

Nun kann das Basic-Programm ohne Fehler ausgeführt werden. Wird der Programmablauf unterbrochen (durch STOP-Taste, Fehlermeldungen, Programmende und so weiter), wird das zweite Maschinenprogramm über den Basic-Warmstartvektor angesprochen. Dort wird die Zeilennummer wieder hochgesetzt, der Warmstartvektor wieder auf den normalen Wert gebracht und die Warmstartroutine angesprochen. Das Programm liegt nun wieder in der geschützten Form vor.

(Ulrich von Gaisberg/rg)

```

0 rem *****
1 rem *           u. v. gaisberg           *
2 rem *           am zuckerberg 70        *
3 rem *           7140 ludwigsburg        *
4 rem *           tel. 07141/55910        *
5 rem *****
6 for i=0to340:reada:b=b+a:poke50000+i,a
7 next i
8 if b <> 33527 then print"fehler in dat
as !":end
9 print"ok !":end
10 rem datas fuer maschinenprogramm
11 data169,0,141,32,208,141,33,208,169,1
,141,134,2,32,68,229,174,3,8,172
12 data4,8,192,0,208,7,224,2,176,3,76,20
6,195,162,0,142,134,2,169,32,32
13 data210,255,232,224,50,208,246,162,0,
189,21,196,157,0,4,232,224,29,208
14 data245,169,24,157,0,4,232,224,69,208
,246,162,0,189,50,196,157,80,4,232
15 data224,8,208,245,162,0,189,58,196,15
7,120,4,232,224,8,208,245,162,10
16 data160,0,24,32,240,255,169,19,162,13
,160,4,141,119,2,142,120,2,142,121
17 data2,142,122,2,132,198,96,162,0,189,
99,196,32,210,255,232,224,31,208
18 data245,96,32,68,229,162,10,160,0,24,
32,240,255,162,1,142,134,2,202,189
19 data66,196,32,210,255,232,224,33,208,
245,169,20,162,17,160,255,141,18
20 data8,142,29,8,140,4,8,162,0,189,130,
196,157,32,8,232,224,34,208,245
21 data96,48,18,5,13,34,148,148,148,148,
148,148,148,148,148,34,12,12
22 data9,19,20,19,3,8,21,20,26,26,76,49,
19,25,19,50,48,57,56,19,25,19,53
23 data48,49,52,48,80,82,79,71,82,65,77,
77,32,45,32,40,67,41,32,85,46,86
24 data46,71,65,73,83,66,69,82,71,32,32,
49,57,56,52,66,73,84,84,69,32,90
25 data69,73,76,69,32,48,32,85,78,68,32,
49,32,69,78,84,70,69,82,78,69,78
26 data32,33,169,255,141,4,8,169,131,162
,164,141,2,3,142,3,3,76,131,164
27 data165,2,141,4,8,169,32,162,8,141,2,
3,142,3,3,96,0

```

ready.

Programm für List- und Lösch-Schutz

Stringy: C64-Erweiterung

Stringy stellt eine Basic-Interpretererweiterung dar, die den Befehlssatz des C 64 um acht Befehle ergänzt. Mit diesen Befehlen ausgestattet, kann man sich einen Programmgenerator von Basic aus programmieren.

Das Listing zu Stringy entstand mit Hilfe von Stringy. Dabei wurden die Zahlen formatiert, die Prüfsummen berechnet und nach jeder vierten Zeile angefügt. Mit Stringy kann man Strubs-ähnliche Erweiterungen programmieren (Der Grund, weshalb ich Stringy schrieb). Man könnte auch ein Programm schreiben, das die in einem Basic-Programm vorkommenden Grafikzeichen durch die entsprechenden CHR\$-Funktionen ersetzt, damit sie im Listing besser zu erkennen sind. Auch Sprite- oder Bildschirmmasken-Generatoren sind recht einfach zu programmieren. Der wichtigste Befehl von Stringy ist der !INPUT-Befehl. Mit ihm kann man einen String, der eine Basic-Zeile mit Zeilennummer darstellt, bei laufendem Programm in das Basic-Programm übernehmen — ohne, daß dabei die Programmausführung unterbrochen wird.

