
KAPITEL II

SOFTWARE

Inhaltsverzeichnis

1. Monitor

- 1.1 Aufbau, Funktionsweise und Aufruf
- 1.2 Befehlsvorat

2. Basic Interpreter

- 2.1 Allgemeines zum BASIC - Interpreter
- 2.2 BASIC - Kommandos

3. MS - DOS

4. UCSD-P System

5. Diagnose-Programme

1.1 Aufbau, Funktionsweise und Aufruf des MOS

Das MOS (Mikrocomputer- Operating- Systems) besteht aus einer Reihe von Programmen zur Abwicklung der MOS-Komandos.

Das MOS liest, analysiert und verarbeitet den JOB-Eingabe- Strom. Der JOB- Eingabestrom ist eine Folge von Anforderungen an Programmen sowie Dateien und Einheiten, welche von dieser verwendet werden.

Nach dem Betätigen des Netzschalters am Gerät und Durchlaufen der Einschaltoutine erscheint am Bildschirm eine Abfrage ob mit BASIC, MONITOR oder TERMINAL gearbeitet werden soll. (Kalt Start)

Die zweite Möglichkeit zum Aufrufen der Abfrage besteht durch das Drücken der RESET-Taste (Warm Start)

Die Abfrage erscheint auf dem Bildschirm nach der Einschaltoutine, wenn kein ROM - Pack vorhanden ist, und kein SCSI - Controller sich meldet.

Um ins BASIC - Programm zu gelangen muß die Funktionstaste **F1** gedrückt werden.

Mit Break und Ok meldet sich das BASIC-Programm bereit. Das Programm kann mit dem Kommando **SYSTEM** verlassen werden.

MONITOR-Programm erreicht man durchs Betätigen der Funktionsteste **F2**

Das TERMINAL-Programm erreicht man durchs Betätigen der Funktionstaste **F3**
(Wird in Verbindung mit Autonome Prozessorplatte benötigt)

Am Bildschirm erscheint eine weitere Abfrage ob eine Datei in Speicher geladen oder aus dem Speicher Datei auf Kassette geschrieben werden soll. Weiterhin kann auch eine Datei gedruckt oder das Programm beendet werden.

BTX-Zuschaltung erfolgt durch Betätigen der Funktionstaste **F4**.

Die CPU 8088 kann bis 1 MB direkt adressieren. Sie besitzt einen 16-bit-Rechner, jedoch nur einen 8-bit-breiten Datenbus. Der Rechner holt sich mit dem entsprechenden Befehl automatisch 2 Byte (= ein Wort).

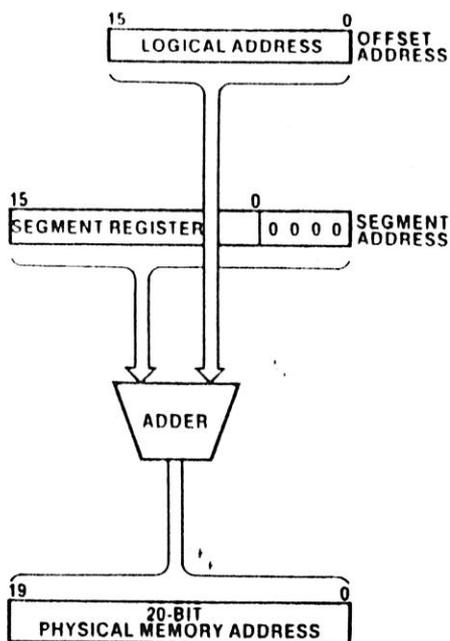
Ein Wort ist eine Folge von 2 Byte. Ein Wort steht so im Speicher, daß das höherwertige Byte auf der higher Memory Adresse steht.

Der Adreßbus ist 20 bit (2 1/2 Byte) breit. Das höchste Halb-Byte bezeichnet die sogenannte "Segment-Adresse". Der Speicher kann in beliebig viele Segmente eingeteilt werden. Ein Segment kann jedoch höchstens 64 KB groß sein. Somit ergibt sich die Speichergröße von $16 \times 64 = 1 \text{ MB}$.

Bit 0 - 15 ist die logische Adresse (Offset)

Bit 16 - 19 ist die Segment-Adresse (16 mögliche 64 KB Segmente)

Beispiel:



Eingabe: 1 0 0 0

Eingabe: 0 0

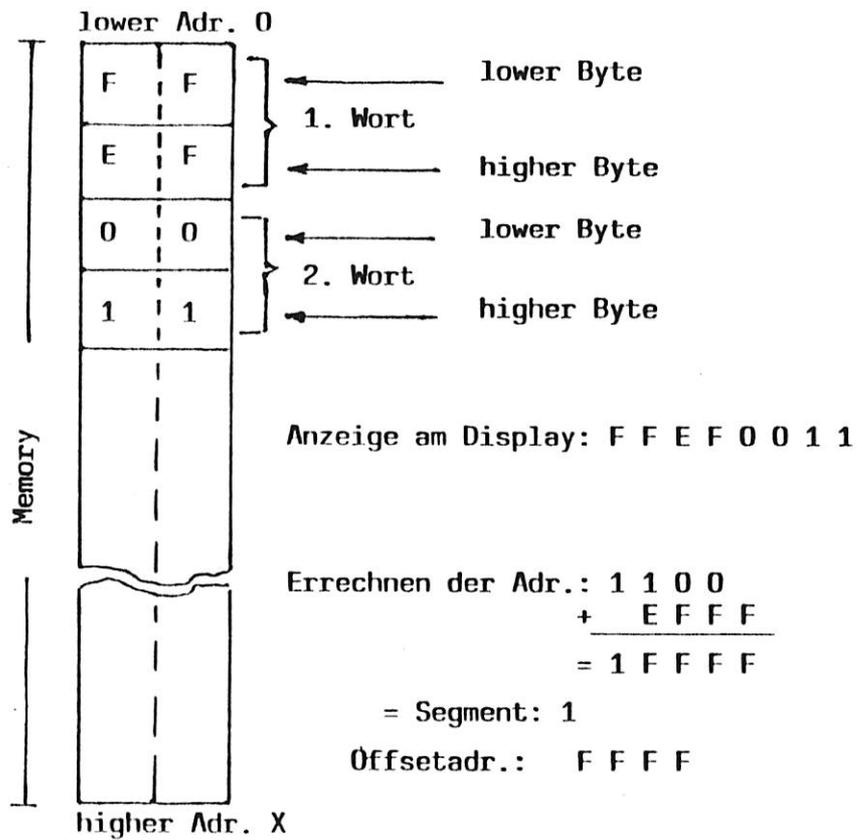
= Adresse: 0 1 0 0 0

Segment: 0 (erste 64 K)

Adresse: 1 0 0 0 (viertes K)

Erklärung zur Display-Anzeige von Speicheradressen

Die CPU beginnt mit der niedrigen Adresse eines Wortes (lower Adr.)



1.2 Befehlsvorrat

Durch Eingabe von zulässigen Befehlen wird dann das Programm veranlasst, bestimmte Aufgaben auszuführen.

Folgende Monitorfunktionen stehen über die Tastatur zur Verfügung:

D Display	Speicherbereich anzeigen
E End	Monitor Ende
F Fill	Speicherbereich füllen
G Goto	Programm starten ab Adresse
M Move	Inhalt im Speicher verschieben
R Register	Alle CPU- Register sichtbar machen
S Subtitue	Speicherzellen ändern
T Tracer	Progr. im Single - Step testen
U Uhrzeit	Zeit im Speicher setzen
P Printer	Drucker ein/aus

Alle Adresseingaben werden von dem Monitor gespeichert und falls keine Eingabe vorhanden als default Eingabe übernommen.

Die Eingabe geschieht über einen Zeilenbuffer und es ist somit eine Sofortkorrektur über die Backspace - Taste möglich.

Die Eingabe wird erst nach Betätigung der CR - Taste überprüft. Eine falsche Eingabe wird nochmals bis zu dem ersten falschen Zeichen ausgegeben.

Befehlsbeschreibung

Display

Mit dieser Funktion wird der Speicherinhalt byteweise in HEXA und ASCII angezeigt.

Die laufende Bildschirmausgabe kann mit den Tasten **CONTR** und **S** angehalten und mit **CONTR** und **Q** wieder gestartet werden.

Mit der Taste **BRACK** bzw. **CONTR** und **C** kann die Anzeige jederzeit abgebrochen werden.

Eingabe:

1. **D** und **CR**
Der Speicher wird ab der default - Adresse 16 Bildschirmzeilen ausgegeben.
2. **D** und **Startadresse**
Der Speicher gibt ab dem default-Segment und dem Offset (Startadesse) 16 Bildschirmzeilen aus.
3. **D** und **Segment: Startadresse**
Der Speicher gibt ab (Segment: Startadresse) 16 Bildschirmzeilen aus.
4. **D** und **, Endadresse**
Der Speicher gibt ab der default - Adresse bis zur Endadresse aus.
5. **D** und **Segment: Startadresse , Endadresse**
Der Speicher gibt über den angegebenen Adressbereich aus.

End

Mit dieser Eingabe wird zum aufrufenden Programm zurückgekehrt, wenn der Monitor über die Funktion **O** aufgerufen wurde.

Eingabe: **E** und **CR**

Fill

Der Speicher wird mit einer Konstanten gefüllt.

Eingabe:

1. **F** und **CR**
keine Funktion
2. **F** und **Startadresse, Endadresse, Füllwert**
Der Speicher wird vom default Segment: Startadresse bis Endadresse mit dem Wert (Füllwert) gefüllt.
3. **F** und **Segment: Adresse, Endadresse, Füllwert**
Der Speicher wird von (Segment. Adresse) bis (Endadresse) mit dem Wert (Füllwert) gefüllt.

Goto

Mit diesem Kommando kann ein im Speicher stehendes Programm bzw. Programmteile gestartet werden. Die mit dem **G** - Aufruf übergebenen Register können mit dem **R** - Kommando verändert bzw. angezeigt werden. Wird bei dem **G** - Aufruf eine Stop - Adresse, angegeben, so muß diese Stop - Adresse im RAM - Bereich liegen. Sonst sehe Trace - Kommando. Mit dem **G** - Aufruf können 2 Stopadressen übergeben werden.

Bei dem G - Kommando müssen alle Adressen eine Segment-information haben.

Eingabe:

1. **G und Segment: Adresse**
Es wird ein Programm ab der Adresse (Segment: Adresse) gestartet.
2. **G und Segment: Adr. , Stopsegment: Adr.**
Es wird ein Programm ab der Adresse (Segment: Adr.) gestartet. Wenn dieses Programm während seiner Ausführung zu der Adresse (Stopsegment: Adresse) kommt, so wird das Programm angehalten und in den Monitor verzweigt.
3. **G und Segment:Adr. , Stop 1 segment:Adr. , Stop 2 segment:Adr**

Es wird ein Programm ab der Adresse (Segment. Adr.) gestartet. Wenn dieses Programm während seiner Ausführung zu den Adressen (Stop 1 segment: Adr. oder Stop 2 segment:Adr.) kommt, so wird das Programm angehalten und in den Monitor verzweigt.

Move

Mit diesem Kommando kann ein Speicherbereich von einer Adresse zu einer anderen Adresse kopiert werden. Die max. Kopierlänge ist 64K.

Eingabe:

1. **M und Startadresse,Endadresse,Zieladresse**
Es wird ein Speicherbereich von default Segment: (Startadresse) bis (Endadresse) zu der default Segment: (Zieladresse) kopiert.
2. **M und Startsegment:Startadresse , Endadresse, Zielsegment:Zieladresse**
Es wird ein Speicherbereich von (Startsegment: Startadresse) bis (Endadresse) zu der Adresse (Zielsegment: Zieladresse) kopiert.

Printer

Mit dieser Funktion wird das Switch - Byte im
CO - Treiber geschaltet.
Bei dem ersten Aufruf wird der Drucker
zugeschaltet, und beim zweiten Aufruf wieder
abgeschaltet.

Eingabe: **P** und **CR**

Register

Mit dieser Funktion können die Registerinhalte der
CPU für das **G** - bzw. **T** - Kommando angesehen bzw.
geändert werden.

Eingabe:

1. **R** und **CR**
 Mit diesem Kommando werden alle CPU - Register
 am Bildschirm ausgegeben.

2. **R** und **Registerkennung**
 Mit diesem Kommando wird das spezifizierte
 Register am Bildschirm angezeigt.

3. **R** und **Registerkennung = Wert**
 Mit diesem Kommando wird das spezifizierte
 Register mit dem angegebenen Wert geladen.

Substitute

Mit dieser Funktion können Speicherinhalte geändert werden.

Wird **S (...)** eingegeben, so wird dieses Kommando byteorientiert arbeiten.

Wird **SW (...)** eingegeben, so wird dieses Kommando wortorientiert arbeiten.

Wird **SP (...)** eingegeben, so wird dieses Kommando pointerorientiert arbeiten.

Fortführungseingaben sind Eingaben, die die Bearbeitung eines Speicherbereiches abschließen.

Für diese Eingabe gibt es folgende Regel:

Wagenrücklauf Die Eingabe wird übernommen und die Funktion beendet.

Leerschritt Die Eingabe wird übernommen und bei der nächst höheren Speicheradresse weitergemacht.

Minus (-) Die Eingabe wird übernommen und bei der nächst tieferen Speicheradresse weitergemacht.

Eingabe:

1. **S und CR**
Mit dieser Funktion wird die default Adresse zum Ändern angezeigt.

Anzeige: Seg.: Adr. Inhalt =

Mit der Eingabe eines Hexawertes wird diese Eingabe in die angezeigte Speicheradresse gespeichert.

2. **S und Adresse**
Mit dieser Funktion wird das default Segment: Adresse zum Ändern angezeigt.

Anzeige: Seg: Adr. Inhalt =

Ändern des Inhaltes siehe 1.

3. **S und Segment: Adresse**
Mit dieser Funktion wird die Adresse Segment: Adresse zum Ändern angezeigt.

Anzeige: Seg: Adr. Inhalt =

Ändern des Inhaltes siehe 1.

Trace

Mit diesem Kommando kann ein Programm im Single - Step ausgetestet werden.

Es werden die CPU - Register mit dem unter Register änderbaren Werten geladen, und das Programm um einen Step gestartet. Nach diesem Programmschritt wird wieder in den Monitor verzweigt. Mit dem T - Kommando können auch Programme im PROM / ROM - Bereich getestet werden.

Eingabe:

1. **T und CR**
Mit diesem Kommando wird ein Programm ab der Adresse CS:IP gestartet und nach dem nächsten Programmschritt abgebrochen und wieder zum Monitor verzweigt.
2. **T und Segment: Adresse**
Mit diesem Kommando werden die Register CS:IP mit der Eingabe Segment: Adresse geladen und dann das Programm wie bei 1. gestartet.
3. **T und Segment:Adr,Stop 1 segment:Stopladresse, Stop 2 segment,Stop 2 adresse**

Mit diesem Kommando wird ein Programm ab der Adresse (Segment: Adresse) gestartet. Kommt das Programm während seiner Ausführung auf eine der beiden Stopadressen (Stop 1 seg:Stop 1 adresse oder (Stop 2 seg: Stop 2 adresse) so wird das Programm gestopt und zum Monitor verzweigt.

Bei dem Kommando 3. läuft das zu testende Programm nicht in Echtzeit ab, es können aber damit Programme im ROM getestet werden.

Uhrzeit

Mit diesem Kommando kann die System - Uhrzeit angezeigt bzw. verändert werden.

Eingabe:

1. **U** und **CR**
Die aktuelle Systemzeit wird angezeigt.

2. **U** und **Stunde, Minute, Sekunde**
Die eingegebene Uhrzeit wird als System - Zeit übernommen.

2.1 Allgemeines zum BASIC-Interpreter ver. 5.2

Nach dem Einschalten und Betätigen der Funktionstaste **F1** steht der Interpreter zur Verfügung.
Der Interpreter kennt drei verschiedene Zustandsarten:

Direktmodus, in dem eingegebene BASIC-Befehle sofort ausgeführt werden.

Programmiermodus, in dem BASIC-Befehle nicht ausgeführt, sondern als Programm gespeichert werden. Vor jeder Eingabezeile muß eine Zeilennummer stehen, die zwischen 0 - 65529 liegen kann.

Eine Programmzeile beginnt stets mit der Zeilen-Nr. und kann bis zu 255 Zeichen lang sein. Sind mehrere Befehle in einer Zeile zusammengefaßt, so sind sie durch einen Doppelpunkt : zu trennen. Der Abschluß einer Programmzeile wird durch Drücken der Eingabetaste gekennzeichnet.

Ausführungsmodus, in dem abgespeicherte BASIC-Programme ausgeführt werden.
(Aufrufen mit dem Kommando RUN)

Ist der BASIC-Interpreter arbeitsbereit, erscheint auf dem Bildschirm **Ok**. Im Direktmodus nimmt er jetzt Kommandos an.

Folgende Kommandos dienen zur Pflege und Ausführung von Programmen.

NEW	LLIST	TRON
RUN	DELETE	TROFF
AUTO	EDIT	CONT
LIST	RENUM	

Einige können innerhalb eines Programmes auch als BASIC-Befehle verwendet werden.

