketten, einer Eingabehilfe für DATA-Zeilen und einem Programm, das beliebige Texte in Abenteuerspielen aufspürt.

Es ist Ihnen nicht auch schon einmal passiert: Sie haben in mühevoller Kleinarbeit eine Diskette mit allen wichtigen Programmen zusammengestellt, und wie es der Zufall will, plötzlich befinden sich durch falsche Eingabe Files auf der Diskette, die nicht dort hingehören oder die Diskette ist neu formatiert. Gut, Files lassen sich löschen. Mit einer neu formatierten Diskette sieht das schon anders aus. Die auf ihr gespeicherten Programme sind verloren.

Der mechanische Schreibschutz (Aufkleber) ist auch nicht der Weisheit letzter Schluß, um solchen Zufällen zu begegnen. Wenn häufiger Daten auf der Diskette geändert werden sollen, muß der Schreibschutz jedesmal entfernt und nach erfolgreicher Änderung wieder angebracht werden. Um diesen mechanischen Schreibschutz zu umgehen, kann eine Diskette auch softwaremäßig geschützt werden (Listing 1).

```

10 OPEN 15,8,15,"I":OPEN 2,8,2,"#" <076>
20 PRINT#15,"U1 2 0 18 0" <248>
30 PRINT#15,"B-P 2 2":GET#2,FO$ <113>
50 PRINT"CLR,2DOWN,3SPACE"ALTES FORMATZEI
CHEN = ";FO$ <189>
55 IF FO$="A"THEN X$="X" <031>
56 IF FO$<>"A"THEN X$="A" <136>
60 PRINT"DOWN,3SPACE"NEUES FORMATZEICHEN
= ";X$;"LEFT"; <146>
70 POKE 198,0:WAIT 198,1 <042>
80 A$=X$:GET X$:IF X$=CHR$(13)THEN X$=A$:G
OTO 90 <043>
85 PRINT X$ <191>
90 IF FO$="A"THEN 200 <172>
100 PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)CHR
$(65) <189>
200 PRINT#15,"B-P 2 2":PRINT#2,X$; <071>
220 PRINT#15,"U2 2 0 18 0":PRINT <119>
230 IF X$="A"THEN PRINT"2DOWN,3SPACE,RVSD
N)SCHREIBSCHUTZ ENTFERNT" <197>
240 IF X$<>"A"THEN PRINT"2DOWN,3SPACE,RVS
D)SCHREIBSCHUTZ AKTIVIERT" <229>
250 PRINT#15,"I":CLOSE 2:CLOSE 15 <047>

```

Listing 1. Schreibschutz. Eingabehinweise auf Seite 99.

Wie das funktioniert, soll kurz erklärt werden.

Auf der Diskette ist im 2. Byte auf Spur 18 Sektor 0 das Formatkennzeichen der 1541 gespeichert und zwar die Zahl \$41. Wird in dieses Byte eine andere Zahl geschrieben (mit einem Disketten-Monitor), so funktioniert das Beschreiben einer Diskette nur so lange, bis die Floppy-Station neu initialisiert wird (zum Beispiel aus- und wieder einschalten). Wird jetzt versucht, ein Programm oder Daten zu speichern, meldet die Floppy-Station einen Formatfehler und nichts geht mehr. Dieser Software-Schutz funktioniert auch dann, wenn versucht wird, die Diskette mit

```
OPEN 15,8,15,"N:TEST"
```

zu formatieren. Das heißt, formatiert wird die Diskette mit die-

ser Befehlssequenz eigentlich nicht. Es wird nur das Inhaltsverzeichnis und ein Diskettenbereich (BAM) gelöscht, der Informationen darüber enthält, welche Diskettenblöcke belegt sind und welche nicht. Dieser Vorgang nennt sich auch »weich formatieren«.

Soll eine softwaremäßig geschützte Diskette »entschärft« werden, taucht ein Problem auf. Es läßt sich ja absolut nichts mehr speichern beziehungsweise ändern. Man muß also das Betriebssystem der Floppy-Station davon überzeugen, daß das richtige Formatkennzeichen an der richtigen Stelle steht. Dazu existiert im Speicher der Floppy-Station eine Speicherzelle, in der nach jeder Initialisierung das aktuelle Formatkennzeichen gespeichert wird. Es handelt sich um die Speicherzelle \$101. Wird in diese Speicherzelle mit dem Befehl

```
PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)CHR$(65)
```

(Zeile 100) wieder das 1541 Formatkennzeichen (\$41 beziehungsweise 65 oder A) geschrieben, läßt sich die Diskette wieder ganz normal beschreiben. Unter anderem kann nun auch wieder das Formatkennzeichen auf der Diskette geändert werden. (Bruno Henze/ah)

DATA Eingabehilfe

Das kleine Programm ist eine Hilfe zum Eintippen von DATAs. In Zeile 1 wird nach der Startzeilennummer gefragt, ab der die DATAs beginnen und in welcher Schrittweite die Zeilennummer erhöht werden soll.

