Kudiplo auch für den C 64

In der Ausgabe 8 des 64'er Magazins war das Programm »Kudiplo« für den VC 20 abgedruckt, das mit dem 1520-Printer-Plotter eine komplette Kurvendiskussion ausgibt. Hier sind die erforderlichen Änderungen, um dieses nützliche Programm auch auf dem C 64 laufen zu lassen.

Nach der Veröffentlichung meines Programms Kudiplo für den VC 20, erreichten mich viele Leserbriefe mit der Bitte um nähere Auskunft dazu, wie das Programm für den C 64 abzuändern ist. Probleme gab es dabei mit der Routine in den Zeilen 185, 190 und 230.

Die genannten Zeilen bewirken in der veröffentlichten Version für den VC 20 ein »Verbiegen« des Vektors für die Fehlerbehandlungsroutine. Der in den Speicherstellen $0300 und $0301 stehende Zeiger wird so verändert, daß er nicht mehr zu der im Basic-ROM stehenden Routine zur Ausgabe von Fehlermeldungen zeigt. Statt dessen zeigt er nun auf einen Sprungbefehl, der mit Hilfe der Zeile 185 in den Kassettenpuffer geschrieben wurde. Dieser Sprungbefehl führt zurück ins Basic-Programm, dessen nächste Zeile gesucht und so abgearbeitet wird, als sei kein Fehler aufgetreten.

Die Ähnlichkeit des VC 20 mit seinem großen Bruder ist oft zitiert. Auch bei ihm läßt sich eine solche Fehlerblockade einrichten. Allerdings ist beim großen Bruder zu diesem Zweck ein kleines Maschinenprogramm in den Kassettenpuffer zu schreiben, in welchem abgefragt wird, ob ein Fehler vorgekommen ist und das dann abhängig vom Ergebnis entweder zum nächsten Statement oder zur nächsten Zeile verzweigt.

Für den C 64 müssen darum die Zeilen 185 und 230 wie folgt geändert werden:

```
185 DATA 138, 48, 3, 76, 59, 169, 76, 116, 164 : FOR I=823
TO 840 : READ A : POKE I, A : NEXT
230 NEXT : POKE 768, 139 : POKE 769, 227
```

Mit diesen Änderungen läuft Kudiplo dann endlich auch auf dem »großen Bruder«.

(Jürgen Curdt/ev)

POKE mal wieder

Viele C 64-Benutzer haben sich sicher schon mit dem Basic des C 64 herumgeärgert: Egal, was man machen will, fast alles läuft über PEEK und POKE. Doch gerade diese POKEs helfen manchmal erheblich, wenn es um Probleme geht, die mit einfachen Basic-Befehlen nicht zu lösen sind.

Hier nun eine Liste von wichtigen PEEKs, POKEs und SYS-Befehlen.