Umgekehrt kann es sinnvoll sein, eine Zeile aus dem Basic-Programm herauszuholen, um sie einer Stringvariablen zuzuordnen. Dies ermöglicht der !GET-Befehl.

Damit es keine Komplikationen mit den Basic-Zeilennummern gibt, teilt der !NEXL-Befehl Ihnen die Folge der Zeilennummern mit.

Die anderen fünf Befehle dienen der Stringverarbeitung. Vier davon sind dem Sinn nach identisch mit den entsprechenden Stringoperationen aus Simons Basic, mit dem Unterschied, daß die Parameter beliebig komplizierte Ausdrücke sein können (dies gilt für alle Befehle von Stringy).

Der letzte der fünf Stringbefehle ist der !REPLACE-Befehl.

Die Stringy-Befehle

Nachfolgend bedeuten str1, str2, str3 immer Stringausdrücke und m, n, p, w, z immer numerische Ausdrücke.

!PLACE

Format: !PLACE (str1,str2)
!PLACE (str1,str2,m)
!PLACE (str1,str2,m,n)

Funktion: Bestimmung der Position, an der str2 in str1 steht. Die Angabe von m und n grenzt str1 auf einen Teilstring ein. Nur dieser Teilstring von str1 wird dann durchsucht, und nicht der ganze String. m gibt den Beginn dieses Teilstrings an, gerechnet vom Anfang von str1, n bestimmt das Ende des Teilstrings. Vorsichtig: n wird vom Ende von str1 aus gezählt, also in anderer Richtung als m.

Beispiel: »PRINT !PLACE ("COMMODORE", "O")« liefert 2 als Antwort.

»PRINT !PLACE ("COMMODORE", "O",3)« liefert 5 als Antwort, da nur in "MMODORE" gesucht wurde

»PRINT !PLACE ("COMMODORE", "E", 1, 4)« liefert 0 als Ergebnis, da "E" nicht in dem Teilstring "COMMODO" enthalten ist.

!REPLACE

Format: !REPLACE (str1,str2,str3)
!REPLACE (str1,str2,str3,m)
!REPLACE (str1,str2,str3,m,n)

Funktion: Ersetzen aller str2, die in str1 vorkommen, durch str3. Dabei kann str1, wie beim !PLACE-Befehl beschrieben, durch n und m eingegrenzt werden.

Beispiel: 10 A\$="INDEX=B\$+C\$"
20 B\$=!REPLACE(A\$,"INDEX","IN\$")

Nach Ausführung gilt: B\$="IN\$=B\$+C\$"

10 N\$="PETER PAUL MARY"

20 M\$=!REPLACE(N\$,"PETER", " ")

Nach Ausführung gilt: M\$="PAUL MARY"

!INSERT

Format: !INSERT (str1,str2,p)

Funktion: Fügt str2 in str1 ein. Die Position p bestimmt, an welcher Stelle str2 in str1 eingefügt werden soll.

Ist dabei p=0 oder p=len (str1), so wird angefügt.

Beispiel: »PRINT !INSERT ("ABCEF";"D",3)« liefert: "ABCDEF"

»PRINT !INSERT ("ABCEF";"D",0)« liefert: "DABCEF"

»PRINT !INSERT ("ABCEF";"D",5)« liefert: "ABCEFD"

!STOVER

Format: !STOVER (str1,str2,p)

Funktion: Überschreibt str1 mit str2.

Die Position, ab der str1 überschrieben werden soll, wird durch p angegeben.

Ist str2 länger als str1, oder ist wegen der Positionsangabe p ein Überschreiben nicht möglich, so erfolgt ein ILLEGAL QUANTITY ERROR.

Beispiel: »PRINT !STOVER ("GOTO XXXX", "0169",6)« liefert: "GOTO 0169"

!DUP

Format: !DUP (str,w)

Funktion: Es wird str w-mal dupliziert.

Beispiel: A\$=!DUP(" ",255) liefert einen String mit 255 einzelnen Punkten.

