

Janich & Klass

Computersysteme

```
***** Betriebssystem für Z80 - Computer *****
*
*      *****      *****      ***      *****      *
*      *      *      *      *      *      *      *      *
M      *      *      *      *      *      *      *      M
K      *      *      *      *      *      *      *      K
C      *      *      *      *      *      *      *      C
*      *      *      *      *      *      *      *
*      *****      *****      ***      *****      *
*
***** Version 5.b *****
```

Copyright (C) 1981, 82, 83 by
H. K. M.

ZDOS Version 5b1 ist im wesentlichen kompatibel zu CP/M 2.2. Es laufen alle Programme, die unter CP/M laufen, auch unter ZDOS. Einige Einschränkung sind die Programme XSUB und SUBMIT. Diese sind im ZDOS-Betriebssystem durch das Programm DO ersetzt. Von diese Einschränkung sind allerdings auch die Programme betroffen, die die CP/M-Version der Prozeduren benutzen. (Beispiel: DBASE Quit to program).

Ferner erlaubt ZDOS nur 1K oder 2K Blockgrößen. Diese Einschränkung tritt nur bei gravierenden Änderungen im HEAS auf. HEAS ist nicht vollständig kompatibel zu BIOS, d.h. HEAS kann erst nach leichten Änderungen als BIOS für CP/M-Systeme benutzt werden. Ein BIOS muß erweitert werden, um die HEAS-Funktionen zu erfüllen.

CP/M-Programme laufen unter ZDOS. Dies heißt aber nicht, daß ZDOS-Programme unter CP/M laufen müssen. Dies ist nur möglich, wenn beim Erstellen eigener Programme die CP/M-Restriktionen eingehalten wurden.

Da bei ZDOS die Ladeadresse des Files im File-Controll-Block steht, dieses Byte aber bei CP/M immer 0 ist, kann es bei manchen Kopierprogrammen Änderungen in der Ladeadresse geben. PIP arbeitet auch unter ZDOS richtig; I(nterchange) setzt die Ladeadresse konstant auf 0.

```

***** Betriebssystem für Z80 - Computer *****
*
*          *****          ***          ****          *
*          *      *      *      *      *      *      *
M          *      *      *      *      *      *      M
K          *      *      *      *      *      ****     K
C          *      *      *      *      *      *      C
*          *      *      *      *      *      *      *
*          *****          ***          ****          *
*
***** Version 5.b1 *****

```

Copyright (C) 1981, 82, 83 by
H.K.M.

Inhaltsverzeichnis

0.0	Einleitung	2
0.1	ZDOS-Übersicht	3
0.2	ZDOS-Module	4
0.3	Bsp: Erstellen einer Kopie	5
1.0	Eingaben von der Konsole	8
1.1	ZDOS-Fehlermeldungen	9
1.2	KI-Befehlsübersicht	11
1.3	Programmübersicht	17
1.4	CONFIG (Schnittstellenparameter)	19
1.5	COPY (Dateikopierprogramm)	21
1.6	DUMPLOD (Drucken von .LOD-Dateien)	23
1.7	DO (Prozeduren)	24
1.8	FILCOM (Dateivergleichsprogramm)	26
1.9	FILES (Diskettenhilfsprogramm)	27
1.10	FORMAT (Formatierprogramm)	28
1.11	GENCOM (Generieren von Programmen)	32
1.12	ID (Zuordnung Peripheriegeräte)	33
1.13	MTEST (Speichertestprogramm)	34
1.14	PSP (Maschinensprachenhilfspaket)	35
1.15	SYSGEN (Kopierprogramm)	43
1.16	USER (Benutzerschutzprogramm)	46
2.0	Prozeduren	47
3.0	Die IOBYTE-Funktion	49
4.0	ZDOS-Systemdisketten erstellen	50
4.1	ZDOS-Inbetriebnahme	51
5.0	HEAS-Aufrufe	53
5.1	Kaltstart / Warmstart	54
6.0	LEAS-Aufrufe	55
7.0	PSP-Befehlsübersicht + ASCII	57

Ein Computersystem besteht aus folgender Hardware (physikalisch vorhandenen Baugruppen):

1. der CPU, das ist die zentrale Rechen- und Verarbeitungseinheit
2. dem Arbeitsspeicher, in dem die Programme ausgeführt werden. Als Speicher werden heute meistens dynamische RAMs, d.h. Halbleiterspeicher verwendet. RAMs sind flüchtige Speicherbausteine, sie verlieren ihren Inhalt beim Ausschalten des Computers. Daher wird ein Massenspeicher (oder Hintergrundspeicher) zur dauerhaften Speicherung benötigt.
3. dem Hintergrundspeicher, meistens Disketten- oder Winchesterlaufwerken, auf denen alle Programme und Daten abgespeichert sind. Gespeichert wird auf Magnetplatten, diese sind dauerhafte wiederverwendbare Speichermedien. Disketten sind flexible Magnetplatten, die in Diskettenlaufwerken gelesen und beschrieben werden können (häufigster Hintergrundspeicher bei kleinen Computersystemen.)
4. einem Terminal oder Datensichtgerät zur Kommunikation zwischen Computer und Benutzer. Hier gibt der Benutzer seine Befehle an den Computer und hier erhält er auch die Antworten des Computers.
5. möglicherweise einem Drucker, um Ausgaben des Computers in schriftlicher Form zu erhalten.

