MCS FDOS

(V2.Ø)

Das Plattenbetriebssystem MCS FDOS

1.

Das MCS FDOS (nachfolgend kurz FDOS genannt) besteht aus folgenden Softwaremodulen:

FDOS-Disk Driver
FDOS-Utilities (Resident)
FDOS-Exekutive

Zusammen mit den MCS Editor-Assembler-Debug-Programmen steht dem Anwender damit ein leistungsfähiges Softwarepaket zur eigenen Programmunterstützung zur Verfügung.

1.1 FDOS-Disk Driver

Der FDOS-Disk Driver befindet sich resident im EPROM auf der mitgelieferten Interface-Karte. Dieses Modul enthält die physikalischen Schreib/Lese-Routinen und verwaltet die Dateien auf der Diskette. Diese Routinen sind auch einem Benutzerprogramm zugänglich und ermöglichen z. B. einen Direktzugriff auf einzelne Records der Diskette.

1.2 FDOS-Utilities

Dieses Unterprogrammpaket beinhaltet Dienstprogramme wie: "Datei reservieren", "Datei kopieren", "Diskette formatieren" usw.

Es befindet sich resident im EPROM (Adreßbereich: $EBØØ - $EFFF) auf der Aufsatzkarte M630/1, auf der auch das MONA2-Betriebssystem untergebracht ist.

1.3 FDOS-Exekutive

Die FDOS-Exekutive ist entweder auf EPROM (Bereich: $EØØØ - $E7FF) untergebracht oder wird von der Diskette direkt ins RAM ($EØØØ - $E7FF) geladen. Es dient dazu, die vom Benutzer eingegebenen Befehle zu interpretieren und auszuführen (s. FDOS-Befehlstabelle).

2. Organisation der Diskette

Jede formatierte Diskette (BASF 6106 o. ä.) hat eine Speicherkapazität von 80k Byte (1k = 1024 Byte). Information wird in Blöcken (Sektoren à 128 Byte) auf der Diskette abgelegt.

Die Diskette besteht aus 40 Spuren (Ø-39), die wiederum jeweils 16 Sektoren (1-16) enthalten. Dem Benutzer stehen auf einer formatierten Diskette nur die Spuren 1-39 zur Verfügung. Spur Ø dient zur Aufnahme der Directory, in der dateispezifische Kenndaten (Name, Länge, usw.), sowie ein Verzeichnis der belegten Sektoren abgelegt sind.

2.1 Dateinamen

Dateinamen bestehen aus maximal 6 ASCII-Zeichen mit Ausnahme von:

SPACE (Leerzeichen)
Komma
Doppelpunkt

Außerdem sollten ASCII-Kontroll-Zeichen bei Dateinamen vermieden werden.

Beispiele gültiger Namen:

KLAUS
P1.SRC
J1+J2

2.2 Gerätenummer

Bei der Eingabe des Filenames kann eine Gerätenummer spezifiziert werden. Diese ist vom eigentlichen Dateinamen durch einen Doppelpunkt zu trennen.

Beispiel: BERND:2
allgemein:NAME:D
wobei D = Ø oder D = 1 für Drive #1
D = 2 für Drive #2

Wird im Dateinamen keine Gerätenummer spezifiziert, so wird automatisch Drive #1 selektiert!

2.3 Aufbau der Directory

Die Directory befindet sich auf der Grundspur (Spur Ø) einer initialisierten Diskette. Sektor #1 enthält das VTOC (Volume Table of Contents). Das VTOC gibt an, wieviele Sektoren durch Direktory und Benutzerdateien belegt sind. Die übrigen Sektoren (2-16) können je 10 "Dateikennblöcke" 'FCB's (File Control Blocks) aufnehmen. Mithin kann eine Diskette bis zu 150 Benutzerdateien enthalten. Jeder FCB besteht aus genau 12 Byte:

1. Byte: bis 6. Byte:	Datei Name (6 ASCII-Zeichen)
7. Byte:	File Status Byte
8. Byte:	Spur # des 1. Records
9. Byte:	Sektor # einer Datei
10. Byte:	Länge der Datei in Records (=Sektoren)
11. Byte:	(Low/High)
12. Byte:	Anzahl der Bytes auf dem letzten Sektor

Eine Datei kann zwischen einem (1) und maximal 624 Records (Sekoren) lang sein!

