Delete

Diese in Maschinensprache geschriebene Routine ermöglicht es, Basic-Programmzeilen in einem vorzugebenden Zeilennummernbereich zu löschen.

Ein ärgerlicher Nachteil des C 64-Basic und auch von Simons-Basic ist das Fehlen einer Delete-Routine zum schnellen Löschen mehrerer Programmzeilen. Das folgende Programm hilft dem ab.

Der Basic-Lader speichert das Maschinenprogramm im Kassettenpuffer ab Adresse 828 (dez.). Das Programm kann dann auf drei verschiedene Arten aufgerufen werden:

— SYS 828, ZN - ZN Bereich löschen
— SYS 828, - ZN Bis Zeile alles löschen
— SYS 828, ZN - Ab Zeile alles löschen

Es werden jeweils die Zeilen inklusive der angegebenen gelöscht. Da die Routine im Kassettenpuffer untergebracht ist, belegt sie keinen Basic-Speicherplatz.

(Hans-Herbert Hagedorn / ev)

```
10 REM    **********************          <053>
15 REM    *                    *          <242>
20 REM    *       DELETE       *          <170>
25 REM    *                    *          <252>
30 REM    *   H.H.HAGEDORN     *          <053>
35 REM    *                    *          <006>
40 REM    *   RUPPRECHTSTR.30  *          <082>
45 REM    *                    *          <016>
50 REM    *   83 LANDSHUT      *          <227>
55 REM    *                    *          <026>
60 REM    *   TEL. 0871/67337  *          <059>
65 REM    *                    *          <036>
70 REM    **********************          <113>
75 :                                      <133>
80 FOR I=828 TO 990 : READ A : POKE I,A   <061>
85 S=S+A : NEXT                           <084>
90 IF S <> 17132 THEN PRINT "DATENFEHLER" : END
                          <250>
95 PRINT "OK"                             <214>
100 DATA 032,253,174,032,121,000,144,006  <206>
105 DATA 240,004,201,171,208,023,032,107  <210>
110 DATA 169,032,019,166,165,095,133,025  <249>
115 DATA 165,096,133,026,032,121,000,240  <227>
120 DATA 004,201,171,240,005,162,011,076  <226>
125 DATA 058,164,032,115,000,032,107,169  <245>
130 DATA 208,243,165,020,005,021,208,008  <244>
135 DATA 169,255,133,020,133,021,208,006  <253>
140 DATA 230,020,208,002,230,021,032,019  <237>
145 DATA 166,165,095,133,036,165,096,133  <033>
150 DATA 037,056,165,036,229,025,165,037  <033>
155 DATA 229,026,144,201,165,045,229,036  <030>
160 DATA 133,095,165,046,229,037,133,096  <047>
165 DATA 024,165,025,101,095,133,045,165  <033>
170 DATA 026,101,096,133,046,160,000,177  <033>
175 DATA 036,145,025,230,025,208,002,230  <029>
180 DATA 026,230,036,208,002,230,037,056  <039>
185 DATA 165,095,233,001,133,095,165,096  <067>
190 DATA 233,000,133,096,016,225,032,089  <057>
195 DATA 166,032,051,165,076,145,227,000  <064>
200 DATA 000,000,000                      <083>
```

Commodore-Basic erweitert

Mit dem hier vorgestellten Maschinenprogramm wird der Basic-Befehlssatz des VC 20 oder des C 64 um sechs Befehle erweitert.

Die 6 Befehle lauten, in die Basic-Schreibweise übersetzt, GOTO N, GOSUB N, RESTORE N, READ D,A, READ N,D,A, und POP. Einen kleinen Nachteil muß man dabei allerdings in Kauf nehmen, denn diese Routinen kann man nur dem SYS-Befehl ansprechen. Es ist also nicht möglich, eine der Routinen direkt mit einem Basic-Befehlswort aufzurufen.

Der Zugriff auf diese Befehle kann insbesondere dann von großem Nutzen sein, wenn man Programme von anderen Computern umschreiben will, die diese Befehle benutzen.

Will man einen der neuen Befehle in einem Basic-Programm benutzen, muß man nur das Basic-Wort in der oben aufgeführten Liste durch ein »SYS (Adresse)« ersetzen. Die Parameter hinter dem Befehl werden genauso hinter den SYS-Befehl geschrieben, als ob sie hinter dem Basic-Befehl stehen würden.

