

# Tips & Tricks für Profis

Nachdem die Anfänger unter den C 64-Fans nun ihre eigene Rubrik haben, sollen natürlich die Profis auch nicht zu kurz kommen. Sicher wird auch für diese Lesergruppe noch der eine oder andere unbekannte Trick dabei sein.

Wer sich als echter Profi bezeichnet, wird bei der Durchsicht unserer Tips & Tricks-Sammlung sicher ab und zu in ein leichtes Schmunzeln verfallen. Es ist bei manchen Beiträgen nämlich gar nicht so leicht, sie eindeutig den Anfängern beziehungsweise den Profis zuzuordnen. Wir hoffen aber trotzdem, eine gute Auswahl getroffen zu haben.

Nun unsere Bitte an Sie: Helfen Sie den vielen Anfängern unter unseren Lesern mit kleinen, leicht verständlichen Tricks, die vielleicht für Sie schon ein »alter Hut« sein mögen, den Einsteigern jedoch eine große Hilfe sein werden. Jeder Beitrag, ob Einzeiler, Programmiertrick oder sonstiges, ist willkommen. Achten Sie auch bitte auf eine leicht verständliche, einfach geschriebene Anleitung.

(tr)

## OLD-Funktion für Variablen

Dieses Programm (Listing 1) holt nicht nur ein Basic-Programm nach NEW, CLR oder einem Reset zurück, sondern auch sämtliche Variablen, Strings und Felder.

Da arbeitet man an einem phänomenalen Basic-Programm und tippt »CLR« statt »CLOSE« oder »NEW« statt NEXT in seinen Computer ein, oder dieser hängt sich selbst auf — und die Variablen sind weg, die womöglich den Grund für den Absturz in sich bergen!

Oder: Man lädt ein Maschinenprogramm in den \$C000-Bereich, und nach »RUN« erfolgt ein »?OUT OF MEMORY ERROR«, weil die Basic-Ende-Zeiger verstellt wurden.

Das Maschinenprogramm, welches dabei entstand, steht ab \$C000 im Speicher (kann verschoben werden), benötigt 115 Byte Speicherplatz und wird einfach mit »SYS 49152« aufgerufen.

Es können, wie gesagt, alle Werte und sogar definierte Funktionen und Felder zurückgeholt werden. Allerdings bestehen da noch drei kleine Bedingungen:

1. Es muß vorher bereits ein Basic-Programm im Speicher gestanden haben (mindestens: »10 :«).
2. Es darf vor Aufrufen der Routine kein Variablen-Programm eingegeben oder ein Feld abgefragt worden sein.
3. Die erste Dimensionierung eines Feldes muß ein Stringfeld sein. Beispiele:

— »DIM DE(20),DE\$(20)« muß umgestellt werden zu  
»DIM DE\$(20),DE(20)«

— »DIM E(34)« muß ergänzt werden zu

»DIM A\$(0),E(34)« (A\$(0) genügt schon!!)

Diese Ergänzung braucht der Computer nämlich, um zu erkennen, daß in die Array-Behandlung gesprungen werden muß, sonst entsteht ein Fehler.

### Funktionsweise

Der Computer untersucht das erste Byte nach dem Basic-Programm-Ende. Wenn dieses ein Buchstabe ist, prüft er es auf Stringvariable. Wenn nicht, wird der Zähler einfach um 7 Byte (zwei Namensbyte und fünf Informationsbyte) erhöht und diese übergangen.

Wenn aber eine Stringvariable gefunden wurde, untersucht er Byte 6 und 7. Diese sind bei einem String nämlich immer Null, weil sie hier nicht gebraucht werden; bei einem Feldkopf hingegen ist die Anzahl der Elemente der Dimensionierung des folgenden Feldes darin abgelegt: Wenn der Computer also eine numerische als erste Dimensionierung findet, hält er sie für eine normale numerische Variable (denn eine andere Erkennung ist hier leider nicht möglich!) und nicht für den ersten Feldkopf der Arrays, gibt einen falschen und unsinnigen Wert für die meist nicht vorhandene einfache Variable und findet auch die restlichen Felder nicht mehr!

Wenn keine Buchstaben mehr gefunden werden, wird noch eine Garbage-Collection durchgeführt, um die Stringzeiger wieder nachzustellen.

Beachten Sie bitte:

Falls Sie zum Beispiel eingeben

```
10 DIM A$(30),B$(10)
20 CLR
30 DIM A$(30)
40 SYS 49152
```

wird automatisch A\$(30), aber auch B\$(10) zurückgeholt!!

Tip: Das Programm kann auch nach einem »NEW«, »CLR« oder »Reset« geladen werden. Sie können das Programm auch auf anderen Speicherbereichen laufen lassen (zum Beispiel im Kassettenpuffer). Dazu müssen Sie die beiden Sprungbefehle

```
C04B JMP C018
C06A JMP C04E
```

jeweils anpassen (und die SYS-Adresse). Mit dem SMON ist das Verschieben ja kein Problem.

Übrigens: Bild 1 zeigt das dokumentierte Assemblerlisting von »RE-CLR«.

