

Tips zu Textverarbeitungsprogrammen

Bei der Arbeit mit Textprogrammen wie »SM-Text«, »StarText« und »Vizawrite« kann man sich durch Tricks das Leben leichter machen. Wir zeigen Ihnen wie es geht.

Kann man Basic-Programme mit Textverarbeitungsprogrammen bearbeiten? Natürlich geht das, Sie brauchen nur ein Programm, das das Basic- in ein Text-File konvertiert, und diesen Vorgang nach der Bearbeitung wieder rückgängig macht. Ein solches Programm ist der »text-transposer«, dessen Listing wir nachfolgend abgedruckt haben.

Bevor wir allerdings dazu kommen, haben wir noch ein paar Tips zu »SM-Text« und »StarText«.

Tips zu SM-Text

Lädt man das »!«-Programm (Lader-Programm), so erscheint zunächst ein Menü, in dem man nochmals ein »t« für Textverarbeitung eingeben muß. Wem das zuviel ist, braucht nur im »!«-Programm in Zeile 60 folgende Erweiterung voranzustellen:

60 POKE631,84:POKE632,13: POKE 198,2:..

Es wird ein »t« im Tastaturpuffer abgelegt, und bei der Abfrage eingelesen. So ist es nun möglich, »SM-Text« in einem Stück zu laden.

Bei der neuesten Version (SM-Text+) braucht man diesen Kniff nicht, da das Programm mit »**« geladen werden kann.

Hier gibt es aber die Möglichkeit, kleine Routinen zu programmieren und diese zu speichern. Dadurch kann man einen kleinen Mangel kompensieren, der beim Speichern eines Textes auftritt. Normalerweise wird der Text dann gelöscht und man müßte ihn, wenn man weiterarbeiten möchte, wieder laden. Das Abarbeiten der einzelnen Menüpunkte können Sie dem C 64 überlassen. Nachstehend ist untereinander aufgelistet, was einzugeben ist. Daneben steht die Erklärung der Tastenoperationen:

Beim Eingeben der Routine sollte man einen Text benutzen, der be-

Eingabe	Erklärung
RESTORE 1	einschalten der Routine 1 (lesen)
zweimal Pfeil nach oben	ins Hauptmenü (egal von welcher Stelle aus)
F5	Speichern
F1	1. Text
j (RETURN)	alte Datei löschen: ja
F3	laden
F1	Text laden
einmal Pfeil nach oben	Hauptmenü
F1	editieren
RESTORE	ausschalten der Routine (Ende speichern)

reits auf Diskette vorhanden ist, damit man die Routine (die ja beim Speichern gleichzeitig ausgeführt wird), eingeben kann.

Will man den laufenden Text zwischendurch einmal speichern, so genügt das Drücken von »RESTORE« und der F1-Taste.

Ein weiterer Trick ist folgender: Es geht darum, einen Info-Text in den vorhandenen Text, den man gerade bearbeitet, einzublenden und wieder zu löschen (Fenstertechnik). In diesem Infotext können zum Beispiel die gängigsten Steuerzeichen erklärt sein. So geht man vor:

Nun kann man durch Drücken von RESTORE und der F3-Taste einen beliebigen Text einblenden und ihn durch Drücken von RESTORE und der F4-Taste wieder löschen.

Tips zu StarText

Dieses Programm ist fast als perfekt zu bezeichnen. Es fehlt jedoch der Tabulator. Hier kann man sich aber leicht behelfen.

In einer Zeile setzt man an der gewünschten Tabulatorstelle einen Punkt. Dann vervielfältigt man diese Zeile so oft wie nötig. CTRL, a, e, i; etc. Hat man sich so ein mit Punkten ausgefülltes Feld aufgebaut, kann man mit der RUN/Stop-Taste (Worttabulator) schnell von Punkt zu Punkt springen.