Zum Betrieb benötigt ein Computersystem außer der Hardware noch Software. Software sind die Programme und Daten, die dem Computer angeben, was er zu tun hat. Sie werden in der Hardware des Computers (dem Speicher) gespeichert. Jedes Computersystem benötigt mindestens ein Betriebssystem. Das Betriebssystem übernimmt die Verwaltung der Computerkomponenten, es organisiert die Speicherung von Dateien (Programmen und Daten, z.B. Texte) auf der Diskette und es ermöglicht das Aufrufen von Programmen. Das Betriebssystem erlaubt also dem Benutzer den Umgang mit dem Computer. Es gibt unterschiedliche Betriebssysteme für Computer. Minimalforderung an ein Betriebssystem sind folgende Punkte:

- Verwaltung des Haupt- und Massenspeichers
- Verwaltung der Peripheriegeräte
- Kommunikation mit dem Benutzer
- Bereitstellen von Dienstprogrammen

Ferner sind die folgenden Forderungen wünschenswert:

- Portabilität (d.h. nicht nur für genau eine Hardware-Konfiguration geeignet)
- Benutzerführung (d.h. Hinweistexte auf der Diskette; der Idealfall wäre erreicht, falls kein Handbuch mehr erforderlich ist)
- breites Angebot von Programmiersprachen und Anwendungsprogrammen

Das hier beschriebene Betriebssystem ZDOS erfüllt im Wesentlichen alle oben genannten Anforderungen. Es ist kompatibel zu CP/M Version 2.2 von Digital Research, d.h. alle für CP/M angebotenen Programme laufen auch unter ZDOS. Somit stehen für ZDOS Compiler für fast alle Programmiersprachen (z.B. FORTRAN, BASIC, PASCAL, COBOL usw.) und mehr als hundert Anwenderprogramme zur Verfügung.

Zusätzlich zu den CP/M-Funktionen ermöglicht ZDOS es, den Drucker im Hintergrund zu bedienen, d.h. während des Druckens kann normal am System weitergearbeitet werden. Ferner bietet ZDOS ab Version 5.b die Möglichkeit, Dateien, die auf dem Betriebslaufwerk nicht gefunden werden, von Laufwerk A zu holen. Zur Ausnutzung dieser Erweiterung sollte man B als Betriebslaufwerk wählen, die Systemprogramme werden dann, falls nicht auf B, von Laufwerk A geholt. Dies vereinfacht die Bedienung erheblich.

In ZDOS wird bei jedem Aufruf an LEAS kontrolliert, ob der Benutzer durch Eingabe von ^S das System anhalten will; dies ermöglicht ein Abbrechen eines beliebigen Programmes. Da diese Erweiterung bei Programmen, die die HEAS-Sprungleiste benutzen, eventuell zu Störungen führt, ist sie durch Aufruf der LEAS-Funktion 250 ausschaltbar. Sie wird automatisch nach jedem Warmstart wieder eingeschaltet.

ZDOS-Aufrufe ändern nur die zur Ausgabe benötigten Register (im Gegensatz zu CP/M, das fast alle Register zerstört).

ZDOS benötigt einen Z80-Computer, bestehend aus mindestens 32K RAM, einem Floppy-Laufwerk als Massenspeicher und einem Terminal. Es unterstützt maximal 64K Byte RAM, bis zu 8 Floppylaufwerke, ein logisches Terminal, einen logischen Drucker und je einen logischen Streifenleser und -stanzer oder ähnliche Peripheriegeräte. Jedes dieser logischen Geräte kann einem von vier physikalischen Geräten zugewiesen werden. Weiterhin unterstützt das zu ZDOS gehörende HEAS als Option auch RAM-Disketten, d.h. Halbleiterspeicher, der von ZDOS wie eine Diskette verwaltet wird.

Durch die Sektorgröße von 512 Byte und die für 4 MHz Systemtakt optimierte Verschränkung ist HEAS bei Diskettenzugriffen sehr schnell.

ZDOS wartet auf Peripheriegeräte, d.h. ein angesprochenes, aber nicht eingeschaltetes oder nicht angeschlossenes Gerät hält den Computer an (Fehlermeldung: Geräte**n**ame, Fragezeichen; z.B. DISK? oder LPT?). Wenn also der Computer nichts tut, sollte zuerst überprüft werden, ob alle Geräte eingeschaltet und angeschlossen sind und ob die richtigen Disketten eingelegt worden sind. Erst danach sollte mit ^C oder ^S und ^C versucht werden, das System neu zu starten. Nur als letzte Möglichkeit dient der RESET-Schalter. Hierbei ist zu beachten, daß es Programme gibt, die sehr lange Ausführungszeiten benötigen!!

