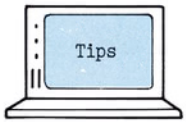
Tips & Tricks für Einsteiger



Neben vielen nützlichen Programmierhinweisen zeigen wir Ihnen diesmal zwei Spielereien mit dem Cursor, einen interessanten Tip zu Textverarbeitungsprogrammen und vieles weitere. Erwähnenswert ist auch ein Programm, mit dem man ein häufig benötigtes Sprite-Entwurfsblatt auf einem Commodore MPS 801-Drucker ausgeben kann.

Hatten Sie auch schon mal während der Arbeit mit dem C 64 plötzlich aus heiterem Himmel einen »out of memory error«? Abgesehen davon, daß Sie wirklich ein äußerst langes Programm im Speicher haben könnten, taucht diese Meldung auch nach dem Laden von vielen Maschinenprogrammen auf (diese erkennen Sie daran, daß sie nicht mit RUN, sondern mit einem SYS-Befehl gestartet werden). Die Fehlermeldung kommt meistens dann, wenn Sie nach dem Laden entweder ein Basic-Programm mit RUN starten, oder aber ein weiteres Programm laden möchten.

Warum taucht dieser Fehler auf? Wenn Sie zum Beispiel eine Variable mit einem Wert belegen (A = 10), dann legt der C 64 diese Zahl 10 direkt hinter dem aktuellen (Basic-)Programm im Speicher ab. Dazu muß er aber wissen, wo das Programm aufhört. Zu diesem Zweck gibt es einen sogenannten »Zeiger«, der die letzte, vom Programm belegte Adresse angibt. Der Zeiger wird aktualisiert, wenn Sie das Programm in seiner Länge verändern, also zum Beispiel eine Zeile hinzuschreiben oder löschen. Der Zeiger wird aber auch gesetzt, wenn Sie ein Programm von Diskette oder Kassette laden.

Maschinenprogramme werden vom Programmierer meistens in Speicherbereiche gelegt, die von Basic-Programmen aus normalerweise nicht überschrieben werden. Diese Bereiche haben sehr hohe Adressen, auf die der eben beschriebene Programmende-Zeiger nach dem Laden gesetzt wird. Wenn Sie jetzt einen Befehl eingeben, der eine Variable definiert, erkennt der C 64, daß hinter dem Programmende-Zeiger bereits ein Speicherbereich liegt, der von einem Basic-Programm nicht benutzt werden darf.

Ein Beispiel: Basic-Programme haben im allgemeinen den Speicherbereich von Adresse 2049 bis Adresse 40959 zur Verfügung. Maschinenprogramme verwenden jedoch häufig den etwas höher liegenden Bereich zwischen Adresse 49152 und Adresse 53247. Wenn ein Programm in diesen hohen Bereich geladen, also der Zeiger zum Beispiel auf Adresse 51234 gesetzt wird, kommt es beim späteren Belegen einer Variablen zum »out of memory error«.

Auch beim Laden von Programmen wird übrigens eine Variable belegt: Der C 64 merkt sich den Namen des zu ladenden Programms am Ende des Basic-Speichers. Vorher wird jedoch anhand des Programmende-Zeigers überprüft, ob dafür überhaupt noch Platz ist. In unserem Beispiel mit der Adresse 51234 ist dies nicht der Fall. Also kommt es auch beim Nachladen von Programmen zum »out of memory error«.

Doch was tun? Geben Sie einfach NEW ein. Keine Angst: Dadurch wird das Maschinenprogramm nicht gelöscht. Lediglich der Programmende-Zeiger wird auf Adresse 2051 gesetzt, also festgelegt, daß sich kein Basic-Programm mehr im Speicher befindet.

(tr)

Professionelles Auswahlmenü

Wenn man ein Programm schreibt, in dem der Benutzer aus mehreren Programmfunktionen auswählen kann, kommt man um die Erstellung eines Menüs nicht hinweg. Dieses Menü kann zum Beispiel so aussehen: Die verschiedenen Wahlmöglichkeiten bekommen eine Nummer und der Benutzer gibt beim Programmablauf die gewünschte Nummer ein. Das Programm verzweigt dann entsprechend der Eingabe.

Eine optisch schon wesentlich ansprechendere Methode sehen Sie in Listing 1. Tippen Sie dieses Listing bitte ein und beachten dabei unsere Eingabehinweise auf Seite 78.

Das Programm ist nur ein Beispiel, um das Funktionsprinzip zu erklären. Es enthält sechs Menüpunkte, deren Texte in der Zeile 90 abgelegt sind. Mittels der Cursor-Tasten kann man sich einen Menüpunkt aussuchen und durch Druck auf die <RETURN>-Taste aktivieren. In Zeile 80 könnte zum Beispiel eine »ON A GOTO 200,300,...«-Verzweigung erfolgen.

