Schleifen mit Format

In manchen Pascal-Versionen geht es automatisch, beim C 64 nur mit einem Trick: das Einrücken von Schleifen. Anhand eines Programms in Maschinensprache bringen Sie Form in Ihre Listings. Wir zeigen auch, wie es geht.

Anhänger der strukturierten Programmierung haben es gelernt, selbst der in Schulungen: ein Programm modular aufbauen, in kleine Schritte zerlegen, übersichtlich gestalten. Leider ist es mit dem C 64 nicht so ganz einfach. Außer IF..THEN, GOTO, GOSUB und FOR..NEXT wird strukturiertes Codieren nicht unterstützt.

Eine Möglichkeit ist, FOR..NEXT-Schleifen einzurücken. Normalerweise macht man das, indem nach der Zeilennummer ein Doppelpunkt gesetzt wird und erst danach die gewünschte Anzahl Leerstellen. Ein kurzes Beispiel:
Schleife ohne einrücken:

```
10 FOR I=1 TO 10
20 A=A+I
30 GOSUB 100
40 FOR J=1 TO 10
50 PRINT I,J,A*J
60 NEXT J
70 NEXT I
```

Schleife mit einrücken durch Doppelpunkte:

```
10 FOR I=1 TO 10
20 : A=A+I
30 : GOSUB 100
40 : FOR J=1 TO 10
50 :    PRINT I,J,A*J
60 : NEXT J
70 NEXT I
```

Das sieht schon viel besser aus. Bei langen Listings werden Sie diese Form schätzen lernen. Allerdings, das Nonplusultra ist es auch nicht. Besser wäre es, das Listing würde so aussehen:

```
10 FOR I=1 TO 10
20    A=A+I 100
30    GOSUB 100
40    FOR J=1 TO 10
50       PRINT I,J,A*J
60    NEXT J
70 NEXT I
```

Um das zu erreichen, müssen wir eine kleine Maschinenroutine schreiben. Aber wie?

Ein Gedanke liegt nah: Warum nicht die normale LIST-Routine des C 64 verwenden? Sie erledigt ja schon einen großen Teil der Aufgabe, nämlich das normale LISTen. Was fehlt, ist nur noch das Einrücken. Zuerst muß also ein dokumentiertes ROM-Listing her. Dort finden wir die LIST-Routine des Basic-Interpreters ab $A69C. Sie geht bis $A740. Soweit, so gut. Aber wie greifen wir in diese Routine ein? Sieht man sich die LIST-Routine etwas genauer an, findet man einen indirekten Sprung ab Adresse $A717 (JMP ($0306)). Das bedeutet, an dieser Stelle springt die LIST-Routine zu der Adresse, die sich aus dem Inhalt der Adressen $0306/$0307 (LO/HI-Byte) ergibt. Man spricht in so einem Fall auch von einem Vektor. Dort kurz spioniert, finden wir in $0306/$0307 die Zahlen $1A und $A7, zusammengesetzt also die Adresse A71A, das heißt genau die Adresse, die dem JMP($0306) folgt. Damit haben wir genau das, was wir suchen, eine Möglichkeit, in die LIST-Routine einzugreifen. Denn $0306 steht im RAM, kann also (von uns) geändert werden.

Im Prinzip brauchen wir also nur folgendes zu machen: Wir schreiben die Anfangsadresse unserer Routine in die Speicherstellen $0306/$0307. Damit springt die Original-LIST-Routine unsere neue Routine an und führt sie aus. In »Fachkreisen« würde man sagen, der LIST-Vektor ist verbogen worden. Am Ende unseres Programms müssen wir noch dafür sorgen, daß die alte LIST-Routine wieder fortgesetzt wird und zwar machen wir das mit JMP $A71A.

Grundsätzlich kennen wir nun also das Wie. In der Zwischenzeit taucht aber noch ein Gedanke auf: Wie soll das Programm gestartet werden? Nach dem Lesen des Assembler-Kurses (Assembler ist keine Alchimie) im 64'er und einigen anderen Artikeln fällt das Stichwort: Interrupt.