3. FDOS-Betrieb

3.1 Startprozedur

Die FDOS-Exekutive wird vom Betriebssystem MONA aus durch die Befehlsequenz

```
XXX  EØØØ (SPACE)
EØØØ XX   YYY  G
```

gestartet (d. h., Startadresse = $EØØØ). Befindet sich die FDOS-Exekutive nicht resident auf EPROM, so muß das Programm (Name = DOS) zuerst von der Diskette geladen werden. Dies geschieht durch ein EPROM-residentes Urladeprogramm:

1. Systemdiskette, die die FDOS-Exekutive enthält nach Drive #1
2. Adresse $EFAA anwählen
3. G-Taste drücken

Meldet sich das Betriebssystem MONA mit der Adresse $ØØØØ, so ist das Programm fehlerfrei geladen worden und startbereit.

3.2 Befehlseingabe

Die FDOS-Exekutive erwartet eine Kommandozeile, wenn der Promt Character '.' (ASCII $2F) auf der Konsole erscheint. Die Kommandozeile hat das Format:

Befehlswort Parameter 1, Parameter 2, ...

Das Befehlswort besteht aus minimal 3 Buchstaben und muß vom 1. Parameter (wenn dieser spezifiziert ist) durch ein Leerzeichen oder ein Komma getrennt werden. Alle spezifizierten Parameter sind wahlweise durch Leerzeichen oder Kommata zu trennen. Die Befehlszeile muß mit einem "Carriage Return" (Kurzzeichen ; ASCII $ØD) abgeschlossen werden! Bevor das "Carriage Return" eingegeben ist, kann eine Befehlszeile durch die Backspace-Taste eventuell noch modifiziert bzw. korrigiert werden.

Es folgt eine Aufstellung der gültigen FDOS-Befehle (FDOS-Befehle bestehen aus minimal drei Buchstaben des Alphabets).

LOAd : Datei laden

Syntax: LOAD Name (:D) (, Startadresse)

Beispiel: LOAD PRG.1
LOAD PRG.2:2, D2ØØ

Die betreffende Datei"Name" wird vom Gerät D in den Speicher geladen. Die ersten beiden Byte des Files geben die Pufferstartadresse der Datei an. Wird in der Befehlszeile eine Startadresse expliziert spezifiziert, so ignoriert FDOS die von der Diskette geladenen ersten 2 Byte und legt die Datei in dem vom Benutzer angegebenen Bereich ab. (Relokativ laden!)

RUN : Programm laden und ausführen

Syntax: RUN Name (:D) (, Programmstartadresse)

Beispiel: RUN DEMOS:2
RUN EDASM D2ØØ

Dieser Befehl bewirkt das Laden des Programms von der Diskette und automatischen Start ab der vom Benutzer angegebenen Programmstartadresse. Fehlt dieser Parameter, so wird die Programmanfangsadresse als Programmstartadresse genommen. Ein relocatives Laden ist mit dem RUN-Befehl nicht möglich! Der Parameter D im Dateinamen gibt wie üblich die Gerätenummer an:

D = Ø oder 1 = Drive #1
D = 2 = Drive #2

SAVe : Datei auf Diskette übertragen

Syntax: SAVE Name(:D), Startadresse, Endadresse

Beispiel: SAVE EDITOR D3ØØ DFFF

Die Startadresse (2Byte) und der durch "Startadresse, Endadresse" angegebene Speicherbereich wird unter dem Namen "Name" auf Diskette abgelegt.

Beim SAVE-Befehl darf die Diskette nicht schreibgeschützt sein!

Achtung:

Ein unter dem angegebenen Namen auf der Diskette schon vorhandenes File wird durch diesen Befehl deaktiviert (Kennung"D"= DELETED).

REName: Datei umbenennen

Syntax: RENAME Name 1(:D), Name 2

Beispiel: REN OTTO:1 EMIL

Die Datei mit dem Namen "Name 1" wird in "Name 2" umbenannt.

Ein eventuell deaktiviertes File wird mit diesem Befehl wieder aktiviert, ist also wieder zugreifbar. Stehen mehrere deaktivierte Dateien gleichen Namens auf der Diskette, so wird das erste File wieder aktiviert!

