

100-11



**OS/M—Software
Systemhandbuch**

EAW *electronic*

P8000

Diese Dokumentation wurde von einem Kollektiv des Kombinates

VEB ELEKTRO-APPARATE-WERKE
BERLIN-TREPTOW "FRIEDRICH EBERT"

erarbeitet.

Nachdruck und jegliche Vervielfaeltigungen, auch auszugsweise, sind nur mit Genehmigung des Herausgebers zulaessig. Im Interesse einer staeendigen Weiterentwicklung werden die Nutzer gebeten, dem Herausgeber Hinweise zur Verbesserung mitzuteilen.

Herausgeber:

Kombinat
VEB ELEKTRO-APPARATE-WERKE
BERLIN-TREPTOW "FRIEDRICH EBERT"
Hoffmannstrasse 15-26
BERLIN
1193

Verantwortlicher Bearbeiter: R. Kuehle

WAE/03-0104-01

Ausgabe: 12/86 BG 117/05/87

Aenderungen im Sinne des technischen Fortschritts vorbehalten.

Die vorliegende Dokumentation unterliegt nicht dem Aenderungsdienst.

Spezielle Hinweise zum aktuellen Stand der Softwarepakete befinden sich in README-Dateien auf den entsprechenden Vertriebsdisketten.

Inhaltsverzeichnis	Seite
1. Einleitung	4
2. Systemueberblick	6
2.1. Laden des Systems	6
2.2. Speicherkonzept	7
3. OS/M-Funktionen	10
3.1. Residente Kommandos	12
3.2. Nichtresidente Kommandos	12
3.2.1. Kommando zur Systemgenerierung	13
3.2.2. Starten und Steuern von Befehlsdateien	13
3.2.3. Assemblieren, Laden und Auflisten von Programmen	14
3.2.4. Ausfuehren, Testen und Abspeichern von Programmen	15
3.2.5. Handhabung von Dateien mit dem FIP- Kommando	16
3.2.5.1. Kopieren von Dateien	16
3.2.5.2. Kopieroperationen zu einzelnen Gerate- einheiten	17
3.2.5.3. Spezielle Kopieroperationen	18
3.3. Kursbeschreibung aller OS/M-Kommandos	20
3.4. OS/M-Applikationssoftwarepakete	55
4. Disketten und Dateien	56
4.1. Wahl des aktuellen Diskettenlaufwerks	56
4.2. Systemdiskette	57
4.3. Diskettenformat	57
4.4. Dateikonzept	58
4.5. Dateizeichnungen	59
4.6. Dateikennungen	59
4.7. Dateiattribute	60
5. BIOS	62
5.1. Adressenplan BIOS-Rufe	62
5.2. Treiber fuer serielle Kanale - System- Konsole, Drucker und serieller Koppelkanal	64
5.3. Druckertreiber	68
5.4. Floppy-Disk-Treiber	68
6. BDOS-Funktionen	70
6.1. BDOS-Funktion 0 - System Reset	70
6.2. BDOS-Funktionen fuer Konsole, Drucker und seriellen Koppelkanal	71
6.3. BDOS-Funktionen fuer Diskettenzugriff	72
7. Allgemeine Fehlermeldungen	80
8. Literaturempfehlungen	82

1. Einleitung

Das OS/M ist ein auf dem 8-Bit-Mikrorechnerteil des Programmier- und Entwicklungssystem P8000 implementiertes Betriebssystem.

Es ist ein "disk-orientiertes" Betriebssystem und ermöglicht dem Bediener mit wenigen einfachen Kommandos vor allem das Programm- und Daten-Speichermedium "Diskette" zu handhaben. Dateiverarbeitung ist sowohl mit sequentiellen als auch wahlfreiem Zugriff möglich.

Die Anwendung von OS/M ist begründet in der Verfügbarkeit einer grossen Anzahl von Dienstprogrammen, von Compilern/Interpretern fuer höhere Programmiersprachen und Applikationssoftwaresystemen. Sie betreffen den Komplex der kommerziellen Datenverarbeitung, den Komplex der Mikrorechnerentwicklungsoftware, aber auch viele weitere Anwendungslinien.

Die Schnittstelle von OS/M auf dem P8000 zu den unter seiner Steuerung laufenden Programmen ist voll kompatibel mit der Schnittstelle des Betriebssystems CP/M Version 2.2. Mit zwei optional an das P8000 anschliessbaren Standardlaufwerken wird die Dateieubnahme von CP/M-Standarddisketten einfacher Aufzeichnungsdichte (Standardformat/3740) gewaehrleistet. Damit ist eine Vielzahl hochwertiger Software, wie Textverarbeitung, Kalkulation, Datenbanksysteme u.a.m. unter dem Betriebssystem OS/M nutzbar.

Ausserdem ist die Kompatibilitaet mit dem in der DDR verfügbaren Betriebssystem SCPX 1526 fuer die Buerocomputer A5120/30 und PC1715 gegeben.

Vom Betriebssystem OS/M werden standardmaessig folgende Ein-/Ausgabegeraete des P8000 bedient:

- ein serieller V24-Kanal fuer die OS/M-Systemkonsole
- ein bis maximal vier Floppy-Disk-Laufwerke
- ein Drucker (Typen robotron 1152, 1157, k631x)
- ein serieller asynchroner Koppelkanal zur Systemkopplung (Kommunikationsschnittstelle)

Das Betriebssystem OS/M verwaltet einen Operationsspeicherbereich (RAM) von 64 KByte.

Alle Dienstprogramme koennen unter ihrem Namen aufgerufen werden. Sie werden dann von der Diskette geladen und abgearbeitet. Dabei werden keinerlei Unterschiede zwischen Dienst- und Anwenderprogramme gemacht.

Fuer die Programmentwicklung gehoeren ein Editor, K5801K80-Assembler, Lader und ein Testhilfeprogramm zum Systembestand.

Das Erarbeiten von Anwenderprogrammen hat vor allem dann Bedeutung, wenn nicht auf das breite Spektrum von Standardsoftware zurueckgegriffen werden kann, die Standardsoftware zu modifizieren ist oder Sonderfunktionen zu loesen sind.

Der zur Verfügung stehende Assembler gestattet das Programmieren auf Maschinenkodeniveau. Dies ermöglicht effektivste Programme bezueglich Laufzeit und Speicherbedarf.

Das ist vor allem wichtig bei der Steuerung von E/A-Geraeten. Der Lader des Systems verarbeitet die ueber den Assembler erzeugten Dateien in eine entsprechende Befehlsdatei, die dann durch Eingabe des Namens gestartet werden kann. Der Debugger gestattet die Testung der Programme im Schrittbetrieb oder in Echtzeit.

Fuer die Formatierung von Disketten steht ein Formatierungsprogramm zur Verfuegung.

Das Textverarbeitungssystem TP und das Datenbanksystem REDABAS koennen ebenfalls verwendet werden.

Die Bedienung ist fuer das System selbst sowie fuer die System- und Anwenderprogramme unterschiedlich und wird an entsprechender Stelle erlaeutert.

Fehlerbehandlungen des Systems sind ebenfalls einfach und auf das Wesentliche beschraenkt

2. Systemueberblick

Das Betriebssystem OS/M besteht aus drei Basis-Systemkomponenten:

- . BIOS Basic Input/Output System
- : BDOS Basic Disc Operating System
- . CCP Consol Command Processor

Sowohl der CCP-Modul als auch der BDOS-Modul sind die hardwareunabhaengigen Teile des OS/M.

Alle Hardware-Zugriffe, wie Eingabe-/Ausgaberufe zur Floppy-Disk-Einheit, zur Konsole, zum Drucker gehen grundsaeztlich ueber den BIOS-Modul. Hier sind die entsprechenden Treiberprogramme fuer die FD-Laufwerke, fuer die Systemkonsole und fuer den Drucker eingebunden.

2.1. Laden des Systems

Nach dem Einschalten des Rechners ("Kaltstart") oder durch Ausloesen von RESET der CPU wird der Lade-PROM des P8000 aktiviert.

Das Programm, das in diesem Lade-PROM gespeichert ist, wird ab der Adresse 0 abgearbeitet und fuehrt folgende wesentliche Schritte aus:

- Eigentest (wird nur bei Einschalten des Rechners durchgefuehrt)
- Einstellen Interruptmode 2
- Programm aus Lade-PROM in den RAM-Bereich einlesen und abarbeiten
- Systemstart - Laden von Spur 0
(wenn keine Diskette in Laufwerk A eingelegt ist, erfolgt eine Ausschrift in folgender Form:
 Insert System Error
 >
Ist eine Diskette in Laufwerk A eingelegt, die keine Systemdiskette ist, so erfolgt die Ausschrift:
 DISK ERROR
 >
Durch Einlegen einer Systemdiskette in Laufwerk A und Eingabe von 0 bzw. durch Druetzen der Taste RESET kann ein erneuter Systemstart durchgefuehrt werden)
- Abschalten Lade-PROM

Vom Programm Urlader wird das Laden des BIOS-Moduls realisiert.

Die Kaltstartroutine fuehrt u.a. folgende wesentliche Funktionen aus:

- Hardware neu initialisieren
- Meldung "PS000 OS/M 2.2 (C) ZPT/KEAW" wird auf dem Bildschirm angezeigt
- Verbindung zu BIOS (Warmstart) und BDOS werden eingetragen
- CCP und BDOS werden geladen
- Die aktuelle DMA-Adresse wird auf 80H eingestellt
- Sprung zu CCP zur Kommandoeingabeaktivierung

Warmstart ist das Nachladen des CCP und von Teilen des BDOS von der Systemdiskette ohne weitere Veraenderungen im System. Bei einem Warmstart wird das Laufwerk A ausgewaehlt und alle anderen Diskettenlaufwerke in den Zustand R/W (read/write) gebracht. Der Warmstart wird ausgefuehrt bei einem RST 0 (Sprung auf Adresse 0000H) oder nach Eingabe des Zeichens <CTRL> C bei einer Eingabeaufforderung. Danach kehrt das Betriebssystem wieder in den Systemgrundzustand zurueck. Im Systemgrundzustand wird das zuletzt aktuelle logische Laufwerk angewaehlt.

2.2. Speicherkonzept

Das OS/M ist fuer einen Gesamtspeicherbereich von 64 KByte vorgesehen. Die EPROMs auf der 8-Bit-Rechnerkarte sind abgeschaltet. Nach dem Laden des Programms von der Systemdiskette ("Kaltstart" Laufwerk A) ist der zur Verfuegung stehende Speicherbereich in 5 Bereiche aufgeteilt.

BIOS	Adresse: FFFFH (64 KByte - 1) Adresse: 0FOOH
BDOS	
CCP	
TPA	Adresse: 0100H
Systemdatenbereich	Adresse: 0000H

- BIOS:** Basic Input/Output System stellt den variablen, anwendungsspezifischen Teil des Betriebssystems dar. In ihm sind die einzelnen Routinen zur Bedienung der im System eingebundenen physischen E/A-Geraete zusammengefasst.
- BDOS:** Basic Disc Operating System uebernimmt als einer der beiden hardwareunabhaengigen Teile des OS/M die Automatisierung des gesamten Datenaustausches mit den logischen Geraeten (Floppy-Disk-Einheit, Konsole, Drucker). Fuer alle im BDOS veranerkten Funktionen- sie werden als BDOS-System-Calls bezeichnet- existiert ein einheitliches Parameteruebergabeschema. BDOS bedient sich hierbei der physischen BIOS-Routinen.
- CCP:** Consol Command Processor ist analog dem BDOS-Modul ein voll unabhaengeriger Programmmodul, dessen Hauptaufgabe die Befehlsentschluesselung ist. Mit diesem Systemteil werden die residenten Kommandos DIR, REN, TYPE, ERA, SAVE, USER realisiert.
- TPA:** Transient Program Area ist der Speicherbereich, der den transienten Kommandos und Programmen zur Verfuegung steht. Ab der Adresse 100H werden die transienten Kommandos oder Programme geladen und auch bei dieser Adresse gestartet. Dieser Bereich steht ebenfalls dem Anwenderprogramm zur Verfuegung.

Systemdaten-
bereich:

In diesem Bereich sind Adressen, Kennbytes, Spruenge- und Standardpuffer fuer die Arbeit des OS/M abgelegt.

Ein Kommando-/Programmaufruf erfolgt durch die Angabe des Kommandos oder des Dateinamens ohne die Dateitypkennzeichnung ".COM". Nach dieser Kommando- bzw. Programmnameneingabe ist eine Eingabe von Parametern noch moeglich. Diese Parameter werden generell auf den Adressen 0084H bis 00FBH abgelegt, die tatsaechliche Anzahl eingegebener Zeichen (max.123 Zeichen) wird auf der Adresse 0080H gespeichert.

Der Programmierer hat aber darauf zu achten, dass die Adresse 0080H nach einem Kalt- und Warmstart die aktuelle DMA-Adresse fuer Floppy-Disk-Zugriffe ist.

Neben der Speicherung der Parameter ab 0080H werden ausserdem die direkt dem Kommando bzw. Programmnamen moeglicherweise folgende 2 Dateibezeichnungen (Laufwerkskode, Dateiname, Dateityp) auf 005CH und 006CH abgelegt (zwischen Dateiname und Dateityp ist bei der Eingabe ein Punkt einzugeben

Bsp.: A:LAUF TEST.MEX ELF.DE abc).

Die Adresse 005CH ist aber auch als Standardadresse fuer den FCB nach Kalt- und Warmstart definiert und ist ebenfalls vom Programmierer zu beachten.

Adresse	Bedeutung
0000H - 0002H	Sprung an den Sprungbefehl fuer Warmstart in der BIOS-Sprungtabelle. Die Adresse 0001H und 0002H ist die Anfangsadresse BIOS+3H.
0004H - 0004H	Aktuelles logisches Laufwerk (0=A...,3=D)
0005H - 0007H	Sprung an BDOS. Die Adresse 0006H und 0007H ist die Anfangsadresse von BDOS.
0038H - 003AH	Reserviert fuer das Pgr. DDT
005CH - 007CH	FCB - File Control Block (Standard-Dateisteuerblock s.a. 6.4)
0080H - 00FFH	DMA - Direct Memory Access (Standard Ein-/Ausgabe-Puffer s.a. 6.4)

3. OS/M-Funktionen

In diesem Abschnitt werden alle unter OS/M verfügbaren Kommandos behandelt.

Zunächst werden einzelne Abgrenzungen verfügbarer OS/M-Befehle vorgenommen: Kontrollzeichen, residente Befehle (built-in commands), nicht residente Befehle (transient commands).

Anschliessend wird auf das Assemblieren, Laden, Listen, die Fehlerberichtigung, das Abspeichern (ASM, LOAD, DUMP, DDT, SAVE) von Programmen und die Generierung von Steuerdateien (SUBMIT, XSUB) eingegangen. Ebenso wird die Behandlung von Dateien (PIP) erläutert.

Danach folgt eine Kurzbeschreibung aller OS/M-Kommandos sowie der unterstützten Dienstprogramme.

Wie bereits beschrieben, meldet sich OS/M mit der Ausgabe des selektierten Laufwerkes und dem Prompt-Character ":". Mit Eingabe "A:", ..., "D:" wird die Selektierung (permanente Änderung) des aktuellen Laufwerks veranlasst. Das System ist damit zur Befehlsannahme bereit. Nach der Kommandoeingabe können sich je nach Art des Kommandos ein oder mehrere, durch Leerschritte getrennte Parameter, anschliessen. Die durch "RETURN" abgeschlossene Befehlszeile wird dann unmittelbar vom System interpretiert.

Viele Kommandos sind sehr einfach, andere jedoch komplex strukturiert. Alle Befehle sind in Wirklichkeit Programme bzw. Routinen im Maschinencode. Einige Kommandos sind immer resident, andere nicht resident.

Die fünf residenten Kommandos (DIR, ERA, REN, SAVE und TYPE) können immer ausgeführt werden, weil sie sich ständig im OS/M-Betriebssystembereich befinden. Die nicht residenten Kommandos können nur dann ausgeführt werden, wenn sie als Kommandodateien (.COM) existieren. Ein nicht residentes Kommando ist ein Programm im Maschinencode, das kopiert, gelöscht und im Arbeitsspeicher verschoben werden kann und dennoch als Kommando ausgeführt wird.

Eigene nicht residente Kommandos können mit einigen Programmierkenntnissen leicht selbst generiert werden. Jede Dateibezeichnung eines nicht residenten Kommandos hat immer die Dateibezeichnung ".COM". Die Dateikennung darf allerdings bei Aufruf eines solchen Befehls nie mit angegeben werden.

Wenn eine solche Datei kopiert, gelöscht oder sonst verändert werden soll, muss neben dem Dateinamen zusätzlich die ".COM"-Kennung angegeben werden.

Das OS/M ermöglicht verschiedene Zeilen-Editor-Funktionen bei der Eingabe von Kommandozeilen.

Die in der nachfolgenden Tabelle beschriebenen Steuerzeichen können während der Eingabe eines OS/M-Kommandos bzw. einer OS/M-Routine eingegeben werden.

Funktion	Ausloesende Taste	Operation
Loeschen des letzten Zeichens	BACKSPACE <CTRL> H	Cursor wiederholt das Zeichen
Loeschen der gesamten Zeile (Zeile wird weiter angezeigt)	<CTRL> U	Cursor springt zur naechsten Zeile und zeigt durch "#" Bereitschaft fuer neue Eingabe an.
Loeschen der gesamten Zeile	<CTRL> X	Cursor springt zum Anfang der Zeile zurueck und loescht diese
Nochmaliges Anzeigen der Befehlszeile	<CTRL> R	Zeigt eine bereinigte Zeile an (nach Fehlerkorrektur)
Abschluss (Ausfuehren der Befehlszeile)	RETURN	der eingegebene Befehl wird ausgefuehrt. Cursor springt zur naechsten Zeile fuer neue Befehlseingabe
Eingabe einer langen Befehlszeile (laenger als Zeilenlaenge des Bildschirmterminals)	<CTRL> E	Cursor springt zur naechsten Zeile, ohne dass der Befehl ausgefuehrt wird
Warnstart des Systems	<CTRL> C	Erneuter Start von OS/M (Warnstart). Abbruch des gerade ausgefuehrten Befehls
Beenden von PIP-Operationen	RETURN <CTRL> H	Beendet PIP und uebergibt die Steuerung wieder an OS/M (auch mit <CTRL> C)
Sendet alle Zeichen, die am Bildschirm gezeigt werden, zum Drucker; Beenden des Parallelbetriebes des Druckers	<CTRL> P	Parallelschaltung von Drucker und Bildschirm; EIN-/AUS-Schalter fuer Drucker ECHO
Abbruch und erneuter Start des Listings am Bildschirm	<CTRL> S	EIN-/AUS-Schalter fuer unterbrochenen Bildschirmausgaben

Start des Listings
am Bildschirm

<CTRL> Q

BIN-Schalter fuer
mit <CTRL> S unter-
brochene Bild-
schirmausgaben

Wird waehrend der Arbeit mit OS/M eine Diskette gewechselt, so ist nach dem Wechsel mit <CTRL> C ein Warmstart auszuloesen, um die vom System logisch schreibgeschuetzte Diskette in den "read/write"-Status zu bringen. Die Kommandos unterteilen sich in 2 Gruppen.