ZDOS besteht aus den in sich abgeschlossenen Modulen:

- dem Konsolinterpreter (KI), der zur Kommunikation zwischen Benutzer und Computer dient. Der KI nimmt Befehlszeilen entgegen und interpretiert sie durch LEAS-Aufrufe.
- dem logischen Ein/Ausgabe-System (LEAS). Dieses enthält die zur Speicherverwaltung (RAM und Diskette) benötigten Routinen und zusätzlich die Ein-/Ausgabe-Routinen. LEAS ist im Prinzip eine Sammlung von Unterprogrammen, die alle auch von Anwenderprogrammen aufgerufen werden können. Die Diskettenverwaltung läßt die auf der Diskette vorhandenen Sektoren und Spuren (physikalisch) für den Anwender unsichtbar werden. Unter LEAS besteht eine Diskette aus Inhaltsverzeichnis, Daten und Programmen (Dateien). Eine Datei besteht unter ZDOS aus mehreren zusammengehörigen Sektoren, die unter einem Namen ansprechbar sind. So ist es für den Anwender nicht mehr erforderlich, zu wissen, wo auf der Diskette Programme und Daten abgespeichert sind. Sie sind mit dem im Inhaltsverzeichnis abgelegten Namen jederzeit wieder aufrufbar.
- dem hardwareabhängigen Ein/Ausgabe-System (HEAS). HEAS besteht aus den Treiberrouinen für die Peripheriegeräte. Durch Änderung von nur HEAS kann ZDOS an unterschiedliche Hardware angepasst werden.
- dem Programmbereich (PB). Hier werden Programme des Anwenders und Hilfsprogramme des Betriebssystems ausgeführt. Der Programmbereich beginnt bei Adresse 100H und endet am Beginn von LEAS. Der Konsolinterpreter darf von Programmen überschrieben werden. Er wird beim Programmende durch einen LEAS-Aufruf (Warmstart) neu geladen.

Unter ZDOS dürfen Disketten nicht beliebig gewechselt werden. Disketten dürfen keinesfalls während eines Zugriffs des Computers auf das entsprechende Laufwerk entnommen werden. Zu diesem Zeitpunkt darf auch der Computer keinesfalls unterbrochen werden.

Es wird davor gewarnt, billige und schlechte Disketten zu verwenden. Eine Diskette, die sich nicht fehlerfrei formatieren läßt, gehört in den Papierkorb, aber keinesfalls in den Computer. Eine Diskette, die permanente Schreib- oder Lesefehler verursacht, kann meist nach neuer Formatierung weiter verwendet werden. Die auf ihr stehenden Daten gehen beim Formatieren verloren. (Kopien von allen wichtigen Disketten).

Eine Diskette, bei der ein BAD-SECTOR-Fehler gemeldet wird, der nicht durch mehrfaches Versuchen übergangen werden kann, wird an genau dieser Stelle immer wieder denselben Fehler melden. In diesem Fall sollte sie soweit möglich kopiert werden, und ab dann mit der Kopie weitergearbeitet werden. Ein Weiterarbeiten mit der Originaldiskette ist riskant, da eventuell Daten nicht mehr abgespeichert werden können. ZDOS übergeht fehlerhafte Sektoren nicht.

Als Beispiel für den Umgang mit dem Computer (unter ZDOS) soll jetzt die Systemdiskette dupliziert werden. (Dies ist unbedingt notwendig, da auf sie keinesfalls geschrieben werden darf). Folgende Schritte sind abzuarbeiten:

1. Starten des Computers

Einschalten aller Teile des Computersystems

Einlegen der Systemdiskette in Laufwerk A, das ist das obere Laufwerk, so daß der Aufkleber zur Tür des Laufwerks zeigt. Der längliche Schlitz in der Diskettenhülle ist dabei hinten.

Bei einer anderen Lage der Diskette kann der Computer (die Köpfe des Diskettenlaufwerks) beschädigt werden, sie ist also in jedem Fall zu vermeiden.

2. Bis jetzt ist weiter noch nichts geschehen. Der Urlader, ein Programm, das in einem PROM (dauerhafter Nur-Lese-Speicher) ist, wartet auf die Eingabe von Leerzeichen an dem Datensichtgerät. Erst wenn die Baudrate (Übertragungsgeschwindigkeit) des Datensichtgerätes bestimmt ist, erscheinen Meldungen des Urladers auf dem Bildschirm. Nach mehreren Diskettenzugriffen meldet sich dann anschließend das Betriebssystem (der Konsolinterpreter KI), das vom Urlader (im PROM) von der Diskette in den Arbeitsspeicher geladen worden ist, mit

OA>

Das heißt: Benutzer O, Betriebslaufwerk A

Sollte diese Meldung nicht erscheinen, liegt ein Fehler oder eine Fehlbedienung vor. Hierzu sind weitere Hilfen im "Urlader"-Kapitel angegeben. Ab jetzt befinden wir uns im Konsolinterpreter KI.

