Dr. Ralf Wiegandt

# Kopieren mit einem Laufwerk

## CP/M-Dateien von Diskette zu Diskette

**Mit dem CP/M-Kommando „PIP“ lassen sich Dateien in einfacher Weise auf andere Disketten übertragen. Man gibt dazu lediglich den Dateinamen und das Laufwerk der Quell-Datei sowie das Laufwerk für die Ziel-Datei an, und die Kopie wird vom Betriebssystem automatisch durchgeführt. Verfügt man jedoch nur über ein einziges Diskettenlaufwerk, so läßt sich diese einfache Methode zum Anlegen von Sicherheitskopien nicht anwenden. Hier sollen zwei Methoden vorgestellt werden, die es dennoch möglich machen, Kopien einzelner Diskettendateien mit einem Laufwerk anzufertigen.**

Die erste der beiden vorgestellten Möglichkeiten wird direkt vom Betriebssystem unterstützt und beruht auf dem CP/M-Programm „DDT“ („Dynamic Debug Tool“), welches in der Regel zur Untersuchung von Dateien und Speicherbereichen sowie zur Fehlersuche in Maschinenprogrammen verwendet wird. Mit diesem Standard-Programm ist es möglich, eine Diskettendatei vollständig in den Speicher einzulesen und später wieder abzuspeichern. Leider erfordert diese Methode aber einige Aufmerksamkeit und kleinere Berechnungen, die im folgenden beschrieben sind.

### Kopieren per Betriebssystem

Zunächst einmal ist es notwendig, daß sich die zu kopierende Datei (z. B. „FILE.DAT“) zusammen mit dem Programm „DDT.COM“ auf derselben Diskette befindet (dies erreicht man notfalls dadurch, daß man die hier beschriebene Prozedur zunächst einmal für die Datei DDT.COM selbst anwendet!). Nun wird DDT mit dem Kommando

DDT FILE.DAT

aufgerufen, was bewirkt, daß die zu kopierende Datei (in unserem Beispiel „FILE.DAT“) von DDT in den Speicher eingelesen wird. Ist dies geschehen, meldet sich DDT mit den Angaben:
NEXT PC
xxxx yyyy

Dabei sind xxxx und yyyy Hexadezimalzahlen, die die Adresse des ersten, auf die eingelesene Datei folgenden, Bytes, bzw. die Adresse des ersten abzuarbeitenden Programmbefehls (in der Regel: 0100) bezeichnen. Wichtig für die Durchführung der Dateikopie ist die unter „NEXT“ stehende Hexadezimalzahl: von ihr muß die unter PC stehende Zahl subtrahiert und das Resultat durch 100 (hex) dividiert werden. Es ergibt sich die Anzahl von „Pages“ (Blöcke zu 256 Bytes), die von der Datei belegt werden.

Ein Beispiel: Wenn sich DDT nach dem Einlesen einer Datei mit

NEXT PC
3800 0100

meldet, so ist die Zahl der Pages: (3800–0100)/100 = 37. Die Datei belegt also 37 (hex), d. h. 55 (dez) Speicherseiten.

Das Resultat muß also in eine Dezimalzahl verwandelt werden, um später verwendet werden zu können. Ist dies geschehen, wird das Programm DDT mit der Tastenkombination „CTRL-C“ verlassen und die eingelegte „Quell-Diskette“ mit der „Ziel-Diskette“ vertauscht.

Nun muß noch einmal „CTRL-C“ eingegeben werden, um die neu eingelegte Diskette für den folgenden Schreibzu-

```
100 REM ************************************************
110 REM *                                              *
120 REM *                C O P Y 1                     *
130 REM *                                              *
140 REM *      MBASIC-Routine zum Kopieren von         *
150 REM *  CP/M-Dateien mit einem Diskettenlaufwerk    *
160 REM *                                              *
170 REM *  März 1983                 Dr. R. Wiegandt   *
180 REM *                                              *
190 REM ************************************************
200 REM
210 REM ================================================
220 REM              VORBELEGUNGEN :
230 REM ================================================
240 DEFINT A-Z
250 REM
260 REM  Zeichenfolge zum Löschen des Bildschirms :
270 REM
280 CLR$=CHR$(27)+"*"
290 REM
300 KBYTE = 40 :              REM  Maximale Dateigröße (40 KB)
310 BLOCK = 128 :             REM  Blocklänge (128 Bytes)
320 RECS=1024/128 * KBYTE
330 DIM BUF$(RECS) :          REM  Dateipuffer (40 KB)
340 REM ================================================
350 REM              STARTDIALOG :
360 REM ================================================
370 PRINT CLR$ :              REM  Bildschirm löschen
380 PRINT TAB(20)"CP/M Dateikopie für Einzellaufwerk"
390 PRINT:PRINT
400 PRINT "Es können beliebige Dateien bis zu einer Größe von ";
410 PRINT KBYTE; "KB kopiert werden !"
420 PRINT : PRINT
430 PRINT TAB(13);"QUELL-DISKETTE einlegen und <RETURN> drücken !";
440 A$=INPUT$(1) :            REM  Tastendruck abwarten
450 PRINT : PRINT
```

griff vorzubereiten. Erst dann kann mit dem CP/M-Befehl

SAVE nn FILE.DAT

die noch im Speicher befindliche Datei auf die eingelegte Diskette kopiert werden. Dabei ist „nn" die Zahl der vorher berechneten Speicherseiten (in dezimaler Schreibweise). In unserem obigen Beispiel muß der Befehl also

