Tips und Tricks für Einsteiger

Es ist für einen Einsteiger anfangs immer schwer, aus seinem Computer die optimale Leistung herauszuholen. Die hier vorgestellten Routinen und Programme sollen Ihnen helfen, mit Ihrem Computer besser »auszukommen«.

Wir haben ein paar Leckerbissen für Sie herausgepickt: neben einem AUTO-Number, Listschutzmethoden und einem besseren INPUT-Befehl finden Sie unter anderem auch noch einen OLD-Befehl und einen Peripherietest.

Hilfe nach NEW oder RESET

Es kann schon mal vorkommen, daß man NEW eingibt und danach erschrocken feststellt: es wurde vergessen, das Programm zu speichern. Normalerweise sind diese Daten dann verloren. Mit Hilfe des Programms »Unnew« (Listing 1) kann jedoch das alte Programm wieder gerettet werden. Dazu müssen Sie jedoch das Hilfsprogramm schon vor dem NEW auf einer Diskette (oder Kassette) parat haben. Geben Sie bitte Listing 1 ein und legen eine formatierte Diskette (oder Kassette) in Ihr Laufwerk. Nach anschließendem Start mit RUN wird daraus ein Maschinenprogramm erzeugt und selbständig auf Ihren Datenträger gespeichert.

Ist Ihnen nun ein Mißgeschick passiert (NEW eingegeben?), so legen Sie die Diskette mit dem Unnew-Programm ein, laden dieses mit

```
LOAD"UNNEW",8,1
```

und starten es durch

```
SYS 525
```

Ihr Basic-Programm wird damit gerettet, selbst wenn Sie einen Reset ausgelöst haben sollten.

(Daniel Kossmann/dm)

Directory ohne Programmverlust

Gelegentlich möchte man das Directory einer Diskette ansehen, ohne ein eventuell im Speicher befindliches Programm zu zerstören. Befindet sich das DOS 5.1 (Erleichtert den Umgang mit der Floppy – zu finden auf Ihrer Test-/Demodiskette und im Artikel in diesem Heft) gerade nicht im Speicher, so behilft man sich meist mit der zeitaufwendigen Zwischenspeicherung des Programms auf Diskette. Es geht jedoch auch anders. Geben Sie einfach den folgenden Befehl im Direktmodus ein:

```
POKE44,PEEK(46)+1
```

Damit wird der Basic-Anfang auf einen freien Speicherbereich gesetzt. Sie können jetzt wie gewohnt mit

```
LOAD"$",8
```

das Directory laden und anschließend LISTen. Mit

```
POKE44,8
```

gelangen Sie dann wieder zurück in das eigentliche Programm.

Eine zweite, komfortablere Methode ist, das folgende kleine Programm (Listing 2) in sein eigenes Programm einzubauen und je nach Bedarf dieses dann auszuführen (zum Beispiel als Unterroutine, die mittels GOSUB aufgerufen werden kann – dabei muß das END in Zeile 20 durch ein RETURN ersetzt werden).

(Heinzpeter Oelkers/dm)

```
10 OPEN 1,8,0,"$":GET#1,A$,A$                          <124>
20 GET#1,A$,A$:IF ST=64 THEN CLOSE 1:END               <007>
30 GET#1,A$,B$:PRINT ASC(A$+CHR$(0))+256*A
   SC(B$+CHR$(0));                                     <078>
40 GET#1,A$:PRINT A$;:IF A$<>""THEN 40                 <109>
50 PRINT:GOTO 20                                       <083>
```

Listing 2. Das Directory ausgeben, ohne ein Programm zu zerstören

Mehr Diskettenkapazität

So können Sie die Kapazität Ihrer Diskette verdoppeln und Ihre Kosten halbieren: Sie müssen nur auf der anderen Seite noch mal genau die gleiche Schreibschutzöffnung einschneiden. Das geht am besten mit einem Papierlocher (siehe Bild 1). (dm)

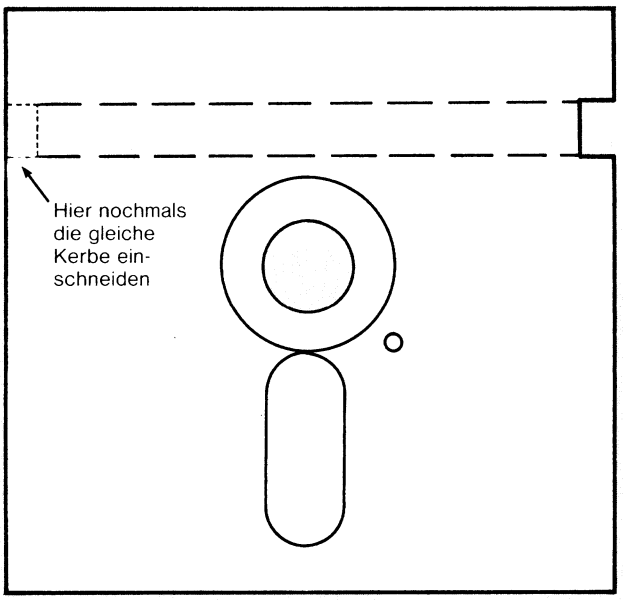


Bild 1. Verdoppeln Sie Ihre Diskettenkapazität

Listschutz für Basic-Programme

Ein Listschutz dient dazu, sein Programm und seinen Programmierstil vor fremden Augen zu schützen.

```
100 FOR I=525 TO 578                                   <086>
110 READ A:POKE I,A:NEXT I                             <156>
120 POKE 43,525 AND 255:POKE 44,2                      <135>
130 POKE 45,578 AND 255:POKE 46,2                      <056>
140 CLR:SAVE"UNNEW",8,1                                <138>
150 REM FUER CASSETTENBETRIEB:                         <117>
160 REM CLR:SAVE"UNNEW",1,1                            <095>
170 DATA 160,3,200,177,43,208                          <103>
180 DATA 251,200,200,152,160,0                         <203>
190 DATA 145,43,165,44,200,145                         <194>
200 DATA 43,133,60,160,0,132                           <195>
210 DATA 59,162,0,200,208,2,230                        <206>
220 DATA 60,177,59,208,245,232                         <092>
230 DATA 224,3,208,242,200,208                         <018>
240 DATA 2,230,60,132,45,164,60                        <101>
250 DATA 132,46,96,255                                 <065>
```

Listing 1. Ein Programm, das Ihnen bei NEW oder RESET hilft

Möchte man ein Basic-Programm mit einem einfachen Listschutz versehen, so verfährt man folgendermaßen:
1. Man ergänzt die Zeile, ab der der Listschutz wirksam werden soll, mit »:REM" "«.
2. Man fährt mit dem Cursor auf das zweite Anführungszeichen und drückt fünfmal die Tastenkombination SHIFT-INST.
3. Nun wird ebenfalls fünfmal die Taste DEL gedrückt, so daß zwischen den Anführungszeichen fünf reverse T stehen.
4. Zuletzt bewegt man den Cursor hinter das zweite Anführungszeichen und drückt die Tastenkombination SHIFT und L. Anschließend RETURN nicht vergessen.

Versucht man nun, das Programm zu listen, gelangt der Computer nur bis zu der Zeile, in der der Listschutz steht und bricht den Vorgang dann mit der Meldung SYNTAX ERROR ab.

Dieser Listschutz ist allerdings relativ leicht wieder aufzuheben.

(Thomas Lopatic/dm)

Ein geänderter Zeichensatz

Mit Hilfe dieses Programmes kann man auf ganz einfache Art einen neuen Zeichensatz (etwas dünner als der gewohnte) erstellen. Nach einer kleinen Wartezeit, in der dieser aufgebaut wird, können Sie über ein neues Schriftbild verfügen.

```
10 R=56334:POKE 53272,24:POKE R,0
20 POKE 1,51:FOR I=6?5 TO R/5
30 POKE I;PEEK(I+45056) AND 60
40 NEXT:POKE 1,7:POKE R,1
```

(dm)

Negative Bytes?

Die FRE-Funktion liefert beim C64 gelegentlich negative Werte (negative Bytes?) Dies ist jedoch nur ein kleiner Fehler im Betriebssystem des Computers. Abhilfe schafft hierbei, zu der negativen Zahl den Wert 65535 zu addieren, also PRINT FRE(0)+65535. (dm)

Sequentielle Datei retten

Wenn auch nicht oft, so kann es doch gelegentlich einmal vorkommen, daß beim Schreiben einer sequentiellen Datei auf Diskette etwas passiert. Sei es, daß das Programm abstürzt oder durch einen Programmier- oder Bedienungsfehler die Datei nicht richtig geschlossen wurde. Durch geschickte Ausnutzung des COPY-Befehles läßt sich die Datei aber noch retten. Man muß nur im Direktmodus folgendes eingeben:

```
D$="NAME"   (Name der verlorenen Datei)
OPEN1,8,15
PRINT#1,"C:X="+D$
PRINT#1,"S:"+D$
PRINT#1,"R:"+D$+"=X"
CLOSE1
```

(Erwin Fröhlich/dm)

Kassettendirectory

Eine einfache, wenn auch sehr langwierige Methode, sich ein Kassettendirectory zu erstellen, ist folgende (ein Drucker wird vorausgesetzt): Geben Sie diese Direktbefehle über die Tastatur ein:

```
OPEN1,4:CMD1:LOAD"QAY"
```

oder einen anderen, wirren Namen, der bestimmt nicht auf der Kassette zu finden ist. Nach dieser Eingabe muß die Space-Taste festgeklemmt (etwa mit einem Stück Tesafilm) und der Recorder gestartet werden. Dieser läßt die Kassette durchlaufen und gibt jedesmal, wenn er etwas findet, ein FOUND PROGRAMMNAME auf dem Drucker aus. Dieses Verfahren funktioniert auch mit Beschleunigern wie »Turbo Tape« oder »Super Tape«. (dm)

Restore für Unterprogramme

Wenn beide Programmteile, Hauptprogramm und Unterprogramm, DATAs enthalten, muß sichergestellt werden, daß auch die richtigen Werte gelesen werden. Wenn man nicht aufpaßt, kann es passieren, daß das Unterprogramm DATAs aus dem Hauptprogramm liest. Wie läßt sich das verhindern? Es gibt eine umständliche Methode: Man kann eine kleine Basic-Erweiterung einbauen, den RESTORE X-Befehl. Es geht aber auch einfacher. Die Zeropage, das sind die ersten 256 Bytes des Speichers, hilft uns bei der Lösung des Problems. Genauer gesagt, die Adressen 65/66 und 122/123. Schlagen wir im Commodore-Handbuch nach, dann steht dort:

65 bis 66 Adresse des aktuellen DATA-Elementes
122 bis 123 Basic-Zeiger innerhalb der Subroutine

Mit diesen Informationen läßt sich schon etwas anfangen. Wenn das Unterprogramm angesprungen wird, dann sollte der Zeiger in Speicherstelle 122/123 auf die Adresse des Unterprogramms im Speicher stehen. POKEt man diese Werte in die Zeilen 65/66 mit

```
POKE 65,PEEK(122)
POKE 66,PEEK(123)
```

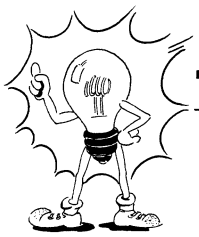
so wird beim nächsten READ der Wert gelesen, der hinter dieser Basic-Zeile mit den POKEs steht, also das erste DATA-Element innerhalb des Unterprogrammes. Nach dem Rücksprung aus dem Unterprogramm muß der Zeiger eventuell auch im Hauptprogramm wieder gestellt werden.