Interrupt war für mich immer ein Wort, vor dem ich mich etwas gedrückt habe, aber so schlimm ist es gar nicht. Doch zuerst einmal zur Aufgabe: Die neue LIST-Routine soll mit der Funktionstaste F7 an- und mit F1 ausgeschaltet werden. Um das zu erreichen, muß noch einmal ein Vektor verbogen werden, und zwar der Interrupt-Vektor in Adresse $0314/$0315 auf unser eigenes Programm (siehe Listing 2, Zeilen 290 bis 360). Wir müssen dafür sorgen, daß die Tasten F7 und F1 dauernd abgefragt werden. Das erreicht man, indem die Interrupt-Routine erweitert wird. Auch hier hilft ein kleines Programm (Listing 2, Zeilen 364 bis 372). Im Prinzip soll unsere neue LIST-Routine so ablaufen:

1. Initialisieren des Programms mit SYS adresse (»adresse« können wir selbst festlegen)
2. mit F7 einschalten und mit F1 ausschalten.
3. mit dem ganz normalen LIST-Befehl ein beliebiges Basic-Programm auf dem Bildschirm oder Drucker ausgeben. Auf dem Drucker sollen die Basic-Befehle FOR und NEXT fettgedruckt werden.

Und damit auch zu sehen ist, ob die neue oder die normale Routine aktiviert ist, soll der Rahmen beim Drücken von F7 die Farbe wechseln (Listing 2, Zeile 452/453) und bei F1 ebenso (Listing 2, Zeile 522/523).

Die LIST-Routine

Die eigentliche neue LIST-Routine finden Sie in Listing 2, Zeile 1000 bis 1340. Danach folgen einige Unterprogramme, zum Beispiel Fettdruck ein-/ausschalten und Leerzeichen ausgeben.

Um das Programm zu verstehen, muß man folgendes wissen:

1. Wenn die neue LIST-Routine angesprungen wird, steht im Akku ein Zeichen aus dem Basic-Listing, das wir ausgeben wollen. Das kann jedes Zeichen sein zwischen der Basic-Zeilennummer und dem Ende einer Basic-Zeile (die Zeilennummer selbst wird vorher, von der alten List-Routine selbst, ausgegeben.).
2. Der Akku-Inhalt muß am Ende unserer LIST-Routine wieder an das Original-LIST übergeben werden. Aus diesem Grund wird er sicherheitshalber am Anfang der Routine mit PHA gesichert und am Ende mit PLA zurückgeholt.
3. Jeder Basic-Befehl wird im Speicher als Token abgelegt, eine Abkürzung. Der Befehl FOR hat den Wert $81 und NEXT den Wert $82.

Der Algorithmus ist jetzt nicht mehr schwer zu entwickeln. Da man FOR..NEXT-Schleifen (fast) beliebig schachteln kann, setzen wir einen Zähler ein (im Listing 2 ZAEHLER genannt), der die Anzahl der Verschachtelungen zählt: Bei jedem FOR wird ZAEHLER um 1 erhöht (Zeile 1210 bis

1240, bei jedem NEXT um 1 vermindert (Zeile 1160 bis 1180). Der Inhalt von ZAEHLER bestimmt auch die Anzahl der Leerzeichen, die am Anfang der Zeile, nach der Zeilennummer, ausgegeben werden (Zeile 1190 bis 1207). Der Anfang der Zeile wird durch eine 4 im Y-Register gekennzeichnet. Das hängt mit dem Aufbau einer Basic-Zeile im C 64 zusammen und mit der indirekt-indizierten Adressierung, die die LIST-Routine verwendet (Y=0 und Y=1 sind die Startadresse der Basic-Zeile, Y=2 und Y=3 die Zeilennummer und ab Y=4 folgt der Rest der Basic-Zeile).

Das Unterprogramm BLANKOUT (Zeile 2010 bis 2070 gibt Leerzeichen aus, und zwar so viele, wie in ZAEHLER stehen.

