

DATEN IM (RELATIVEN)

DIREKTZUGRIFF

Kommen wir gleich zu den Vor- und Nachteilen. Der Hauptvorteil der relativen Datei ist der schnelle Zugriff auf Daten. Ein weiterer Vorteil ist, daß eigentlich nur wenig Speicherplatz des Computers notwendig ist, nämlich nur so viel, wie das Programm selbst benötigt. Deshalb ist eine sinnvolle Anwendung dieser Dateiform auch mit dem VC 20 in der Grundversion möglich. Allerdings ist ein Diskettenlaufwerk unbedingt notwendig. Die sequentielle Datei hingegen benötigt viel Speicherplatz im Computer, und ein schneller Zugriff auf Daten ist nicht einfach zu realisieren, zumindest muß die gesamte Datei erst in den Speicher des Computers geladen werden, bevor man sinnvoll mit ihr arbeiten kann. Jedoch ist nicht unbedingt ein Diskettenlaufwerk notwendig, die Datensätze tut's auch, wenn auch erheblich langsamer. Der Nachteil der relativen Datei ist die Art des Zugriffs auf bestimmte Daten. Er ist nämlich nur über die Datensatznummer möglich. Das heißt: Angenommen, wir haben uns eine Adressendatei aufgebaut und auch schon eine Anzahl Adressen eingegeben. Wenn wir uns jetzt die Adresse von Anton Huber ausgeben lassen wollen, so ist das nicht möglich, indem wir den Namen »Huber« eingeben und dann das Ergebnis, seine Adresse, auf einen Schlag vor uns haben. Dazu müßten wir seine Satznummer kennen. Vereinfacht kann man sagen, daß die Satznummer angibt, die wievielte Adresse gemeint ist. Satznummer 14 bedeutet also die 14. Adresse. Diese Adresse findet der Computer, weil er vom Dateianfang ausgeht, den er sich merkt, und dann 13 Datensätze überspringt (auf den nächsten Datensatz positioniert), um direkt den 14. Datensatz zu lesen (daher der Name relative Datei: relativ zum Dateianfang).

In diesem Bericht werden die Unterschiede zwischen der relativen und der sequentiellen Datei aufgezeigt und anhand eines Beispielprogramms die Programmierung einer relativen Datei erklärt.

```

100 REM *****
110 REM * ADRESSENDATEI 64'ER/7 REL *
120 REM *****
130 CLOSE 15:OPEN 15,8,15
140 CLOSE 1:OPEN 1,8,2,"ADR.REL,L,"+CHR$(82)
1000 REM -----
1010 REM - AUSWAHL -
1020 REM -----
1030 :
1040 PRINT "M RELATIVE DATEI 64'ER 7"
1050 PRINT :PRINT :PRINT
1060 PRINT " 1 = DATEI EINRICHTEN"
1070 PRINT " 2 = DATEN EINGEBEN"
1080 PRINT " 3 = SUCHEN UND ANZEIGEN"
1090 PRINT
1100 PRINT " X = PROGRAMMENDE"
1110 PRINT :PRINT
1120 PRINT "WAENLE"
1130 GET R$:IF R#="" THEN 1130
1140 IF R#="1" THEN GOSUB 11000
1150 IF R#="2" THEN GOSUB 4000
1160 IF R#="3" THEN GOSUB 3000
1170 IF R#="X" THEN CLOSE 1:CLOSE 15:END

1180 GOTO 1000
2000 REM -----
2010 REM - BILDSCHIRMANZEIGE
2020 REM -----
2030 :
2040 PRINT "M ANZEIGE DATENSATZ"
2050 PRINT
2060 PRINT " NUMMER ";RN:PRINT
2070 PRINT
2080 PRINT " NAME : "NV$ "NN$"
2090 PRINT " STRASSE : "SR$
2100 PRINT " PLZ/ORT : "PL$ "OT$"
2110 PRINT " TELEFON : "TE$
2120 PRINT :PRINT :PRINT
2130 RETURN
2140 :