Nachdem das Programm mit RUN gestartet und die beiden Parameter eingegeben wurden, erscheinen 20 Zeilen mit der gewünschten Startzeilennummer und dahinter das Wort »DATA«.

Diese Zeilen können nun editiert werden.

Das Programm läßt sich beliebig verändern. Statt DATA kann man auch PRINT oder ähnliches einsetzen.

```

1 INPUT "STARTZEILENNUMMER";X: INPUT "SCHRITTWEITE";Y: FOR
Z=1 TO 20
2 PRINT X "DATA": X=X+Y: NEXT:PRINT "{HOME}{CRSR DOWN}": END
(Frank Besler/ah)

```

8925 Blocks free

Das nun folgende kleine Programm wird Sie verblüffen. Mit ihm läßt sich beim LISTen des Directory (Inhaltsverzeichnis) die Meldung »8925 BLOCKS FREE« erzeugen. Frei sind natürlich nur 664 Blocks. Das Programm schreibt einfach so oft 255 in die BAM (Block Availability Map oder Blockbelegungsplan), bis die gewünschte Zahl erreicht ist.

Alle auf der Diskette gespeicherten Programme können problemlos geladen werden. Speichern sollte man auf dieser Diskette jedoch nichts mehr, es sei denn, Sie haben vorher den Befehl

```
OPEN 15,8,15,"V" <RETURN>
einggegeben.
```

```

10 INPUT "WIEVIELE BLOCKS FREE";B
13 IF B>8925 THEN 10
15 HI=INT(B/255): LO=B-HI*255
20 OPEN 1,8,15,"IO": OPEN 2,8,2,"#"
25 PRINT#1,"U1";2;0;18;0
27 IF HI<1 THEN N=4: GOTO 60
30 FOR N=4 TO (HI+1)*4 STEP 4
40 PRINT#1,"B-P";2;N
45 PRINT#2,CHR$(255)
50 NEXT
60 IF N<144 THEN PRINT#1,"B-P";2;N
70 IF N<144 THEN PRINT#2,CHR$(LO)
75 IF N<136 THEN GOSUB 100
80 PRINT#1,"U2";2;0;18;0
90 CLOSE 2: PRINT#1,"IO": INPUT#1,A$,B,C:
PRINT A,A$,B,C: CLOSE 1
95 END

```

```
100 FOR I=N+4 TO 140 STEP 4: PRINT #1,"B-P";2;I:
PRINT #2,CHR$(0): NEXT: RETURN
```

(Tobias Wörmann/ah)

Adventurelister

Kommt man bei Abenteuerspielen (Adventures) nicht mehr weiter, bietet es sich an, die Texte zu untersuchen. Als Werkzeug könnte zum Beispiel ein Maschinensprachemonitor (SMON) dienen. Aber Basic-Programmierer können mit solchen Werkzeugen für Assemblerfreaks recht wenig anfangen. Daher wurde ein Basic-Programm entwickelt, das die Texte in Adventures sucht und auf dem Bildschirm ausgibt. So wird das Programm gehandhabt:

1. Adventure laden
 2. Reset-Knopf drücken (siehe Ausgabe 6/85)
 3. Programm »Adventurelister« abtippen oder von Diskette beziehungsweise Kassette laden
 4. RUN <RETURN> eingeben
- Eine Programmunterbrechung erreicht man durch Drücken einer beliebigen Taste. Weiter geht es bei einem erneuten Druck auf eine Taste. Einen Programmabbruch erzielt man mit <Q>.

```
10 X=2048
20 Y=PEEK(X): IF Y>64 AND Y<91 THEN PRINT CHR$(Y):
GOTO 40
30 PRINT". ";
40 X=X+1: GET R$: IF R$<>" " THEN 60
50 GOTO 20
60 IF R$="Q" THEN END
70 GET Y$: IF Y$=" " THEN 60
80 GOTO 20
```

(Peter Riedlberger/ah)

Der kleinste Sprite-Editor

Mit diesem Zweizeiler läßt sich ein Sprite entwerfen. Die Sprite-Daten werden dabei automatisch auf dem Bildschirm ausgegeben.

Wenn das Programm geladen beziehungsweise abgetippt worden ist, darf es zunächst nicht mit RUN <RETURN> gestartet werden. Zuerst ist im Direktmodus mit der <SHIFT+CLEAR/HOME>-Taste der Bildschirm zu löschen. Jetzt kann, angefangen in der linken oberen Bildschirmcke, das Sprite entworfen werden:

<*> setzt einen Punkt
<SPACE> löscht beziehungsweise überspringt einen Punkt
Das Eingabefeld hat eine Größe von 21 Spalten und 24 Zeilen.

Sind Sie mit Ihrem Kunstwerk zufrieden und wollen die Sprite-Daten ausgeben, ist der Cursor am Anfang der vorletzten Bildschirmzeile zu positionieren.