(Andreas Blödow/tr)

LABEL	LOC	CODE	STATEMENT	
STPRG	0000		* = \$C000	
Old	C000	A0 01	LDY #01	
	C002	98	TYA	
	C003	91 2B	STA (2B)Y	eine »NEW-Nul« löschen (mit 1 überschreiben)
	C004	20 33	JSR A533	Basic-Zellen neu binden
	C008	18	CLC	
	C009	A5 22	LDA 22	
	C008	69 02	ADC #02	
	C009	65 2D	STA 2D	
	C00F	A5 23	LDA 23	
	C011	69 00	ADC #00	
	C013	85 2E	STA 2E	
	C015	20 60	A6 JSR A660	zum CLR-Befehl
Variablen	C018	A0 00	LDY #00	
	C01A	B1 2F	LDA (2F)Y	1. Byte nach Programmende retten
	C01C	AA	TAX	Bit 7 löschen
	C01D	29 7F	AND #7F	Buchstabe?
	C01F	20 13	JSR B113	nein, dann fertig
	C022	90 49	BCC C06D	
	C024	8A	TXA	
	C025	0A	ASL	war Bit 7 gesetzt?
	C026	B0 10	BCS C038	ja, kein String (sondern Integervariable oder Funktion), Arraykennungsschleife überspringen
	C028	C8	INY	2. Namensbyte
	C029	B1 2F	LDA (2F)Y	kleiner als \$80, also REAL-Variable, kein String
	C02B	10 08	BPL C038	
	C02D	A0 05	LDY #05	
	C02F	B1 2F	LDA (2F)Y	6. Byte des Strings
	C031	D0 1B	BNE C04E	ungleich Null => Array-Feldkopf, kein String
	C033	C8	INY	
	C034	B1 2F	LDA (2F)Y	7. Byte des Strings
	C036	D0 16	BNE C04E	ungleich Null => Array-Feldkopf, kein String
	C038	A0 07	LDY #07	
	C03A	E6 2F	INC 2F	
	C03C	D0 02	BNE C040	
	C03E	E6 30	INC 30	7 Bytes überlesen
	C040	88	DEY	
	C041	D0 F7	BNE C03A	
	C043	A5 2F	LDA 2F	
	C045	B5 31	STA 31	
	C047	A5 30	LDA 30	Array-Ende = Variablen-Ende
	C049	B5 32	STA 32	und weitermachen: nächstes Element prüfen
	C04B	4C 18	C0 JMP C018	
ARRAY	C04E	A0 00	LDY #00	
	C050	B1 31	LDA (31)Y	1. Byte nach Variablenende
	C052	29 7F	AND #7F	Bit 7 löschen
	C054	20 13	JSR B113	Buchstabe?
	C057	90 14	BCC C06D	nein, dann fertig, keine Felder mehr
	C059	C8	INY	
	C05A	C8	INY	
	C05B	18	CLC	
	C05C	B1 31	LDA (31)Y	Feldlänge Low-Byte
	C05E	65 31	ADC 31	zu Array-Ende-Zeiger addieren und merken
	C060	AA	TAX	
	C061	C8	INY	
	C062	B1 31	LDA (31)Y	Feldlänge High-Byte
	C064	65 32	ADC 32	zu Array-Zeiger addieren
	C066	B5 32	STA 32	Feld überlesen und
	C068	B6 31	STX 31	Array-Zeiger auf Feld-Ende positionieren
	C06A	4C 4E	C0 JMP C04E	und weitermachen
Ende	C06D	20 26	B5 JSR B526	Garbage-Collection: Stringzeiger nachstellen
	C070	4C AE A7 JMP A7AE		zurück zu Interpreterschleife

Bild 1. Das dokumentierte Assembler-Listing zu »RE-CLR«

```

programm : re-cir      c000 c073
c000 : a0 01 98 91 2b 20 33 a5 45
c008 : 18 a5 22 69 02 85 2d a5 f5
c010 : 23 69 00 85 2e 20 60 a6 4b
c018 : a0 00 b1 2f aa 29 7f 20 3d
c020 : 13 b1 90 49 8a 0a b0 10 35
c028 : c8 b1 2f 10 0b a0 05 b1 c4
c030 : 2f d0 1b c8 b1 2f d0 16 ab
c038 : a0 07 e6 2f d0 02 e6 30 14
c040 : 88 d0 f7 a5 2f 85 31 a5 12
c048 : 30 85 32 4c 18 c0 a0 00 5b
c050 : b1 31 29 7f 20 13 b1 90 57
c058 : 14 c8 c8 18 b1 31 65 31 a2
c060 : aa c8 b1 31 65 32 85 32 63
c068 : 86 31 4c 4e c0 20 26 b5 75
c070 : 4c ae a7 00 bf bf 1e 01 72

```

**Listing 1. »RE-CLR«. Verwenden Sie zur Eingabe bitte den MSE.**

## Wersiboard und Sound-Editor

Das Programm »Sound-Editor« aus der 64'er ist eines der besten (oder sogar das beste) Synthesizerprogramm, das es für den C 64 gibt.

Einzigstes Manko ist, daß man sich jedes Mal die Finger verknotet, wenn man versucht, eine Melodie zu spielen. Überdies muß man eine Ewigkeit (vergleichen mit anderen Programmen) warten, ehe eine angeschlagene Note auch tatsächlich erklingt. So erkannte ich also, daß zwei Dinge fehlten:  
 a) ein Programm, welches die gespielten Noten schneller zu Gehör bringt, und  
 b) eine professionelle Klaviatur.

Das kurze Maschinenprogramm (Listing 2) übernimmt die Abfrage des Wersiboards und ahmt im wesentlichen die Arbeitsweise der originalen Basic-Routine in Ausgabe 7/85 (Zeilen 2025 bis 2160) nach. Der größte Unterschied besteht darin, daß die Maschinenroutine eine Tabelle mit Frequenzwerten ab Speicherstelle dezimal 52948 erwartet, aus der sie die aktuelle Frequenz holt. Im Basic-Programm wird dies mit »f\*bf(sn)« gelöst. Meine Vorgehensweise hat den Nachteil, daß die Frequenztabelle bei jedem Wechsel der Frequenz (Untermenü »Stimmen-Parameter« Punkt »F«) neu angelegt werden muß.

