Aufgewickelt

Die LIST-Funktion des VC 20 hat bekanntlich eine Reihe von Nachteilen. Das hier vorgestellte Programm ermöglicht Endloslistings vorwärts und rückwärts und soll als Anregung zur eigenen Beschäftigung mit Betriebssystemroutinen dienen.

Wie unendlich langwierig kann es doch sein, wenn man im Listing eine Stelle sucht, von der man nur ungefähr weiß, wo sie liegt! Manche Basic-Erweiterungen bieten die Möglichkeit, das LISTing per Cursor hin- und herzuschieben. Das gleich zu beschreibende Programm leistet ähnliches. Mehr noch! Es läßt sich durch leichte Ergänzung des einbettenden Basic-Programms beliebig komfortabel gestalten. Im einzelnen bewirkt das Programm nach Listing 1 folgendes:

(1) Ansprung per GOSUB 500. Nach Drücken der Return-Taste Rücksprung ins Ausgangsprogramm oder, falls der Ansprung vom Direktmodus aus erfolgte, in den Direktmodus. Solange das in den Kassettenpuffer gelegte Maschinenprogramm nicht durch andere Operationen (LOAD, SAVE, VERIFY) zerstört wurde, kann jeder weitere Ansprung per GOSUB 580 vorgenommen werden, was eine Einlesezeit von 2,5 sec einspart.

(2) Das LISTing läuft mit der von LIST her bekannten Geschwindigkeit über den Bildschirm (Anfang zufällig), und zwar endlos: Nach Erscheinen der letzten Basic-Zeile erscheint wieder die erste und so weiter. (Die CTRL- und Run/Stop-Taste haben die üblichen Funktionen, sind hier aber überflüssig.)

(3) Das LISTing stoppt nach Drücken der * -Taste und nimmt seinen Lauf nach abermaligem Drücken der * -Taste wieder auf. Alle weiteren Operationen (einschließlich Rückkehr per Return-Taste) können nur am gestoppten LISTing vorgenommen werden.

(4) Drücken der Cursor-Down-Taste läßt die jeweils nächste Basic-Zeile erscheinen (mit der üblichen Repeat-Funktion der Cursor-Taste und dem Bildschirm-Scrollen nach oben bei vollem Bildschirm).

(5) Drücken der Cursor-Right-Taste sucht diejenige Basic-Zeile auf, die auf die letzte auf dem Bildschirm befindliche Zeile folgt, addiert zur Zeilennummer den Wert 4 x 256 und läßt, angefangen bei der sich so ergebenden Zeile, nach Löschung des Bildschirms, fünf aufeinanderfolgende Basic-Zeilen erscheinen. (Der Wert 4 in 256 kann im Programm nach Listing 1 in Zeile 630 abgeändert werden.) Das Ganze bewirkt also, grob gesagt, ein schnelles Durchsuchen des Listings in Sprüngen von 1024 (oder einem vom Benutzer abgeänderten Vielfachen von 256). Ist die Zeile mit der um den Wert 1024 erhöhten Nummer gar nicht vorhanden, was den Normalfall darstellt, da ja 1024 für das Listing als willkürlich gewählter Wert erscheint, wird die darauffolgende Zeile als erste der fünf auszugebenden Zeilen genommen. Genauer gesagt, werden nur vier ausgegeben, da die Suchschleife in diesem Fall einen Leerlaufschritt macht.

(6) Drücken der Cursor-Up-Taste (Cursor mit Shift) löscht den Bildschirm und läßt, ausgehend von der jeweils letzten Basic-Zeile auf dem Bildschirm (genauer: von der auf diese folgenden Zeile), fünf aufeinanderfolgende Basic-Zeilen erscheinen. (Die Zahl 5 kann im Programm nach Listing 1 in Zeile 770 abgeändert werden.) Jedes weitere Drücken der Cursor-Up-Taste nimmt die zur ersten auf dem Bildschirm befindlichen Basic-Zeile vorhergehende Zeile und läßt fünf aufeinanderfolgende Zeilen erscheinen (mit der üblichen Repeat-Funktion der Cursor-Up-Taste). Bei Erreichen der absolut ersten Zeile des LISTings wird dieser Vorgang mit der absolut letzten Zeile fortgesetzt und so weiter. (Rückwärtsverschieben des LISTings über die Anfangszeile hinaus).