!INPUT

Format: !INPUT(str)

Funktion: Hat str keine Zeilennummer am Anfang, so geht der Computer in den Direktmodus über und führt str sofort aus.

Soll der Computer anschließend zum Programm zurückkehren, so muß der letzte Befehl in str ein »GOTO (Zeilennummer)« sein.

Beginnt str mit einer Zeilennummer, so wird str als Basic-Zeile in das laufende Programm eingefügt, sofern in dem Programm nicht bereits eine Zeile mit derselben Zeilennummer existiert. Andernfalls wird die betreffende Zeile vor dem Einfügen gelöscht. Wenn allerdings diese zu löschende Zeile eine noch offene FOR...TO-Anweisung oder ein noch nicht durch RETURN abgeschlossenes GOSUB enthält, so erfolgt ein CAN'T CONTINUE ERROR.

Die gleiche Fehlermeldung erscheint auch, wenn Sie eine Zeile löschen wollen, in der sich der DATA-Zeiger momentan befindet. Beispiel:

```
10 DATA56
20 READA:INPUT(STR$(10))
```

Nach RUN erfolgt ein CAN'T CONTINUE ERROR, da sich der DATA-Zeiger in Zeile 10 befindet. Durch einen RESTORE-Befehl, läßt sich diese Zeile dennoch löschen:

```
10 DATA56
20 READA:RESTORE:INPUT("10")
```

Nach Ausführung dieser beiden Zeilen ist die Zeile 10 gelöscht. Enthielt die gelöschte Basic-Zeile eine DEF-Anweisung, so gilt diese Funktion als nicht definiert. Enthielt die gelöschte Basic-Zeile eine Stringvariablenzuordnung der Art »AA\$="ABCD"« oder »A\$(n)="ABCDE"«, so ist anschließend die Variable nur noch als Leerstring definiert.

Soll der !INPUT-Befehl direkt nach einem THEN stehen, dann ist ein Doppelpunkt einzufügen »...THEN:INPUT...«

!GET

Format: !GET(z)

Funktion: Es wird die Basic-Zeile mit der Zeilennummer z in Stringformat geholt. Der Parameter z darf dabei nicht den Wert 0 haben. Beispiel:

```
10 REM !GET-DEMO
20 PRINT!GET(10);PRINT!GET(20)
30 A$=!GET(30)
40 PRINT MID$(A$,!PLACE(A$,"")+1)
```

!NEXL

Format: !NEXL(z)

Funktion: Es wird die auf z folgende Basic-Zeilenummer geholt. Hat !NEXL(z) den Wert 0, so bedeutet dies, daß auf z keine Basic-Zeilen mehr folgen. Beispiel:

```
10 REM !NEXL-DEMO
20 REM SIMULATION DES LIST-BEFEHLS
30 Z=0
40 Z=!NEXL(Z):IF Z=0 THEN END
50 PRINT !GET(Z):GOTO 40
```

Zum Schluß noch einige Daten zu Stringy. Stringy belegt den Speicher von \$c100 bis \$c85a. Der Bereich von \$c000 bis \$c0ff dient als Stringzwischenpeicher (je nach Befehl wird dieser Raum benutzt). Die Speicherplätze \$c85b bis \$c865 dienen als Zwischenpeicher für einige wichtige Betriebssystemdaten. Der unter dem Basic-ROM liegende Speicherbereich wird mitbenutzt.