Die Basic-Zeilen 20000 bis 20080 im Listing 3 erfüllen diese Aufgabe. Dieses Basic-Listing ist übrigens für sich als Programm nicht lauffähig und nur in Verbindung mit dem Sound-Editor (Ausgabe 7/85) sinnvoll!

Die Bedienung des wersiboardfähigen Sound-Editors ist denkbar einfach: Die normale Computertastaturabfrage bleibt erhalten. Ändern Sie die Frequenzwerte, ist die Änderung zunächst nur über diese zu bemerken; das Wersiboard ist noch auf die alten Frequenzwerte gestimmt.

Dies können Sie jedoch ändern, indem Sie die Tastenkombination »Shift-W« betätigen. Die oben erwähnte Frequenztabelle wird sodann erneuert.

Viel mehr ist beim erweiterten Sound-Editor nicht zu beachten, alle übrigen Parameter des erweiterten Sound-Editors sind direkt nach einer Änderung ohne zusätzliche Maßnahmen auch über das Wersiboard spielbar.

Sie sollten jedoch darauf achten, daß Sie keine Oktave über vier einstellen; denn der C 64 kann nur acht Oktaven darstellen. Das Wersiboard reicht aber über vier Oktaven, und rechnen Sie mal 5+4... Achten Sie also im Untermenü »Stimmen-Parameter« auf eine entsprechende Einstellung.

Ein Tip zum Sound-Editor: Schalten Sie im Untermenü Filter/Laut. immer »FiltEx« ein. Das Hintergrundrauschen wird dadurch beträchtlich vermindert.

Und nun zur Vorgehensweise bei der Eingabe der Erweiterung:

1. Laden Sie den Sound-Editor und geben Sie die zusätzlichen

Basic-Zeilen ein. Diese enthalten alle neuen Zeilen sowie jene, die nur geändert werden.

2. Speichern Sie den erweiterten Sound-Editor wieder (zum Beispiel unter dem Namen »Sound-Editor +«).

3. Laden Sie den MSE und geben Sie mit ihm die wenigen Hexazeilen ein; »SAVEn« Sie das Programm unter dem Namen »Wersi« auf der Diskette, auf der sich auch die anderen zum Sound-Editor gehörenden Routinen befinden.

4. Fertig!

Laden Sie nun ganz normal den erweiterten Sound-Editor und starten Sie ihn mit »RUN«.

Und nun viel Spaß mit dem Wersiboard plus Sound-Editor!  
 (Helge Kautz/tr)

```

PROGRAMM : WERSI      CB64 C92
CB64 : A9 72 8D E0 03 A2 07 A0 6B
CB6C : 08 BD FF DE CE E0 03 CE CC
CB74 : E0 03 2A 90 07 88 D0 F4 54
CB7C : CA D0 EC 60 AE E0 03 AD 4F
CB84 : D5 03 C9 01 D0 12 BD D4 AC
CB8C : CE BD 00 C0 BD D5 CE 8D 1A
CB94 : 01 C0 AD DB 03 8D 04 D4 32
CB9C : AD D6 03 C9 01 D0 12 BD 09
CB94 : 36 CF BD 07 C0 BD 37 CF 7D
CBAC : BD 08 C0 AD DC 03 BD 0B 55
CB84 : D4 AD D7 03 C9 01 D0 12 C1
CBBC : BD 98 CF BD 0E C0 BD 99 7C
CB84 : CF BD 0F C0 AD DD 03 BD 27
CB8C : 12 D4 AD DE 03 C9 01 D0 B4
CB84 : 08 AD 48 C0 09 01 BD 48 3C
CBDC : C0 AD DF 03 C9 01 F0 3C AC
CB84 : 4C EE CB AD E0 03 C5 0A E0
CBEC : D0 18 A9 72 85 0A A2 07 C2
CB84 : A0 08 BD FF DE C6 0A C6 E1
CBFC : 0A 2A 90 E7 88 D0 F6 CA BD
C904 : D0 EE AD 48 C0 29 FE BD 2C
C90C : 4B C0 AD D8 03 AE D9 03 4E
C914 : AC DA 03 BD 04 D4 BE 0B D7
C91C : D4 8C 12 D4 60 FF FF FF 5B

```

**Listing 2. »WERSI«. Verwenden Sie zur Eingabe bitte den MSE.**

```

1060 IF W=0 THEN W=1:LOAD" WERSI",8,1 <025>
1160 DIM V%(8,255),TN%(255),TH(49) <023>
1530 SYS 51300:GET A$:IF A$=""THEN 1530 <053>
1535 IF A$="W"THEN GOSUB 20000:GOTO 1530 <164>
3362 POKE 984+SN,C(SN):POKE 987+SN,C1(SN) <095>
3565 POKE 981+I,ABS(S(I)) <129>
3735 POKE 990,ABS(EG) <241>
3835 POKE 991,ABS(SU) <201>
4295 SYS PR,18,1;"{WHITE}SH W{GREEN,5SPAC
E}WERSIBOARD STIMMEN" <246>
7425 POKE 981+I,ABS(S(I)) <181>
7429 POKE 50334+I,C(I):POKE 984+I,C(I) <020>
7431 POKE 987+I,C1(I) <143>
7441 POKE 990,ABS(EG) <137>
7443 POKE 991,ABS(SU) <255>
7444 GOSUB 20000:RETURN <153>
8035 PRINT"BITTE 40 SEKUNDEN WARTEN" <076>
8460 FOR I=0 TO 40:TH(I)=2^(I/12):NEXT <119>
8484 F1$=CHR$(30):F1=5 <125>
8576 POKE 984+SN,C(SN):POKE 987+SN,C1(SN) <229>
8582 POKE 981+SN,ABS(S(SN)) <092>
8631 POKE 990,ABS(EG):POKE 991,ABS(SU) <131>
8666 GOSUB 20000:W=1 <075>
8880 POKE 53280,0:POKE 53281,0 <116>
20000 IF W=1 THEN SYS PR,18,1;"{WHITE}SH
W{5SPACE,RVSON}WERSIBOARD STIMMEN{RV
OFF}" <218>
20010 FOR XX=0 TO 2:ZZ=-2 <212>
20020 FOR YY=0 TO 48 <141>
20025 ZZ=ZZ+2 <163>
20030 WW=TH(YY)*BF(XX) <172>
20040 IF WW>65535 THEN WW=0 <109>
20050 SYS DD,52950+(98*XX)+ZZ,WW <044>
20060 NEXT YY,XX <040>
20070 IF W=1 THEN SYS PR,18,1;"{WHITE}SH
W{GREEN,5SPACE}WERSIBOARD STIMMEN" <121>
20080 RETURN <070>