Alle verwendeten Cursor-Tasten behalten ihre Repeat-Funktion bei. Das Rückwärtslisten ist dreimal langsamer als das Vorwärtslisten über die * - oder Cursor-Down-Taste und nur zum Einpendeln gedacht: Überblick mit *, Grobabstimmung mit Cursor right, Feinabstimmung mit Cursor down, Korrektur mit Cursor up, endgültige Feinabstimmung mit Cursor down.

Will man das LISTing mit einer vorgegebenen Zeilennummer Y*256+X beginnen lassen, was wegen der oben erwähnten schnellen Durchsuchungsmöglichkeit in Schritten von 1024 an sich nicht nötig ist, so ersetze man den Aufruf GOSUB 500, beziehungsweise GOSUB 580, durch POKE 780, Y:POKE 781, X:GOSUB 500, beziehungsweise POKE 780, Y: POKE 781, X: GOSUB 580.

Für das Maschinenprogramm wird der Trick verwendet, geeignete Teile des Betriebssystems in den Kassettenpuffer zu kopieren und dort durch einige wenige POKE-Anweisungen abzuändern und miteinander zu verbinden. Es werden keine DATA-Zeilen benötigt, wodurch die bekannten Schwierigkeiten mit dem im VC 20-Basic nicht vorhandenen »RESTOREn« umgangen werden.

Neben der eben beschriebenen langen Form des Programms zum Endloslisten nach Listing 1 schlagen wir in Listing 2 noch eine kurze Form vor. Für diese gelten die oben beschriebenen Punkte 1 bis 5 (* -Taste: Lauf, *-Taste: Stillstand, Cursor-Down-Taste: Zeile für Zeile — endlos, Cursor-Right-Taste: Schrittweite 1024, Return-Taste: Rückkehr). Sowohl das Maschinenprogramm als auch das einbettende Basic-Programm sind wesentlich kürzer. Der Hauptaufand wurde im Programm nach Listing 1 für die Organisation des Rückwärtslaufes benötigt. (Fred Behringer/ev)

```
500 REM  ENLOSLISTING                                  <031>
501 REM (LANGE FASSUNG)                                <084>
502 REM                                                <134>
503 REM                                                <135>
510 POKE 828,134:POKE 829,20:POKE 830,133
   :POKE 831,21                                        <041>
520 FOR I=0 TO 35:POKE 832+I,PEEK(50707+I):NEXT
                    <020>
530 FOR I=0 TO 9:POKE 868+I,PEEK(50952+I):NEXT
                    <250>
540 FOR I=0 TO 8:POKE 879+I,PEEK(50743+I):NEXT
                    <003>
550 FOR I=0 TO 120:POKE 888+I,PEEK(50889+I):NEXT
                    <115>
560 POKE 845,42:POKE 853,34:POKE 865,22
   :POKE 867,20:POKE 875,253:POKE 877,254              <129>
570 POKE 878,136:POKE 887,204:POKE 949,19
   :POKE 963,96                                        <149>
580 GOSUB 780:GET X$:IF X$<>"*"THEN 580                <032>
590 GET X$:IF X$=""THEN 590                            <233>
600 IF X$=CHR$(17)THEN GOSUB 780:GOTO 590              <199>
610 IF X$="*"THEN 580                                  <207>
620 IF X$<>CHR$(29)THEN 640                            <157>
630 POKE 780,PEEK(780)+4:GOSUB 770                     <207>
640 IF X$=CHR$(13)THEN 800                             <246>
650 IF X$<>CHR$(145)THEN 590                           <238>
660 X=PEEK(253):Y=PEEK(254)                            <064>
```