Die Routinen FETTDRUCK und NORMDRUCK schalten bei Ausgabe auf Drucker (mit OPEN 1,4:CMD1:LIST) Fettdruck an und aus. Interessant sind hier vielleicht die Zeilen 1510 bis 1530 oder 1710 bis 1730. Durch die Abfrage der Speicherstelle $9A (dezimal 154) kann die mit CMD gewählte Geräteadresse überprüft werden (4 steht für Drucker). Die Sequenz, die hier gewählt wurde, um Fettschrift

```
100  -;.LI1,4
110  -;.SY1,4
120  -;.OB"NEULIST.OBJ $5,P,W"
130  -;---------------------------------
140  -; PROGRAMM : LIST-ROUTINE VERAENDERT
142  -; NAME: NEULIST.$5000.SRC
144  -; INIT MIT SYS 5*4096
150  -; MIT F7 AN
160  -; MIT F1 AUSSCHALTEN
170  -; RUECKT FOR..NEXT-SCHLEIFEN EIN
180  -; UND FOR..NEXT AUF DRUCKER FETT
190  -;---------------------------------
200  -.EQ ZAEHLER= $FE;ANZAHL FOR-SCHACHTELUNGEN
210  -.EQ CHROUT = $FFD2;AUSGABE ZEICHEN
212  -.EQ STROUT = $AB1E;AUSGABE STRING
220  -.EQ IRQVEC = $0314;VEKTOR AUF IRQ
230  -.EQ IRQ    = $EA31;IRQ
240  -.EQ CTRL   = $028D ;FLAG FUER CTRL
250  -.EQ KEY    = $00CB ;LETZTE TASTE
260  -;---------------------------------
270  -.BA $5000 ;STARTADRESSE
280  -;---------------------------------
290  -;INTERRUPT AUF EIGENE ROUTINE VERBIEGEN
300  -           SEI
310  -           LDA #<(START)
320  -           STA IRQVEC      ;LO-BYTE
330  -           LDA #>(START)
340  -           STA IRQVEC+1    ;HI-BYTE
350  -           CLI
360  -           RTS
362  -;---------------------------------
364  -START      LDA KEY         ;WELCHE TASTE?
366  -F7         CMP #3          ;F7 ?
367  -           BNE F1          ;NEIN
368  -           JSR LISTNEU     ;NEUE LISTROUTINE
369  -F1         CMP #4          ;F1 ?
370  -           BNE OLDIRQ      ;NEIN
371  -           JSR LISTALT     ;ALTE LISTROUTINE
372  -OLDIRQ     JMP IRQ
373  -;
374  -;---------------------------------
380  -;
390  -;      LISTVEKTOR AUF EIGENE LISTROUTINE
395  -;      START BEI LABEL"LIST"
400  -LISTNEU    LDA #<(LIST)
410  -           STA $0306       ;LISTVEKTOR LO
420  -           LDA #>(LIST)    ;HI-BYTE
430  -           STA $0307
440  -           LDA #0
450  -           STA ZAEHLER
452  -           LDA #$F6        ;BLAUER
453  -           STA $D020       ;RAHMEN
460  -           RTS
470  -;
480  -;
485  -;ALTEN LISTVEKTOR (A71A)
486  -;WIEDERHERSTELLEN MIT F1
490  -LISTALT    LDA #$1A
500  -           STA $0306
510  -           LDA #$A7
520  -           STA $0307
522  -           LDA #$FB        ;GRAUER
523  -           STA $D020       ;RAHMEN
530  -           RTS
540  -;*********************************
1000 -;.LI1,4
1010 -LIST       PHA
1020 -           JSR NORMDRUCK
1160 -           CMP #$82        ;NEXT?
1170 -           BNE LI1         ;NEIN
1180 -           DEC ZAEHLER     ;N=N-1
1182 -           JSR FETTDRUCK
1190 -LI1        CPY #4          ;ZEILENANFANG?
1200 -           BNE LI2
1205 -           JSR BLANKOUT
1207 -           JSR BLANKOUT
1210 -LI2        CMP #$81        ;FOR ?
1220 -           BNE LIOUT
1222 -           JSR FETTDRUCK
1240 -           INC ZAEHLER     ;N=N+1
1330 -LIOUT      PLA
1340 -           JMP $A71A       ;LIST
1350 -;
1360 -;---------------------------------
1500 -FETTDRUCK PHA
1510 -           LDA $9A         ;CMD = 4?
1520 -           CMP #4
1530 -           BNE FEOUT       ;NEIN
1540 -           TYA
1550 -           PHA
1560 -           LDA #<(FETT)    ;DRUCKER
1570 -           LDY #>(FETT)    ;AUF
1580 -           JSR STROUT      ;FETTDRUCK
1590 -           PLA
1600 -           TAY
1610 -FEOUT      PLA
1620 -           RTS
1630 -FETT       .BY 27,"E",0    ;FETTDRUCK
1640 -;---------------------------------
1700 -NORMDRUCK PHA
1710 -           LDA $9A         ;CMD4?
1720 -           CMP #4
1730 -           BNE NOOUT       ;NEIN
1740 -           TYA
1750 -           PHA
1760 -           LDA #<(NORM)    ;FETTDRUCK
1770 -           LDY #>(NORM)    ;aus
1780 -           JSR STROUT
1790 -           PLA
1800 -           TAY
1810 -NOOUT      PLA
1820 -           RTS
1830 -NORM       .BY 27,"F",0    ;FETTDRUCK AUS
1840 -;---------------------------------
2000 -;---------------------------------
2010 -BLANKOUT   LDX ZAEHLER     ;AUSGABE N BLANKS
2015 -           BPL BL1         ;WENN < 128
2016 -           INC ZAEHLER
2017 -           BMI BLANKOUT    ;WENN>128
2020 -BL1        BEQ BLOUT
2030 -           JSR BLANK       ;LEERZEICHEN
2050 -           DEX
2060 -           JMP BL1
2070 -BLOUT      RTS
2080 -;
2090 -;---------------------------------
2100 -CR         PHA             ; ZEILENVORSCHUB
2110 -           LDA #13         ;RETURN
2120 -           JSR CHROUT
2130 -           PLA
2140 -           RTS
2150 -;-----------------------------
2160 -;
2170 -BLANK      PHA             ;LEERZEICHEN
2180 -           LDA #" "
2190 -           JSR CHROUT
2200 -           PLA
2210 -           RTS
2220 -;
2230 -;---------------------------------
9999 -.EN

READY.
```

