Betriebssystem selbst gemacht

Ein eigenes Betriebssystem zu schreiben, ist nicht jedermanns Sache. Aber warum soll man nicht das vorhandene für sich abändern?

Wie schön wäre es doch, sein eigenes, kleines Betriebssystem für den C 64 zu schreiben. Doch gewiß ist das nicht gerade jedermanns Sache. Daraus könnte sich schnell ein »Fulltime-Job« mit Beschäftigung bis ins hohe Rentenalter entwickeln. Es liegt viel näher, das vorhandene Betriebssystem einfach so lange abzuändern, bis ein eigenes, den Bedürfnissen angepaßtes Kernal entsteht. Beste Beispiele sind das Hypra-Kernal und Hypra-Perfekt (64'er, Ausgaben 12/84, 1/85 und 4/85). Bestimmt hat so mancher inzwischen schon eine kleine Kernal-Sammlung auf Diskette oder EPROM.

Ganz interessant wäre es dann, auf einfache Weise diese Systeme weiter abzuändern. Ihnen beispielsweise einen deutschen Basic-Interpreter zu verpassen, mit deutschen System- und Fehlermeldungen.

Gesagt, getan: Vorliegendes Programm erledigt dies alles. Es ist abgestimmt auf die Kernal-Version aus der 64'er-Ausgabe 1/85 sowie auf das Hypra-Perfekt-Kernal aus Ausgabe 4/85, arbeitet jedoch auch mit vielen anderen Betriebssystemen. Probieren geht hier über Studieren.

Im Gegensatz zu den oben genannten EPROM-Versionen, erzeugt dieses Programm ein lauf- und damit testfähiges Betriebssystem im freien RAM-Bereich unter dem Kernal-ROM ab $E000. Nicht jeder wird wohl gleich ein EPROM programmieren, sondern lieber zuerst etwas experimentieren wollen. Zuerst werden Basic und Kernal ins RAM kopiert (Init-Routine ab 190). Erst dann kann geändert werden. Das Programm (siehe Listing) erlaubt auch (falls vorhanden) ein Betriebssystem von Diskette zu laden (Zeile 23, siehe Tabelle 1 und 2). Danach kann die Farbkombination beim Warm- und Kaltstart ausgetestet werden. Der SYS-Befehl in Zeile 201 löst einen Reset des Bildschirms innerhalb der geänderten Kernalroutine aus, so daß die Kombination sofort beurteilt werden kann. Erst nach Bestätigen verzweigt das Programm nach Zeile 213. Dort kann eine OLD-Routine aktiviert werden. Beginn ist ab Adresse 64608. Dies gilt es zu beachten, wenn ein Kernal abgeändert werden soll, das eventuell diesen Bereich benutzt. Es handelt sich dabei um einen Teil der normalerweise überschriebenen Kassettenroutinen des Betriebssystems. Hat man ein Betriebssystem geladen, das mit der OLD-Routine kollidiert, muß letztere verschoben werden.

Die genannten Hypra-Kernal sind diesbezüglich unkritisch. Beachtet werden muß lediglich, daß sich deren Adressen für Funktionstasten-Belegung unterscheiden. Soll OLD also auch auf Funktionstaste gelegt werden, fragt das Programm selbst nach der verwendeten Version (Zeile 218). Bei anderen Kernal-Versionen kommt es auf das berühmte Experiment an. Betriebssysteme ohne eine Belegung der Funktionstasten können selbstverständlich auch nicht den OLD-Befehl per Funktionstaste unterstützen. Es fehlen ganz einfach die nötigen Routinen (siehe Original Kernal V2.0)!

Hier bleibt ja immer noch der SYS-Befehl zum Aufruf von OLD, den man sich eben merken muß. Für ganz zerstreute Freaks ist die nachfolgende Routine am idealsten. OLD als gewöhnlicher Basic-Direktbefehl (also auch abkürzbar mit »O SHIFT L«): Zeile 224 sorgt für die entsprechende Programmierung. Da hier, wie nachher auch bei den Fehlermeldungen in deutscher Sprache, das Basic mitverändert wird, muß natürlich auch das Basic-ROM (beziehungsweise -RAM) abgespeichert werden. Wie, das wird zum Schluß erklärt.

