

## Datenkreislauf

# Die sequentielle Datenspeicherung

**Als es noch keine Diskettenlaufwerke gab, standen lediglich Bandgeräte zur Datenspeicherung zur Verfügung. Und auf Bändern ist nur sequentielles Speichern möglich. Auch mit der Floppy kann man Daten sequentiell speichern, obwohl der Vorteil der Diskette, der direkte Zugriff auf Daten, nicht ausgenutzt wird.**

**W**enn Sie sich das Diskettenlaufwerk gerade erst gekauft haben, werden Sie sicher zu Beginn einige Schwierigkeiten haben, es auch richtig einzusetzen. Das erste, was man macht, wird das Laden und Abspeichern von Programmen sein. Daß Sie mit der Floppy noch viel mehr machen können, wissen Sie ja auch schon längst. Und sicher fallen Ihnen einige Möglichkeiten ein, die Sie wohl gerne realisieren möchten. Meistens sind das am Anfang eine Adressenkartei, eine Schallplattensammlung erfassen oder etwa seine Dias ordnen und auswerten. Wir möchten Ihnen den Einstieg etwas erleichtern und Sie Schritt für Schritt mit den Methoden der Datenverwaltung vertraut machen.

Es gibt verschiedene Arten von Dateien und unterschiedliche Methoden ihrer Verwaltung. Hauptsächlich sind das die sequentielle, die relative Datei und der Direktzugriff auf die Diskette.

Gerade für Anfänger ist die sequentielle Datei eine geeignete Methode, Daten zu verwalten. Diese Einfachheit ist allerdings verbunden mit einer gewissen Langsamkeit. Aber wenn sie sich erst einmal mit der sequentiellen Datenspeicherung auskennen, wird auch die relative Datei und die Direktzugriffsmethode kein großes Problem mehr für Sie darstellen. Außerdem gibt es genügend Probleme, für die eine se-

quentielle Datenspeicherung völlig ausreichend ist.

Zuerst sollen Sie einige wichtige Begriffe kennenlernen.

Sequentiell bedeutet hintereinander. Also alle Daten, die wir benutzen, werden hintereinander, sequentiell auf der Diskette gespeichert (bei Kassetten ist eine andere Methode gar nicht möglich).

Immer, wenn man von und über Dateien spricht, werden solche Vokabeln wie FELD, DATENSATZ oder auch RECORDS genannt. Was ist das eigentlich?

## Der Computer ersetzt Karteikarten

Die meisten von Ihnen kennen eine Karteikarte. Auf dieser Karteikarte können mehrere Eintragungen stehen. Angenommen, Sie besitzen eine Adressenkartei, dann stehen auf jeder Karteikarte normalerweise der Vorname, Nachname, Straße, Postleitzahl und Ort und die Telefonnummer, manchmal auch der Geburtstag und andere wichtige Daten. Das sind die einzelnen Felder eines Datensatzes. Die Gesamtheit aller Felder auf einer Karteikarte bezeichnet man als einen Datensatz, also alle Daten, die zusammengehören. Somit besteht ein Datensatz aus mehreren Feldern. Und der Begriff