Listing 1. Programm für Endloslisting, »lange Form« (vorwärts und rückwärts). Ansprung per GOSUB 500. Einlesezeit für Maschinenprogramm in den Kassettenpuffer 2,5 sec. Jeder weitere Ansprung per GOSUB 580.

```
670 IF X+256*Y<=PEEK(781)+256*PEEK(780)THEN 730
                                        <205>
680 POKE 780,0:POKE 781,0:GOSUB 770          <110>
690 GET X$:IF X$=""THEN 690                  <078>
700 IF X$<>CHR$(145)THEN 600                 <024>
710 POKE 781,255:POKE 780,249                <002>
720 FOR I=0 TO 3:SYS 828:X=PEEK(253):Y=PEEK(254)
   :POKE 781,X:POKE 780,Y:NEXT               <089>
730 GOSUB 770                                <007>
740 GET X$:IF X$=""THEN 740                  <124>
750 IF X$<>CHR$(145)THEN 600                 <074>
760 POKE 781,X:POKE 780,Y:GOSUB 780:GOTO 660
                        <111>
770 PRINT CHR$(147);:FOR I=1 TO 5:GOSUB 780:NEXT
   :RETURN                                   <055>
780 SYS 828:IF PEEK(782)=1 THEN POKE 781,0
   :POKE 780,0:GOTO 800                      <012>
790 PRINT CHR$(145);                         <159>
800 RETURN                                   <145>
```

Listing 1. Schluß

Zeile	
510-570	Aufbau des Maschinenprogramms im Kassettenpuffer durch Aneinanderreihung von Teilen des Betriebssystems und leichte Ergänzungen und Abänderungen.
510	Übergabe der X/A-Register (dort steht die Nummer der zu listenden Basic-Zeile) an $0014/$0015 (20/21).
520	$C613-$C636 (50707-50742). Startadresse einer Basic-Programmzeile berechnen. Zeilennummer in $0014/$0015 (20/21).
530	$X708-$C711 (50952-50961). Festhalten der jeweils erreichten Zeilennummer in der Suchschleife und Ablegen nach $00FD/$00FE (6253/254. Nach Erreichen der Adresse der gesuchten Zeilennummer aus $0014/$0015 (20/21) steht in $OOFD/$00FE (253/254) die Nummer der vorhergehenden Zeile. Ist die Nummer in $0014/$0015 (20/21) nicht größer als die erste Zeile des zu listenden Programms, dann wird $C708-$C711 nicht wirksam und $00FD/ $00FE (253/254) behält seinen ursprünglichen Wert bei.
540	$C637-$C63F (50743-50751). Weiter in der Suchschleife zur Berechnung der Adresse derjenigen Zeile, deren Nummer in $0014/$0015 (20/21) steht. Die Adresse wird an $005F$/0060 (95/96) ausgegeben Existiert diese Zeile nicht, dann erscheint die Adresse der nächstfolgenden Zeile. In den X5A-Registern steht jeweils die Nummer (Low-Byte, High-Byte) derjenigen Zeile, welche auf die Zeile folgt, deren Adresse in $005F/$0060 (95/96) gelegt wurde. Ist die in $0014/$0015 (20/21) eingegebene Zeilennummer nicht kleiner als die Nummer der größten Zeile des zu listenden Programms, dann werden X/A mit (für unsere Zwecke) unbrauchbaren Werten belegt; jedoch erhält dann (und nur dann) das Y-Register den Wert 1 (siehe unten Zeile 780). Das Basic-Rahmenprogramm erhält die jeweiligen A-, X-, Y-erte über $030C-$030E (780-782).
550	$C6C9-$C741 (50889-51009). Eigentliches Programm zum Auflisten der gewünschten Zeile. Normalerweise Zeilenvorschub, der in 790 (siehe unten) korrigiert wird. Bei Erreichen der letzten Zeile Austritt ohne Zeilenvorschub, was in Zeile 780 (siehe unten) berücksichtigt wird.
560	Abänderungen von Sprungadressen und Korrektur so, daß die Nummer der jeweils vorhergehenden Zeile in $00FD/$00FE (253/254) abgelegt wird (siehe oben unter Zeile 530).
570	Ergänzung um einen DEY-Befehl $0088 (136).
	Abänderung zweier Sprungadressen. Sorge dafür, daß nach LISTen der betreffenden Zeile nicht zum Basic-Warmstart gesprungen wird, sondern an die SYS-Rücksprungadresse des aufrufenden Basic-Programms. Das wird durch Überschreiben des Inhalts von $C714 (50964) (natürlich in der Kassettenpuffer-Kopie) mit dem RTS-Befehl $0060 (96) erreicht.
580	Listen, solange nicht *.
590-600	Sonst: Wenn {Cursor down}, dann eine Zeile listen.
610	Wenn wieder *, dann listen, solange nicht abermals *.
620-630	Wenn {Cursor right·, dann Sprung um 1024.
640	Wenn {Return}, dann zurück.
650	Wenn nicht {Cursor up}, dann Neubeginn der Eingabeschleife.
660	Nummer der vorhergehenden Zeile in X/Y.
670-680	Wenn größer, dann Korrektur für Null und fünf Zeilen listen.
690	Neubeginn der Eingabeschleife.
700-720	Wenn nicht {Cursor up}, dann Korrekturen für Nulldurchgang und fünf Zeilen listen.
740	Abermaliger Beginn der Eingangsschleife.
750-760	Wenn wiederum {Cursor up}, dann Vorgänger der Zeile, deren Nummer in X/Y liegt (Anfangszeile des gerade ausgegebenen Fünferblocks) als Anfangszeile des neuen Fünferblocks nehmen und fünf Zeilen ausgeben.
770	Ausgabe von fünf aufeinanderfolgenden Zeilen.
780-800	Ausgabe einer Zeile
790	und Korrektur für Programmende.