1	Inhalt 55 = normal Inhalt 54 = Basic ausgeschaltet (auf RAM umgestellt) Inhalt 53 = Basic und Kernal auf RAM umgestellt. (Es empfiehlt sich dabei, das Basic und das Kernal vorher ins RAM zu POKEn, damit der Computer bei der Umschaltung nicht aussteigt.)
17	Mit diesem PEEK läßt sich abfragen, wie die letzte Variable zugewiesen wurde. Ist PEEK(17) = 00, dann war die letzte Variablenzuweisung ein INPUT, oder es hat noch keine Zuweisung stattgefunden. Ist PEEK(17) = 64, dann wurde die letzte Variable durch GET geholt. Bei PEEK(17) = 152 erfolgte die letzte Variablenübergabe durch einen READ-Befehl.
19	Durch POKE 19,64 wird beim nächsten INPUT-Befehl kein Fragezeichen mehr ausgegeben. Allerdings kann man nachher durch Drücken der RETURN-Taste nicht mehr in die nächste Zeile gelangen. Es empfiehlt sich daher, nach dem INPUT-Befehl diesen Befehl wieder mit POKE 19,0 rückgängig zu machen.
43/44	Der Anfang des zur Zeit im Speicher befindlichen Basic-Programms errechnet sich durch PEEK(43)+PEEK(44)*256.
45/46	Das Ende des Basic-Programms erhält man durch ?PEEK(45)+PEEK(46)*256.
61/62	Zeiger auf Basic-Statement für CONT: Durch PEEK(61)+PEEK(62)*256 erhält man die Speicherstelle, die nach dem zuletzt ausgeführten Basic-Befehl liegt, das heißt die Speicherstelle, von der sich der Basic-Interpreter bei CONT den nächsten Befehl holt. Tip: Bei CONT kommt öfter CAN'T CONTINUE ERROR vor, wenn man nach dem Stoppen ein CLR eingegeben oder in irgendeiner Programmzeile etwas geändert hat. Liest man die Werte mit PEEK(61) und PEEK(62) nach der Unterbrechung aus, dann macht ein CLR oder ähnliches nichts aus, wenn man vor CONT die zuvor ausgelesenen Werte wieder in die Speicherstellen POKEt.
63/64	Nummer der aktuellen DATA-Zeile: Mit ?PEEK(63)+PEEK(64)*256 erhält man die Nummer der DATA-Zeile, aus der gerade das letzte Datum geholt wurde. (Gut zum Finden von Fehlern in DATA-Zeilen geeignet.)
69/70	Name der zuletzt zugewiesenen Variable: Bei normalen Fließkommavariablen liest man

den Wert mit PRINT CHR$(PEEK(69)) + CHR$(PEEK(70)) aus.
Bei Integervariablen (zum Beispiel XY%) erhält man den Namen durch ?CHR$(PEEK(69)-128)+CHR$(PEEK(70)-128).
Strings (zum Beispiel VX$) erhält man durch ?CHR$(PEEK(69))+CHR$(PEEK)70)-128).

120 Nach Ausführung dieses POKEs nimmt der C 64 keinerlei Befehle mehr an:
POKE 120,2

147 Wenn man die LOAD-Routine im Betriebssystem anspringt, holt es sich aus der Speicherstelle 147 die Information, ob LOAD oder VERIFY durchgeführt wird.
Inhalt 0 = LOAD
Inhalt 4 = VERIFY

157 Ausgabe-Kontrolle:
Inhalt 000 = Programm-Modus
Inhalt 128 = Direktmodus
Damit bei LOAD-Befehlen vom Programm aus die Mitteilungen SEARCHING, LOADING oder VERIFYING auf dem Bildschirm erscheinen, setzt man vor dem LOAD-VERIFY- oder SAVE-Befehl ein POKE157,128.

197 Derzeitiger Tastendruck:
PEEK(197)

200 Zeiger auf Zeilenende. PEEK(200) gibt an, wieviel Zeichen die zuletzt eingegebene Zeile hatte.

204 Nach POKE 204,0 bleibt der Cursor an, auch bei GET-Befehlen. Mit POKE 207,0:POKE 204,1 kommt man dann wieder auf den Normalzustand zurück.

641-644 Start- und Endadresse des Basic-RAMs:
Durch Ändern dieser Werte kann man die Größe des Basic-RAMs verändern, zum Beispiel:
POKE 643,0 : POKE 644,128 : SYS 64764 setzt das Ende des Basic-RAMs um 8 KByte nach unten. Anderes Beispiel: POKE 641,0 : POKE 642,16 : SYS 64764 setzt das Basic-RAM um 2 KByte nach oben.

646 POKE 646,Farbwert setzt die Cursorfarbe.

653 PEEK für Shift-, Commodore- und für CTRL-Taste:
Bit 0 = Shift-Taste, Bit 1 = Commodore-Taste und Bit 2 = Control-Taste.