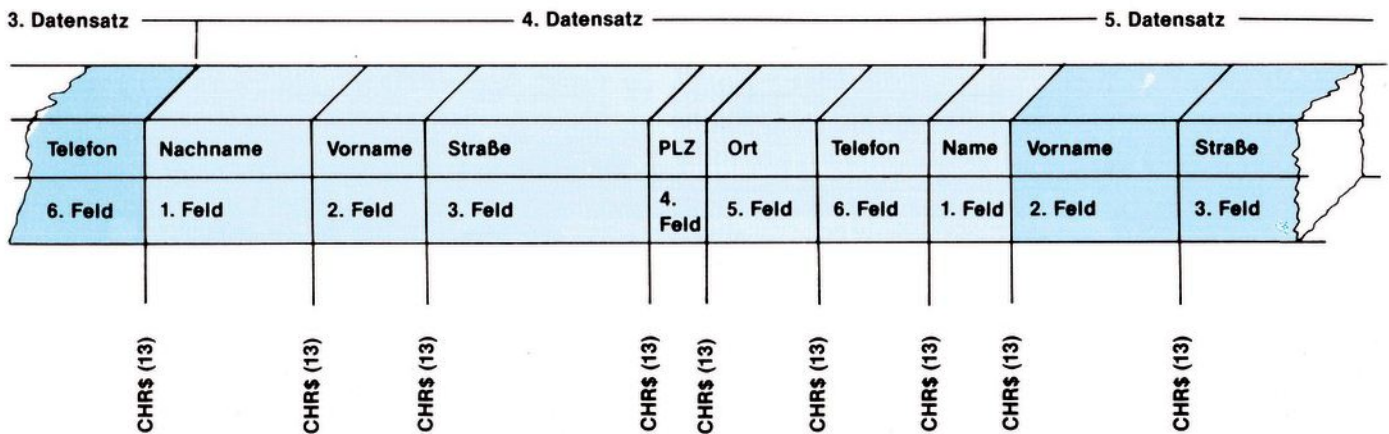
Record ist nur die englische Bezeichnung für Datensatz. Alle Karteikarten, also alle Datensätze, nennt man eine Datei. Damit wären die wichtigsten Begriffe geklärt und wir können zum Wesen der sequentiellen Datei kommen.

Wie schon gesagt, werden bei dieser Dateiart alle Daten hintereinander (sequentiell) auf der Diskette gespeichert. Das sieht dann zum Beispiel so aus wie in Bild 1.

Angenommen, unsere Datei besitzt schon 200 Adressen und eine Adresse enthält den Namen Huber, Anton. Wollen wir jetzt seine Anschrift wissen, so muß die Datei von Anfang an durchsucht werden, bis zu dem Datensatz, in der der Huber steht. Steht der gesuchte Name ziemlich am Ende der Datei, so kann die Suche einige Zeit in Anspruch nehmen; je größer die Datei, desto mehr Zeit beansprucht die Suche. Diese Suche, aber auch alle anderen Operationen wie Ergänzungen, Veränderungen oder Sortieren der Datei spielen sich im Speicher des Computers ab. Das bedeutet, daß, wenn man mit dieser Adressendatei arbeiten will, zuerst die gesamte Datei von der Diskette in den Speicher geladen werden muß. Und am Ende, nach getaner Arbeit, wird sie wieder auf die Diskette zurückgespeichert.

Wenden wir uns dem Beispiel in Bild 2 zu. Es ist ein kleines und noch nicht einmal fertiges Adressen-Datei-Programm (die Funktionen »Suchen«, »Löschen« und »Ändern«) sind noch nicht codiert worden. Jedoch kann sich jeder dieser Module selber schreiben. Auf eine Besonderheit möchte ich gleich hinweisen. In den Zeilen 133 bis 135 stehen die Erklärungen der verwendeten Steuerzeichen, die diesmal codiert ausgedruckt wurden. Deswegen





In dieser Reihenfolge stehen die Daten in einer sequentiellen Datei (bezogen auf das Beispielprogramm)

brauchen Sie nicht mehr mit der Lupe und viel Fantasie, die manchmal schwierig zu erkennenden Commodore-Steuerzeichen versuchen, zu interpretieren. Drücken Sie einfach die entsprechende Cursortaste (innerhalb der PRINT-Anweisung) wenn die Zeichen »CRR« und »CRL« im Listing erscheinen. Auf Ihrem Bildschirm sollten sie dann die entsprechenden Grafikzeichen erkennen. Gefällt Ihnen diese Art der Dar-

stellung? Wir würden gerne Ihre Meinung dazu erfahren.

Der Aufbau des Programms dürfte nicht schwer zu durchblicken sein. Aber es gibt einige Programmzeilen und -abschnitte, die man sich einmal näher anschauen sollte.