```

**64'er Listing 3. »Sound-Editor-Zusatz«. Diese Zeilen müssen Sie zusätzlich zum Sound-Editor eintippen.**

## Paint Magic und Basic-Programme

Um Paint Magic-Bilder in einem Basic-Programm aufrufen zu können, geht man wie folgt vor:

- Laden des Bildes
- Starten des Bildes mit »RUN«
- RUN-STOP/RESTORE drücken
- Im Direktmodus »POKE 24565,96« eingeben
- Speichern mit einem Monitor (zum Beispiel SMON) von Hex \$4000 bis \$6400

Das Bild kann jetzt, wenn es zuvor absolut geladen wurde, in einem Basic-Programm mit »SYS 24513« aufgerufen werden.

Verschwinden des Bildes:

POKE 53272,21  
POKE 56576,151  
POKE 53265,27  
POKE 53270,200

(Frank Hoffmann/tr)

## Zerstörtes Directory retten

Jedem eifrigeren Benutzer der Floppy, der nicht gleichzeitig ein Übermensch ist, wird es wohl schon mindestens einmal passiert sein, daß er das Directory einer Diskette (vielleicht auch nur teilweise) »vermurkt« hat.

Sei es nun durch Formatieren (ohne ID) oder eine sonstige Unachtsamkeit. Der arme Mensch wird wohl alles daranset-

zen, das oder die Programme wiederzubekommen. Eine mögliche Vorgehensweise besteht darin, die einzelnen Blöcke der Diskette nach den Zahlen für die Anfangsadresse des Programms abzusuchen.

Diese steht im dritten und vierten Byte des ersten Blocks jedes Programms in der Form Low-Byte/High-Byte. Dies ist notwendig, damit der Computer bei absolutem Laden (mit LOAD "Name",8,1) weiß, wo er das Programm im Speicher ablegen soll.

Nun kann man natürlich jeden Block einzeln »per Hand« absuchen. Dies ist aber in Anbetracht der Tatsache, daß es der 664 gibt, nicht ratsam. Deshalb wird man sich für diesen Zweck wohl ein Programm schreiben. Hierbei wird wohl der erste Versuch eine Basic-Version sein. So auch bei mir. Ich mußte aber leider feststellen, daß es zwar gut funktionierte (kein Wunder, es war ja auch von mir!), aber auch einen sehr großen Zeitbedarf hatte.

Also mußte ein schnelleres Programm her. Denn man wartet schließlich nicht gerne, und schon gar nicht, wenn man um ein geliebtes Programm bangen muß.

Und so kam ich auf die Idee, das Problem dadurch zu lösen, daß man ein Maschinenprogramm im Floppyspeicher ablegt. Durch diese Methode läßt sich das umgehen, was beim Basic-Programm den größten Teil des Geschwindigkeitsverlustes ausmacht:

Denn da muß man jeden Block vollständig einlesen. Und bis die Zahlen mit GET verarbeitet sind und der nächste Block ge-