(Ulrich Wichelhaus/tr)

```
10 A=1                                                  <183>
15 PRINT"{CLR,4DOWN}"                                   <002>
20 FOR I=1 TO 6                                         <249>
30 READ M$(I)                                           <211>
40 PRINT TAB(10)M$(I)                                   <150>
50 NEXT                                                 <060>
60 GOSUB 100                                            <004>
70 PRINT"{CLR}SIE WAEHLTEN PUNKT"A                      <176>
80 END:REM HIER KOENNTE DANN VERZWEIGUNG E
   RFOLGEN                                              <007>
90 DATA PUNKT 1,PUNKT 2,PUNKT 3,PUNKT 4,PU
   NKT 5,PUNKT 6                                        <109>
100 REM ******************************                 <238>
105 REM HIER BEGINNT DIE AUSWAHLROUTINE                 <135>
110 POKE 214,A+4:POKE 211,10:SYS 58640:PRI
    NT"{RVSON}"M$(A)                                    <163>
120 GET X$:IF X$=""THEN 120                             <189>
130 IF X$=CHR$(13)THEN RETURN                           <111>
140 IF X$="{DOWN}"THEN 180                              <029>
150 POKE 214,A+4:POKE 211,10:SYS 58640:PRI
    NT M$(A)                                            <055>
160 A=A-1:IF A<1 THEN A=1                               <118>
170 GOTO 110                                            <114>
180 POKE 214,A+4:POKE 211,10:SYS 58640:PRI
    NT M$(A)                                            <085>
190 A=A+1:IF A>6 THEN A=6                               <023>
200 GOTO 110                                            <144>
```

Listing 1. Ein professionelles Auswahlmenü

Bildschirmeffekt

Zwei interessante Bildschirm-Flimmer-Effekte zum Ausprobieren:

```
10 FOR I=18 TO 30:POKE 53265,I:NEXT I:GOTO 10
10 FOR I=1 TO 255:POKE 53270,I:NEXT I:GOTO 10
```

oder beide Effekte kombiniert:

```
10 FOR I=18 TO 30:POKE 53270,I:POKE 53265,I:NEXT I:GOTO 10
```

(Tzimas Kosta/tr)

Hilfe beim Eintippen von DATA-Zeilen

Wenn Sie ein Programm abtippen, das sehr viele DATA-Zeilen enthält, schleichen sich fast automatisch irgendwelche Tippfehler ein. Meist werden diese DATA-Zeilen über eine FOR-NEXT-Schleife mit READ- und POKE-Anweisungen bearbeitet. Ein Tippfehler macht sich dann durch einen »illegal quantity error« bemerkbar. Nun kann man mit

```
LIST (PEEK(63)+PEEK(64)*256)
```

die fehlerhafte DATA-Zeile sofort auf den Bildschirm bringen. Voraussetzung ist natürlich, daß Sie sich nicht schon beim Eingeben der FOR-NEXT-Zeile vertippt haben.

(Oliver Twesten/tr)

Sprite-Entwurfsblatt für MPS-Drucker

Das Mini-Programm in Listing 2 druckt auf einem MPS 801 oder MPS 803 ein Sprite-Entwurfsblatt. Rechts daneben können dann die errechneten Werte für jede Zeile eingetragen werden. Siehe dazu auch das entsprechende Kapitel im C 64-Handbuch.

(Steffen Unterreiner/tr)

```
10 REM SPRITE-ENTWURFSBLATT MPS801/803-DRU
   CKER                                    <088>
20 OPEN 4,4,1:PRINT#4,CHR$(14);"{2SPACE}SP
   RITE-ENTWURFSBLATT":PRINT#4             <086>
30 FOR I=1 TO 21                           <050>
40 PRINT#4,CHR$(15)"{8SPACE}00000000000000
   0000000000 000,000,000"CHR$(8)          <130>
50 NEXT                                    <060>
60 PRINT#4,CHR$(15)"{8SPACE}TTTTTTTTTTTTTT
   TTTTTTTTT"CHR$(8)                       <169>
70 CLOSE 4                                 <105>
```

Listing 2. Sprite-Entwurfsblatt für Commodore-Drucker

Professioneller Cursor

Kleines Listing mit großem Effekt: Listing 3 zeigt, wie man mit wenig Aufwand bei Texteingaben einen professionell aussehenden Cursor erzeugen kann. Einfach eintippen und ausprobieren!