In dem kurzen Demo-Listing (Listing 3) werden drei Unterprogramme in zufälliger Reihenfolge aufgerufen.

(Stephan Pätzold/dm)

```
1 REM ***********************                <058>
2 REM *        DEMO         *                <161>
3 REM * SUBROUTINE/RESTORE *                <075>
4 REM ***********************                <061>
5 PRINT"{CLR,6SPACE}TASTE DRUECKEN!"         <161>
6 PRINT:PRINT                                <214>
10 X=INT(RND(TI)*3)+1                        <235>
20 ON X GOSUB 1000,2000,3000                 <012>
25 POKE 65,PEEK(122):POKE 66,PEEK(123)       <084>
30 READ A$:PRINT A$                          <249>
50 DATA"HAUPTPROGRAMM"                       <118>
100 GOTO 10                                  <022>
999 :                                        <213>
1000 REM *** SUBROUTINE 1 ***                <208>
1005 :                                       <219>
1010 POKE 65,PEEK(122):POKE 66,PEEK(123)     <051>
1020 FOR I=1 TO 4:READ A:PRINT A;:NEXT       <051>
1030 READ A$:PRINT A$;                       <095>
1040 POKE 198,0:WAIT 198,1                   <252>
1050 DATA 1,11,111,1111,"UP 1 "              <118>
1060 RETURN                                  <102>
1070 :                                       <030>
2000 REM *** SUBROUTINE 2 ***                <193>
2005 :                                       <203>
2010 POKE 65,PEEK(122):POKE 66,PEEK(123)     <035>
2020 FOR I=1 TO 4:READ A:PRINT A;:NEXT       <035>
2030 READ A$:PRINT A$;                       <077>
2040 POKE 198,0:WAIT 198,1                   <234>
2050 DATA 2,22,222,2222,"UP 2 "              <214>
2060 RETURN                                  <086>
2070 :                                       <014>
3000 REM *** SUBROUTINE 3 ***                <178>
3005 :                                       <187>
3010 POKE 65,PEEK(122):POKE 66,PEEK(123)     <019>
3020 FOR I=1 TO 4:READ A:PRINT A;:NEXT       <019>
3030 READ A$:PRINT A$;                       <061>
3040 POKE 198,0:WAIT 198,1                   <218>
3050 DATA 3,33,333,3333,"UP 3 "              <051>
3060 RETURN                                  <068>
3070 :                                       <252>
```

Listing 3. Ein Demo-Programm zur Erklärung des »RESTORE X«



Fehler im Floppy-DOS

Wenn Sie schon einmal versucht haben, ein Programm mit der REPLACE-Funktion
(SAVE"@:NAME",8)
zu speichern und das Programm dann zerstört war, so liegt es an einem Fehler im Floppy-Betriebssystem, der allerdings sehr selten auftritt. Abhilfe schafft hier nur die Methode, das alte Programm erst zu löschen und dann das Neue zu speichern. (dm)

Neues INPUT

Haben Sie schon einmal versucht, professionelle Programme zu schreiben? Dann hat Ihnen mit Sicherheit die INPUT-Routine des C64-Betriebssystems zu schaffen gemacht, die sämtliche Zeichen zuläßt, auch solche, die zu völligem Unsinn oder Fehlermeldungen führen. Dieser Mißstand wird in Basic-Programmen meist mit einer GET-Schleife umgangen. Bei dieser Methode wird ständig ein Zeichen von der Tastatur geholt und auf Zulässigkeit überprüft. Die einzelnen Zeichen werden in einem String addiert. Wenn Sie mit INS oder DEL Korrekturen ausführen wollen, können Sie sich vorstellen, wie kompliziert und zeitraubend eine solche Routine werden kann.

Eine schnelle Lösung des Problems schafft dieses Maschinenprogramm (Listing 4), das einen neuen INPUT-Befehl generiert. Die Syntax des Befehls lautet:
INPUT > Eingabelänge, Spalte, Zeile, Variable

Wie Sie schon an der Syntax sehen, ein recht komfortabler INPUT-Befehl. Mit der Eingabelänge geben Sie an, wie lang das Eingabefeld höchstens sein darf. Durch Angabe von Zeile und Spalte können Sie das Eingabefeld direkt an jeder beliebigen Bildschirmstelle positionieren. Als Variable müssen Sie eine Stringvariable einsetzen, zum Beispiel A$. Sollen ausschließlich Zahlen eingegeben werden, braucht der String nur noch auf Buchstaben hin überprüft werden. Beispielsweise mit folgender Routine:

```
10 INPUT>4,1,1,A$
20 IF STR$(VAL(A$))=" "+A$ THEN PRINT"ES IST EINE
ZAHL":END
30 PRINT"KEINE ZAHL":GOTO10
```

Tips zur Eingabe

Das Programm (Listing 4) müssen Sie mit dem MSE eingeben und speichern. Den MSE finden Sie auf Seite 136 in dieser Ausgabe. Laden Sie dann die neue Routine mit
LOAD"E-ROUTINE 64",8,1 (für die Floppy)
oder
LOAD"E-ROUTINE 64",1,1 (für Datasette)
und geben Sie
NEW
und
SYS 49152
zur Initialisierung ein. Danach können Sie Ihre Basic-Programme eintippen und mit dem neuen INPUT-Befehl verwenden. Die Routine können Sie natürlich auch mit dem Basic-Programm nachladen und mit dem neuen INPUT-Befehl verwenden.

Zusammenfassung

- Die Routine wird mit
SYS 49152
aktiviert.
- Das Programm benötigt den Speicherbereich von Adresse 49152 bis 49617
- Die Syntax lautet:
INPUT > Eingabelänge, Zeile, Spalte, Stringvariable
- Die CLR-; HOME-; RUN-; STOP- und SHIFT-Taste ist bei der Eingabe gesperrt.
- Das Eingabefeld kann mit den Cursortasten nicht verlassen werden.
- INST/DEL wirkt nur innerhalb des Eingabefeldes.
- Leerzeichen am Ende der Eingabe werden nicht mit übernommen.
- Benötigen Sie Leerzeichen am Schluß, geben Sie einfach
POKE 49298,131
ein.
POKE 49298,94
schaltet wieder in den Normalzustand.

Ein Nachteil soll nicht verschwiegen werden: Fragetext kann nicht wie beim serienmäßigen INPUT-Befehl ausgegeben werden. Der Text muß mit PRINT gedruckt werden. Das bedeutet, Sie müssen erst den Text mittels PRINT ausgeben und dann quasi den INPUT »drüberlegen«.

Eine komfortablere Routine finden Sie neben anderen Befehlen in der Basic-Erweiterung »Datawork-Basic«, erschienen in der 64'er, Ausgabe 1/86.

(Thomas Graf/dm)

```
 programm : e-routine 64 c000 c1f4
-----------------------------------

c000 : a9 0b a0 c0 8d 08 03 8c  ad
c008 : 09 03 60 20 73 00 c9 85  18
c010 : f0 06 20 79 00 4c e7 a7  8c
c018 : 20 73 00 c9 b1 f0 06 20  26
c020 : bf ab 4c ae a7 a9 00 8d  81
c028 : 6e c1 20 9b b7 8e 6f c1  23
c030 : 20 fd ae 20 9e b7 e0 28  7a
c038 : b0 0c 86 fd 20 fd ae 20  3d
c040 : 9e b7 e0 19 90 03 4c 48  f8
c048 : b2 8e 70 c1 a4 fd 20 0c  68
c050 : e5 20 24 ea a5 d2 85 fe  b1
c058 : 98 18 65 d1 90 02 e6 fe  43
c060 : 85 fd a5 f4 85 fc 98 18  bf
c068 : 65 f3 90 02 e6 fc 85 fb  90
c070 : a5 c6 85 cc 8d 92 02 f0  cb
c078 : f7 78 a5 cf f0 0c a5 ce  b2
c080 : ae 87 02 a0 00 84 cf 20  2a
c088 : 13 ea 20 b4 e5 c9 0d d0  31
c090 : 03 4c 71 c1 c9 14 f0 46  db
c098 : c9 9d f0 42 c9 1d f0 29  50
c0a0 : c9 94 f0 61 c9 91 d0 03  8e
c0a8 : 4c 41 c1 c9 11 d0 03 4c  7b
c0b0 : 54 c1 ae 18 d0 e0 15 f0  de
c0b8 : 08 c9 db b0 b3 c9 c1 b0  a4
c0c0 : 08 c9 60 b0 ab c9 20 90  86
c0c8 : a7 c9 22 f0 a3 ae 6e c1  e7
c0d0 : ec 6f c1 f0 9b ee 6e c1  71
c0d8 : 20 16 e7 4c 70 c0 ac 6e  23
c0e0 : c1 f0 8d ce 6e c1 c9 9d  ae
c0e8 : f0 ee b1 fb aa b1 fd 88  7c
c0f0 : 91 fd 8a 91 fb c8 c8 f0  60
c0f8 : 07 cc 6f c1 90 ec f0 ea  83
c100 : a9 9d 4c d8 c0 ac 6f c1  59
c108 : ce 6f c1 88 c0 ff f0 2b  35
c110 : cc 6e c1 90 26 b1 fd c9  11
c118 : 20 f0 f0 cc 6f c1 f0 1b  85
c120 : b1 fb aa b1 fd c8 91 fd  18
c128 : 8a 91 fb 88 88 c0 ff f0  fb
c130 : 05 cc 6e c1 b0 ea c8 a9  48
c138 : 20 91 fd ee 6f c1 4c 70  95
c140 : c0 aa 38 ad 6e c1 e9 28  06
c148 : 90 07 8d 6e c1 8a 4c d8  e0
c150 : c0 4c 70 c0 aa 18 ad 6e  69
c158 : c1 69 28 b0 07 cd 6f c1  0e
c160 : f0 05 90 03 4c 70 c0 8d  be
c168 : 6e c1 8a 4c d8 c0 ea ea  f8
c170 : ea 4c 96 c1 88 c0 ff d0  8e
c178 : 13 20 73 00 c9 00 f0 07  e7
c180 : c9 3a f0 03 4c 79 c1 a0  dc
c188 : 00 4c eb c1 b1 fd c9 20  54
c190 : f0 e2 c8 8c 6f c1 20 fd  36
c198 : ae 20 8b b0 48 98 48 20  fa
c1a0 : a3 b6 68 85 65 68 85 64  e2
c1a8 : ad 6f c1 20 75 b4 84 fb  88
c1b0 : a0 00 91 64 c8 8a 91 64  31
c1b8 : c8 a5 fb 91 64 a0 01 b1  37
c1c0 : 64 48 c8 b1 64 85 65 68  89
c1c8 : 85 64 ac 6f c1 88 b1 fd  bb
c1d0 : ae 18 d0 e0 15 f0 08 c9  67
c1d8 : 41 90 04 09 80 d0 06 c9  be
c1e0 : 20 b0 02 09 40 91 64 c0  9e
c1e8 : 00 d0 e2 ae 70 c1 20 0c  8c
c1f0 : e5 4c ae a7 00 00 00 00  9c
```