```

Mit diesem Programm können Sie bis zu 1978 Datensätze (Adressen) verwalten. Das gilt für C 64 und für den VC 20!

```

3000 REM -----
3010 REM - SUCHEN
3020 REM -----
3030 :
3040 GOSUB 13000:REM DATENSATZNR.EINGAB
E
3050 GOSUB 9000:REM LESEN
3060 IF RC#<>CHR$(255) THEN 3120
3070 PRINT "E"
3080 PRINT :PRINT :PRINT
3090 PRINT " NOCHT NICHT BESCHRIEBEN"
3100 PRINT :PRINT
3110 GOTO 3140
3120 GOSUB 5000:REM AUFTEILEN
3130 GOSUB 2000:REM ANZEIGEN
3140 PRINT " DRUECKE TASTE"
3150 GET R#: IF R#="" THEN 3150
3160 R#=""
3170 RETURN
3180 :
4000 REM -----
4010 REM - EINGABE -
4020 REM -----
4030 :
4040 GOSUB 13000:REM DATENSATZNR.
4050 GOSUB 6000:REM EINGABE
4060 GOSUB 7000:REM VERKETTEN
4070 GOSUB 8000:REM SPEICHERN
4080 RETURN
4090 :
5000 REM -----
5010 REM AUFTEILEN DATENSATZ IN FELDER
5020 REM -----
5030 :
5040 NV#=LEFT$(RC$,15)
5050 NN#=MID$(RC$,16,15)
5060 SR#=MID$(RC$,31,20)
5070 PL#=MID$(RC$,51,4)
5080 OT#=MID$(RC$,55,15)
5090 TE#=MID$(RC$,70,12)
5100 RETURN
6000 REM -----
6010 REM - EINGABE NEUE DATEN -
6020 REM -----
6030 :
6040 PRINT "L"
6050 PRINT " EINGABE NEUE DATEN"
6060 PRINT :PRINT
6070 PRINT " NUMMER ";KN
6080 PRINT :PRINT
6090 INPUT "NACHNAME ";NN#:NN#=LEFT$(NN#
,15)
6100 INPUT "VORNAME ";NV#:NV#=LEFT$(NV#
,15)
6110 INPUT "STRASSE ";SR#:SR#=LEFT$(SR#
,20)
6120 INPUT "POSTLZ. ";PL#:PL#=LEFT$(PL#
,4)
6130 INPUT "ORT ";OT#:OT#=LEFT$(OT#
,15)
6140 INPUT "TELEFON ";TE#:TE#=LEFT$(TE#
,12)
6150 PRINT :PRINT
6160 PRINT " ADRESSE OK (J/N) ?"
6170 GET R#: IF R#="" THEN 6170
6180 IF R#="N" THEN 6040
6190 IF R#<>"J" THEN 6170
6200 RETURN

```

Programmierung
einer relativen Datei
(Fortsetzung)

Dazu ist eine Voraussetzung notwendig: Die Datensätze müssen alle die gleiche Länge haben. Wie das realisiert wird, erkläre ich noch.

Um einer möglichen Frustration vorzubeugen, sei gesagt, daß auch eine direkte Suche über einen Namen möglich ist, indem man die Vorteile der relativen Datei mit der der sequentiellen Datei verknüpft.

Eine relative Datei besteht im Prinzip aus 3 Teilen:

1. Einrichten einer Datei
2. Speichern (Schreiben) eines Datensatzes
3. Lesen eines Datensatzes

Gehen wir diese Teile Schritt für Schritt durch. Anhand des Listings können Sie diese Schritte mitverfolgen.

1. Einrichten einer relativen Datei

Um mit einer relativen Datei arbeiten zu können, müssen 2 Kanäle zur Floppy geöffnet werden. Zum ersten ist das der Kommandokanal (Kanal 15) und zweitens ein beliebiger anderer Kanal. Der Kommandokanal ist notwendig um den Positionierbefehl zu übertragen. Mit dem anderen Kanal wird die Datei eröffnet und bearbeitet.

1.a) Öffnen des Kommandokanals

130 CLOSE 15:OPEN 15,8,15

Sicherheitshalber sollte man vor jedem OPEN ein CLOSE setzen, um eine eventuelle Fehlermeldung »FILE OPEN ERROR« zu verhindern.