Basic-Programme bearbeiten

Basic-Programme sind im Speicher normalerweise als Gemisch von Klartext und Token vorhanden. Damit kann ein Textprogramm natürlich nichts anfangen. Es ist also erforderlich, das Programm im Klartext zu haben. Der LIST-Befehl benutzt eine Betriebssystem-Routine,

Eingabe	Erklärung	Eingabe	Erklärung
RESTORE 3	einschalten der Routine 3 (lesen)	RESTORE	ausschalten der Routine
zweimal Puffer nach oben	ins Hauptmenü	RESTORE 4	einschalten der Routine 4
F1	editieren	Mehrmals F3	Zeile löschen (Anzahl identisch Zeile einfügen)
F5	Textbewegung	einmal Pfeil nach oben	editieren
Mehrmals F1	Zeilen nach Belieben einfügen	RESTORE	Speichervorgang beenden
Beliebige Zeichen	Text schreiben		

die dies bewerkstelligt. Es besteht die Möglichkeit, sein Basic-Programm als Text-File zu speichern. Man gibt im Direktmodus ein:
open8,8,"NAME,PW":cmd8:list:
print #8
und schließt das File mit
close8

Ein solches Text-File kann zwar von vielen Textprogrammen geladen werden (auch als sequentielle File denkbar). Leider treten aber verschiedene unangenehme Effekte auf: Anführungsstriche verschwinden; die Formatierung stimmt nicht; Zeichen sind grundsätzlich codemäßig anders definiert.

Das Programm »text-transposer« (Listing) paßt das Text-File an das entsprechende Textprogramm an. Nun kann man mit diesem modifizierten File im Textprogramm handieren. Beim Speichern macht nun jedes Textprogramm seinen eigenen Spaghetti-Code daraus. Mit »text-transposer« kann dieses File wieder in ein lauffähiges Basic-Programm umgewandelt werden.

Programmbeschreibung und Benutzung

Im Hauptmenü besteht die Auswahl zwischen
(a) »prg-file« in »textprg-file« wandeln und
(b) »textprg-file« in »basic-file« wandeln.

Hat man sein Basic-Programm in der oben erwähnten Weise abgespeichert, wählt man (a). Dann wählt man sein Textprogramm 1 bis 3 und lädt sein Text-Quellprogramm. Danach speichert man das modifizierte Textfile (unter anderem Namen).

Liegt nach der Bearbeitung mit einem Textprogramm ein von diesem gespeicherten File vor, wählt man (b) und lädt dieses Programm. Es besteht die Möglichkeit, das zu wandelnde Text-File komplett zu übernehmen oder einzeln nach Zeilennummern zu editieren. Editiert man im Textprogramm, sollte man die READY-Meldung unbedingt stehen lassen. »text-transposer« erkennt dadurch das Ende durch einen SYNTAX ERROR. Ist dies geschehen, kann man mit RUN 61500 den »text-transposer« löschen.

Bemerkungen zu SM-Text

Bevor man das gewandelte Text-File (sequentiell) lädt, muß man die Zeilenbreite auf die längste auftre-

tende Zeile stellen (zirka 100). SM-Text hat die Eigenschaft, Buchstaben zu verlieren, wenn das Format größer ist als die eingestellte Zeilenzahl. Weiter ist die Umstellung vom deutschen Zeichensatz auf Commodore(ASCII-)Tastatur sinnvoll, aber nicht unbedingt nötig.

Hat man den Text geladen, bleibt als einzige Forderung nur, daß eine Leerzeile nach dem Textnamen stehenbleiben muß. Weiterhin sollte das Programm nicht mehr als 100 Zeilen haben, da SM-Text sonst indiskutabel langsam wird.

Bemerkungen zu StarTexter

Das Programm verarbeitet maximal 250 Zeilen ohne merkbar langsamer zu werden. Die Pfeil-nach-links-Taste wird durch das Telefon-symbol (Commodore-Taste »t«), und der Klammeraffe durch Pac-man dargestellt. Man fängt in der ersten Zeile an. Auch hier muß die READY-Meldung als Endekennzeichen am Schluß stehenbleiben.