Listing 4.
Ein komfortabler INPUT-Befehl

Hinweis zum Abtippen
Dieses Listing muß mit dem MSE eingegeben werden. Den MSE finden Sie in dieser Ausgabe auf Seite 136.

```
PROGRAMM : EYSSELE          C900 CBD7
-----------------------------------

C900 : A9 5A A0 C9 8D 1A 03 8C  06
C908 : 1B 03 A9 91 A0 C9 8D 1C  08
C910 : 03 8C 1D 03 A9 AD A0 C9  1F
C918 : 8D 1E 03 8C 1F 03 A9 C8  49
C920 : A0 C9 8D 20 03 8C 21 03  2B
C928 : A9 E3 A0 C9 8D 26 03 8C  53
C930 : 27 03 A9 FF 8D 03 DD AD  07
C938 : 02 DD 09 04 8D 02 DD 60  0D
C940 : 48 A9 10 2C 0D DD F0 FB  62
C948 : 68 8D 01 DD AD 00 DD 09  D7
C950 : 04 8D 00 DD 29 FB 8D 00  7F
C958 : DD 60 A6 B8 F0 05 20 0F  FC
C960 : F3 D0 03 4C FE F6 A6 98  79
C968 : E0 0A 90 03 4C FB F6 E6  20
C970 : 98 A5 B8 9D 59 02 A5 B9  6C
C978 : 09 60 9D 6D 02 A5 BA 9D  3A
C980 : 63 02 C9 04 F0 04 C9 10  4E
C988 : 90 02 18 60 C9 00 4C 77  E8
C990 : F3 20 14 F3 F0 02 18 60  57
C998 : 20 1F F3 BA 48 A5 BA C9  C6
C9A0 : 10 B0 07 C9 04 F0 03 4C  70
C9A8 : 9D F2 4C F1 F2 20 0F F3  64
C9B0 : F0 03 4C 01 F7 20 1F F3  3A
C9B8 : A5 BA C9 04 F0 04 C9 10  24
C9C0 : 90 03 4C 0A F7 4C 19 F2  52
C9C8 : 20 0F F3 F0 03 4C 01 F7  11
C9D0 : 20 1F F3 A5 BA C9 04 F0  1D
C9D8 : 04 C9 10 90 03 4C 75 F2  25
C9E0 : 4C 5B F2 48 85 9E A5 9A  B9
C9E8 : C9 10 B0 07 C9 04 F0 03  4D
C9F0 : 4C CD F1 98 48 BA 48 A5  F7
C9F8 : 9E A4 9A C0 10 D0 06 20  87
CA00 : 40 C9 18 90 1F C0 11 D0  1B
CA08 : 06 20 5C CA 18 90 15 C0  6B
CA10 : 04 F0 04 C0 12 D0 06 20  A5
CA18 : 2B CA 18 90 07 C0 13 D0  25
CA20 : 03 20 44 CA 68 AA 68 A8  6D
CA28 : 68 18 60 C9 41 90 12 C9  62
CA30 : 5F B0 04 09 20 D0 0A C9  4E
CA38 : C1 90 06 C9 DE B0 02 29  CA
CA40 : 7F 4C 40 C9 C9 FF F0 18  BF
CA48 : C9 60 B0 03 4C 40 C9 E9  90
CA50 : 40 10 02 E9 40 A0 D0 84  AB
CA58 : 06 4C C4 CA C9 FF D0 06  FB
CA60 : A2 5E A0 D0 D0 5B 48 A4  C6
CA68 : B9 C0 FF D0 02 E6 B9 29  2C
CA70 : 7F C9 20 90 2C A8 A5 B9  00
CA78 : 29 01 F0 10 68 C9 A0 90  D8
CA80 : 04 C9 C0 90 03 4C 2B CA  80
CA88 : E9 40 D0 0E 68 C9 60 B0  3F
CA90 : 03 4C 40 C9 E9 40 10 02  E7
CA98 : E9 40 A0 D0 84 06 4C C4  17
CAA0 : CA 68 24 0F 30 03 4C 40  56
CAA8 : C9 18 69 40 30 02 69 40  19
CAB0 : AA A5 B9 29 02 D0 71 A0  6E
CAB8 : D0 A5 B9 29 01 F0 02 A0  CF
CAC0 : D8 84 06 8A A8 A9 00 A2  CA
CAC8 : 07 9D C0 02 CA 10 FA 98  58
CAD0 : 4A 4A 4A 4A 4A 18 65 06  22
CAD8 : 85 06 98 0A 0A 0A 85 05  D9
CAE0 : A9 01 85 03 78 A5 01 29  D7
CAE8 : FB 85 01 A0 07 B1 05 85  17
CAF0 : 02 A2 07 06 02 90 08 BD  06
CAF8 : C0 02 05 03 9D C0 02 CA  D9
CB00 : 10 F1 06 03 88 10 E6 A5  DB
CB08 : 01 09 04 85 01 58 A2 00  9D
CB10 : BD D2 CB 20 40 C9 E8 E0  E5
CB18 : 05 D0 F5 A2 07 BD C0 02  BC
CB20 : 20 40 C9 CA 10 F7 18 60  0E
CB28 : E0 D5 90 01 CA 8A 29 7F  DC
CB30 : C9 50 90 02 E9 03 C9 45  EE
CB38 : 90 02 E9 03 C9 41 90 02  91
CB40 : E9 21 C9 1C 90 02 E9 08  81
CB48 : C9 11 90 02 E9 0B 38 E9  AA
CB50 : 04 AA A9 3C 20 40 C9 A0  08
CB58 : FF CA F0 08 C8 B9 7A CB  D5
CB60 : 10 FA 30 F5 C8 B9 7A CB  94
CB68 : 30 06 20 40 C9 18 90 F4  35
CB70 : 29 7F 20 40 C9 A9 3E 4C  E4
CB78 : 40 C9 57 48 D4 43 52 C4  B6
CB80 : 52 4F CE 48 4F CD 52 45  6E
CB88 : C4 43 52 D2 47 52 CE 42  A3
CB90 : 4C D5 4F 52 C7 46 B1 46  E7
CB98 : B3 46 B5 46 B7 46 B2 46  A9
CBA0 : B4 46 B6 46 B8 42 4C CB  54
CBA8 : 43 52 D5 52 4F C6 43 4C  A5
CBB0 : D2 42 52 CE 48 52 D4 47  0B
CBB8 : 52 B1 47 52 B2 48 47 CE  27
CBC0 : 48 42 CC 47 52 B3 50 55  F4
CBC8 : D2 43 52 CC 59 45 CC 43  E3
CBD0 : 59 CE 1B 2A 04 08 00 FF  1D
```

Listing 5. Ein preiswertes und sehr leistungsfähiges Software-Interface. Bitte mit dem MSE eingeben.

Ein Centronics-Interface

Ein Interface wird benötigt, um die Datenübertragung von einem C64 auf Nicht-Commodore-Drucker (zum Beispiel Epson FX-80/85) zu ermöglichen. Dazu muß das Interface die vom Computer gesendeten Zeichen und Befehle so wandeln, daß der andere Drucker sie versteht und ausführt. Dies kann auf teuere Weise mittels eines Hardware-Interfaces oder billiger über eine Schnittstelle in Form eines Programmes erfolgen (Hardware-Interface bedeutet: In den Drucker muß ein meist teures Modul eingebaut werden, das diese Aufgabe übernimmt – wie etwa das Görlitz-, Wiesemann- oder HDS-Interface).

Die hier vorgestellte Software-Schnittstelle (unter Software versteht man Programme) übertrifft mit ihren Leistungsmerkmalen viele käufliche Hardware-Schnittstellen. Nicht aber im Preis: Nur etwa 20 Mark kostet diese »Selbstbau«-Centronics-Schnittstelle

Es handelt sich hierbei um ein Maschinenspracheprogramm mit 726 Byte Länge, das den Adreßbereich $C900 bis $CBD1 (dezimal: 51456 bis 52182) belegt (Listing 5). Ein Vorteil des Programmes besteht in der Verträglichkeit mit DOS 5.1. Das bedeutet: Es können beide Programme auf einmal im Speicher des Computers liegen. Die Befehle beider Routinen können eingegeben werden, ohne daß man mit Fehlern rechnen muß.

Das Programm ist ohne Einschränkung für alle Epsonkompatiblen Drucker anwendbar, die über einen Bitmustermodus verfügen. Dieser Modus wird für die Ausgabe der Commodore-eigenen Grafikzeichen benötigt. Das Programm besteht aus mehreren Programmteilen, von denen die meisten Erweiterungen bestehender Ein-/Ausgaberoutinen des Betriebssystems sind und bei der Initialisierung in diese eingebunden werden. Dadurch können schon bestehende Basic-Befehle wie OPEN und PRINT # zum Drucken benutzt und Programme müssen nicht umgeschrieben werden. Zur Ansteuerung verschiedener Druckmodi wurden jedoch zusätzliche Gerätenummern definiert, deren Bedeutung in Tabelle 1 erklärt ist.