3. Wenn die Meldung OA> erschienen ist, erwartet das Betriebssystem (der Konsolinterpreter) weitere Befehle vom Benutzer. Wir wollen jetzt eine Diskette duplizieren. Dazu ist eine neue Diskette nötig. Diese kommt in Laufwerk B, das untere Laufwerk.

Jede neue Diskette muß vor der Benutzung formatiert werden, d.h. mit speziellen Daten vorbeschrieben werden.

4. Wir formatieren also jetzt die neue Diskette. Dazu gibt es ein Hilfsprogramm FORMAT. Dieses wird aufgerufen mit der Eingabe:

FORMAT<cr>

Nach jeder Eingabe ist das Wagenrücklaufzeichen (Carriage Return oder New-Line) notwendig (hier mit <cr> angedeutet), damit der Computer arbeitet. Falsch getippte Zeichen können durch Eingabe von CONTROL H, d.h. gleichzeitiges Betätigen der CONTROL und der H-Taste korrigiert werden. Hierdurch wird das letzte eingegebene Zeichen gelöscht. Ferner kann die gesamte bereits eingegebene Zeile durch Eingabe von CONTROL X oder RUBOUT gelöscht werden. Eingaben werden erst nach Carriage-Return beachtet.

Ab jetzt befinden wir uns im dem soeben aufgerufenen Program "FORMAT", also nicht mehr im Konsolinterpreter; d.h. ab jetzt gelten die KI-Befehle nicht mehr. Jetzt sind nur die Befehle für "FORMAT" zulässig (siehe bei FORMAT).

Wir wollten eine Diskette formatieren. Dazu muss die neue Diskette in Laufwerk B, das ist das untere Laufwerk, eingelegt werden.

Das Programm "FORMAT" meldet sich mit:

```
FORMAT V. 3.2 (26.10.82)
(B) 5,25", MFM, DS, 40 TRK
FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) =
```

oder einer ähnlichen Meldung. Falls die Meldung NOT INSTALLED, PLEASE INSTALL erscheint, ist zuerst das für den jeweiligen Computer erforderliche Format durch Eingabe von A bis I zu wählen. Danach geht es für beide Fälle gleich weiter:

Es ist B einzugeben, um die Diskette in Laufwerk B zu formatieren. FORMAT schreibt die Nummer der momentan bearbeiteten Spur auf den Bildschirm:

```
FORMATTING TRACK 00 (als Beispiel für Spur 0)
```

Bei zweiseitigen Disketten wird zweimal gezählt, einmal für die Vorder- und nochmals für die Rückseite. Erst danach ist die Diskette fertig formatiert; jetzt kann die Aufforderung:

```
HIT CARRIAGE RETURN TO RETURN TO ZDOS
```

durch Eingabe von Carriage-Return befolgt werden.

5. Jetzt sind wir wieder im Konsolinterpreter (KI). Als nächstes wird das Programm SYSGEN zum direkten Kopieren von Disketten aufgerufen:

```
SYSGEN
```

Es meldet sich mit:

```
SYSTEM GENERATION AND COPY PROGRAMM
```

```
(TYPE X FOR EXIT)
```

```
(B) 5,25", MFM, DS, 40 TRK
```

```
SELECT: SYSGEN (S), COPY (C), COPY TRACK (T) OR INSTALL (I)
```

oder einer ähnlichen Meldung. Falls die Meldung NOT INSTALLED, PLEASE INSTALL erscheint, ist zuerst das für den jeweiligen Computer erforderliche Format durch Eingabe von A bis I zu wählen (wie bei FORMAT). Danach geht es für beide Fälle gleich weiter:

Es soll die Diskette in A auf die in B kopiert werden. Damit ist COPY durch Eingabe von C auszuwählen; die anschließende Frage:

SOURCE DISK FOR COPY =

ist mit A und die nächste:

DESTINATION DISK =

ist mit B zu beantworten. Jetzt wird kopiert; dabei wird die momentane Spur wie bei FORMAT angezeigt. Falls ein READ ERROR gemeldet wird, ist die gesamte Kopie nicht gelungen; es ist mit FORMAT neu zu beginnen.

Im Normalfall erscheint die Anforderung:

HIT CARRIAGE RETURN TO RETURN TO ZDOS

Nach Eingabe von Carriage-Return meldet sich wieder der Konsolinterpreter mit OA). Damit ist die Kopie der Systemdiskette fertig.

Die soeben erstellte Kopie kann durch Aufruf von SYSGEN mit dem Original verglichen werden (VERIFY). Wenn kein VERIFY-ERROR gemeldet wird, stimmen die beiden Disketten überein.