Bei den nun folgenden Erläuterungen wird davon ausgegangen, daß sich das Maschinenprogramm im Kassettenpuffer ab Adresse 828 befindet. Wurde eine andere Anfangsadresse gewählt, ändern sich auch die Adressen der einzelnen Routinen.

Der Befehl GOTO N sieht in der Form, wie er im Programm verwendet werden muß, so aus: SYS(828)N; also doch noch recht einfach. Dieser Befehl bewirkt, daß man direkt zu einer beliebigen Zeile springen kann, deren Zeilenummer »N« vorher berechnet wurde. Nun zur Syntax. Bei diesem, wie auch bei allen folgenden Befehlen ist darauf zu achten, daß die Startadresse der Routine nach dem SYS-Befehl, (hier 828) in Klammern steht, um Sie von der darauf folgenden Parameterliste zu trennen und so als Adresse kenntlich zu machen. »N« repräsentiert hier, wie auch bei den weiter folgenden Befehlen, eine beliebige gültige numerische Variable, eine Zahl oder einen numerischen Ausdruck. Für »N« ist also beispielsweise auch der Ausdruck »INT(RND(1)*20)*10+100« erlaubt. Der Ausdruck muß nur einen Ganzzahlenwert zum Ergebnis haben. Noch zu bemerken ist, daß zwischen der geschlossenen Klammer der Adresse und der ersten Variablen oder dem Ausdruck kein Komma stehen darf. Das Komma wirkt wie bei PRINT oder READ wie ein Trennzeichen. Da dieser Befehl aber nur eine Variable oder einen Ausdruck enthalten darf, würde das zu einem »SYNTAX ERROR« führen. Dies gilt auch bei allen folgenden Befehlen. Ist die berechnete Zeilennummer nicht im Programm enthalten, erfolgt die Fehlermeldung »UNDEF'D STATEMENT ERROR«.

Für den Befehl GOSUB N gilt das gleiche, was auch zu GOTO N gesagt wurde, unter Berücksichtigung der Tatsache, daß es sich hier um einen Unterprogramm-Aufruf handelt. Mit diesem Befehl kann man also zu einer vorher berechneten Unterprogramm-Adresse springen (SYS(834) N).

RESTORE N ermöglicht es, den DATA-Zeiger auf eine bestimmte Zeile zu setzen. SYS(866)100 beispielsweise setzt den DATA-Zeiger auf das erste Datum der Zeile 100. Mit einem anschließenden READ-Befehl kann man dann gezielt auf die-

sen Datensatz zugreifen. Ist die angegebene Zeilennummer im Programm nicht vorhanden, erfolgt ein »UNDEF'D STATEMENT ERROR«.

READ D, A (SYS(890) d, a) liest direkt den D-ten DATA-Wert in die Variable A. Anstelle von »A« kann sowohl eine numerische als auch eine Stringvariable stehen. »SYS(890) 5, A$« entspricht beispielsweise der Basic-Befehlsfolge »FOR I = 1 TO 5 : READ A$: NEXT«. Auf etwas ist noch zu achten: Will man numerische Daten mit einer numerischen Variable lesen, darf keiner der vorhergehenden DATA-Werte ein String sein. Dies ist programmtechnisch bedingt und liegt daran, daß in Wirklichkeit die entsprechende Anzahl von READ-Befehlen durchgeführt wird. Ein direktes Lesen nur des gesuchten Datums würde das Maschinen-Programm dreimal so lang machen. Sollte es doch einmal vorkommen, daß man versucht, in eine numerische Variable einen String einzulesen, wird ein »SYNTAX ERROR« mit Angabe der entsprechenden DATA-Zeile ausgegeben. Am besten benutzt man immer Stringvariable zum Lesen.

Dann wäre da noch der Befehl »READ N,D,A« (entspricht SYS(927)N,D,A). Dieser Befehl ist eine Mischung des RESTORE- und des READ-Befehls.

SYS(927)N,D,A liest aus der Zeile N das D-te Datum dieser Zeile in die Variable »A«. Ist D größer als die Anzahl der Daten in dieser Zeile, wird in der nächsten DATA-Zeile weitergelesen.

Nun noch zu »POP« (SYS(937)). Springt man aus einem Unterprogramm anstatt mit »RETURN« mit einem direkten Sprungbefehl in die nächsthöhere Ebene (zum Beispiel ins Hauptprogramm) zurück, dann kann mit SYS(937) die letzte gespeicherte Rücksprungadresse im Stack, die dann nicht mehr gebraucht wird, gelöscht werden. Damit wird verhindert, daß der Stack überläuft, da er maximal 23 Rücksprungadressen speichern kann. Außerdem wird ein korrekter Programmablauf sichergestellt, wenn man »hart« aus einem Unterprogramm herausspringt, beispielsweise, um in eine Fehlerbehandlungsroutine zu gehen.