Bemerkungen zu Vizawrite

Das Text-File wird als sequentielle File eingelesen. Dazu geht man ins Editierfeld und benutzt die Merge-Funktion: Commodore-Taste drücken, großes M, Namen eingeben, bei page ein »s« eingeben, RETURN. Nun wird das Text-File als sequentielle File geladen. Dabei sollte die Formatzeile unbedingt 75 Zeichen lang sein (sonst klappt die Rückwandlung nicht richtig).

Ist das Text-File geladen, muß man die ersten zwei Leerzeilen unbedingt stehenlassen. Man sollte sich nicht daran stören, wenn eine Basic-Zeile etwas unkonventionell abgekürzt wird (durch word-wrapping, also automatischem Zeilenumbruch); den Abschluß einer Basic-Zeile bildet immer das Dreieck-Sonderzeichen für RETURN.

Die Anführungsstriche werden von Vizawrite normalerweise eliminiert und sind daher vom »text-transposer« durch ein Sonderzeichen ersetzt. Beim weiteren Editieren des Programms kann man natürlich Anführungsstriche weiterverwenden. Beim Rücktransformieren werden die Anführungsstriche und das Sonderzeichen in jeden Fall als Anführungsstriche ausgegeben.

Funktionsweise von »text-transposer«

Der Klartext wird ab Adresse 20000 geladen (die Lade- und Speicheroutine ist angelehnt an file-copy der Test/Demo-Disk). Die Parameter sind ab Zeile 61070 aufgeführt (lb = 32, hb = 78). Je nach verwendetem Textprogramm wird etwas unelegant von außen in ein Maschinenprogramm hineingePOKEt, um eine Verzweigung zu erreichen. Bei Vizawrite und SM-Text werden die Zeichen nacheinander eingelesen. Bei StarTexter wird beim Einlesen gleich formatiert, das heißt eine Zeile wird auf 80 Zeichen mit Leerzeichen aufgefüllt.

Vor dem Speichern werden, je nach Textprogramm, die entsprechenden Parameter (a) gesetzt (Zeile 60710, 60830, 60940). Zur Erklärung der einzelnen Parameter:

(a) parameter-viza:
Text-File als sequentielle File (f\$ = "s") speichern ab Adresse 20000 (lb = 32, hb = 78); Anführungsstriche Code 34 in Code 191 wandeln.

(a) parameter-star:
Als Programm-File speichern ab Adresse 20152 (unnötige Leerzeichen); Code-Wandlung 95 in 163 und 13 in 95

(a) parameter-sm:
Als sequentielle File speichern, ab Adresse 19983; Vorschreiben der Namenskennungszeile(nn\$); Code-wandlung 34 in 191

Wünscht man weitere Code-Wandlungen, so braucht man dies nur mit POKE TA + (ALTER CODE),(NEUER CODE) zu machen. TA ist eine Tabelle, die ab 50000 im Speicher steht.

Hat man das Text-File mit einem Textprogramm bearbeitet und gespeichert, muß eine erneute Code-Wandlung erfolgen (b).

(b) parameter-viza:
Als Programm-File laden; Code-Wandlung vornehmen (Zeile 60780, 60790); dann 279 Byte überlesen (POKE 250/251 Anfangsadresse; nur spezifische Informationen)

(b) parameter-star:
Als Programm-File laden; durch POKE 985,234 auf Routine »verzweigen«, die den Text wieder kürzt (alle Leerzeichen nach einer Zeile entfernt); Code-Wandlung 95 in 13, 163 in 95; dann die ersten beiden Byte überlesen (Sonderzeichen)

(b) parameter-sm:
Als sequentielle File laden; Code-Wandlung 191 in 34; dann die ersten

19 Byte überlesen (Namenskenntniszeile; nn\$).

Beim Editieren (Schleife 60310 bis 60390) wird durch SYS 864 eine Basic-Zeile nach der anderen auf den Bildschirm geschrieben, gleichzeitig der Code geändert, und die Zeile durch ein »Carriage Return« im Tastaturspuffer übernommen. Bei Zeilen, die länger als 80

Zeichen sind, erfolgt die Aufforderung zum Kürzen.