Trotzdem sollte die soeben neu erstellte Diskette (die Kopie) jetzt vorsichtshalber in Laufwerk A ausprobiert werden.

Benutzt wurden in obigem Beispiel die Programme: FORMAT, SYSGEN und das Betriebssystem ZDOS, bestehend aus KI, LEAS und HEAS.

Eingaben von der Konsole

Die Eingabe an das System erfolgt zeilenweise. Zeilen werden durch Carriage-return oder New-line abgeschlossen, d.h. Kommandos werden erst dann ausgeführt. Die Line-Feed-Taste dient zur logischen Trennung von Eingaben, die eigentlich in unterschiedlichen Zeilen stehen müßten. Innerhalb einer Zeile sind folgende Korrekturmöglichkeiten gegeben:

^H oder Backspace löscht das zuletzt eingegebene Zeichen

^X oder Rubout bzw. Delete löscht die gesamte Zeile

^I oder Tab springt auf die nächste Tabulatorposition (Tabulatoren sind alle 8 Zeichen gesetzt)

Folgende Zeichen haben Sonderbedeutungen:

^C als erstes Zeichen veranlaßt einen Warmstart.

^P protokolliert alle Ein- und Ausgaben auf dem Drucker, das zweite ^P schaltet den Drucker wieder aus.

Line-Feed ermöglicht die Eingabe von mehreren Kommandos in einer Zeile. Es wird durch das '&'-Zeichen dargestellt.

^S stoppt jedes Programm beim nächsten LEAS-Aufruf. Jetzt gibt es 2 Möglichkeiten:

^Q setzt die Programmausführung fort,

^C bricht das Programm ab und übergibt die Kontrolle wieder an das Betriebssystem.

Alle anderen Eingaben nach ^S werden ignoriert.

Im KI können Listings auf die Konsole nur durch Eingabe von Carriage-Return, Escape oder Blank abgebrochen werden. Bei TYPE mit der P-Angabe wird das Listing alle 20 Zeilen gestoppt; es kann durch Eingabe von Carriage-Return, Line-Feed oder Escape fortgesetzt werden; das zur Fortsetzung verwendete Zeichen kann nicht zum Abbrechen des Listings verwendet werden (dies verhindert ein ungewolltes Abbrechen des Listings bei Terminals, die Autorepeat haben).

^char heißt CONTROL char und bedeutet, daß die CONTROL-Taste und der entsprechende Buchstabe (char) gleichzeitig betätigt werden.

Fehler können von allen Teilen des Betriebssystems gemeldet werden.

Fehlermeldungen des Kommandointerpreters

NO FILE	Die im Kommando angegebene Datei ist nicht vorhanden.
BAD LOAD	Die angegebene Datei kann nicht geladen werden; sie würde KI überschreiben.
NO SPACE	Für den zu rettenden Speicherbereich reicht der auf der Diskette vorhandene freie Platz nicht aus.
FILE EXISTS	Eine Datei mit dem neuen Namen existiert bereits. (bei RENAME)
READ ERROR	Die angegebene Datei kann nicht vollständig gelesen werden. (Meist wurde versucht, Random-Dateien zu lesen.)
xx?	Dieses Kommando ist unverständlich. Der nicht interpretierbare Teil steht vor dem Fragezeichen.
ALL (Y/N)?	Keine wirkliche Fehlermeldung; es wird nur vorsichtshalber nochmal gefragt, ehe alle Dateien gelöscht werden. Y löscht, N läßt bestehen.

LEAS Fehlermeldungen

Alle LEAS-Fehlermeldungen haben folgende einheitliche Form:

ZDOS ERR ON laufwerk: erklärung

Hierbei ist laufwerk der Name des betroffenen Laufwerks (A...H) und erklärung einer der folgenden Texte:

SELECT	Das ausgewählte Laufwerk existiert nicht. Bei Betätigung einer beliebigen Taste führt das System einen Neustart durch und selektiert Laufwerk A.
BAD SECTOR	Fehler bei Lese- oder Schreibzugriffen auf die Diskette. CARRIAGE RETURN veranlasst einen weiteren Versuch, ^C bricht ab.
R/O	Das angegebene Laufwerk befindet sich im Nur-Lese-Zustand. Wahrscheinlich wurde die Diskette vorher ohne Meldung an ZDOS gewechselt. Die Betätigung einer beliebigen Taste führt zu einem Neustart.

FILE R/O	Die ausgewählte Datei ist eine Nur-Lese-Datei, d.h. sie kann nicht überschrieben oder umbenannt werden. CARRIAGE RETURN ermöglicht das Überschreiben nur für USER 0, ^C bricht ab.
SPOOLING	Diese Fehlermeldung tritt auf, wenn während des Druckens im Hintergrund (LIST) versucht wird, den Drucker anzusprechen. In diesem Fall hat LIST immer Vorrang. CARRIAGE RETURN setzt das Programm, das zu drucken versucht hat, fort, übergeht aber alle Druckeraufrufe dieses Programms. ^C bricht das Programm ab.
ERASE SPOOLING	Es wurde versucht, eine Datei, die momentan gedruckt wird, zu löschen. Dies ist erst nach Eingabe von STOP oder Aufruf von LEAS-Funktion 254 möglich.