Tips für die Eingabe

Als erstes sollte man nur den Basic-Lader (Listing 1 oder 2, je nach Computer) ab Zeile 10 000 eintippen und dann, ohne ihn zu starten, sicherheitshalber erst mal abspeichern. Dann gibt man noch eine Testzeile ein, und zwar: »10 GOSUB 10 000:PRINT"Prüfsumme=";AS:END« und startet das Ganze mit »RUN«. Ergibt sich für die Prüfsumme ein anderer Wert als 18 413 für den VC 20 oder 17 901 für den C64, dann hat man sich irgendwo vertippt und die Datazeilen sind mit dem abgedruckten Listing noch einmal zu vergleichen. Eine genaue Kontrolle sollte man sowieso vornehmen, da sich durch Zufall eine richtige Prüfsumme ergeben kann, obwohl vielleicht zwei Werte falsch sind, die sich aber gegeneinander aufheben.

Sind alle Datazeilen fehlerfrei, löscht man Zeile 10 und speichert das Programm noch einmal ab, damit man den Basic-Lader an jedes gewünschte Programm anhängen kann.

Ist dies geschehen, kann man das Umrechnungsprogramm (Listing 3) eingeben. Es dient dazu, das Maschinenprogramm aus dem Kassettenpuffer an eine andere Stelle im Speicher zu verschieben. Der Kassettenpuffer hat ja den Nachteil, daß das Maschinenprogramm bei jeder Kassettenoperation zerstört wird. Außerdem können auch andere Befehlserweiterungen diesen Bereich benutzen, um Werte zwischenzuspeichern. Dadurch würde dann das Maschinenprogramm auch zerstört. Werden in einem Programm, das den Basic-Lader enthält, noch weitere DATA-Zeilen verwendet, so ist sicherzustellen, daß deren Zeilennummern größer sind als die höchste Zeilennummer des Basic-Laders, da sonst falsche Werte eingelesen würden.

Und noch ein Tip. Um die Adressen der Befehle nicht ändern zu müssen, wenn man das Maschinenprogramm in einen anderen Speicherbereich verlegt, verwendet man am besten Variablen, denen man am Anfang des Programms die Adresse zuweist. Dies könnte zum Beispiel so aussehen: »SM%=828«.Für die einzelnen Befehle würde dann folgendes gelten:

GOTO N	=	SYS(SM%)
GOSUB N	=	SYS(SM%+6)
RESTORE N	=	SYS(SM%+38)
READ D,A	=	SYS(SM%+62)
READ N,D,A	=	SYS(SM%+99)
POP	=	SYS(SM%+109)

Da alle diese Routinen weitgehend in das Betriebssystem des Computers eingebunden sind, werden bei Fehlern in der Ausführung die entsprechenden Systemfehlermeldungen ausgegeben.

(Wolfgang Thauer / ev)

```
1 REM BASIC-LADER VC 20                                          <130>
2 REM                                                            <145>
3 REM WOLFGANG THAUER                                            <176>
4 REM AM SCHIEDSBERG 45                                          <173>
5 REM 5205 ST.AUGUSTIN 2                                         <215>
6 REM                                                            <149>
7 REM                                                            <150>
10000 AS=0:FOR I=828+AB TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN                             <126>
10005 DATA 32                                                    <036>
10010 DATA 198,3:REM**                                           <226>
10020 DATA 76,163,200,169,3,32,251,195,165,123,
   72,165,122,72,165,58,72                                       <061>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                    <002>
10040 DATA 198,3:REM**                                           <000>
10050 DATA 32,163,200,76,174,199,32                              <150>
10060 DATA 198,3:REM**                                           <020>
10070 DATA 32,19,198,176,3,76,227,200,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32                        <247>
10080 DATA 29,200,32                                             <196>
10090 DATA 198,3:REM**                                           <050>
10100 DATA 165,20,133,253,166,21,232,134,254,32,
   253,206,32,6,204                                              <071>
10110 DATA 198,253,240,11,32,28,204,198,253,208,
   249,198,254,208,245,96,32                                     <060>
10120 DATA 98,3 :REM**                                           <031>
10130 DATA 32,253,206,32                                         <188>
10140 DATA 125,3:REM**                                           <090>
10150 DATA 96,169,255,133,74,32                                  <062>
10170 DATA 192,3:REM**                                           <124>
10180 DATA 154,201,141,240,3,76,224,200,104
                    <134>
10190 DATA 104,104,104,104,76,248,200,186,232,
   232,76,139,195,32,138,205,32,247,215,96                       <196>
```