3.1. Residente Kommandos

Diese Kommandos sind die im OS/M direkt installierten Kommandos. Sie sind Bestandteil des von der Systemdiskette in den Speicher geladenen Betriebssystems und werden als residente Kommandos (built-in commands) bezeichnet. Auf sie kann nach einem Kalt- oder Warmstart sofort zugegriffen werden.

Diese Kommandos sind:

DIR	Ausgabe der Directory einer Diskette
ERA	Loeschen von Diskettendateien
REN	Umbenennen von Diskettendateien
TYPE	Ausgabe von ASCII-Dateien auf die Konsole
SAVE	Anlegen einer Speicherabzugsdatei (ab 100H)
USER	Einteilung einer Diskette in unterschiedliche Benutzerbereiche

3.2. Nicht residente Kommandos

Diese Kommandos (transient commands) sind auf der Diskette als Datei mit dem Dateityp ".COM" verfuegbar. Sie werden vom aktuellen Laufwerk in den Anwenderspeicher durch Eingabe des Dateinamens geladen.

Die standardmaessig verfuegbaren, nicht residenten Kommandos fuer OS/M sind:

SYSGEN, PIP, ED, STAT, ASM, LOAD, DUMP, DDT, SUBMIT
XSUB, FORMAT, COPYDISK, GENHEX, SETLP, SETPD, ERRORS

Diese sind auf der Systemdiskette als ".COM"-Datei verfuegbar.

3.2.1. Kommando zur Systemgenerierung

Das Kommando SYSGEN wird zur Erstellung einer OS/M-Kopie eingesetzt. Mit dem PIP-Kommando koennen alle nicht residenten (transiente) Kommandos auf eine neue Diskette uebertragen werden. Dies sind alles Dateien mit der Dateikennung ".COM".

Das eigentliche OS/M-System, das sich auf 3 reservierten Spuren der Systemdiskette befindet, kann mit dem PIP-Kommando nicht kopiert werden. Erst mit dem SYSGEN-Kommando werden die 3 Spuren auf die neue Diskette uebertragen. Das SYSGEN-Kommando generiert auf einer neuen Diskette ein System, so dass aus einer normalen Diskette eine Systemdiskette wird. SYSGEN holt das System von der Originaldiskette in den Arbeitsspeicher (transient program area TPA) und speichert es von dort auf die reservierten Spuren der Systemdiskette.

Nach Aufruf des Kommandos SYSGEN muss das Quellaufwerk angegeben werden, von dem kopiert werden soll (i.allg. Laufwerk A). Danach wird das Betriebssystem in die TPA eingelesen. Ohne Angabe eines Quellaufwerks wird das Einlesen uebersprungen, was notwendig ist, wenn in der TPA schon ein Betriebssystem gespeichert wurde. Danach wird das Ziellaufwerk eingegeben und anschliessend der Arbeitsspeicher auf die Systemspuren geschrieben. Auf eine so erstellte Diskette kann nun mit dem PIP-Kommando (B:=A:*. *[V]) der transiente Teil des Betriebssystems kopiert werden.

3.2.2. Starten von Befehlsdateien

In vielen Faellen ist es nuetzlich oder auch notwendig eine Reihe von OS/M-Kommandos oder -Routinen automatisch wie Programminstruktionen ablaufen zu lassen.

OS/M stellt hierfuer den nicht residenten Befehl SUBMIT bereit. Das SUBMIT-Kommando akzeptiert eine Datei mit dem Dateityp ".SUB" und fuehrt die dort aufgefuehrten Befehle, die ausserdem bei der eigentlichen Ausfuehrung noch mit Argumenten ergaenzt werden koennen, der Reihe nach aus. Eine ".SUB"-Datei kann mit dem Editor ED generiert werden. Die Kommando-Datei "BEISPIEL.SUB" enthaelt z.B. folgende Zeilen:

```
DIR #1:#2
PIP A:=#1:#2
```

Die Zeichen #1 und #2 stellen wahlweise eingebbare Argumente dar. Sie entsprechen Variablen eines Programms und koennen waehrend des Ablaufs mit Argumenten gefuehlt werden. Mit der Eingabe

```
A:SUBMIT BEISPIEL B DATEI.TXT
```

wuerde #1 mit dem Laufwerkskennzeichen und #2 durch eine Dateibezeichnung ersetzt.

Bei der Ausfuehrung der SUBMIT-Routine wird der Wert B

anstelle von #1 und DATEI.TXT anstelle von #2 gesetzt. Es wird dann die Dateibezeichnung in der Directory von Laufwerk B angezeigt und danach die Datei "DATEI.TXT" von B nach A mit der gleichen Dateibezeichnung kopiert.

Die SUBMIT-Kommandodateien koennen nur von Laufwerk A gestartet werden, in den Kommandodateien kann aber auf beliebige Laufwerke umgeschaltet werden.

OS/M verfuegt ueber eine zusaetzliche Funktion im nicht residenten Kommando XSUB. Die XSUB-Routine erlaubt die spaetere Hinzufuegung weitere Eingabeinformationen in den gestarteten Programmen.

Mit der XSUB-Funktion (XSUB.COM ist notwendig) koennen Eingaben waehrend der Abarbeitung von OS/M-Kommandos in den Kommandodateien simuliert werden.

Enthaelt die Datei SAVER.SUB z.B. folgende Zeilen:

```
XSUB
DDT
I#1.HEX
R
GO
SAVE 1 #2.COM
```

so werden mit SUBMIT SAVER X Y im DDT-Kommando (Debugger) die Eingaben I X.HEX, R und GO ausgefuehrt.

Sind alle Kommandos durchgefuehrt worden, uebernimmt der CCP wieder die Steuerung und wartet auf neue Kommandos. Allerdings bleibt die XSUB-Routine weiterhin aktiv, bis ein Programm geladen wird, das den von XSUB belegten Bereich ueberschreibt oder bis ein erneuter Kaltstart bzw. Systemstart durchgefuehrt wird.

3.2.3. Assemblieren, Laden und Auflisten von Programmen

Die Programme ASM, LOAD und DUMP sind Begriffe fuer Operationen, die es ermoeöglichen, Assemblerprogramme zu starten und auszufuehren.

Mit dem ASM- und dem LOAD-Kommando kann eine Assembler-Quellprogrammdatei in ein neu definiertes Kommando ueberfuehrt werden. Mit dem DUMP-Kommando kann der Inhalt einer solchen Datei aufgelistet werden.

Das ASM-Kommando startet den K5801K80-Assembler, der als Datei "ASM.COM" vorliegt. Mit dem ASM-Kommando koennen K5801K80-Quellprogramme des Typs ".ASM" in Maschinenprogramme uebersetzt werden.

Existiert beispielsweise die Datei "DAT.ASM", so bewirkt das Kommando

```
A:ASM DAT ,
```

dass der Inhalt von "DAT.ASM" assembliert wird. Treten waehrend der Uebersetzung Fehler auf, so werden diese auf der Konsole angezeigt. Es wird automatisch die Datei "DAT.PRN" erzeugt, welche das Quellprogramm und die ent-

sprechenden Adressen und Befehle in Maschinencode enthaelt. Ausserdem wird die Datei "DAT.HEX" erzeugt, die das Maschineprogramm im Standard-Hex-Format enthaelt.

Wenn eine ".HEX"-Datei durch die ASM-Routine generiert wurde, kann diese in eine ".COM"-Datei (direkt ausfuehrbarer, nicht residenter Befehl) durch die Laderoutine LOAD umgewandelt werden. Das LOAD-Kommando ist ein nicht residenter Kommando mit der vollstaendigen Dateibezeichnung LOAD.COM.

A:LOAD DAT

erzeugt z.B. aus der Datei DAT.HEX die Datei DAT.COM.

Das DUMP-Kommando listet den Inhalt einer Datei im hexadezimalen Format am Bildschirm auf. In einem DUMP-Kommando muss die vollstaendige Dateibezeichnung angegeben werden (Dateiname und Dateityp).

3.2.4. Ausfuehren, Testen und Abspeichern von Programmen

Ist von einer ".HEX"-Datei ueber das LOAD-Kommando eine ".COM"-Datei generiert worden, so kann dieses Programm durch die Eingabe des Dateinamens ausgefuehrt werden.

A:DAT

Dieser Befehl startet das Programm DAT.COM (von Laufwerk A)

Das mit dem OS/M-System mitgelieferte Programm "DDT.COM" dient der Fehlersuche und Fehlerbeseitigung in Programmen. DDT bringt jede Datei in den Arbeitsspeicher und erlaubt Manipulationen ueber spezielle DDT-Befehle. Es kann fuer die Korrektur von Fehlern oder fuer das Laden von Dateien in den Speicher verwendet werden.

Mit dem SAVE-Kommando kann eine 1:1 Abbildung eines gewuenschten Speicherbereichs (TPA) unabhaengig vom Inhalt auf die Diskette gespeichert werden.

Das SAVE-Kommando uebernimmt eine oder mehrere Speicherseiten des TPA-Bereichs und legt diese auf der Diskette unter einer spezifizierten Dateibezeichnung ab (eine Seite enthaelt 256 Bytes).

Das Kommando hat die folgende Form:

SAVE p Dateibezeichnung

Mit p wird die Zahl (dezimal) der Speicherseiten angegeben, wobei p innerhalb des DDT-Programms aus der Angabe des R-Befehls im Feld "Next" errechnet werden kann. Diese Zahl gibt die naechst hoechere Adresse nach dem Bereich des bearbeitenden Programms im TPA-Bereich an. Die Umrechnung in Seiten kann wie folgt vorgenommen werden:

Die unter "Next" angegebene Adresse um 1 verringern und

diese Hexadezimalzahl (entspricht nun der Endadresse des Programms) in eine Dezimalziffer umwandeln. Die errechnete Zahl ist die Anzahl der Seiten.

3.2.5. Handhabung von Dateien mit dem PIP-Kommando

PIP ist die Abkuerzung fuer Peripheral Interchange Program. PIP ist ein leistungsfaehtiges Programm zur Abwicklung allgemeiner Dateiuebertragungen. PIP deckt ein breites Spektrum von Funktionen ab:

- . Uebertragung von Dateien zwischen Diskettenlaufwerken und anderen externen Gerueten.
- . Dateigruppenuebertragungen und Dateikonzentrationen
- . Dateiverarbeitung, -ueberpruefung, -formatierung

Die PIP-Routine kann sowohl auf Teile einer Datei als auch auf mehrere Dateien angewendet werden.

3.2.5.1. Kopieren von Dateien

PIP kann auf 2 Arten gestartet werden;

1. als einzelne Befehlszeile
2. als "Programm"

```
A:PIP B:KOPIE1=DATEI1
A:
```

PIP wird hierbei als Einzelbefehl eingesetzt. Dieser Befehl teilt der PIP-Routine mit, dass eine Kopie der Datei DATEI1 erstellt und als Datei KOPIE1 auf der Diskette von Laufwerk B generiert werden soll.

```
A:PIP
*B:KOPIE2=A:DATEI2
*A:=B:BEISPIEL
*RETURN
A:
```

PIP laeuft in diesem Beispiel als "Programm" ab, so dass hierin die einzelnen Kopierbefehle als Folge eingegeben werden koennen. Mit der ersten Zeile wird beispielsweise die Erstellung einer Kopie KOPIE2 auf Laufwerk B von Datei DATEI2 auf Laufwerk A veranlasst. Mit der naechsten Zeile wird eine Kopie von der Datei BEISPIEL von Laufwerk B erstellt und mit der gleichen Dateizeichnung auf der Diskette im Laufwerk A gespeichert.

Dateigruppenbefehle im PIP (multiple PIP-commands) erlauben das Kopieren von Dateigruppen. Fuer das Kopieren von Dateigruppen werden im PIP die Sondersymbole "*" und "?" bereitgestellt. Das Zeichen "?" kann in einer Dateibezeichnung an beliebiger Stelle gesetzt werden, wo es jedes erlaubte Zeichen ersetzt. Das zweite Gruppensymbol "*" kann noch effektiver eingesetzt werden. Es korrespondiert mit allen Zeichen im ganzen Feld, unabhaengig von der Anzahl der Zeichen.

A:PIP A:=B:DATEI?

Mit diesem einzigen Befehl werden mit Hilfe des Gruppensymbols "?" z.B. folgende Dateien DATEI1, DATEI2, DATEI3 von Laufwerk B nach Laufwerk A kopiert.

A:PIP B:=*.COM

Mit Hilfe dieses PIP-Kommandos werden alle Dateien vom Dateityp ".COM" von Laufwerk A nach Laufwerk B kopiert. Fuer das Kopieren aller Dateien kann eine umfassende Dateigruppenbezeichnung verwendet werden. Sollen z.B. alle Dateien der Diskette vom Laufwerk A auf eine Diskette im Laufwerk B kopiert werden, genuegt die Anweisung:

A:PIP B:=A:*.*

3.2.5.2. Kopieroperationen zu einzelnen Geraeteeinheiten

Die PIP-Routine verfuegt ueber allgemein verwendbare Uebertragungsmoeglichkeiten, die nicht nur die Uebertragung von Laufwerk zu Laufwerk, sondern auch zwischen verschiedene Geraeteeinheiten ermoeglichen. So kann eine Datei z.B. vom Diskettenlaufwerk zum Drucker oder zum Bildschirm uebertragen werden. Ausgabegeraete koennen logischerweise nur als Zieldateien und Eingabegeraete nur als Quelldateien verwendet werden.

Fuer die problemlose Abwicklung von PIP-Routinen muessen die Geraeteeinheiten definiert zugewiesen werden. Ein physikalischer Geraetenname ist der aktuelle Name einer vom System selektierten Geraeteeinheit. Ein logischer Geraetenname entspricht einem symbolischen Namen fuer eine Gruppe von Geraeteeinheiten.

In PIP-Kommandos sind die folgenden logischen Namen erlaubt:

CON:	fuer Konsole (Eingabe/Ausgabe)
LST:	fuer Drucker (Ausgabe)

Diese logischen Gerætenamen koennen problemlos in PIP-Kommandos Verwendung finden.

```
A:PIP
*CON:=BEISPIEL1
*LST:=B:BEISPIEL2
*
A:
```

Der erste PIP-Befehl sendet eine Kopie der Datei BEISPIEL1 zur Konsoleinheit (Bildschirm). Der zweite PIP-Befehl uebertraegt eine Kopie der Datei BEISPIEL2 von Laufwerk B zur Druckeinheit.

3.2.5.3. Spezielle Kopieroperationen

Zur Abwicklung spezieller Uebertragungsprozeduren werden im PIP zusaetzliche Gerætenamen verwendet:

- NUL: sendet 40 "Nullen" (ASCII-Code) zur Geræteeinheit
- EOF: Sendet ein Dateiendezeichen (EOF=ASCII-<CTRL> Z) zur Geræteeinheit fuer Nicht-ASCII-Files. ASCII-Files werden immer mit EOF (<CTRL> Z) abgeschlossen.
- PRN: entspricht der LST-Einheit mit der Ausnahme, dass die Tabulatorsteuerungen alle 8 Spalten festgesetzt, die Zeilen numeriert und die ausgedruckten Seiten ueber automatische Formularvershube (alle 60 Zeilen) gesteuert werden.
- INP: Aufruf eines vom Benutzer geschriebenes spezielles Eingabeprogramm auf Adresse 103H und Dateiuebergabe im Speicherbereich ab Position 109H (Paritaetsbit muss 0 sein)
- OUT: Aufruf eines vom Benutzer geschriebenes spezielles Ausgabeprogramm auf Adresse 106H. Daten werden im CPU-Register C bereitgestellt. Die Speicherpositionen 109H bis 1FFH sind im PIP-Programm nicht benutzt und koennen deshalb durch Routinen fuer spezielle Geræetreiber ersetzt werden.

Die Unterscheidung der Dateiformen, ob es sich um eine Datei im HEX-Format (2 Hex-Ziffern je Byte) oder im ASCII-Format (ein Zeichen je Byte) handelt, ist fuer die Verwendung von speziellen Kopierbefehlen sehr wichtig. Das sind z.B. solche Anwendungen wie das Ersetzen von Grossbuchstaben durch Kleinbuchstaben, das Kopieren einzelner Dateibe-
reiche, das Aneinanderfuegen mehrerer Dateien u.a.m.

Das Zusammenfuegen mehrerer Textdateien zu einer einzigen Datei ist eine haeufige Anwendung des PIP-Kommandos.

```
A:PIP
*GESAMT.TXT=TEXT1.TXT,TEXT2.TXT
```

In diesem Beispiel werden die 2 Dateien TEXT1.TXT, TEXT2.TXT in der Reihenfolge kopiert und in der Datei GESAMT.TXT zusammengefasst. Voraussetzung fuer ein solche PIP-Operation ist immer, dass es sich hierbei um ASCII-Dateien handelt.

Beim Zusammenfuegen von Dateien, die nicht ASCII-Dateien sind, muss ein spezieller Uebertragungsparameter mit angegeben werden.

```
A:PIP
*GESAMT.HEX=TEMP1[0],TEMP2[0],TEMP3
```

Der O-Parameter veranlasst die PIP-Routine, das Ende der Datei TEMP1 und TEMP2 zu erkennen und das Verbinden einzuleiten.

Die Parameter fuer spezielle Kopieroperationen werden in eckigen Klammern [] und direkt nach den entsprechenden Dateien gesetzt:

- B Uebertragung in "block mode". Die Dateien werden in einen Puffer geschoben, bis ein ASCII-"X-off"-Zeichen (<CTRL> S) erkannt wird.
- Dn Waehrend der Uebertragung werden die ueber die "n-te" Spalte am Bildschirm hinausgehenden Zeichen geloescht.
- E Die Kopiervorgaenge werden in der Reihenfolge am Bildschirm gemeldet (echoed).
- F Alle Formularvorschubzeichen werden ausgefiltert
- H Ueberpruefung der Daten auf zulaessiges HEX-Format.
- I Ignorieren von "OOH"-Saetzen innerhalb der Uebertragung von HEX-Dateien.
- L Umsetzung aller Grossbuchstaben in Kleinbuchstaben.
- N Einfuegen von Zeilennummern in eine Dateikopie, beginnend mit der Zeile 1.
- O Die PIP-Routine ignoriert beim Zusammenfassen von Objektdateien (keine ASCII-Dateien) jeweils das physische Datenende.
- Pn Generierung eines Seitenvorschubs nach jeder n-ten Zeile

- Q string <CTRL> Z Abbruch der Uebertragung von einer Geraeteinheit oder Datei, sobald die angegebene Zeichenkette (string) erkannt wird. Mit diesem Befehl kann das Kopieren einzelner Dateiabschnitte erreicht werden.
- S string <CTRL> Z Beginn der Uebertragung von einer Geraeteinheit oder Datei, sobald die angegebene Zeichenkette erkannt wird. Mit diesem Parameter kann ein Dateibereich ab einer bestimmten Startposition uebertragen werden.
- Tn Die Tabulatorabstaende werden waehrend der Uebertragung auf "n"-Spalten gesetzt.
- U Umsetzung aller Kleinbuchstaben in Grossbuchstaben.
- V Alle in die Kopie uebertragenen Daten werden nochmals mit der Originaldatei verglichen (verify) und die ordnungsgemaesse Uebertragung gemeldet.
- Z Das Paritaetsbit eingelesener Daten wird auf Null gesetzt.
- Gn Abruf einer Datei vom Benutzerbereich "n"
- W Ueberschreiben von Nur-Lese-Dateien (R/O-Attribut) werden ignoriert.
- R Lesen von Systemdateien (Daten mit "SYS"-Attributen).