2.2 BASIC - Kommandos

Die nachstehend beschriebenen Formate für BASIC-Kommandos und BASIC-Befehle sind folgendermaßen zu interpretieren:

- Der in Großbuchstaben angegebene erste Begriff stellt den BASIC-Befehl oder das Kommando dar.
- In eckigen Klammern angegebene Zusätze zu den Befehlen können wahlweise angegeben werden.
- Angaben in spitzen Klammern stellen Parameter dar, die entsprechend zu besetzen sind.
- Nach jedem BASIC-Kommando ist mind. **1 Leerschritt** zwischen dem Kommando und den nachfolgenden Parametern einzufügen. Ansonsten wird **Syntax error** angezeigt.

NEW

Dieses Kommando löscht alle Programmzeilen, setzt alle numerischen Variablen auf Null und alle String-Variablen auf Blank. Das NEW-Kommando ist stets vor der Eingabe eines neuen Programmes zu geben.

BASIC kehrt nach der Ausführung des NEW-Kommandos stets in den Kommandomode zurück.

RUN [**< Zeilen-Nr.>**]

Nachdem das Anwenderprogramm in den Speicher geladen wurde, kann es mit dem RUN-Kommando zur Ausführung gebracht werden. Wurde keine Zeilen-Nr. angegeben, beginnt die Ausführung mit der niedrigsten Zeilen-Nr., sonst bei der angegebenen Zeilen-Nr.

Da das RUN-Kommando auch als BASIC-Befehl innerhalb eines Programmes angewendet werden kann, bestehen die Möglichkeiten, von einer bestimmten Programmstelle aus erneut das Programm zu starten.

AUTO [**<Zeilen-Nr.>** [**<Schrittweite>**]]

Als Folge dieses Kommandos wird automatisch nach jedem Drücken der Eingabe-Taste eine Zeilen-Nummer generiert. Als Standardwert wird für beide Parameter der Wert 10 angenommen.

Wird bei der Automatischen Zeilenummerierung eine Zeilen-Nr. angezeigt, unter der bereits eine Programmzeile im Speicher ist, so erscheint ein Stern * neben der Zeilen-Nr.

Soll diese im Speicher befindliche Zeile erhalten bleiben, drücken Sie die Eingabetaste unmittelbar nach dem Stern. Jede Eingabe überschreibt die im Speicher vorhandene Programmzeile.

Der AUTO-Modus wird durch Eingabe von **CTRL + C** oder **BREAK** abgeschlossen. Dabei wird die Zeile, in der die **CTRL + C** Taste eingegeben wurde, nicht abgespeichert. Nach dem Drücken von **CTRL + C** kehrt der Interpreter in den Kommando-Modus zurück.

Die Taste **BREAK** hat die gleiche Funktion wie **CTRL + C**.

LIST [**<Zeile 1>**[- **<Zeile 2>**]]

Dieses Kommando listet das im Speicher befindliche Programm auf dem Bildschirm auf.

Als Standardwerte werden für **Zeile 1** die erste und für **Zeile 2** die letzte Zeile angenommen. **Zeile 2** darf nur im Zusammenhang mit " - " angegeben werden. Wird als Parameter ein Punkt angegeben, so wird die zuletzt bearbeitete bzw. eingegebene Zeile angezeigt.

Das Auflisten kann mit **CTRL + S** unterbrochen werden. Das Betätigen einer beliebigen Taste setzt das Auflisten fort. **CTRL + C** oder **BREAK** brechen die Auflistung entgültig ab.

Nach der Auflistung kehrt BASIC in den Kommando-Modus zurück.

LLIST

Dieses Kommando wirkt wie der LIST-Befehl, die Anzeige der Programmzeilen erfolgt jedoch auf dem Drucker.

Voraussetzung zur Anwendung dieses Kommandos ist allerdings, daß der Drucker angeschlossen ist.

LLIST setzt für den Drucker 132 Zeichen pro Zeile voraus. Eine Änderung ist mit dem Befehl **WIDTH LPRINT** und Eingabe der Zeilenlänge möglich.

DELETE [**<Zeilen-Nr.>**][- **<Zeilen-Nr.>**]

Dieses Kommando dient zum Löschen von im Speicher befindlichen Programmzeilen.

Werden **Zeile 1** und **Zeile 2** eingegeben, so werden nur die dazwischen liegende Programmzeilen gelöscht.

Wird eine nicht existierende Zeilen-Nr. angegeben, so erscheint die Fehler-Meldung: **ILLEGAL FUNCTION CALL**

BASIC kehrt nach Ausführung des Kommandos immer in den Kommando-Modus zurück.

EDIT <Zeilen-Nr.>

Dieses Kommando erlaubt das Ändern oder Korrigieren einer Programmzeile, ohne die gesamte Zeile erneut einzugeben. Programmzeilen werden mit den nachfolgend erläuterten Folge-Kommandos modifiziert. Diese Folge-Kommandos werden nur im sog. EDIT-Modus wirksam und dienen folgenden Zwecken:

Folge-Kommandos:

L	Listen der Programmzeile
E	Beenden des EDIT-Modus
Q	Änderungen löschen und den EDIT-Modus beenden
A	Änderungen löschen und erneut den EDIT-Modus starten
X	Listen der Zeile und umschalten in den Einfüge-Modus
I	Umschalten in den Einfüge-Modus
H	Rest der Zeile löschen und umschalten in den Einfüge-Modus
n D	n Zeichen löschen
n C	n Zeichen ändern
n S c	Suchen von Zeichen
n K c	Suchen und löschen von Zeichen
n Leertaste	Cursor in der Zeile nach rechts bewegen
Eingabe-Taste	Änderungen anzeigen und den EDIT-Modus beenden
CTRL + A	Editieren der zuletzt bearbeiteten Zeile
ESC	Beenden des Einfügemodus und umschalten in den normalen EDIT-Modus

Wird **n** nicht angegeben, so wird stets der Wert **1** angenommen
Die nähere Erläuterung der Folge-Kommandos entnehmen Sie bitte dem BASIC-Handbuch.

RENUM [[<neue Nr.>] [, <alte Nr.>] , <Schrittweite>]]]

Diese Anweisung numeriert ein im Speicher befindliches Programm neu durch.

<neue Nr.> = die erste Zeilen-Nr. der neuen Nummerierung

<alte Nr.> = ab dieser Zeilen-Nr. der alten Nummerierung beginnt die Neu-Nummerierung

<Schrittweite> = um diesen Wert wird die Zeilen-Nr. jeweils hochgezählt

RENUM ändert alle Zeilennummern entsprechend der neuen Vorgabe mittels der o.a. Parameter.

Die bei **GOTO**, **GOSUB**, **THEN**, **ON...GOTO**, **ON...GOSUB** und **ERL** angegebenen Zeilen-Nummern-Referenzen werden ebenfalls gegen die neue Zeilennummerierung ausgetauscht.

Wird in einer Anweisung eine nicht existierende Zeilen-Nr. festgestellt, erscheint der Fehler:

Undefined line XXXX in YYYY

Die ungültige Zeilen-Nr. (XXXX) wird durch RENUM nicht verändert, jedoch die Zeilen-Nr., in der der Fehler auftrat (YYYY).

Ist der Wert <neue Nr.> nicht angegeben, wird stets 10 angenommen. Wird bei dem Wert <alte Nr.> nichts eingegeben, wird bei der kleinsten Zeilen-Nr. des Programms mit der neuen Nummerierung begonnen. Für <Schrittweite> wird als Standardwert stets 10 angenommen.

RENUM bringt die Fehlermeldung **Illegal function coll**, wenn Zeilen-Nummern größer als 65529 generiert werden, oder beispielsweise folgende Anwendung von RENUM vorgegeben wird:

RENUM 15,30

und nur die Programmzeile 10, 20 und 30 existieren.

TRON

TROFF

Mit diesen Befehlen besteht die Möglichkeit, den Programmablauf während einer Ausführung zu verfolgen. Dies erleichtert das Testen und die damit verbundene Fehlersuche erheblich. Jede Zeilen-Nr., die vom Programm durchlaufen wird, erscheint in Klammern auf dem Bildschirm.

Durch **TRON** wird diese Trace-Funktion eingeschaltet und durch **TROFF** wieder ausgeschaltet.

Auch das **NEW** Kommando eliminiert die Trace-Funktion.

Bei deutschen Zeichensatz treten an der Stelle der eckigen Klammern die Großbuchstaben **Ä** und **Ü**.

CONT

Dient zur Fortsetzung einer Programmausführung, die mit **CTRL + C**, **BREAK**, **STOP** oder **END** Anweisung unterbrochen wurde.

Die Fortsetzung des Programms erfolgt an derselben Stelle, bei der die Unterbrechung auftrat.

Während der Unterbrechung besteht die Möglichkeit, Variable im Direkt-Modus durch **PRINT** anzuzeigen oder zu verändern und anschließend mit **CONT** das Programm fortzusetzen.

CONT ist unwirksam,

- wenn das Programm während der Unterbrechung verändert worden ist,
- wenn es während einer **INPUT\$** - Anweisung abgebrochen wurde, oder
- wenn keine Zeile mehr folgt.

BASIC-Kommandos zum Abspeichern von Programmen auf Kassette

Der Programmspeicher im PC ist ein flüchtiger Speicher (Halbleiterspeicher). Die Programme gehen deshalb bei Netzausfall oder Ausschalten des PC verloren.

Um die Programme zu erhalten, können sie mit eine Kassettenrecorder auf Magnetbandkassetten gespeichert werden.

Im Bedarfsfall (Netzwiederkehr oder nach erneutem Einschalten des PC) können die Programme von der Magnetbandkassette wieder in den Arbeitsspeicher geladen werden.

Für das Arbeiten mit dem Kassettenrecorder werden folgende Kommandos benötigt:

SAVE

LOAD

Die Kommandos sind folgendermaßen zu interpretieren:

SAVE "Name"

Die BASIC-Anweisung **SAVE** dient dem Speichern von Programmen.

Sie gehen dabei wie folgt vor:

Das Magnetband ist auf den Anfang, auf einen freien oder auf einen nicht mehr benötigten Bereich zu spulen.

Nach der Eingabe: **SAVE "Name"** und betätigen der Eingabetaste erfolgen am Bildschirm einige Abfragen.

- I. Abfrage: **Remote Casette ?** **Y/N**
(fern Ein-/Ausschalten des Motors.)
- II. Abfrage: **Baudrate = 1200 Bd ?** **Y/N**
(das Speichern auf die Kasette kann mit einer Geschwindigkeit von 1200 Bd. (Y), oder von 600 Bd. (N) erfolgen.)
- III. Abfrage: **Press Record and Play OK ?** **Y/BREAK**

Ist das Programm auf der Kasette aufgezeichnet, so erscheint die Abfrage:

Casette Stop OK ? **Y/BREAK**

Wird die Taste **Y** betätigt, kehrt BASIC in Kommando-Modus zurück.

LOAD "Name"

Wollen Sie ein Programm von einer Magnetbandkassette laden, so müssen Sie das BASIC-Kommando **LOAD** verwenden. Beim Laden gehen Sie wie folgt vor:

Die Magnetbandkassette ist auf den Programmanfang zu spulen. Anschließend **LOAD "Name"** eingeben und Eingabetaste betätigen.

Am Bildschirm erfolgt die Abfrage:

Press Play OK ? Y/BREAK

Der Ladevorgang beginnt. Sind auf der Kassette mehrere Programme gespeichert, erscheint auf dem Bildschirm die Information:

Found "Name"

Ist das richtige Programm gefunden, so erscheint die Information:

Load "Name"

und das Programm wird in den Speicher geladen.

Ist der Ladevorgang korrekt beendet, erfolgt die Abfrage:

Cassett Stop OK ? Y/BREAK

Wird die **Y** Taste betätigt, so kehrt BASIC in Kommando Mode zurück.

Mit dem Kommando **RUN** kann das Programm gestartet werden

1.2 Zusammenfassung der DOS-Befehle.

Befehl	Bedeutung	Syntax
ASSIGN	Laufwerke zuweisen	ASSIGN (d=x)...
CHDIR	Ändert das aktuelle Inhaltsverzeichnis	CHDIR ((d:)Pfad) CD ((d:)Pfad)
CHKDSK	Prüft Disketten und gibt Statusbericht	CHKDSK (d:)(Dateiname)(F)(V)
CLS	Bildschirm löschen	CLS
COPY	Kopieren von Dateien	COPY (d:)(Pfad)Dateiname(.ext) (d:)(Pfad)Dateiname(.ext) COPY (d:)(Pfad)Dateiname(.ext) +(d:)(Pfad)Dateiname(.ext) (d:)(Pfad)Dateiname(.ext)
DATE	Datum rufen/ändern	DATE (tt.mm.jj) DATE (tt.mm.19jj) DATE (tt/mm/jj) DATE (tt-mm-jj)
DEL	Löschen von Dateien	DEL (d:)Dateiname(.ext) DEL (d:)Pfad ERASE (d:)Dateiname(.ext)
DIR	Inhaltsverzeichnisse auflisten	DIR (d:)(Pfad)(Dateiname)(.ext) (/P)(/W)
DISKCOPY	Disketten kopieren	DISKCOPY (d:) (d:)
FIND	Suchen einer Zeichenkette	FIND (/V)(/C)(/N) "Zeichenkette" (d:)(Pfad)Dateiname(d:) (Pfad)Dateiname
FORMAT	Formatieren einer Diskette	FORMAT (d:){/V}{/S}
MKDIR	Inhaltsverzeichnis einrichten	MKDIR Pfad((ØPfadname)Ø..) MD Pfad
MORE	Bildschirmseite ausgebenöMORE

MODE	Drucker-Modus setzen	MODE com1:baud,parität,datenbits, stoppbits(,p) MODE lpt#:druckbreite,zeilen/inch (,p) MODE lpt#:=com1:
PATH	Suchpfad für ausführbare Dateien	PATH (d:)(Pfad)(;)((d:)Pfad;...)
PRINT	Druckt Dateien aus einer Warteschlange	PRINT ((d:)Dateiname(.ext) (/T)(/C)(/P)....)
PROMPT	Ändern des Prompters	PROMPT (Prompter-Text)
RECOVER	Wiederherstellen von Dateien	RECOVER (d:)(Pfad)Dateiname(.ext) RECOVER d:
RENAME	Dateien umbenennen	RENAME (d:)(Pfad)Dateiname(.ext) Dateiname(.ext) REN (d:)(Pfad)Dateiname(ext) Dateiname(.ext)
RMDIR	Inhaltsverzeichnis löschen	RMDIR Pfad RD Pfad
SORT	Sortieren von Texten	Sort(/R)(/+n)<DateiöSort(/R)(/+n)
SYS	Übertragen von Systemdateien	SYS d:
TIME	Uhrzeit aufrufen/ändern	TIME (hh.mm.ss.xx)
TREE	Zeigt die Zugriffspfade und Dateien	TREE (d:)(/F)
TYPE	Anzeige des Dateiinhaltes	TYPE (d:)(Pfad)Dateiname(.ext)
VER	Versionsnummer abfragen	VER
VERIFY	Überprüft Daten nach dem Aufzeichnen.	VERIFY (on)(off)
VOL	Abfragen des Datenträgerkennsatzes	VOL (d:)

Inhaltsverzeichnis

1. Monitor
 - 1.1 Aufbau, Funktionsweise und Aufruf
 - 1.2 Befehlsvorat

2. Basic Interpreter
 - 2.1 Allgemeines zum BASIC - Interpreter
 - 2.2 BASIC - Kommandos

3. MS - DOS

4. UCSD-P System

5. Diagnose-Programme

1.1 Aufbau, Funktionsweise und Aufruf des MOS

Das MOS (Mikrocomputer- Operating- Systems) besteht aus einer Reihe von Programmen zur Abwicklung der MOS-Komandos.

Das MOS liest, analysiert und verarbeitet den JOB-Eingabe-Strom. Der JOB-Eingabestrom ist eine Folge von Anforderungen an Programmen sowie Dateien und Einheiten, welche von dieser verwendet werden.

Nach dem Betätigen des Netzschalters am Gerät und Durchlaufen der Einschalt routine erscheint am Bildschirm eine Abfrage ob mit BASIC, MONITOR oder TERMINAL gearbeitet werden soll. (Kalt Start)

Die zweite Möglichkeit zum Aufrufen der Abfrage besteht durch das Drücken der RESET-Taste (Warm Start)

Die Abfrage erscheint auf dem Bildschirm nach der Einschalt routine, wenn kein ROM - Pack vorhanden ist, und kein SCSI - Controller sich meldet.

Um ins BASIC - Programm zu gelangen muß die Funktionstaste **F1** gedrückt werden. Mit Break und Ok meldet sich das BASIC-Programm bereit. Das Programm kann mit dem Kommando **SYSTEM** verlassen werden.

MONITOR-Programm erreicht man durchs Betätigen der Funktionsteste **F2**

Das TERMINAL-Programm erreicht man durchs Betätigen der Funktionstaste **F3**
(Wird in Verbindung mit Autonome Prozessorplatte benötigt)

Am Bildschirm erscheint eine weitere Abfrage ob eine Datei in Speicher geladen oder aus dem Speicher Datei auf Kassette geschrieben werden soll. Weiterhin kann auch eine Datei gedruckt oder das Programm beendet werden.

