

Eine der interessantesten Möglichkeiten, BASIC und ein Plattenbetriebssystem (DOS) zusammenarbeiten zu lassen, ist das Erstellen von Dateien, die durch ein oder mehrere BASIC-Programme verwaltet werden. Die folgende Programmier-technik ist ausführlich in den Unterlagen von Ohio beschrieben; Junior-Besitzer können von diesen Unterlagen auch profitieren. Das Buchhaltungsprogramm, das uns ein Leser zuschickte, war für uns Anlaß genug, uns einmal mit diesem interessanten Thema zu beschäftigen.

# Indirekte Dateien mit dem Junior

Zum Beispiel: Ein Buchhaltungsprogramm

L. Germain

Zuerst muß man natürlich wissen, was der Unterschied zwischen einer direkten und einer indirekten Datei ist. Wenn ein Programmierer sich hinsetzt, ein Programm mit Zeilennumerierung schreibt, es ändert, einen Programmausdruck macht usw., dann hat er einen direkten Zugang zu seiner Datei. Wenn das gleiche Programm später Daten holt, Berechnungen macht, die Ergebnisse in einer neuen Datei ablegt und das ganze mit Hilfe der Befehle OPEN, CLOSE, GET und PUT auf Diskette speichert, spricht man von einem indirekten Zugriff. Bei solchen Dateien kann der Anwender nichts mehr direkt ändern, hinzufügen usw.

## Wie macht man indirekte Dateien?

Ganz allgemein muß jede Datei erst mal einen Namen erhalten. In unserem Beispiel heißen die Dateien "DATA1" und "DATA2" (indirekte Dateien), "BANK" (Hauptprogramm) und "PRPDA3" (PRE-Pare DATA3).

"DATA3" ist eine indirekte Datei mit wahlfreiem Zugriff ("random access"), die man nur braucht, um den Saldo eines Kontos, die Kennziffer der letzten Eintragung (diese beiden Daten werden automatisch durch "PRPDA3" initialisiert) zu speichern. Außerdem enthält diese Datei eine Art Geheimkode, so daß nur der Benutzer selbst auf die Datei zugreifen kann; dazu muß lediglich der Befehl RUN durch den Befehl NEW in Zeile 45 von "BANK" ersetzt werden. Wenn sich Unbefugte am Programm zu schaffen machen, löscht es sich selbst. Am Anfang von Zeile 20 kann auch noch der Befehl POKE 741,10 eingefügt werden, damit das Programm nicht ständig aufgelistet wird.

"DATA2" enthält die eigentlichen Buchhaltungsdaten (Summen, Datum, Zusatzinformationen), die vom Programm "BANK" erfaßt und verarbeitet wurden.

Bevor die Programme "BANK" und "PRPDA3" gespeichert werden, müssen mit Hilfe von Option 7 von BEXEC\* Pufferzonen hergestellt werden. Für das Programm "BANK" braucht man zwei Pufferzonen ("DATA2" und "DATA3"); für "PRPDA3"

braucht man nur eine. Dann müssen auf der Diskette vier Spuren für "BANK", eine für "DATA3", eine für "PRPDA3" und gut ein Dutzend für "DATA2" reserviert werden. Ist erst einmal das Programm aus Tabelle 1 auf Diskette, läßt man "PRPDA3" laufen, gibt einen Kode ein und läßt schließlich das Hauptprogramm "BANK" laufen. Sobald der richtige Kode eingegeben ist, erhält man einen Überblick über die 8 Möglichkeiten, die im Folgenden erläutert werden.

## Anwender-Routinen

Zu Beginn jeder Routine sucht das Programm auf der Diskette ("DATA3") zwei Grunddaten: den vorhandenen Saldo (Variable S) und die Bezeichnung der letzten Eintragung (Variable C). Das passiert ab Programmzeile 500. Ab Zeile 525 läuft ein Unterprogramm, das dafür sorgt, daß die zuvor aus dem Speicher geholten Daten wieder zurücktransportiert werden.

### Ein- und Ausgabe-Routine

Zuerst fragt das Programm ab, wieviele Kredit- oder Debit-Befehle es ausführen soll. Danach werden Betrag, Art (oder Kategorie) und Datum der ersten Eingabe abgefragt. Die Datumsangabe ist immer 6-stellig und hat die Form TT (Tag)/MM (Monat)/JJ (Jahr). die Art der Eingabe wird in einer Zeichenfolge ("string") registriert. Es kann sich dabei also um Namen (wie "Steuer"), Zahlen (Schecknummer beispielsweise) oder Abkürzungen handeln.

### Suchen

Es ist möglich, alle Kredit- und Debit-Vorgänge vollständig aufzulisten, zum Beispiel für einen Monat, oder eine Liste für einen bestimmten Vorgang machen zu lassen. Wenn einer der notwendigen Parameter (Betrag, Art, Datum) unbekannt ist, einfach ein "X" eingeben.