HEAS Fehlermeldungen

DISK?	Die Diskette ist nicht richtig im Laufwerk eingelegt. Diskette richtig einlegen und dann Carriage-Return betätigen. Die Operation wird dann fortgesetzt. ^C bricht ab!
LPT?	Drucker nicht bereit, einschalten bzw. anschließen oder selektieren. Eingabe wie bei DISK?
CONA?	V24-Schnittstelle A hat Handshake on und das angeschlossene Gerät wird nicht ready.
CONB?	Wie bei CONA?, aber Schnittstelle B
*** WRITE PROTECTED ***	Die Diskette ist schreibgeschützt.

Der letzte Fehler ist nur eine Information, die Kontrolle wird an LEAS übergeben, LEAS meldet
ZDOS ERROR ON laufwerk: BAD SECTOR
 Jetzt kann, falls gewünscht, die Operation durch Eingabe von CARRIAGE RETURN nach Entfernen des Schreibschutzes fortgesetzt werden. Der Schreibschutz ist ein Streifen, der von der Diskette entfernt (8") oder auf die Diskette geklebt (5,25") werden muß.

Weitere Fehler können von den Dienstprogrammen gemeldet werden. Sie sind dann dort beschrieben.

In der folgenden Beschreibung bedeutet:

laufwerk: Ausgewähltes Laufwerk A:, B:, ..., H:
 dateiname Name der gewünschten Datei (max. 8 Zeichen)
 .dateityp Typ der gewünschten Datei (max. 3 Zeichen)
 länge Länge des Bereichs in Einheiten von 256 Bytes (zwischen 0 und 255)
 \$adresse Startadresse des Bereichs (hexadezimal)

Angaben in Klammern können weggelassen werden.

Falls in einem Kommando das laufwerk: weggelassen wird, ist das Betriebslaufwerk oder, falls auf ihm die Datei nicht gefunden werden kann, Laufwerk A gemeint.

Folgende Befehle sind im KI implementiert:

HELP (kommando)

Ausdrucken eines Hinweistextes auf dem Schirm. Die ersten 4 Zeichen des Kommandos sind signifikant. Benötigt wird die Datei SYSTEM.HLP.

Bsp: 4C>HELP DIR

DIR ((laufwerk:)(dateiname.dateityp))

Ausdrucken des Inhaltsverzeichnisses

Bsp: 0A>DIR B:

TYPE (laufwerk:)dateiname.dateityp (P)

Ausgabe von Textdateien auf dem Terminal, das P bewirkt eine seitenweise Ausgabe, d.h. alle 20 Zeilen wird das Listing gestoppt, es kann mit CR, ESC oder Blank fortgesetzt werden.

Bsp: 0A>TYPE A:TEXT.TXT

ERA (laufwerk:)dateiname.dateityp

Löschen einer Datei

Bsp: 0A>ERA TEXT.TXT

REN (laufwerk:)neuename.neuertyp=altername.altertyp

Umbenennen einer Datei

Bsp: 4A>REN A:TEXT1.ASC=TEXT.TXT

SAVE länge (laufwerk:)dateiname.dateityp (\$adresse)

Ablegen eines Speicherbereichs auf Diskette

Bsp: 15A>SAVE 25 PROG.COM \$4000

LOAD (laufwerk:)dateiname.dateityp (\$adresse)

Laden einer Datei in den Speicher

Bsp: 0A>LOAD PROG.COM

fett gedruckte Zeichen in Beispielen sind Eingaben am Terminal; normale sind Ausgaben. Vor dem > steht die Benutzernummer (0 bis 15) und das Betriebslaufwerk (A bis H). Blanks sind signifikant.

laufwerk:
Auswahl des Betriebslaufwerks
Bsp: OB>A:

LIST (laufwerk:)dateiname.dateityp
Ausgabe einer Textdatei auf den Drucker im Hintergrund (Tabs müssen vom Drucker expandiert werden)
Bsp: OB>LIST A:HILFE

STOP
Abbrechen des LIST-Kommandos (Dies ist die einzige Möglichkeit, ein LIST-Kommando abzubrechen!)