Listing 2. Der Assembler-Quelltext zu »Schleifen mit Format«

einzuschalten, gilt für Epson-kompatible Drucker (Zeile 1630 und Normalschrift Zeile 1830). Sie können diese Routinen natürlich entfernen, es ist Geschmacksache, die FOR..NEXT-Schleife auch noch fettgedruckt zu sehen (Listing 1).

Hinweise zum Abtippen

Listing 2 ist der Quelltext oder, wie man auch sagt, der Sourcecode der neuen Listroutine. Wenn Sie wollen, können Sie ihn mit Hypra-Ass eingeben. Das hat den Vorteil, daß Sie das Programm weiterentwickeln oder verändern können. Wollen Sie das Programm lediglich benutzen, tippen Sie am besten Listing 3 mit dem MSE ab. Gestartet wird mit SYS 5*4096 oder SYS 20480. Mit F7 ist die neue LIST-Routine aktiv, mit F1 abgeschaltet. Auch während des LISTens kann umgeschaltet werden. Eine korrekte FOR..NEXT - Schleife wird daran erkannt, daß sich FOR und NEXT auf der gleichen Höhe befinden (Listing 1).

Wenn Sie das Programm etwas genauer analysieren, dürfte es nicht schwerfallen, eigene Wünsche zu verwirklichen.

(H.Zwartscholten/gk)

```
10 PRINTI:PRINTA
20 FORI=1TO4:PRINT"LIST TEST  ":NEXTI
30 A=1
40 B=2
50 FORI=1TO2
55 NEXTI
60 D=4:E=5
70 FORI=1TO2
80   FORJ=1TO2
90     I=3
91   NEXTJ
92   A=5
93 NEXTI
100 FORI=1TO20:PRINTI;:NEXTI
110 A=4:B=5
120 FORJ=1TO2
130   X=X+2:Z=A
140   PRINTX;"ALFA=5:3"
150 NEXTJ
1100 FORI=1TO20:PRINTI;:NEXTI
1110 A=4:B=5
1120 FORJ=1TO2
1130   X=X+2:Z=A
1131   FORQ=1TO2
1132     X=X+2:Z=A
1133     PRINTX;"ALFA=5:3"
1134     IFA=3THENB=4:C=5
1136   NEXTQ
1140   PRINTX;"ALFA=5:3"
1150 NEXTJ

READY.
```

Listing 1. Ein-Beispielprogramm

```
 PROGRAMM : NEULIST.OBJ $5  5000 50B8
-------------------------------------

5000 : 78 A9 0D 8D 14 03 A9 50  E2
5008 : 8D 15 03 58 60 A5 CB C9  E2
5010 : 03 D0 03 20 20 50 C9 04  F4
5018 : D0 03 20 34 50 4C 31 EA  FA
5020 : A9 44 8D 06 03 A9 50 8D  E9
5028 : 07 03 A9 00 85 FE A9 F6  00
5030 : 8D 20 D0 60 A9 1A 8D 06  BB
5038 : 03 A9 A7 8D 07 03 A9 FB  D3
5040 : 8D 20 D0 60 48 20 7F 50  42
5048 : C9 82 D0 05 C6 FE 20 68  DD
5050 : 50 C0 04 D0 06 20 96 50  78
5058 : 20 96 50 C9 81 D0 05 20  04
5060 : 68 50 E6 FE 68 4C 1A A7  2A
5068 : 48 A5 9A C9 04 D0 0B 98  87
5070 : 48 A9 7C A0 50 20 1E AB  96
5078 : 68 A8 68 60 1B 45 00 48  C7
5080 : A5 9A C9 04 D0 0B 98 48  BE
5088 : A9 93 A0 50 20 1E AB 68  9F
5090 : A8 68 60 1B 46 00 A6 FE  E5
5098 : 10 04 E6 FE 30 F8 F0 07  E0
50A0 : 20 B0 50 CA 4C 9E 50 60  41
50A8 : 48 A9 0D 20 D2 FF 68 60  9C
50B0 : 48 A9 20 20 D2 FF 68 60  68
```