programm : programmsucher 0801 0cda	0991 : e6 fb 4c 4a 09 a6 f9 e8 51 0999 : e0 12 f0 fb e0 24 f0 05 3b 09a1 : 86 f9 4c 1a 09 a5 fc f0 0e 09a9 : 1a a9 d2 a0 0b a2 06 86 4b 09b1 : fd 20 31 0a 20 64 0a 48 2a 09b9 : 20 e1 09 68 c9 4a d0 03 51 09c1 : 4c 09 a9 ef a0 0b a2 7f 09c9 : 07 86 fd 20 31 0a 20 64 43 09d1 : 0a 48 20 e1 09 68 c9 4a d3 09d9 : 0d 03 4c 0f 08 4c 66 fe 9a 09e1 : 18 a2 0b a0 00 20 f0 ff e6 09e9 : a9 12 20 d2 ff a9 9a 20 f6 09f1 : d2 ff a2 28 a9 20 20 d2 32 09f9 : ff ca d0 fa a9 92 20 d2 46 0a01 : ff 60 85 f7 86 f8 a9 08 77 0a09 : 20 b1 ff a9 6f 20 93 ff 7d 0a11 : b1 f7 20 a8 ff 88 10 f8 51 0a19 : 60 a9 08 20 ae ff 60 a9 14 0a21 : 08 20 b4 ff a9 6f 20 96 2a 0a29 : ff 60 a9 08 20 ab ff 60 e4 0a31 : 48 98 48 a2 0b a4 fd 18 2a 0a39 : 20 f0 ff a9 12 20 d2 ff 74 0a41 : a9 05 20 d2 ff 68 a8 68 86 0a49 : 20 1e ab a9 92 20 d2 ff 0e 0a51 : 60 a9 04 85 fd a9 a6 a0 41 0a59 : 0b 20 31 0a 20 64 0a 20 90 0a61 : e1 09 60 a9 00 85 c6 a5 a7 0a69 : c6 f0 fc a9 00 85 c6 ad be 0a71 : 77 02 60 a2 04 dd c5 0b b2 0a79 : ca b0 fa bd ca 0b bc ce a7 0a81 : 0b 60 20 52 a2 0c 8a 0a 0a89 : 48 20 ff e9 68 aa e8 e0 60 0a91 : 19 d0 f4 a9 0b 85 f8 a2 8b 0a99 : 0d a0 00 18 20 f0 ff 60 44 0aa1 : 93 96 11 11 20 20 20 20 aa 0aa9 : 20 20 20 d5 c0 c0 cd 0ab1 : c0 c0 c0 c0 c0 c0 b0 0ab9 : c0 c0 c0 c0 c0 c0 b8 0ac1 : c0 c0 c0 c0 d0 20 20 dd 0ac9 : 20 20 20 20 dd 20 50 18 0ad1 : 52 4f 47 52 41 4d 53 41 0ad9 : 55 43 48 45 52 20 56 20 4a 0ae1 : 32 2e 30 20 dd 0d 20 20 41 0ae9 : 20 20 20 20 20 20 ad b2 45 0af1 : c0 c0 c0 c0 c0 c0 c0 f0 0af9 : c0 c0 c0 c0 c0 c0 c0 f8 0b01 : c0 c0 c0 c0 b2 bd 0d 20 f7 0b09 : 20 20 20 20 20 20 20 09 0b11 : dd 20 31 39 38 35 20 56 cc 0b19 : 44 4e 20 4d 2e 54 48 49 7a 0b21 : 45 52 45 52 20 dd 0d 20 90 0b29 : 20 20 20 20 20 20 20 20 29 0b31 : ca c0 c0 c0 c0 c0 c0 3a	0b39 : c0 c0 c0 c0 c0 c0 c0 c0 38 0b41 : c0 c0 c0 c0 c0 cb 0d 11 6a 0b49 : 11 9e 20 20 20 20 20 20 79 0b51 : 50 52 4f 47 52 41 4d 4d 86 0b59 : 2d 53 54 41 52 54 41 44 c2 0b61 : 52 45 53 53 45 20 3a 20 14 0b69 : 32 30 34 39 0d 00 20 03 3f 0b71 : 00 57 2d 4d 66 04 00 52 3d 0b79 : 2d 4d 03 93 45 2d 4d 01 75 0b81 : 00 0a 00 64 00 e8 03 10 86 0b89 : 27 7f 20 20 54 52 41 43 ef 0b91 : 4b 20 3a 20 90 00 9f 20 47 0b99 : 20 20 53 45 4b 54 4f 52 80 0ba1 : 20 3a 20 90 00 57 45 49 5b 0ba9 : 54 45 52 20 3a 20 42 49 79 0bb1 : 54 54 45 20 54 41 53 54 ca 0bb9 : 45 20 44 52 55 45 43 4b 8d 0bc1 : 45 4e 20 21 00 24 1f 19 29 0bc9 : 12 11 12 13 15 07 06 05 f7 0bd1 : 03 4e 4f 43 48 20 45 49 65 0bd9 : 4e 4d 41 4c 20 41 4e 5a a2 0be1 : 45 49 47 45 4e 20 3f 20 68 0be9 : 28 4a 2f 4e 29 00 4e 4f 36 0bf1 : 43 48 20 45 49 4e 20 44 19 0bf9 : 55 52 43 48 47 41 4e 47 97 0c01 : 20 3f 20 28 4a 2f 4e 29 78 0c09 : 00 a5 06 20 4b f2 85 46 50 0c11 : a9 00 85 07 20 7f 03 a2 4c 0c19 : 03 20 56 f5 50 fe b8 ca f6 0c21 : d0 fa 50 fe b8 ad 01 1c 97 0c29 : 9d cc 03 e8 e0 04 d0 f2 62 0c31 : 6e cc 03 6e cd 03 6e ce e0 0c39 : 03 6e cf 03 6e cc 03 6e fe 0c41 : cd 03 6e ce 03 6e cf 03 ee 0c49 : a9 00 85 30 a9 03 85 31 85 0c51 : a9 cd 85 34 20 e6 f7 a2 27 0c59 : 02 a5 52 cd ca 03 d0 09 96 0c61 : a5 53 cd cb 03 d0 02 38 cc 0c69 : 04 18 3e 00 04 ca 10 fa d6 0c71 : a6 07 e8 e4 46 f0 05 86 7f 0c79 : 07 4c 0b 03 ee 62 03 ee b5 0c81 : 62 03 ee 62 03 4c 9e fd 76 0c89 : a5 12 85 16 a5 13 85 17 93 0c91 : a5 06 85 18 a5 07 85 19 79 0c99 : 20 27 f5 60 20 18 c1 20 e0 0ca1 : 42 d0 a9 0f 8d 05 1c a9 5c 0ca9 : 00 a8 99 00 04 c8 d0 fa 23 0cb1 : a2 01 86 06 a9 e0 85 00 ee 0cb9 : a5 00 30 fc c9 02 b0 60 93 0cc1 : e8 e0 12 f0 fb e0 24 d0 b5 0cc9 : e9 20 23 c1 60 a2 00 20 1f 0cd1 : 0a e6 60 00 00 55 55 55 11 0cd9 : 55 4c 5f b4 20 cf ff c9 d7
-------------------------------------	--	---

Listing 4. »Programmsucher«. Verwenden Sie zur Eingabe bitte den MSE.

lesen werden soll, ist der Lesekopf schon weiter. Mit dieser Methode braucht man zirka eine Umdrehung.