BTX-Zuschaltung erfolgt durch Betätigen der Funktionstaste **F4**.

Die CPU 8088 kann bis 1 MB direkt adressieren. Sie besitzt einen 16-bit-Rechner, jedoch nur einen 8-bit-breiten Datenbus. Der Rechner holt sich mit dem entsprechenden Befehl automatisch 2 Byte (= ein Wort).

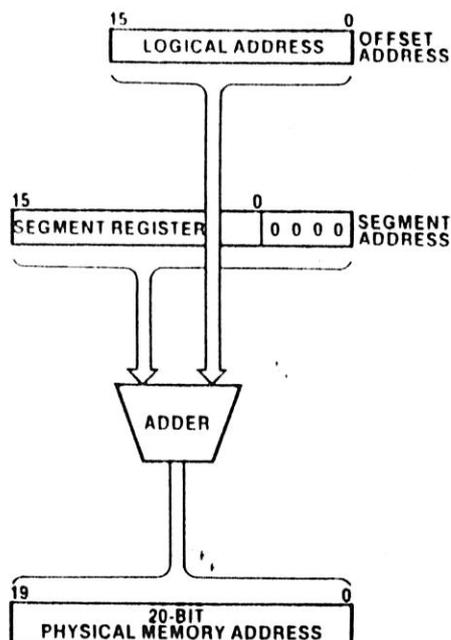
Ein Wort ist eine Folge von 2 Byte. Ein Wort steht so im Speicher, daß das höherwertige Byte auf der higher Memory Adresse steht.

Der Adreßbus ist 20 bit (2 1/2 Byte) breit. Das höchste Halb-Byte bezeichnet die sogenannte "Segment-Adresse". Der Speicher kann in beliebig viele Segmente eingeteilt werden. Ein Segment kann jedoch höchstens 64 KB groß sein. Somit ergibt sich die Speichergröße von $16 \times 64 = 1 \text{ MB}$.

Bit 0 - 15 ist die logische Adresse (Offset)

Bit 16 - 19 ist die Segment-Adresse (16 mögliche 64 KB Segmente)

Beispiel:



Eingabe: 1 0 0 0

Eingabe: 0 0

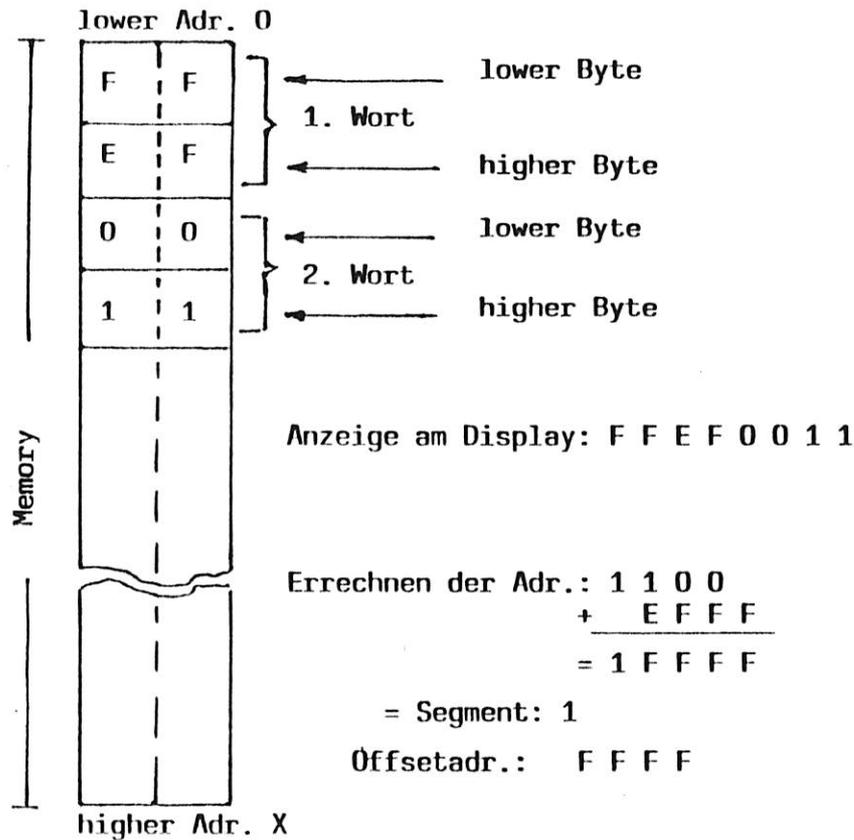
= Adresse: 0 1 0 0 0

Segment: 0 (erste 64 K)

Adresse: 1 0 0 0 (viertes K)

Erklärung zur Display-Anzeige von Speicheradressen

Die CPU beginnt mit der niedrigen Adresse eines Wortes (lower Adr.)



1.2 Befehlsvorrat

Durch Eingabe von zulässigen Befehlen wird dann das Programm veranlasst, bestimmte Aufgaben auszuführen.

Folgende Monitorfunktionen stehen über die Tastatur zur Verfügung:

D	Display	Speicherbereich anzeigen
E	End	Monitor Ende
F	Fill	Speicherbereich füllen
G	Goto	Programm starten ab Adresse
M	Move	Inhalt im Speicher verschieben
R	Register	Alle CPU- Register sichtbar machen
S	Subtitue	Speicherzellen ändern
T	Tracer	Progr. im Single - Step testen
U	Uhrzeit	Zeit im Speicher setzen
P	Printer	Drucker ein/aus

Alle Adresseingaben werden von dem Monitor gespeichert und falls keine Eingabe vorhanden als default Eingabe übernommen.

Die Eingabe geschieht über einen Zeilenbuffer und es ist somit eine Sofortkorrektur über die Backspace - Taste möglich.

Die Eingabe wird erst nach Betätigung der CR - Taste überprüft. Eine falsche Eingabe wird nochmals bis zu dem ersten falschen Zeichen ausgegeben.

Befehlsbeschreibung

Display

Mit dieser Funktion wird der Speicherinhalt byteweise in HEXA und ASCII angezeigt.

Die laufende Bildschirmausgabe kann mit den Tasten **CONTR** und **S** angehalten und mit **CONTR** und **Q** wieder gestartet werden.

Mit der Taste **BRACK** bzw. **CONTR** und **C** kann die Anzeige jederzeit abgebrochen werden.

Eingabe:

1. **D** und **CR**
Der Speicher wird ab der default - Adresse 16 Bildschirmzeilen ausgegeben.
2. **D** und **Startadresse**
Der Speicher gibt ab dem default-Segment und dem Offset (Startadesse) 16 Bildschirmzeilen aus.
3. **D** und **Segment: Startadresse**
Der Speicher gibt ab (Segment: Startadresse) 16 Bildschirmzeilen aus.
4. **D** und **, Endadresse**
Der Speicher gibt ab der default - Adresse bis zur Endadresse aus.
5. **D** und **Segment: Startadresse , Endadresse**
Der Speicher gibt über den angegebenen Adressbereich aus.

End

Mit dieser Eingabe wird zum aufrufenden Programm zurückgekehrt, wenn der Monitor über die Funktion **O** aufgerufen wurde.

Eingabe: **E** und **CR**

Fill

Der Speicher wird mit einer Konstanten gefüllt.

Eingabe:

1. **F** und **CR**
keine Funktion
2. **F** und **Startadresse, Endadresse, Füllwert**
Der Speicher wird vom default Segment: Startadresse bis Endadresse mit dem Wert (Füllwert) gefüllt.
3. **F** und **Segment: Adresse, Endadresse, Füllwert**
Der Speicher wird von (Segment. Adresse) bis (Endadresse) mit dem Wert (Füllwert) gefüllt.

Goto

Mit diesem Kommando kann ein im Speicher stehendes Programm bzw. Programmteile gestartet werden. Die mit dem **G** - Aufruf übergebenen Register können mit dem **R** - Kommando verändert bzw. angezeigt werden. Wird bei dem **G** - Aufruf eine Stop - Adresse, angegeben, so muß diese Stop - Adresse im RAM - Bereich liegen. Sonst sehe Trace - Kommando. Mit dem **G** - Aufruf können 2 Stopadressen übergeben werden.

Bei dem G - Kommando müssen alle Adressen eine Segment-information haben.

Eingabe:

1. **G und Segment: Adresse**
Es wird ein Programm ab der Adresse (Segment: Adresse) gestartet.
2. **G und Segment: Adr. , Stopsegment: Adr.**
Es wird ein Programm ab der Adresse (Segment: Adr.) gestartet. Wenn dieses Programm während seiner Ausführung zu der Adresse (Stopsegment: Adresse) kommt, so wird das Programm angehalten und in den Monitor verzweigt.
3. **G und Segment:Adr. , Stop 1 segment:Adr. , Stop 2 segment:Adr**
Es wird ein Programm ab der Adresse (Segment. Adr.) gestartet. Wenn dieses Programm während seiner Ausführung zu den Adressen (Stop 1 segment: Adr. oder Stop 2 segment:Adr.) kommt, so wird das Programm angehalten und in den Monitor verzweigt.

Inhaltsverzeichnis

Kapitel

- 1 MS-DOS
- 2 MS-DOS Befehle
- 3 EDITOR
- 4 Batch Verarbeitung
- 5 MS-DOS Initialisierung / File Allocation / Fehlermeldungen

- Inhaltsverzeichnis Kapitel 1

- 1.0.0 Allgemeines
- 1.0.1 Diskettenorganisation
- 1.0.2 Bestückung
- 1.0.3 Laden des Betriebssystems
- 1.0.4 Boot Sektor

- 1.1.0 Sicherung von Disketten
- 1.1.1 Laufwerksauswahl
- 1.1.2 Formatieren und kopieren
- 1.1.3 Hidden Files

- 1.2.0 Dateien
- 1.2.1 Dateinamen
- 1.2.2 Ersatzzeichen
- 1.2.3 Reservierte Namen
- 1.2.4 Inhaltsverzeichnisse

- 1.3.0 DOS-Kommandos
- 1.3.1 Interne und externe Kommandos
- 1.3.2 Kommando-Optionen
- 1.3.3 Batch-Ausführung
- 1.3.4 Input-Output

- 1.4.0 Filter
- 1.4.1 Beispiele

1.0.0

Allgemeines

MS-DOS ist die Abkürzung für " Mikro-Soft Disk Operating System ".

Das MS-DOS Betriebssystem besteht aus Kommandos und Programmen, die es dem Anwender ermöglichen Daten im Computer zu verwalten und eigene Software zu entwickeln. Durch das Laden des Betriebssystems ist der Rechner in der Lage, Kommandos vom Anwender entgegen zu nehmen. Dabei wird zwischen Internen- und Externen Befehlen unterschieden. Interne-Befehle sind Bestandteil des Command-Prozessors, der als ein Teil des Betriebssystems, in der Lage ist, diese direkt auszuführen. zB. COPY DATE TIME DEL REN DIR usw.

Bei Externen-Befehlen muß zuvor eine Datei von der Betriebssystemdiskette geladen werden. Diese Datei stellt ein in sich geschlossenes, meist längeres Programm dar, das eine bestimmte Aufgabe lösen kann. zB. FORMAT SORT DISKCOPY usw.

Durch diese Aufteilung wird nur ein kleiner Teil des Programmspeichers konstant vom Betriebssystem belegt. Nur bei Bedarf werden andere Dienstprogramme wie Basic-Interpreter oder Compiler dazu geladen.

Das freigegebene IO.SYS basiert noch größtenteils auf dem zugekauften Produkt, wobei IA-eigene Modifikationen in den Bereichen Videotreiber, Floppy-Disk-I/O und Druckerausgabe durchgeführt wurde. Das Disketten-Handling erfolgt noch mit sehr unbefriedigender Geschwindigkeit - dieser Mangel wird bei der nächsten Version weitgehend beseitigt sein.

Diskette/Platte im Laufwerk A: hat keinen Kennsatz
Katalog von A:Ö

COMMAND	COM	16613	8.03.85	10.21
FORMAT	COM	16663	14.03.85	7.53
DEBUG	COM	11764	1.02.83	10.13
		3 Datei(en)	327680 Bytes frei	

1.0.4

Boot Sektor

Auf unseren Maschinen können zwar die Systemdisketten des IBM-PC und der meisten kompatibelen "gebootet" werden - unsere Systemdisketten jedoch nicht auf diesen Rechnern. Der Grund liegt in unserem BOOT SEKTOR, welcher nicht ganz IBM kompatibel ist. Bei späteren Versionen kann dies jedoch geändert werden.

1.1.0

Sicherung von Disketten

Um sich davor zu schützen, daß Disketten oder Dateien unbeabsichtigt zerstört werden, ist es unerläßlich, folgende Punkte stets zu beachten.

1. Verriegeln der Disketten, erst nachdem der PC eingeschaltet wurde.
2. Vor dem Ausschalten des PC müssen die Disketten aus beiden Laufwerken entfernt werden.
3. Häufige Datensicherung durch Kopieren der Datendisketten.
4. Diskette, wenn nicht damit gearbeitet wird, aus dem Laufwerk entnehmen.
5. Möglichkeit des Schreibschutzes nützen. Die Diskette ist vor dem Überschreiben geschützt, wenn die Schreibschutzkerbe der Diskettenhülle überklebt wird.

1.1.1

Laufwerksauswahl

MS-DOS meldet sich nach dem Laden mit dem Prompt A> . Das bedeutet, daß der Command-processor auf Eingaben des Bedieners wartet. Das Laufwerk "A" ist als Standard-Laufwerk vorgemerkt. Wird kein anderes Laufwerk angegeben, beziehen sich alle Kommando-Ausführungen auf dieses Laufwerk. (Standard-Laufwerk = default device).

Auf die Eingabe von B: reagiert MS-DOS wie folgt:

```
A>b: <RETURN>
B>
```

Das bedeutet, daß nun das Laufwerk "B" als Standard-Laufwerk behandelt wird.

1.1.2

Formatieren und kopieren

Nach dem ersten Einschalten sollte als erstes die Betriebssystem-Diskette gesichert werden. Dazu wird der DOS-Befehl "DISKCOPY" verwendet.

```
A>DISKCOPY a: b:
```

Hierbei wird der gesamte Inhalt der Diskette im Laufwerk "B" übertragen. Bei fabrikneuen Disketten im Laufwerk "B" wird eine automatische Formatierung der Diskette vor dem Kopieren durchgeführt.

1.1.3

Hidden Files

Die beiden Systemdateien MSDOS.SYS und IO.SYS besitzen das Attribut "hidden" und tauchen nicht im Directory auf. Sie werden auch nicht bei "COPY*.*" mitkopiert, sondern können nur mit dem Kommando "SYS" auf eine andere Diskette gebracht werden. Das dritte Systemfile COMMAND.COM ist nicht hidden, da es jederzeit auch auf einer NICHT-SYSTEM-Diskette benötigt werden kann.

1.2.0

Dateien

Alle Programme, Texte oder sonstige Daten werden unter eindeutigen Namen als Datei, oder auch Files, gespeichert. (s. Punkt 1.2.4)

Die Dateien auf des Betriebssystems können folgendermaßen aufgeteilt werden:

- Original MS-DOS-Files von Mikrosotft (z.B. CHKDSK, DEBUG, SORT usw.).

1.2.1

Dateinamen

Der Dateiname kann bis zu 8 Zeichen lang sein und durch einen Punkt getrennt eine 3-stellige Erweiterung enthalten.

Beispiel: NEWFILE.EXE
Name 12345678.123

Für den Dateinamen und die Erweiterung können folgende Zeichen verwendet werden:

A-Z 0-9 \$ & % ' () - @ ^ { } ~ ` !
(kein Leerschritt)

Beispiel: A>DIR <RETRURN>

Die Dateinamen einer Arbeitsebene werden von der Diskette im Laufwerk "A" gelesen und auf dem Bildschirm angezeigt.

1.2.2

Ersatzzeichen (Global Parameter)

Die beiden Zeichen "?" und "*" werden immer dann angewandt, wenn mehrere ähnliche Dateinamen mit einem gemeinsamen Ausdruck angesprochen werden sollen.

Das Fragezeichen ersetzt ein beliebiges Zeichen innerhalb des Dateinamens oder der Dateierweiterung.

Beispiel: Es befinden sich folgende Dateien in einem Inhaltsverzeichnis:

```
TEST1RUN.EXE
TEST2RUN.EXE
TEST3RUN.EXE
```

Sie sollen mit einer Namensangabe aufgelistet werden.

```
A>DIR TEST?RUN.EXE
```

Der Stern kann ein beliebiges, die restlichen oder alle Zeichen in einem Dateinamen oder einer Dateierweiterung ersetzen.

Obiges Beispiel würde auch mit folgender Namensgebung aufgelistet werden.

```
A>DIR TEST*.EXE
A>DIR *.EXE
```

1.2.3

Reservierte Namen

Einige 3-stellige Namen dürfen für Dateinamen nicht verwendet werden, da sie für MS-DOS reserviert sind.

```
AUX  CON  LST  oder  PRN  NUL
```

1.2.4

Inhaltsverzeichnisse

Die Namen aller Dateien einer Diskette sind in einem Inhaltsverzeichnis auf der Diskette abgelegt. Zu dem Namen werden auch die Länge der Datei, ihre Lage auf der Diskette und das Datum der Erstellung bzw. der letzten Änderung abgespeichert.

