

Befehlsenerweiterung für Simons Basic

Die Fähigkeiten des Commodore 64 sind mit dem vorhandenen Befehlsvorrat des Basic 2.0 nur sehr schwer auszunutzen. Dafür bietet Commodore eine Erweiterung an, die diesen Mangel weitgehend behebt: Simons Basic. Dieser Artikel gibt eine grobe Speicherbelegung und behebt einige Mängel.

Leider muß im voraus erwähnt werden, daß nur Besitzer der Disk-Version POKEs anwenden können, da bei der Modul-Version ein unveränderbares ROM vorliegt und somit POKEs hier unwirksam sind.

Obwohl nur 8 KByte im Basic-Speicher verbraucht werden, ist Simons Basic eine 16-KByte-Basic-Erweiterung. Die zweiten 8 KByte liegen unter dem Basic-ROM im RAM, so daß Simons Basic immer trickreich zwischen beiden Ebenen umschalten muß. Zusätzlich wird auch der Bereich von \$C400-\$CBFF (dezimal 50176-52223) benutzt; zum Beispiel liegt der KEY-Funktionstastenspeicher ab \$C64D (50765).

Bei MEM (dem Befehl zum Kopieren des Original-Zeichen-ROMs ins RAM und dessen Einschalten) liegt der Zeichensatz ab \$E000 (57344) im RAM unter dem Kernal, der Bildschirm von nun an ab \$CC00 (52224), die Sprite-Pointer ab \$CFF8 (53240) und die Sprites von \$C000-\$C3FF (49152-50175).

Bei HIRES und MULTI (den Grafik- und Farbgrafik-Modi) liegt der Grafik-Speicher unter dem Kernal im RAM (\$E000-\$FFFF, 57344-65535), der Farbspeicher ab \$C000 (49152), und Sprites finden sich in den Blocks 48-63 (ab \$CC00, 52224).

Man sieht also, daß \$C000-CFFF (49152-53247) laufend belegt sind und kleine Maschinenroutinen höchstens in Sprite-Speichern plaziert werden können.

Hier erst einmal Beanstandungen zur Beschreibung einiger Befehle:

```

70 REM *****
71 REM *           P A G E           *
72 REM *****
73 REM * BERICHTIGUNG DES           *
74 REM * SIMON'S BASIC-BEFEHLS PAGE *
75 REM *****
79 K=0:PRINT" P A G E   ?CHECKSUM "
;
80 FOR I=$BFF2 TO $BFFE:READ J:POKE I,J:
K=K+J:NEXT
81 DATA 32,225,255,208,5,104,104,76,60
82 DATA 191,76,228,255
83 POKE $BF12,242:POKE $BF13,191
84 IFK=1819THENPRINT"OK":ELSE:PRINT"ERRO
R
READY.

```

Listing 2. Verbesserte Version des Page-Befehls

```

50 REM *****
51 REM *           D U M P           *
52 REM *****
53 REM * BERICHTIGUNG DES           *
54 REM * SIMON'S BASIC-BEFEHLS DUMP *
55 REM *****
59 K=0:PRINT" D U M P   ?CHECKSUM "
;
60 FOR I=$9F9C TO $9FA7:READ J:POKE I,J:
K=K+J:NEXT
61 DATA 160,4,32,57,160,136,208,250,234
62 DATA 234,234,234:POKE $80B9,221:POKE
$808C,68
63 POKE $9FD6,240:POKE $9FD7,27:POKE $9F
D8,200
64 FOR I=$9FF0 TO $9FFC:READ J:POKE I,J:
K=K+J:NEXT
65 DATA 160,4,44,160,5,169,34,32,210,255
,76,158,159
66 IFK=3409THENPRINT"OK":ELSE:PRINT"ERRO
R
READY.

```

Listing 1. Variablen-Liste mit Vorzeichen

```

10 REM *****
11 REM *
12 REM * INTERPRETERCODETABELLE *
13 REM *
14 REM *****
20 A=0:B=33762
30 A=A+1:IF A=128 THEN B=41118
40 IF A=204 THEN END
50 PRINT:PRINT A,
60 PRINT CHR$(PEEK(B) AND 127);:B=B+1
61 IF A<128 AND PEEK(B)<>64 THEN 60
62 IF A>127 AND PEEK(B-1)<128 THEN 60
70 IF A<128 THEN B=B+1
80 GOTO 30
READY.

```

Listing 3. Alle Befehle des C 64 mit Simons Basic

```

10 REM *****
11 REM *           E R R O R           *
12 REM *****
13 REM * ERGAENZUNG ZU SIMON'S BASIC *
14 REM * FORMAT: ERROR                *
15 REM * GIBT DISKSTATUS AUS          *
16 REM * > BEISPIEL: ERROR            *
17 REM * ERGEBNIS: 00, OK,00,00      *
18 REM *****
19 K=0:PRINT"██ E R R O R █ ?CHECKSUM "
;
20 FORI=$BFBB TO $BFE2:READ J:POKE I,J:K
  =K+J:NEXT
21 DATA32,115,,201,176,240,3,76,138,136
22 DATA169,8,133,186,32,180,255,169,111
23 DATA133,185,32,150,255,32,165,255,32
24 DATA210,255,201,13,208,246,32,171
25 DATA255,76,116,145
26 POKE$87BE,186:POKE$87BF,191
27 IFK=5527THENPRINT"OK":ELSE:PRINT"ERRO
R"
READY.