Beschreibung des Programms zum Endloslisten nach Listing 1 (»lange Form«, mit Rückwärtslisten).

```
900 REM  ENDLOSLISTING                       <245>
901 REM (KURZE FASSUNG)                      <016>
902 REM                                      <024>
903 REM                                      <025>
910 POKE 828,134:POKE 829,20:POKE 830,133
   :POKE 831,21                              <187>
920 FOR I=0 TO 45:POKE 832+I,PEEK(50707+I):NEXT
                        <166>
930 FOR I=0 TO 120:POKE 878+I,PEEK(50889+I):NEXT
                        <239>
940 POKE 939,19:POKE 953,96                  <142>
950 GOSUB 1020:GET X$:IF X$<>"*"THEN 950     <184>
960 GET X$:IF X$=""THEN 960                  <093>
970 IF X$=CHR$(17)THEN GOSUB 1020:GOTO 960   <095>
980 IF X$=CHR$(13)THEN 1040                  <120>
990 IF X$="*"THEN 950                        <077>
1000 IF X$<>CHR$(29)THEN 960                 <031>
1010 POKE 780,PEEK(780)+4:PRINT CHR$(147);
   :FOR I=1 TO 5:GOSUB 1020:NEXT:GOTO 960    <017>
1020 SYS 828:IF PEEK(782)=1 THEN POKE 781,0
   :POKE 780,0:GOTO 1040                     <041>
1030 PRINT CHR$(145);                        <144>
1040 RETURN                                  <162>
```

Listing 2. Programm für Endloslisting, »kurze Form« (nicht rückwärts). Ansprung per GOSUB900. Einlesezeit wie in Listing 1. Jeder weitere Ansprung per GOSUB950.

Das Programm Listing 2 ist eine verkürzte Form des Programms nach Listing 1. In Zeile 920 von Listing 2 werden die Zeilen 520 bis 540 von Listing 1 zusammengefaßt. Die Einschubzeile 530 in Listing 1, die die Nummer der jeweils vorhergehenden Zeile zwischenspeichert, wird in Listing 2 nicht benötigt. Die Zeilen 650 bis 760 in Listing 1 organisieren das Rückwärtslisten und werden im Programm nach Listing 2 nicht benötigt.

Beschreibung des Programms zum Endloslisten nach Listing 2 (kurze Form, ohne Rückwärtslisten).