Gerätenummer 16 realisiert einen sogenannten Direktmodus, mit dem die internen Commodore-Zeichencodes ohne Wandlung an den Drucker gelangen. Der Direktmodus ist zur Ausgabe von Steuerzeichen oder bei der Verwendung des

GERÄTENUMMER 16	=	DIREKTMODUS
GERÄTENUMMER 18,4	=	TEXTMODUS
GERÄTENUMMER 19	=	GRAFIKMODUS
GERÄTENUMMER 17	=	LISTMODUS

SEK.-ADR.	MODUS	STEUERZEICHEN
0	NORMAL	NORMAL
1	KLEIN	NORMAL
2	NORMAL	ERKLÄRT
3	KLEIN	ERKLÄRT

Tabelle 1. Zusätzliche Gerätenummern zur Ansteuerung verschiedener Druckmodi

	USER PORT - CENTRONICS	
A	GND	16
B	FLAG - BUSY	11
C	D0	2
D	D1	3
E	D2	4
F	D3	5
H	D4	6
J	D5	7
K	D6	8
L	D7	9
M	PA2 - STROBE	1

Tabelle 2. Verbindungsplan von User-Port auf Centronics

Druckers als Plotter zur Einzelnadelsteuerung beziehungsweise zur Ausgabe von Bitmustern geeignet. Der Textmodus (Groß- und Kleinschreibung) ist unter Gerätenummer 18 und, weil er wohl am häufigsten bei bereits bestehenden Programmen benutzt wird, unter Gerätenummer 4 ansprechbar. Die Gerätenummer 19 realisiert den Großschrift-/Grafikmodus, wie er beim C64 gleich nach dem Einschalten voreingestellt ist. Der wichtige Modus zum Listen von Programmen wurde mit Gerätenummer 17 realisiert. Es läßt sich über die Sekundäradresse noch zwischen vier Fällen unterscheiden. Einmal kann gewählt werden, ob das Listing, wie vom Bildschirm her gewohnt, mit Großbuchstaben und Grafikzeichen oder im Textmodus mit großen und kleinen Buchstaben gedruckt wird. Zum anderen kann man wählen, ob die Steuerzeichen wie bei Bildschirmausgabe als inverse Zeichen oder durch

Abkürzungen wie <CLR> (Bildschirm löschen) im Klartext gedruckt werden. Durch all diese Möglichkeiten kann der Anwender zum einen auf seinem Drucker Ausgaben erzeugen, wie man sie von Commodore-Druckern her gewohnt ist, zum anderen auch alle Möglichkeiten seines Druckers voll nutzen.

Das Programm (Listing 5) müssen Sie mit dem MSE eingeben und auf Diskette oder Kassette speichern. Vor dem Start sollten Sie das Verbindungskabel Userport-Centronics zusammenlöten. Dazu ist in Tabelle 2 ein Verbindungsplan angegeben. Das Kabel sollte für eine störungsfreie Funktion nicht länger als einen Meter sein und aus abgeschirmtem, mehradrigem Steuerkabel bestehen, das man in (fast) jedem Elektronik-Bastelgeschäft findet. Dort sind in der Regel ebenfalls der Centronics-Stecker und der Stecker für den Userport erhältlich.

Initialisiert wird die ganze Treiberroutine mit

```
SYS 51456
```

Hierbei wird die Routine in das Betriebssystem eingebunden. Jedoch Vorsicht: Nach einem Abbruch des Programms, zum Beispiel durch die Betätigung der Tasten RUN/STOP und RESTORE ausgelöst, muß die Routine erneut initialisiert werden.

Die einzelnen Druckmodi spricht man mit den üblichen Basic-Befehlen an. Geöffnet wird der Ausgabekanal mit:

OPEN log.Dateinummer, Geräteadr. [,Sekundäradr.]

Die eckigen Klammern kennzeichnen optionale Angaben. Nach Öffnen des Kanals zum Drucker kann dann mit PRINT# log.Dateinummer der Text ausgegeben werden. Ein Programm-Listing wird zum Beispiel erzeugt mit den Befehlen

```
OPEN 17,17,0:CMD17:LIST
PRINT#17:CLOSE17
```

Der PRINT-Befehl vor dem CLOSE ist notwendig, damit der CMD-Modus aufgehoben wird.

Umstellen auf beliebige Drucker mit Centronics-Schnittstelle

Das Programm wurde für einen Epson-Drucker geschrieben. Unverändert ist es für jeden anderen Drucker mit Centronics-Schnittstelle verwendbar, sofern auf die Ausgabe von Commodore-eigenen Grafikzeichen verzichtet wird (dazu gehören auch die reversen Zeichen).

(H. Eyssele/dm)

Automatische Zeilennummernvorgabe

Das lästige Eingeben der Zeilennummern beim Programmieren kann Ihnen dieses kleine Programm abnehmen. Die Syntax des AUTO-Befehls lautet:

```
←A Anfangszeilennummer, Schrittweite
```

Nach Eingabe dieses Befehls wird die Zeilennummer vorgegeben und nach RETURN um »Schrittweite« erhöht. Um aus dem AUTO-Modus wieder herauszukommen, muß man nach Vorgabe einer Zeilennummer das Zeichen »←« und RETURN eingeben.

Falls man nach Vorgabe einer Zeilennummer die RETURN-Taste betätigt, wird die entsprechende Zeile, falls vorhanden, gelöscht. Hiermit lassen sich auch sehr schnell Programmblöcke löschen, falls man die RETURN-Taste gedrückt hält, da die Zeilenvorgabe weiterläuft und die entsprechenden Zeilennummern gelöscht werden.

Das Programm (Listing 6) geben Sie bitte mit dem MSE ein und speichern es auf Ihren Datenträger. Sie starten das Programm mit

```
SYS 49152
```

(Frank Siedel/dm)

```
programm : auto-number      c000 c0aa
-------------------------------------

c000 : a9 0b 8d 08 03 a9 c0 8d  2f
c008 : 09 03 60 20 73 00 08 c9  9a
c010 : 5f f0 04 28 4c e7 a7 20  d0
c018 : 73 00 c9 41 d0 f5 20 73  4a
c020 : 00 18 20 6b a9 a5 14 85  c5
c028 : 26 a5 15 85 27 20 fd ae  e0
c030 : 18 20 6b a9 a5 14 85 28  ca
c038 : a5 15 85 29 a9 81 8d 02  cf
c040 : 03 a9 c0 8d 03 03 a9 80  ea
c048 : 8d 8a 02 a5 27 85 62 a5  c3
c050 : 26 85 63 a2 90 38 20 49  44
c058 : bc 20 dd bd a2 00 bd 01  76
c060 : 01 f0 09 9d 00 02 20 d2  06
c068 : ff e8 d0 f2 20 12 e1 c9  1b
c070 : 5f f0 1e c9 0d f0 2d 9d  50
c078 : 00 02 e8 20 62 a5 4c 86  49
c080 : a4 18 a5 26 65 28 85 26  58
c088 : a5 27 65 29 85 27 4c 4b  99
c090 : c0 a9 83 8d 02 03 a9 a4  e0
c098 : 8d 03 03 a9 00 8d 8a 02  37
c0a0 : 28 4c 74 a4 20 76 a5 4c  85
c0a8 : 86 a4 c1 c9 11 d0 03 4c  66
```

Listing 6. Erweitern Sie Ihr Basic um den AUTO-NUMBER-Befehl. Bitte mit dem MSE eingeben.

Angeschlossene Geräte eingeschaltet?

Dieses kleine Unterprogramm (Listing 7) dient dazu, ein Peripheriegerät (Drucker, Floppy) zuerst darauf zu testen, ob es überhaupt eingeschaltet ist. Dieses ist möglich durch Abfrage des Statusbytes. Enthält es einen anderen Wert als -128, so ist ein Fehler (DEVICE NOT PRESENT) aufgetreten. Bei nicht vorhandener Abfrage kann es schlimmstenfalls passieren, daß das Hauptprogramm abstürzt. Falls Sie diese Unterroutine in Ihr Programm einbauen wollen, so ersetzen Sie die jeweiligen PRINT-Anweisungen durch Rücksprünge in Ihr Hauptprogramm. (dm)

```
100 REM DRUCKERTEST                          <043>
120 OPEN 4,4                                 <223>
135 POKE 768,185                             <077>
155 PRINT#4:CLOSE 4                          <098>
170 POKE 768,139                             <115>
190 IF ST<>-128 THEN 205                     <170>
195 PRINT"DRUCKER NICHT EINGESCHALTET !      <063>
200 GOTO 215                                 <056>
205 PRINT"DRUCKER EINGESCHALTET !            <206>
215 REM HIER GEHT'S WEITER IM PROGRAMM       <245>
225 REM FLOPPYTEST                           <047>
240 POKE 768,185                             <182>
260 OPEN 15,8,15,"I":CLOSE 15                <182>
275 POKE 768,139                             <222>
295 IF ST<>-128 THEN 310                     <004>
300 PRINT"FLOPPY NICHT EINGESCHALTET !       <062>
305 GOTO 320                                 <027>
310 PRINT"FLOPPY EINGESCHALTET !             <138>
320 REM HIER GEHT'S WEITER IM PROGRAMM       <096>
```

Listing 7. Keine Probleme mehr mit ausgeschalteten Druckern oder Floppies

Zugriffszeit der Floppy verkürzen

Das vorliegende kleine Programm dient dazu, die Zugriffszeit der Floppy 1541 drastisch zu verkürzen. Der Schrittmotor, der den Schreib-Lesekopf bewegt, kann erfahrungsgemäß wesentlich schneller arbeiten, ohne daß eine sichere Funktion der Floppy gefährdet wird. Da der Schrittmotor im Interrupt bedient wird, genügt es, die Größe des Interrupt-

intervalls zu verändern, um die Drehzahl des Motors zu beeinflussen. Standardmäßig wird etwa alle 15 Millisekunden ein Interrupt ausgelöst, der den Stepper um eine Viertelspur bewegt. Durch das vorliegende Programm wird diese Zeit auf etwa 4 Millisekunden verkürzt. Alle Bewegungen des Kopfes werden dadurch etwa viermal schneller. Das hat neben der Zeitersparnis noch zwei weitere wesentliche Vorteile: Das Laufgeräusch des Kopfes wird angenehm leise und kurz, und im Falle einer Kopfjustage (MG-salvenartiges Geräusch) fährt der Kopf mit erheblich verminderter Kraft gegen den Anschlag, so daß die Gefahr einer Dejustage deutlich vermindert wird.

```
10 OPEN 1,8,15,"M-W"+CHR$(7)+
   CHR$(28)+CHR$(1)+CHR$(15)
```

(Robert Loos/dm)

Zahlensystemwandlung

Hier wollen wir einen leicht verständlichen Lösungsweg zur Umrechnung der beiden Zahlensysteme Dezimal und Hexadezimal erarbeiten.

Warum Hexadezimal?

Wir sind von Kind auf an das dezimale Zahlensystem gewöhnt. In der Computertechnik wird jedoch auch häufig das hexadezimale Zahlensystem verwendet. Warum? Um diese Frage zu beantworten, müssen wir etwas ausholen.