In einem Inhaltsverzeichnis können bis zu 112 Dateinamen abgelegt werden.

MS-DOS erlaubt auch das Anlegen von Unterverzeichnissen, so daß der Aufbau der Inhaltsverzeichnisse eine hierarchische Struktur (Baumstruktur) erhält.

Beispiel:

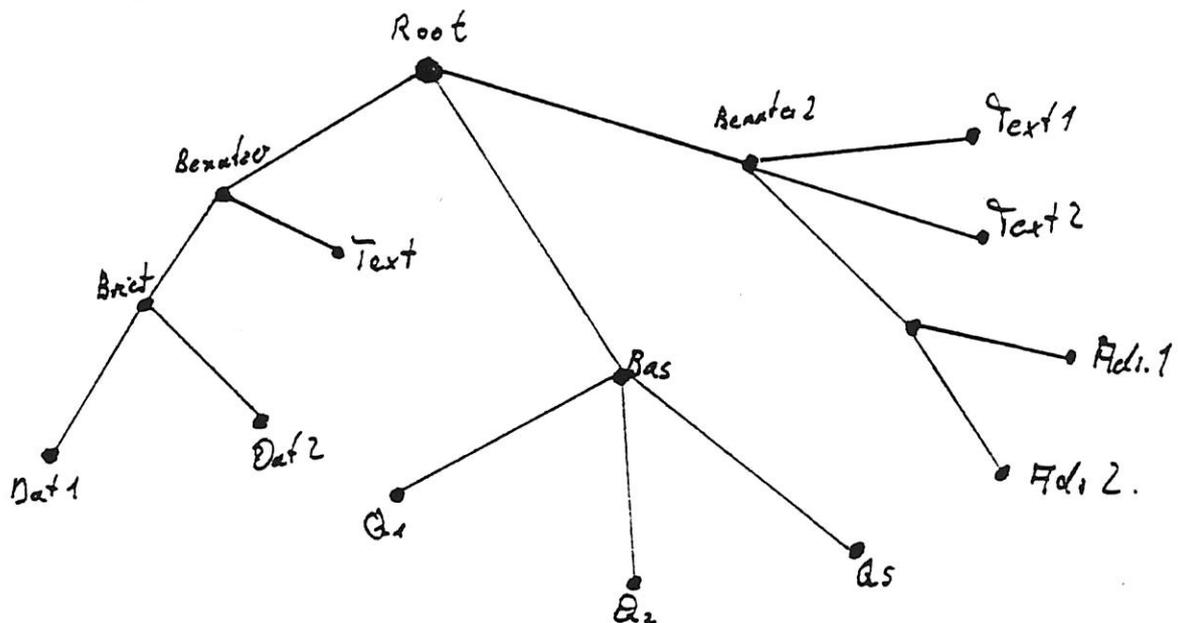
(Ö =)

Ebene	Inhaltsverz.	Dateien im Inhaltsverzeichnis
1	Stammverz. ---->	Datei.1, Datei.2, Datei.3,
	:	
	Ö	
	:--->	weitere Unterverz.
	v	
2	Unterverz.1 --Ö-->	Datei.1,
	:	
	Ö	
	:--->	weitere Unterverz.
	v	
3	Unterverz.2 --Ö-->	Datei.1, Datei2,
	:	
	Ö	
	:--->	weitere Unterverz.
	v	
4	Unterverz.3 --Ö-->	Datei.1, Datei.2, Datei3, ...

Jedes Inhaltsverzeichnis kann mehrere Unterverzeichnisse enthalten. Limitiert wird eine Baumstruktur nur durch die Größe des externen Speichers.

Auf das Anlegen und Löschen von Unterverzeichnissen wird in einem späteren Kapitel eingegangen.

Abb.: 1



1.3.0

DOS-Kommandos

- 1 Diese Kommandos dienen zum Erstellen , Vergleichen, Kopieren, Anzeigen, Löschen und Umbenennen von Dateien.
- 2 Zum Kopieren und Formatieren von Disketten.
- 3 Zum Ausführen von Systemprogrammen sowie auch Anwender-Programmen.
- 4 Zum Analysieren und Listen von Directories.
- 5 Zur Eingabe und Anzeige von Datum und Uhrzeit.
- 6 Zum Setzen verschiedener Printer- und Bildschirm-Optionen.
- 7 Zum Kopieren von Systemdateien auf eine andere Diskette.
- 8 Fordern MS-DOS auf, Programme über eine bestimmte Zeit zu unterbrechen.

1.3.1

Interne und externe Kommandos

Interne DOS-Kommandos werden beim Laden von DOS in den Speicher übernommen und stehen für ihren Aufruf unverzüglich zur Verfügung. Diese internen Kommandos werden beim Auflisten mit dem Kommando "DIR" nicht angezeigt. Zu den internen Kommandos zählen:

DIR	TIME	CHDIR
TYPE	EXIT	REM
COPY	VER	REN
VER	DATE	RMDIR
DEL	MKDIR	VERIFY

Externe DOS-Kommandos verbleiben auf der Diskette und werden erst bei Bedarf in den Speicher geladen. Externe Kommandos besitzen eine Namenserverweiterung die jedoch beim Aufruf nicht angegeben wird.

CHKDSK	MORE	EDLIN
DISKCOPY	PRINT	MASM
FIND	RECOVER	DEBUG
FORMAT		

1.3.2

Kommando-Optionen

Der Kommando-Aufruf besteht aus dem Kommando und zusätzlichen Parametern, die als Information für das System benötigt werden. Sind diese Parameter nicht angegeben, so nimmt das System Standardwerte an oder lehnt die Abarbeitung des Kommandos ab.

Beispiel: A>DIR

Kommando <-----I-I Kommando ohne Optionen. Das System nimmt Standardlaufwerk "A" an und listet den Inhalt der Diskette in "A" auf.

 A>DIR b:TEST.001 Kommando mit Optionen
Kommando<-I-I
Optionen<-----I-----I
Laufwerk<-----II
Dateiname<-----I--I

Erweiter. <-----I--I

Zu den Optionen zählen:

d:	Laufwerksangabe (a:,b:usw)	:Dateispezi-
filename	Dateiname	:fikation
.ext	Namenserweiterung (.EXE)	:(filespec)
pathname	Pfadnamen	
switches	Schalter (/p,/ /w)	
arguments	Argumente (on,off)	

Allgemeines zu den MS-DOS Kommandos:

- 1 Dem Kommando folgt gewöhnlich eine oder mehrere Optionen.
- 2 Die Eingaben können in großen, kleinen oder auch in gemischten Buchstaben erfolgen. Es wurden alle Namen in Großbuchstaben umgewandelt und abgelegt.
- 3 Kommandos und Optionen müssen mit Trennzeichen separiert werden. Neben Komma und Leerschritt können auch Semikolon, Gleichheitszeichen oder <TAB> verwendet werden.

-
- 4 Bei Dateispezifikationen werden nur Doppelpunkt oder Punkt zugelassen.
 - 5 Wird die Instruktion "Press any key" auf dem Bildschirm angezeigt, so kann eine beliebige Taste betätigt werden.
 - 6 Laufende Kommandos können mit <CONTROL-C> abgebrochen werden.
 - 7 Kommandos werden nur gestartet, wenn die Taste <RETURN> betätigt wird.
 - 8 Die Globalparamete "*" und "?" sowie reservierte Namen dürfen in Kommandonamen nicht verwendet werden.
 - 9 Die laufende Ausgabe auf den Bildschirm kann mit <CONTROL-S> gestoppt werden. Neustart mit einer beliebigen Taste.
 - 10 Die DOS Editorfunktionen können auch bei der Kommandoeingabe verwendet werden. (s. EDLIN)

1.3.3

Batch-Ausführung

Häufig benutzte gleiche Kommandosequenzen, die zeitraubende Eingaben erfordern, können in einer Batch-Datei zusammengefaßt werden. Durch Eingabe des Dateinamens werden alle Kommandos in der Batch-Datei sequentiell abgearbeitet, so, als würden sie einzeln eingegeben und mit der Taste <RETURN> aktiviert werden. Eine Batch-Datei muß die Erweiterung .BAT erhalten. Erstellt wird eine Batch-Datei über den Zeileneditor (EDLIN) oder mit Hilfe des COPY-Kommandos.

Gestartet wird der Batch durch Eingabe seines Dateinamens ohne seine Erweiterung .BAT.

Wird ein Batch unter dem Dateinamen "AUTOEXEC.BAT" angelegt, wird mit jedem Systemstart der Batch ausgeführt, sofern der Batch auf der Systemdiskette liegt.

Die Batch-Datei wird in einem der folgenden Kapitel ausführlich behandelt. (Kapitel 4)

1.3.4

Input-Output

Grundsätzlich nimmt DOS als Standard die Tastatur als Eingabe- und den Bildschirm als Ausgabe-Gerät an. Dieser Standard kann geändert werden.

Beispiel: A>DIR Befehlseingabe über die Tastatur, Ausgabe auf den Bildschirm.

Mit dem Zeichen ">" kann die Ausgabe umdirigiert werden. Beispielsweise in eine Datei.

A>DIR>TEST

Wenn eine Datei mit dem Namen Test auf der Diskette in Laufwerk "A" noch nicht vorhanden ist, wird die Datei angelegt und das Inhaltsverzeichnis der Diskette in diese Datei abgelegt. War die Datei TEST bereits vorhanden, so wird der alte Inhalt mit den neuen Daten überschrieben.

Sollen die neuen Daten an den alten Inhalt der Datei angehängt werden, so müssen zwei Zeichen ">>" angegeben werden:

A>DIR>>TEST

Soll eine Datei als Eingabe verwendet werden, wird das Zeichen "<" angewandt:

A>SORT<TEST>>LIST

Der Inhalt der Datei TEST wird alphabetisch geordnet an den an den Inhalt der Datei LIST angehängt.

1.4.0

Filter

Ein Filter ist ein Kommando, das Dateien lesen, sortieren, durchsuchen und wieder ausgeben kann.

DOS stellt folgende Filter zur Verfügung:

- FIND Durchsucht eine Datei nach einer vorgegebenen Zeichenfolge.
- MORE Gibt nur eine Bildschirmseite Daten aus. Die nächste Seite wird erst ausgegeben, wenn der Bediener eine beliebige Taste betätigt.
- SORT Sortiert Texte. Der sortierte Text kann auf dem Bildschirm ausgegeben oder in eine Datei abgelegt werden.

FIND /v/c/n zeichenfolge dateiname

durchsucht eine Datei nach vorgegebener Zeichenfolge.

/v zeigt alle Zeilen, die die Zeichenfolge nicht enthalten.

/c zeigt, wie oft die Zeichenfolge in der Datei gefunden wurde.

/n zeigt die relative Zeilennummer einer Datei, in der die Zeichenfolge gefunden wurde.

MORE füllt bei Ausgabe auf den Bildschirm nur eine Bildschirmseite auf einmal.

SORT/r/+n sortiert Textzeilen alphabetisch.
/r sortiert umgekehrt Z --> A.
/+n sortiert ab Teilung n.

1.4.1

Beispiele

1. dir b:
2. dir b:ösort
3. dir b:ösort/r
4. dir b:ösort/+14
5. dir b:ösort>b:sort.txt
6. type b:sort.txt
7. find "zur"b:edtast
8. find/c"zur"b:edtast
9. find n"zur"b:edtast
10. find/v"zur"b:edtast
11. type b:edtastömore

Inhaltsverzeichnis Kapitel 2

2.0.0		MS-DOS Befehle
2.0.1	CHDIR / CD	Wechsel / Anzeige des aktuellen Inhaltsverzeichnis
2.0.2	CHKDSK	Konsistenzüberprüfung des Inhaltsverzeichnis
2.0.3	CLS	Bildschirm löschen
2.0.4	COPY	Kopieren einer oder mehrerer Dateien
2.0.5	DATE	Aktuelles Datum
2.0.6	DEL	Löschen von Dateien
2.0.7	DIR	Inhaltsverzeichnis
2.0.8	DISKCOPY	Physikalische Kopie einer Diskette
2.0.9	FIND	Suchen eines Textstrings
2.1.0	FORMAT	Formatieren einer Diskette
2.1.1	MKDIR / MD	Einrichten eines neuen Inhaltsverzeichnisses
2.1.2	MORE	Bildschirmausgabe seitenweise
2.1.3	PATH	Einrichten eines Kommando-Pfades
2.1.4	PRINT	Hintergrunddruck
2.1.5	PROMPT	Abänderung des Prompters
2.1.6	RECOVER	Restaurierung einer Datei oder Diskette
2.1.7	REMARK/REM	Dient zur Kommentierung einer Batchdatei
2.1.8	RENAME/REN	Änderung von Dateinamen

-
- 2.1.9 RMDIR / RD Löschen eines Inhaltsverzeichnisses
 - 2.2.0 SORT Alphabetisches Sortieren von Textdateien
 - 2.2.1 SYS Übertragung der MS-DOS Systemdateien
 - 2.2.2 TIME Systemzeit lesen setzen
 - 2.2.3 VER Ausgabe der Versionsnummer
 - 2.2.4 VERIFY Ein- und Ausschalten des Verify-Modus
 - 2.2.5 VOL Zeigt den aktuellen Namen

 - 2.3.0 Beispiele zu den MS-DOS Befehlen

2.0.1

CHDIR / CD

Wechsel / Anzeige des aktuellen Inhaltsverzeichnisses.

Syntax: CHDIR (Pfad)
CD (Pfad)

Bei Angabe eines neuen Pfades wird das neue Inhaltsverzeichnis eingestellt. Wird kein Pfad angegeben, so wird der Name des aktuellen Inhaltsverzeichnisses gelistet.

Beispiel: CD ÜBINÖUSERÖMEYERÖLISTEN
führt in das Inhaltsverzeichnis LISTEN

CD..
führt in das nächst übergeordnete Inhaltsverzeichnis.

CD Ö
führt in das Hauptinhaltsverzeichnis.

CD zeigt das aktuelle Inhaltsverzeichnis.

2.0.2

CHKDSK

Konsistenzüberprüfung des Inhaltsverzeichnisses der angegebenen Diskette. Statusbericht.

Syntax: CHKDSK (d:) (Dateiname) (/F) (/V)

Wird ein Dateiname angegeben, so zeigt CHKDSK die Anzahl von nicht zusammenhängenden Bereichen, die von der Datei belegt werden.

/F: Fehler auf der Diskette werden korrigiert.
CHKDSK korrigiert nicht automatisch!

/V: Es werden ausführlichere Fehlermeldungen gemacht.

Die Bildschirmausgabe kann durch Redirection auf eine Datei geschrieben werden.

Beispiel: CHKDSK a: Disk A wird überprüft.
CHKDSK a:/v
Disk A wird überprüft, Status- und Fehlermeldungen werden auf Datei Bericht ausgegeben.

Fehlerbearbeitung durch CHKDSK:
Automatisch falls /F gesetzt:

Invalid drive specification	ungültige Laufwerks-Auswahl
Invalid parameter	ungültiger Parameter
Invalid sub-directory	ungültiges Inhaltsverzeichnis

Cannot CHDIR to (filename)	Dateiname nicht gefunden
First cluster nr. is invalid	Cluster Nr. ungültig
Allocation error, size adjusted	Fehler in der Datei-Belegungstabelle, Größe korrigiert

Has invalid cluster, file truncated	ungültige Clusterangabe, Datei abgeschnitten
Disk error reading FAT	Lesefehler in der Datei-Belegungstabelle
Disk error writing FAT	Schreibfehler in der Datei-belegungstabelle

Manuell

Invalid current directory	ungültiges, aktuelles Verzeichnis; RESTART und CHKDSK wiederholen
---------------------------	---

Cannot CHDIR to root	findet Hauptverzeichnis nicht; Versuchen Sie RESTART und das Kommando RECOVER
----------------------	---

(file) is cross linked on cluster	Datei ist doppelverzeigert gewünschte Datei kopieren, danach doppelverzeigerte Dateien löschen
-----------------------------------	--

Probable non-DOS disk	evtl. nicht DOS Diskette
-----------------------	--------------------------

2.0.3

CLS

Bildschirm löschen, keine Parameter

2.0.4

COPY

Kopieren von einer oder mehreren Dateien auf derselben Diskette, auf eine andere Diskette oder in ein anderes Directory.

Syntax: COPY Quelldatei Zieldatei (/V)
 Quell/Zieldatei = (d:) (Pfad) Dateiname
 COPY Quelle + Quelle2 Zieldatei

Wird die Zieldatei nicht angegeben, so wird in das aktuelle Directory kopiert, sofern es sich von dem Quelldirectory unterscheidet.

/V setzt den Verify-Mode beim Kopieren.

Für Quelldateien ist die Kennzeichnung von Dateigruppen mit ? oder mit * erlaubt. Ebenfalls können mit der Operation + Dateien zusammengefaßt werden.