788/789 IRQ, Hardware-Interrupt: Das Betriebssystem springt ständig in diese Routine, durch Ändern des Inhalts kann man eigene, »interruptgesteuerte« Maschinenroutinen ständig laufen lassen.

792/793 Restore-Vektor: PEEK(792) + PEEK(793)*256 ergibt die Speicherstelle, an die bei Restore-Tastendruck gesprungen wird.
Beispiel: Bei POKE 792,226 : POKE 793,252 wird bei Drücken der Restore-Taste ein Reset ausgelöst.

828-1019 Kassettenpuffer: Nach Laden oder Verify stehen im Kassettenpuffer folgende Informationen:
828: 1 = normales File, 3 = wurde mit SAVE"Name",1,1 abgespeichert. Solche Programme werden bei LOAD automatisch ab der Adresse geladen, von der sie abgespeichert wurden.
829/830: Hier ist die Startadresse des Programms abgelegt (829 ist das Low-Byte, 830 das High-Byte).
831/832: Endadresse des Programms.
833-1019: 186 Zeichen langer Programmname (auf dem Bildschirm werden nur 16 angezeigt, aber es lassen sich bis 186 Stellen lange Programmnamen abspeichern).
Der Kassettenpuffer ist auch gut zum Ablegen eigener Maschinenprogramme geeignet, sofern mit der Floppy gearbeitet wird.

42291 Koppeladressen angleichen: Falls Programme mit NEW gelöscht wurden, kann man mit diesem SYS-Befehl die Bytes 2049 und 2050 wieder in Ordnung bringen, wenn vorher etwas anderes als 0 in diese Speicherzellen gePOKEt wird.

56320 Joystick Port 2:
WAIT 56320,16,16 wartet auf Feuerknopf
WAIT 56320,4,4 wartet auf Linksbewegung des Joysticks
WAIT 56320,1,1 wartet auf Joystick nach oben
WAIT 56320,2,2 wartet auf Joystick nach unten
WAIT 56320,8,8 wartet auf Joystick nach rechts

56321 Wie 56320, aber Joystick in Port 1.

56576 Mit PEEK(56576) kann man die Pins PB0-PB7 vom User-Port (auf der Unterseite des Ports, siehe Handbuch) auslesen. Mit POKE in diese Speicherstelle kann man auch Ausgaben über den User-Port laufen lassen.

56578 Datenrichtungsregister für User-Port: Jedes der Bits gibt die Datenrichtung für die Pins PB0-PB7 des User-Ports an. Ist das entsprechende Bit gesetzt, so fungiert der dem Bit zugeordnete Pin als Ausgang, bei nicht gesetztem Bit als Eingang.

65409 SYS 65409 setzt den Video-Chip des C 64 auf den Ursprungszustand zurück.

65493 LOAD-Routine des Betriebssystems. Mit folgender kleiner Routine kann man Unterprogramme nachladen, ohne irgendwelche Basic-Pointer (wie zum Beispiel die Zeiger auf die Endadresse, 45 und 46) zu verändern:
POKE 186,1 : POKE 780,0 :
POKE 781,0 : POKE 782,96 :
POKE 183,0 : SYS 65493
Erklärung: 186,1 = Geräteadresse für Recorder
781 und 782 gibt die Startadresse an, ab der das Programm geladen werden soll.
183,0 = kein Programmname.
SYS 65493 = LOAD-Routine.

65511 Durch SYS 65511 lassen sich alle Files schließen. So erspart man sich das lästige Eintippen von CLOSE1:CLOSE2:CLOSE3... Dabei sollte aber beachtet werden, daß so nur der Kanal geschlossen wird, aber keine Dateien auf einer Disk.

Auf PEEKs und POKEs für Grafik und Sprites wurde hier verzichtet, da die Grafik und die Sprites im Grafikkurs von H. Ponnath schon sehr ausführlich beschrieben sind.

(M. Kohlen/gk)