Wie Sie vielleicht wissen, kennt der Computer im Prinzip nur das binäre Zahlensystem, also die Kombination von 0 und 1. Aus diesen beiden Ziffern wird jede Zahl gebildet. Beispiel:

binär 1101 = dezimal 13 = hexadezimal D
binär 1100000000000000 = dezimal 49152 = hexadezimal C000

In Fachzeitschriften finden Sie oft Abkürzungen oder andere Schreibweisen, um klarzustellen, welches Zahlensystem gemeint ist. Beispiel:

binär = bin = %
hexadezimal = hex = $
dezimal = dez = #

Die Zeichen %, $ und # werden vor allem von Programmen benutzt, die irgendetwas mit Assembler (Maschinensprache) oder mit dem Betriebssystem des Computers zu tun haben, zum Beispiel Maschinensprache-Monitore. Wenn Sie mehr darüber wissen wollen, empfehle ich Ihnen das 64'er Sonderheft 8/85 (Assembler für Anfänger und Fortgeschrittene). Doch zurück zu den Zahlensystemen:

Die ersten Computer konnten nur mit binären Werten gefüttert werden, eine sehr anstrengende Sache für die Programmierer. Eine wesentliche Vereinfachung ergab das hexadezimale System. Jeweils 4 Binär-Ziffern konnten ersetzt werden durch eine einzige Hex-Ziffer. Beispiel:

bin 1101 = hex D = dez 13
bin 1000 = hex 8 = dez 8

Das liegt daran, daß das binäre Zahlensystem die Basis 2 besitzt, sich jede Zahl also in 2er-Potenzen darstellen läßt, während das hexadezimale System die Basis 16 hat. Da 2 hoch 4 den Wert 16 ergibt, können 4 Binär-Ziffern durch eine Hex-Ziffer ersetzt werden.

Unser C64 und auch fast alle anderen Heimcomputer sind 8-Bit-Computer. Der C64 kann 64 KByte Speicherplatz adressieren, das sind 2 hoch 16 = dezimal 65536 = hex FFFF. Sie sehen, überall taucht die 2 auf, beziehungsweise Potenzen von 2. Und das macht die Verwendung von hex-Zahlen so einfach, weil man sehr oft mit runden Hex-Zahlen zu tun hat. Beispiel: Als C64-Besitzer werden Sie immer wieder mit der Speicherstelle 49152 zu tun haben. Mit SYS 49152 werden sehr viele Maschinenprogramme gestartet. Hexadezimal gesehen entspricht das der Zahl C000. Weitere Beispiele liefert die Speicheraufteilung des C64:

hex	dez	Inhalt
100	256	Ende der Zeropage
400	1024	Anfang Bildschirmspeicher
800	2048	Anfang Basic-Speicher
A000	40960	8 KByte Basic-ROM
C000	49152	4 KByte RAM
D000	53248	8 KByte Interpreter-ROM

Sie sehen, daß viele wichtige Adressen »krumme« dezimale Zahlen ergeben, die sich in der Regel schlechter merken lassen. Leider kann man beim C64 nicht auf das dezimale System verzichten. SYS-, POKE- und PEEK-Befehle lassen nur dezimale Parameter zu.

Beispiel: MSE

Ein Programm, mit dem Sie in jeder 64'er Zeitschrift und auch in diesem Sonderheft konfrontiert werden, ist der MSE. Mit dem MSE lassen sich Maschinenspracheprogramme sehr einfach, schnell und fehlerfrei abtippen. Tippen Sie ein MSE-Listing ab, haben Sie es mit Hex-Zahlen zu tun, und das aus zwei Gründen:

1. Hex-Zahlen benötigen weniger Platz als Dezimal-Zahlen, nämlich anstatt mit drei Ziffern kommt man mit zwei Ziffern aus. Das bedeutet weniger Tipp-Arbeit.
2. Wir können mehr Listing pro Seite unterbringen. Eine MSE-Zeile besteht aus der Adresse und einem Doppelpunkt (= 5 Ziffern), gefolgt von 8 Bytes plus einem Prüfbyte. Insgesamt ergibt das 23 Ziffern. Würde der MSE dezimale Zahlen verwenden, wären 33 Ziffern notwendig.

Beispiel: SMON

Selbst wenn Sie jetzt noch Anfänger sind, wird es nicht sehr lange dauern, bis Sie die ersten Schritte in Maschinensprache wagen. Spätestens dann werden Sie nicht mehr auf das hexadezimale Zahlensystem verzichten können. Denn beim SMON und auch bei den meisten anderen Maschinensprache-Monitoren wird mit Hex-Zahlen gearbeitet. Das geschieht auch hier hauptsächlich aus Platzgründen, aber auch, weil es üblich ist.

Ich hoffe, daß Sie jetzt etwas die Scheu vor diesem Zahlensystem verloren haben. Selbst wenn Sie vorerst nicht mit Hex-Zahlen arbeiten, werden Sie wissen, was gemeint ist, wenn Sie lesen: »Das Programm beginnt bei $C000 und wird mit SYS 49152 gestartet«.

1. Was ist das Hexadezimalsystem?

In unserem normalen Zahlensystem repräsentiert jede Stelle einer Zahl eine Zehnerpotenz. Ein Beispiel: Die Zahl 4714 läßt sich auch als Summe von Zehnerpotenzen schreiben.

$4714 = 4*10^3 + 7*10^2 + 1*10^1 + 4*10^0 = 4*1000 + 7*100 + 1*10 + 4*1$

Beim Hexadezimalsystem wird nun jede Stelle einer Zahl nicht mehr durch eine Zehner-, sondern durch eine Sechzehnerpotenz repräsentiert. Auch hier wieder ein Beispiel: Die Hexadezimalzahl 0324 bedeutet nichts anderes als $3*16^2 + 2*16^1 + 4*16^0$ (=3*256+2*16+4*1).

Dies hat aber noch weitere Konsequenzen:

Im Dezimalsystem wird eine Stelle immer von 0 bis 9 (insgesamt 10 Ziffern) durchgezählt, bevor die nächste Stelle um eins erhöht wird. Also

00 01 02 03 04 05 06 07 08 09
10 11 12 13 14...

Hex, Dez, Bin . . .?

Im Hexadezimalsystem jedoch wird eine Stelle um sechzehn Werte erhöht, bevor zur nächsten Stelle ein Wert hinzugefügt wird. Da aber unsere Ziffern von 0 bis 9 dazu nicht ausreichen, wurden zusätzlich die Buchstaben A bis F herangezogen. Sie vertreten die Zahlenwerte 10 bis 15 (von 0 bis 15 sind es 16 Werte!). Es bedeuten:

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

0 1 2 3 4 5 6 7 8 9 A B C D E F
10 11 12 13 14... 1A 1B 1C...

Die Dezimalzahl 10 ist also gleichwertig mit dem Hexadezimalwert A. Damit wären wir auch schon bei der Umrechnung.

2. Dezimal - Hexadezimal

Wenn wir eine Dezimalzahl in Hexadezimal umrechnen wollen, so bauen wir den Hex-Wert Stelle für Stelle von links nach rechts auf.

Nehmen wir also an, wir möchten die Dezimalzahl 41717 in Hexadezimal umrechnen. Dazu teilen wir sie erst einmal durch 16^3

$41717 : 16^3 = 10{,}1848145$

Uns interessiert hier nur die Vorkommastelle 10. Sie ist gleichbedeutend mit dem Hex-Wert **A**. Er bildet die letzte Stelle unserer Hexadezimalzahl.

Nun müssen wir von unserer Dezimalzahl $10 * 16^3$ abziehen.

Also $41717 - 10 * 16^3 = 757$

Um die nächsten Stellen unserer Hex-Zahl zu erhalten, führen wir diese Prozedur nun noch mit 16^2 und 16^1 durch:

$757 : 16^2 = 2{,}9703125$ (=**2**)
$757 - 2 * 16^2 = 245$
$245 : 16^1 = 15{,}3125$ (=**F**)
$245 - 15 * 16^1 = 5$

Als endgültige Umrechnung der Zahl 41717 ins Hexadezimalsystem erhalten wir also **A2F5**.

3. Hexadezimal - Dezimal

Diese Umrechnung ist schon wesentlich einfacher. Um die Hex-Zahl A2F5 wieder zurückzurechnen, geht man wie folgt vor:

$A * 16^3 + 2 * 16^2 + F * 16^1 + 5 * 16^0$

Da man aber mit den Buchstaben A und F nicht rechnen kann, müssen diese als Dezimalzahlen angegeben werden.

$10 * 16^3 + 2 * 16^2 + 15 * 16^1 + 5 * 16^0$

Wenn Sie dies auf Ihrem C64 einmal ausrechnen, so werden Sie als Ergebnis wieder die Zahl 41717 erhalten!

Als Abschluß unseres kleinen Kurses könnten Sie einmal versuchen, ein Basic-Programm zu schreiben, das diese Berechnungen ausführt.

Kleine Merkhilfe

Gerade wenn man anfängt, sich mit dem hexadezimalen Zahlensystem zu beschäftigen (und dafür gibt es einige gute Gründe), ist man froh über ein paar kleine Eselsbrücken. Am Anfang kommt man immer wieder ins Stolpern, weil man sich nicht so schnell an den Dezimalwert von zum Beispiel Hex C erinnern kann. Dies geht wesentlich leichter, wenn man die Anfangsbuchstaben der beiden Systeme gegenüberstellt.

C = **Z**wölf
D = **D**reizehn
F = **F**ünfzehn

Dezimal	Hexadezimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15
10	16
11	17
12	18
4096	1000
49152	C000 = 12*4096

Tabelle 3.
Einige Beispiel, wie Hex-Werte in dezimal und umgekehrt aussehen würden. Man sieht deutlich die unterschiedliche Darstellungsweise der Zahlensysteme.



Zeichen in vierfacher Größe

Ein tolles Programm zur Darstellung des normalen Zeichensatzes in vierfacher Breite und Höhe. »Blow Up« ist gedacht für große Überschriften auf dem Bildschirm, für Nummerierungen innerhalb von mehrstufigen Menü-Programmen und die hervorgehobene Ausgabe von Ergebnissen. Der Trick dabei besteht darin, daß die Zeichen aus normalen Blockgrafik-Elementen zusammengesetzt werden. Dadurch erspart man die Umstellung auf den Grafikmodus und auf spezielle Zeichensätze.

Bitte geben Sie zuerst das Maschinenprogramm »Blow Up« (Listing 8) mit Hilfe des MSE ein und speichern es. Das Demo-Programm (Listing 9) können Sie ebenfalls abtippen, müssen aber nicht. Nachdem Sie es ebenfalls gespeichert haben, starten Sie es mit RUN.

Zur Bedienung: Das Programm ist mit LOAD"BLOW UP",8,1 zu laden. Die Übergabe des auszudruckenden Textes erfolgt mit

`SYS 51968($)`

Dabei kann eine Stringvariable ($) in der Form A$ oder "TEST" übergeben werden, zum Beispiel SYS 51968 ("TEST"). Auch kombinierte Strings (A$+B$) sind möglich. Der Ausdruck muß jedoch in Klammern stehen, da sonst die Fehlermeldung SYNTAX ERROR ausgegeben wird. Die verschiedenen Zeichensätze des C64 werden über einen POKE-Befehl erreicht. Dabei gilt:

`POKE 52073,208`
– Großschrift/Blockgrafik (Grundeinstellung)
`POKE 52073,212`
– Großschrift/Blockgrafik invertiert
`POKE 52073,216`
– Groß-/Kleinschrift
`POKE 52073,220`
– Groß-/Kleinschrift invertiert

Steuerzeichen innerhalb eines Strings werden ähnlich wie beim LISTen dargestellt, aber nicht ausgeführt. Die Eingabe eines Leerstrings führt zur Ausgabe eines 255 Zeichen langen Bereichs aus dem Basic-Speicher. Dieser »Fehler« wurde nicht beseitigt, da diese wirren Zeichen ganz reizvoll sein können.