```

Listing 4. Disk-Status-Ausgabe

```

30 REM *****
31 REM *           J O Y           *
32 REM *****
33 REM * ERGAENZUNG ZU SIMON'S BASIC *
34 REM * FORMAT: JOY N                *
35 REM * STELLT JOYSTICKPORT EIN     *
36 REM * > BEISPIEL: JOY1 (FUER PORT1)*
37 REM *           JOY2 (FUER PORT2)*
38 REM *****
39 K=0:PRINT"██ J O Y █ ?CHECKSUM "
;
40 FORI=$BFE3 TO $BFF1:READ J:POKE I,J:K
  =K+J:NEXT
41 DATA32,115,,32,6,129,138,41,1,141
42 DATA213,136,76,42,130
43 POKE$88DA,111:POKE$88FC,16
44 POKE$8764,226:POKE$8765,191
45 IFK=1232THENPRINT"OK":ELSE:PRINT"ERRO
R"
READY.

```

Listing 5. Joystick für beide Ports

Beim **TEXT**-Befehl (Einsetzen von Text in die HiRes-Grafik-Seite) müssen Zeichentyp, Größe und Abstand als Konstanten gegeben sein, da der Text sonst nicht korrekt ausgedruckt wird. <CTRL-A> und <CTRL-B> können beliebig in einem String benutzt werden. Kein CTRL am Anfang wird als Großschrift ausgelegt. <RVS ON/OFF>-Zeichen werden auch richtig ausgeführt.

Beim Befehl **FETCH** (Eingaberoutine mit bestimmbarer Zeicheneingabe-Beschränkung) fehlt die Angabe, daß die Eingabelänge über ein Zeichen (wie angegeben) hinausgehen darf. Das Limit liegt bei 88 Zeichen. Danach ist der Eingabepuffer, in den die eingegebenen Zeichen abgelegt werden, gefüllt und dahinterliegende Systemvariablen (siehe C 64-Handbuch) können zerstört werden. Ein Fehler ist außerdem, daß nach der Eingabe des letzten Zeichens keine Korrektur (mit) mehr möglich ist. Es bleibt keine andere Wahl, als <RETURN> zu drücken.

Bei **DUMP** (Ausgabe der Inhalte aller nicht indizierten Variablen) werden die Werte leider ohne Vorzeichen ausgegeben. Dies kann man beheben mit **POKE 32953,221**. Zudem werden leere Strings (Länge 0) als Strings mit zufälligen 255 Zeichen und Integervariablen als 16-Bit-Adressen ausgegeben. Das Programm in Listing 1 beseitigt alle drei Fehler.

Im Handbuch wird bei den Bildschirmroll-Befehlen das Format leider falsch angegeben. Die Parameter werden mit der Routine geholt, die auch INV und MOVE bedient. Daher ist auch das Format das gleiche: Richtung W/B r,c,w,d.

Die Parameter entsprechen denen im Simons Basic-Handbuch in Abschnitt 7.6.

Bei **MERGE** kann es vorkommen, daß ein Programm nicht ordnungsgemäß angehängt wird. Dieser Fehler kommt nicht mehr vor, wenn man immer **OLD** vor MERGE eingibt, da dieser

Befehl das Programmende noch einmal überprüft und gegebenenfalls bereinigt.

Bei **PAGE** kann man das Listen nicht, wie in der Anleitung angegeben, durch Drücken der RUN/STOP-Taste abbrechen. Das Programm in Listing 2 behebt diesen Fehler.

Befehle, die zwar nicht im Handbuch, jedoch in der Befehlsliste im RAM zu finden sind, wurden bereits in früheren Ausgaben behandelt.

Zur Speicherung der Simons Basic-Befehle:

Die Befehle werden als Zwei-Byte-Kombination abgespeichert. Das erste Byte hat den Wert 100, und das zweite einen Wert zwischen 1 und 127. Dies ergäbe eine Befehlsmenge von 127 Befehlen, einige Tokens sind aber nicht belegt.

Um alle Befehls-codes auszugeben, kann man Listing 3 verwenden. Nach RUN wird eine Liste der Interpreter-codes ausgegeben, die direkt aus dem RAM von Simons Basic und aus dem Basic-ROM entnommen wird. Setzt man in Zeile 20 für A den Wert 127 ein, so wird nur der normale Befehls-vorrat ausgedruckt. Mit diesem geänderten Programm kann sich jeder C 64-Anwender eine Interpreter-codetabelle erstellen, die leider im C 64-Handbuch fehlt.

Nun zwei kurze Maschinenprogramme, die nach einmaligem Lauf als neue Befehle zur Verfügung stehen:

ERROR:

Ausgabe des Diskettenstatus auf dem Bildschirm (Listing 4).

JOY n:

Nach Ausführung dieses Befehls liest die Funktion JOY den Control Port n (n=1 oder 2) (Listing 5).

Die Listings 1 und 5 lassen sich zum Beispiel als Vorprogramm nach dem Start von Simons Basic laden und starten und stehen danach bis zum Ausschalten des Computers bereit.

(Dieter Temme/gk)