Wurden alle Eingaben gemacht, startet das Programm die Initialisierungs-Routine, die Kernal und Basic mit deutschen System- und Fehlermeldungen versieht. Zu guter Letzt wird das neue Betriebssystem aktiviert (Zeile 133). Jetzt können ruhig mal alle Neuheiten durchgespielt werden. Gar nicht so einfach, alle Fehler-Meldungen zu »provozieren«.

Und dabei gibt es immer noch Programmierer, die sich bei Meldungen wie »Syntax Error« die Haare raufen. Auf jeden Fall: einen »Syntax Error« wird es für jene jetzt wohl kaum mehr geben.

Jetzt noch kurz zu den Änderungen gegenüber den EPROM-Versionen. Um ein EPROM zu brennen, muß das Programm in einem Bereich liegen, auf den das EPROM-Programmiergerät zugreifen kann. Daher liegen die Hypra-Kernal ab $6 000 im Speicher. Die Adressen bleiben unkorrigiert, so daß als EPROM ein lauffähiges Betriebssystem im Originalbereich ab $E000 entsteht. Ausgetestet werden kann aber nur, wenn das Betriebssystem wieder an seinen Ursprung verschoben wird. Das kann am komfortabelsten mit einem Monitorprogramm durchgeführt werden.

Ohne Monitor muß mit einer Basic-PEEK/POKE-Schleife der Speicherbereich $6000 bis $7FFF auf die Adressen ab $E000 verschoben werden:

```
forx=24576to49151:fory=57344to65535
pokey,peek(x):nexty:nextx
```

Zuvor muß sichergestellt sein, daß das RAM unter dem ROM ab $E000 aktiviert wurde (wie durch die Routine ab Zeile 193). Am besten, man speichert sich diese Basic-Routine separat als Programm, da sie beim Einlesen eines Betriebssystems immer wieder benötigt wird. Aus Zeitgründen wäre eine solche Routine in Maschinen-Code auch nicht zu verachten.

Der verschobene Speicherbereich muß nun lediglich noch auf Disk abgespeichert werden. Ohne Monitor kann dazu zum

100-101	Feststellen ob ROM/RAM aktiv Feststellen ob Kernal nachgeladen
102	RAM aktivieren
103	Routine zum Data-Einlesen
123-127	Anfangs- und Endadressen der POKE-Schleifen
128-132	Variablenübergabe an DATA-Einleseroutine und Prüfsummen der Datenblöcke
133	Aktivieren des neuen Kernal/Basic
134-154	Datenblock 1 Fehlermeldungen
155-159	Datenblock 2 Interpreter-Meldungen
160-166	Datenblock 3 Startkennung
167-175	Datenblock 4 System-Meldungen Kernal
176-182	Datenblock 5 Old-Routine
183-189	Prüfsummen-Statement
190-197	ROM-RAM Kopierroutine
198-209	Austesten der Farbgebung
210-230	Old-Routine anpassen
231-239	Nachladen eines Betriebssystems

▲ **Tabelle 1. Programmbeschreibung**

N$	= Filename des nachgeladenen Kernals
P	= Zeropage-Adresse 1 auf ROM oder RAM
A	= Anfangsadresse der DATA-Schleifen
B	= Endadresse der DATA-Schleifen
S	= Prüfsumme
Z	= Zeichenfarbe
R	= Rahmenfarbe
H	= Hintergrundfarbe
K	= verwendete Kernalversion

◀ **Tabelle 2. Die wichtigsten Variablen**

Beispiel das kleine Programm im 64'er, Ausgabe 2/85, Seite 91 verwendet werden.

Mit einem Maschinensprache-Monitor geht's einfacher:
Verschieben: .T6000,7FFF,E000-
Save Kernal: .S »K-Name«,08,E000,FFFF
Save Basic: .S »B-Name«,08,A000,BFFF

Beim SMON hat der Verschiebe-Befehl den Buchstaben »W«, und Komma, Minuszeichen und Geräteadresse entfallen.

Übrigens kann auch das ROM ins RAM mittels Monitor kopiert werden. Die Befehle dazu lauten
Kernal: T E000, FFFF, E000-Basic: T A000,BFFF,A000- (danach POKE 1,53).