SAVE 55 FILE.DAT

lauten.
Dieses direkt vom Betriebssystem unterstützte Verfahren erfüllt zwar seinen Zweck, ist aber recht umständlich und fehlergefährdet. Daher soll nun als zweite Möglichkeit ein Basic-Programm vorgestellt werden, das – mit wesentlich weniger Aufwand durch den Benutzer – den gleichen Zweck erfüllt.

```
460 PRINT TAB(13);:INPUT"Welche Datei soll kopiert werden ? ",FI$
470 REM ================================================
480 REM           QUELL-DATEI LESEN :
490 REM ================================================
500 REM
510 REM    Datei als "Random-Access-File" öffnen :
520 REM
530 OPEN "R",#1,FI$,BLOCK
540 FIELD #1,BLOCK AS IN$
550 REM
560 PRINT : PRINT
570 PRINT TAB(30); "BLOCK : ";
580 REM
590 REM    Dateiblöcke einlesen und zählen :
600 REM
610 FOR I = 1 TO RECS
620   GET #1,I       :          REM  Block lesen
630   IF EOF(1) THEN 1030 :     REM  Dateiende ?
640   PRINT USING "####";I; : REM  Blocknummer ausgeben
650   PRINT STRING$(4,8); :     REM  Backspaces ausgeben
660   BUF$(I)=IN$ :             REM  Dateipuffer füllen
670 NEXT I
680 REM
690 REM  Fehlermeldung, falls Datei > 40 KB :
700 REM
710 PRINT : PRINT
720 PRINT CHR$(7); "FILE ZU GROSS !"
730 PRINT : PRINT
740 CLOSE : STOP
1000 REM
1010 REM   Datei vollständig eingelesen :
1020 REM
1030 CLOSE :                    REM  Quell-Datei schließen
1040 LAST=I-1 :                 REM  Letzter Block
1050 PRINT : PRINT : PRINT
1060 PRINT TAB(13); "ZIEL-DISKETTE einlegen und <RETURN> drücken !";
1070 A$=INPUT$(1) :             REM  Tasteneingabe abwarten
1080 RESET :                    REM  Diskettenzugriff initialisieren
1090 REM ================================================
1100 REM            ZIEL-DATEI SCHREIBEN :
1110 REM ================================================
1120 REM
1130 REM    Datei als "Random-Access-File" öffnen :
1140 REM
1150 OPEN "R",#2,FI$,BLOCK
1160 FIELD #2,BLOCK AS IN$
1170 REM
1180 PRINT : PRINT
1190 PRINT TAB(30); "BLOCK : ";
1200 REM
1210 REM   Dateiblöcke schreiben und zählen :
1220 REM
1230 FOR I=1 TO LAST
1240   LSET IN$=BUF$(I) :       REM  Dateipuffer auslesen
1250   PUT #2,I :               REM  Block schreiben
1260   PRINT USING "####"; I; :REM  Blocknummer ausgeben
1270   PRINT STRING$(4,8); :    REM  Backspaces ausgeben
1280 NEXT I
1290 CLOSE :                    REM  Ziel-Datei schließen
1300 PRINT : PRINT : PRINT
1310 PRINT TAB(17) ; "DATEI "FI$" WURDE KOPIERT !"
1320 PRINT
1330 END
```

## Eleganter mit Kopierprogramm

Das Programm (*Bild*) dient dazu, beliebige Dateien (also Text-, Daten- und Programmdateien) bis zu einer Größe von 40 KByte auf eine andere Diskette zu kopieren, ohne ein zweites Laufwerk verwenden zu müssen. Es beruht darauf, daß jede Datei als Random-Access-File geöffnet werden kann, d. h. als Datei mit fester Satzlänge, die es erlaubt, im Direktzugriff auf jeden beliebigen Satz zuzugreifen. Als Satzlänge wird hier sinnvollerweise die Größe eines CP/M-Blocks (128 Bytes für den APPLE-II mit Z80-Softcard) gewählt. Die Datei wird satzweise eingelesen und alle Sätze in einem Pufferfeld gespeichert. Ist dies geschehen, muß die Ziel-Diskette eingelegt werden, so daß der angelegte Dateipuffer im gleichem Format wieder abgespeichert werden kann.

Das Programm wird folgendermaßen angewendet: Nach dem Programmstart wird der Benutzer dazu aufgefordert, die Quell-Diskette in das Laufwerk einzulegen und danach die Taste RETURN zu drücken. Ist dies geschehen, wird der Name der zu kopierenden Datei erfragt, diese anschließend geöffnet und blockweise eingelesen (ein Block enthält 128 Bytes). Während dieses Vorgangs können die Blocknummern der eingelesenen Blöcke auf dem Bildschirm abgelesen werden.

Ist die Datei größer als 40 KBytes, so wird das Programm mit einer entsprechenden Fehlermeldung abgebrochen (falls ein Rechner mit einem größeren Hauptspeicher als 64 KByte verwendet wird, kann diese maximale Dateigröße in Programmzeile 300 entsprechend erhöht werden). Im Normalfall wird jedoch der Benutzer nach dem Einlesen aller Dateiblöcke dazu aufgefordert, die Ziel-Diskette in das Laufwerk einzulegen und danach wiederum die RETURN-Taste zu betätigen. Sobald dies geschehen ist, wird die eingelesene Datei auf der Ziel-Diskette abgespeichert, und der Kopiervorgang ist beendet.

Das abgedruckte Programm wurde auf einem Apple-II mit 64 KByte und einer Z80-Softcard entwickelt und getestet. Da es in Standard-Basic (Microsoft) geschrieben wurde, dürfte es aber auch auf jedem anderen CP/M-Rechner (einschließlich CP/M-86) lauffähig sein. Lediglich die in Zeile 28 definierte Zeichenfolge zum Löschen des Bildschirms muß an den jeweiligen Rechner angepaßt werden.