Der Programmsucher (Listing 4) geht hier, wie schon ange deutet, einen etwas anderen Weg: Es wird nicht der ganze Block, sondern nur die zur Bestimmung der benötigten Zahlen notwendigen GCR-Byte eingelesen (siehe dazu auch 64'er 6/85, Seite 116-119).

Soviel zu den programmtechnischen Dingen. Wer sich mit Maschinensprache und insbesondere der Floppy-Programmierung auskennt, wird das Programm wohl ohnehin verstehen. Wer nicht, sollte sich den Floppy-Kurs im 64'er durchlesen.

Nun zur Bedienung des Programms:

Diese ist zum Glück sehr einfach. Das Programm wird mit »RUN« gestartet und fragt dann nach der Startadresse des zu suchenden Programms. Hier nicht die Zahl eingeben, mit der das Programm eventuell über »SYS« gestartet wird, sondern die, an der es im Speicher beginnt. Diese sind aber in den meisten Fällen identisch. Bei Programmen, die am Basic-Start stehen (dies trifft für alle zu, die mit »LOAD "Name",8« geladen werden), braucht nur »RETURN« gedrückt zu werden. Danach startet das Programm die Suche auf der Diskette. Nach etwa 14 Sekunden beginnt das Programm, wenn kein Fehler aufgetreten ist, mit der Ausgabe der (möglichen) Startblocks.

Wenn man die Startadresse(n) hat, trägt man diese am besten mit einem Disk-Monitor, zusammen mit irgendwelchen Pseudonamen (zum Beispiel X1, X2, ...), in das Directory ein und probiert dann aus, welches das richtige Programm ist. Danach kann man dann mit »SCRATCH« und »RENAME« die anderen wieder löschen und dem gesuchten den richtigen Namen geben. Validate nicht vergessen! (Martin Thierer/tr)

## 59 390 Bytes free

Mit Double-Basic (Listing 5) können Sie zwei unabhängige Basic-Programme im Speicher halten, die zusammen 59 390 Byte (!) lang sein können. Die Programme können 38 911 (PRG 1) und 20 480 Byte (PRG 2) belegen.

Durch CTRL + F1 werden beide Programme gegeneinander ausgetauscht, während alle Variablen des jeweiligen Programmes erhalten bleiben. Wenn Sie also erst Programm 1 bearbeitet haben, können Sie nach CTRL + F1 mit Programm 2 fortfahren und umgekehrt. Zusätzlich besitzt Double-Basic zwei andere, ungewöhnliche Funktionen:

1. CTRL + Pfeil links verlangsamt die Ausführungszeiten des Computers. Der Grad der Verzögerung kann durch »POKE 53024,...« eingestellt werden (255 = größte Verzögerung).
2. CTRL + RUN/STOP hält den Computer an, bis SPACE gedrückt wird. Sehr praktisch für das Listen eines Programmes: CTRL + STOP gedrückt halten und für jede nächste Zeile zusätzlich kurz SPACE drücken.

Wenn Sie Double-Basic mit dem MSE abgetippt haben, speichern Sie es. Nach dem Laden mit »8,1« und NEW können Sie Double-Basic mit SYS 52992 initialisieren (SYS 52992 ist nach jedem RUN/STOP-RESTORE notwendig!). Es stehen nun die beschriebenen Funktionen zur Verfügung.

Hinweise: CTRL + F1 ist nur möglich, wenn sich der Computer im Direktmodus befindet. Während des Listens führt der Druck dieser Tastenkombination zum Absturz! Also: erst STOP, dann CTRL + F1. Da alle Variablen erhalten bleiben, kann ein Programm nach einem Tausch mit CONT fortgeführt werden, wenn es vorher unterbrochen wurde. Es sei denn, das System hat seit dem letzten Tausch eine FOR-, NEXT- oder GOSUB-RETURN-Struktur bearbeitet. In diesem Fall reagiert der Computer, wenn er auf NEXT oder RETURN stößt, mit Un sinn, Fehlermeldungen oder gar Absturz.

Einige Daten zu Double-Basic. Double-Basic ist in den System-Interrupt eingebunden und belegt den Speicherbe-

programm : dbasic cf00 d000

```

cf00 : a9 0b bd 14 03 a9 cf bd ed
cf08 : 15 03 60 ad 8d 02 c9 04 85
cf10 : f0 03 4c 31 ea a5 cb c9 5a
cf18 : 04 f0 29 c9 39 d0 13 a0 bf
cf20 : 1e a2 ff ca ea ea ea ea 70
cf28 : ea ea ea d0 f6 88 d0 f1 37
cf30 : f0 e0 c9 3f d0 dc a9 7f 84
cf38 : 8d 00 dc a9 10 2d 01 dc 5a
cf40 : d0 fb f0 ce a5 9d c9 80 93
cf48 : d0 c8 ad fa cf c9 31 f0 39
cf50 : 08 a9 31 8d fa cf 4c 5e 47
cf58 : cf a9 32 8d fa cf 78 a9 9d
cf60 : 34 85 01 a9 01 85 fb a9 4c
cf68 : 08 85 fc a9 00 85 fd a9 1f
cf70 : a0 85 fe a2 20 20 88 cf ab
cf78 : a0 00 84 fd a9 d0 85 fe 2e
cf80 : a2 30 20 88 cf 4c a6 cf ed
cf88 : a0 00 b1 fb 85 02 b1 fd 3f
cf90 : 91 fb a5 02 91 fd e6 fb 65
cf98 : d0 02 e6 fc e6 fd d0 ea 3a
cfa0 : e6 fe ca d0 e5 60 a2 37 2c
cfa8 : 86 01 a2 18 b5 2b bc d9 b6
cfb0 : cf 9d d9 cf 94 2b ca 10 ac
cfb8 : f3 ac 21 d0 c8 a2 28 a9 f9
cfc0 : 20 9d ff 03 ca d0 fa a2 73
cfc8 : 0d bd f1 cf 9d 01 04 98 4d
cfd0 : 9d 01 d8 ca d0 f3 4c 31 bd
cfdb : ea 01 08 03 08 03 08 03 64
cfec : 08 00 58 00 58 00 58 30 46
cfef : ff 00 2a 2a 2a 2a 20 10 12 bf
cff0 : 07 00 2a 2a 2a 2a 20 10 12 cf
cff8 : 07 2d 31 20 2a 2a 2a df 43