Die Ausgabe der vergrößerten Zeichen beginnt ab der Position des Cursors. Die Cursor-Position wird durch »Blow Up« nicht verändert. Zeichen, die im unteren Bildschirmbereich nicht mehr vollständig abgebildet werden können, wandern an den Bildschirmanfang. Da Strings bis zu 255 Zeichen lang sein können, kann der Bildschirm von einem String mehrfach überschrieben werden.

(Ludwig Wolff/dm)

```
 PROGRAMM : BLOW UP          CBOO CBFC
-------------------------------------

CB00 : 20 FA AE 20 9E AD 20 A3   6C
CB08 : B6 86 28 84 29 8D EA CB   DE
CB10 : AD 0E DC 29 FE 8D 0E DC   6F
CB18 : 18 A5 D1 65 D3 AA A5 D2   F3
CB20 : 69 00 A8 86 FA 86 71 98   5F
CB28 : 85 FB 49 DC 85 72 A9 0A   3F
CB30 : 8D EB CB A0 04 A2 00 86   1C
CB38 : FF 86 02 A5 FB C9 07 90   FB
CB40 : 06 A5 FA C9 6D B0 DC A1   24
CB48 : 28 30 0C C9 40 90 0C 29   D0
CB50 : 3F C9 40 90 06 29 1F 29   0E
CB58 : 7F 09 40 18 0A 26 FF 0A   55
CB60 : 26 FF 0A 26 FF 85 FE 18   26
CB68 : A9 D0 65 FF 85 FF A5 01   C3
CB70 : 29 FB 85 01 A2 00 A1 FE   C7
CB78 : 85 FC E6 FE A1 FE 85 FD   39
CB80 : E6 FE A5 01 09 04 85 01   38
```

Listing 8. Zaubern Sie viermal so große Buchstaben auf Ihren Bildschirm. Bitte mit dem MSE eingeben.

```
CB88 : 66 FC 2A 66 FC 2A 66 FD   7A
CB90 : 2A 66 FD 2A 29 0F AA 88   79
CB98 : BD EC CB 91 FA AD 86 02   2C
CBA0 : 91 71 C4 02 D0 E2 98 18   12
CBA8 : 69 28 85 02 69 04 A8 C9   B4
CBB0 : A4 D0 BB CE EB CB D0 09   F7
CBB8 : A9 0A 8D EB CB A9 7C D0   E5
CBC0 : 02 A9 04 18 65 FA 90 04   13
CBC8 : E6 FB E6 72 85 FA 85 71   DD
CBD0 : E6 28 D0 02 E6 29 CE EA   07
CBD8 : CB F0 03 4C 33 CB AD 0E   CA
CBE0 : DC 09 01 8D 0E DC 20 F7   6B
CBE8 : AE 60 00 04 20 7B 6C 62   9B
CBF0 : 7E 61 7F FC 7C FF E1 FE   EB
CBF8 : E2 EC FB A0 00 00 00 00   63
```

Listing 8. Schluß

```
100 REM ******************************     <238>
101 REM *   DEMO + BOOT  PROGRAMM    *     <061>
102 REM *           FUER             *     <167>
103 REM *                            *     <152>
104 REM *       B L O W   U P        *     <191>
105 REM *                            *     <154>
106 REM *  EINE BASIC-ZUSATZFUNKTION  *     <148>
107 REM *   FUER ZEICHENDARSTELLUNG   *     <120>
108 REM *  IN 4-FACHER HOEHE + BREITE *     <149>
109 REM *                            *     <158>
110 REM *           VON              *     <243>
111 REM *                            *     <160>
112 REM *       LUDWIG WOLFF         *     <009>
113 REM *      NIEDERHUTSTR.42       *     <208>
114 REM *      5483 AHRWEILER        *     <045>
115 REM *      TEL.02641/34867       *     <213>
116 REM *                            *     <165>
117 REM *        15.11.1985          *     <173>
118 REM ******************************     <000>
200 :                                      <176>
300 :                                      <022>
400 :                                      <122>
500 IF A=0 THEN A=1:LOAD"BLOW UP",8,1      <099>
520 C=646:POKE 53280,0:POKE 53281,0        <221>
530 Z1$="@ABCDEFGHIJKLMNOPQRSTUVWXYZ £↑123
    4567890"                               <112>
540 Z2$="ZVXSHQ*M*X-+LFGJ@YR/£RNB*U+-Z↑UYT
    @RFCDT UYT GHJK±LNDVD B+[]←"           <222>
600 GOSUB 1000:S=9:Z=6:GOSUB 1100          <150>
610 PRINT"EIN DEMO-PROGRAMM FUER"          <118>
620 S=12:Z=11:GOSUB 1100:PRINT" B L O W{3S
    PACE}U P":T=600:GOSUB 1200             <221>
650 GOSUB 1000:PRINT"{HOME,2DOWN,2RIGHT,SP
    ACE}BLOW UP{2SPACE}IST EIN HILFSPROGRA
    MM ZUR"                                <033>
660 PRINT"{DOWN,2RIGHT}DARSTELLUNG VON ZEI
    CHEN AUS DEM C64-"                     <136>
670 PRINT"{2DOWN,4RIGHT}ZEICHENSATZ IN 4-F
    ACHER GROESSE:":T=300:GOSUB 1200       <159>
680 S=6:Z=11:POKE C,1:GOSUB 1100:SYS 51968
    ("BLOW UP"):T=150:GOSUB 1200           <124>
690 S=4:Z=16:POKE C,10:GOSUB 1100:SYS 5196
    8("DEMO-PRG")                          <052>
700 T=400:GOSUB 1200                       <083>
710 GOSUB 1000:S=11:Z=13:GOSUB 1100:PRINT"
    DAS KANN BLOW UP:"                     <150>
720 T=150:GOSUB 1200:PRINT"{CYAN,HOME}":SY
    S 51968(Z1$)                           <035>
730 S=13:Z=19:GOSUB 1100:PRINT"{YELLOW}GRO
    SSE{2SPACE}ZEICHEN."                   <065>
740 T=450:GOSUB 1200:POKE 52073,216:PRINT"
    {LIG.RED,HOME}":SYS 51968(Z1$)         <170>
750 S=13:Z=19:GOSUB 1100:PRINT"{YELLOW}KLE
    INE":GOSUB 1200:GOSUB 1000             <058>
760 S=7:Z=2:POKE C,10:GOSUB 1100:SYS 51968
    ("GROSSE,")                            <020>
770 S=8:Z=6:POKE C,14:GOSUB 1100:SYS 51968
    ("KLEINE")                             <017>
780 S=0:Z=10:POKE C,7:GOSUB 1100:SYS 51968
    ("BUCHSTABEN")                         <042>
790 GOSUB 1200:GOSUB 1000:S=10:Z=10:GOSUB
    1100:PRINT"UND DIE BLOCKGRAHIK:"       <042>
800 T=150:GOSUB 1200:PRINT"{HOME}";:POKE 5
    2073,208                               <059>
810 SYS 51968(Z2$)                         <141>
820 GOSUB 1200:POKE 52073,212              <039>
830 SYS 51968(Z2$)                         <161>
```

Listing 9. Ein Demo-Programm zu »BLOW UP«

```
840 GOSUB 1200:GOSUB 1010:POKE 52073,216:S
    =13:Z=7:POKE C,14:GOSUB 1100                <222>
850 SYS 51968("AUCH"):S=8:Z=13:GOSUB 1100:
    SYS 51968("INVERS")                         <068>
860 GOSUB 1200:GOSUB 1000:S=11:Z=7:GOSUB 1
    100:PRINT"{YELLOW}UND HIER EIN TRICK:"      <216>
870 L$="{20SPACE}"                              <181>
880 PRINT SPC(90)L$SPC(20)L$"{LIG.RED}"SPC
    (20)L$SPC(20)"{LIG.BLUE}"L$                 <181>
890 POKE 52128,165:S=10:Z=10:GOSUB 1100:SY
    S 51968("64'ER")                            <030>
900 POKE 52128,145:POKE 52073,208:T=350:GO
    SUB 1200:GOTO 600                           <187>
999 STOP                                        <047>
1000 REM -- BILDSCHIRM MASKE --                 <236>
1005 PRINT"{CLR}"                               <231>
1010 PRINT"{HOME,YELLOW}R**********************
     **********************S";                  <071>
1020 FOR I=0 TO 20:PRINT"▒"SPC(38)"▒";:NEX
     T                                          <155>
1030 PRINT"▒ TASTE DRUECKEN ODER WARTEN !"
     SPC(9)"▒";                                 <239>
1040 PRINT"Z**********************************
     ********X";                               <195>
1050 PRINT" * BLOW UP * L.WOLFF * AHRWEILE
     R 1985 *";:RETURN                          <184>
1100 REM -- CRSR SET --                         <152>
1110 POKE 211,S:POKE 214,Z:SYS 58640:RETUR
     N                                          <038>
1200 REM -- GET --                              <027>
1210 I=0                                        <127>
1220 I=I+1:IF I>T THEN RETURN                   <168>
1230 GET A$:IF A$=""THEN 1220                   <008>
1240 RETURN                                     <026>
```

Listing 9. »BLOW-UP« (Schluß)

Formatierte Zahlenausgabe

Wie allgemein bekannt und vielfach bemängelt, bietet das Commodore-Basic keinen PRINT USING-Befehl. Doch gerade bei kommerziellen Problemen kann auf eine Druckaufbereitung von Zahlen nicht verzichtet werden. Eine Rechnung, in der nicht einmal die Dezimalpunkte untereinander stehen, ist eben unübersichtlich und keine Reklame. Für die Druckaufbereitung gibt es verschiedene Lösungen.

Die beste Lösung ist ein Maschinenprogramm. Doch zuerst stellt sich die Frage, wie ein Maschinenprogramm für die Druckaufbereitung aufgerufen werden soll. Offensichtlich ist die USR-Funktion am geeignetsten, da sie sowohl in einer PRINT- als auch in einer PRINT#- oder LET-Anweisung verwendet werden kann. Die Funktion wird aufgerufen durch USR(X),L,NK.

Dabei ist X die Zahl, die aufbereitet werden soll,
L die Gesamtfeldlänge der aufbereiteten Zahl einschließlich Vorzeichen und Dezimalkomma und
NK die Anzahl der darzustellenden Nachkommastellen.