Die Schleife wird beendet, wenn die READY-Meldung auf dem Bildschirm erscheint, und bei dem Versuch, diese mit RETURN zu übernehmen, ein Syntax Error entsteht. Man hat nun die Möglichkeit, mit RUN 61500 den »text-transposer« zu löschen.

Hinweise zum Abtippen

Alle REM-Zeilen können weggelassen werden. Umnummerierung ist jedoch nicht ratsam, da aus dem Direktmodus heraus Zeilen direkt angesprungen werden, und durch ein Renumber nicht geändert würden (Zeilen 60380 und 61550).

(M. Schacht/og)

```

60060 PRINT CHR$(147):POKE 788,52:REM RUN/STOP BLOCKIEREN <225>
60070 AN=32:EN=78:REM LB/HB BASIC-ENDE=ANFANG TEXTSPEICHE
R <044>
60080 ZS=20000:TA=50000:REM ADRESSEN ZWISCHENSPEICHER/TAB
ELLE <063>
60090 POKE 53280,0:POKE 53281,0 <015>
60100 FOR I=1 TO 38:BL$=BL$+CHR$(32):BS$=BS$+CHR$(163):NE
XT <198>
60110 GOSUB 61220:GOSUB 61300:REM DATAS EINLESEN <130>
60120 POKE 250,80:POKE 251,195:SYS 927:REM TABELLE VORBEL
EGEN <238>
60130 : <159>
60140 REM -----MENUE/ABFRAGEN <252>
60150 PO=4:AZ=18:GOSUB 60690:PRINT CHR$(159)CHR$(14)CHR$(8):
POKE 214,1:PRINT <147>
60160 PRINT CHR$(18)LEFT$(BL$,14); "TEXT-EDITOR"; LEFT$(BL$,
13):PRINT BL$:PRINT BS$ <033>
60170 POKE 214,22:PRINT:PRINT CHR$(18)LEFT$(BL$,13); "MIT
+ ZURUECK"; LEFT$(BL$,12) <031>
60180 POKE 214,8:PRINT:FOR I=0 TO 1:PRINT ME$(I):PRINT:NE
XT <058>
60190 POKE 198,0:WAIT 198,1:GET X$:IF X$="e"THEN PRINT CH
R$(147):POKE 788,49:END <150>
60200 IF ASC(X$)<65 OR ASC(X$)>66 THEN 60190 <048>
60210 POKE 214,2:PRINT:PRINT TAB(2)RIGHT$(ME$(ASC(X$)-65)
,34) <131>
60220 ON(ASC(X$)-64)GOSUB 60420,60510 <075>
60230 PO=4:AZ=18:GOSUB 60690:GOTO 60150 <196>
60240 : <015>
60250 REM -----FILE EDITIEREN <149>
60260 POKE 214,20:PRINT:PRINT CHR$(18)"EINZELN ODER KOMPL
ETT EDITIEREN (E/K)?" <178>
60270 POKE 198,0:WAIT 198,1:GET X$:IF X$<>"E"AND X$<>"K"A
ND X$<>"e"THEN 60270 <056>
60280 IF X$="e"THEN RUN 60070 <072>
60290 POKE 788,49:POKE 49155,0:IF X$="E"THEN POKE 49155,1
:REM FLAG <016>
60300 : <075>
60310 REM.....SCHLEIFE ANFANG..... <092>
60320 PRINT CHR$(147):POKE 214,0:SYS 58640 <180>
60330 SYS 864:REM CODEWANDLUNG UND PRINT <231>