```

Listing 5. »Double Basic«. Verwenden Sie zur Eingabe bitte den MSE.

reich von \$CF00 bis \$D000, ist also verwendbar mit Turbo-Tape und SMON (ohne Diskmonitor).

In der Interrupt-Routine (ab \$CF0B) wird auf die Betätigung einer der drei Tasten-Kombinationen geprüft. Wird der Programmaustausch aufgerufen, erfolgt zuerst noch ein Test, ob sich der Computer auch im Direktmodus befindet. Daraufhin wird der Inhalt des Speichers von 2049 bis 22528 mit dem des RAM-Bereichs unter dem ROM vertauscht. Genauso wird mit den Basic-Vektoren in den Zellen 43 bis 66 (die unter anderem die Programmlänge und den freien Speicherplatz angeben) und einer Tabelle ab \$CFD9 verfahren. Bevor in die normale Interrupt-Routine gesprungen wird, wird noch die Nummer des aktuellen Programmes angezeigt. (Julian Ziersch/tr)

## Tips für Schreibmaschinen-Profis

Jeder, der einmal einen Schreibmaschinenkurs absolviert hat, hat sich auch schon über die Vertauschung der Tasten »Y« und »Z« auf der Tastatur des C 64 geärgert. Vor allem, wenn man in einem Basic-Programm in PRINT-Zeilen einen kleinen Text schreiben möchte, ist dieser Umstand sehr störend. Eine naheliegende, wenn auch sehr umständliche Methode ist, das Betriebssystem zu ändern. Es empfiehlt sich, diese Änderung dann in ein EPROM zu »brennen«.

Um die genannten Tasten in die deutsche Norm zu bringen, ist zuerst das Betriebssystem ins RAM zu kopieren (zum Beispiel mit einem Monitor), dann sind die folgenden POKEs einzugeben:

```

POKE 60301,89
POKE 60314,90
POKE 60366,217
POKE 60379,218
POKE 60431,183
POKE 60444,173

```

Übrigens: Wenn das Betriebssystem ins RAM kopiert wurde, besteht im allgemeinen das Problem, diese Kopie dauerhaft eingeschaltet zu lassen. Wenn man nämlich RUN/STOP-RESTORE drückt, wird das RAM aus- und das ROM wieder eingeblendet. Durch einen POKE 64982,53 wird dies verhindert. (Stefan Zellin/tr)

## Tips zum Gedichtgenerator

Das Programm »Lyrik« aus der Ausgabe 11/85 ist ja wirklich sehr schön, vor allem ist die Programmstruktur so gewählt, daß problemlos eigene Erweiterungen eingebaut werden können. Im Programm sind jedoch keine Bereichskontrollen für die Anzahl der Verse vorgesehen. Hier muß man selbst aufpassen. Auch sollte im Wortschatz wegen Zeile 371 kein »Ei« vorkommen! Nebenbei: Die Satzstrukturen passen größtenteils nicht zu den angegebenen Mnemonic-Strukturen. Ein richtiger Fehler ist jedoch in Zeile 196, dort muß es heißen:

for a=1toaz-1 ....

Geben Sie mal folgende Struktur ein: 2 Strophen, 1. Strophe 10 Verse, 2. Strophe 2 Verse, und geben Sie eigene Mnemonics ein. Es klappt nicht! Die Verse der zweiten Strophe erscheinen auch in der ersten Strophe. Der Grund ist die Codierung der Indizes von mn\$. Zur Abhilfe muß ein eigener Verszähler vz eingeführt werden:

```

107  vz=-1:....
181  vz=vz+1:inputmn$(vz):....
192  vz=vz+1:inputmn$(vz):...
196  fora=1toaz-1:forb=Otova%(0)-1:vz=vz+1:mn$(vz)=mn$(b):
      nextba:goto300
242  vz=vz+1:mn$(vz)=pm$(.....
301  ....:vz=-1
305  ....:vz=vz+1:w$=mn$(vz):....
816  fora=0tor-1:b=int(rnd(1)*q0):vz=vz+1:mn$(vz)=co$(b):next
818  goto196

```

Durch eine weitere sehr kleine Änderung kann mit VRB10 beziehungsweise VRB20 ein Imperativ wie »höre« eingebaut werden:

```

418  ifepthen....
419  e$(z2)=left$(vb$(wa,1),len(vb$(wa,1))-1)
420  ifright$(e$(z2),1)<>"e"thene$(z2)=e$(z2)+"e"
421  z1=z1+6:goto710

```

Folgende kleine Änderung ergibt ein neues Mnemonic, das einen sehr viel lebendigeren Versaufbau erlaubt. ADV erzeugt ein Adverb, somit sind Verse wie »Der Blitz lächelt schrecklich« (sub2001 vrb11 adv) möglich.

```

462  rem --- adv ---
464  ifm$<>"adv"then468
466  wa=int(rnd(1)*q4):e$(z2)=left$(no$(wa,2),len(no$(wa,2))-1)
467  z1=z1+4:goto710
468  :