1.b) Eröffnen der Datei und Festsetzen der Datensatzlänge

Wie schon erwähnt, ist ein Merkmal der relativen Datei, daß jeder Datensatz die gleiche Länge besitzen muß. Diese Angabe muß beim Einrichten der Datei angegeben werden. Das geschieht mit folgendem Befehl:

OPEN lfn,ga,kanal,"dateiname,L," + CHR\$(datensatzlänge)

(im Listing Zeile 140 und 11120)
lfn = logische filenummer
ga = Geräteadresse (normalerweise 8)

kanal (2-14)

dateiname = Name der Datei, wird so im Directory abgelegt, im Beispiel »ADR.REL«.

L = Kennzeichen für eine relative Datei

datensatzlänge = Summe aller Feldlängen (1-254)

Der Buchstabe »L« sagt dem DOS, daß eine relative Datei eröffnet wird. Diesem Buchstaben muß die Angabe der Datensatzlänge folgen. Sie wird mit dem CHR\$ gesandt. Im Beispiel setzt sich ein Datensatz folgendermaßen zusammen:

NAME = 15 Buchstaben
 VORNAME = 15 Buchstaben
 STRASSE = 20 Buchstaben
 POSTLZ = 4 Buchstaben
 ORT = 15 Buchstaben
 TELEFON = 12 Buchstaben

Das ergibt insgesamt 81 Buchstaben pro Datensatz. Da jeder Datensatz (auch RECORD genannt) mit einem RETURN abgeschlossen wird, erhöht sich die Datensatzlänge auf insgesamt 82 Zeichen.

1.c) Positionieren

Der Computer findet einen bestimmten Datensatz über die Datensatznummer, indem er sich den Dateianfang merkt und die entsprechende Anzahl Datensätze überspringt. Genauer gesagt, er positioniert auf diesen Datensatz. Und das wird mit dem Positionierbefehl erledigt. Dieses Kommando wird über den Kommandokanal (15) der Diskette gesandt. Seine Form ist:

PRINT # lfn,"P"+CHR\$(kanal+CHR\$(low)+CHR\$(high)+CHR\$(byte)

(siehe dazu Zeile 11130 - 11190)

Die Parameter »low« und »high« geben die Datensatz (=Record)nummer an. Da ein Byte maximal den Wert 255 annehmen kann, die Recordnummer aber höher sein darf, muß sie aufgeteilt werden in ein Low-Byte und ein High-Byte. Das geschieht einfach mit den Anweisungen

HB=INT(RN/256)

LB=RN-HB*256

RN = Recordnummer

HB = Höherwertiges Byte

LB = Niederwertiges Byte

Der letzte Parameter (Byte) positioniert auf eine bestimmte Stelle innerhalb eines Records. Beispiel:

PRINT # 15,"P"+CHR\$(2)+CHR\$(12)+CHR\$(1)+CHR\$(8)

Hier wird auf das 8. Byte des 268. Records positioniert. Die 268 ist die Recordnummer und wird aufgeteilt in ein Low-Byte(12) und ein High-Byte(1). (HighByte*256 + Low-Byte=Recordnummer). Beim Schreiben eines Records muß jedoch immer auf das erste Byte positioniert werden.

Zum Schluß wird die Datei freigegeben.

1.d) Freigeben der Datei

PRINT # 15,CHR\$(255)