Auf der Diskette hat man jetzt ein Betriebssystem, das nur mehr vom Programm nachgeladen zu werden braucht. Es kann aber auch schon vorher ein Betriebssystem im RAM aktiv sein. Das Bearbeitungsprogramm erkennt automatisch, ob das Original oder ein neues Kernal aktiv ist (Adresse 1 gleich 55 oder 53, Zeile 100/101)! Andernfalls wird immer das Original-Betriebssystem bearbeitet.

Grundsätzlich muß vor jedem Laden einer Kernal- oder Basic-Version von ROM auf RAM umgePOKEt werden.

Arbeiten mit dem neuen Betriebssystem:
1. ROM auf RAM umPOKEn
2. Zeropage-Adresse 1 mit POKE 1,53 umschalten
3. Kernal laden mit »LOAD "Name",8,1«
4. Basic laden mit »LOAD "Name",8,1«

Zur Kontrolle, daß tatsächlich das neue System läuft, kann mit SYS 58648, SYS 58260 in den Warmstart gesprungen werden. Ein SYS 64738 schaltet immer auf das Original-ROM zurück und sollte vermieden werden. Falls er doch einmal ausgelöst wird (zum Beispiel Run-Stop/Restore), kann mit POKE 1,53 jederzeit wieder ins RAM geschaltet werden.

Da das Bearbeitungsprogramm auch das Basic-ROM ändert, muß auch immer Basic von Disk geladen werden. Andernfalls hat man englische Fehlermeldungen und OLD nur als SYS-Befehl.

Zum Abschluß muß noch erwähnt werden, daß der OLD-Befehl den Basic-Befehl END eliminiert. Bei Fremdprogrammen sollte im Hinblick auf Syntax-Fehler daran gedacht werden. Eine Verlängerung der Basic-Befehlsliste ist aus Platzgründen in dieser Programmversion nicht realisiert worden.