RUN <RETURN> an dieser Stelle startet schließlich den Sprite-Editor, der die Daten errechnet und auf dem Bildschirm ausgibt.

```
1 FOR I=0 TO 2: A=0: FOR N=0 TO 7: A=A-2↑(7-N)*
(PEEK (1024+N*I*8) = 42): NEXT
```

```
2 PRINT A;: NEXT: PRINT: X=X+1: IF X<21 THEN 1
```

(Daniel Wicker/ah)

Druckersequenzen

Möchte man seinen Epson-kompatiblen Drucker dazu bewegen von nun an alles zu unterstreichen, ist nach dem entsprechenden OPEN- und PRINT-Befehl folgende Befehlssequenz erforderlich:

```
CHR$(27)"-CHR$(49)
```

Man muß also für diese Sequenz insgesamt 19 Tasten drücken.

Es geht aber auch einfacher und platzsparender. Der Basic-Interpreter erlaubt es uns nämlich, die Anzahl der zu drückenden Tasten im Gänsefußmodus auf drei zu reduzieren.

```
10 OPEN 1,4,1 <252>
20 PRINT#1,CHR$(27);CHR$(83);CHR$(1);:REM <094>
INDEX
30 PRINT#1,CHR$(27);CHR$(15);:REM <090>
132 ZEICHEN/ZEILE
40 PRINT#1,CHR$(27);CHR$(51);CHR$(14);:REM <159>
14/216 ZOLL ZEILENABS.
50 PRINT#1,"TEST" <093>
60 PRINT#1,"TEST" <103>
70 PRINT#1,CHR$(27);CHR$(64):REM <236>
DRUCKER NORMIEREN
100 REM JETZT KUERZER <015>
110 PRINT#1," <CTRL-[>S1 <CTRL-[,<CTRL-O,<CTRL- <197>
-[>3<CTRL-N>"
120 PRINT#1,"TEST" <163>
130 PRINT#1,"TEST" <173>
140 PRINT#1," <CTRL-[>e" <133>
150 CLOSE 1 <161>
200 REM BYTE-VERBRAUCH OH. REM-ZEILEN, <093>
201 REM DER TESTDRUCKE, DES OPEN, DER <129>
210 REM ZEILENPOINTER U. -NUMMERN(HI/LO) <103>
220 REM 1. PRG 38 BYTE (KURZVERSION) <145>
230 REM 2. PRG 20 BYTE (AUCH OH. NOR- <053>
MIERUNG D. DRUCKERS
```

Listing 2. Drucker-Sequenzen lassen sich platzsparender eingeben

<SHIFT+CTRL+:> + <1>

Dabei steht <SHIFT+CTRL+:> für CHR\$(27) und <1> für CHR\$(49).

Diese Methode läßt sich natürlich auch auf andere Steuerzeichen anwenden.

Aus einer einfachen Beziehung heraus kann man sich die Tastenkombinationen errechnen, um die Codes 1 bis 26 zu senden.

Zuvor muß jedoch gesagt werden, daß sich diese Methode nicht auf die Codes 0, 13 und 20 anwenden läßt.

Die Stelle, die ein Buchstabe im Alphabet einnimmt (zum Beispiel B=2) ergibt den gewünschten Code, wenn der Buchstabe zusammen mit der <SHIFT>- und <CTRL>-Taste gedrückt wird.

Das kleine Beispielprogramm (Listing 2) veranlaßt einen FX-80, Text in der kleinstmöglichen Schrift mit angepaßtem Zeilenabstand auszugeben.

(Harald Wengner/ah)

SYS-Zeile für Basic-Programme

Geben Sie zunächst NEW <RETURN> ein, um ein eventuell gespeichertes Basic-Programm zu löschen. Im nächsten Schritt läßt sich die SYS-Zeile eingeben, die ungefähr so aussehen könnte:

```
1986 SYS(1111)"64'er" <RETURN>
```

Wichtig ist, daß diese SYS-Zeile nach dem Schema Zeilennummer, Space, SYS-Befehl mit nachfolgender, in Klammern eingeschlossener, vierstelliger Zahl aufgebaut ist. anschließend führen Sie im Direktmodus folgende Befehlssequenz aus:

```
S=2297:A$+STR$(S):FOR L=2 TO 5:POKE
2053+L,ASC(MID$(A$,L,1)):NEXT <RETURN>
```

Mit der nächsten Zeile, die ebenfalls im Direktmodus einzugeben ist, legen Sie ein winziges Maschinenprogramm an die Startadresse:

```
POKE S,169:POKE S+1,9:POKE S+2,133:POKE S+3,44:POKE
S+4,76:POKE S+5,142:POKE S+6,166 <RETURN>
```

Nun läßt sich nach dem Hochsetzen der Basic-Startadresse mit den drei Befehlen

```
POKE 44,9:POKE 2304,0:NEW <RETURN>
```

ein Basic-Programm laden oder eingeben. Ist das geschehen, müssen Sie das Programm mit der Zeile

```
POKE 44,8:SAVE"name",geräteadresse <RETURN>
```

speichern. In Zukunft ist Ihr Basic-Programm vor Neugierigen sicher (bei LIST erscheint nur die SYS-Zeile).

(K. Deisenhofer/ah)