Beispiel: COPY a:Datei.txt *)
 COPY a:Datei.txt b:
 COPY a:Datei.txt b:Datei2.txt
 COPY a:ÖBINÖMEYERÖDAT DATEI.txt
 COPY a:*.*)
 COPY a:*.LST b:LISTE
 COPY Datei+Datei2 Datei3

*) = nicht, wenn a default device

Soll direkt von der Tastatur in eine Datei geschrieben werden kann das Kommando:

COPY CON: <d> filename.ext

Die Eingabe wird mit Control Z abgeschlossen.

2.0.5

DATE

Mit diesem Kommando kann die aktuelle Datumseinstellung im System geändert werden. Das Datum wird jeweils bei Dateien eingetragen, die neu angelegt oder geändert werden.

Syntax: DATE (mm-dd-yy)

Wird nur DATE eingegeben, so wird das aktuelle Datum angezeigt und nach dem neuen Datum gefragt. Als Separatoren sind - oder / erlaubt.

Beispiel: DATE 7-16-84
DATE 7/16/84
DATE 07-16-1984

2.0.6

DEL

Löschen von einer oder mehreren Dateien.

Syntax: DEL (d:) (Pfad) Dateiname
ERASE (d:) (Pfad) Dateiname

Zum Löschen von Dateigruppen sind die Globalzeichen ? und * im Dateinamen erlaubt.

Beispiel: DEL a:ÖBINÖUSERÖDatei.txt
DEL Datei
DEL Datei.*
DEL a:*. *
ERASE Datei

2.0.7

DIR

Inhaltsverzeichnis insgesamt oder selektiert auflisten mit Angabe der Dateigröße und dem Erst- bzw. Update-Datum.

Syntax: DIR (d:) (Pfad) (Dateiname) (/P) (/W)

DIR allein listet das Inhaltsverzeichnis der Default-Disk. Bei weiteren Angaben wird selektiert gelistet. Für Dateigruppen sind die Globalzeichen ? und * m Dateinamen erlaubt.

/P setzt den Seitenmode (je 1 Bildschirmseite)

/W setzt breites Display.
Nur Dateinamen (5 per Zeile).

Beispiel: DIR
DIR a:ÖBINÖUSEÖDatei.txt
DIRa:ÖBINÖUSERÖ*.*
DIR*.com
DIR*.*

2.0.8

DISKCOPY

Physikalische Kopie einer Diskette.

Syntax: DISKCOPY (d:) (d:)

Die erste Option gibt die Quelldiskette an, die zweite die Zieldiskette. Sind beide Diskangaben gleich oder keine gemacht, so erfolgt eine Einlaufwerkkopie. Wird die zweite Diskangabe nicht gemacht, so wird auf die Default-Disk kopiert. Beide Disketten müssen die gleiche Zahl physikalischer Sektoren und gleicher Größe haben. Die Seitenzahl ermittelt DISKCOPY aus der Quelldiskette.

Fragmentierte Disketten werden besser durch COPY sequentiell reorganisiert.

Beispiel: DISKCOPY a: b:

2.0.9

FIND

Suchen eines Texttringes in einer oder mehreren Textdateien.

Syntax: FIND (/V /C /N) "String" (Datei)
Datei = (d:) (Pfad)Dateiname

FIND ist ein Filter und hat als Eingabeparameter einen String und eine Gruppe von Textdateien. Der String wird in den Dateien gesucht und alle zutreffenden Zeilen auf dem Bildschirm angezeigt. Ist keine Datei angegeben, so wird der Bildschirm durchsucht.

/V Es werden alle Zeilen gezeigt, die den String nicht enthalten.

/C Es wird nur die Anzahl der Zeilen bestimmt, die den String enthalten.

/N Jede Zeile erhält als Vorsatz die relative Zeilennummer in der Datei.

Beispiel: FIND "Juliet" RO&JU.txt LEAR.txt
DIR b:öFIND /V "DAT"

(ö =)
(siehe auch Kapitel)

2.1.0

FORMAT

Formatieren einer Diskette.

Syntax: FORMAT (d:)(/O)(/V)(/S)

Zusätzlich werden der Directory-Bereich und die FATs (File Allocation Tables) initialisiert. Fehlt die Diskangabe, so wird die Default-Diskette formatiert.

/V Nach der Formatierung wird der Disketten-Name (Volume-Label) abgefragt.

/S Die Systemdateien werden von der Quelldiskette zur Zieldiskette kopiert.

Beispiel: FORMAT b:/V/S

2.1.1

MKDIR / MD

Einrichten eines neuen Inhaltsverzeichnis.

Syntax: MKDIR Pfad
MD Pfad

Mit diesem Kommando werden neue Inhaltsverzeichnisse in der hierarchischen Verzeichnisstruktur eingerichtet.

Am Anfang eines Verzeichnisses stehen die Referenzdateien:

- . Referenz Arbeitsbereich
- .. Referenz nächst übergeordneter Arbeitsber.

Beispiel: MD BENUTZER
MD ÜBINÖUSERÖMEYER

2.1.2

MORE

Bildschirmausgabe seitenweise.

Syntax: MORE (keine Parameter)

MORE ist ein Filter, der die Standardausgabe seitenweise zum Bildschirm sendet. Ist eine Bildschirmseite angezeigt, so unterbricht die Ausgabe mit der Anfrage --MORE--. Nach (RETURN) wird die nächste Seite angezeigt, usw. bis zum Ende der Ausgabe. Dieser Filter empfiehlt sich bei der Einsicht in größere Textdateien.

Beispiel: TYPE LANGER.txtöMORE

(MORE benötigt Hilfsdateien. Schreibschutz beachten.)

2.1.3

PATH

Einrichten eines Kommando-Pfades.

Syntax: PATH (Pfad(Pfad(...)))

PATH setzt für MS-DOS den Pfad zu einem oder mehreren Inhaltsverzeichnissen. Diese werden von MS-DOS durchsucht, falls eine Datei oder ein externes Kommando im aktuellen Verzeichnis nicht gefunden wird. Defaultmäßig ist für MS-DOS kein Pfad gesetzt.

PATH ohne Optionen listet den aktuellen Pfad PATH; setzt den NULL-Pfad, d.h. die Defaulteinstellung. Nur das aktuelle Verzeichnis wird durchsucht.

Beispiel.: PATH ÜBINÖUSERÖMEYER
PATH ÜBINÖUSER;SYSTEM

(MS-DOS durchsucht nur nach Befehlen oder Stapel-
verarbeitungs-Dateien.)

2.1.4

PRINT

Hintergrunddruck

Syntax: PRINT ((Datei)(/T)(/C)(/S)...

PRINT veranlaßt das Ausdrucken im Hintergrund
während an der Konsole weiter gearbeitet werden
kann. Die Dateien werden in eine Warteschlange ge-
setzt.

/T Terminate:

Alle Daten in der Warteschlange werden ge-
löscht.

/C Cancel:

Die angegebenen Dateien werden aus der Warte-
schlange herausgenommen. Der Hintergrunddruck
wird bis zum nächsten /P unterdrückt.

/P Print:

Der Hintergrunddruck wird (wieder) aufgenommen.
Die verbliebenen Dateien werden wieder in die
Warteschlange gesetzt, die neuen Dateien werden
angehängt.

Beispiel: PRINT /T
 PRINT /T *.ASM

2.1.5

PROMPT

Abänderung des Prompters.

Syntax: PROMPT (Prompter-.ext)

PROMPT ohne Text setzt den Defaultprompter
(Defaultdisk). Der Text wird in den Prompter über-
nommen. Mit \$ werden die folgenden, speziellen
Funktionen eingeleitet.

\$ Der "\$" Character
t Die aktuelle Zeit
d Das aktuelle Datum
p Das aktuelle Verzeichnis auf der Def.-Disk
v Die Versionsnummer
n Die Defaultdisk
g Der ">" Character
l Der "<" Character
b Der " " Character
Die CR-LF Sequenz
s Vorangehender Leerplatz
h Backspace
e ESC vor nachfolgenden ASCII Code

Beispiel: PROMPT \$
PROMPT Zeit = \$t\$ Datum = \$d
PROMPT \$e 7m \$n:\$e m

2.1.6

RECOVER

Restaurierung einer Datei oder einer Disk mit defekten Sektoren.

Syntax: RECOVER Dateiname
RECOVER d:

RECOVER erkennt und kennzeichnet defekte Sektoren für weiteres, korrektes Arbeiten auf der Diskette. RECOVER Dateiname liest und schreibt die Datei unter Auslassung der schlechten Sektoren. RECOVER d: restauriert die gesamte Diskette einschließlich der Inhaltsverzeichnisse. (Reicht Root-Direcotry nicht aus, so werden extra Dateinamen in der FAT eingetragen. Sie können mit einem zweiten RECOVER restauriert werden, nachdem für Platz im Root-Directory gesorgt wurde.)

Beispiel: RECOVER b:
RECOVER a:test.txt

(Root-Directory=Stammverzeichnis)

2.1.7

REMARK / REM

Dient zur Kommentierung einer Batchdatei.

Syntax: REM (Bemerkungen)

Als Separatoren sind nur Leerplatz, Tabulator und Komma erlaubt.

Beispiel: REM Prüfung von Disketten

2.1.8

RENAME / REN

Änderung von Dateinamen

Syntax: REN (d:)(Pfad)Dateiname Dateiname

Die Datei im ersten Parameter erhält den Namen im zweiten Parameter.

Es wird nur im gleichen Inhaltsverzeichnis umbenannt. (d:), (Pfad) Spezifikationen werden im zweiten Parameter ignoriert.

Die Globalkennzeichen * und ? spezifizieren:
in der Quellangabe eine Dateigruppe
in der Zielangabe Buchstaben, die nicht geändert werden.

Beispiel: REN a:Datei.txt Datei2.txt
REN *.LST *.PRN
REN b:ABODE ?D?B?

2.1.9

RMDIR / RD

Löschen eines Inhaltsverzeichnisses

Syntax: RMDIR Pfad
RD Pfad

Nur ein leeres Inhaltsverzeichnis kann gelöscht werden. Unterverzeichnisse enthalten dabei noch die durch . und .. gekennzeichneten Referenzdateien.

Beispiel: RD ÜBINÜUSERÜJOE

2.2.0

SORT

Alphabetisches Sortieren von Textdateien

Syntax: SORT (/R) (/+n)

SORT ist ein Filter und wird zum Sortieren von Dateien benutzt. Die Dateiangaben erfolgen durch Redirection.

/R Kehrt die Sortierfolge um: Z-A
/+n Spezifiziert die Anfangsspalte für das Sortieren. Default ist Spalte 1.

Beispiel: SORT /R <Unsort.txt >Sort.txt
DIR öSORT/+14
DIR öSORT/+14 öMORE

(siehe auch Kapitel 6)

2.2.1

SYS

Übertragung der MS-DOS Systemdateien auf eine andere Diskette.

Syntax: SYS d:

SYS überträgt von der Defaultdiskette das MS-DOS-System auf die Zieldiskette d. Die Dateien werden in folgender Reihenfolge übertragen:

Als versteckte Dateien: IO.SYS
MSDOS.SYS
Als normale Datei: COMMAND.COM

Beispiel: SYS b:

2.2.2

TIME

Die Systemzeit lesen oder setzen.

Syntax: TIME (hh(:mm))

TIME ohne weitere Angaben zeigt die Uhrzeit am Bildschirm und fragt nach dem neuen Wert. TIME mit Zeitangabe setzt ohne weiteres Echo die Zeit.

Beispiel: TIME
TIME 12:00

2.2.3

TYPE

Ausgabe von Textdateien am Bildschirm

Syntax: TYPE (d:)(Pfad)Dateiname mit Erweiterung.

Nur sinnvoll für Textdateien. Mit CTRL-P kann parallel der Drucker betrieben werden.

Beispiel: TYPE b:TEXT.txt
TYPE b:Öuser10autoexec.bat

2.2.4

VER

Ausgabe der Versionsnummer von MS-DOS.

Syntax: VER

2.2.5

VERIFY

Ein- und Ausschalten des Verify-Modus für Dateitransfer.

Syntax: VERIFY (ON/OFF)

VERIFY ohne weitere Angabe zeigt den aktuellen Verify-Modus.

VERIFY ON setzt den Verify-Modus bis zum nächsten System-Reset oder dem Befehl VERIFY OFF

Beispiel: VERIFY ON
COPY Dateil Datei2

2.2.6

VOL

Zeigt den aktuellen Namen (Volume-Name) auf der spezifischen Diskette.

Syntax: VOL (d:)

Beispiel: VOL b:

2.3.0 Beispiele zu den MS-DOS Befehlen

Baumstruktur, Pfadnamen MKDIR, CHDIR, RMDIR.

DIR b: Inhalt des Stammverzeichnisses.

Volume in drive B is TESTDISK 1
Directory of B:Ö

IO	SYS	21632	7-12-84	2:13p
MSDOS	SYS	16768	7-12-84	2:13p
COMMAND	COM	15488	7-12-84	2:13p
EBENE2	<DIR>		7-23-84	9:45a
4 File(s)		294912 bytes free		

DIR b:Öebene2 Inhalt des 2.Unterverzeichnisses.

Volume in drive B is TESTDISK1
Directory of B:Öebene2

.	<DIR>		7-23-84	9:45a
..	<DIR>		7-23-84	9:45a
EBENE3	<DIR>		7-23-84	9:45a
ÜBUNG1		4	7-23-84	10:07a
4 File(s)		294912 bytes free		

DIR b:Öebene2Öebene3 Inhalt des 2.Unterverzeichnisses.

Volume in drive B is TESTDISK 1
Directory of B:Öebene2Öebene3

.	<DIR>		7-23-84	9:45a
..	<DIR>		7-23-84	9:45a
EBENE4	<DIR>		7-23-84	9:46a
ÜBUNG2		4	7-23-84	10:07a
4 File(s)		294912 bytes free		

DIR b:Öebene2Öebene3Öebene4 Inhalt des 3.Unterverzeichnisses.

Volume in drive B is TESTDISK 1
Directory of B:Öebene2Öebene3Öebene4

.	<DIR>		7-23-84	9:46a
..	<DIR>		7-23-84	9:46a
TEST1		23	7-11-84	12:04p
TEST2		23	7-11-84	12:05p
TEST3		23	7-11-84	12:06p
TEST4		23	7-11-84	12:24p
STAPEL	BAT	687	7-23-84	10:34a

START BAT 47 7-23-84 10:35a
 8 File(s) 294912 bytes free

CHKDSK

DIR auf Diskette mit zerstörtem FAT - Bereich. (36A=00).

Volume in drive B is TESTDISK 1
Directory of B:Ö

IO	SYS	21232	7-12-84	2:13p
MSDOS	SYS	16768	7-12-84	2:13p
COMMAND	COM	15488	7-12-84	2:13p
EBENE2	<DIR>		7-23-84	9:45a
4 File(s)		294912 bytes free		

CHKDSK b:/v

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:

B:ÖIO.SYS
B:ÖMSDOS.SYS
B:ÖCOMMAND.COM
B:ÖTESTDISK

Directory B:ÖEBENE2

Directory B:ÖEBENE2ÖEBENE3

Directory B:ÖEBENE2ÖEBENE3ÖEBENE4

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST1
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST2
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST3
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST4

Errors found, F parameter not specified.
Corrections will not be written to disk.

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST4 : werden bei An-
Has invalid cluster, file truncated. : gabe von /f
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST4 : korrigiert
Allocation error, size adjusted. :

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTAPEL.BAT

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTART.BAT

B:ÖEBENE2ÖEBENE3ÖÜBUNG2

B:ÖEBENE2ÖÜBUNG1

B:ÖEBENE2ÖÜBUNG1

Has invalid cluster, file truncated.

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
65536 bytes in 11 user files
293888 bytes available on disk

CHKDSK b:/v/f

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:

B:ÖIO.SYS
B:ÖMSDOS.SYS
B:ÖCOMMAND.COM
B:ÖTESTDISK

Directory B:ÖEBENE2

Directory B:ÖEBENE2ÖEBENE3

Directory B:ÖEBENE2ÖEBENE3ÖEBENE4

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST1
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST2
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST3
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST4
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTAPEL.BAT
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTART.BAT
B:ÖEBENE2ÖEBENE3ÖÜBUNG2
B:ÖEBENE2ÖÜBUNG1

B:ÖEBENE2ÖÜBUNG1

Has invalid cluster, file truncated.

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
65536 bytes in 11 user files
293888 bytes available on disk

262144 bytes total memory
223616 bytes free

CHKDSK b:/v

Kontrolle.