(Richard Diezmann/rg)

```
100 IF N$=""THEN P=PEEK(1):IF P<>53 THEN 1
    91                                      <198>
101 IF N$=""THEN P=PEEK(1):IF P=53 THEN 23
    2                                       <136>
102 POKE 1,53:GOTO 201                      <212>
103 FOR I=A TO B:READ D:POKE I,D:S=S+D:NEX
    T I                                     <220>
104 RETURN                                  <162>
105 REM*****************************        <117>
106 REM*                           *        <155>
107 REM*   BETRIEBSSYSTEM-EDITOR   *        <100>
108 REM*                           *        <157>
109 REM*       COMMODORE 64        *        <244>
110 REM*     KERNAL UND BASIC      *        <176>
111 REM*                           *        <160>
112 REM*         1985 BY RIDI      *        <095>
113 REM*   ---------------------   *        <209>
114 REM*     RICHARD DIEZMANN      *        <058>
115 REM*     LOISACHSTRASSE 5      *        <182>
116 REM*     8400 REGENSBURG       *        <099>
117 REM*                           *        <166>
118 REM*     TEL.(0941)49542       *        <192>
119 REM*                           *        <168>
120 REM*****************************        <132>
121 POKE 53280,1:POKE 53281,1:PRINT CHR$(1
    47)CHR$(152)                            <190>
122 PRINT"DEUTSCHES BETRIEBSSYSTEM WIRD";:
    PRINT CHR$(13)"INITIALISIERT !"         <170>
123 A1=41374:E1=41767                       <221>
124 A2=41828:E2=41865                       <117>
125 A3=58463:E3=58540                       <017>
126 A4=61629:E4=61738                       <109>
```

```
127 A5=64608:E5=64667                       <198>
128 A=A1:B=E1:GOSUB 103:IF S<>29905 THEN 1
    83                                      <167>
129 A=A2:B=E2:S=0:GOSUB 103:IF S<>1919 THE
    N 183                                   <088>
130 A=A3:B=E3:S=0:GOSUB 103:IF S<>4464 THE
    N 183                                   <071>
131 A=A4:B=E4:S=0:GOSUB 103:IF S<>7664 THE
    N 183                                   <122>
132 A=A5:B=E5:S=0:GOSUB 103:IF S<>6755 THE
    N 183                                   <169>
133 POKE 41812,219:SYS 58648:SYS 58260      <154>
134 REM DATENBLOCK 1
          BASIC-FEHLERMELDUNGEN             <240>
135 REM --------------------------------    <181>
136 DATA 90,85,32,86,73,69,76,69,32,70,73,
    76,69,211,79,70,70,69,78,161,32         <038>
137 DATA 32,32,78,73,67,72,84,32,79,70,70,
    69,78,161,32,78,73,67,72,84,32,71       <190>
138 DATA 69,70,85,78,68,69,206,71,69,82,65
    ,69,84,32,65,85,83,32,186,32,32         <202>
139 DATA 32,32,32,32,75,69,73,78,32,69,73,
    78,71,46,70,73,76,197,75,69,73,78        <044>
140 DATA 32,65,85,83,71,65,46,70,73,76,197
    ,70,73,76,69,78,65,77,69,32,70,69        <018>
141 DATA 72,76,84,161,32,32,73,76,76,69,71
    ,65,76,69,32,71,69,82,65,69,84,78        <225>
142 DATA 85,77,77,69,210,78,69,88,84,32,79
    ,72,78,69,32,70,79,210,32,32,32          <143>
143 DATA 83,80,82,65,67,200,82,69,84,85,82
    ,78,32,79,72,78,69,32,71,79,83,85        <242>
144 DATA 194,32,32,32,68,65,84,69,78,90,69
    ,73,76,69,206,70,65,76,83,67,72          <057>
145 DATA 69,32,90,65,72,76,186,32,32,32,90
    ,65,72,76,32,90,85,190,75,69,73          <090>
146 DATA 78,32,82,65,77,32,77,69,72,210,90
    ,69,73,76,69,32,78,73,67,72,84,32        <075>
147 DATA 68,65,32,161,32,78,73,67,72,84,32
    ,73,78,32,68,73,77,161,82,69,68          <099>
148 DATA 73,77,32,65,82,82,65,89,32,173,68
    ,73,86,73,83,73,79,78,32,68,85,82        <190>
149 DATA 67,72,32,176,78,85,82,32,80,82,71
    ,46,77,79,68,85,83,161,90,65,72          <051>
150 DATA 76,32,83,84,65,84,84,32,83,84,82,
    73,78,71,161,32,90,85,32,76,65,78        <134>
151 DATA 71,161,32,70,73,76,69,32,68,65,84
    ,193,83,84,82,73,78,71,32,90,85          <112>
152 DATA 32,75,79,77,80,76,69,88,161,32,75
    ,65,78,78,32,78,73,67,72,84,161          <214>
153 DATA 32,32,32,85,78,68,69,70,46,32,70,
    85,78,75,84,73,79,78,161,80,82,85        <237>
154 DATA 69,70,197,76,65,68,197              <244>
155 REM                                      <217>
156 REM DATENBLOCK 2
          MELDUNGEN DES INTERPRETERS        <175>
157 REM --------------------------------    <023>
158 DATA 13,79,75,13,0,32,70,69,72,76,69,8
    2,0,32,73,78,32,0,13,10,66,69,82         <016>
159 DATA 69,73,84,13,10,0,13,10,80,65,85,8
    3,69,0,160                               <128>
160 REM                                      <222>
161 REM DATENBLOCK 3
          STARTKENNUNG BETRIEBSSYSTEM       <145>
162 REM --------------------------------    <028>
163 DATA 0,32,66,89,84,69,83,32,70,65,83,8
    4,76,79,65,68,73,78,71,0,147,13          <221>
164 DATA 32,32,32,32,32,32,42,32,67,79,77,
    77,79,68,79,82,69,32,54,52,32,75         <243>
165 DATA 69,82,78,65,76,32,86,50,46,48,32,
    42,32,13,13,40,67,41,49,57,56,53         <088>
166 DATA 32,66,89,32,82,73,68,73,46,32,0,1
    29                                       <147>
167 REM                                      <229>
168 REM DATENBLOCK 4
          SYSTEMMELDUNGEN KERNAL            <052>
169 REM --------------------------------    <035>
170 DATA 13,69,47,65,32,70,69,72,76,69,82,
    186,13,83,85,67,72,69,32,78,65,67        <084>
171 DATA 72,13,32,32,32,160,68,82,85,69,67
    ,75,69,32,80,76,65,89,84,65,83,84        <089>
172 DATA 69,160,68,82,85,69,67,75,69,32,82
    ,69,67,79,82,68,43,80,76,65,89,84        <240>
173 DATA 65,83,84,197,32,32,32,13,13,76,65
    ,68,69,160,32,32,13,13,83,65,86          <209>
```