3.3. Kurzbeschreibung aller OS/M-Kommandos

Die einzelnen Kommandos und Programme sind in alphabetischer Reihenfolge geordnet und koennen ueber die Kommando- und Programmnamen leicht gefunden werden. Jede Einzelbeschreibung ist nach dem gleichen Format aufgebaut. In Kurzform wird der Zweck des Kommandos bzw. Programms und darunter in Klammern die Dateibezeichnung mit dem Dateityp (.COM, residentes Programm) angegeben.

Als naechstes ist das Format des Befehls mit den moeglichen Argumenten angezeigt. Ist dieses Argument in eckigen Klammern "[...]" angegeben, so kann es auch weggelassen werden (optional), sonst ist es zwingend notwendig. In einigen Faellen ist ein Teil eines Arguments wahlweise, waehrend ein anderes zwingend notwendig ist (z.B. [d:] filename, wobei [d:] fakultativ und filename zwingend ist. Die geschweiften Klammern "{...}" zeigen eine Auswahlmoeglichkeit an.

Weiterhin sind einige generelle Konventionen unterstellt. So ist zum Beispiel vereinbart, dass Dateiargumente, ob

benoetigt oder nicht, in jedem Fall eine Laufwerkskennzeichnung enthalten koennen (z.B. B:DATEI). Hiermit wird angegeben, dass auch ein anderes als das selektierte Laufwerk angesprochen werden kann. Wegen dieser generellen Vereinbarung sind die Laufwerkssymbole in den Befehlsbeschreibungen weggelassen worden.

Eine andere Vereinbarung ist die Abmachung, dass alle ".COM"-Dateien direkt ausgefuehrt werden koennen. Soll beispielsweise "BEISPIEL.COM" gestartet werden, so genuegt die Eingabe von "BEISPIEL". Wenn "BEISPIEL.COM" auf der Diskette von Laufwerk B abgespeichert ist und Laufwerk B nicht selektiert ist, dann muss "B:BEISPIEL" eingegeben werden. Eine Kommandoeingabe (transientes Kommando) wird mit einem "RETURN" abgeschlossen.

Das Symbol "^" mit einem nachfolgenden Buchstaben steht fuer ein Steuerzeichen (CTRL-Zeichen), das durch gleichzeitiges Druucken der <CTRL>-Taste mit einer anderen Taste (z.B. ^C) erzeugt wird.

Assemblieren einer Datei

(ASM.COM;Standard-Dienstprogramm)

FORMAT:

1. ASM Dateiname
2. ASM Dateiname [.shp]

ARGUMENTE:

Dateiname Der Name einer "ASM"-Quelldatei, die Assembler-Informationen und ASCII-Texte enthaelt. ASM sucht nach der Dateibezeichnung "Dateiname.ASM", so dass der Dateityp ".ASM" nicht mit angegeben werden muss.

.shp Sind wahlweise angebbare Parameter. Fuer s (source) muss der Buchstabe des Laufwerkes gesetzt werden (A,B,C,D), das die ASM-Quelldatei enthaelt (falls nicht auf dem selektierten Laufwerk). Fuer h (hex) muss entweder das Laufwerk (A,B,C,D) fuer die durch ASM erzeugte "HEX"-Datei oder "Z" angegeben werden, wobei das Generieren einer "HEX"-Datei unterbunden wird. Mit p (print) wird das Laufwerk fuer die ebenfalls von ASM erzeugte "PRN"-Datei, X fuer das Auflisten der "PRN"-Datei auf dem Bildschirm oder "Z" angegeben, wodurch das Generieren einer "PRN"-Datei verhindert wird.

BESCHREIBUNG:

Der Assembler (ASM.COM) generiert aus einer Quelldatei mit K5801K80-Befehlen eine Zieldatei mit Maschinencode und dem Dateityp ".HEX". Diese Objektdatei kann dann mit dem LOAD-Befehl in das System als transientes Kommando geladen werden. Zusaetzlich wird noch von dem Assembler eine LIST-Datei mit dem Dateityp ".PRN" generiert. Sie enthaelt alle Befehlszeichen des Assembler-Quellprogramms in den entsprechenden Maschinenbefehlen in hexadezimaler Darstellung sowie eventuelle Fehlermeldungen.

Assembler-Fehlermeldungen:

NO SOURCE FILE PRESENT	ASM findet nicht die spezifizizierte Quelldatei oder die Datei hat nicht den Dateityp ".ASM".
NO DIRECTORY SPACE	Directory der Diskette ist voll
SOURCE FILE NAME ERROR	Falsche Eingabe des Dateinamens
SOURCE FILE READ ERROR	Die Quelldatei kann von der ASM-Routine nicht gelesen werden.

OUTPUT FILE WRITE ERROR Die Zieldatei (".HEX" oder ".PRN") kann nicht auf die Diskette geschrieben werden (Diskette z.B. voll).

CANNOT CLOSE FILE Die Zieldatei (".HEX" oder ".PRN") kann nicht geschlossen werden (z.B. Diskette R/O).

ANWENDUNG:

Wenn die "ASM"-Quelldatei sich auf dem selektierten Laufwerk befindet und dort auch die ".HEX"- und die ".PRN"-Datei generiert werden sollen, kann das Format 1 zur Anwendung kommen. Ansonsten muss Format 2 mit allen Parametern eingegeben werden.

In beiden Formaten wird vom Assembler das Quellprogramm in die hexadezimale Notation uebersetzt. Sie entspricht dem Maschinenkode im Binuerkode. Tritt waehrend des Assemblierens ein Fehler auf, so werden die Nummer der fehlerhaften Befehlszeile und ein Fehlerkode angezeigt.

BEISPIELE:

A:ASM PROG

(Assemblieren von PROG.ASM vom selektierten Laufwerk)

A:ASM DOTHIS.ABZ

(Assemblieren der Datei DOTHIS.ASM vom Laufwerk A, Speichern DOTHIS.HEX auf Laufwerk B und keine Generierung der LIST-Datei DOTHIS.PRN)

Physisches Kopieren von Disketteninhalten
(COPYDISK;Standard-Dienstprogramm)

FORMAT:

COPYDISK

BESCHREIBUNG:

COPYDISK.COM ist ein dialogorientiertes Programm zum physischen Kopieren des gesamten Disketteninhalts vom Quelllaufwerk zum Ziellaufwerk mit der optionalen Möglichkeit eines Fehlersuchlaufs des Quelllaufwerks.

ANWENDUNG:

Start des Programms COPYDISK.COM

A: COPYDISK

copy for formatted disks track by track
Type source disk (A,B,...):A

Die Frage nach dem Quelllaufwerk (Type source disk) muss mit der entsprechenden Laufwerksbezeichnung beantwortet werden (z.B. A).

Type destination disk

RETURN=verify disk for errors:B

Die Frage nach dem Ziellaufwerk kann mit der entsprechenden Laufwerksbezeichnung oder mit einem RETURN erfolgen.

Die Beantwortung mit RETURN veranlasst ein Fehlersuchlauf des Quelllaufwerks (z.B. A).

Load disk in A

READY (Y/N)?

Die Beantwortung mit 'Y' startet den Fehlersuchlauf

Verify disk in A

Die Beantwortung der Laufwerksfrage mit der entsprechenden Laufwerksbezeichnung (Type destination disk) (z.B. B) veranlasst einen Kopierlauf.

Load source disk in A

Load destination disk in B

READY (Y/N)?

Die Beantwortung mit 'Y' initiiert den Kopiervorgang.

Copy A to B

Wenn ein Kopiervorgang mit Beteiligung des Laufwerks A erfolgte, meldet sich das Programm ueber

Ready for repeat (Y/N)?

zurueck. Der physische Inhalt der Quelldiskette (z.B. A) wurde auf die Zieldiskette (z.B. B) kopiert. Die Beantwortung von "Ready for repeat (Y/N)? mit 'Y' kann eine Wiederholung des Kopiervorganges erfolgen, anderenfalls erfolgt ein Warmstart (Systemdiskette in A) des Systems. Ist ein Kopiervorgang ohne Beteiligung des Laufwerks A erfolgt, wird ein Warmstart initiiert.

Auftretende Kopierfehler generieren Fehlerausschriften in folgender Form:

source and destination disk are not compatible	- Disketten sind nicht kompatibel
Drive x not ready	- Laufwerk ist nicht bereit
source and destination are the same drives	- Laufwerksbezeichnung fuer Ziel- und Quelldatei ist gleich
source disk Error nn track yyy	- Fehler Quelldiskette
destination disk Error nn track yyy	- Fehler Zieldiskette
nn - UPCS Completion Code	C1 - ungueltiger Requestkode C2 - Laufwerk nicht bereit C3 - Schreibschutz Diskette C4 - Sektorfehler C5 - Spurfehler
x - Laufwerksname (drive)	
yyy - Spurnummer (track)	

BEISPIEL:

```

.
A: COPYDISK
copy for formatted disks track by track
Type source disk (A,B,...):A
Type destination disk
RETURN=verify source disk for errors:B
Load source disk in A
Load destination disk in B
READY (Y/N)?Y
Copy A to B
Ready for repeat (Y/N)?N
A:

```

(Kopieren Diskette in Laufwerk A nach Diskette in Laufwerk B)

Debug - Ausfuehren des Debugger-Programms
fuer das Laden, Aendern und Testen von Programmen

(DDT.COM;Standard-Dienstprogramm)

FORMAT:

DDT Dateibezeichnung

ARGUMENT:

Dateibe- wahlweise angebbares Argument fuer eine
zeichnung Datei mit dem gewuenschten Dateinamen und
 den spezifizierten Dateityp ".COM" oder
 ".HEX"

BESCHREIBUNG:

DDT ersetzt den CCP im Arbeitsspeicher und laedt die gewahlte Datei in den TPA.

DDT verfuegt ueber eigene Befehle fuer das Einfuegen von Zeichen, das Anzeigen von Speicherinhalten, Sichern von Speicherinhalten, Setzen von Unterbrechungspunkten und andere Funktionen zur Fehlererkennung und -beseitigung. Der Aufruf ist in folgender Form moeglich:

DDT

DDT filename.HEX

DDT filename.COM

Folgende Kommandos stehen im DDT zur Verfuegung. Alle Zahlenangaben (Kleinbuchstaben) werden als Hexadezimalzahlen interpretiert.

As Programmeingabe mit Zeilenassembler ab Adresse s. Verlassen des Assemblermodes mit "RETURN".

D Ds Ds,f Hexadump von Adresse s bis f

Fs,f,c Fuellen des Speichers von Adresse s bis f mit Constante c

Gs Gs,b Starten eines Echtzeitlaufes von akt. Adresse
Gs,b,c oder Startadresse s mit Angabe von Haltepunkten b
Gb G,b,c und c

I filen. Laden einer Datei in den TPA-Bereich

L Ls Ls,f Ausgabe des Speicherinhaltes als Assemblerprogramm

Ms,f,d Verschieben des Speicherblocks von s bis f auf neue Adresse ab d.

R Rb Lesen von dem mit dem I-Kommando vorbereiteten ".COM"- oder ".HEX"-File. Mit b kann eine Offsetadresse angegeben werden. ".COM"-Files werden ab 100H+b, ".HEX"-Files auf ihrer Ladeadresse+b

abgelegt.

- Ss Schreiben von Speicherinhalten ab Adresse s. Ende mit ".".
- T Tn Programmablauf von n-Schritten (Befehlen) mit eingeschalteter Tracefunktion
- U Un Programmablauf von n-Programmbefehlen ohne eingeschaltete Tracefunktion
- X Angabe des aktuellen Inhalts der CPU-Register
- Xr Aendern des Inhalts des CPU-Registers

ANWENDUNG:

Laden, Aendern und Testen von Programmen.

BEISPIEL:

A:DDT PIP.COM

Directory - Anzeige einer Liste aller auf der Diskette des selektierten Laufwerks verzeichneten Dateien

(DIR;Residentes Kommando)

FORMAT:

```

DIR      {[Dateibezeichnung]
          [Dateigruppenbezeichnung]}

```

ARGUMENTE:

Dateibezeichnung Wahlweise angebbares Argument zur Uebernahme im DIR-Kommando, das dann nur die angegebene Datei anzeigt.

Dateigruppenbezeichnung Wahlweise angebbares Gruppensymbol zur Uebernahme im DIR-Kommando, das dann alle hiermit korrespondierenden Dateien anzeigt.

Sowohl Dateibezeichnung als auch Dateigruppenbezeichnung koennen ein Laufwerkskennzeichen enthalten.

BESCHREIBUNG:

Ist keine Dateibezeichnung oder Dateigruppenbezeichnung spezifiziert, werden alle Dateien des selektierten Laufwerkes (nur die Dateien mit DIR-Attributen, nicht die Dateien mit SYS-Attributen) vom DIR-Kommando angezeigt.

Wenn eine Dateibezeichnung angegeben ist, wird nur die Datei angezeigt. Bei einer angegebenen Laufwerksbezeichnung (d.h. A:,...,D:) wird das Kommando auf dieses Laufwerk bezogen.

Ist anstelle der Dateibezeichnung eine Dateigruppenbezeichnung angegeben, so werden alle Dateien angezeigt, die mit diesem Gruppensymbol auf dem selektierten Laufwerk korrespondieren.

ANWENDUNG:

Das DIR-Kommando wird vorwiegend zur Pruefung existierender Dateien einer Diskette eingesetzt. Genauso kann nach dem Generieren oder Loeschen von Dateien mit dem DIR-Kommando die richtige Anwendung der Operation nachgeprueft werden. Die Dateien werden im Vierspaltenformat aufgelistet.

BEISPIEL:

```

.
.
A:DIR
A:NOVCPM COM :ASM COM :STAT COM
A:ED COM :LOAD COM :PIP COM
A:SUBMIT COM :SYSGEN COM :FORMAT COM
A:DUMP COM :DDT COM

```

Dateiausdruck auf dem Terminal
Zeigt in hexadezimaler Form den Inhalt einer Diskettendatei

(DUMP.COM;Standard-Dienstprogramm)

FORMAT:

DUMP Dateibezeichnung

ARGUMENT:

Dateibezeich- Die Datei-
nung bezeichnung mit Dateiname und Dateityp

BESCHREIBUNG:

DUMP zeigt in hexadezimaler Darstellung den Inhalt einer beliebigen Diskettendatei auf dem Bildschirm an. Hierbei werden in jeder Zeile links die absolute Adresse und 16 Bytes angezeigt.

ANWENDUNG:

Das DUMP-Programm (DUMP.COM) kann wie ein Kommando mit der gewünschten Datei-
bezeichnung als Argument angegeben werden. Ist die auszugebende Datei nicht auf dem selektierten Laufwerk verfügbar, muss das Laufwerkskennzeichen zusätzlich angegeben werden.

BEISPIELE:

A:DUMP SCRATCH.HEX

.
.
.

A:DUMP B:NAME.COM

Editieren einer Datei

(ED.COM; Standard-Dienstprogramm)

FORMAT:

ED Dateibezeichnung

ARGUMENTE:

Dateibezeichnung Bezeichnung der zu editierenden Datei. Sie muss eine Textdatei oder eine sonstige ASCII-Zeichen enthaltene Datei sein. Die Datei-Bezeichnung muss Dateiname und Dateityp enthalten.

BESCHREIBUNG:

Das Editierprogramm ED.COM kann durch die Eingabe von "ED" und einer nachfolgenden Datei-Bezeichnung gestartet werden. Das ED-Programm generiert einen Editorpuffer und erlaubt dem Benutzer, in diesem Puffer die Modifizierung von Text. Die mit dem Dateinamen korrespondierende ".BAK"-Datei (z.B. TEXT.BAK fuer TEXT.TXT) wird gelöscht, dann kann vom Nutzer Text aus der Quelldatei in den Puffer geladen und dort modifiziert werden. Der erstellte oder modifizierte Text kann während des gesamten Editiervorganges in einer Zwischendatei gespeichert werden. Ebenso kann Text aus Bibliotheksdateien eingefügt werden. Wenn der Editor mit dem E-Befehl verlassen wird, wird die ursprüngliche Quelldatei in eine "Backup"-Datei umbenannt und die modifizierte Quelldatei als neue Quelldatei gespeichert.

Die meisten ED-Befehle bestehen aus speziellen Buchstaben mit einer Zahl oder einem Symbol, das die Anzahl angibt. Diese Befehle werden mit RETURN abgeschlossen. Andere Befehle sind spezielle Steuerbefehle (wie ^Z und ^C), die bei der Eingabe sofort ausgeführt werden. Das ED-Programm meldet sich mit dem Bereitschaftszeichen "*".

Folgende ED-Befehle stehen im ED zur Verfügung:

Befehl	Bedeutung
nA	Laden von n Zeilen (oder 1, wenn n nicht angegeben) aus der Quelldatei in den Editorpuffer. Ein "#" fuer n erzeugt das Laden von 65535 Zeilen (Fuellen des gesamten Puffers) und 0 fuer n laedt soviele Zeilen, bis die Haelfte des Puffers gefuellt ist.
+/-B	Positioniert den CP' an den Beginn (+B) oder das Ende (-B) des Puffers.
+/-nC	Positioniert den CP' um (+) n-Zeichen vorwaerts oder (-) n-Zeichen rueckwaerts.
E	Das Editieren wird beendet. Der E-Befehl sichert den gepufferten Text und den Rest der Quelldatei in eine Uebergangsddatei; benennt die Quelldatei in die entsprechende "BAK"-Datei um und belegt die Uebergangsd-

- tei mit dem Namen der urspruenglichen Quelldatei.
- nFstring {^Z} Sucht die n-te durch die Zeichenkette angegebene Zeichenfolge (wenn n nicht angegeben, wird die erste Zeichenkette gesucht). Nach dem Auffinden der Zeichenkette wird CP' auf das folgende Zeichen gesetzt. Ein folgendes ^Z erlaubt das zusaetzliche Anfuegen weitere Befehle..
- H Das Editieren wird beendet (wie E-Befehl). Die erzeugte neue Quelldatei wird erneut aufgerufen und der Editierbetrieb kann erneut fortgesetzt werden.
- I Einfuegen einer neuen Textzeile nach dem CP' Itext^Z Fuegt die eingegebenen Zeichen ein und positioniert den CP' hinter das zuletzt eingegebene Zeichen.
- nJstring1^Z Nebeneinanderstellen von Zeichenketten. string2^Z Zuerst wird die erste Zeichenkette aufgesucht, die zweite Zeichenkette an die erste angefuegt und alle Zeichen bis zum Erkennen der dritten Zeichenkette werden geloescht. string3 {^Z} Loescht die folgenden (+) bzw. die vorhergehenden (-) n-Zeilen und positioniert den CP' direkt hinter die geloeschten Zeilen
- +/-nK Positioniert den CP' um n-Zeilen vorwaerts (+) oder um n-Zeilen rueckwaerts (-). Ist n=0 wird der CP' an den Beginn der aktuellen Zeile gesetzt.
- +/-nL Fuehrt den mit "string" angegebenen ED-Befehl n-mal in derselben Weise aus. Ist n=0 oder 1, wird der ED-Befehl sooft ausgefuehrt, bis das Ende der Datei erreicht ist oder ein Fehler erkannt wurde.
- nHstring {^Z} Aufsuchen der n-ten mit "text" angegebenen Zeichenfolge im Textpuffer und falls dort nicht vorhanden auch in der Quelldatei. Hierbei werden so viele Zeichen geladen, bis die gesuchte Zeichenfolge gefunden ist. CP' wird an das Ende des gesuchten Texts gesetzt
- 0 Ignoriert den editierten Puffer und erhaelt die Originaldatei in der urspruenglichen Form.
- +/-nP Anzeige von n-Bildschirmseiten (24 Zeilen je Seite) aus dem aktuellen Textpuffer. Gleichzeitig wird der CP' um die mit n vorwaerts (+) oder rueckwaerts (-) angezeigten Anzahl der Seiten bewegt. Mit n=0 wird die aktuelle Zeile beginnend von CP' mit den folgenden 23 Zeilen angezeigt.
- Q Abbruch des Editiervorganges ohne Aenderungen. Es wird keine ".BAK"-Datei erstellt, existiert jedoch vorher so eine Datei, so wird sie geloescht.
- R Liest vorhandene Zeilen aus der Datei "XXXXXXXXX.LIB" ein und fuegt sie direkt hinter den CP' ein.