NEW
Meldung eines Diskettenwechsels an das System

(laufwerk:)dateiname (parameterliste)
Laden und Ausführen eines Maschinenprogramms mit dem dateityp COM. Die Parameterliste wird an das Programm übergeben.
Bsp: OA>B:ASS PROGRAMM/L

Die Angabe des Laufwerkes ist nur erforderlich, falls weder das Betriebslaufwerk noch Laufwerk A gemeint ist. (Auf Laufwerk A wird automatisch gesucht, falls die Datei sonst nicht gefunden werden kann). Ausnahmen sind: ERA, REN, DIR und SAVE.

dateiname.dateityp können auch mehrdeutig angegeben werden. Ein Fragezeichen ersetzt ein beliebiges Zeichen an seiner Position. Ein Stern ist gleichbedeutend mit Fragezeichen bis zum Ende von Dateiname oder -typ.
Bsp: TEST?.* kann stehen für TEST1.COM, TEST.TXT oder TESTA.AAA, aber nicht für TEST12.COM; TEST*.* könnte für TEST12.COM stehen. Bei mehrdeutigen Dateibezeichnungen werden alle hiervon erfaßten Dateien beeinflusst (d.h. bei ERA *.* werden alle Dateien der Betriebsdiskette gelöscht).

Falls eine Datei mit dem Namen eines KI-Kommandos ausgeführt werden soll, muß zur Unterscheidung vor dem Namen eine Laufwerksbezeichnung angegeben werden. Beispiel:
Das Programm mit dem Namen LOAD.COM soll ausgeführt werden. Zur Unterscheidung zum KI-Befehl LOAD ist:
OA>A:LOAD einzugeben, falls das Programm LOAD auf der Diskette in Laufwerk A ist.

Ausgaben des KI können durch Eingabe von ^S gestoppt, anschließend durch ^Q fortgesetzt, und durch Eingabe von Carriage-Return, Escape oder Blank abgebrochen werden. ^S, ^C veranlaßt einen Systemwarmstart. TYPE mit P kann durch Eingabe von Carriage-Return, Escape oder Blank nach einem Stop fortgesetzt werden. Das hierfür verwendete Zeichen ist als Abbruchzeichen gesperrt.

Die nun folgenden Befehlsbeschreibungen sind mit `HELP` befehl im System aufrufbar. Signifikant sind die ersten 4 Zeichen eines Befehls.

KI-Kommando `D I R`

`D I R` zeigt das Inhaltsverzeichnis einer Diskette in folgender Form an:

A: filename.ext \$adr filename.ext \$adr filename.ext \$adr

Eingabemöglichkeiten:

<code>DIR</code>	Inhaltsverzeichnis des Betriebslaufwerkes
<code>DIR A:</code>	Inhaltsverzeichnis von "A" (alle Files)
<code>DIR A:*. *</code>	Inhaltsverzeichnis von "A" (alle Files)
<code>DIR *.ext</code>	nur Files vom Typ "ext"
<code>DIR X*. *</code>	alle Files, deren Name mit "X" anfaengt

`D I R` zeigt Files des aktuellen USERS und die Files von USER 0 an, ohne sie explicit zu unterscheiden. SYS-Files werden von `D I R` nicht erfasst.

KI-Kommando `T Y P E`

`T Y P E` druckt ein File auf dem Terminal und bei `^P` auch auf dem Drucker aus. Die Eingabe von `P` bewirkt ein seitenweises Drucken (jeweils 23 Zeilen); fortgesetzt wird das Listing nach Eingabe von `<CR>`, `<ESC>` oder `<BLANK>`. Listings werden durch Eingabe von `^S` gestoppt, mit `^Q` nach einem `^S` fortgesetzt und durch Eingabe von `<CR>`, `<ESC>` oder `<BLANK>` abgebrochen. Ein `^C` nach einem `^S` fuehrt einen Systemwarmstart durch.

Eingabemöglichkeiten:

<code>TYPE filename.ext</code>	liste die Datei "filename.ext"
<code>TYPE filename.ext P</code>	liste die Datei "filename.ext", Stop nach jeweils 23 Zeilen
<code>TYPE A:*.MAC</code>	drucke das erste MAC-File der Diskette "A".

Zur Verhinderung von Fehleingaben ist das Zeichen, das soeben zum Fortsetzen des Listings eingegeben wurde, nicht als Abbruchzeichen zulaessig. Es ist darauf zu achten, nur Files zu "typen", die nur druckbare ASCII-Zeichen enthalten.

KI-Kommando L I S T

L I S T druckt ein File im Hintergrund auf dem Systemdrucker. L I S T kann nur durch Eingabe von S T O P abgebrochen werden. Waehrend des LISTENS eines Files sind die Programme FORMAT und SYSGEN nicht benutzbar. Die Diskette, die das LIST-File enthaelt, darf waehrend des "listens" nicht gewechselt werden.

Eingabemoeglichkeiten:

LIST filename.ext	Drucken von "filename.ext" im Hintergrund; Datei ist auf dem Betriebslaufwerk.
LIST B:TEXT.ABC	Drucken des Files "TEXT.ABC" der Diskette "B"

L I S T arbeitet waehrend der Wartezeiten auf die Tastatur bei LEAS-Aufrufen; d.h. es arbeitet nicht bei HEAS-Aufrufen, z.B. in Verbindung mit dem WORDSTAR.

KI-Kommando E R A

E R A loescht Dateien von der Diskette. Es sind sowohl einzelne Dateien wie auch ganze Dateigruppen erfassbar.