COPy : Datei kopieren

Syntax: COPY QUELLE(:D1), ZIEL(:D2)

Beispiel: COPY DATA; DATA9:2

Die Datei QUELLE (auf Gerät D1) wird auf die Datei ZIEL (Gerät D2) kopiert.

Auf der entsprechenden Diskette muß genügend Platz vorhanden sein, um die Zieldatei aufzunehmen.

DELete: Datei löschen

Syntax: DELETE NAME(:D)

Beispiel: DEL PRG1

Die Datei NAME wird mit der Kennung "D" (Deleted) versehen. Gelöschte Files können von FDOS nicht gelesen werden.

Beim "Packen" einer Diskette (s. REPACK-Befehl) werden gelöschte Files von der Diskette entfernt!

Bemerkung:

Der RENAME-Befehl gestattet geschlöschte Files (Kennung = D) zu reaktivieren (Kennung = P = Permanent)!

SHOw : Listen der Directory

Syntax: SHOW (D)

Beispiel: SHOW 2
SHO

Dieser Befehl bewirkt die Ausgabe der Directory, der sich im Drive #D befindlichen Diskette auf die Konsole in folgender Form:

```
# OF AVLBL RCDS : XXX     Anzahl der freien Sektoren

NAME              LENGTH    STATUS
Name 1            XXXX      P oder D
Name 2            XXXX      P oder D
usw.
```

LENGTH: Länge der Datei in Sektoren

STATUS: D = Deleted
P = Permanent

INIt : Formatieren einer Diskette

Syntax: INI (D)

Beispiel: INIT 2
INI

Die im Drive D befindliche Diskette wird im IBM-Format (128 Byte/Sector) formatiert und kontrollgelesen. Danach wird die Directory auf der Grundspur initialisiert.

Um versehentliches Löschen von Disketten zu vermeiden, muß die FDOS Systemmeldung

R U SURE?

mit "Y" (für Yes) quittiert werden, ehe der Befehl ausgeführt wird.

Bemerkung:

Neue Disketten müssen vor der erstmaligen Benutzung mit dem INIT-Befehl formatiert werden.

REPack: Packen einer Diskette

Syntay: REPACK

Beispiel: REP

Mit dem Repack-Befehl wird der Inhalt einer Dieskette komprimiert, indem alle "gelöschten" Dateien entfernt werden und dadurch der von diesen Dateien belegte Platz frei wird. Vor Befehlsausführung muß die zu packende Diskette nach Drive #2 und eine neue formatierte Diskette nach Drive #1 geladen werden. Ausführung des REPACK-Befehls bewirkt eine Kopie der gültigen (nicht gelöschten) Dateien von Drive #2 nach Drive #1! Die in Drive #2 befindliche Diskette kann dann z. B. neu formatiert und weiterbenutzt werden.

Bemerkung:

"Repack" bietet eine einfache Möglichkeit, Kopien von Disketten anzufertigen!

EXIt : Rücksprung ins MONA-Betriebssystem

Syntax: EXIT

Der Befehl EXIT übergibt die Programmkontrolle dem MONA-Betriebssystem.

Mit (1) EØØØ (Space)

(2) G

kann die FDOS-Exekutive von MONA aus wieder gestartet werden.

EDItor: Übergang zum MCS-Editor

Syntax: EDITOR

Beispiel: EDI

Dieser Befehl bewirkt einen Sprung auf den Kaltstartpunkt des Texteditors.

Ist der Editor nicht im Speicher vorhanden, so hängt das System!

Bemerkung:

Drücken der Tasten CRTL C
bewirkt einen Sprung auf den Warmstartpunkt des Editors, falls dieser geladen und initialisiert (Kaltstart!) ist.

SSA: Show Starting Address

Syntax: SSA Name (:D)

Beispiel: SSA QUELLE

Es wird (hexadezimal) die Anfangsadresse der unter dem Namen "Name" angegebenen Datei auf die Konsole ausgegeben. Der Befehl ist eine reine "Orientierungshilfe" für den Benutzer.

DEBug: Übergang zum DEBUG-Programm

Syntax: DEBUG

Beispiel: DEB

Dieser Befehl bewirkt lediglich einen Sprung auf die Adresse $6000, die Einsprungadresse für das DEBUG-Programm (falls dieses geladen ist). Zur Funktionsweise der DEBUG-Software verweisen wir auf die entsprechende Programmbeschreibung.