R	Dateiname	Liest vorhandene Zeilen aus der mit den Dateinamen spezifizierten ".LIB"-Datei ein und fuegt sie direkt hinter den CP' ein.
nStextalt		Aufsuchen der mit "textalt" eingegebenen
^Ztextneu {^Z}		Zeichenfolge und Ersetzen durch die mit "textneu" eingegebenen Zeichenfolge. n gibt die Anzahl der Austauschoperationen an.
+/-n ²		Ist n positiv (+), werden die folgenden n-Zeilen (einschliesslich der aktuellen Zeile) angezeigt. Ist n negativ (-), so werden die n vorangehenden Zeilen ausschliesslich der aktuellen Zeile angezeigt.
+/-U		Uebersetzt alle folgenden eingegebenen Kleinbuchstaben nach der Eingabe von +U in Grossbuchstaben. Bei Eingabe von -U wird diese Umsetzung beendet.
V		Einschalten der Anzeige von Zeilennummern der Textdatei im Textpuffer.
OV		Zeigt die verfuegbare und die gesamte Grosse des Textpuffers in Bytes (dezimal) an. So bedeutet z.B. die Anzeige "27648/28832", dass 27648 Bytes im Textpuffer noch nicht belegt sind und der Textpuffer eine Gesamtgrosse von 28832 Bytes hat.
nW		Schreiben von n Zeilen beginnend von CP' in die entsprechende Uebergangsdatei der Kennung ".###".
nX		Kopiert die folgenden n Textzeilen in die Datei "XXXXXXXX.LIB". Mit n=0 wird mit diesem Befehl die Datei "XXXXXXXX.LIB" gelöscht.
nZ		Unterbricht laufende ED-Operationen um n Zeittakte (etwa n Sekunden).
+/-n		Bewirkt einen +/-nLT-Befehl.
n:		Positioniert den CP' an den Beginn der Zeile n.
n1::n2		Spezifiziert einen Bereich von Zeilennummern beginnend mit n1 und endend mit n2.

- . CP' ist der Zeichenzeiger (Character Pointer)
- . in allen ED-Befehlen, wenn mit n eine bestimmte Anzahl von Operationen angegeben wird, kann mit "#" die hoechstmoeegliche Zahl (65535) angegeben werden.
- . die geschweiften Klammern {} weisen auf 2 Wahlmoeglichkeiten hin. Der Befehl kann mit ^Z abgeschlossen und danach ein neuer Befehl angefuegt werden; anderenfalls ist ein Befehlsabschluss auch durch RETURN moeglich.

ED-Fehlermeldungen:

Fehlermeldungen werden vom Editor im allgemeinen in der Form:

"ERRDAK x AT c"

ausgegeben, wobei x ein Fehlersymbol und c den angeführten ED-Befehl angibt. Fuer x kann eine der folgenden Meldungen angegeben sein:

x	Bedeutung
?	Befehl wurde nicht erkannt
>	Der Puffer ist voll; Eingabe eines der folgenden Befehle D, K, S, W, E oder H oder die Zeichenfolge nach F, N oder S ist zu lang
#	Das Kommando kann nicht in der gewuenschten Haeufigkeit wiederholt werden
0	LIB-Datei kann mit dem R-Befehl nicht eroeffnet werden

Genauso kann u.U. vom OS/M-System ein Systemfehler in der Form:

"PERM ERR DISK d"

erkannt werden, wobei durch d das selektierte Laufwerk angezeigt wird. Der Fehler kann ignoriert werden, in dem eine beliebige Taste gedrueckt wird. Besser ist es jedoch in jedem Fall, das Vorhandensein der Originalkopie der Quelldatei zu ueberpruefen, um aus der Hintergrund-Datei die editierte Datei neu herzuleiten.

ANWENDUNG:

Das Standard-Dienstprogramm ED.COM dient der Erstellung und Modifizierung von Anwenderdateien.

BEISPIEL:

A:ED BEISPIEL.TXT

Loeschen einer oder mehrerer Dateien auf der Diskette

(ERA;Residentes Kommando)

FORMAT:

ERA {Dateibezeichnung
Dateigruppenbezeichnung}

ARGUMENTE:

Dateibezeichnung Bei Anwendung des ERA-Kommandos muss eine vollstaendige Dateibezeichnung angegeben werden. Eine Laufwerksbezeichnung kann wahlweise hinzugefuegt werden.

Dateigruppenbezeichnung Wird eine Dateigruppenbezeichnung beim ERA-Kommando angegeben, so koennen damit mehrere Dateien angesprochen und geloescht werden. Bei Angabe der Dateigruppenbezeichnung "*. *" werden alle Dateien des aktuellen Laufwerks geloescht.

BESCHREIBUNG:

Das ERA-Kommando loescht jede Datei, die mit der angegebenen Dateibezeichnung angesprochen wird. Davon ausgenommen sind Dateien mit R/O-Attributen. Wird die angesprochene Datei nicht gefunden, so wird die Ausschrift "NO FILE" ausgegeben.

Dateien auf Disketten in anderen Laufwerken koennen durch die zusaetzliche Angabe des entsprechenden Laufwerkskennzeichen angesprochen werden. Zum Beispiel ERA B:TEXT.* loescht alle Dateien mit dem Dateinamen TEXT und beliebigen Dateitypen auf Laufwerk B.

ANWENDUNG:

ERA ist ein residentes Kommando. Es wird zum Loeschen von Dateien eingesetzt und sollte mit aeusserster Vorsicht angewendet werden, da schnell Dateien irrtuemlich geloescht werden koennen.

BEISPIELE:

A:ERA BEISPIEL.TXT

(Das Kommando loescht die Datei BEISPIEL.TXT im Laufwerk A)

A:ERA B:SPRUNG.TXT

(Dieses Kommando loescht die Datei SPRUNG.TXT im Laufwerk B)

A:ERA *.*

(Dieses Kommando loescht alle Dateien im Laufwerk A)

A:ERA *.HEX

(Dieses Kommando loescht im Laufwerk A alle Dateien mit dem Dateityp ".HEX").

Ausgabe von Diskettenzugriffsfehlern seit dem Systemstart
(ERRORS.COM;Standard-Dienstprogramm)

FORMAT:

ERRORS

BESCHREIBUNG:

Über das Programm ERRORS.COM erfolgt die Ausgabe der Anzahl von Software- und Hardwarefehlern bei Diskettenzugriffen, die seit dem Systemstart aufgetreten sind.

ANWENDUNG:

Ein Start von ERRORS.COM initiiert die Ausgabe von Diskettenzugriffsfehlern in folgender Form:

THE FOLLOWING RECOVERABLE ERRORS HAVE OCCURED SINCE SYSTEM START

xxxx	SEEK ERRORS
yyyy	SECTOR ADDRESS ERROR
zzzz	DATA TRANSFER ERRORS

Wobei

xxxx - Anzahl der Such-Fehler
 yyyy - Anzahl der Sektor-Fehler
 zzzz - Anzahl der Daten-Fehler
 angibt.

BEISPIEL:

```

.
.
.
A:ERRORS

```

THE FOLLOWING RECOVERABLE ERRORS HAVE OCCURED SINCE SYSTEM START

```

0000 SEEK ERRORS
0002 SECTOR ADDRESS ERRORS
0000 DATA TRANSFER ERRORS

```

(Seit dem Systemstart sind 2 Sektor-Fehler aufgetreten).

Formatierung von Disketten

(FORMAT.COM;Standard-Dienstprogramm)

FORMAT:

FORMAT d:

ARGUMENTE:

d: Ueber d: wird das Laufwerkskennzeichen (A, ... D) angegeben.

BESCHREIBUNG:

Ueber die Laufwerksangabe d erfolgt eine Formatierung der Diskette im entsprechenden Laufwerk d. Das FORMAT-Programm stellt sich ueber das FDCONF-Word auf die Laufwerkstypen ein (s.a. Kommando SETFD, 4.1 und 5.4)

Erfolgt eine Formatierung von Disketten im Laufwerk 0 (A), so wird nach Beendigung des Formatiervorgangs folgende Meldung auf dem Bildschirm ausgegeben:

Ready for repeat?

Die Beantwortung mit 'Y' veranlasst eine erneute Formatierung der eingelegten Diskette auf dem Laufwerk 0. Das Einlegen einer Systemdiskette und die Beantwortung der Meldung mit 'N' initiiert einen Warmstart des Systems.

Erfolgt eine Formatierung von Disketten in den Laufwerken 1,2 oder 3 (B,C,D), so erfolgt nach Beendigung des Formatiervorgangs ein Warmstart.

Auftretende Eingabe- bzw. Formatierungsfehler generieren Fehlerausschriften auf der Konsole.

Im einzelnen haben die Fehlerausschriften folgende Bedeutung:

'MISSING OR INVALID OPERANDS'	- fehlerhaftes Kommando
'DISK VERY ERROR nn on track yyy'	- Disk-Fehler
'DISK FORMAT ERROR nn on track yyy'	- Formatierungsfehler
nn - UDOS Completion Code	C1 - ungueltiger Requestkode
	C2 - Laufwerk nicht bereit
	C3 - Schreibschutz Diskette
	C4 - Sektorfehler
	C5 - Spurfehler

yyy - Spurnummer (track)

ANWENDUNG:

A:FORMAT d:
READY?

d=A,B,C oder D

Bei der Beantwortung von 'READY?' mit 'Y' beginnt die Formatierung der durch d: angegebenen Diskette.

BEISPIEL:

A:FORMAT B:
READY?Y
A:

(Formatierung der Diskette im Laufwerk B)

Transformation einer COM-Datei in eine HEX-Datei

(GENHEX.COM; Transientes Programm)

FORMAT:

GENHEX Programmname.COM offset

ARGUMENTE:

Programm- Der Name des zu transformierenden Programms
name (von Dateityp ".COM"). Dem Programmnamen
kann ein Laufwerkskennzeichen vorangesetzt
werden.

offset Gewünschter Offset der zu generierenden
".HEX"-Datei (hexadezimal).

BESCHREIBUNG:

Dieses Kommando generiert aus einer Datei von Typ ".COM"
eine Datei vom Typ ".HEX" und legt hier den durch das
Argument "offset" angegebenen Offsetbereich an.

ANWENDUNG:

Diese Kommando wird allgemein zur Generierung seitenweise
verschiebbarer Programme verwendet.

BEISPIEL:

.
.
.
A:GENHEX ACTION.COM 100
.
.
.

Umwandeln einer ".HEX"-Datei in einen ausfuehrbaren
Befehl als ".COM"-Datei

(LOAD.COM;Standard-Dienstprogramm)

FORMAT:

LOAD Dateiname

ARGUMENT:

Dateiname Name einer Datei vom Dateityp ".HEX". Der
Dateityp braucht nicht ausdruecklich angege-
ben zu werden.

BESCHREIBUNG:

Das LOAD-Kommando laedt ein in "HEX"-Format geschriebenes
Programm und formt es in eine entsprechende Befehlsdatei
".COM-Datei" um, die dann durch Eingabe des Namens gestar-
tet werden kann.

ANWENDUNG:

Eine mit dem LOAD-Kommando generierte Datei kann dann durch
den Aufruf des Dateinamens direkt gestartet werden.

BEISPIEL:

A:LOAD BEISPIEL
("BEISPIEL.HEX" wird in "BEISPIEL.COM" umgeformt)

.
.
.

A:BEISPIEL
(Das Programm BEISPIEL.COM wird ausgefuehrt)

Peripheral Interchange Program
Ausfuehrung von Kopieroperationen

(PIP.COM; Standard-Dienstprogramm)

FORMAT:

1. PIP {d:Zielfdatei } =d:Quelldatei[p]
d: }
2. PIP
*d:Zielfdatei=d:Quelldatei[p]
* ...
* ...
* ...
* RETURN
3. PIP {dev: } {dev } [p],
{d:Dateibez.} {d:Dateibez.}
{dev: }
{d:Dateibez.}, ... [p]
4. PIP d:=d:Dateigruppenbezeichnung[p]

ARGUMENTE:

- d:Zielfdatei
d: Der Benutzer muss hier im Format 1 oder 2 angeben, ob die neue Datei eine neue Bezeichnung erhalten oder ob die neue Datei (Kopie der Quelldatei) die gleiche Bezeichnung wie die alte Datei (Quelldatei) eines anderen Laufwerkes (d:) erhalten soll.
- d:Quelldatei Die Dateibezeichnung muss in beiden Formaten 1 und 2 angegeben werden (wahlweise mit Laufwerkskennzeichen)
- [p]
c Vom Nutzer kann in allen Formaten wahlweise ein PIP-Parameter gesetzt werden, der dann auf die davor angegebene Datei wirkt.
- dev:
d:Dateibez. In Format 3 muss vom Nutzer entweder fuer "dev:" die Kurzbezeichnung der anzusprechende Gerateeinheit (z.B. IST) oder eine Dateibezeichnung (wahlweise mit Laufwerksbezeichnung) angegeben werden. Um den Inhalt einer Datei zu einer Gerateeinheit zu senden oder empfangene Daten in eine Datei zu schreiben, kann diese Form benutzt werden.
- d:=d:Dateigruppenbez. Das Ziellaufwerk als auch die Dateigruppenbezeichnung muss hierbei angegeben werden. Mit Format 4 koennen mehrere Dateien (Datei-

gruppenbezeichnung) von einer Diskette zu einer anderen kopiert werden.

BESCREIPUNG:

Format 1: Wenn ein d: angegeben ist, erhaelt die neue Datei (Zieldatei) die gleiche Bezeichnung wie die alte Datei (Quelldatei). Das auf der linken Seite angegebene Laufwerkskennzeichen muss dann von dem auf der rechten bzw. vom akt. Laufwerkskennzeichen unterschiedlich sein. Wenn d: weggelassen wird, so wird das akt. Laufwerk genommen. Wenn fuer die Zieldatei eine Bezeichnung angegeben wurde, dann wird unter dieser Bezeichnung eine Datei mit dem Inhalt der Quelldatei kopiert.

Format 2: In diesem Format koennen mehrere PIP-Operationen abgewickelt werden. Das PIP-Programm wartet nach Ausfuehrung einer PIP-Operation auf erneute Eingabe (wird durch "*" angezeigt). Durch Betaetigung der RETURN-Taste wird diese Betriebsart abgebrochen. PIP-Anweisungen folgen den gleichen Regeln wie unter Format 1.

Format 3: In diesem Format koennen mit dev: (device name) die verfügbaren Geraeteinheiten wie Dateien angesprochen werden. Die linke Seite des Ausdrucks ist immer die Zieldatei (-datei) und die rechte Seite immer die Quelleinheit (-datei). Vom Benutzer koennen auch mehrere Quelldateien (durch Kommandos getrennt) zu einer Zieldatei zusammengefasst werden.

Format 4: Bei der Angabe einer Dateigruppenbezeichnung und einem wahlweise angebbaren Laufwerkskennzeichen koennen mit einem Befehl mehrere Dateien kopiert werden. Das Laufwerkskennzeichen auf der linken Seite muss unbedingt angegeben werden. Nach dem Kopieren haben die Zieldateien die gleiche Bezeichnung wie die entsprechenden Quelldateien auf der Quelldiskette.

PIP-Fehlermeldungen:

- DISK READ ERROR	Lesefehler
- DISK WRITE ERROR	Schreibfehler
- BAD PARAMETER	undefinierter Parameter
- INVALID FORMAT	fehlerhaftes PIP-Kommando
- NO DIRECTORY SPACE	keine Verzeichniseintragung moeglich
- NO FILE	keine Datei vorhanden
- START NOT FOUND	Zeichenketten fuer begrenztes Kopieren wurde nicht gefunden
- QUIT NOT FOUND	
- UNRECOGNIZED DESTINATION	unerlaubte Zieldatei
- CANNOT READ	unerlaubte oder fehlende Quelldatei

- | | |
|----------------------|--|
| - CANNOT WRITE | unerlaubte oder fehlende Ziel-datei |
| - INVALID PIP FORMAT | fehlendes Gleichheitszeichen oder unerlaubte Leerzeichen |
| - INVALID SEPARATOR | unerlaubte Trennzeichen in der Parameterliste |

ANWENDUNG:

Das Programm PIP.COM kann von jedem Laufwerk durch zusätzliche Angaben des Laufwerkskennzeichens gestartet werden (z.B. B:PIP). Nach dem Start des PIP-Programms können die gewünschten Kopieroperationen durchgeführt werden. Durch Drücken der RETURN-Taste kann das PIP-Programm abgebrochen werden.

BEISPIELE:

A:PIP

*DATEI2=TEST2

(Nach Eingabe diese PIP-Ausdrucks wird von der Datei TEST2 eine Datei DATEI2 kopiert).

*LST:=DATEI2

(Dieser Ausdruck sendet eine Kopie der DATEI2 zur LST-Einheit).

*B:=PROG.ASM

(Generieren einer Kopie der Datei PROG.ASM auf Laufwerk B derselben Bezeichnung)

Umbenennen einer Datei

(REN;Residentes Kommando)

FORMAT:

REN [d:]neue Dateibezeichnung=[d:]alte Dateibezeichnung

ARGUMENTE:

neue Dateibe-	Dateiname und Dateityp sind erforderlich.
zeichnung	Ein Laufwerkskennzeichen ist zulaessig, muss
und alte	aber bei beiden Dateibezeichnungen identisch
Dateibezeich-	oder nur bei der "neuen" angegeben sein.
nung	

BESCHREIBUNG:

Das REN-Kommando benennt die Datei mit der alten Dateibezeichnung in eine neue Dateibezeichnung um. Es muss hierbei die vollstaendige Dateibezeichnung einschliesslich Dateityp angegeben werden.