60340 AN=PEEK(5)+PEEK(250)+1:EN=PEEK(251):IF AN>255 THEN
AN=AN-256:EN=EN+1 <059>
60350 POKE 250,AN:POKE 251,EN:REM NEUE ANFANGSADRESSE <128>
60360 POKE 651,19:POKE 632,13:POKE 633,13:POKE 634,13:POK
E 198,4:IF PEEK(49155)=1 THEN POKE 198,2 <070>
60370 IF PEEK(214)>2 AND PEEK(1104)<>32 THEN POKE 198,2:P
OKE 214,3:POKE 211,0:PRINT:PRINT"KUERZEN!?" <048>
60380 PRINT CHR$(18)"RUN 60320":END <220>
60390 REM....SCHLEIFE ENDE..... <034>
60400 : <175>
60410 REM ----- (A)-PRG => TEXT <237>
60420 PO=8:AZ=3:GOSUB 60690:POKE 214,4:PRINT <208>
60430 PRINT TAB(4)CHR$(28)ME$(5):PRINT ME$(6)CHR$(34)ME$(7)
CHR$(34):ME$(8) <162>
60440 PRINT ME$(9)CHR$(159):PRINT BS$ <148>
60450 POKE 214,6:PRINT:GOSUB 60580:IF T$="e"THEN RETURN <042>
60460 GOSUB 61080:RW$="R":AD=828:GOSUB 61140:IF N$="e"THE
N RETURN <173>
60470 POKE 214,19:PRINT:ON VAL(T$)GOSUB 60720,60840,60950 <140>
60480 RW$="W":AD=885:GOSUB 61140:RUN 60070 <166>
60490 : <011>
60500 REM ----- (B)-TEXT => PRG <037>
60510 PD=8:AZ=3:GOSUB 60690 <012>
60520 GOSUB 60580:IF T$="e"THEN RETURN <213>
60530 POKE 214,19:PRINT:ON VAL(T$)GOSUB 60770,60890,61020 <060>
60540 RW$="R":POKE 936,96:AD=828:GOSUB 61140:POKE 936,234
:IF N$="e"THEN RETURN <091>
60550 GOTO 60260:REM EDITIEREN <219>
60560 : <081>
60570 REM.....AUSWAHL..... <127>
60580 POKE 214,8:PRINT:FOR I=2 TO 4:PRINT:PRINT TAB(12)ME
$(I):NEXT <051>
60590 POKE 198,0:WAIT 198,1:GET T$:IF T$="e"THEN RETURN <099>
60600 IF ASC(T$)<49 OR ASC(T$)>51 THEN 60590 <034>
60610 POKE 214,VAL(T$)*2+7:PRINT:PRINT TAB(12)CHR$(18)ME$(
VAL(T$)+1):RETURN <094>
60620 : <141>
60630 REM -----FEHLER <247>
60640 INPUT#15,FE,FE$:IF FE=0 THEN RETURN <184>
60650 POKE 214,20:PRINT:PRINT TAB(19)CHR$(28)LEFT$(FE$+BL
$,19)CHR$(159) <214>
60660 FOR I=0 TO 2000:NEXT:PO=19:AZ=3:GOSUB 60690:POKE 21
4,PO:PRINT:RETURN <012>
60670 : <191>
60680 REM -----TEILBEREICH LOESCHEN <093>
60690 POKE 214,PO:PRINT:FOR Q=1 TO AZ:PRINT BL$:NEXT:RETU
RN <041>
60700 : <223>
60710 REM ----- (A)-PARAMETER-VIZA <041>
60720 F$="S":AN=32:EN=78 <028>
60730 POKE TA+34,191 <095>
60740 RETURN <089>
60750 : <017>
60760 REM ----- (B)-PARAMETER-VIZA <220>

```

Listing. »text-transposer«