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:

B:ÖIO.SYS
B:ÖMSDOS.SYS
B:ÖCOMMAND.COM
B:ÖTESTDISK

Directory B:ÖEBENE2

Directory B:ÖEBENE2ÖEBENE3

Directory B:ÖEBENE2ÖEBENE3ÖEBENE4

B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST1
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST2
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST3
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖTEST4
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTAPEL.BAT
B:ÖEBENE2ÖEBENE3ÖEBENE4ÖSTART.BAT
B:ÖEBENE2ÖEBENE3ÖÜBUNG2
B:ÖEBENE2ÖÜBUNG1

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
65536 bytes in 11 user files
293888 bytes available on disk

262144 bytes total memory
223616 bytes free

DIR auf Diskette mit fehlerhaften Stammverzeichnis.(B5A=00)

Volume in drive B is TESTDISK 1
Directory of B:0

IO	SYS	21632	7-12-84	2:13p
MSDOS	SYS	16768	7-12-84	2:13p
COMMAND	COM	15488	7-12-84	2:13p
EBENE2	<DIR>		7-23-84	9:45a
		4 File(s)	294912 bytes free	

CHKDSK b:/v

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:
B:0IO.SYS
B:0MSDOS.SYS
B:0COMMAND.COM

Errors found, F parameter not specified.
Corrections will not be written to disk.

B:0COMMAND.COM
Allocation error, size adjusted.
B:0TESTDISK
Directory B:0EBENE2
Directory B:0EBENE20EBENE3
Directory B:0EBENE20EBENE30EBENE4
B:0EBENE20EBENE30EBENE40TEST1
B:0EBENE20EBENE30EBENE40TEST2
B:0EBENE20EBENE30EBENE40TEST3
B:0EBENE20EBENE30EBENE40TEST4
B:0EBENE20EBENE30EBENE40STAPEL.BAT
B:0EBENE20EBENE30EBENE40START.BAT
B:0EBENE20EBENE300UBUNG2
B:0EBENE200UBUNG1

16 lost clusters found in 1 chains. : mit "n" wird
Convert lost chains to files (Y/N)? n : Datei Command.com
16384 bytes disk space : nicht gerettet
would be freed. :

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
48128 bytes in 11 user files
294912 bytes available on disk

262144 bytes total memory
223616 bytes free

CHKDSK b:/v/f

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:

B:ØIO.SYS
B:ØMSDOS.SYS
B:ØCOMMAND.COM

B:ØCOMMAND.COM

Allocation error, size adjusted.

: Datei Command.com
: wird auf "0" gesetzt

B:ØTESTDISK

Directory B:ØEBENE2

Directory B:ØEBENE2ØEBENE3

Directory B:ØEBENE2ØEBENE3ØEBENE4

B:ØEBENE2ØEBENE3ØEBENE4ØTEST1
B:ØEBENE2ØEBENE3ØEBENE4ØTEST2
B:ØEBENE2ØEBENE3ØEBENE4ØTEST3
B:ØEBENE2ØEBENE3ØEBENE4ØTEST4
B:ØEBENE2ØEBENE3ØEBENE4ØSTAPEL.BAT
B:ØEBENE2ØEBENE3ØEBENE4ØSTART.BAT
B:ØEBENE2ØEBENE3ØÜBUNG2
B:ØEBENE2ØÜBUNG1

16 lost clusters found in 1 chains.
Convert lost chains to files (Y/N)? y

: die Cluster wer-
: den in neue Datei
: übernommen

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
48128 bytes in 11 user files
16384 bytes in 1 recovered files
294912 bytes available on disk

: gerettete Datei

262144 bytes total memory
223616 bytes free

DIR auf restaurierte Diskette.

Volume in drive B is TESTDISK 1
Directory of B:0

IO	SYS	21632	7-12-84	2:13p	
MSDOS	SYS	16768	7-12-84	2:13p	
COMMAND	COM	0	7-12-84	2:13p	: gerettete Datei
FILE0000	CHK	16384			:
EBENE2	<DIR>		7-23-84	9:45a	
	5 File(s)		294912 bytes	free	

CHKDSK b:/v

Kontrolle.

Volume TESTDISK 1 created Jul 11, 1984 12:03p

Directory B:

- B:0IO.SYS
- B:0MSDOS.SYS
- B:0COMMAND.COM
- B:0TESTDISK
- B:0FILE0000.CHK

Directory B:0EBENE2

Directory B:0EBENE20EBENE3

Directory B:0EBENE20EBENE30EBENE4

- B:0EBENE20EBENE30EBENE40TEST1
- B:0EBENE20EBENE30EBENE40TEST2
- B:0EBENE20EBENE30EBENE40TEST3
- B:0EBENE20EBENE30EBENE40TEST4
- B:0EBENE20EBENE30EBENE40STAPEL.BAT
- B:0EBENE20EBENE30EBENE40START.BAT
- B:0EBENE20EBENE30ÜBUNG2
- B:0EBENE20ÜBUNG1

362496 bytes total disk space
0 bytes in 1 hidden files
3072 bytes in 3 directories
64512 bytes in 12 user files
294912 bytes available on disk

262144 bytes total memory
223616 bytes free

RECOVER

DIR b:

sämtliche Dateien wurden gerettet

Volume in drive B has no label
Directory of B:0

FILE0001	REC	22528	7-25-84	3:07p
FILE0002	REC	17408	7-25-84	3:07p
FILE0003	REC	16384	7-25-84	3:07p
FILE0004	REC	1024	7-25-84	3:07p
FILE0005	REC	1024	7-25-84	3:07p
FILE0006	REC	1024	7-25-84	3:07p
FILE0007	REC	1024	7-25-84	3:07p
FILE0008	REC	1024	7-25-84	3:07p
FILE0009	REC	1024	7-25-84	3:07p
FILE0010	REC	1024	7-25-84	3:07p
FILE0011	REC	1024	7-25-84	3:07p
FILE0012	REC	1024	7-25-84	3:07p
FILE0013	REC	1024	7-25-84	3:07p
FILE0014	REC	1024	7-25-84	3:07p

14 File(s) 294912 bytes free

Beispiel:

```
A> CD MAKEPR          MAKEPR ist jetzt
                        Arbeitsdirectory
A> DIR                Inhalt von MAKEPR ausgeben
```

Auf dem Monitor erscheinen jetzt alle Files die in dieser Directory abgelegt sind.

```
  .      <DIR>          8-09-82      10:09a
  ..     <DIR>          8-09-82      10:09a
```

Die Files " ." und " .. " werden automatisch vom Betriebssystem erzeugt. Dort sind die Namen der Arbeitsdirectory unter " ." und der nächst höheren Directory " .. " abgelegt. Durch die Eingabe von DIR . bzw. DIR .. werden diese entsprechend ausgegeben.

Hinwies: Die Root-Directory ist über den Buchstaben ö anzusprechen. zB. DIR ö

Beispiel:

In das Directory MAKEPR sollen alle zur Programmerstellung in BASIC wichtigen Teile zusammengefasst werden wie zb.:

```
BASCOM.COM
BASCOM.LIB
BASEXE.EXE
LINK.EXE
usw.
```

Es können alle Hilfen die auch im CP/M bekannt sind angewendet werden.

```
A> COPY BASCOM.* MAKEPR
oder
A> COPY BAS???.* MAKEPR
usw.
```

INHALTSVERZEICHNIS Kapitel 3

- 3.0.0 Der Zeileneditor
- 3.0.1 Die EDLIN-Befehle
- 3.0.2 Zusammenfassung der EDLIN-Befehle
- 3.0.3 Beispiele

3.0.0

Der Zeileneditor

Aufruf: edlin d:pfad dateiname.erw/b

Wenn der Parameter /b nicht angegeben wird, so endet das Einlesen der Datei in den Speicher beim ersten CTRL-Z. Mit /b wird die gesamte Datei eingelesen.

3.0.1

Die EDLIN-Befehle

Zeilen anfügen

nA

n Zeilen werden von der Diskette gelesen und an den in dem Hauptspeicher befindlichen Text angehängt. Höchstens aber soviel, bis 75 % des zur Verfügung stehenden Platzes im Hauptspeicher gefüllt sind. Nach dem Einladen der letzten Zeile aus der Datei kommt die Nachricht: End of input file.

Zeilen kopieren

zeile,zeile,zeile,wiederholzahl C

1. u. 2. Parameter = Zeile von bis
3. Parameter = Kopie vor Zeile in diesem Parameter
4. Parameter = so oft soll die Operation wiederholt werden.

Zeilen löschen

zeile,zeile D von Zeile bis Zeile löschen

Zeile editieren

zeile die Zeile wird mit der Zeilennummer auf dem Bildschirm angezeigt. Auf der darunter liegenden Zeile erscheint die Zeilennummer nochmals. Jetzt kann, wie bereits im Kapitel

"Editiertasten" beschrieben, die Zeile editiert werden.

Editieren beenden

E Abspeichern der editierten Datei. Die Originaldatei erhält die Erweiterung .BAK.

Zeilen einfügen

zeile I es können eine oder mehrere Zeilen vor die angegebene Zeile eingefügt werden. Mit <ctrl>+c kann der Einfügemodus verlassen werden.

Zeilen auflisten

zeile,zeile L der im Speicher befindliche Text wird von Zeile bis Zeile aufgelistet.
 L ohne Zeilennummer werden +10 / -10 Zeilen ab der aktuellen Zeile angezeigt

Zeilen verschieben

zeile,zeile,zeile M der Textblock vom 1. bis 2. Parameter wird vor den 3. Parameter verschoben. Fehlt der 1. Parameter, wird von der laufenden Zeile bis zum 2. Parameter vor den 3. Parameter verschoben. Die Zeilen werden neu durchnumeriert.

Seiten auflisten

zeile,zeile P es wird jeweils nur seitenweise auf den Bildschirm aufgelistet.

Abbrechen der Editierung

Q die Editierung wird ohne Übernahme der Änderungen abgebrochen.

Text ersetzen

zeile,zeile ? R zeichenfolge<F6>zeichenfolge
der angegebene Zeilenbereich wird auf die erste Zeichenfolge durchsucht und die aufgefundenen Stellen durch die zweite Zeichenfolge ersetzt.
Der Parameter ? stoppt an jeder gefundenen Zeichenfolge und wartet auf die Bestätigung, daß geändert werden soll.

Text suchen

zeile,zeile ? S zeichenfolge
der angegebene Zeilenbereich wird auf die angegebene Zeichenfolge durchsucht.

Zeilen übertragen

zeile T d: dateiname
der Inhalt der angegebenen Datei wird vor die angegebene Zeile eingefügt.

Zeilen speichern

nw
die angegebene Anzahl von Zeilen werden auf die Diskette gespeichert. Dieser Befehl wird nur bei großen Dateien verwendet.

3.0.2

Zusammenfassung der EDLIN-Befehle

Aufruf:

A> EDLIN <FILENAME>

nA	Zeile anfügen
n,n1,n2,n3C	Zeile kopieren
n,n1D	" löschen
E	Editor beenden
nI	Zeile einfügen
n,n1L	" listen
n,n1,n2M	" verschieben
n,n1P	Seiten auflisten
Q	Abbrechen
n,n1?R<string>F6<string>	Text ersetzen / ? stoppt bei Zeichenfolge
n,n1?S<string>	Text suchen
nTd:<filename>	Datei hinter Zeile n
nW	n Zeile in Datei

- n	Zeilennummer
- n,n1	Zeile von n bis n1
- n,n1,n2	Zeile von n bis n1 hinter n2
- n,n1,n2,n3	n3 mal wiederholen

3.0.3

Beispiele

End of input file

*1,6l

- 1: *Diese Datei
- 2: wurde erstellt
- 3: um Editor - Befehle
- 4: durch praktische
- 5: Übungen besser
- 6: kennenzulernen.

*1,6,1,2c

*1,20l

- 1: *Diese Datei
- 2: wurde erstellt
- 3: um Editor - Befehle
- 4: durch praktische
- 5: Übungen besser
- 6: kennenzulernen.
- 7: Diese Datei
- 8: wurde erstellt
- 9: um Editor - Befehle
- 10: durch praktische
- 11: Übungen besser
- 12: kennenzulernen.
- 13: Diese Datei
- 14: wurde erstellt
- 15: um Editor - Befehle
- 16: durch praktische
- 17: Übungen besser
- 18: kennenzulernen.

*7,18d

*1,20l

- 1: Diese Datei
- 2: wurde erstellt
- 3: um Editor - Befehle
- 4: durch praktische
- 5: Übungen besser
- 6: kennenzulernen.

*4

- 4: *durch praktische
- 4: *^C

*1,6l

- 1: Diese Datei
- 2: wurde erstellt
- 3: um Editor - Befehle
- 4: *durch praktische
- 5: Übungen besser
- 6: kennenzulernen.

*-2,+11
2: wurde erstellt
3: um Editor - Befehle
4:*durch praktische
5: Übungen besser

*4,5,1m
*1,61
1:*durch praktische
2: Übungen besser
3: Diese Datei
4: wurde erstellt
5: um Editor - Befehle
6: kennenzulernen.

*1,2,6m
*1,61
1: Diese Datei
2: wurde erstellt
3: um Editor - Befehle
4:*durch praktische
5: Übungen besser
6: kennenzulernen.

*1,6?rEditor^ZEd.
3: um Ed. - Befehle
O.K.? j
*1,61
1: Diese Datei
2: wurde erstellt
3: um Editor - Befehle
4:*durch praktische
5: Übungen besser
6: kennenzulernen.

*1,6?rEditor^ZEd.
3: um Ed. - Befehle
O.K.?
*1,61
1: Diese Datei
2:*wurde erstellt
3: um Ed. - Befehle
4: durch praktische
5: Übungen besser
6: kennenzulernen.

*1,6rEd.^ZEditor
3: um Editor - Befehle
*1,61
1: Diese Datei
2:*wurde erstellt
3: um Editor - Befehle
4: durch praktische
5: Übungen besser
6: kennenzulernen.

*1,6?sle
3: um Editor - Befehle
O.K.? j
*1,6?sle
3:*um Editor - Befehle
O.K.? j
6: kennenzulernen.
O.K.? j
Not found

*3tb:edbei
*1,121
1: Diese Datei
2: wurde erstellt
3:*Diese Datei
4: wurde erstellt
5: um Editor - Befehle
6: durch praktische
7: Übungen besser
8: kennenzulernen.
9: um Editor - Befehle
10: durch praktische
11: Übungen besser
12: kennenzulernen.

*3,8d
*1,61
1: Diese Datei
2: wurde erstellt
3:*um Editor - Befehle
4: durch praktische
5: Übungen besser
6: kennenzulernen.
*q -
Abort edit (Y/N)? y

INHALTSVERZEICHNIS Kapitel 4

- 4.0.0 Die Stapelverarbeitungs-Datei
- 4.1.0 Die Stapelverarbeitungs-Befehle
 - 4.1.1 Anlegen einer Batch Datei
 - 4.1.2 Beispiele
- 4.2.0 Autoexec.Bat

4.0.0

Die Stapelverarbeitungs-Datei

Eine Stapelverarbeitungsrechner- oder Batch-Datei muß immer die Erweiterung .BAT aufweisen. Sie enthält einen oder mehrere Befehle, die von DOS nacheinander ausgeführt werden.

Das Erstellen einer Batch-Datei kann über:

A> copy con:d:dateiname.bat

oder im

A> edlin d:dateiname.bat

durchgeführt werden.

Wird eine Batch-Datei unter dem Namen AUTOEXEC.BAT auf der Diskette, mit der DOS geladen wird, abgelegt, so wird beim Starten von DOS der Batch automatisch ausgeführt.

4.1.0

Die Stapelverarbeitungs-Befehle

ECHO on die Befehle in der Batch-Datei werden in der Reihe ihrer Verarbeitung auf dem Bildschirm angezeigt.

ECHO off stoppt die Anzeige der Befehle auf dem Bildschirm.

ECHO nachricht eine Nachricht wird unabhängig vom on/off-Status angezeigt.

FOR %%variable IN (satz) DO befehl
der hinter DO angegebene Befehl wird als Variable auf jedes Element im (satz) gesetzt.

GOTO sprungziel die Fortsetzung des Batches erfolgt auf der Zeile nach dem Sprungziel. Das Sprungziel wird mit einem : gekennzeichnet. (z.B. :LOOP)

IF bedingung befehl bedingte Ausführung von DOS Befehlen.

NOT bedingung befehl wie oben, nur als IF NOT.

SHIFT alle Parameter einer Befehlszeile werden um eine Stelle nach links verschoben.

PAUSE bemerkung die Systemverarbeitung wird unterbrochen. Auf dem Bildschirm wird -bemerkung- und die Nachricht "Wenn bereit, eine beliebige Taste betätigen" ausgegeben.

REM bemerkung Bemerkungen werden auf den Bildschirm ausgegeben.

4.1.1 Anlegen einer Batch Datei

Eine Batch-Datei ermöglicht es, eine Folge von Programmen und Betriebssystemkommandos zu programmieren. Dabei wird in einer Batch-Datei, pro Zeile eine Anweisung geschrieben und später dann auch so ausgeführt. Parameter werden verschlüsselt durch die Ziffern 0-9, miteinander führenden % Zeichen angegeben. Eine Batch-Routine kann grundsätzlich mit Control C abgebrochen werden.

Anlegen einer Batch-Datei:

>A COPY CON filename.BAT Jetzt kann zeilenweise die batch-Prozedur eingegeben werden.

Beispiel:

PROMPT \$\$	Prompt wechseln
CD MAKEPR	Directory MAKEPR anwählen
EDLIN %1.BAS	% entspricht dem filename des zu erstellenden Programms
BASCOM %1.BAS	Compiler aufrufen
PAUSE	Die Routine wird an dieser Stelle unterbrochen und kann durch drücken einer Taste fortgesetzt werden.
LINK %1.OBJ	Linker aufrufen
PAUSE	
%.EXE	Das erstellte Basic-Programm wird aufgerufen.