```

Wie das Listing zeigt, werden die Adverbien aus den Adjektiven erzeugt, das schränkt die Wahl der Adjektive geringfügig ein: »größte« oder »dunkle« kann nicht mehr verwendet werden. Noch zwei Vorschläge für neue Mnemonics: FIX setzt einen vorher eingegebenen String ein, so lassen sich Gedichte für »unsere Oma« erzeugen! KMP setzt eine Vergleichsstruktur wie »so sehr wie« oder »größer als« ein. Strukturen wie »schöner als« oder »so blutrüinstig wie« erzeugt das Programm selbst unter Verwendung der Adjektive. Deshalb dürfen dort zum Beispiel »große« »gute« »viele« und ähnliche nicht auftreten. Auch »scharfe« oder »fromme« sind nicht möglich. Eine Einschränkung zwar, aber der Gewinn ist größer!

```

468  rem --- fix 469ifm$=--fix--thene$(z2)=fi$:z1=z1+4:
      goto710
470  rem --- vergleich ---
471  ifm$<>"kmP"→then700
472  wa=int(rnd(1)*(q9+4)):ifwa>=q9then475
473  e$(z2)=v9$(wa)
474  z1=z1+4:goto710
475  w1=int(rnd(1)*q4)
476  if wa>q9+1thene$(z2)=no$(w1,2)+"r als ":"goto474
477  e$(z2)="so "+left$(no$(w1,2),len(no$(w1,2))-1)+"wie":goto474

```

Für das Mnemonic »FIX« sind noch folgende Zeilen im Programm nötig:

```

99  fi$="der C 64"
150  Print:Print:inPut"String für FIX";fi$

```

Das Mnemonic KMP benötigt noch folgende Zeilen:

```

11  q9==xx Für xx müssen Sie die Anzahl der von Ihnen ab Zeile
      9000 eingegebenen Vergleiche einsetzen.
26  ....v9$(q9-1)
65  fora=0toq9-1:readv9$(a):next:rem vergleich
      9000 data besser als, so gut wie, und so weiter.

```

Wichtig: Um auch beim automatischen Dichten in den Genuß der eigenen Mnemonics zu kommen, sollten die Satzstrukturen ab Zeile 1001 überarbeitet beziehungsweise ergänzt und mit Strukturen, die die neuen Mnemonics enthalten, versehen werden.

Wenn es Sie stört, daß Komma und Leerzeichen vertauscht gedruckt werden, daß am Versende auch nach einem Komma ein Punkt gesetzt wird und daß bei mehr als 80 Zeichen pro Vers der Ausdruck falsch formatiert wird, dann helfen Ihnen folgende Zeilen:

```

712  forx=0toz2-1:p$=p$+e$(x)+"":ife$(x)", "then714
713  iflen(p$)>3thenp$=left$(p$, len(p$)-4)+e$(x)
715  ifright$(p$,2)="," thenp$=left$(p$, len(p$)-1):goto722
716  p$=left$(p$, len(p$)-1)+""
722  printp$
723  ifdr$<>jthen739
724  x=left$(p$):y=y:ifx<81then738
725  x=x-1:ifmid$(p$,x,1)<>""orx>80then725
726  print #1, left$(p$,x-1)
727  print #1, spc(78+x-y):mid$(p$,x+1,y-x)
730  goto739
738  print #1,p$
739  p$=""":nextb:print:ifdr$="j"thenprint #1

```

Haben Sie sich auch schon geärgert, wenn Sie lange Mnemonic-Ketten eingegeben haben und das Programm lapidar mit »UNKNOWN MNEMONIC« aussteigt? Dann helfen Ihnen die folgenden Zeilen. Nach Erkennen eines Fehlers wird die fehlerhafte Versstruktur ausgegeben, nach Berichtigung wird das Gedicht neu erzeugt.

```

700  print"ERROR...UNKNOWN MNEMONIC:"m$:goto13000
13000  close1:gosub13100
13010  forx=0to2:e$(x)=""":next:goto300
13100  print"(home,11 Cursor down,2 Spaces)"mn$(vz)
13101  input"(home,11 Cursor down)":mn$(vz)
13102  return

```

Wenn Sie zusätzlich noch Groß- und Kleinschrift einführen wollen, müssen Sie sämtliche DATA-Zeilen ab 2060 anpassen! Die DATA-Elemente müssen dabei in Anführungszeichen stehen, da sonst Großbuchstaben nicht übernommen werden. Dieser Aufwand lohnt sich wohl nur, wenn Sie sowieso den Wortschatz ändern wollen. Folgende Zeilen müssen dann zusätzlich geändert werden:

```

304  .... open1,4,7
714  nextx:x=asc(left$(p$,1)):x$=chr$(xor128):p$=x$+mid$(p$,2)

```

Man könnte nun daran denken, eine Korrekturmöglichkeit der Mnemoniczeilen vorzusehen. Weiter wäre es sicher reizvoll, von gelungenen Gedichten die Struktur speichern beziehungsweise laden zu können. Auch wäre daran zu denken, bei der Struktureingabe eine Abfrage des Stringzählers einzubauen, um einen »illegal quantity error« zu vermeiden. Wollen Sie obiges nicht alles selbst machen, so können Sie eine leere Diskette mit Rückporto an Uli Beyenbach, Glückstr. 12, 75 Karlsruhe-21 schicken. Ich schicke Ihnen dann das verbesserte Programm zu.

(Uli Beyenbach/tr)