Listing 1. Basic-Lader »6 neue Befehle« (VC 20-Version)

```
1 REM BASIC LADER C 64                                           <007>
2 REM                                                            <145>
3 REM WOLFGANG THAUER                                            <176>
4 REM AM SCHIEDSBERG 45                                          <173>
5 REM 5205 ST.AUGUSTIN 2                                         <215>
6 REM                                                            <149>
7 REM                                                            <150>
10000 AS=0:FOR I=828 TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN                             <081>
10005 DATA 32                                                    <036>
10010 DATA 198,3:REM**                                           <226>
10020 DATA 76,163,168,169,3,32,251,163,165,123,
   72,165,122,72,165,58,72                                       <069>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                    <002>
10040 DATA 198,3:REM**                                           <000>
10050 DATA 32,163,168,76,174,167,32                              <158>
10060 DATA 198,3:REM**                                           <020>
10070 DATA 32,19,166,176,3,76,227,168,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32                        <255>
10080 DATA 29,168,32                                             <209>
10090 DATA 198,3:REM**                                           <050>
10100 DATA 165,20,133,253,166,21,232,134,254,32,
```

```
   253,174,32,6,172                                    <079>
10110 DATA 198,253,240,11,32,28,172,198,253,208,
   249,198,254,208,245,96,32                           <064>
10120 DATA 98,3:REM**                                  <031>
10130 DATA 32,253,174,32                               <192>
10140 DATA 125,3:REM**                                 <090>
10150 DATA 96,169,255,133,74,32                        <062>
10160 DATA 192,3:REM**                                 <114>
10170 DATA 154,201,141,240,3,76,224,168,104
                                                       <137>
10180 DATA 104,104,104,104,76,248,168,186,232,
   232,76,139,163,32,138,173,32,247,183,96             <202>
```