(Karl Szameitat/ev)

```

0 REM STRINGY BY KARLSZAMEITAT, MUEHLENSTR 8B,
  2355 WANKENDORF
100 DATA 238,000,192,208,003,076,113,165,238,
  017,193,208,003,238,018,193
101 DATA 141,071,192,096,165,001,041,254,133,
  001,096,165,001,009,001,133
102 DATA 001,096,169,000,162,160,133,020,134,
  021,133,002,162,192,160,000
103 DATA 177,020,145,020,200,208,249,230,021,
  228,021,208,241,169,096,141,7522
104 DATA 020,167,141,038,181,160,005,169,234,
  153,209,166,136,016,250,160
105 DATA 002,153,237,164,136,016,250,169,000,
  162,193,141,072,171,142,073
106 DATA 171,169,243,141,000,003,142,001,003,
  169,122,141,008,003,142,009
107 DATA 003,169,170,141,010,003,142,011,003,
  096,032,115,000,008,201,033,6890
108 DATA 240,004,040,076,231,167,032,115,000,
  201,133,208,004,104,076,015
109 DATA 196,234,234,234,234,234,234,234,234,
  234,234,234,234,234,234,234
110 DATA 234,234,234,234,234,234,234,234,076,008,
  175,169,000,133,013,032,115
111 DATA 000,008,201,033,240,004,040,076,141,
  174,104,032,115,000,160,006,9045
112 DATA 217,077,200,240,005,136,016,248,048,
  199,185,084,200,168,032,115
113 DATA 000,217,032,200,208,187,200,201,040,
  208,243,185,032,200,072,185
114 DATA 033,200,072,076,115,000,169,000,162,
  192,141,017,193,142,018,193
115 DATA 076,016,193,138,048,010,169,000,133,
  002,032,027,193,076,058,164,7638
116 DATA 076,116,164,032,020,193,134,035,104,
  133,020,104,133,021,169,006
117 DATA 032,251,163,230,002,160,011,185,165,
  000,072,136,016,249,165,021
118 DATA 072,165,020,072,166,035,134,174,169,
  165,162,000,133,175,134,176
119 DATA 032,158,173,032,143,173,160,002,177,
  100,145,175,136,016,249,165,7306
120 DATA 100,164,101,032,219,182,198,174,240,
  011,032,253,174,230,175,230
121 DATA 175,230,175,208,219,162,001,032,121,
  000,201,041,240,006,032,253
122 DATA 174,032,158,183,134,174,162,001,201,
  041,240,006,032,253,174,032
123 DATA 158,183,134,175,076,247,174,032,230,
  193,165,073,072,165,074,072,8831
124 DATA 032,138,173,032,247,183,032,247,174,
  032,019,166,032,020,193,032
125 DATA 189,166,032,027,193,104,133,074,104,
  133,073,169,001,162,192,133
126 DATA 111,134,112,173,017,193,076,192,180,
  166,168,165,166,197,175,240
127 DATA 023,160,000,177,166,209,169,208,006,
  202,240,011,200,208,244,230,8555
128 DATA 166,208,230,230,167,208,226,024,096,
  165,174,240,025,165,175,240
129 DATA 021,198,174,198,175,024,165,174,101,
  175,176,010,101,168,176,006
130 DATA 197,165,240,005,144,003,076,072,178,