Diese Anweisung bewirkt, daß alle Records, die unter der angegebenen Recordnummer liegen, mit CHR\$(255) beschrieben werden, sofern sie noch nicht anders belegt wurden. Das dauert seine Zeit. Je mehr Datensätze eingerich-

```

6210 :
7000 REM -----
7010 REM VERKETTEN DER FELDER -
7020 REM -----
7030 :
7040 BL$=" "
7050 RC$=NV$+LEFT$(BL$,15-LEN(NV$))
7060 RC$=RC$+NN$+LEFT$(BL$,15-LEN(NN$))
7070 RC$=RC$+SR$+LEFT$(BL$,20-LEN(SR$))
7080 RC$=RC$+PL$+LEFT$(BL$,4-LEN(PL$))
7090 RC$=RC$+OT$+LEFT$(BL$,15-LEN(OT$))
7100 RC$=RC$+TE$+LEFT$(BL$,12-LEN(TE$))
7110 RETURN
7120 :
8000 REM -----
8010 REM SPEICHERN DATEN AUF DISK -
8020 REM -----
8030 :
8040 PRINT# 15,"P"+CHR$(2)+CHR$(LB)+CHR$(
(HB)+CHR$(1)
8050 PRINT# 1,RC$
8060 RETURN
8070 :
9000 REM -----
9010 REM LESEN DATENSATZ VON DISK -
9020 REM -----
9030 F=0
9040 PRINT# 15,"P"+CHR$(2)+CHR$(LB)+CHR$(
(HB)+CHR$(1)
9050 INPUT# 15,ER:REM FEHLERKANAL
9060 IF ER<>50 THEN 9110:REM RECORD NOT
PRESENT-ERROR
9070 PRINT " DATENSATZNUMMER ZU HOCH"
9080 PRINT :PRINT " DRUECKEN SIE EINE TA
STE"
9090 GET R$:IF R$="" THEN 9090
9100 RUN
9110 INPUT# 1,RC$
9120 IF ASC(RC$)<>255 THEN 9140
9130 REM DATENSATZ IST NOCH FREI
9140 RETURN
11000 REM -----
11010 REM NEUE DATEI ANLEGEN
11020 REM -----
11030 PRINT " ACHTUNG, EINE BESTEHENDE
DATEI MIT"
11040 PRINT :PRINT " DEM NAMEN 'ADR.RE
L' WIRD GELOESCHT"
11050 PRINT
11060 PRINT " (W)EITER (Z)URUECK"
11070 GET R$:IF R$="" THEN 11070
11080 IF R$<>"W" THEN RETURN
11090 PRINT
11100 PRINT :PRINT "BITTE WARTEN"
11110 CLOSE 15:OPEN 15,8,15,"S:ADR.REL"
11120 CLOSE 1:OPEN 1,8,2,"ADR.REL,L,"+CH
R$(B2)
11130 PRINT "WIEVIELE DATENSAETZE SOLL D
IE DATEI "
11140 PRINT "VERWALTEN? ";
11150 INPUT RN
11160 HB=INT(RN/256)
11170 LB=RN-HB*256
11180 PRINT :PRINT "BITTE WARTEN"
11190 PRINT# 15,"P"+CHR$(2)+CHR$(LB)+CHR$(
(HB)+CHR$(1)
11200 INPUT# 15,ER:REM FEHLERKANAL
11210 IF ER<>52 THEN 11250:REM DATEI ZU
GROSS

```

```

11220 PRINT "      DATEI  ZU GROSS"
11230 PRINT :PRINT " DRUECKEN SIE EINE T
ASTE"
11240 GET R$:IF R#="" THEN 11270
11250 PRINT# 1,CHR$(255)
11260 CLOSE 1:CLOSE 15
11270 RUN
13000 REM -----
13010 REM -- EINGABE DATENSATZNUMMER
13020 REM -----
13030 :
13040 PRINT "C"
13050 PRINT :PRINT :PRINT
13060 PRINT " EINGABE DER DATENSATZNUMM
ER"
13070 INPUT RN
13080 HB=INT (RN/256)
13090 LB=RN-HB*256
13100 RETURN

```

READY.

Listing zur relativen Datei (Schluß)

tet (freigegeben) werden, desto mehr Zeit beansprucht diese Arbeit. Aber erst diese Prozedur erlaubt ein fehlerfreies Lesen und schnelles Schreiben von Datensätzen. Wollen Sie einen Record beschreiben, der oberhalb des zuletzt freigegebenen Records liegt, so werden automatisch alle Records, die zwischen dem letzten und dem gerade beschriebenen Record liegen, freigegeben, das heißt, mit CHR\$(255) beschrieben. Um diese Prozedur zu vermeiden, sollten Sie nur das erste Mal, beim Einrichten der Datei, die maximal zu erwartende Anzahl Records freigeben. Die Fehlermeldung »RECORD NOT PRESENT«, die dann im Floppy-Fehlerkanal erscheint, kann ignoriert werden, da dieser Record nur beschrieben (freigegeben) wird.

2. Speichern (Schreiben) eines Datensatzes

Zum Speichern eines Datensatzes sind folgende Funktionen durchzuführen:

2.a) Eingabe der Recordnummer und Aufteilung in Low- und High-Byte (siehe oben).

2.b) Eingabe der Daten (zum Beispiel über Input). Hier wird auch sichergestellt, daß die einzelnen Felder nicht länger werden, als vorher geplant wurde.

2.c) maximale Datensatzlänge bilden.