Zeile 9110: Die Variable ST ist eine Statusvariable, die der C 64 und der VC 20 für sich beanspruchen. Hier gibt sie Auskunft darüber, ob das Dateiende erreicht ist. In diesem

Fall wird nämlich das 6. Bit gesetzt (2 hoch 6 ergibt 64). Da aber in dieser Systemvariablen auch andere Nachrichten abgelegt werden, die andere Bits beeinflussen und ST auch bei Dateiende einen von 64 ungleichen Wert zuweisen können, müssen wir mit der AND-Verknüpfung feststellen, ob speziell dieses 6. Bit gesetzt ist.

Aus Sicherheitsgründen habe ich vor jedem OPEN-Befehl ein CLOSE

Bild 2. Kleine Adressdatei

```

100 REM *****
110 REM * ADRESSENDATEI 64 ER/6 *
120 REM *****
130 :
131 REM ***** STEUERZEICHEN *****
132 REM *
133 REM * <CLR> = CLR/HOME *
134 REM * <CRR> = CURSOR RECHTS *
135 REM * <CRL> = CURSOR LINKS *
136 REM *
137 REM *****
138 :
140 DI=200
150 DIM NV$(DI),NN$(DI),SR$(DI)
160 DIM PL$(DI),OT$(DI),TE$(DI)
170 POKE 53281,0
180 CLOSE 15:OPEN 15,8,15
1000 REM -----
1010 REM - MENUE -----
1020 REM -----
1030 :
1040 PRINT "<CLR>"
1050 PRINT :PRINT
1060 PRINT " ADRESSENDATEI"
1070 PRINT :PRINT
1080 PRINT " X = PROGRAMMENDE"
1090 PRINT
1100 PRINT " 1 = ANZEIGEN GESAMTE DATEI"
1110 PRINT
1120 PRINT " 2 = SUCHEN"
1130 PRINT
1140 PRINT " 3 = LOESCHEN"
1150 PRINT
1160 PRINT " 4 = AENDERN"
1170 PRINT
1180 PRINT " 5 = NEUE ADRESSEN EINGEBE"
1190 PRINT
1200 PRINT " 6 = SPEICHERN"
1210 PRINT :PRINT :PRINT
1220 PRINT "WAELHLEN SIE ";
1230 GET R$:IF R$="" THEN 1230
1240 IF R$="X" THEN 1500
1250 IF R$="1" THEN GOSUB 2000:REM ANZ.
1260 IF R$="2" THEN GOSUB 3000:REM SUCH
1270 IF R$="3" THEN GOSUB 4000:REM DEL
1280 IF R$="4" THEN GOSUB 5000:REM AEN
1290 IF R$="5" THEN GOSUB 6000:REM NEU
1300 IF R$="6" THEN GOSUB 8000:REM SPEI
1310 GOTO 1000
1320 :
1500 REM -----
1510 REM - PROGRAMM ENDE -----
1520 REM -----
1530 REM SS=2 WENN NICHT GESPEICHERT
1540 IF SS=1 THEN GOSUB 9000
1550 CLOSE 15
1560 END
1570 :
1580 :
2000 REM -----
2010 REM - ANZEIGEN GESAMTE DATEI -----
2020 REM -----
2030 :
2040 IF L=0 THEN GOSUB 9000:REM LADEN
2050 FOR I=1 TO ND
2060 :PRINT "<CLR> ANZEIGE DER DATEI"
2065 :PRINT
2070 :PRINT " NUMMER ";I:PRINT
2075 :PRINT
2080 :PRINT " NAME : "NV$(I) " NN$(I)
2100 :PRINT " STRASSE : "SR$(I)
2110 :PRINT " PLZ/ORT : "PL$(I) " OT$(I)
2130 :PRINT " TELEFON : "TE$(I)
2150 :PRINT :PRINT
2160 :PRINT " W = WEITER X = ENDE"
2170 :GET R$:IF R$="" THEN 2170
2180 :IF R$="X" THEN 2200
2190 NEXT I
2200 RETURN
2210 :
6000 REM -----
6010 REM - EINGABE NEUE DATEN -----
6020 REM -----
6030 :
6040 IF L=0 THEN GOSUB 9000:REM LADEN
6050 PRINT "<CLR>"
6060 PRINT " EINGABE NEUE DATEN"
6070 PRINT :PRINT
6075 ND=ND+1
6076 PRINT " NUMMER ";ND
6078 PRINT :PRINT
6080 INPUT "NACHNAME <CRR><CRR><CRR>.<CRL><CRL><CRL>";NN$(ND)
6090 INPUT "VORNAME <CRR><CRR><CRR>.<CRL><CRL><CRL>";NV$(ND)
6100 INPUT "STRASSE <CRR><CRR><CRR>.<CRL><CRL><CRL>";SR$(ND)
6110 INPUT "POSTLZ. <CRR><CRR><CRR>.<CRL><CRL><CRL>";PL$(ND)