Listing 3. Der Objectcode zu »Schleifen mit Format«. Benutzen Sie zur Eingabe den MSE.

ÜberLISTet

Mit dieser Betriebssystemerweiterung wird das Schreiben von Basic-Programmen zum Vergnügen. Scrolling aufwärts und abwärts sowie einige Zusatzfunktionen machen es möglich.

Dieses Maschinenprogramm (eventuell in Verbindung mit einem Toolkit) macht das Editieren von Basic-Texten fast so komfortabel wie mit einem guten Textverarbeitungssystem. Man kann damit nämlich nicht nur, wie von vielen teuren Basic-Erweiterungen bekannt, das Listing mit den Cursortasten hinauf und hinunterscrollen, sondern auch:

- andere Erweiterungen verwenden
- einzelne Zeilen löschen und einfügen
- eine Zeile bis beziehungsweise ab Cursorposition löschen
- eine ganze Zeile mit Leerzeichen füllen
- einen Zeilenausschnitt beliebig oft an andere Positionen kopieren.

Dies alles geschieht auf einfachen Tastendruck und ohne Absturzgefahr.

Um mit möglichst vielen anderen Programmen zusammenarbeiten zu können, liegt das zirka 1300 Byte lange Programm im Bereich $7A00 - 7F20 (Hexadezimal). Dazu muß das Ende des Basic-Bereichs gegebenenfalls herabgesetzt werden (POKE 56,122:POKE 55,0: NEW). Mit »SYS 31232« wird die Erweiterung in den Interrupt eingehängt und ist dann sofort einsatzbereit. Es wird nur der IRQ-Vektor verändert.

Das Listing läßt sich im Direktmodus mit Hilfe der Cursortasten fließend hoch- und runterscrollen. Die Programmzeilen können dabei natürlich wie gewohnt geändert und mittels der RETURN-Taste übernommen werden. Stößt man jedoch mit dem Cursor gegen den oberen beziehungsweise unteren Bildschirmrand, wird der Inhalt des Bildschirms in die entsprechende Richtung gescrollt und die nächste Zeile ausgegeben.

Das Programm geht dabei so vor: Es sucht die am nächsten zum Cursor stehende Zeilennummer. Steht keine Zahl in der ersten Spalte, wird normal gescrollt (auch nach unten). Ist die gefundene Zahl größer gleich 64000, wird der Cursor auf diese Zeile gesetzt und das Scrolling verhindert. Andernfalls wird die nächsthöhere beziehungsweise nächstniedrigere Zeile aus dem Basic-Listing gesucht, der Bildschirm in die richtige Richtung gescrollt und die gefundene Zeile gelistet.

Die Erweiterung arbeitet nur im Direktmodus, das heißt wenn in einem Programm ein »INPUT«-Befehl oder ähnliches vorkommt, verhält sich der Bildschirm normal.

Zusätzlich bietet dieses Programm noch einige andere Möglichkeiten, die das Editieren erleichtern. Diese Funktionen werden durch gleichzeitiges Drücken der Control-Taste und einer Buchstabentaste aufgerufen. Zwischen zwei Anführungszeichen werden diese »Control-Codes« jedoch normal ausgegeben.

CTRL-i Einfügen (Insert) einer Leerzeile. Der untere Teil des Bildschirms wird nach unten geschoben. Dies dient zur Übersichtlichkeit, wenn eine neue Basic-Zeile eingegeben werden soll, da keine anderen Zeilen überschrieben werden brauchen.

CTRL-d Löschen (Delete) einer Zeile mit Nachrücken des unteren Bildschirmteils. Auch dies dient der Übersichtlichkeit. Soll eine Zeile gänzlich aus dem Listing