Listing 2. Basic-Lader »6 neue Befehle« (C 64-Version)

```
1 REM ADRESSEN UMRECHNEN                               <138>
2 REM                                                  <145>
3 REM WOLFGANG THAUER                                  <176>
4 REM AM SCHIEDSBERG 45                                <173>
5 REM 5205 ST.AUGUSTIN 2                               <215>
6 REM                                                  <149>
10 :                                                   <068>
20 REM DAS MACHINENPROGRAMM IST 145 BYTES LANG
                                                       <086>
30 :                                                   <088>
40 REM ZEICHERERKLAERUNG                               <177>
50 REM CHR$(147)= CLEAR HOME                           <124>
60 REM CHR$(17) = CURSOR DOWN                          <216>
70 REM CHR$(18) = REVERSE ON                           <134>
80 REM CHR$(13)  = RETURN                              <178>
90 :                                                   <148>
100 :                                                  <158>
110 :                                                  <168>
120 PRINT CHR$(147)                                    <197>
130 PRINT"DIESES[SPACE]PROGRAMM[SPACE]RECHNET
   [SPACE]DIE[SPACE]ABSOLUTEN[SPACE]ADRESSEN
   [SPACE]IN[SPACE]DER[SPACE]MASCHINENROU";
                                                       <151>
135 PRINT"TINE[SPACE]FUER[SPACE]EINEN[SPACE]
   ANDEREN[SPACE]SPEICHERBEREICH[SPACE]UM[SPACE]
   UND[SPACE]POKET[SPACE]ES";                          <086>
140 PRINT"LAUFFAEHIG[SPACE]IN[SPACE]DIESEN
   [SPACE]SPEICHERBEREICH."                            <247>
150 PRINT CHR$(17);"GEBE[SPACE]DIE[SPACE]
   GEWUENSCHTE[SPACE2]ANFANGSADRESSE[SPACE]EIN,
   AB"                                                 <229>
160 PRINT"DER[SPACE]DAS[SPACE]MASCHINENPRO-
   [SPACE]GRAMM[SPACE]IM[SPACE]SPEICHER[SPACE5]
   LIEGEN[SPACE]SOLL."                                 <093>
170 PRINT CHR$(17);"(DIE[SPACE]NORMALE[SPACE]
   ADRESSE[SPACE3]IST[SPACE]828[SPACE]=[SPACE]
   KASSETTEN-[SPACE2]PUFFER)."                         <030>
180 PRINT CHR$(17):INPUT AA                            <014>
190 PRINT CHR$(17);"DIE[SPACE]ANFANGSADRESSE
   [SPACE4]IST[SPACE]";AA                              <218>
200 PRINT CHR$(17);"IST[SPACE]DAS[SPACE]RICHTIG
   [SPACE]?"                                           <113>
210 INPUT"(J/N)";A$                                    <083>
220 IF A$<>"J"AND A$<>"N"THEN 210                      <002>
230 IF A$="N"THEN PRINT:GOTO 150                       <179>
240 AB=AA-828                                          <244>
245 PRINT CHR$(147);"BITTE[SPACE]WARTEN"               <010>
250 GOSUB 700                                          <030>
260 IF AA=828 THEN 490                                 <170>
270 :                                                  <073>
280 REM ANGABE DER ZU AENDERNDEN DATAZEILEN
                                                       <133>
290 REM DIESE DATAZEILEN SIND IM LISTING DURCH
   EIN ANGEHAENGTES    ':REM**'GEKENNZEICHNET
                                                       <105>
300 :                                                  <103>
310 PRINT CHR$(17);"DIE[SPACE]ZU[SPACE]
   AENDERNDEN[SPACE]DATAZEILEN[SPACE]MIT[SPACE]
   DEN[SPACE]ENT-[SPACE3]SPRECHENDEN[SPACE]
   NEUEN"                                              <132>
320 PRINT"WERTEN[SPACE]LAUTEN:"                        <246>
330 PRINT                                              <228>
340 AD=AA+138:GOSUB 630                                <180>
350 PRINT"10010[SPACE]DATA";AL;",";AH                  <127>
360 PRINT"10040[SPACE]DATA";AL;",";AH                  <140>
370 PRINT"10060[SPACE]DATA";AL;",";AH                  <152>
380 PRINT"10090[SPACE]DATA";AL;",";AH                  <165>
390 AD=AA+38:GOSUB 630                                 <181>
400 PRINT"10120[SPACE]DATA";AL;",";AH                  <179>
410 AD=AA+65:GOSUB 630                                 <201>
420 PRINT"10140[SPACE]DATA";AL;",";AH                  <201>
430 AD=AA+132:GOSUB 630                                <008>
440 PRINT"10170[SPACE]DATA";AL;",";AH                  <224>
450 PRINT CHR$(17);"WEITER[SPACE]MIT[SPACE]
   <RETURN>"                                           <111>
460 GET A$:IF A$=""THEN 460                            <052>
470 IF A$<>CHR$(13)THEN 460                            <232>
480 :                                                  <027>
490 PRINT                                              <132>
500 PRINT"DIE[SPACE]STARTADRESSEN[SPACE]DER
   [SPACE]ROUTINEN[SPACE]LAUTEN:"                      <217>
510 PRINT CHR$(17);"GOTO[SPACE]N[SPACE2]=[SPACE]
   SYS(";AA;")";CHR$(17)                               <162>
520 PRINT"GOSUB[SPACE]N[SPACE]=[SPACE]SYS(";
   AA+6;")";CHR$(17)                                   <025>
530 PRINT"RESTORE[SPACE]N=SYS(";AA+38;")";
   CHR$(17)                                            <252>
540 PRINT"RESTORE[SPACE]D,A$"
   :PRINT"[SPACE8]=[SPACE]SYS(";AA+62;")";
   CHR$(17)                                            <161>
550 PRINT"RESTORE[SPACE]N,D,A$"