ANWENDUNG:

Umbenennen von Dateien.

Bei dem Kommando sind zwei Fehlermeldungen moeglich:

- . NO FILE Eine Date mit der alten Bezeichnung existiert auf dem adressierten logischen Laufwerk nicht
- . FILE EXISTS Es besteht schon eine Datei mit der neuen Bezeichnung.

Nach diesen Fehlern wird die Ausfuehrung des Kommandos abgebrochen und OS/M geht in den Systemgrundzustand.

BEISPIEL:

A:REN TEXTNEU.HEX=TEXTALT.HEX

(Dieses Kommando benennt die Datei TEXTALT.HEX in TEXTNEU.HEX um).

Sichern eines Speicherinhaltes als DISK-Datei

(SAVE;Residenter Befehl)

FORMAT:

SAVE p Datei-Bezeichnung

ARGUMENTE:

p Anzahl fuer die Datei benoetigten Speicherseiten
 Datei-Bezeichnung Gewuenschte vollstaendige Datei-Bezeichnung der zu erstellenden Datei, einschliesslich Dateityp

BESCHREIBUNG:

Das SAVE-Kommando sichert den Inhalt des EPA-Bereiches beginnend von 100H bis zu einer Grosse von p-Speicherseite zu 256 Byte-Segmenten in einer Datei mit der angegebenen Datei-Bezeichnung.

ANWENDUNG:

Zur Berechnung der benoetigten Anzahl fuer p muss zunaechst mit dem DDT das gewuenschte Programm geladen und der Wert der Variablen "NEXT" ermittelt werden. Die Adresse NEXT ist um 1 grosser als die letzte Adresse des Programms. Zur Berechnung des Wertes fuer p kann der folgende Algorithmus verwendet werden:

Wenn die letzten 2 Ziffern des NEXT-Wertes 00 sind, dann muss hiervon 1H abgezogen werden (z.B. 1C00H-1=1BF9H). Wenn die letzten zwei Ziffern ungleich 00 sind, kann diese Zahl direkt verwendet werden. Von dem so erhaltenen Wert koennen dann die ersten beiden Ziffern (die hoehwertigen Ziffern) zur dezimalen Umrechnung von p direkt herangezogen werden. Im obigen Beispiel ergibt sich fuer

$$p=1BH=16+11=27$$

BEISPIEL:

```
A:DDT BEISPIEL.COM
NEXT PC
1D00 00
-GO
```

Der Wert unter NEXT ist 1D00H. Die Subtraktion von 1H ergibt 1CFFH. Die Hexadezimalzahl 1CH ergibt den Dezimalwert 28.

```
A:SAVE 28 KOPIE.COM
```

Ausgabe und Aenderung der Floppy-Disk-Konfiguration

(SDTWD.COM;Standard-Dienstprogramm)

FORMAT:

1. SDTWD
2. SDTWD [abcd [Sd]]

ARGUMENTE:

abcd Sd sind wahlweise angebbare Parameter. Werden diese Parameter weggelassen, so erfolgt eine Ausgabe der Floppy-Disk-Konfiguration des PC8000 auf dem Bildschirm.

Die Parameter "abcd" entsprechen den logischen Laufwerken A,B,C,D (a=logisches Laufwerk A, b=logisches Laufwerk B, ...). Bei einer Aenderung der Floppy-Disk-Konfiguration muessen fuer die Parameterfolge "abcd" entsprechende Konfigurationsgroessen angegeben werden. Die Konfigurationsgroessen sind wie folgt vereinbart:

- 0 - kein Laufwerk
- 1 - 8"-Laufwerk
Format: SD-SS 26x128x77
Operation:Lesen und Schreiben
- 2 - 5 1/4"-Laufwerk (K5600.10)
Format: DD-SS 16x256x40
Operation:Lesen und Schreiben
- 3 - 5 1/4"-Laufwerk (K5600.20)
Format: DD-SS 16x256x40
Operation:Lesen
- 4 - 5 1/4"-Laufwerk (K5600.20)
Format: DD-SS 16x256x80
Operation:Lesen und Schreiben
- 5 - 5 1/4"-Laufwerk
Format: DD-DS 32x256x80
Operation:Lesen und Schreiben

DD=double density
SD=single density
DS=double sided
SS=single sided

Der Parameter "Sd" kennzeichnet eine Auswahlmoeglichkeit hinsichtlich der Aenderung der Floppy-Disk-Konfiguration. Wird der Parameter weggelassen, so erfolgt eine temporaeere Aenderung (RAM) der Konfiguration entsprechend der angegebenen Parameterfolge "abcd".

Wird "Sd" angegeben, so ist die Aenderung der Konfiguration permanent, das heisst die

Aenderung wird in die Systemspuren der entsprechenden Systemdiskette eingetragen, die sich in dem mit d=0,1,2,3 selektierten Laufwerk befindet. Die Angabe "SO" bewirkt eine Aenderung der Floppy-Disk-Konfiguration sowohl im RAM (temporaer) als auch auf der Systemdiskette (permanent) im Laufwerk 0 (A).

BESCHREIBUNG:

Dieses Kommando ist ausschliesslich fuer den Systembetreuer vorgesehen. Es dient zur Ausgabe und Aenderung der Floppy-Disk-Konfiguration des P8000. Die Floppy-Disk-Konfiguration ist auf der Systemdiskette vermerkt. Dieses Kommando ist bei Anschluss eines Beistellers bzw. bei Anschluss eines Beistellers mit anderen Laufwerkstypen fuer jede Systemdiskette auszufuehren.

ANWENDUNG:

Anzeige bzw. softwaeremaessige Einbindung der Floppy-Disk Konfiguration.

BEISPIEL:

```
.  
.  
.  
A:SETFD 5511 SO
```

(Der Anschluss eines Beistellers mit 8"-Standardlaufwerken wird auf der Systemdiskette im Laufwerk 0 (permanent) und im RAM (temporaer) initiiert).

Anzeige und Aenderung der Druckparameter

(SETLP.COM; Standard-Dienstprogramm)

FORMAT:

1. SETLP
2. SETLP [lines cols indent [Sd]] [A/N]

ARGUMENTE:

lines cols indent Sd A/N Sind wahlweise angebbare Parameter. Ueber "lines" kann die Anzahl der Zeilen pro Seite geaendert werden. Mit dem Argument "cols" kann die Anzahl der Zeichen pro Zeile (rechter Rand) angegeben werden. Mit "indent" wird die Anzahl der Leerzeichen fuer den rechten Rand festgelegt. Der Parameter "Sd" kennzeichnet eine Auswahlmoeglichkeit hinsichtlich der Aenderung der Druckparameter. Wird der Parameter weggelassen, so erfolgt eine temporaere Aenderung (RAM) der Druckparameter entsprechend der angegebenen Parameterfolge "lines cols indent". Wird "Sd" angegeben, dann ist die Aenderung der Druckparameter permanent, das heisst die Aenderung wird in die Systemspuren der entsprechenden Diskette eingetragen, die sich in dem mit d=0,1,2,3 selektierten Laufwerk befindet. Die Angabe "SO" bewirkt eine Aenderung der Druckparameter sowohl im RAM (temporaer) als auch auf der Systemdiskette (permanent) im Laufwerk 0 (logisches Laufwerk A). Ueber den Parameter "A/N" kann der Zeilenabstand generiert werden. "A" erzeugt einen alternativen Zeilenabstand (1 1/2-zeilig) und "N" einen normalen Zeilenabstand (1-zeilig). Die Parameterangabe "A/N" ist nur wirksam bei Anwendung des Textverarbeitungssystem TP. Werden die Parameter weggelassen, so erfolgt eine Anzeige der entsprechenden aktuellen Druckparameter "lines", "cols" und "indent" auf dem Bildschirm.

BESCHREIBUNG:

Das Kommando SETLP dient der Anzeige und Aenderung (temporaer und permanent) der Druckparameter.

BEISPIEL:

·
·
·
A:SETLP 03 80 4 80

(Anzahl der Zeilen pro Zeile=63; Anzahl der Zeichen pro Zeile=80; Anzahl der Leerzeichen=4. Die Aenderung erfolgt im RAM und auf der Systemdiskette im Laufwerk 0).

Ausfuehren einer Befehlsfolge

(SUBMIT.CON;Standard-Dienstprogramm)

FORMAT:

SUBMIT Dateiname [v1 v2 v3 . . . v9]

ARGUMENTE:

Dateiname Hier muss der Name einer Textdatei angegeben werden, die mit einzelnen Befehlszeilen gefuellt ist und die den Dateityp ".SUB" aufweist. Der Dateityp ".SUB" braucht jedoch nicht mit angegeben zu werden.

v1 v2 v3...v9 Wahlweise angebbare Werte, die anstelle der in der SUBMIT-Datei aufgefuehrten Variablen uebernommen werden sollen. Die Variablen muessen die Form #1,#2,#3 usw. aufweisen, die fuer v1,v2,v3 usw. angegebenen Werte werden hierin uebernommen.

BESCHREIBUNG:

Die SUBMIT-Routine generiert automatisch eine temporaere Datei "###.SUB", in der die abzuarbeitenden Befehle stehen. Bei der Erstellung der Uebergangsdatei werden vom SUBMIT-Programm die Variablen #1,#2,#3 usw. von den mit v1,v2,v3 usw. angegebenen Werten ersetzt. Erst wenn alle Befehle abgearbeitet sind bzw. ein Fehler erkannt wurde, wird die Datei geloescht. Falls die Datei aus irgendeinem Grunde noch existiert, so fuehrt das System bei einem Systemstart (Warmstart) sofort wieder diese Routine aus. SUBMIT-Dateien werden nur dann vom System abgearbeitet, wenn sie ueber laufwerk A zugreifbar sind.

SUBMIT-Fehlermeldungen:

- No 'SUB' File Present: die Datei "name.SUB" existiert nicht
- Parameter Error Parameter-Fehler des Kommandos in der Datei "name.SUB"
- Command Too Long die Kommandozeile ist zu lang (max. Zeilenlaenge 126 Zeichen)
- Command Puffer Overflow: die neue Datei "###.SUB" ist zu lang (max. Laenge 1023 Zeichen)
- Directory Full kein freier Bereich fuer die Datei "###.SUB"
- Cannot close, Read/Only die Datei "###.SUB" kann nicht beendet werden, Diskette nur fuer Lesen zugelassen

- Disk Write Error Hardwarefehler Laufwerk
- Invalid Control Character fehlerhaftes Steuerzeichen in
der Datei "###.SUB"

ANWENDUNG:

Eine "SUBMIT"-Datei kann mit jedem Editor (z.B. ED) generiert werden. Die Befehlszeilen koennen noch zusaetzlich Variablen aufweisen (#1,#2,#3 usw.).

BEISPIEL:

Die Befehlsdatei TEST.SUB enthaelt folgende Textzeilen:

```
DIR #1.*
PIP #2:=#1.BAK
ERA #1.BAK
```

Die Ausfuehrung der einzelnen Befehlszeilen kann mit folgenden SUBMIT-Befehl gestartet werden:

```
A:SUBMIT TEST PROG B
```

Die Befehlszeilen werden in der Datei ###.SUB mit den Werten der Variablen wie folgt transformiert:

```
DIR PROG.*
PIP B:=PROG.BAK
ERA PROG.BAK
```

Nach der Transformierung werden vom System automatisch die einzelnen Befehlszeilen von ###.SUB abgearbeitet.

Anzeige wichtiger Statusinformationen und zugeordneter Gerateeinheiten

(STAT.COM;Standard-Dienstprogramm)

FORMATE:

1. STAT d:=R/O
2. STAT d: {Dateibezeichnung}
{Dateigruppenbezeichnung}
3. STAT d: {Dateibezeichnung} {#S}
{Dateigruppenbezeichnung} {#R/O}
{#R/W}
{#SYS}
{#DIR}
4. STAT d: {DSK:}
{USR:}

ARGUMENTE:

d:=R/O mit d:=R/O wird durch R/O das mit d: angegebene Laufwerk so verwaltet, dass von ihm nur Daten gelesen werden koennen. Wird nur d: angegeben, so wird dessen Status (z.B. R/O oder R/W) angezeigt.

{Dateibez.} Mit diesem Format kann der Status einer
d: Datei oder ganzer Dateigruppen angezeigt
{Dateigr.} werden (Anzahl der Records, Bytes, Extents usw.).

#S Das Argument #S erlaubt die gleiche Abfrage
#R/O von Zusatzinformationen wie mit Format 4.
#SYS Bei Angabe der anderen Argumente werden den
#R/W angesprochenen Dateien die entsprechenden
#DIR Attribute zugeordnet (z.B. R/O (read only) schuetzt vor Ueberschreiben).

DSK: Mit DSK: koennen Statusinformationen des
d: aktuellen bzw. des mit d: angegebenen Laufwerks angezeigt werden. Zur Anzeige des
USR: aktuellen und aktiven Benutzerbereiches ist die Eingabe von USR: erforderlich.

BESCHREIBUNG:

Der STAT-Befehl erlaubt die Abfrage von Statusinformationen von Dateien und Disketten. Mit dem STAT-Kommando koennen Laufwerke zu Nur-Lese-Dateien umgeschaltet und Informatio-

nen ueber aktuelle und aktive Benutzerbereiche abgefragt werden.

STAT-Fehlermeldungen:

- | | |
|---------------------------|--|
| - File not found | Falsche oder unvollstaendige Angabe der Datei, Datei nicht enthalten. Falsche Angabe der Parameter (z.B. DSK anstelle DSK:). |
| - Invalid File Indicator | Falsche Parameterangabe (z.B. Leerzeichen zwischen #R/O). |
| - Invalid Disc Assignment | Falsche Attributzuweisung fuer Diskette (z.B. STAT A:=SYS nicht moeglich). |
| - Bad Delimiter | Falsches Trennzeichen. |
| - Invalid Assignment | Fehlerhafte Kanalzuordnung (z.B. STAT RDR:=LPT:) |

ANWENDUNG:

Das Programm STAT.COM kann zur Anzeige von Statusinformationen mit einem der angegebenen Formate gestartet werden.

BEISPIELE:

```
A:STAT PIP.COM #S
```

```
Size Recs Bytes      Ext  Acc
55   55   12K         1   R/O  A:PIP.COM
```

(Die Datei PIP.COM hat 55 Records mit insgesamt 12K und ist als Datei mit einem R/O-Attribut auf der Diskette in Laufwerk A abgelegt).

```
A:STAT BEISPIEL.COM #R/O
```

```
A:STAT B:
```

```
BYTES REMAINING ON B:192K
```

```
B:R/O
```

(Laufwerk B hat ein R/O-Attribut und verfuegt noch ueber 192K freien Speicherplatz).

Generieren einer OS/M-Kopie
Laden des Systems in den Arbeitsspeicher
und (oder) Erstellen einer Systemkopie

(SYSGEN.COM, Standard-Dienstprogramm)

FORMAT:

SYSGEN

BESCHREIBUNG:

Das SYSGEN-Programm initialisiert eine Systemdiskette (beschreibt die Systemspuren einer Diskette nach dem Formatieren mit dem System, die Anzahl der Systemspuren ist abhängig vom gewählten Diskettenformat). Darüber hinaus wird das System in den Arbeitsspeicher geladen.

ANWENDUNG:

Start des Programms SYSGEN.COM

A:SYSGEN
SYSGEN VERSION 2.2
SOURCE DRIVE NAME (OR RETURN TO SKIP)A

(Die Frage nach dem Laufwerk der Systemdiskette (SOURCE DRIVE NAME) muss mit der entsprechenden Laufwerksbezeichnung beantwortet werden, wenn das System sich noch nicht im Arbeitsspeicher befindet. Ist das gewünschte System bereits im Arbeitsspeicher, muss die Frage durch Druecken der RETURN-Taste quittiert werden).

SOURCE ON A, THEN TYPE RETURN
FUNCTION COMPLETE

(Das System wurde vollstaendig gelesen und befindet sich nun im Arbeitsspeicher)

DESTINATION DRIVE NAME (OR RETURN TO REBOOT)B

(Die Frage nach dem Laufwerk fuer die zu generierende Systemdiskette muss mit dem Laufwerkssymbol beantwortet werden, in dem die neu zu erstellende Systemdiskette liegt.

DESTINATION ON B, THEN TYPE RETURN
FUNCTION COMPLETE

(Das System wurde auf die neue Systemdiskette geschrieben und ist nun von dort aus aufrufbar)

DESTINATION DRIVE NAME (OR RETURN TO REBOOT)
A:

(Die SYSGEN-Routine wird durch Druecken der RETURN-Taste abgebrochen)

Umschalten des aktuellen Benutzerbereichs(USER; Residentes Kommando)

FORMAT:

USER n

ARGUMENTE:

n Dieses Argument steht fuer einen gewählten Benutzerbereich (0 ... 15).

BESCHREIBUNG:

Der aktuelle Benutzerbereich wird auf Bereich n umgeschaltet.

Das bedeutet z.B. fuer die folgenden Kommandos DIR,REN,SAVE:

- . DIR zeigt nur die Dateien des USER n an
- . REN kann nur die Dateien USER n umbenennen
- . SAVE speichert nur Dateien des USER n ab

ANWENDUNG:

Aufteilung des Verzeichnisses der Disketten auf verschiedene Benutzer.

BEISPIEL:

A:USER 3
A:

Erweiterte SUBMIT-Funktion zur Einfuegung
gewuenschter Eingabedaten bzw. -zeilen
in eine ausgefuehrte SUBMIT-Datei

(XSUB.COM;Standard-Dienstprogramm)

FORMAT:

XSUB

BESCHREIBUNG:

Wenn XSUB vor einer ".SUB"-Datei angegeben wird, wird XSUB direkt unter den CCP-Bereich geladen, um so die in einer ".SUB"-Datei stehenden Befehlsdaten abzuarbeiten und die zusaetzlich ueber die Konsole abgefragten Eingaben zu verarbeiten.

ANWENDUNG:

Zur Anwendung dieses Befehls muss XSUB als erste Befehlszeile in der ".SUB"-Datei aufgefuehrt sein. XSUB bleibt bis zum naechsten Kaltstart aktiv und die ".SUB"-Datei wird bis zum Ende weiterbearbeitet.

BEISPIEL:

```
Datei NEU.SUB: XSUB  
DDT  
I=1.HEX  
R  
GO  
SAVE 1 =2.COM
```

A:SUBMIT NEU DIES DAS

"DIES" ersetzt =1 und "DAS" ersetzt =2. Das XSUB-Programm startet mit DDT und fuehrt die DDT-Befehlszeilen "IDIES.HEX", "R" und "GO" aus. Danach veranlasst das XSUB-Programm mit "SAVE 1 DAS.COM" zur Abwicklung eines SAVE-Befehls. XSUB bleibt bis zum naechsten Kaltstart aktiv.