```



gesetzt, damit nicht durch einen Programmierfehler ein »FILE OPEN ERROR« entstehen kann. Zum CLOSE-Befehl sei noch so viel gesagt, daß er unbedingt existieren muß. Sonst kann es vorkommen, daß nicht alle Daten auf die Diskette übertragen werden, und noch schlimmer, daß die Datei nicht ordnungsgemäß geschlossen werden kann. Dadurch kann die gesamte Datei unrettbar verloren sein. Also Vorsicht.

Wenn Sie sich das Programm etwas näher anschauen, könnten Sie über die Variablen »SS« und »L« stolpern. Diese Variablen habe ich eingesetzt als »Flags«. Sie regeln, ob ein Aufruf dieses Menüpunktes sinnvoll ist und das entsprechende Unterprogramm ausgeführt werden soll. Zum Beispiel verhindert die Variable »L« in Zeile 8025, daß Daten gespeichert werden, wenn die Datei noch gar nicht in den Speicher des Computers geladen wurde. Das würde nämlich bewirken, daß (in

diesem Fall die Datei »ADRESS« überschrieben würde ohne es zu wollen; auf Diskette vorhandene Daten wären verloren. Also eine reine Sicherheitsmaßnahme.

## Sicherheit ist Trumpf

Die Variable »SS« verhindert, daß das Programm beendet wird, ohne neu eingegebene Daten vorher abzuspeichern.

Zuletzt möchte ich noch auf einen Programmteil hinweisen, der ab Zeile 10000 steht. Dieses Unterprogramm wird immer dann angesprungen, wenn eine Datei eröffnet wird. Es können nämlich auch dann Fehler auftreten, wenn das Programm sonst einwandfrei funktioniert. Es kann zum Beispiel sein, daß sie eine schreibgeschützte Diskette benutzen wollen, um Daten abzuspeichern. Das funktioniert natürlich nicht. Damit das Programm dann nicht abstürzt, wird jedesmal

nach einem Öffnen eines Floppy-Kanals diese Routine angesprungen. Sie können dann im Fehlerfall das Problem beseitigen, ohne das Programm neu starten zu müssen und damit eventuell alle eingegebenen Daten zu verlieren. Der Fehler »62« in Zeile 10062 weist übrigens darauf hin, daß ein »FILE NOT FOUND ERROR« gemeldet wurde. In diesem Fall kann eine neue Datei angelegt werden (ab 11000). Man kann natürlich auch andere Fehler direkt abfragen. Schauen Sie nach im Floppy-Handbuch von Commodore. Dort sind alle Fehlermeldungen erklärt.

Ich glaube, daß Sie viele Möglichkeiten finden werden, eine sequentielle Datei sinnvoll einzusetzen. Setzen Sie sich hin und fangen Sie an zu programmieren und zu experimentieren. Erweitern Sie Ihr Wissen durch eigene Erfahrungen. Tippen Sie nicht nur ab, seien Sie kreativ!