Listing »Betriebssystem-Editor«. Bitte beachten Sie die Eingabehinweise auf Seite 6

```
174 DATA 69,160,32,13,13,80,82,85,69,70,69
    ,160,13,70,65,78,68,160,32,13,79        <067>
175 DATA 75,141                             <229>
176 REM                                     <238>
177 REM DATENBLOCK 5
          KERNAL-OLDROUTINE                 <000>
178 REM ------------------------------      <044>
179 DATA 165,43,164,44,133,34,132,35,160,3
    ,200,177,34,208,251,200,152,24,101      <098>
180 DATA 34,160,0,145,43,165,35,105,0,200,
    145,43,136,162,3,230,34,208,2,230       <102>
181 DATA 35,177,34,208,244,202,208,243,165
    ,34,105,2,133,45,165,35,105,0,133       <079>
182 DATA 46,96                              <182>
183 PRINT"FEHLER IN DER DATEN-SUMME "       <154>
184 IF A=A1 THEN PRINT"IN BLOCK 1!"         <057>
185 IF A=A2 THEN PRINT"IN BLOCK 2!"         <124>
186 IF A=A3 THEN PRINT"IN BLOCK 3!"         <191>
187 IF A=A4 THEN PRINT"IN BLOCK 4!"         <003>
188 IF A=A5 THEN PRINT"IN BLOCK 5!"         <070>
189 PRINT"BITTE UEBERPRUEFEN.":END          <161>
190 REM                                     <252>
191 REM INIT-ROUTINE ROM-RAM                <169>
192 REM ------------------------------      <058>
193 POKE 53280,0:POKE 53281,0:PRINT CHR$(1
    47)CHR$(152)                            <000>
194 PRINT"RAM WIRD AKTIVIERT, BITTE 60 SEC
     WARTEN!"                               <010>
195 FOR X=40960 TO 49152:POKE X,PEEK(X):NE
    XT                                      <017>
196 FOR X=57344 TO 65535:POKE X,PEEK(X):NE
    XT                                      <070>
197 GOTO 234                                <053>
198 REM                                     <004>
199 REM FARBZUSAMMENSTELLUNG                <212>
200 REM ------------------------------      <066>
201 SYS 58648                               <094>
202 PRINT"FARBEN BEIBEHALTEN ?"             <142>
203 GET F$:IF F$=""THEN 203                 <152>
204 IF F$="J"THEN 213                       <102>
205 INPUT"ZEICHENFARBE ... ";Z              <082>
206 INPUT"RAHMENFARBE .... ";R              <035>
207 INPUT"HINTERGRUNDFARBE ";H              <253>
208 POKE 58677,Z:POKE 60633,R:POKE 60634,H  <051>
209 SYS 58648:GOTO 202                      <031>
210 REM                                     <016>
211 REM OLD-BEFEHL EINRICHTEN               <003>
212 REM ------------------------------      <078>
213 PRINT CHR$(147)"SOLL DIE FUNKTIONSTAST
    E F8 MIT DEM "                          <057>
214 PRINT"OLD-BEFEHL BELEGT WERDEN (J/N)"   <016>
215 GET F$:IF F$=""THEN 215                 <229>
216 IF F$="J"THEN 218                       <194>
217 GOTO 224                                <057>
218 INPUT"KERNAL STANDARD/HYPRA 1{17SPACE}
    KERNAL HYPRA-PERFECT{2SPACE}2{2SPACE}"
    ;K                                      <197>
219 ON K GOTO 220,221                       <222>
220 POKE 64346,54:POKE 64347,48:GOTO 224    <170>
221 POKE 64469,83:POKE 64470,121:POKE 6447
    1,54:POKE 64472,52:POKE 64473,54        <144>
222 POKE 64474,48:POKE 64475,56:POKE 64476
    ,13:POKE 64477,136:POKE 64778,255       <062>
223 POKE 64479,255                          <045>
224 PRINT CHR$(147)"SOLL DER OLD-BEFEHL AL
    S BASIC-DIREKT-"                        <253>
225 PRINT"BEFEHL VERWENDET WERDEN (J/N)"    <060>
226 GET F$:IF F$=""THEN 226                 <177>
227 IF F$="J"THEN 229                       <229>
228 GOTO 121                                <220>
229 POKE 40972,95:POKE 40973,252            <085>
230 POKE 41118,79:POKE 41119,76:POKE 41120
    ,196:GOTO 121                           <079>
231 REM                                     <037>
232 REM LADEN EINES VORHANDENEN KERNALS     <140>
233 REM ------------------------------      <099>
234 POKE 1,55:PRINT CHR$(147)"SOLL EIN BES
    TEHENDES BETRIEBSSYSTEM VON"            <008>
235 PRINT"DISK GELADEN WERDEN ?"            <245>
236 GET F$:IF F$=""THEN 236                 <251>
237 IF F$="N"THEN POKE 1,53:GOTO 201        <115>
238 INPUT"FILENAME ";N$                     <180>
239 LOAD N$,8,1                             <203>
```