3.4. OS/M-Applikationssoftwarepakete

Unter dem Betriebssystem OS/M stehen z.B. die Applikationssoftwarepakete TP und REDABAS zur Verfügung.

Durch seine grosse Vielseitigkeit gehoert TP zu den am meisten verbreiteten Textverarbeitungssystemen.

Spezielle Merkmale von TP sind:

- schnelle Installation sowie leicht erlernbare Aufbereitung der Daten machen die Anwendung von TP so effizient.
- bildschirmorientierte Formatierung
- falls der Text breiter ist als der Bildschirm, verschiebt TP den Bildschirminhalt auch horizontal
- TP bietet die Anzeige von Hilfsinformationen auf dem Bildschirm an
- hilfreiche Trennmöglichkeiten der Woerter fuer den rechten Randausgleich erleichtern das saubere Formatieren der Arbeiten
- Einfuegung von Zusatzinformationen bei der Aufbereitung des Textes (z.B. fuer den spaeteren Druck)
- Sondereinrichtungen zum Suchen und Aendern nach spezifizierten Zeichenfolgen
- einfache Verschiebung von Textblocken erlauben es, bestimmte Textabschnitte an eine beliebige Stelle zu schieben oder zu kopieren
- es koennen an beliebiger Stelle Buchstaben, Woerter oder ganze Wortgruppen eingefuegt, geloescht, verschoben oder kopiert werden
- schnelle Positionierung des Cursors durch bis zu zehn frei wahlbare Positionsmarkierungen
- Hilfsfunktionen zum Trennen von Woertern am Zeilenende, Blocksatz mit Randausgleich, Seitenumbruch, Hoch- und Tiefstellen von Texten sind moeglich
- verschiedene Steuerzeichen fuer den Druckvorgang, wie z.B. Fettdruck, variable Zeilengroesse und Zeilenabstaende

REDABAS ist ein relationales Datenbanksystem. Es zeichnet sich durch bequeme und sichere Bedienung sowie durch grosse Flexibilitaet in der Anwendung aus. Die wesentlichen Merkmale von REDABAS sind:

- bietet alle Moeglichkeiten der Datei- und Datenhandhabung, wie Erfassen, Aendern, Einfuegen, Mischen und Suchen; damit arbeiten alle Anwendungen auf einem gemeinsamen Datenbestand, der nur einmal erfasst werden muss
- sequentieller oder wahlfreier Zugriff nach frei wahlbaren/kombinierbaren Kriterien; beliebige Indizierung ueber Schluessel
- eingebauter Maskengenerator vereinfacht die Entwicklung von Erfassungsmasken; Report-Generator zum Schreiben von Berichten; Schema und Inhalt wird im Dialog definiert
- vorprogrammierte Anwendungen ueber Befehlsdateien bis hin zur Menuesteuerung
- ermoeoglicht Aufbau kompletter Anwendungen, wie Finanzbuchhaltung, Lagerverwaltung etc.

Beschreibung, Anwendung und Installationsvorschriften von TP und REDABAS befinden sich auf der entsprechenden Diskette.

4. Disketten und Dateien

Das Floppy-Disk-orientierte Betriebssystem OS/M benutzt die schnelle direkte Zugriffsmöglichkeit zu Disketten-Dateien, verwaltet diese weitgehend automatisch und befreit den Nutzer von Verwaltungsarbeit.

4.1. Wahl des aktuellen Diskettenlaufwerks

Zum Starten des Systems durch Laden des OS/M-Betriebssystems benötigt man eine Systemdiskette. Die Systemdiskette ist eine spezielle Diskette, die das Betriebssystem OS/M enthält. Der Start von OS/M erfolgt grundsätzlich über das Laufwerk A. Dazu wird die Systemdiskette in das Laufwerk A eingeschoben und durch Einschalten des Systems oder durch Betätigung der Taste RETURN das System gestartet.

Mit der Systembereitschaftsmeldung (System prompt)

```
P8000 OS/M 2.2 (C) ZPT/KEAW
```

```
A:
```

meldet sich das System.

Ein "prompt" wird von OS/M immer dann gesendet, wenn es fuer die naechste Eingabe von Kommandos bereit ist. OS/M verwendet das Startsymbol "A:". Der Buchstabe A steht fuer das "logische Laufwerk A" (bestehend aus Laufwerk A und der darin befindlichen Diskette). Das Systembereitschaftszeichen teilt jeweils mit, welches Diskettenlaufwerk das aktuelle Laufwerk ist. Alle Programm- und Dateinamen in nachfolgenden Kommandos beziehen sich ausschliesslich auf das Laufwerk A. Unter der Voraussetzung, dass an das System 2 Laufwerke angeschlossen sind, kann mit folgendem Befehl auf das Laufwerk B umgeschaltet werden:

```
A:B:
```

Wenn dort eine geeignete Diskette vorhanden ist, erscheint

```
B:
```

womit Laufwerk B als aktuelles Laufwerk gemeldet wird. Alle Eingaben koennen in Gross- und Kleinbuchstaben erfolgen. Die Anzeige

```
"d:"
```

(d allgemein fuer Laufwerk A ... D) wird als Systemgrundzustand bezeichnet.

Im OS/M sind 2 ... 4 logische Laufwerke (A ... D) vorgesehen, die durch 2 ... 4 physische Laufwerke realisiert sind. Das erste physische Laufwerk wird zum logischen Laufwerk A, das zweite zum logischen Laufwerk B usw.

Die Floppy-Disk-Laufwerke 0 und 1 sind im P8000 eingebaute Minilaufwerke mit doppelter Aufzeichnungsdichte und 80

Spuren. Optional kann ein Beisteller mit zwei Standardlaufwerken, mit zwei 80-Spur- oder zwei 40-Spur-Minilaufwerken betrieben werden.

Alle einmal "aktiv" geschalteten Laufwerke werden im System als "on line" vermerkt.

4.2. Systemdiskette

Die Systemdiskette ist eine spezielle Diskette, die das Betriebssystem OS/M auf den ersten Spuren enthaelt. Mit einer derartigen "Systemdiskette" koennen Kalt- und Warmstart durchgefuehrt werden, bei denen das gesamte oder ein Teil des Betriebssystems in den Speicher geladen werden.

Kaltstart wird durch Einschalten oder durch die RESET-Funktion durchgefuehrt und bewirkt einen voelligen Neubeginn ueber die Systembereitschaftsmeldung des Systems (s.a. 3.1). Mit dem Ausdruck "Kaltstart" wird die Grundoperation bezeichnet, mit der das OS/M von der Diskette geladen wird und sich danach selbst startet.

Ein "Warmstart" kann durch Eingabe von <CTRL> C oder von Programmen veranlasst werden. Dadurch wird das System in den Systemgrundzustand gebracht, das aktuelle Laufwerk wird beibehalten. Ein Warmstart kann dazu benutzt werden, um eine "read-only"-Diskette (vom System geschuetzt) in "read/write" zu bringen. Das heisst, wird eine neue Diskette in ein Laufwerk eingeschoben und werden auf dieser neuen Diskette Schreibvorgaenge durchgefuehrt, so muss vorher unbedingt ein Warmstart durchgefuehrt werden. Wenn allerdings von der neuen Diskette nur gelesen werden soll, ist ein Warmstart nicht erforderlich.

4.3. Diskettenformat

Unter OS/M sind folgende Laufwerke vorhanden:

A:	(Drive 0)	5 1/4"	-	Systemlaufwerk
B:	(Drive 1)	5 1/4"	-	Anwenderlaufwerk

Optional:

C:	(Drive 2)	8"	-	Anwenderlaufwerk
D:	(Drive 3)	8"	-	Anwenderlaufwerk
C:	(Drive 2)	5 1/4"	-	Anwenderlaufwerk
D:	(Drive 3)	5 1/4"	-	Anwenderlaufwerk

Bevor eine Diskette verarbeitbar ist, muss sie mit dem Standard-Dienstprogramm FORMAT formatiert werden. Fuer OS/M sind folgende Formate bezueglich Kapazitaet, Arbeitsgeschwindigkeit und Kompatibilitaet realisiert:

	5 1/4"-Disketten			8"-Disketten
Aufzeichnungsverfahren:		KPM		PM
Spuren:	40	80	80	77
reservierte Systemspuren:	3	3	2	--
Sektoren/Spur:	16	16	32	26
Byte/Sektor:	256	256	256	128
Datenkapazitaet: (in KByte)	148	308	624	240

4.4. Datei-Konzept

Das Speichern von Informationen (Daten/Programme) auf der Diskette erfolgt generell als Dateien. Um diese Informationen verwenden zu koennen, muss auf der entsprechenden Diskette die Datei ueber eine Bezeichnung (filename) aufrufbar sein. Alle Dateien einer Diskette sind in einem Verzeichnis (directory) registriert. Jede Datei belegt in diesem Verzeichnis mindestens eine 32 Byte grosse Eintragung.

OS/M arbeitet bei der Aufteilung des Diskettenraumes mindestens mit jeweils 8 Saetzen (records). Auf Standard-Disketten (8") besteht ein Satz aus 128 Bytes. Aus diesem Grunde ist der kleinste Speicherplatz, der auf der Diskette belegt werden kann, aus 1 KByte. Minidisketten (5 1/4") haben Saetze zu 256 Bytes. In diesem Fall bestehen die kleinsten Diskettenbereiche bei OS/M aus 2 KByte-Blocken.

Um eine dynamische Disketten-Speicherplatz-Verwaltung zu realisieren, sind die Blocke der gesamten Diskette (mit Ausnahme der Systemspuren) durchnummeriert. Im Verzeichnis der Diskette sind neben der Eintragung des Dateinamens auch die Nummer der Blocke enthalten, die diese Datei auf der Diskette belegt.

Wird z.B. eine Datei geloescht, stehen deren Blocke sofort fuer ander Dateien zur Verfuegung. Wird eine Datei ver-groessert, so wird ein freier Block in der Umgebung des letzten Blockes auf der Diskette gesucht und dessen Nummer in der Eintragung der Datei gespeichert.

Die minimale Groesse einer Datei ist damit bei Standard-disketten 1 KByte und bei Minidisketten 2 KByte. Die max. Groesse einer Datei entspricht der Speicherkapazitaet einer Diskette.

4.5. Dateizeichnungen

Die Dateizeichnung (filename) dient der Identifikation einer Datei auf Diskette. Jedes Directory enthaelt Dateizeichnungen fuer alle Dateien einer Diskette. Ein Beispiel einer Dateizeichnung ist "NAME.TXT", wobei "NAME" aus bis zu 8 Zeichen bestehen kann und ".TXT" den Dateityp bezeichnet. Dateizeichnungen koennen aus Buchstaben, Ziffern und Sonderzeichen bestehen. Folgende Zeichen duerfen jedoch nicht verwendet werden:

< > . , = ; * ? []

Wenn ein Kommando auf mehrere Disketten oder alle Dateien angewendet werden soll, ist hierzu ein spezielles Format notwendig. Kommandos, die eine solche Mehrfachanwendung in ihrem Kommando erlauben, benoetigen anstelle der reellen Dateizeichnung sogenannte Dateigruppenzeichnungen. Sie schliesst eine bestimmte Gruppe von Dateien ein. Die Dateigruppenzeichnung koennen Buchstaben, Ziffern und Sonderzeichen sowie die beiden Zeichen "*" und "?" enthalten. In der Kombination der Dateigruppensymbole "*" und "?" mit den anderen Zeichen werden mit dem zugehoerigen Kommando bestimmte Dateigruppen angesprochen. Das Kommando bezieht sich dann auf alle Dateien, die an der Position des oder der Fragezeichen ein bzw. mehrere beliebige Zeichen haben. Waehrend sich Fragezeichen auf einzelne Zeichenpositionen beziehen, werden mit einem Stern mehrere Zeichen angesprochen.

Beispiele:

Dateigruppenzeichnung	Beispiel fuer angesprochene Datei	Beispiel fuer nicht angesprochene Datei
0???	OTTO,ORTE	ORT,OTTER
T????Y.*	TEDDY.COM TARBY.COM TILLY	TINY.COM TARABY.ASM TONY
.	jede Datei	

4.6. Dateikennungen

Dateizeichnungen enthalten meist auch Zeichen fuer die Kennung des Dateityps (bis zu 3 Zeichen nach dem Punkt). Diese Zeichen koennen zur Charakterisierung des Dateityps benutzt werden. Bei einigen Dateien ist die Angabe eines Dateityps unbedingt notwendig, bei anderen aus organisatorischen Gruenden erwuenscht. Wird eine Dateizeichnung als Argument eines Kommandos bzw. einer OS/M-Routine benoetigt, dann muss die vollstaendige Bezeichnung eingegeben werden. Die einzige Ausnahme ist die Anwendung von nicht residenten

Kommandos (hier wird intern der richtige Dateityp ".COM" ueberprueft).

Folgende Dateitypen sind festgelegt:

Dateityp	Zuordnung	Beispiel
.COM	fuer Befehlsdateien (command files) reserviert	PIP.COM LOAD.COM
.ASM	fuer Assembler-Quellprogramme reserviert, die mit der ASM-Routine assembliert werden	PROG1.ASM
.PRN	fuer Listausdruck der assemblierten Maschinensprachprogramme reserviert	PROG1.PRN
.HEX	fuer Programmdateien im "HEX-Format" (Maschinenkode) reserviert, die mit der LOAD-Routine geladen werden koennen	PROG1.HEX
.BAS	fuer BASIC-Quellprogramme reserviert	PROGBAS.BAS
.INT	fuer BASIC-Zwischendateien reserviert, die mit dem Interpreter ausgefuehrt werden sollen	PROGBAS.INT
.BAK	fuer Sicherungsdateien (backup) reserviert	LETTER.BAK
.MMN	fuer Zwischendateien reserviert, werden wieder geloescht	LETTER.MMN
.SUB	Textdateien mit residenten oder transienten Kommandos oder Programmen, die nacheinander durch das SUBMIT-Programm ausgefuehrt werden sollen	TRANS.SUB

4.7. Dateiattribute

Jeder Datei koennen bestimmte Attribute zugeordnet werden. Ueber das Kommando STAT koennen die Attribute angezeigt werden.

Es gibt vier Attribute:

- R/W (read/write) : Dateien mit diesem Attribut koennen gelesen, ueberschrieben und geloescht werden.
- R/O (read/only) : Dateien mit diesem Attribut koennen nur gelesen werden.

- DIR (directory-
Datei) : Dateien mit diesem Attribut werden
durch das Kommando DIR angezeigt.
- SYS (System-Datei) : Dateien mit diesem Attribut werden
bei dem Kommando DIR nicht ange-
zeigt. Sie koennen nur unter zu-
saetzlichen Angaben mittels PIP
kopiert werden.

5. BIOS

Im BIOS sind alle Treiber der E-/A-Geraete enthalten. Ueber eine Sprungtabelle, eine Aneinanderfuegung von Sprungbefehlen, koennen die Treiberrountinen der geratespezifischen BIOS-Unterprogramme aufgerufen werden. In der Tabelle sind auch BYTE- und WORD-Werte unterschiedlicher Bedeutung enthalten.

5.1. Adressenplan BIOS-Rufe

Adresse	Bezeichnung	Erlaueuterung
F000H	BOOTE JP BOOT	Kaltstartroutine
F003H	WBOOTE JP WBOOT	Warmstartroutine
F006H	JP CSTE	Console Input Status Abfrage, ob ein Zeichen von der Systemkonsole eingegeben wurde A=00H kein Zeichen A=FFH ein Zeichen liegt vor
F009H	JP CI	Console Input Einlesen von einem Zeichen von der Systemkonsole in Register A; RETURN erst, wenn ein gueltiges Zeichen vorliegt
F00CH	JP CO	Console Output Ausgabe eines Zeichens von Register C an die Systemkonsole
F00FH	JP LO	List Output (Drucker) Ausgabe eines Zeichens vom Register C an den Drucker
F012H	JP SO	Serieller Koppelkanal - Output Ausgabe eines Zeichens vom Register C an den seriellen Koppelkanal
F015H	JP SI	Serieller Koppelkanal - Input Einlesen eines Zeichens vom seriellen Koppelkanal ins Register A; RETURN erst wenn ein gueltiges Zeichen vorliegt
F018H	JP HOME	Kopfpositionierung des akt. Laufwerks auf Spur 0
F01BH	JP SELDSK	Auswaehlen des Laufwerks fuer eine folgende FD-Operation

(im Register C 0=A, 1=B, ...)

F01EH	JP SETTRK	Einstellen der Spurnummer des ausgewählten Laufwerks fuer nachfolgende PD-Operationen. Das Registerpaar BC enthaelt die Spurnummer
F024H	JP SETSEC	Einstellen der Sektornummer des ausgewählten Laufwerks fuer nachfolgende PD-Operationen. Im Registerpaar BC ist die Sektornummer enthalten.
F024H	JP SETDMA	Einstellen der DMA-Adresse. Im Registerpaar BC steht die einzustellende Adresse
F027H	JP READ	Lesen eines Sektors nach entsprechend eingestellten Parametern (SETDSK, SETTRK, SETSEC, SETDMA) fuer A=0 fehlerfrei A=1 Fehler
F02AH	JP WRITE	Schreiben eines Sektors; Parameter, Rueckmeldung und Wiederholung wie bei READ
F02DH	JP LSTS	List Status Status vom Drucker abfragen, wobei A=00H Drucker nicht bereit A=FFH Drucker bereit
F030H	JP SECTRN	Sector Translate Zuordnung log. - physischen Sektor DE - Adresse der Zuordnungstabelle BC - log. Sektornummer HL - Rueckmeldung der physischen Sektornummer
F033H	JP SISTS	Serieller Koppelkanal - Input Status Abfrage, ob ein Zeichen vom Koppelkanal anliegt A=00H kein Zeichen A=FFH ein Zeichen liegt vor
F036H	JP SOSTS	Serieller Koppelkanal - Output Status Abfrage, ob ein Zeichen vom Koppelkanal ausgegeben werden kann A=00H nicht bereit A=FFH Zeichen kann ausgegeben werden
F04BH	JP FLOPPY	Aufruf des Floppy-Disk-Treibers mit dem UDOS-Parametervektor; Le-

sen und Schreiben von Sektoren
ohne Fehlerausschrift
IY=Adr. UDOS-Parametervektor

Konstantenbereiche

FO4EH	PDCOMP DEFB 2H	enthaelt die Floppy-Disk-Konfiguration s.a. 5.4.
FO50H	CRGERR DEFW 0	enthaelt die Anzahl der CRC-Fehler bei Diskettenzugriffen seit dem Systemstart.
FO52H	SECCRR DEFW 0	enthaelt die Anzahl der Sektorfehler seit dem Systemstart
FO54H	TRKERR DEFW 0	Enthaelt die Anzahl von Spur-Fehler seit dem Systemstart
FO56H	DEFW SIOBASE	Anfangsadresse der SIO-Parametertabelle (s.a. 5.2.)