(Christian Schlüter/gk)

```

6120 INPUT "ORT          <CRR><CRR><CRR>.<CR
L><CRL><CRL>";OT$(ND)
6130 INPUT "TELEFON     <CRR><CRR><CRR>.<CR
L><CRL><CRL>";TE$(ND)
6140 PRINT :PRINT
6150 PRINT "W = WEITERE ADRESSE   X = EN
DE"
6160 GET R$:IF R$="" THEN 6160
6170 IF R$="W" THEN 6050
6175 SS=1:REM   DATEN NEU EINGEGEBEN
6180 RETURN
6190 :
8000 REM -----
8010 REM - SPEICHERN DATEI AUF DISK -
8020 REM -----
8025 REM L=0 WENN NOCH NICHT GELADEN
8030 IF L=0 THEN 8170
8040 CLOSE 1:OPEN 1,8,2,"@:ADRESSE,S,W"
8050 GOSUB 10000:REM  DISKETTENFEHLER?
8060 IF A1<>0 THEN 8040
8070 PRINT# 1,ND
8080 FOR I=1 TO ND
8083 :PRINT "<CLR>   SPEICHERN DER DATE
N"
8095 :PRINT I" VON "ND
8090 :PRINT# 1,NV$(I)
8100 :PRINT# 1,NN$(I)
8110 :PRINT# 1,SR$(I)
8120 :PRINT# 1,PL$(I)
8130 :PRINT# 1,OT$(I)
8140 :PRINT# 1,TE$(I)
8150 NEXT I
8160 CLOSE 1
8165 SS=2:REM  LETZTEN STAND GESPEICHERT
8170 RETURN
8180 :
9000 REM -----
9010 REM - LADEN DER DATEI VON DISK -
9020 REM -----
9030 I=0
9040 CLOSE 1:OPEN 1,8,2,"ADRESSE,S,R"
9045 GOSUB 10000:REM  DISKETTENFEHLER?
9046 IF A1<>0 THEN 9030
9047 INPUT# 1,ND$:ND=VAL(ND$)
9048 I=I+1
9049 PRINT "<CLR>I" VON "ND
9050 INPUT# 1,NV$(I)
9060 INPUT# 1,NN$(I)
9070 INPUT# 1,SR$(I)
9080 INPUT# 1,PL$(I)
9090 INPUT# 1,OT$(I)
9100 INPUT# 1,TE$(I)
9110 IF (ST AND 64)<>64 THEN 9048
9120 CLOSE 1
9130 L=1:REM   DATEI IST GELADEN
9140 RETURN
9150 :
10000 REM -----
10010 REM - DISKETTENFEHLER -
10020 REM -----
10030 PRINT "<CLR>"
10050 INPUT# 15,A1,A2$,A3,A4
10060 IF A1=0 THEN 10170
10062 IF A1=62 THEN GOSUB 11000:GOTO 101
70
10065 PRINT
10070 PRINT A1,A2$,A3,A4
10080 PRINT :PRINT
10090 PRINT "   DISKETTENFEHLER"
10100 PRINT :PRINT
10110 PRINT "   BEHEBEN SIE DEN FEHL
"
10120 PRINT "   UND DRUECKEN SIE"
10130 PRINT
10140 PRINT "   >> F <<"
10150 GET R$:IF R$="" THEN 10150
10160 PRINT "<CLR>"
10170 RETURN
10180 :
11000 REM -----
11010 REM - NEUE DATEI ANLEGEN
11020 REM -----
11030 :
11040 PRINT "<CLR>"
11050 PRINT "   DIE DATEI EXISTIERT NOCH
NICHT"
11060 PRINT :PRINT
11070 PRINT "N = NEUE DATEI   X = ENDE"
11080 GET R$:IF R$="" THEN 11080
11090 IF R$="X" THEN RUN
11100 IF R$<>"N" THEN 11080
11140 CLOSE 1:OPEN 1,8,2,"ADRESSE,S,W"
11150 FOR I=1 TO 7
11160 :PRINT# 1,"."
11170 NEXT I
11180 CLOSE 1
11200 RETURN

```

**Bild 2. Unsere kleine Adressendatei ist nicht ganz vollständig. Aber das ist zum Verständnis auch nicht notwendig. Beachten Sie bitte die Zeilen 131 bis 137 und 6080 bis 6130. Diese Symbole stehen anstelle der sonst so schwer zu erkennenden Steuerzeichen.**