Listing »Betriebssystem-Editor« (Schluß)

Mehr Platz auf der Diskette durch »File-Compactor«

Die Zeit, in der man aus Platzgründen nur ein oder zwei Programme auf einer Diskette unterbringen kann, ist vorbei. Mit diesem neuartigen File-Compactor sind Sie in der Lage, nicht nur Grafiken, sondern sogar beliebige Programme um bis zu 55 Prozent zu kürzen.

Das Problem, Programme zu packen, um so Platz auf Disketten zu sparen, ist mit einem einfachen File-Compactor, der sich darauf beschränkt, Folgen gleicher Bytes zu erkennen und zu vereinfachen, noch lange nicht beseitigt. Ein solcher Compactor (zum Beispiel der File-Compactor aus dem 64'er, Ausgabe 7/85) kann mit Maschinensprache-Routinen wenig anfangen, da dort selten mehr als zweimal das gleiche Byte hintereinander vorkommt. Ein guter Programm-Compactor muß daher anders arbeiten.

Ein neuartiges Computerverfahren

Hier ist nun ein Compactor, der sich die geringe Unordnung in Programmen zunutze macht. Dadurch wird wirklich jedes Programm (mit einer Länge von mindestens 25 Blocks) erheblich gekürzt. Es gibt praktisch kein Programm, in dem die verschiedenen Zeichen gleich häufig vorkommen. In Basic-Programmen wird das besonders deutlich, weil dort Leerzeichen, Doppelpunkte, das $00-Byte oder auch einzelne Buchstaben (wie zum Beispiel das »e«) besonders oft zu finden sind. Es gibt aber auch Zeichen, die in einem Programm nur ganz selten oder überhaupt nicht vorkommen. Diese Tatsache kann man sich zunutze machen, indem man beispielsweise für häufig vorkommende Zeichen, einen 3 Bit langen Code definiert. Jeder kann sich vorstellen, was das für eine Platzersparnis mit sich bringt. Je mehr in dem Programm einige wenige Zeichen dominieren, desto geringer ist der durchschnittliche Informationswert und desto stärker läßt es sich kürzen. Natürlich müssen dabei für die weniger oft vorkommenden Zeichen längere Codes (mit einer Länge von 8 Bit und länger) definiert werden, damit die Codierung eindeutig bleibt.

Zusätzlich wurde in diesem Compactor das Verfahren aus Ausgabe 7/85 (Zählen gleicher Bytefolgen) angewandt, da es besonders bei Spielen mit viel Grafik sehr effizient arbeitet.

Das Programm, das gekürzt werden soll, muß ab dem Basic-Start (Adresse $0801) in den Speicher geladen werden (also laden mit »LOAD"Name",8«) und darf nicht länger als 48639 Bytes sein. Nach dem Kürzen steht eine SYS-Zeile am Anfang, die den Entpacker startet. Dieser wandelt den komprimierten Code wieder in ein lesbares Programm. Auch Basic-Programme können auf diese Weise von dem Computer bearbeitet werden. Mit »RUN« wird dann zuerst der Entpacker ge-