### Saldo

Mit dieser Routine wird der vorhandene Saldo festgestellt.

### Abschluß

Wie der Name schon sagt, können alle Eintragungen mit diesen Befehlen zusammengefaßt werden.

Beim ersten Ausprobieren des Programms, muß man mit der Ein- und Ausgabe-Rou-

Tabelle 1. Das Programm "PRPDA3" dient nur dazu, einen (geheimen) Kode in die Datei "DATA3" einzutragen.

Tabelle 1.

```
5 REM PRPDA3
10 PRINT:PRINT:INPUT"INPUT CODE ";B$:C=0:S=0
20 DISK OPEN,6,"DATA3":DISK GET,4
30 PRINT$6,4;" ";B$:DISK PUT:DISK GET,2
40 PRINT$6,2;" ";C;" ";S:DISK PUT: DISK CLOSE,6
```

Tabelle 2.

```

1 REM BANK
5 PRINT:PRINT:POKE 2888,0:POKE8722,0
10 PRINTTAB(16)"**FINANCIAL CONTROL PROGRAM**":PRINT:PRINT
20 PRINT:PRINT:INPUT"CODE ";A$:PRINT:PRINT
30 DISK OPEN,6,"DATA3":DISK GET,4
45 INPUT $6,R,B$:DISK CLOSE,6:IFA$<>B$ THEN RUN
70 CLEAR:PRINT:PRINTTAB(3)"1 = CREDITS"
80 PRINTTAB(3)"2 = DEBITS"
90 PRINTTAB(3)"3 = REQUESTS"
95 PRINTTAB(3)"4 = BALANCE"
96 PRINTTAB(3)"5 = MONTHLY TOTAL"
97 PRINTTAB(3)"6 = ANNUAL TOTAL"
98 PRINTTAB(3)"7 = TOTAL PER CATEGORY"
100 PRINTTAB(3)"8 = EXIT"
105 PRINT:INPUT"NUMBER SELECTED ";A
140 ON A GOTO 430,200,250,330,340,380,430,4000
120 GOTO70
121 REM
130 GOSUB500:PRINT:PRINT:PRINTTAB(16)"CREDIT ROUTINE"
145 PRINT:PRINT:INPUT"NUMBER OF CREDITS";D:PRINT:PRINT
150 DISK OPEN,6,"DATA2"
155 FOR R=C+1 TO C+D
160 INPUT"AMOUNT ";E:S=S+E:PRINT:PRINT
161 INPUT"OPERATION OR CATEGORY ";A$:PRINT:PRINT
170 INPUT"DATE (DAY/MONTH/YEAR) ";B$:S$="C":PRINT:PRINT:DISK GET,R
180 PRINT$6,R,"";E,"";A$,"";B$,"";S$
185 DISK PUT:NEXTR:C=R-1:GOSUB525:GOTO70
186 REM
200 GOSUB500:S$="D":PRINT:PRINT:PRINTTAB(16)"DEBIT ROUTINE":PRINT:PRINT
205 DISK OPEN,6,"DATA2":INPUT"NUMBER OF DEBITS ";B:PRINT:PRINT
215 FOR R=C+1 TO C+B
220 INPUT"AMOUNT ";E:PRINT:PRINT
221 IF(S-E)<0THEN PRINT"YOU ONLY HAVE ";S;" POUNDS!":PRINT:PRINT:GOTO220
225 S=S-E
226 INPUT"OPERATION OR CATEGORY ";A$:PRINT:PRINT
227 INPUT"DATE (DAY/MONTH/YEAR) ";B$:PRINT:PRINT
235 DISK GET,R:PRINT$6,R,"";E,"";A$,"";B$,"";S$
240 DISK PUT:NEXTR:DISK CLOSE,6:GOSUB525:GOTO70
241 REM
250 PRINT:PRINTTAB(16)"REQUEST ROUTINE":PRINT:PRINT
255 PRINT"DO YOU WANT TO LOOK AT ";
256 INPUT"CREDITS OR DEBITS (C/D) ";Z$:PRINT:PRINT
260 INPUT"DO YOU WANT A LIST (Y/N) ";X$:IF X$<>"Y" THEN 265:PRINT:PRINT
262 INPUT"DO YOU WANT A LIST PER MONTH (Y/N)";H$:PRINT:PRINT:IF H$<>"Y"THEN280
263 GOSUB 500:GOSUB 740:V$=Z$:GOTO345
265 PRINT:PRINT:INPUT"AMOUNT ";I$:PRINT:PRINT:IF I$<>"X" THEN 280
270 INPUT"OPERATION OR CATEGORY ";I$:PRINT:PRINT
271 IF I$<>"X" THEN 280
275 INPUT"DATE ";I$:IF I$="X" THENPRINT:PRINT"IMPOSSIBLE REQUEST !!":GOTO70
276 REM
280 GOSUB 500:DISK OPEN,6,"DATA2"
285 FOR R=1 TO C:DISK GET,R
290 INPUT$6,R,E,A$,B$,S$:IF Z$<>S$ THEN 320