Die USR-Funktion wandelt zunächst die Zahl X in einen ASCII-String um und berechnet die Stringlänge und die Anzahl der Nachkommastellen. Wenn bei der Wandlung von X der Interpreter die Exponentialform wählt, dann wird die Exponentialdarstellung zunächst in die Fließkommadarstellung umgewandelt. Danach folgt die Aufbereitung der Nachkommastellen. Fehlende Nachkommastellen werden durch angehängte Nullen ergänzt. Müssen Nachkommastellen abgeschnitten werden, dann rundet der Interpreter die Zahl, wenn die erste abgeschnittene Dezimalstelle größer oder gleich 5 ist. Wenn die Anzahl der gewünschten Nachkommastellen null ist, wird die Zahl als ganze Zahl (integer) ohne Dezimalkomma aufbereitet. Wegen der kaufmännischen Anwendung ersetzt die Routine den Dezimalpunkt durch ein Dezimalkomma. Nach der Aufbereitung der Nachkommastellen erhält der String durch Voranstellen von Leerzeichen die erforderliche Länge. Ist der String nach der Aufbereitung der Nachkommastellen schon länger als gewünscht, so wird er nicht mehr verändert, sondern in voller Länge ausgegeben, um einen Datenverlust zu verhindern.

Das hier vorgestellte Maschinenprogramm kann vom Anwender in den Speicherbereich geladen werden, der ihm am geeignetsten erscheint. Wenn druckaufbereitete Werte einer Variablen zugewiesen werden sollen, dann muß die Anweisung

```
A$=""+USR(X),L,NK
```

oder

```
A$=(USR(X),L,NK)+""
```

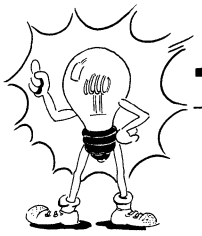
lauten, da dann das Ergebnis der Stringverknüpfung in den Speicherbereich kopiert wird und der Variablen A$ dauerhaft zugewiesen ist.

Vor dem ersten Aufruf der USR-Funktion muß jetzt noch in Adresse 785 (Low-Byte) und 786 (High-Byte) die Startadresse der USR-Funktion hinterlegt werden. Listing 10 zeigt das Basic-Ladeprogramm für die USR-Funktion. Die Ladeadresse können Sie selbst bestimmen. Das Ladeprogramm setzt die Startadresse der USR-Funktion in den Speicherstellen 785 und 786 entsprechend. Zur Verdeutlichung der Anwendung der USR-Funktion enthält das Ladeprogramm verschiedene Druckaufbereitungen der Zahl. Das Ergebnis des Beispiels ist in Bild 2 wiedergegeben.

(Dr. Michael Irskens/dm)

```
100 S=0:INPUT "STARTADRESSE";B                  <182>
110 FOR I=B TO B+335                            <000>
120 READ A                                      <160>
130 S=S+A:POKE I,A                              <062>
140 NEXT I                                      <224>
150 IF S<>36986 THEN PRINT "{RVSON,SPACE}F
    EHLER IN DEN DATAZEILEN{SPACE,RVOFF}":
    STOP                                        <119>
160 POKE 785,B-256*INT(B/256):POKE 786,B/2
    56                                          <040>
170 PRINT " ALLES OK. "                         <036>
199 REM TESTBEISPIELE                           <050>
200 FOR E=0 TO 9                                <238>
210 F=↑*10↑E                                    <176>
220 A$=(USR(F),11,0)+""                         <178>
230 PRINT A$;USR(F),13,1;USR(F),14,2            <024>
240 NEXT E                                      <036>
250 END                                         <252>
299 REM DATAZEILEN                              <190>
300 DATA 32,141,173,32,221,189,32,253,174,
    32,158,183,134,88,32,253,174,32             <212>
301 DATA 158,183,134,87,104,104,162,255,16
    0,0,232,189,0,1,240,117,201,69,240          <057>
302 DATA 8,201,46,208,242,138,168,208,238,
    173,2,1,201,46,208,12,202,160,1             <250>
303 DATA 200,185,1,1,153,0,1,208,247,189,2
    ,1,41,15,10,133,2,10,10,101,2,125           <139>
304 DATA 3,1,233,47,188,1,1,192,45,240,23,
    105,3,134,2,229,2,168,169,48,157            <106>
305 DATA 0,1,232,136,208,249,169,0,157,0,1
    ,240,168,133,2,169,0,157,0,1,138            <207>
306 DATA 24,101,2,168,189,0,1,240,8,201,48
    ,176,4,169,48,208,1,202,153,0,1             <048>
307 DATA 136,208,236,169,46,141,1,1,208,12
    9,152,240,18,165,87,208,7,152,170           <211>
308 DATA 189,1,1,208,119,169,44,153,0,1,20
    8,12,196,87,240,40,169,44,157,0             <159>
309 DATA 1,232,208,16,132,2,56,138,229,2,5
    6,233,1,197,87,240,19,176,72,168            <011>
310 DATA 169,48,157,0,1,232,200,196,87,208
    ,247,169,0,157,0,1,173,1,1,201,48           <115>
311 DATA 176,17,232,138,168,185,255,0,153,
    0,1,136,208,247,169,48,141,1,1,228          <091>
312 DATA 88,176,20,164,88,189,0,1,153,0,1,
    136,202,16,246,169,32,153,0,1,136           <163>
313 DATA 16,250,169,0,160,1,76,135,180,56,
    229,87,133,2,138,56,229,2,170,189           <206>
314 DATA 0,1,201,53,144,179,138,168,136,24
    0,24,185,0,1,201,48,144,246,24,105          <086>
315 DATA 1,201,58,153,0,1,208,157,169,48,1
    53,0,1,208,229,138,168,185,0,1,153          <028>
316 DATA 1,1,136,208,247,169,49,141,1,1,23
    2,208,131                                   <030>
```

Listing 10. Formatierte Zahlenausgabe auf Ihrem Drucker – nun kein Problem mehr



3	3,1	3,14
31	31,4	31,42
314	314,2	314,16
3142	3141,6	3141,59
31416	31415,9	31415,93
314159	314159,3	314159,27
3141593	3141592,7	3141592,65
31415927	31415926,5	31415926,50
314159265	314159265,0	314159265,00
3141592650	3141592650,0	3141592650,00

Bild 2. Beispiele für verschiedene Druckaufbereitungen

Directory sortieren

Ist Ihnen das auch schon öfters passiert: Sie hatten auf einer Diskette ein schön geordnetes Directory und beim Speichern eines weiteren Files stand dieses nicht sauber am Schluß des Directorys, sondern mitten zwischen den anderen Programmen. Mit dem hier vorgestellten Sortierprogramm (Listing 11) ist es nun möglich, diese »falsch hineingeratenen« Files herauszunehmen und an geeigneter Stelle wieder einzusetzen. Auch das Einsetzen von Trennstrichen ist möglich. Bild 3 und Bild 4 zeigen Ihnen den Unterschied zwischen einem unsortierten zu einem sortierten Directory. Mit etwas Geschick kann man seine Disketten auch so umsortieren, daß an erster Stelle immer ein Ladeprogramm für das Hauptprogramm steht. Man kann sich dadurch viel Sucharbeit sparen, denn man lädt einfach das erste Programm.

Nach dem Starten des Programms legt man die Diskette ein, die sortiert werden soll und drückt eine Taste. In der linken oberen Ecke wird nun die Sektornummer des Directory-Blocks angezeigt, den der Computer gerade einliest, rechts daneben die Anzahl der in den Speicher eingelesenen Files.

Hinweise zum Abtippen
Im folgenden Listing tauchen eventuell unterstrichene oder überstrichene Zeichen auf. Diese sind folgendermaßen einzugeben:
unterstrichen: SHIFT-Taste und Buchstabe
überstrichen: COMMODORE-Taste und Buchstabe
Bei Begriffen in geschweiften Klammern, zum Beispiel {CLR}, muß die Taste SHIFT und CLR gedrückt werden und weder die Klammer noch das Wort CLR.
Die <Zahl> am Ende jeder Basic-Zeile darf nicht eingegeben werden.
Genaueres steht im Beitrag Checksummer 64 auf Seite 135.

```
100 GOTO 200                                          <036>
110 INPUT#1,F1,F$,F2,F3:IF F1=0 THEN RETUR
    N                                                 <139>
120 PRINT F1;F$,F2;F3:END                             <030>
130 IF NO+EN>=C THEN RETURN                           <043>
131 SYS 53056,0,2,2,23,18:NO=NO+1                     <007>
135 POKE 214,23:POKE 211,2:SYS 58732:PRINT
     NA$(ZU(NO+EN-1));:RETURN                         <057>
140 IF NO=0 THEN RETURN                               <227>
141 SYS 53056,1,2,2,23,18:NO=NO-1                     <023>
145 POKE 214,2:POKE 211,2:SYS 58732:PRINT
    NA$(ZU(NO+1)):RETURN                              <250>
200 POKE 53280,6:POKE 53281,6                         <108>
201 PRINT"{CLR,YELLOW,RVSON}"TAB(9)"EPLOX'
    S DIRECTORY-SORT"                                 <125>
202 PRINT TAB(8)"------------------------"
    :IF DG=1 THEN 210                                 <002>
203 PRINT" F1{2SPACE}EINTRAG MARKIEREN/EIN
    ORDNEN"                                           <009>
204 PRINT" F2{2SPACE}TRENNSTRICH ERZEUGEN"            <081>
205 PRINT" F3{2SPACE}AUF"                             <146>
206 PRINT" F4{2SPACE}LOESCHEN"                        <200>
207 PRINT" F5{2SPACE}AB"                              <227>
```