5.2. Treiber fuer serielle Kanale - Systemkonsole, Drucker und serieller Koppelkanal

Drei der vier seriellen Kanale des 8-Bit-Teils im P8000 werden von OS/M bedient. Sie werden fuer den Anschluss der Systemkonsole (V24-Terminal), eines Druckers (V24- oder IFPS-Schnittstelle) und fuer einen seriellen asynchronen Koppelkanal (V24 oder IFPS) benutzt. Empfangene Zeichen werden in Eingabepuffern zwischengespeichert. Interrupt ist nur fuer Zeichenempfang zugelassen. Die Uebertragungsparameter sind, fuer jedes der seriellen Gerate getrennt, in Parametertabellen abgelegt. Die Anfangsadresse der Parametertabellen ist in SIOBASE (s.a. Sprungtabelle) gespeichert. Der FLAG-Parameter kann von Anwenderprogrammen modifiziert werden. Das ist besonders fuer den seriellen Koppelkanal wichtig. Fuer eine permanente Modifizierung aller Uebertragungsparameter sind die Initialisierungswerte der Parametertabellen (Byte 0 - Byte 8 und Byte 17 - Byte 19) im ersten Umlader (Spur 0, Sektor 1) der Systemdiskette abgelegt. Der erfahrene Systembetreuer kann dort Aenderungen vornehmen.

Anfangsadressen der SIO-Parametertabellen:

Serieller Koppelkanal: (SIOSI)	Byte 60H im 1. Umlader (SIOBASE)+0 im BIOS
Systemkonsole, Terminal: (SIOCON)	Byte 68H im 1. Umlader (SIOBASE)+10H im BIOS
Drucker: (SIOLP)	Byte 72H im 1. Umlader (SIOBASE)+20H im BIOS

Aufbau der Parametertabelle:Byte 0 - FLAG

- Bit 0: 1 = Die Empfangsinterruptroutine wertet spezielle Zeichen zur Steuerung der Ausgabe aus. Das XOFF-Zeichen stoppt die Zeichenausgabe so lange, bis das XON-Zeichen empfangen wurde.
- Bit 1: 1 = XOFF wird automatisch gesendet, wenn im Eingabepuffer nur noch 20 Bytes frei sind, XON wird gesendet, wenn der Puffer bis auf 10 Zeichen geleert wurde.
- Bit 2: (Nur-Lese-Bit) 1 = Ausgabe wurde durch empfangenes XOFF unterbrochen.
- Bit 3: (Nur-Lese-Bit) 1 = Die Eingabeinterruptroutine hat ein XOFF gesendet.
- Bit 6: 1 = Ueberlauf des Eingabepuffers, empfangenes Zeichen konnte nicht mehr uebertragen werden. Das Bit wird nicht automatisch rueckgesetzt.
- Bit 7: (Nur-Lese-Bit) 0=kein Zeichen im Eingabepuffer
1=Zeichen im Eingabepuffer

Byte 1 - CTCADR

In diesem Byte ist die Adresse des CTC-Kanals enthalten, der fuer die Erzeugung der Baudrate des E/A-Kanals zustandig ist. Es besteht folgende Zuordnung:

- CTC1 Kanal 0 (Adresse 2CH) - SIO0 Kanal A
 CTC1 Kanal 1 (Adresse 2DH) - SIO0 Kanal B
 CTC1 Kanal 2 (Adresse 2EH) - SIO1 Kanal A
 CTC0 Kanal 0 (Adresse 08H) - SIO1 Kanal B

Byte 2 - SPEED (CTC-Zeitkonstante)

In diesem Byte ist die Zeitkonstante fuer die Erzeugung der Baudrate enthalten. Mit einem SIO-Vorteiler von 16 ergeben sich folgende Werte:

- 1 = 38400 bit/s
 2 = 19200 bit/s
 4 = 9600 bit/s
 128 = 300 bit/s

Byte 3 - SIOADR

Diese Byte enthaelt die Zuordnung des E/A-Geraetes zum SIO-Kanal:

- SIO0 Kanal A (Datenadr. 24H) - serieller Koppelkanal
 SIO0 Kanal B (Datenadr. 26H) - Systemkonsole
 SIO1 Kanal A (Datenadr. 28H) - keine Zuordnung
 SIO1 Kanal B (Datenadr. 2AH) - Drucker

Eine Aenderung der Zuordnung sollte nur fuer den seriellen Koppelkanal erfolgen, gleichzeitig ist Byte 1 zu aendern.

Byte 4 - LPARM (Line Parameter)

Bit 1;0	Paritaet	0 = keine Paritaet	
		1 = ungerade Paritaet	
		2 = keine Paritaet	
		3 = gerade Paritaet	
Bit 3;2	Anzahl Stopbits	0 = undefiniert	
		1 = 1 Stopbit	
		2 = 1,5 Stopbits	
		3 = 2 Stopbits	
Bit 4;5	Bits pro Zeichen	0 = 5 Bits	
		1 = 6 Bits	
		2 = 7 Bits	
		3 = 8 Bits	
Bit 6;7	SIC Clock Rate	0 = x1	(Ext. Synchronisation erforderlich)
		1 = x16	
		2 = x32	
		3 = x64	

Byte 5 - LCONTR (Line Control)

Bit 1	1 = Request To Send (RTS) Leitung aktiv
Bit 7	1 = Data Terminal Ready (DTR) Leitung aktiv

Byte 6 - XOFF

In diesem Byte ist das XOFF-Zeichen definiert.

Byte 7 - XON

In diesem Byte ist das XON-Zeichen definiert.

Byte 8 - INMSK (Input Maske)

Dieses Byte wird zum Ruecksetzen der Paritaet der empfangenen Zeichen verwendet.

Byte 9 - GET

GET-Pointer fuer interne Benutzung

Byte 10 - STORE

STORE-Pointer fuer interne Benutzung

Byte 11 - COUNT

Anzahl der gueltigen Zeichen im Eingabepuffer

Byte 12 - BUFLen

Laenge des Eingabepuffers

Byte 13/14 - BUFADR

Anfangsadresse des Eingabepuffers

Byte 15/16 - INTADR

Adresse der Interruptserviceroutine fuer Zeichenempfang

Byte 9-16 duerfen nicht modifiziert werden, sie sind im ersten Urlader auch nicht vorhanden. Fuer den Drucker existiert eine erweiterte Parametertabelle. Die Parameter koennen im BIOS fuer die zeitweilige und im ersten Urlader fuer eine permanente Aenderung modifiziert werden. In ersten Urlader folgt nach Byte 8 das Byte 17 der erweiterten Parametertabelle.

Byte 17 - LINES

Anzahl der Zeilen pro Seite; der Druckertreiber gibt nach dieser Anzahl automatisch einen Seitenvorschub (Form Feed) aus.

Byte 18 - COLS

Anzahl der Zeichen pro Zeile (rechter Rand); alle weiteren Zeichen der Zeile werden nicht ausgegeben.

Byte 19 - INDENT

Einrueckung (linker Rand), vor dem 1. Zeichen einer Zeile wird die in INDENT festgelegte Anzahl Leerzeichen ausgegeben.

Die Parameter sind bei der Auslieferung des PC000 wie folgt eingestellt:

	Konsole	Drucker	Koppelport
FLAG	00H	01H	03H
CTCADR	2DH	00H	2CH
BAUD	04H	04H	04H
SIOADR	26H	2AH	24H
LPARM	7CH	7CH	7CH
LCONTR	82H	82H	82H
XOFF	13H	13H	13H
XOH	11H	11H	11H
INMSK	7FH	7FH	7FH
GET	00H	00H	00H
STORE	00H	00H	00H
COUNT	00H	00H	00H
BUFLEN	keine festen Adressen		
INTADR	keine festen Adressen		
LINES	-	66	-
COLS	-	120	-
INDENT	-	0	-

5.3. Druckertreiber

Zusätzlich zu Punkt 5.2. besitzt der Druckertreiber einige Besonderheiten. Er führt automatisch mit den eingestellten Parametern LINES, COLS, und INDENT eine Seitenformatierung aus. Jedes beliebige ASCII-Zeichen (00H bis 7FH) kann an den Drucker ausgegeben werden. Nichtdruckbare Zeichen, ausser Carriage Return (CRLF), Line Feed (LF), Form Feed (FF), Tabulator (TAB) und Backspace (BS) werden bei der Formatierung nicht berücksichtigt. Tabs werden in Leerzeichen aufgelöst. Mit der Ausgabe der ersten Escape-Folge (beginnend mit 1BH) geht der Treiber in den Direkt-Mode. Die automatische Formatierung wird eingestellt, alle Zeichen werden unverändert an den Drucker ausgegeben. Beendet wird der Direkt-Mode durch Ausgabe des Nicht-ASCII-Zeichens 80H oder spätestens bei Beendigung des laufenden Programms. Vor Beendigung des Drucks im Direkt-Mode sollte der Drucker auf den Anfang einer neuen Seite positioniert werden.

5.4. Floppy-Disk-Treiber

Der Floppy-Disk-Treiber benutzt den UDOS-Parametervektor. Dieser ist im UDOS-Systemhandbuch beschrieben. Der Floppy-Disk-Treiber stellt sich ueber das FDCONF-Word auf die Laufwerkstypen ein. Der Initialisierungswert ist im ersten Umlader auf Byte 7EH/7FH abgelegt. Er kann dort vom Systembetreuer geaendert werden (Kommando SETFD). Das unter Byte

von PDCONF (Adr. 7EH) enthaelt jeweils im Halbbyte (L-Teil: Laufwerk 0; H-Teil: Laufwerk 1) den Typ der im P8000 eingebauten Laufwerke, das obere Byte (7FH) enthaelt jeweils im Halbbyte (L-Teil: Laufwerk 2; H-Teil: Laufwerk 3) den Laufwerkstyp im Beisteller.

Die Typen sind wie folgt vereinbart:

- 0 - kein Laufwerk
- 1 - 8"-Laufwerk
Format: SD-SS 26x128x77
Operation: Lesen und Schreiben
- 2 - 5 1/4"-Laufwerk (K5600.10)
Format: DD-SS 16x256x40
Operation: Lesen und Schreiben
- 3 - 5 1/4"-Laufwerk (K5600.20)
Format: DD-SS 16x256x40
Operation: Lesen
- 4 - 5 1/4"-Laufwerk (K5600.20)
Format: DD-SS 16x256x80
Operation: Lesen und Schreiben
- 5 - 5 1/4"-Laufwerk
Format: DD-DS 32x256x80
Operation: Lesen und Schreiben

DD=double density
SD=single density
DS=double sided
SS=single sided

6. BDOS-Funktionen

Vom OS/M werden folgende logische Gerate unterstuetzt:

- die Konsole
- ein Drucker
- 2 ... 4 FD-Laufwerke mit den logischen Bezeichnungen
A ... D
- ein serieller asynchroner Koppelkanal (Kommunikations-
schnittstelle)

Fuer den Zugriff auf die logischen Gerate sind die BDOS-Funktionen zu verwenden. Zum Aufruf dieser insgesamt 37 Funktionen gibt es ein standardisiertes Verfahren. Die Funktionen sind von 0 - 36 durchnummeriert. Der BDOS ist ein Teil des OS/M, der die logischen Geratetreiber beinhaltet. Jeder Aufruf erfolgt ueber die Anspruchsstelle

Adresse 0005H

mit einem Unterprogrammaufruf (CALL 0005H). Dort ist vom OS/M ein Sprung an den Anfang von BDOS eingetragen. Zur Spezifikation des Kommandos ist die Nummer der BDOS-Funktion in Register C einzustellen. Sind fuer die ausgewaehlte Funktion weitere Zusatzfunktionen erforderlich, sind diese vor dem "CALL 0005H" im Registerpaar DE abzulegen. Demnach sieht der Aufruf einer BDOS-Funktion n im allgemeinen wie folgt aus:

```
LD C,n
{LD DE,parameter}
CALL 5
```

Erfolgt bei der angegebenen BDOS-Funktion eine Rueckkehrmeldung, so ist diese entweder im Register A oder im Registerpaar HL abgelegt.

6.1. BDOS-Funktion 0: - System RESET

Mit dieser Funktion erfolgt die Rueckkehr in den CCP-Status durch einen Warmstart. Das System meldet sich mit dem aktuellen Laufwerk und dem Bereitschaftszeichen.

6.2. BDOS-Funktionen fuer Konsole, Drucker und seriellen Koppelkanal

1: READ CONSOLE

Erwartet die Eingabe eines Zeichens ueber die Konsole nach dem Register A. Die Routine wartet solange, bis ein Zeichen ueber die Tastatur eingegeben wurde

2: WRITE CONSOLE

Das im Register E befindliche Zeichen wird an die Konsole ausgegeben.

3: IN SERIAL

Liest ein Zeichen vom seriellen Koppelkanal in das Register A, wenn dieser aktiv ist.

4: OUT SERIAL

Sendet das im Register E stehende Zeichen an den seriellen Koppelkanal

5: WRITE LIST

Das Zeichen, das im Register E steht, wird an den Drucker ausgegeben.

6: DIRECT CONSOLE INPUT/OUTPUT

Direktverkehr mit der Konsole. Enthaelt das Register den Wert FFH, so wird das Zeichen von der Tastatur in das Register A uebergeben. Enthaelt das Register A den Wert 00H, so wurde keine Taste betaetigt.

Ist der Wert des Registers E ungleich FFH, so wird dieses Zeichen an die Konsole ausgegeben.

7: GET I/O-STATUS

Wird im OS/M nicht benutzt.

8: SET I/O-STATUS

Wird im OS/M nicht benutzt.

9: PRINT BUFFER

Sendet eine Zeichenkette an die Konsole, beginnend mit der im Registerpaar DE angegebenen Speicheradresse. Ihr Ende wird durch Erkennen eines α -Zeichens definiert.

10: READ BUFFER

Eine Zeichenfolge wird von der Konsole nach dem ueber das Registerpaar DE adressierten Zeichenpuffer uebertragen, bis CR oder LF (ODH oder OAH) erscheint. Gleichzeitig erfolgt die Anzeige der eingegebenen Zeichen auf dem Bildschirm.

Der Eingabepuffer muss folgenden Aufbau haben:

Das 1.BYTE des Puffers muss die angebbare Zeichenzahl enthalten (1 - 255). Werden mehr Zeichen eingegeben, wird die Eingabe unterbrochen.

Das 2.Byte des Puffers enthaelt die tatsaechliche Anzahl eingegebener Zeichen nach Ausfuehrung.

Ab der 3. Position des Puffers stehen die eingegebenen Zeichen. Der Puffer wird vor der Eingabe nicht geloescht.

11: INTERROGATE CONSOLE READY

Abfrage der Konsole, ob ein Zeichen eingegeben wurde. Wenn das geringwertigste Bit des Bytes im Register A=1 ist, so wurde eine Taste betaetigt, im anderen Fall wurde keine Taste betaetigt.

12: RETURN VERSION NUMBER

Sendet die Versionsnummer von BDOS zum Register A. Damit wird dem Anwender programmtechnisch die Version von BDOS mitgeteilt.

6.3. BDOS-Funktionen fuer den Diskettenzugriff

Wenn vom BDOS Dateimanipulationen durchgefuehrt werden, wird ein Parameterblock benoetigt, der Informationen ueber die Dateizeichnung und der Speicheraufteilung beinhaltet. Dieser Block wird File Control Block (FCB) genannt. Seine Adresse ist vor dem BDOS-Aufruf im Registerpaar DE bereitzustellen.

Der PCB besteht bei sequentiellen Zugriff aus 33 Bytes, bei direktem Zugriff aus 36 Bytes und setzt sich wie folgt zusammen:

Feld	Byte-Position	Bemerkungen
ET	0	Entry Type. Entspricht dem Laufwerks- Kode: 0 ... akt. Laufwerk 1 ... LW A 2 ... LW B 3 ... LW C 4 ... LW D
FN	1 - 8	File Name. Dateiname
FT	9 - 11	File Type. Die Bits 7 von Byte 9 und 10 haben folgende Bedeutung: Bit 7 von Byte 9 = 1 : R/O-Datei Bit 7 von Byte 10 = 0 : SYS-Datei
EK	12 13 - 14	File Extent. akt. Bereichsnummer, i.allg. Null Fuer Systemzwecke reserviert.
RC	15	Record Count. Anzahl der Saeetze (128- Byte-Satz) des akt. Bereichs (Extent) (Wertebereich 0 - 128)
DM	16 - 31	Disk Allocation Map. Wird vom BDOS gefuehrt und fuer eigene Zwecke verwen- det.
NR	32	Next Record Number. Akt. Satznummer fuer den naechsten sequentiellen Zu- griff.
RR	33 - 35	Optional Record Number. Anzahl der Re- cords im Direktzugriff. Byte 33 - L-Teil } der Satznummer im } Bereich von 0 bis Byte 34 - H-Teil } 65535 Byte 35 - fuer Ueberlauf

Vor jeder Dateioperation ist ein so aufgebauter PCB bereit-
zustellen. Vor dem Eroeffnen einer Datei sind in den
Bytes 0 - 11 die erforderlichen Eintragungen zu machen,
waehrend die anderen Bytes auf 00H zu setzen sind.

13: RESET DISK SYSTEM

Diese Funktion initialisiert das BDOS, setzt alle Laufwerke auf den Status read/write, selektiert Laufwerk A und uebernimmt die voreingestellte DMA-Adresse (80H) fuer den Ladevorgang. Diese Funktion kann innerhalb eines Programms fuer das Einlegen neuer Disketten ohne Eingabe von <CTRL> C oder eines neuen Systemstart benutzt werden.

14: SELECT DISK

Hiermit wird ein Laufwerk selektiert. Der Inhalt von Register E gibt an, welches Laufwerk fuer die folgenden Dateioperationen das aktuelle Laufwerk ist. (0 fuer A, 1 fuer B, 2 fuer C, 3 fuer D).

15: OPEN FILE

Im Registerpaar DE steht die Adresse des FCB. BDOS sucht den FCB in dem Directory der Diskette, entsprechend der benannten Datei, die auf dem durch das Byte 0 des FCB ausgewaehlten Laufwerks existieren muss. Wird der gesuchte FCB in dem Directory gefunden, wird das Directory entsprechend geaendert und die Datei fuer einen folgenden Zugriff vorbereitet.

Wird die Datei gefunden, so wird im Register A der Wert 00H, 01H, 02H oder 03H gesetzt. Im anderen Fall enthaelt das Register den Wert FFH.

Das Byte 32 des FCB muss auf den Wert 0 gesetzt werden, wenn nach dem Eroeffnen der Datei sequentiell auf den ersten Satz zugegriffen werden soll.

16: CLOSE FILE

Mit dieser Funktion werden die im FCB gespeicherten Informationen in das entsprechende Directory uebernommen. Im Registerpaar DE steht die Adresse des FCB. Die Datei muss aber vorher durch die Funktion 15 oder 22 aktiviert sein. Steht nach dem Funktionsaufruf im Register A ein Wert zwischen 00H und 03H, so war die Funktion erfolgreich. Der Wert FFH signalisiert, dass die Datei nicht gefunden wurde.