  024,165,166,101,165,056,229
131 DATA 175,056,229,168,133,175,230,175,024,
  165,166,101,174,133,166,144,9181
132 DATA 002,230,167,096,162,002,032,003,194,
  165,168,240,034,165,165,240
133 DATA 030,197,168,144,026,165,166,072,032,
  201,194,032,169,194,104,176
134 DATA 014,229,166,073,255,170,232,032,051,
  195,138,168,076,162,179,162
135 DATA 000,240,244,104,133,020,104,133,021,
  160,244,104,153,177,255,200,8729
136 DATA 208,249,165,021,072,165,020,072,198,
  002,208,003,032,027,193,096
137 DATA 162,003,032,003,194,024,165,166,133,
  034,101,165,133,037,165,167
138 DATA 133,035,032,230,193,032,201,194,032,
  169,194,144,004,165,037,133
139 DATA 166,165,166,197,034,240,015,160,000,
  177,034,032,000,193,230,034,7186
140 DATA 208,239,230,035,208,235,165,034,197,
  037,240,037,024,160,000,165
141 DATA 034,101,168,133,034,144,002,230,035,
  196,171,240,008,177,172,032
142 DATA 000,193,200,208,244,165,034,197,037,
  240,006,032,191,194,076,107
143 DATA 195,032,051,195,076,155,194,162,002,
  032,003,194,165,168,240,008,8087
144 DATA 165,165,240,004,197,174,176,003,076,
  072,178,024,101,168,176,248
145 DATA 032,125,180,138,208,001,136,202,142,
  017,193,140,018,193,160,000
146 DATA 140,000,192,196,174,240,014,177,166,
  032,000,193,200,198,165,208
147 DATA 242,165,168,240,020,152,170,160,000,
  177,169,032,000,193,200,198,8633
148 DATA 168,208,246,138,168,165,165,208,222,
  032,051,195,076,202,180,160
149 DATA 011,185,165,000,153,091,200,136,016,
  247,032,166,179,032,115,000
150 DATA 032,250,174,032,158,173,032,247,174,
  032,163,182,201,089,144,003
151 DATA 076,113,165,170,208,003,076,055,198,
  165,122,164,123,133,165,132,8466
152 DATA 166,160,000,132,167,132,168,132,169,
  232,202,240,008,177,034,153
153 DATA 000,002,200,208,245,138,153,000,002,
  202,160,001,134,122,132,123
154 DATA 032,115,000,144,003,076,019,200,032,
  107,169,032,121,165,132,011
155 DATA 032,019,166,176,003,076,216,196,032,
  072,198,176,036,166,020,164,7200
156 DATA 021,196,064,208,020,228,063,208,016,
  165,043,229,065,008,201,001
157 DATA 208,015,040,165,044,229,066,208,008,
  196,058,208,009,228,057,208
158 DATA 005,162,026,076,055,164,165,095,072,
  165,096,072,056,160,000,177
159 DATA 095,133,167,229,095,133,169,200,177,
  095,133,168,032,131,198,162,7546
160 DATA 000,056,181,045,229,169,149,045,232,
  176,002,214,045,232,224,006
161 DATA 208,239,104,133,096,104,133,095,173,
  000,002,208,005,133,011,076