PROMPT
CD 0

Prompt wechseln
Zurück in die Root-Directory

Mit < Control Z > und return wird die Eingabe beendet.

Aurufen der Batch-Datei:

A> filename.BAT filename

 %1----- Name des Programms
 %0----- Name der Batch-Datei

Hinweis: Der Parameter %0 entspricht dem Namen der
Batch-Datei.

4.1.1

Beispiele

```
echo on
dir b:/w
pause -jetzt folgt der gleiche Vorgang, aber mit "ECHO OFF".
echo off
dir b:/w
echo -Diese Nachricht wird unabhängig von "ECHO OFF" angezeigt.
echo on
pause -ECHO ON ist wieder eingeschaltet. Nun folgt ein FOR Befehl.
for %%k in (b:test1 b:test2 b:test3) do type %%k
rem -Diese Bemerkung wurde mit "REM" auf den Bildschirm ausgegeben.
:loop
if %1==b:test1 echo test1 ist wahr!
pause -Nun folgen die Befehle "SHIFT" und "GO TO".
if %1==b:test4 copy %1 a:
dir a:/w
pause
if %1==b:test4 goto pause
pause
rem -Anzeige der drei Konstanten.
echo %1 %2 %3
shift
echo %1 %2 %3
shift
echo %1 %2 %3
shift
echo %1 %2 %3
pause
goto loop
:pause
del test4
dir/w
cd b:Ø
rem -Ende des Batchdurchlaufes.
```

4.2.0

AUTOEXEC.BAT

Direkt nach dem Systemstart kann über AUTOEXEC.BAT eine beliebige Folge von Systemkommando- und Programmaufrufen gestartet werden. Die AUTOEXEC.BAT Datei wird in die Rootdirectory abgelegt durch:

A> COPY CON: AUTOEXEC.BAT

Lege folgende Kommandos in die AUTOEXEC.BAT Datei ab.

Beispiel:

C:	Wechsele auf Laufwerk C
CD MAKEPR	Gehe in die Directory MAKEPR
BASIC86	BASIC86 Interpreter aufrufen

Mit < Control Z > und return abschließen.

In diesem Beispiel wird direkt nach dem MS-DOS geladen worden ist, auf das Laufwerk B umgeschaltet, dann die Directory MAKEPR aufgerufen und der BASIC86 Interpreter geladen.

Hinweis: Bei AUTOEXEC.BAT entfällt die Eingabemöglichkeit Datum und Zeit, nach dem Laden des Betriebssystems, im Speicher zu ändern. Dies kann durch hinzufügen von den Kommandos DATE und TIME in die AUTOEXEC.BAT Datei wieder mit eingefügt werden.

Inhaltsverzeichnis Kapitel 5

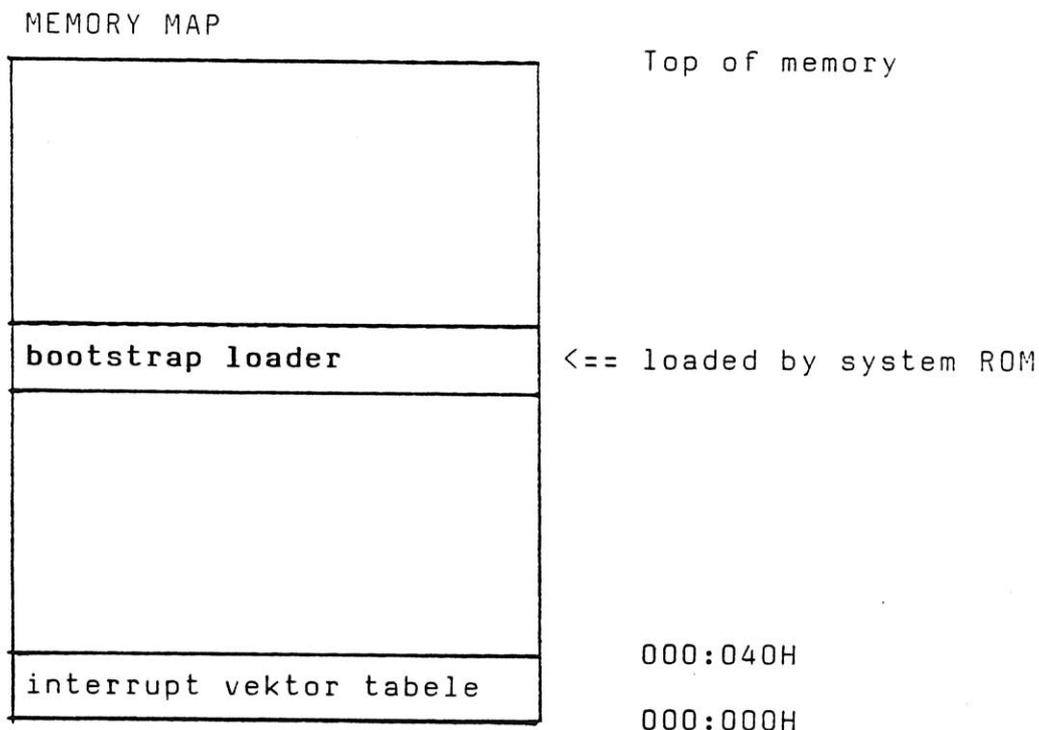
- 5.0.0 MS-DOS Initialisierung
- 5.1.0 MS-DOS File Allocation Tabel
- 5.2.0 Fehlermeldung

5.0.0

MS-DOS Initialisierung

Im unterschied zum CP/M, das in einen Ladevorgang das gesamte Betriebssystem in einen fest definierten Speicherbereich ablegt, vollzieht sich das Initialisieren von MS-DOS in mehreren Schritten.

In einen ROM ist ein kleines Programm abgelegt, das veranlasst, den sogenannten "boot sector" von der Diskette zu laden und beliebig irgendwo in den Speicher zu schreiben. Dieser Teil wird "bootstrap loader" genannt weil er den ersten Einstieg ins System ermöglicht. Außerdem muß zu diesem Zeitpunkt auch ein Teil der "interrupt vector table" initialisiert werden. (von 000:000H bis 000:040H) Für das MS-DOS Betriebssystem sind die Interrupts von 020H bis 03FH reserviert. Die Speicherbelegung zu diesem Zeitpunkt kann durch folgendes Schema dargestellt werden:



Durch den **bootstrap loader** werden folgende Systemprogramme, von der Diskette, in den Speicher übertragen.

IO.SYS

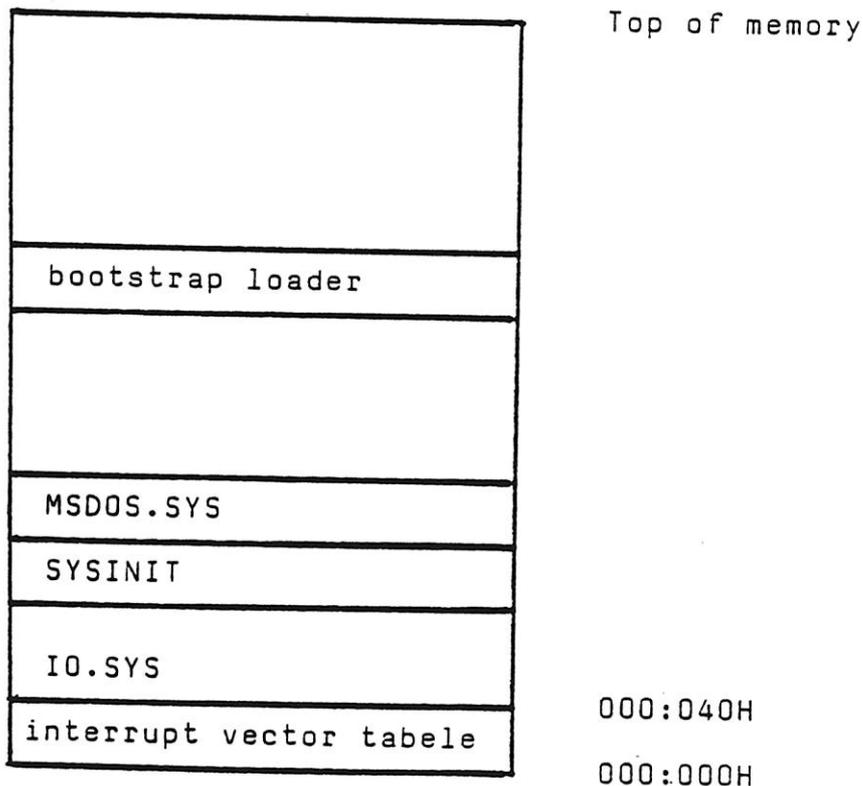
-SYSINIT Systeminitialisierung
-BIOS Default device drivers
(charater und blocks)
I/O handling (OEM written)

MSDOS.SYS

MS supplied hardware independent disk
operating system

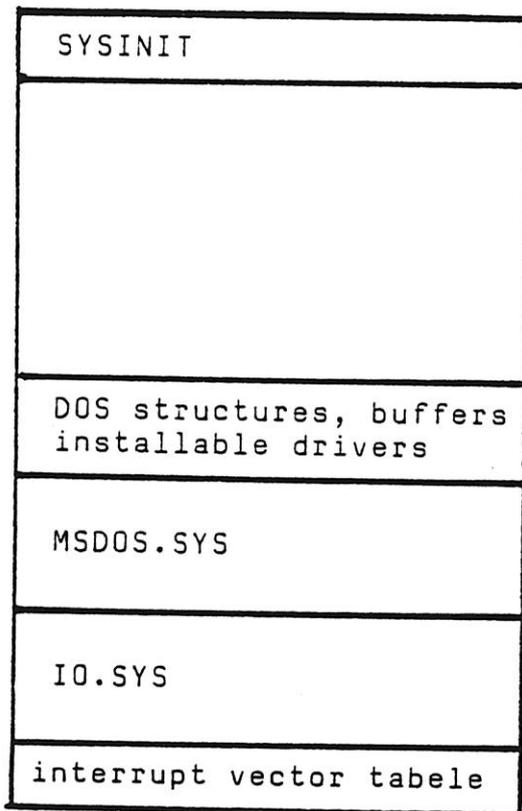
Die Verteilung im Speicher nachdem vom bootstrap loader das IO.SYS und MSDOS.SYS geladen wurde:

MEMORY MAP



Jetzt übernimmt SYSINIT die Steuerung und verschiebt sich selbst als erstes in den obersten Bereich des Speichers. Das MSDOS.SYS wird in die unteren Bereiche relokalisiert. Außerdem überprüft SYSINIT ob über CONFIG.SYS weitere Ladevorgänge gefordert werden. Dies ist der Fall wenn ein AUTOEXEC.BAT File angelegt worden ist. Es werden die Segment-Adressen der als nächstes zu ladenden Programme übergeben.

MEMORY MAP



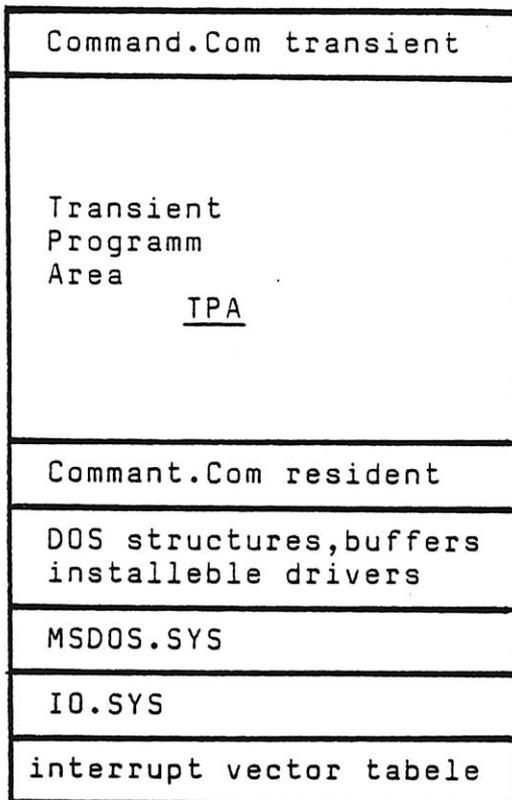
Top of memory

000:040H

000:000H

Als letzter Schritt wird von SYSINIT veranlasst, das der Command-Prozessor geladen wird. Dabei wird der transiente Teil wieder weit oben im Speicher abgelegt. Dieser transiente Teil kann später von Anwenderprogrammen überschrieben werden, er wird vom System dann wieder neu Initialisiert. Der residente Teil des Command-Prozessors schiebt sich in den unteren Speicherbereich. Der transiente Part des Command-Prozessor meldet jetzt durch A> die Betriebsbereitschaft des Systems. Kommandos können nun von der Konsole oder einer Batch-Datei eingegeben und abgearbeitet werden.

MEMORY MAP



Top of memory

000:040H

000:000H

5.1.0 MS-DOS File Allocation Tabel

MS-DOS Standard Disk Formats							
	5-1/4	5-1/4	5-1/4	5-1/4	8	8	8
No. sides	1	1	2	2	1	1	2
Tracks/side	40	40	40	40	77	77	77
Bytes/ sector	512	512	512	512	128	128	1024
Sectors/ track	8	9	8	9	26	26	8
Sectors/allocation unit	1	1	2	2	4	4	1
Reserved sectors	1	1	1	1	1	4	1
No. FATs	2	2	2	2	2	2	2
Root directory entries	64	64	112	112	68	68	192
No. sectors	320	360	640	720	2002	2002	616
Media Descriptor Byte	FE	FC	FF	FD	FE*	FD	FE*
Sectors for 1 FAT	1	2	1	2	6	6	2

*The two media descriptor bytes that are the same for 8" disks (FEH) is not a misprint. To establish whether a disk is single- or double-density, a read of a single-density address mark should be made. If an error occurs, the media is double-density.

Disketten-Organisation PC 16

1 Sektor = 1024 Byte

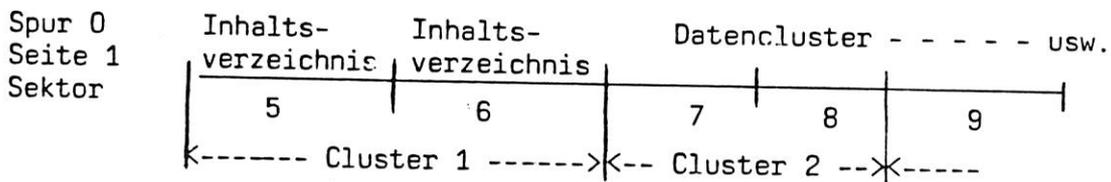
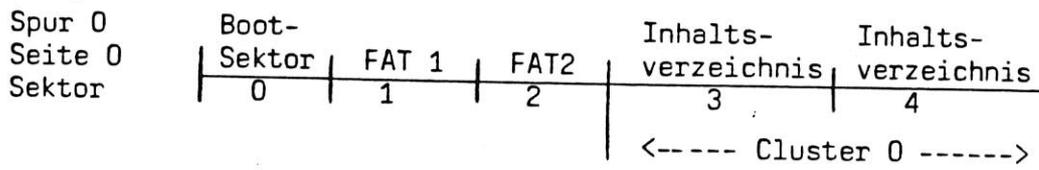
1 Cluster = 2 Sektoren

1 Spur = 5 Sektoren

1 Track = 10 Sektoren = 5 Cluster

1 Diskette = 40 Track (0-39)

Diskettenkapazität = $1024 \times 5 \times 40 \times 2 = 409600$ Byte
= 400 KB



Errechnen des Anfangs-Sektors eines Clusters = (Cluster x 2) + 3

Beispiel: gefragt ist nach dem Anf.Sektor von Cluster 9

$$\begin{aligned}
 &= (9 \times 2) + 3 \\
 &= 18 + 3 \\
 &= 21 \\
 &= 15 \text{ Hex.}
 \end{aligned}$$

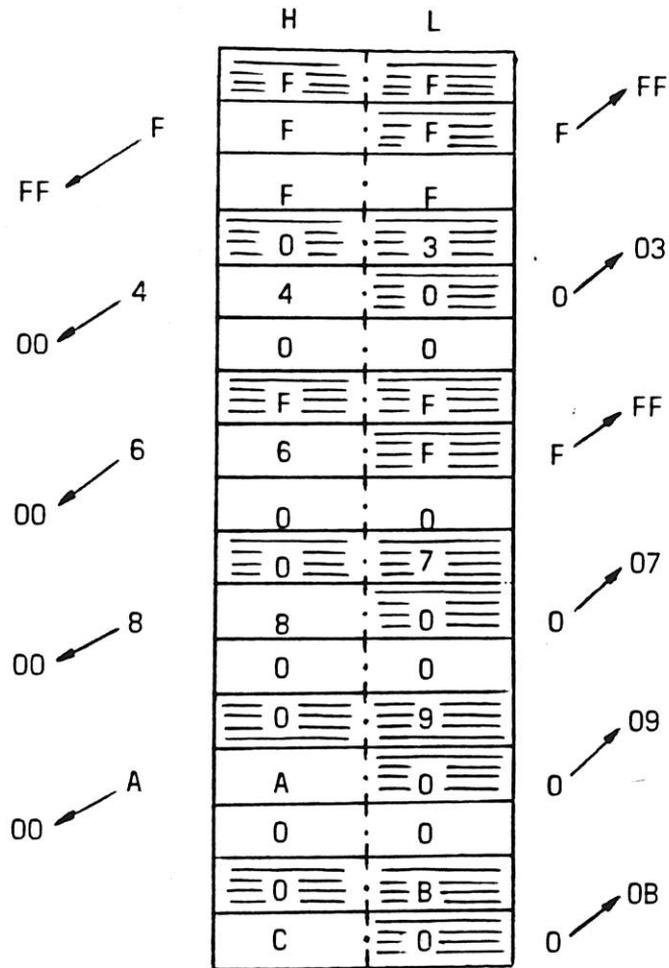
Im Inhaltsverzeichnis belegt eine Datei 32 Bytes.