17: SEARCH FOR FILE

Sendet die im Registerpaar DE stehende Adresse eines File Control Blocks. Das Directory wird von Anfang nach einer Eintragung entsprechend der FCB-Eintragung (ET, FN, FT, EX) durchsucht. Fragezeichen im Namen lassen an dieser Position jeden Buchstaben zu. Ein Fragezeichen im Byte 0 ermoeglicht das Durchsuchen nach allen Eintragungen im aktuellen Laufwerk.

Die Eintragung, die entsprechend der FCB-Vorgaben ermittelt wurde und 32 Bytes lang ist, wird in einem 128 Byte-Puffer (DMA) eingetragen. Dieser beginnt entweder bei 80H oder bei

einer mit der Funktion 26 vorgegebenen Adresse.
In Register A wird

A=0	:	Adresse=DMA
A=1	:	Adresse=DMA+20H
A=2	:	Adresse=DMA+40H
A=3	:	Adresse=DMA+60H
A=FFH	:	Fehler file not found

uebergeben.

18: SEARCH FOR NEXT OCCURENCE

Diese Funktion, angewendet nach der Funktion 17, sucht den naechsten File Control Block in dem Directory, in dem der angegebene Dateiname auftritt. Diese Funktion kommt zur Anwendung, wenn nach weiteren Dateien (im Namen sind Fragezeichen enthalten) oder weitere Eintragungen der gleichen Datei gesucht wird.

19: DELETE FILE

Ermittelt die Adresse des File Control Blocks in dem Directory, das den angegebenen Dateinamen enthaelt und loescht die Datei auf der Diskette. Enthaelt der Dateiname oder Dateityp im FCB Fragezeichen, so werden alle Dateien geloescht, die in den anderen Positionen gleiche Zeichen enthalten. Register A=FFH zeigt an, dass keine Datei mit der Dateibezeichnung im FCB gefunden wurde. Erfolgreiches Loeschen einer Datei wird mit den Werten 00H, 01H, 02H oder 03H im Register A angezeigt.

20: READ SEQUENTIALLY

Wenn die Datei eroeffnet oder mit einer "make"-Funktion aktiviert ist, werden mit dieser Funktion die naechsten 128 Bytes (1 Record) in den Arbeitsspeicher, beginnend von der selektierten DMA-Adresse gelesen. Im Registerpaar DE ist die FCB-Adresse vorzugeben. Die Angabe "NR" im FCB gibt den naechsten zu lesenden Satz an. Fuer NR=00H wird der 1. Satz gelesen. "NR" wird nach dem Lesen automatisch um 1 erhoeht. Wird im Register A ein Wert ungleich 00H gemeldet, so ist das Dateiende erreicht. Der Wert 00H steht fuer einen erfolgreichen Lesevorgang. A=FFH signalisiert einen allgemeinen Lesefehler.

21: WRITE SEQUENTIALLY

Wenn die Datei eroeffnet oder mit einer "make"-Funktion aktiviert wird, werden mit dieser Funktion die naechsten 128 Bytes (1 Record) beginnend mit der selektierten DMA-Adresse in die durch den FCB spezifizierte Datei geschrieben. Im Registerpaar DE ist die FCB-Adresse vorzugeben. Mit "NR" wird die Position des Satzes in der Datei festgelegt

und automatisch nach jedem Schreiben um 1 erhöht.
In A wird mit

A=FFH	allgemeiner Schreibfehler
A=01H	Schreiben in vorhandenes File
A=02H	Ende der Diskette
A=00H	ohne Fehler

angeschigt

Nach dem Schreiben muss die CLOSE-Funktion folgen, damit die Information im Directory auf der Diskette aktualisiert wird.

22: MAKE FILE

Entspricht der Funktion "OPEN FILE" mit der Ausnahme, dass vorher noch die Datei generiert wird. Die Funktion 15 (OPEN FILE) ist nur fuer bereits existierende Dateien wirksam. Im Registerpaar DE ist die FCB-Adresse vorzugeben. Bei Anwendung dieser Funktion muss darauf geachtet werden, dass nicht eine vorhandene Datei auf derselben Diskette neu generiert wird. Um sicher zu gehen, kann mit der Loeschfunktion die angesprochene Datei vorher geloescht werden. Die Datei wurde erfolgreich in dem Directory angelegt, wenn im Register A als Rueckmeldung der Wert 00H bis 03H steht. Ein FFH in Register A sagt aus, dass in dem Directory keine Eintragung mehr frei ist.

23: RENAME FILE

Mit dieser Funktion kann eine Datei auf der Diskette umbenannt werden. Der ueber das Registerpaar DE adressierte FCB hat dabei folgenden Aufbau:
Die Bytes 0 - 15 enthalten die "alte" Dateibezeichnung einschliesslich Laufwerksangabe, die Bytes 16 - 32 die "neue" Dateibezeichnung. Nach erfolgreicher Ausfuehrung steht im Register A ein Wert zwischen 00H und 03H. Steht im Register A der Wert FFH, so wurde die umzubennende Datei nicht gefunden.

24: RETURN LOG-IN VECTOR

Mit dieser Funktion werden die verfuegbaren Laufwerke ermittelt.

Im Register A und L entsprechen nach der Funktion die Bits 0 - 3 den logischen Laufwerken A,B,C,D. Ein gesetztes Bit ("1") bedeutet, dass auf die Diskette in diesem Laufwerk bereits zugegriffen wurde.

25: RETURN CURRENT DISK

Diese Funktion ermittelt eine Zahl, die mit der Laufwerksbezeichnung A,B,C,D korrespondiert und die auf das selektierte (aktuelle) Laufwerk hinweist.

Die Bits 0 - 4 des Registers A geben an, welches Laufwerk das aktuelle ist.

26: SET DMA-ADDRESS

Diese Funktion setzt Directory Memory Address (Adresse, auf die der Pointer nach einer Schreib-/Leseoperation zeigt) auf einen Wert, um so den Beginn bestimmter Datensätze anzuzeigen. Sie ist im Registerpaar DE vorzugeben. Bei einem Kaltstart, Warmstart oder Ruecksetzen der Laufwerke wird diese DMA-Adresse auf 80H gesetzt.

27: GET ADDRESS OF ALLOCATION VECTOR

Das System verwaltet fuer jedes verfügbare Laufwerk eine Zuteilungsvektor innerhalb des Arbeitsspeicher. Diese Funktion ermittelt die Adresse des Vektors des selektierten Laufwerks.

28: WRITE-PROTECT DISK

Die Funktion erlaubt das Setzen eines zeitweisen Schreibschutzes. Jeder Schreibzugriff auf das Laufwerk (bis ein Kalt- oder Warmstart gegeben wird) wird verhindert und veranlasst die Fehlermeldung:

OS/M BRR ON d: R/O

(d steht fuer das angesprochene Laufwerk A,B,C oder D)

29: GET=READ/ONLY VECTOR

Diese Funktion ermittelt einen Vektor, der die schreibgeschuetzten Laufwerke anzeigt, d.h., das read/only-Bit ist gesetzt. In den Registern A und L wird wie bei der Funktion 24 ein Kode abgelegt, aus dem zu entnehmen ist, welches logische Laufwerk den R/O-Status hat.

30 SET FILE ATTRIBUTES

Diese Funktion erlaubt das Setzen und Ruecksetzen von Dateiattributen. Das Register DE adressiert ein FCB. Ist das Bit 7 von Byte 9 gesetzt, so wird die Datei mit dieser Funktion zu einer R/O-Datei. Ist das Bit nicht gesetzt, wird die Datei zu einer R/W-Datei. Ein gesetztes Bit von Byte 10 versetzt die Datei in den SYS-Status. Im anderen Fall erhaelt die Datei das DIR-Attribut.

31: GET DISK PARAMETER ADDRESS

Diese Funktion ermittelt die Blockadresse des Laufwerkparameters. Nach dem Funktionsaufruf steht im Registerpaar HL die Adresse des Parameterblocks. Sie ist sehr nuetzlich bei der Berechnung des Speicherraums und der Aenderung der Laufwerkparameter bei Wechsel der Diskettenlaufwerke.

32: GET OR SET USER CODE

Mit dieser Funktion kann der USER-Code des aktiven Benutzerbereiches ermittelt werden.

Wird im Register E ein FFH eingetragen, so wird nach dem Funktionsaufruf im Register A der aktive Benutzercode zurueckgemeldet.

Ist der Wert im Register E ungleich FFH, so werden durch diese Funktion die unteren 5 Bits dieses Wertes als neuer Benutzercode eingestellt (00H ... 1FH). Da sich mit dem residenten Kommando USER nur die Benutzerbereiche (0 ... 15) einstellen lassen, sollte deshalb generell nur mit dem Benutzercode 00H ... 0FH gearbeitet werden.

33: READ RANDOMLY

Diese Funktion greift auf das Feld RR (Read Record) im File Control Block zu, selektiert und liest diesen Record. Dabei wird die Nummer des Records nicht veraendert. Die Adresse des FCB muss im Registerpaar DE stehen. Im Byte 33 ist der L-Teil und in Byte 34 ist der H-Teil der Adresse des Satzes einzutragen, der gelesen werden soll.

Es ist zu garantieren, dass bei den vorher erforderlichen Funktionen 15 und 22, das Byte "EX" im adressierten FCB auf Null gesetzt wird.

Nach dem Lesen werden durch diese Funktion ausserdem die akt. Satznummer "NR" und die Extentnummer "EX" gesetzt, um ein anschliessendes sequentielles Lesen zu ermoeöglichen. Im Register A kann es zu folgenden Meldungen kommen:

00H	ohne Fehler
01H	Lesen ungeschriebener Daten
03H	Aktualisierung der Directory-Eintragung kann nicht erfolgen aufgrund einer vorhergehenden Funktion Schreiben mit direktem Zugriff
04H	Lesen ungeschriebener Extents
06H	Zugriff ueber physisches Ende der Diskette
FFH	allgemeiner Lesefehler

34: WRITE RANDOMLY

Diese Funktion laeuft auf die gleiche Weise wie die direkt adressierte Leseoperation ab, mit dem Unterschied, dass hier die Daten von der selektierten DMA-Position zur Diskette geschrieben werden. Wie beim Lesen mit direktem Zugriff geben das Registerpaar DE die FCB-Adresse und dort

die Bytes 33 und 34 die Adresse des Satzes an. Die Nummer des Records wird nicht veraendert. Es werden aber die aktuelle Satznummer "NR" und die Extentnummer "EX" im FCB aktualisiert, so dass ein anschliessendes sequentielles Arbeiten moeglich ist.

Die Fehlerkodes sind die gleichen wie beim Lesen mit direktem Zugriff (ausser 01H und 04H), dazu kommt der Fehlerkode 05H. Diese Fehlermeldung signalisiert, dass in dem Directory fuer die erforderlichen Eintragungen kein Platz mehr frei ist.

35: COMPUTE FILE SIZE

Diese Funktion sendet die Adresse eines File Control Blocks und ermittelt die Adresse des "logical records", der sich an das Dateiende anschliesst (virtuelle Dateigroesse). Mit dieser Funktion koenne Daten einer bereits existierenden Datei hinzugefuegt werden, in dem diese Information zum Setzen der Recordposition verwendet und danach die gewuenschte Schreiboperation durchgefuehrt wird. Die virtuelle Groesse korrespondiert dann zur physischen Groesse, wenn die Datei sequentiell geschrieben wurde, ansonsten besitzt die Datei aufgrund der direkt adressierten Schreiboperationen unbenutzte Recordbereiche. Mit dem Registerpaar DE ist ein FCB zu adressieren. Nach Ausfuehrung der Funktion enthalten die Bytes 32 und 33 die Adresse nach dem letzten Satz der Datei.

36: SET RANDOM RECORD POSITION

Mit dieser Funktion wird die aktuelle Adresse eines Records (z.B. nach einer Folge sequentieller Schreib-/Lesevorgaenge) ermittelt. In einigen Faellen erweist es sich als sehr nuetzlich, wenn von sequentieller auf direkter Dateioperation umgeschaltet wird und beide Vorteile der Dateiorganisation mit verwendet werden sollen (schnelles Selektieren und Einlesen).

Das Registerpaar DE adressiert den FCB, in dessen Bytes 33 und 34 durch diese Funktion die Satzadressen des letzten sequentiellen Zugriffs abgelegt werden.

6. Allgemeine Fehlermeldungen

Bei der Anwendung des OS/M-Systems werden neben den kommandospezifischen Fehlern alle vom System erkannten Fehler durch 4 verschiedene Fehlermeldungen in folgendem Format mitgeteilt:

BDOS ERR ON d: Fehlerursache

Der Buchstabe fuer d identifiziert das Laufwerk, auf dem der Fehler erkannt wurde. Fuer das Wort Fehlerursache steht eine der folgenden Meldungen:

BAD SECTOR

SELECT

READ ONLY

FILE R/O

Zusaetzlich zu diesen Fehlerauschriften erfolgt im OS/M eine ueber das BIOS initiierte, vorangestellte Fehlerauschrift in folgender Form:

DISK ERROR nn D=x T=yyy S=zzz

nn	- UDOS Completion Code	C1 - ungueltiger Requestkode
		C2 - Laufwerk nicht bereit
		C3 - Schreibschutz Diskette
		C4 - Sektorfehler
		C5 - Spurfehler
x	- Laufwerksname (drive)	
yyy	- Spurnummer (track)	
zzz	- Sektornummer (sector)	

Fehler: BAD SECTOR

Ein "bad sector" (fehlerhafter Diskettensektor) wird dann erkannt, wenn von der Diskette durch den DISK-Controller keine definierte Information gelesen werden kann. Die Fehlerursache ist dann entweder bei der Diskette oder beim DISK-Controller zu suchen.

Genauso wird diese Fehlermeldung gegeben, wenn die Diskette beim Laufwerkszugriff noch nicht in das Laufwerk eingeschoben war oder wenn Dateien gelesen werden, die durch einen anderen Controller generiert worden sind. Trotz eines verwendeten gleichen Formats koennen geringe Abweichungen in der Aufzeichnung der Records vorliegen. Die gleiche Fehlermeldung wird auch angezeigt, wenn die Diskette durch unsachgemaesse Behandlung oder durch ein fehlerhaftes Programm zerstoert worden ist.

Zur Quittierung dieser Fehlermeldung kann mit <CTRL> C ein erneuter Systemstart eingegeben werden, wodurch das laufen-

de Programm oder die gerade durchgeführte Dateiarbeitung unterbrochen wird. Auch kann diese Fehlermeldung durch Druecken der RETURN-Taste quittiert werden, wodurch die Fehlermeldung - und damit der fehlerhafte Sektor - ignoriert wird.

Das Ignorieren gemeldeter Fehler kann in einigen Faellen zur weiteren Zerstoerungen auf der Diskette fuehren. Tritt der Fehler z.B. beim Beschreiben des Directory auf und wird er ignoriert, so kann die Diskette zerstoert und damit jeder Zugriff auf irgendeine Datei dieser Diskette unmoglich werden.

Fehler: SELECT

Diese Fehlermeldung wird angezeigt, wenn ein nicht existierendes Laufwerk angesprochen wird. Da OS/M maximal 4 logische Laufwerke ansprechen kann (A ... D), wird dieser Fehler stets beim Zugriff auf die logischen Laufwerke E, F, ... gemeldet.

Durch Betaetigen einer beliebigen Taste wird ein Warnotart ausgeloeost, das System geht in den Grundzustand.

Fehler: READ ONLY

Die Fehlermeldung "READ ONLY" wird angezeigt, wenn versucht wird, auf eine Diskette zu schreiben, der das "READ ONLY"-Attribut (z.B. ueber STAT) zugewiesen wurde.

Genauso gilt diese Fehlermeldung, wenn eine neue Diskette eingelegt wurde, ohne dass vor einem Zugriff <CTRL> C eingegeben oder das System neu gestartet worden ist. Immer wenn auf eine neue Diskette geschrieben werden soll, muss deshalb vorher <CTRL> C eingegeben werden.

Eine "READ ONLY"-Fehlermeldung kann durch Druecken einer beliebigen Konsoltaste quittiert werden. Danach wird automatisch ein Systemstart erzeugt, der dann ausserdem die betreffende Diskette wieder in den "READ WRITE"-Betrieb schaltet.

Fehler: FILE R/O

Diese Fehlermeldung wird gegeben, wenn eine Schreiboperation auf eine Datei mit einem "R/O"-Attribut versucht wird. Die "FILE-R/O"-Fehlermeldung kann durch Druecken einer beliebigen Konsoltaste quittiert werden. Danach muss jedoch, wenn noetig, das R/O-Attribut durch ein R/W-Attribut ersetzt werden (z.B. STAT-Kommando).

8. Literaturempfehlungen

- VEB Robotron-Buchungsmaschinenwerk

Karl-Marx-Stadt
Software-Zentrum
9010 Karl-Marx-Stadt
Postschliessfach 129

- . Anleitung fuer den Bediener SCP 1520
- . Anleitung fuer den Programmierer SCP 1520
(Teil I und Teil II (Sprachbeschreibung ASM))
- . Anleitung fuer den Systemprogrammierer SCP 1520
- . Hardwarebeschreibung
- . Anwenderdokumentation BASIC-Interpreter
- . Anwenderdokumentation BASIC-Compiler
- . Anwenderdokumentation C-Compiler
- . Anwenderdokumentation PASCAL
- . Anwenderdokumentation FORTRAN
- . Anwenderdokumentation Textverarbeitungssystem TP
- . Anwenderdokumentation Installierungs-Programm fuer TP
- . Anwenderdokumentation KOMBO-Druck
- . Schulungshandbuch fuer das Textverarbeitungssystem TP

- VEB Robotron

Bueromaschinenwerk Soemmerda

- . Systemhandbuecher fuer den Personalcomputer PC1715

Hinweise des Lesers zu diesem Dokumentationsband

Wir sind staendig bemüht, unsere Unterlagen auf einem qualitativ hochwertigen Stand zu halten. Sollten Sie deshalb Hinweise zur Verbesserung dieses Dokumentationsbandes bzw. zur Beseitigung von Fehlern haben, so bitten wir Sie, diesen Fragebogen auszufüllen und an uns zuzurückzusenden.

Titel des Dokumentationsbandes:

Ihr Name / Tel.-Nr.:

Name und Anschrift des Betriebes:

Genuegt diese Dokumentation Ihren Anspruechen? ja / nein
Falls nein, warum nicht?

Was wuerde diese Dokumentation verbessern?

Sonstige Hinweise:

Fehler innerhalb dieser Dokumentation:

Unsere Anschrift: Kombinat VEB ELEKTRO-APPARATE-WERKE
BERLIN-TREPTOW "FRIEDRICH EBERT"
Abteilung Basissoftware
Hoffmannstrasse 15-26
BERLIN
1193



KOMBINAT VEB
ELEKTRO-APPARATE-WERKE

BERLIN-TREPTOW > FRIEDRICH EBERT <

Hoffmannstraße 15-26, Berlin, DDR-1193

☎ 011 2263 eaw 011 2264 eaw

⚡

Die Angaben über technische Daten entsprechen dem bei Redaktionsschluß vorliegenden Stand. Änderungen im Sinne der technischen Weiterentwicklung behalten wir uns vor.

Ausgabe August 1986