(Jan C. Tchinitchian/tr)

```
10 BL$="{CTRL P}{LEFT}"                    <077>
20 PRINT"{CLR}"                            <008>
30 PRINT"BITTE GEBEN SIE TEXT EIN:"        <088>
40 PRINT BL$;                              <182>
50 GET X$:IF X$=""THEN 50                  <095>
60 IF X$=CHR$(13)THEN 80                   <202>
70 PRINT X$;:TE$=TE$+X$:GOTO 40            <216>
80 PRINT"SIE GABEN EIN:"                   <180>
90 PRINT TE$                               <058>
```

Listing 3. Professioneller Cursor

Hilfe für Datasetten-Besitzer

Angenommen, man hat eine Kassette mit mehreren Programmen und möchte hinter dem letzten Programm etwas speichern. Natürlich hat man sich zwar den Anfangszählerstand des letzten Programms, nicht aber das Ende aufgeschrieben. Um das Ende nun zu finden, gibt es zwei Möglichkeiten:

1. Man lädt das Programm vollständig in den C 64, was von Nachteil sein kann, wenn man bereits ein Programm im Speicher hat.
2. Man spult die Kassette bis kurz vor den Anfang des letzten Programms und gibt »VERIFY« ein. Dadurch wird das Programm auf der Kassette bis zum Schluß abgearbeitet und der Speicher des C 64 bleibt trotzdem unverändert. Anschließend erscheint ein »verify error«, den man jedoch ignorieren kann.

Die Kassette hat jetzt die richtige Bandposition.

(Peter Rothe/tr)

NEW rückgängig

Schnell ist es passiert, daß man einmal aus Versehen NEW eingetippt hat. Das Basic-Programm ist dann rettungslos verloren. Oder doch nicht? Nein, mit einigen POKEs und einem SYS-Befehl kann man es »wiederbeleben«:

```
POKE2050,1:SYS42291:POKE46,PEEK(35):POKE45,PEEK(781)+2:
CLR
```

In ganz seltenen Fällen kann es vorkommen, daß der Computer daraufhin einen »illegal quantity error« meldet. Dann muß man noch zusätzlich

```
POKE46,PEEK(35)+1:POKE45,PEEK(781)-254
```

eingeben.

Achtung: Das Ganze funktioniert nur, wenn man nach dem versehentlichen NEW-Befehl keine Variable definiert hat (zum Beispiel A = 10)!

(Alexander Rühl/tr)

Tip zu Textverarbeitungen

Wenn man ein Textverarbeitungs-Programm besitzt, möchte man es natürlich irgendwann auch zum Schreiben von Briefen verwenden. Tip: Erstellen Sie einen Briefkopf mit zum Beispiel Ihrem Namen, Ihrer Anschrift und eventuell Bankverbindung und so weiter. Speichern Sie diesen »Text« dann auf Diskette. Wenn Sie nun einen Brief schreiben möchten, laden Sie diesen Briefkopf und vervollständigen ihn.

Auf diese Weise können Sie auch zum Beispiel Formbriefe erstellen: Überlegen Sie sich für verschiedene Anlässe (Geburtstag, Heirat, etc.) einen passenden Text und setzen für die persönlichen Daten (Name, Datum, etc.) zum Beispiel »XXX« ein. Wenn der Anlaß da ist, laden Sie diesen Text und setzen nur noch die speziellen Daten ein. Auf diese Weise haben Sie immer »persönlich aussehende« Briefe zur Hand.

Die Sache hat noch einen weiteren Vorteil: Auf diese Weise können Sie sich einmal in aller Ruhe einen ausgefeilten Text überlegen, der dann bei Bedarf schnell bei der Hand ist.

Auch gut zu verwenden, wenn man sich bei mehreren Firmen mit demselben Schreiben bewerben möchte.

(Frank Werner/tr)

Zweifarbiger Bildschirm

Der folgende Einzeiler erzeugt einen zweifarbigen Bildschirmhintergrund.

```
10 POKE53281,7:::::::::::::::::::::POKE53281,13:GOTO10
```

Wichtig ist dabei die richtige Anzahl an Doppelpunkten. Diese verzögern das »Programm« nämlich so, daß sich ein stehendes Bild ergibt. Ändern Sie ruhig einmal die Anzahl an Doppelpunkten und probieren Sie auch andere Farbwerte für die Zahlen 7 und 13 hinter den POKEs.

(Olaf Hein/tr)

Turbo-Cursor

Wenn man seinen Cursor so weit beschleunigen möchte, daß man kein vernünftiges Wort mehr schreiben kann, genügen zwei POKEs:

```
POKE650,128:POKE56325,1
```

Der zweite POKE stellt die eigentliche Geschwindigkeitsbeeinflussung dar. Die Zahl 1 bedeutet größtmögliche Geschwindigkeit (Werte zwischen 1 und 255). Der erste POKE dient nur dazu, die Tastatur auf Wiederholfunktion zu schalten. Versuchen Sie nun einmal, das Wort »COMPUTER« zu schreiben. Sie werden an Ihren Fähigkeiten zweifeln . . .

(Axel Stämmler/tr)