Es stehen 4 Sektoren zur Verfügung also 4096 Bytes.

Somit sind $4096 : 32 =$ 128 Einträge möglich.

Beispiel der FAT Seite 6 und 8

Auf den Adressen 0400 - 040F stehen die Werte wie folgt im Speicher:



Display Anzeige: FF FF FF 03 40 00 FF 6F 00 07 80 00 09 A0 00 0B
 FFF FFF 003 004 FFF 006 007 008 009 00A

Beginnend von 1 muß auf jeder dritten Adresse das lower und das higher Halb-Byte getrennt werden.

Das lower Halb-Byte gehört vor die nächst niedrigere Adresse (= zu teilende ADR-1)

Das higher Halb-Byte gehört hinter die nächst höhere Adresse (= zu teilende ADR+1)

DEBUG

Aufruf: DEBUG

L = Load NNNN : NNNN, N, NNN, NNN
 | ADR auf die
 | gelesen
 | wird
 | (Rel. ADR) | *Laufwerk*
 | *0 oder 1* | *Anfangs-*
 | *Sektor* | *Anzahl der*
 | *Sektoren* |

Wird bei der Adr. 0 eingegeben, so bedeutet dies, daß auf die erste freie Adresse gelesen wird. Bei Display braucht dann auch nur 0 als Adresse eingetastet werden.

Wird eine Adresse angegeben, so muß bei Display dieselbe auch angegeben werden.

D = Display

Die Anzeige auf dem Bildschirm geschieht in Blöcken zu je 128 Byte.
8 solche Blöcke sind 1 Sektor.

Ändern von Zeichen:

Laden des Sektors, der geändert werden soll.

Dump anzeigen auf Bildschirm.

E-(Enter): E<ADR> = Anzeigen vom Inhalt der Adresse.

Wenn der Inhalt am Bildschirm steht, kann nach dem Punkt der neue Inhalt eingegeben werden (auch mehrere Folgebyte).

W (Write): W<ADR> (<Laufwerk>) (<Anf.Sekt>) (<Sekt.Anz>)
= Schreiben auf Diskette

Sektor 0 Boot-Sektor

```

D 0
09F5:0000 EB 4A 90 54 41 42 54 58-50 43 30 00 04 02 01 00 kJ.TABTXPCO.....
09F5:0010 02 80 00 90 01 00 01 00-05 00 02 00 00 00 55 55 .....UU
09F5:0020 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0030 00 00 00 00 00 00 00 00-55 55 55 55 55 55 55 55 .....UUUUUUUU
09F5:0040 00 00 00 00 00 00 00 00-00 00 00 00 8C C8 8E D8 .....4.....H.X
09F5:0050 8E C0 A1 16 00 8A 1E 10-00 32 FF F7 E3 8B F8 A1 .$.!.....2.wc.x!
09F5:0060 11 00 B3 20 F7 E3 F7 36-0B 00 03 C7 03 06 0E 00 ..3 wcw6...G....
09F5:0070 88 26 44 00 A2 45 00 B3-20 A1 0B 00 3D 00 04 74 .&D."E.3 !..=.t
-D
09F5:0080 16 02 DB 3D 00 02 74 0F-02 DB 3D 00 01 74 08 02 ..Ä=...t...Ä=...t..
09F5:0090 DB 3D 80 00 74 01 CB 88-1E 46 00 C6 06 47 00 00 Ä=...t.K..F.F.G..
09F5:00A0 C6 06 43 00 00 C6 06 42-00 08 B8 40 00 8E C0 BF F.C..F.B..8$.$.?
09F5:00B0 08 00 26 C5 1D 8C C8 8E-C0 26 89 1E 48 00 26 8C ..&E..H.$&..H.&.
09F5:00C0 1E 4A 00 8E D8 BA 40 00-B4 05 CD FD 0A C0 74 01 .J..X:$.4.Mü.$t.
09F5:00D0 CB B8 40 00 8E C0 BF 02-00 26 C6 05 01 BA 0B 00 K8$.$.?..&F...
09F5:00E0 FF 2E 48 00 55 55 55 55-55 55 55 55 55 55 55 55 ..H.UUUUUUUUUUUUU
09F5:00F0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
-D
09F5:0100 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0110 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0120 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0130 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0140 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0150 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0160 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0170 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
-D
09F5:0180 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:0190 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01A0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01B0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01C0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01D0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01E0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
09F5:01F0 55 55 55 55 55 55 55 55-55 55 55 55 55 55 55 55 UUUUUUUUUUUUUUUUU
-

```

- Byte 0B und 0C bezeichnen die Sektorengröße = 11hB
- Byte 0D zeigt die Anzahl der Sektoren pro Cluster
- Byte 0E zeigt die reservierten Sektoren (für Boot-Sekt.)
- Byte 10 zeigt die Anzahl der FAT.

Sektor 1. FAT 1 (File Allocation Table)

D																
09F5:0400	FF	FF	FF	03	40	00	FF	6F-00	07	80	00	09	A0	00	0B\$.o..... ..
09F5:0410	C0	00	0D	F0	FF	0F	00	01-11	20	01	13	40	01	15	60	\$.p..... \$.`
09F5:0420	01	FF	8F	01	19	A0	01	1B-C0	01	1D	E0	01	1F	00	02\$.`.....
09F5:0430	FF	2F	02	23	40	02	FF	FF-FF	FF	8F	02	FF	AF	02	2B	./.#\$....../.+
09F5:0440	F0	FF	FF	FF	FF	0F	03-31	20	03	33	F0	FF	35	F0		p.....1 .3p.5p
09F5:0450	FF	37	80	03	39	A0	03	3B-F0	FF	3D	E0	03	3F	F0	FF	.7..9 .;p.=`.?p.
09F5:0460	FF	FF	FF	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:0470	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
-D																
09F5:0480	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:0490	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:04A0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:04B0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:04C0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	:.....
09F5:04D0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:04E0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:04F0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
-D																
09F5:0500	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:0510	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
09F5:0520	00	00	00	00	00	00	00	00-00	55	55	55	55	55	55	55UUUUUUUU
09F5:0530	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:0540	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:0550	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:0560	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:0570	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
-D																
09F5:0580	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:0590	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05A0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05B0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05C0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05D0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05E0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU
09F5:05F0	55	55	55	55	55	55	55	55-55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUU

Byte 00 = Anzahl der Sektoren für eine FAT
 FF = 1 Sektor
 FD = 2 Sektoren

Jeder Eintrag ist in Gruppen zu je 3 Hex. Zeichen einzuteilen.
Erklärung dazu auf Seite 2.

000 = Cluster ist unbenutzt und steht zur Verfügung

FF7 = Cluster hat einen def. Sektor

FF8-FFF zeigt das letzte Cluster eines Files

XXX = Jede andere Zahl zeigt die nächste Cluster-Nummer eines Files. Die erste Cluster-Nummer eines Files steht im Inhaltsverzeichnis.

Sektor 2 FAT 2

D

09F5:0800	FF FF FF 03 40 00 FF 6F-00 07 80 00 09 A0 00 0B\$.o..... ..
09F5:0810	C0 00 0D F0 FF 0F 00 01-11 20 01 13 40 01 15 60	\$.p..... .\$.`
09F5:0820	01 FF 8F 01 19 A0 01 1B-C0 01 1D E0 01 1F 00 02\$.`.....
09F5:0830	FF 2F 02 23 40 02 FF FF-FF FF 8F 02 FF AF 02 2B	./.#\$.`...../..+
09F5:0840	F0 FF FF FF FF FF 0F 03-31 20 03 33 F0 FF 35 F0	p.....1 .3p.5p
09F5:0850	FF 37 80 03 39 A0 03 3B-F0 FF 3D E0 03 3F F0 FF	.7..9 .;p.=`.?p.
09F5:0860	FF FF FF 00 00 00 00 00-00 00 00 00 00 00 00
09F5:0870	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
-D		
09F5:0880	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:0890	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08A0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08B0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08C0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08D0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08E0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:08F0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
-D		
09F5:0900	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:0910	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
09F5:0920	00 00 00 00 00 00 00 00-00 55 55 55 55 55 55UUUUUUUU
09F5:0930	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:0940	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:0950	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:0960	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:0970	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
-D		
09F5:0980	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:0990	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09A0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09B0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09C0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09D0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09E0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU
09F5:09F0	55 55 55 55 55 55 55 55-55 55 55 55 55 55 55	UUUUUUUUUUUUUUUUUU

Sektor 3 Inhaltsverzeichnis (Directory)

D																	
09F5:0C00	49	4F	20	20	20	20	20	20-53	59	53	27	00	00	00	00	IO	SYS'
09F5:0C10	00	00	00	00	00	00	ED	4E-F3	0A	02	00	FF	17	00	00mNs.....	
09F5:0C20	4D	53	44	4F	53	20	20	20-53	59	53	27	00	00	00	00	MSDOS	SYS'
09F5:0C30	00	00	00	00	00	00	12	42-6B	0A	05	00	78	42	00	00Bk...xB..	
09F5:0C40	43	4F	4D	4D	41	4E	44	20-43	4F	4D	20	00	00	00	00	COMMAND	COM
09F5:0C50	00	00	00	00	00	00	B4	52-68	0A	0E	00	E5	40	00	004Rh...e\$..	
09F5:0C60	43	44	47	41	30	31	20	56-31	30	32	08	00	00	00	00	CDGA01	V102.....
09F5:0C70	00	00	00	00	00	00	23	11-21	00	00	00	00	00	00	00#!.....	
-D																	
09F5:0C80	46	4F	52	4D	41	54	20	20-43	4F	4D	20	00	00	00	00	FORMAT	COM
09F5:0C90	00	00	00	00	00	00	59	79-F3	0A	17	00	CB	4F	00	00Yys...KO..	
09F5:0CA0	43	48	4B	44	53	4B	20	20-43	4F	4D	20	00	00	00	00	CHKDSK	COM
09F5:0CB0	00	00	00	00	00	00	8E	54-68	0A	21	00	72	1B	00	00Th.!..r...	
09F5:0CC0	44	49	53	4B	43	4F	50	59-43	4F	4D	20	00	00	00	00	DISKCOPY	COM
09F5:0CD0	00	00	00	00	00	00	74	55-68	0A	25	00	52	06	00	00tUh.%..R...	
09F5:0CE0	53	59	53	20	20	20	20	20-43	4F	4D	20	00	00	00	00	SYS	COM
09F5:0CF0	00	00	00	00	00	00	04	57-68	0A	26	00	C5	03	00	00Wh.&..E...	
-D																	
09F5:0D00	50	52	49	4E	54	20	20	20-43	4F	4D	20	00	00	00	00	PRINT	COM
09F5:0D10	00	00	00	00	00	00	3D	59-72	0A	27	00	6C	0F	00	00=Yr.'..l...	
09F5:0D20	4D	4F	52	45	20	20	20	20-43	4F	4D	20	00	00	00	00	MORE	COM
09F5:0D30	00	00	00	00	00	00	24	59-68	0A	29	00	10	11	00	00\$Yh.).....	
09F5:0D40	42	41	53	49	43	20	20	20-43	4F	4D	20	00	00	00	00	BASIC	COM
09F5:0D50	00	00	00	00	00	00	EC	04-21	00	2C	00	F1	00	00	00l.!.,.q...	
09F5:0D60	43	41	53	53	20	20	20	20-45	58	45	20	00	00	00	00	CASS	EXE
09F5:0D70	00	00	00	00	00	00	33	6A-A7	0A	2D	00	80	07	00	003j'..-.....	
-D																	
09F5:0D80	42	54	58	20	20	20	20	20-43	4F	4D	20	00	00	00	00	BTX	COM
09F5:0D90	00	00	00	00	00	00	D2	5D-99	0A	2E	00	F5	02	00	00RÜ.....u...	
09F5:0DA0	45	44	4C	49	4E	20	20	20-43	4F	4D	20	00	00	00	00	EDLIN	COM
09F5:0DB0	00	00	00	00	00	00	87	56-68	0A	2F	00	1E	20	00	00Vh./.. ..	
09F5:0DC0	52	45	43	4F	56	45	52	20-43	4F	4D	20	00	00	00	00	RECOVER	COM
09F5:0DD0	00	00	00	00	00	00	C4	59-68	0A	34	00	C3	09	00	00DYh.4.C...	
09F5:0DE0	44	45	42	55	47	20	20	20-43	4F	4D	20	00	00	00	00	DEBUG	COM
09F5:0DF0	00	00	00	00	00	00	A5	51-41	06	36	00	F4	2D	00	00%QA.6.t...	

1 Eintrag = 32 Byte lang

Byte 00-07 = Volume Name

Byte 08-0A = Extension

byte 0B = Attribüt

Byte 0C-15 = Frei

Byte 16-17 = Uhrzeit, wann der File erstellt wurde.

Byte 18-19 = Datum, wann der File erstellt wurde.

Byte 1A-1B = Startcluster des Files

Byte 1C-1F = File-Größe in Byte

Fortsetzung Sektor 3

D																		
09F5:0E00	46	49	4E	44	20	20	20	20-45	58	45	20	00	00	00	00	FIND	EXE
09F5:0E10	00	00	00	00	00	00	F6	78-02	09	3C	00	00	19	00	00vx..<.....		
09F5:0E20	53	4F	52	54	20	20	20	20-45	58	45	20	00	00	00	00	SORT	EXE
09F5:0E30	00	00	00	00	00	00	AE	61-4E	0A	40	00	80	06	00	00aN.\$.....		
09F5:0E40	54	45	53	54	20	20	20	20-54	58	54	20	00	00	00	00	TEST	TXT
09F5:0E50	00	00	00	00	00	00	15	0F-21	00	41	00	00	01	00	00!.A.....		
09F5:0E60	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0E70	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
-D																		
09F5:0E80	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0E90	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0EA0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0EB0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0EC0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0ED0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0EE0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0EF0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
-D																		
09F5:0F00	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0F10	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0F20	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0F30	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0F40	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0F50	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0F60	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0F70	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
-D																		
09F5:0F80	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0F90	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0FA0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0FB0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0FC0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0FD0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														
09F5:0FE0	00	E5	E5	E5	E5	E5	E5	E5-E5	E5	.eeeeeeeeeeeeeeee								
09F5:0FF0	E5	E5-E5	E5	eeeeeeeeeeeeeeee														

Erklärung zu Attribut - Zeit - Datum

Offset 17H
 H H H H H M M M
 7 3 2 0

Offset 16H
 M M M S S S S S
 7 5 4 0

Offset 19H
 Y Y Y Y Y Y Y M
 7 1 0

Offset 18H
 M M N D D D D D
 7 4 0

Attribut

- 01 = Read only
- 02 = Hidden file
- 04 = System file
- 08 = Volume Name in den ersten 11 Bytes
- 10 = Sub-Directory
- 20 = Datei geschlossen

5.2.0 Fehlermeldung

Disketten-und Laufwerksfehler.

Format der Fehlermeldungen:

<yyy> ERROR WHILE <I/O Aktion> ON DRIVE <x>
Abort,Ignore,Retry:_

Bei <yyy> sind folgende Einträge möglich:

WRITE PROTECT	Es wurde versucht auf eine schreibgeschützte Diskette zu schreiben.
BAD UNIT	Unbekannte Einheit.
NOT READY	Laufwerk nicht bereit.
BAD COMMAND	Befehl nicht bekannt.
DATA	CRC-Fehler.
BAD CALL FORMAT	Falsche Formatierung der Disk.
SEEK	Positionierfehler.
NON DOS-DISK	Unbekannter Mediumtyp.
SECTOR NOT FOUND	Sektor nicht gefunden.
NO PAPER	Drucker hat kein Papier.
WRITE FAULT	Schreibfehler.
READ FAULT	Lesefehler.
DISK	Allgemeine Störung.

Bei <I/O Aktion> kann einer der beiden Einträge stehen:

READING	Bei Leseoperation.
WRITING	Bei Schreiboperation.

<x> bezeichnet das Laufwerk in dem der Fehler auftrat.

MS-DOS erwartet eine Eingabe des Bedieners:

Abort = A	Beendet das Programm.
Ignore = I	Fehler ignorieren.
Retry = R	Operation wiederholen.