295 IF X$="Y" THEN GOSUB600:GOTO 320
296 REM
300 IF I$=A$ OR I$=B$ OR E=VAL(I$) THEN GOSUB 600
320 IF W=15 THEN INPUT"CR TO CONTINUE ";W$:W=0
321 NEXTR:IF Y=0 THEN PRINT:PRINT"DATA NOT FOUND "
322 DISK CLOSE,6:Y=0
323 PRINT:PRINT:INPUT"CR TO CONTINUE ";W$:GOTO70
324 REM
330 GOSUB500:PRINT:PRINT:PRINT"BALANCE ";S;" POUNDS":GOTO 323
331 REM
340 X$=" MONTH ":GOSUB700
345 PRINT:PRINT:INPUT"MONTH ";M:DISK OPEN,6,"DATA2"
355 FOR R=1 TO C:DISK GET,R
357 INPUT $6,R,E,A$,B$,S$
360 IF V$<>S$ THEN 365
361 L=VAL(MID$(B$,4,2)):O=VAL(RIGHT$(B$,2))
363 IF H$="Y" AND Q=0 AND M=L THEN GOSUB600:IF W=18 THEN 320
364 IF Q=0 AND M=L THEN LET T=T+E
365 NEXTR
370 PRINT:PRINT:PRINT"TOTAL OF ";V$;" FOR ";M;" / ";Q;" : ";T;" POUNDS "
371 T=O:Q=0
375 GOTO 323
376 REM
380 X$="YEAR ":GOSUB700:DISK OPEN,6,"DATA2"
400 FOR R=1 TO C:DISK GET,R
410 INPUT $6,R,E,A$,B$,S$
415 IF V$<>S$ THEN 425
420 O=VAL(RIGHT$(B$,2))
422 IF Q=0 THEN LET T=T+E
425 NEXTR
427 PRINT:PRINT:PRINT"TOTAL OF ";V$;" FOR ";Q;" : ";T;" POUNDS":T=O:GOTO 323
428 REM
430 X$=" CATEGORY ":GOSUB700:PRINT:PRINT:INPUT"CATEGORY ";I$
435 DISK OPEN,6,"DATA2"
440 FOR R=1 TO C:DISK GET,R:INPUT $6,R,E,A$,B$,S$
442 IF V$<>S$ THEN 450
444 L=VAL(RIGHT$(B$,2)):IF Q=L AND A$=I$ THEN LET T=T+E
450 NEXTR
455 PRINT:PRINT:PRINT"TOTAL OF ";I$;" FOR ";Q;" = ";T;" POUNDS":T=O:GOTO323
456 REM
500 DISK OPEN,6,"DATA3"
540 DISK GET,2:INPUT $6,R,C,S:DISK CLOSE,6:RETURN
524 REM
525 C=R-1:DISK OPEN,6,"DATA3":DISK GET,2:PRINT$6,2,"";C,"";S
540 DISK PUT:DISK CLOSE,6:RETURN
550 REM
600 IF LEN(STR$(E))=9 THEN SP=0:GOTO 620
602 K=LEN(STR$(E)):SP=(9-K)
620 PRINT R;TAB(4)"AMOUNT ";SPC(SP);E;"Pds /OPER. OR CAT. ";A$;
621 PRINTTAB(49)"DATE";B$;" ";S$
630 W=W+1: Y=Y+1:RETURN
700 GOSUB 500:PRINT:PRINT"ADDING ROUTINE PER ";X$
705 PRINT:PRINT:INPUT"CREDITS OR DEBITS (C/D) ";V$
710 PRINT:PRINT:INPUT"YEAR ";Q:RETURN
1000 END

```

Indirekte Dateien mit dem  
Junior  
Elektor Dezember 1983

Tabelle 2. Das eigentliche  
Buchhaltungsprogramm  
("BANK") hat zwei  
indirekte Dateien mit  
wahlfreiem Zugriff  
("random access"),  
"DATA2" und "DATA3".  
Diese beiden Dateien  
brauchen je eine Puffer-  
zone, weil sonst die neben-  
stehenden Daten nicht ein-  
getragen werden können.  
Achtung! Das Zeichen  
"\$" muß durch "#" er-  
setzt werden.

tine anfangen, das heißt: Man muß zu-  
mindest eine Kredit-Eingabe machen, weil  
sich das Programm sonst weigert, nach  
Debit-Eintragungen zu suchen. Es wäre hier  
zu mühsam, alle Programmschritte einzeln

durchzukauen, obwohl das Programm schon  
eine stark zusammengefaßte Version ist.  
Trotzdem glauben wir, einen interessanten  
Einstieg in das Thema der indirekten Dateien  
vorgestellt zu haben.