```

Listing »Stringy«

```

162 DATA 031,197,024,165,049,133,090,101,011,
133,088,164,050,132,091,144
163 DATA 001,200,132,089,032,184,163,162,000,
024,181,045,101,011,149,045,6847
164 DATA 232,144,002,246,045,232,224,004,208,
239,165,020,164,021,141,254
165 DATA 001,140,255,001,164,011,136,185,252,
001,145,095,136,016,248,032
166 DATA 051,165,165,165,164,166,032,182,198,
133,122,132,123,165,065,164
167 DATA 066,032,182,198,133,065,132,066,032,
206,198,165,045,166,046,133,8216
168 DATA 034,134,035,160,000,177,034,016,007,
200,177,034,048,007,016,031
169 DATA 200,177,034,048,077,024,165,034,105,
007,133,034,144,002,230,035
170 DATA 165,035,197,048,144,221,165,034,197,
047,144,215,076,187,197,200
171 DATA 200,177,034,170,136,177,034,032,028,
199,176,028,032,050,199,145,6648
172 DATA 034,200,138,145,034,200,200,177,034,
170,136,177,034,032,050,199
173 DATA 145,034,138,200,145,034,208,189,169,
000,168,145,034,200,145,034
174 DATA 208,179,200,200,200,177,034,170,136,
177,034,032,028,199,176,232
175 DATA 032,050,199,145,034,200,138,145,034,
208,154,165,047,166,048,133,8228
176 DATA 034,134,035,228,050,208,013,197,049,
208,009,076,055,198,165,036
177 DATA 166,037,208,235,024,160,002,113,034,
133,036,200,138,113,034,133
178 DATA 037,160,000,177,034,048,231,200,177,
034,016,226,160,004,177,034
179 DATA 010,105,005,101,034,133,034,144,002,
230,035,160,000,177,034,240,6620
180 DATA 022,200,200,177,034,170,136,177,034,
032,028,199,176,034,032,050
181 DATA 199,145,034,200,138,145,034,024,165,
034,105,003,133,034,144,002
182 DATA 230,035,165,035,197,037,144,211,165,
034,197,036,144,205,176,158
183 DATA 169,000,168,145,034,240,224,160,244,
185,103,199,153,177,255,200,8065
184 DATA 208,247,032,121,000,076,231,167,186,
189,003,001,201,141,240,006
185 DATA 201,129,240,027,024,096,189,004,001,
197,020,208,007,189,005,001
186 DATA 197,021,240,241,024,138,105,007,170,
201,248,144,220,176,229,189
187 DATA 017,001,197,020,208,007,189,018,001,
197,021,240,216,024,138,105,7736
188 DATA 018,208,229,160,000,177,095,133,034,
200,177,095,133,035,165,049
189 DATA 133,036,165,050,133,037,136,177,034,
145,095,165,036,197,034,208
190 DATA 007,165,037,197,035,208,001,096,230,
034,208,002,230,035,230,095
191 DATA 208,229,230,096,208,225,196,096,144,
013,240,012,229,169,176,001,7971
192 DATA 136,024,101,011,144,001,200,096,197,
095,144,251,176,236,186,189
193 DATA 003,001,201,141,240,005,201,129,240,
028,096,189,006,001,188,007
194 DATA 001,032,182,198,157,006,001,152,157,
007,001,024,138,105,007,170
195 DATA 201,248,144,219,176,228,188,019,001,
189,020,001,032,182,198,157,7404
196 DATA 020,001,152,157,019,001,189,004,001,
188,005,001,032,182,198,157
197 DATA 004,001,152,157,005,001,024,138,105,
018,208,211,228,168,144,006
198 DATA 208,014,197,167,176,010,228,096,144,
007,208,005,197,095,176,001
199 DATA 024,096,228,052,240,004,176,023,144,
004,197,051,176,017,228,096,6562
200 DATA 144,013,240,012,229,169,176,001,202,
024,101,011,144,001,232,096
201 DATA 197,095,144,251,176,238,162,002,032,
003,194,165,168,240,004,165
202 DATA 165,208,003,076,072,178,024,166,174,
240,248,202,134,174,138,101
203 DATA 168,197,165,240,002,176,236,165,165,
032,125,180,134,033,132,034,8518
204 DATA 164,165,136,177,166,145,098,152,208,
248,024,165,174,101,033,133
205 DATA 033,144,002,230,034,164,168,136,177,
169,145,033,152,208,248,032
206 DATA 051,195,076,202,180,076,227,168,162,
001,032,003,194,165,165,208
207 DATA 003,076,072,178,032,230,193,166,174,
240,246,160,000,177,166,032,8614
208 DATA 000,193,200,196,165,208,246,202,208,
241,032,051,195,076,155,194
209 DATA 032,138,173,032,247,183,032,019,166,
160,000,144,039,177,095,133
210 DATA 020,200,177,095,133,021,177,020,240,
021,200,200,177,020,170,136
211 DATA 177,020,133,099,134,098,162,144,056,
032,073,188,076,247,174,169,8321
212 DATA 000,170,240,238,200,177,095,240,246,
166,095,165,096,134,020,133
213 DATA 021,208,215,169,255,133,058,032,121,
165,032,115,000,076,055,198
214 DATA 040,194,118,076,065,067,069,040,195,
003,069,080,076,065,067,069
215 DATA 040,195,079,078,083,069,082,084,040,
195,182,164,086,069,082,040,7129
216 DATA 199,085,085,080,040,199,167,069,088,
076,040,199,207,161,080,082
219 DATA 073,083,068,078,000,003,010,019,027,
034,039,2291
300 :
310 REM POKE/PRUEFRoutine
320 :
330 RESTORE:AD=49408:ZE=100
340 PR=0
350 READ PO:IF PO>255 THEN 370
360 PR=PR+PO:POKE AD,PO:AD=AD+1:GOTO 350
370 IF PO<>PR THEN PRINT"FEHLER<SPACE>IN<SPACE>
ZEILEN"ZE"BIS"ZE+3:END
380 IF PO=2291 THEN PRINT"DATAS<SPACE>OK<SPACE>
-<SPACE>SAVE<SPACE>PROGRAMM<SPACE>UND<SPACE>
STARTE<SPACE>MIT<SPACE>SYS49442":END
390 ZE=ZE+4:GOTO 340

```

Listing »Stringy« (Schluß). In Zeile 370/380 bei »SPACE« einfach die Leertaste drücken!