   :PRINT"[SPACE8]=[SPACE]SYS(";AA+99;")";
   CHR$(17)                                            <047>
560 PRINT"POP[SPACE5]=[SPACE]SYS(";AA+109;")"
                                                       <011>
570 END                                                <188>
580 :                                                  <128>
590 :                                                  <138>
600 REM SUBROUTINE                                     <249>
610 REM BERECHNUNG VON LOW- UND HIGH-BYTE EINER
   ADRESSE                                             <200>
620 :                                                  <168>
630 AH=INT(AD/256):AL=AD-AH*256:RETURN                 <203>
640 :                                                  <188>
650 :                                                  <198>
670 REM SUBROUTINE                                     <063>
680 REM MASCHINENPROGRAMM WIRD IN SPEICHER GEPO
   KET                                                 <099>
690 :                                                  <238>
700 GOSUB 10000                                        <060>
710 IF AS=18413 THEN PRINT CHR$(17);"PRUEFSUMME
   [SPACE]KORREKT.":RETURN                             <250>
712 REM  18413 = PRUEFSUMME FUER VC 20                 <205>
714 REM FUER C 64 IST DIE PRUEFSUMME 17901             <007>
720 PRINT CHR$(147);"PRUEFSUMME[SPACE]=";AS
                                                       <179>
730 PRINT CHR$(17);"PRUEFSUMMENFEHLER[SPACE]!"
                                                       <162>
740 PRINT                                              <127>
750 PRINT"MOEGLICHE[SPACE]FEHLER-[SPACE5]
   QUELLEN[SPACE]:"                                    <141>
760 PRINT                                              <147>
770 PRINT"*EINE[SPACE]DATE[SPACE]WURDE[SPACE]
   VER-[SPACE2]GESSEN[SPACE]ODER[SPACE]FALSCH
   [SPACE4]EINGE-TIPPT."                               <224>
780 PRINT                                              <168>
790 PRINT"*SIE[SPACE]HABEN[SPACE]DIE[SPACE2]
   DATA-[SPACE2]ZEILEN[SPACE]GEAENDERT,[SPACE]
   UM[SPACE2]DAS[SPACE]MASCHINENPROGRAMM";             <170>
800 PRINT"[SPACE]AN[SPACE]EINEN[SPACE]ANDEREN
   [SPACE6]SPEICHERBEREICH[SPACE]ANZU-[SPACE]
   PASSEN."                                            <131>
810 PRINT"*SIE[SPACE]HABEN[SPACE]DIE[SPACE]
   FALSCHE[SPACE]PRUEFSUMME[SPACE]IN[SPACE]
   ZEILE[SPACE3]710[SPACE]EINGESETZT."                 <028>
820 END                                                <183>
830 :                                                  <123>
960 :                                                  <253>
970 REM UNTERPROGRAMM                                  <079>
980 REM BASICLADER                                     <048>
990 :                                                  <027>

10000 AS=0:FOR I=828+AB TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN                   <126>
10005 DATA 32                                          <036>
10010 DATA 198,3:REM**                                 <226>
10020 DATA 76,163,200,169,3,32,251,195,165,123,
   72,165,122,72,165,58,72                             <061>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                                                       <002>
10040 DATA 198,3:REM**                                 <000>
10050 DATA 32,163,200,76,174,199,32                    <150>
10060 DATA 198,3:REM**                                 <020>
10070 DATA 32,19,198,176,3,76,227,200,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32              <247>
10080 DATA 29,200,32                                   <196>
```

```
10090 DATA 198,3:REM**                                  <050>
10100 DATA 165,20,133,253,166,21,232,134,254,32,
   253,206,32,6,204                                     <071>
10110 DATA 198,253,240,11,32,28,204,198,253,208,
   249,198,254,208,245,96,32                            <060>
10120 DATA 98,3 :REM**                                  <031>
10130 DATA 32,253,206,32                                <188>
10140 DATA 125,3:REM**                                  <090>
10150 DATA 96,169,255,133,74,32                         <062>
10170 DATA 192,3:REM**                                  <124>
10180 DATA 154,201,141,240,3,76,224,200,104
                        <134>
10190 DATA 104,104,104,104,76,248,200,186,232,
   232,76,139,195,32,138,205,32,247,215,96 <196>
```

Listing 3. Der Adressen-Umrechner

Hardcopy im Superformat

Hier ist endlich eine formatfüllende Hardcopy-Routine für den Commodore 64 mit Epson-Drucker und Simons Basic

Das Programm erzeugt eine vergrößerte Hardcopy vom Grafikbildschirm. Die in Simons Basic enthaltene Hardcopy nutzt ja leider nur das halbe Blatt aus, was in vielen Fällen sehr störend ist. Das Programm vergrößert den Ausdruck in X-und Y-Richtung, so daß eine halbe A4-Seite bedruckt wird.

Es ist für den Commodore 64 und Epson-Drucker mit VC-Interface gedacht, läßt sich aber auch an andere Nadeldrucker anpassen.

Für das Görlitz-Interface muß der Wert 1 in Zeile 320 in 12 geändert werden. (Peter Schwabe/ev)

Ein Hardcopy-Beispiel (Originalgröße)