Das wird im Beispiel (siehe Listing) in Zeile 7000 bis 7999 durchgeführt. Es wird ein String mit der maximalen Länge (im Beispiel 81) gebildet. Dazu wird vorher eine Variable (im Listing BL\$) definiert, mit der An-

zahl Leerstellen, die das längste Feld unseres Datensatzes besitzt (im Beispiel das Feld »STRASSE« mit 20 Zeichen, also enthält BL\$ 20 Leerstellen). Jedes Feld unseres Datensatzes wird entsprechend seiner vorher festgelegten Länge und abhängig von der wirklichen Länge bei der Eingabe mit den notwendigen Leerstellen aufgefüllt. Im Beispiel hat die Variable RC\$ dann die Länge von 81 Zeichen und sie enthält den vollständigen Datensatz.

2.d) Speichern

Das Speichern dieses Datensatzes ist dann schnell geschehen: Es wird auf die vorher (in 2.a) angegebene Recordnummer positioniert und danach mit PRINT#1,RC\$ gespeichert.

Mit PRINT# kann man jedoch nur Datensätze abspeichern, die nicht länger als 88 Zeichen sind. Sonst muß man die Daten Zeichen für Zeichen mittels einer GET#-Schleife schreiben und lesen.

3. Lesen eines Datensatzes

Hier erfolgt die Reihenfolge fast umgekehrt wie beim Schreiben. Zuerst muß jedoch auch hier die Nummer des gesuchten Datensatzes eingegeben werden. Nach entsprechender Aufteilung in Low- und High-Byte wird auf den Record positioniert und anschließend mit INPUT#1, RC\$ von der Disk gelesen.

3.a) Eingabe Recordnummer und Aufteilung in Low- und High-Byte.

3.b) Lesen des Datensatzes

3.c) Aufteilen des Datensatzes

Dieser Vorgang muß umgekehrt dem Verketteten (2.c) geschehen. Es wird ja lediglich der komplette Datensatz in die Variable RC\$ gelesen. In Zeile 5000 bis 5999 werden wieder die einzelnen Felder gebildet. Danach fehlt nur noch die

3.d) Anzeige des Datensatzes

Das wars auch schon. Wenn Sie sich einmal alles gut durch den Kopf gehen lassen und die Problematik am Beispiel durchgehen und ausprobieren, können Sie in Zukunft Ihre eigene relative Datei programmieren.

Es ist gar nicht so schwierig, wie es auf den ersten Blick aussehen mag. Sie können auch Teile dieses Programms in Ihre eigenen einbauen. Schließlich ist das der Vorteil eines strukturierten Programmablaufs.

Zum Schluß möchte ich Sie noch auf einige Besonderheiten im Listing hinweisen.

Zeile 11110: Löschen einer eventuell bestehenden relativen Datei.

Zeile 11200-11220: Lesen des Fehlerkanals. Wenn der Fehler 52 gemeldet wird, bedeutet dies, daß die Anzahl Datensätze, die Sie angegeben haben, die Speicherkapazität der Diskette sprengen würde. Der Fehler 50 (RECORD NOT PRESENT, siehe auch Zeile 9050) bedeutet, daß Sie einen Datensatz lesen wollen, der nicht freigegeben wurde. Beim Schreiben braucht diese Fehlermeldung nicht beachtet werden (siehe 1.d).

Zeile 3060: Wenn ein gelesener Datensatz den Wert CHR\$(255) besitzt, wird davon ausgegangen, daß er noch nicht beschrieben wurde. Deshalb wäre es auch sinnlos, ihn anzeigen zu wollen. Dieser Vergleich wird in Zeile 9120-9130 noch einmal in anderer Form durchgeführt, kann dort jedoch weggelassen werden.

Zeile 1170: Wenn das Programm beendet werden soll, muß der zur Positionierung notwendige Kanal 15 wieder geschlossen werden.

Anzumerken ist noch, daß nicht mehr als eine relative Datei gleichzeitig geöffnet werden kann. Es ist lediglich möglich, noch eine zusätzliche sequentielle Datei zur gleichen Zeit zu nutzen. Und diese Möglichkeit erlaubt uns, auch komfortablere Suchkriterien als über die Recordnummer einzusetzen. Doch darüber mehr in der übernächsten Ausgabe des 64'er Magazins. (gk)