4. Fehlermeldungen

Codes

N	Bedeutung
1	Datei nicht vorhanden
2	Schreib/Lese-Fehler
3	Positionierfehler (Spur nicht gefunden)
4	Drive-Klappe nicht geschlossen oder Drive leer
5	Diskette voll
6	Directory voll
7	falscher Parameter im Übergabevektor (DISK DRIVER)
8	Disk ist schreibgeschützt
9	zu viele Files eröffnet (max. 4 in BASIC)

5. FDOS-Driver

Der FDOS-Driver bietet dem Benutzer den Zugang zu sämtlichen Disk-Funktionen ohne Zuhilfenahme der FDOS-Exekutive. Das Driver-Programm enthält Routinen zum sequentiellen Schreiben/Lesen eines Bytes. Der Benutzer muß lediglich vorher die entsprechende Datei "eröffnen".

Die Kommunikation erfolgt durch fünf Subroutinen und eines Vektors im RAM, der die benötigten Parameter enthält:

a) Subroutines:

(1) CLEAR (Adresse $B8ØØ)
initialisiert den FDOS-Driver und löscht evtl. offene Files.

(2) ENTRY (Adresse $B8Ø3)
Einsprung für FDOS-Driver.
Der "Übergabevektor" ($B48Ø-$B48A) bestimmt die auszuführende Funktion.

(3) CHKCNT (Adresse $B8Ø6)
prüft beim sequentiellen Lesen, ob das File zu Ende ist oder nicht.
(Setzt Status Bit Z = 1, wenn letztes Byte)

(4) RDCHDK (Adresse $B8Ø9)
liest ein Byte von der Diskette in den Akkumulator.

(5) WRCHDL (Adresse $B8ØC)
schreibt den Akkumulatorinhalt (1Byte) auf Diskette.

(6) "Übergabevektor":

Name	Adresse (Hex)	Bedeutung
TEMPA	B480	legt die auszuführenden Funktionen fest.
SVDVT	B481	Drive # (Ø oder 1)
FILN	B482-B487	Dateiname (6ASCII'Zeichen)
FLAG	B488	Dateikennung, Programm, Daten usw.
L	B489	Hilfsregister 'HL'
H	B48A	

Mögliche Werte für TEMPA und ihre Bedeutung:

TMP - UEBERGABE-WERTE:

F1 EROEFFNE FILE ZUM SCHREIBEN
E1 SCHLIESSE FILE VOM SCHREIBEN UND ERNEUERE DIRECTORY
F0 EROEFFNE FILE ZUM LESEN
E0 SCHLIESSE FILE VOM LESEN
F2 POSITIONIERE RECORD ZUM LESEN BEI GEOEFFNETEM FILE
E2
F3 POSITIONIERE RECORD ZUM SCHREIBEN BEI GEOEFFNETEM FILE
E3 SCHLIESSE FILE OHNE DIRECTORY-EINTRAG
F4 EROEFFNE DIRECOTRY ZUM LESEN
F5 EROEFFNE ALTES FILE ZUM RENAME
F6 EROEFFNE NEUES FILE ZUM RENAME
E5 SCHLIESSE ALTES FILE VOM RENAME
E6 SCHLIESSE ALTES FILE VOM RENAME
F7 EROEFFNE ZIEL FUER COPY
F8 LOESCHE FILE
F9 LAENGENANGABE FÜR CREATE

6. Benutzung der FDOS-Befehle von der Editor-Ebene

In DOS V2.0 zusammen mit dem Editor/Assembler-Up date können sämtliche FDOS-Befehle vom Editor aus aufgerufen werden :

Syntax:

: ! (FDOS-Befehlszeile)

Beispiel:

: ! SAVE QUELLE

Bemerkungen zu 6.

a) Bei SAVE vom Editor aus brauchen Anfangs- und Endadresse des zu übertragenden Files nicht mit angegeben zu werden! Der Editor trägt die Start-/Endadresse des jeweiligen Edit-Files ein.

b) Bei LOAD vom Editor aus wird die Textdatei automatisch relokativ in den zuvor vereinbarten Textpuffer geladen.

c) Jeder auftretende Fehler führt zurück in die FDOS-Ebene (übergeordnetes Programm!).