Listing 11. Komfortables Sortieren Ihres Directory.

```
208 PRINT" F8{2SPACE}SPEICHERN"                       <022>
209 PRINT "{3SPACE,3DOWN}BITTE DISKETTE EI
    NLEGEN":DG=1:POKE 198,0:WAIT 198,63:GO
    TO 201                                            <027>
210 OPEN 1,8,15,"I":DIM AN$(145),NA$(145),
    RE$(145),ZU(146),SN(19)                           <244>
215 FOR I=1 TO 18:READ SN(I):NEXT                     <171>
218 FOR I=52992 TO 53242:READ X:POKE I,X:N
    EXT                                               <045>
219 GOSUB 110:OPEN 2,8,2,"#":GOSUB 110                <010>
220 S=1:C=1:N$=CHR$(0):NN$=N$+N$+N$                   <157>
223 FOR I=1 TO 10:NU$=NU$+NN$:NEXT                    <234>
225 AN$(0)=CHR$(128)+CHR$(18)+CHR$(1):NA$(
    0)="----------------"                             <176>
226 RE$(0)=NN$+NN$+NN$+N$+N$                          <003>
230 PRINT#1,"U1 2 0 18"S:PRINT"{HOME,LIG.R
    ED}"S"{LEFT,SPACE}"                               <101>
235 GET#2,T$:GET#2,S$:S=ASC(S$+N$)                    <089>
240 FOR BP=0 TO 7:PRINT#1,"B-P 2";BP*32+2             <217>
270 SYS 52992,2,3,X$:AN$(C)=X$:IF LEFT$(X$
    ,1)=N$THEN NEXT:GOTO 320                          <049>
290 SYS 52992,2,16,X$:NA$(C)=X$:SYS 52992,
    2,11,X$:RE$(C)=X$                                 <036>
300 ZU(C)=C:PRINT"{HOME,4RIGHT}"C"{LEFT,SP
    ACE}":C=C+1:NEXT                                  <206>
320 IF T$<>""THEN 230                                 <028>
400 CLOSE 2:POKE 650,128:PRINT"{HOME,2DOWN
    ,BLACK}>{CYAN,UP}";:CP=2:NO=0:EN=C:IF
    EN>23 THEN EN=23                                  <033>
410 PRINT CHR$(13)TAB(2)NA$(NU+1);:NU=NU+1
    :IF NU<EN-1 THEN 410                              <046>
440 GET TA$:IF TA$=""THEN 440                         <197>
450 IF TA$="{F3}"THEN IF CP>2 THEN CP=CP-1
    :SYS 53056,0,2,0,23,0                             <077>
455 IF TA$="{F3}"THEN IF CP=2 THEN GOSUB 1
    40                                                <223>
460 IF TA$="{F5}"THEN IF CP<EN THEN CP=CP+
    1:SYS 53056,1,2,0,23,0                            <027>
465 IF TA$="{F5}"THEN IF CP>=23 THEN GOSUB
     130                                              <231>
470 IF TA$="{F1}"THEN 1000                            <002>
480 IF TA$="{F2}"THEN IF C<145 THEN F=1:C=
    C+1:PRINT"{LIG.RED,HOME,4RIGHT}"C-1"{L
    EFT,SPACE,CYAN}":EN=EN+1:IF EN>23 THEN
     EN=23                                            <128>
485 IF TA$="{F2}"THEN IF F=1 THEN F=0:TE=0
    :TE$=NA$(0):GOTO 1010                             <150>
490 IF TA$="{F8}"THEN 3000                            <045>
495 IF TA$="{HOME}"THEN 3040                          <057>
500 GOTO 440                                          <006>
1000 TE=ZU(NO+CP-1):TE$=NA$(TE)                       <238>
1005 SYS 53056,0,CP,2,23,18:IF C>23 THEN E
     N=EN+1:GOSUB 135:EN=EN-1                         <173>
1007 FOR I=NO+CP-2 TO C-1:ZU(I+1)=ZU(I+2):
     NEXT                                             <072>
1010 POKE 214,CP:POKE 211,19:SYS 58732                <217>
1020 PRINT"TT"TE$                                     <102>
1040 GET TA$:IF TA$=""THEN 1040                       <101>
1050 IF TA$="{F3}"THEN IF CP>2 THEN CP=CP-
     1:SYS 53056,0,2,19,24,37:SYS 53056,0,
     2,0,23,0                                         <235>
1055 IF TA$="{F3}"THEN IF CP=2 THEN GOSUB
     140                                              <061>
1060 IF TA$="{F5}"THEN IF CP<EN THEN CP=CP
     +1:SYS 53056,1,2,19,24,37:SYS 53056,1
     ,2,0,23,0                                        <033>
1065 IF TA$="{F5}"THEN IF CP=23 THEN GOSUB
     130                                              <152>
1070 IF TA$="{F1}"THEN 2000                           <104>
1075 IF TA$="{F4}"THEN TE$="{16RIGHT}":C=C
     -1:EN=C:IF EN>23 THEN EN=23                      <201>
1076 IF TA$="{F4}"THEN PRINT"{HOME,LIG.RED
     ,4RIGHT}"C-1"{LEFT,SPACE,CYAN}":GOTO
     2050                                             <027>
1080 GOTO 1040                                        <132>
2000 SYS 53056,1,CP,2,23,18                           <122>
2040 FOR I=C-1 TO NO+CP-1 STEP-1:ZU(I+1)=Z
     U(I):NEXT:ZU(NO+CP-1)=TE                         <093>
2050 POKE 214,CP:POKE 211,2:SYS 58732                 <199>
2060 PRINT TE$"{19SPACE}"                             <149>
2199 GOTO 440                                         <183>
3000 OPEN 2,8,2,"#":T=18:FOR I=0 TO INT((C
     -2)/8):IF 8*I+8>=C-1 THEN T=0                    <147>
3010 PRINT#1,"B-P 2 0":PRINT#2,CHR$(T)CHR$
     (SN(I+2))                                        <165>
3020 FOR BP=0 TO 7:PRINT#1,"B-P 2";BP*32+2            <203>
3023 IF BP+8*I+1>=C THEN PRINT#2,NU$;:NEXT
```

```
    :GOTO 3030                                    <082>
3025 PRINT#2,AN$(ZU(BP+8*I+1))NA$(ZU(BP+8*
    I+1))RE$(ZU(BP+8*I+1));:NEXT                  <040>
3030 PRINT#1,"U2 2 0 18"SN(I+1):PRINT"{LIG
    .RED,HOME}"TAB(36)SN(I+1)"{LEFT,SPACE
    ,CYAN}":GOSUB 110:NEXT                        <032>
3040 CLOSE 1:CLOSE 2:RUN                          <235>
10000 DATA 1,4,7,10,13,16,2,5,8,11,14,17,3
    ,6,9,12,15,18                                 <072>
10001 DATA 32,253,174,32,158,183,32,30,225
    ,32,253,174,32,158,183,138,72,32,253 <099>
10002 DATA 174,32,139,176,133,73,132,74,32
    ,163,182,104,32,117,180,160,2,185             <074>
10003 DATA 97,0,145,73,136,16,248,200,32,1
    8,225,145,98,200,196,97,208,246,32    <169>
10004 DATA 204,255,96,0,0,0,0,0,32,245,207
    ,138,72,32,245,207,224,0,176,3,76             <032>
10005 DATA 72,178,224,24,176,249,134,251,3
    2,245,207,224,0,144,240,224,39,176    <109>
10006 DATA 236,134,253,32,245,207,224,25,1
    76,227,134,252,232,138,56,229,251      <137>
10007 DATA 144,218,240,216,133,250,32,245,
    207,224,40,176,207,228,253,144,203     <197>
10008 DATA 134,254,104,170,165,172,72,165,
    173,72,165,174,72,165,175,72,224,0     <108>
10009 DATA 208,22,166,251,198,250,240,44,3
    2,240,233,232,189,240,236,133,172      <223>
10010 DATA 181,217,32,219,207,48,236,202,2
    40,3,76,72,178,166,252,198,250,240     <054>
10011 DATA 16,32,240,233,202,189,240,236,1
    33,172,181,217,32,219,207,48,236,164 <107>
10012 DATA 254,32,240,233,32,36,234,169,32
    ,145,209,136,196,253,16,249,76,88      <232>
10013 DATA 233,41,3,13,136,2,133,173,32,22
    4,233,164,254,177,172,145,209,177      <205>
10014 DATA 174,145,243,136,196,253,16,243,
    96,32,253,174,76,158,183               <199>
```

Listing 11. Schluß

```
0 "64'ER 1/86          " 64'ER
16    "1. PRG"              PRG
7     "MSE V1.0"            PRG
58    "APFELMANN"           PRG
13    "HARDCOPY KOALA"      PRG
12    "2. PRG"              PRG
69    "PROFIPRINT"          PRG
4     "APFELROUTINEN"       PRG
69    "LYRIC 3.0"           PRG
12    "CHECKSUMMER V3"      PRG
32    "APFELMANN.PIC"       PRG
41    "0. PRG"              PRG
26 BLOCKS FREE.
READY.
```

Bild 3. Ein normales Diskettendirectory

```
0 "64'ER 1/86          " 64'ER
12    "CHECKSUMMER V3"      PRG
7     "MSE V1.0"            PRG
0     "----------------"    DEL
13    "HARDCOPY KOALA"      PRG
0     "----------------"    DEL
69    "LYRIC 3.0"           PRG
0     "----------------"    DEL
41    "0. PRG"              PRG
16    "1. PRG"              PRG
12    "2. PRG"              PRG
0     "  HYPRA PLATOS  "    PRG
0     "----------------"    DEL
69    "PROFIPRINT"          PRG
0     "----------------"    DEL
```

Bild 4. So kann Ihr Directory nach der Behandlung mit dem Directorysorter aussehen

```
4     "APFELROUTINEN"       PRG
58    "APFELMANN"           PRG
32    "APFELMANN.PIC"       PRG
26 BLOCKS FREE.
```

Bild 4. Schluß

Directory mit System

Nach beendetem Einlesen erscheint links vom obersten File ein schwarzer Pfeil, der sich mit den Funktionstasten F3 nach oben und mit F5 nach unten bewegen läßt. Mittels F1 kann nun ein mit diesem Pfeil gekennzeichnetes Feld nach rechts herausgeschoben und mit F3 und F5 verschoben werden. An gewünschter Stelle wird es mit F1 wieder eingesetzt. Somit ist ein beliebiges Vertauschen aller Files im Directory möglich.

F8 schreibt das sortierte Directory wieder auf die Diskette zurück. Will man das geänderte Directory nicht gespeichert haben, so kann statt dessen mit F6 noch einmal das alte oder ein anderes eingelesen werden.

Zur optischen Abgrenzung mehrerer Files dient der Trennstrich, den man mit F2 erzeugen, mit F3 und F5 verschieben und schließlich mit F1 einfügen kann. Wem übrigens ein anderer Trennstrich besser gefällt, der kann in Zeile 260 für die Minuszeichen andere einsetzen, zum Beispiel SHIFT und +. SHIFT und C ist nicht zu empfehlen, da dieser Strich bei Groß-/Kleinschrift-Umschaltung ein großes C ergibt. Der Trennstrich belegt keinen Block auf der Diskette und ist zur besseren Unterscheidung mit DEL im Directory gekennzeichnet. Ein versehentlich mit F2 erzeugter Trennstrich kann, wenn er nach rechts gebracht wird, mit F4 wieder gelöscht werden. In gleicher Weise können auch Files aus dem Directory gelöscht werden. Dabei werden aber die von dem File belegten Blöcke nicht wieder freigegeben, so daß ein abschließendes VALIDATE bei der Diskette erforderlich ist, falls ein File gelöscht wurde.

Hier noch eine Zusammenstellung der Funktionstastenbelegung:

F1 - Eintrag markieren/einordnen
F2 - Trennstrich erzeugen
F3 - Gekennzeichnetes File aufwärts bewegen
F4 - Gekennzeichnetes File löschen
F5 - Gekennzeichnetes File abwärts bewegen
F6 - Directory nochmal einlesen
F8 - Geändertes Directory auf Diskette zurückschreiben

(Edwin Göbel/dm)