Eingabemoeglichkeiten:

ERA *.*	loesche alle Dateien des Betriebslaufwerks; hier wird vorsichtshalber noch mal nachgefragt und erst nach Eingabe von Y geloescht
ERA B:ABCDEF.*	loesche auf Laufwerk "B" die Files mit dem Namen "ABCDEF", ohne auf den Typ zu achten.
ERA A*.*	loesche alle Files auf dem Betriebslaufwerk, deren Name mit A beginnt.

Die Files werden nicht wirklich geloescht, sie werden nur im Inhaltsverzeichnis als geloescht deklariert. Es koennen nur Dateien des jeweiligen Benutzers geloescht werden; die auch aufrufbaren Dateien des USER 0 (Systemprogramme) sind von anderen nicht loeschbar.

KI-Kommando R E N

R E N benennt Dateien um. Es koennen Dateien und teilweise auch Dateigruppen erfasst werden. R E N kann nur Files des jeweiligen USERS umbenennen.

Eingabemoeglichkeiten:

REN A:ABC.123=DEF.456	Das File "DEF.456" auf Laufwerk "A" wird in "ABC.123" umbenannt
REN *.MAC=*.SRC	Alle Dateien des Typs "SRC" erhalten den neuen Typ "MAC"
REN ?80.XXX=?80.COM	Alle Files, die als 2. und 3. Buchstaben "80" und als Typ "COM" haben, bekommen den Typ "XXX"

Falls der neue Name bereits auf der Diskette gefunden wird, erscheint die Fehlermeldung FILE EXISTS, und REN wird nicht ausgefuehrt; es ist erst das File mit dem neuen Namen zu loeschen, dann kann das REN-Kommando wiederholt werden.

KI-Kommando N E W

N E W teilt dem System einen Diskettenwechsel mit. N E W ist immer dann erforderlich, wenn auf eine nach dem letzten Systemwarmstart gewechselte Diskette geschrieben werden soll; die Diskette waere sonst R/O (read only). Nach einem ^C ist NEW nicht erforderlich.

Eingabemoeglichkeit:

NEW	Meldung Diskettenwechsel an ZDOS
-----	----------------------------------

KI-Kommando S A V E

S A V E rettet einen Speicherauszug auf die Diskette. Angegeben wird die Laenge in 256-Byte, der Name und die Anfangsadresse. Falls keine Adresse angegeben wird, wird 100H genommen.

Eingabemoeglichkeiten:

SAVE 1 TEST.COM \$4500	speichere 256 Byte ab Adresse 4500H unter dem Namen TEST.COM
SAVE 255 B:MEM	speichere 255*256 Byte ab Adresse 100H unter dem Namen MEM auf Diskette B

KI-Kommando L O A D

L O A D laedt ein File in den Speicher. Erforderlich sind: Filename und Startadresse, falls nicht die Originaladresse genommen werden soll.

Eingabemöglichkeiten:

LOAD PROGRAM.COM	Lade das File PROGRAM.COM an seine Startadresse
LOAD PROGRAM.COM \$100	Lade PROGRAM.COM ab Adresse 100H unabhaengig von der Originaladresse

Die Fehlermeldung BAD LOAD sagt aus, dass das File zu lang ist und daher nicht geladen werden kann, ohne Systemteile zu ueberschreiben. Das Laden wird abgebrochen, bevor das System ueberschrieben wird.

KI-Kommando S T O P

S T O P ist ausser dem RESET-Schalter die einzige Moeglichkeit ein LIST-Kommando (Drucken im Hintergrund) abubrechen.

Eingabemöglichkeit:

STOP	"LISTen" eines Files beenden
------	------------------------------

andere Kommandos

Ferner sind alle Dateien mit dem Dateityp .COM genauso wie KI-Kommandos ausfuehrbar; d.h. eine .COM-Datei mit dem Namen TEST.COM ist durch die Eingabe:

OA>TEST

genau wie ein KI-Kommando ausfuehrbar; einziger Unterschied ist, daB sie umbenannt werden kann und nicht Teil des KI ist, sondern vor Ausfuehrung von der Diskette in den Programmbereich PB geladen wird. Das erfordert, daB alle benoetigten .COM-Dateien auf einer Diskette im Computer erreichbar sein muessen.

Dies ermoeglicht es, beliebige neue Kommandos zu generieren. Im Prinzip ist jedes ausfuehrbare Programm unter dem KI ein Kommando (oder einem Kommando vergleichbar).

Wie bereits beim KI gesagt, kann die Kommandoliste durch ausführbare Programme (Dateityp = .COM) erweitert werden. Dies wird bei den nun folgenden Dienstprogrammen (oder auch Kommandos?) ausgenutzt. Es gibt also noch die folgenden Befehle:

CONFIG (parameterliste)

Einstellen von Schnittstellenparametern; falls die parameterliste weggelassen wird, arbeitet CONFIG interaktiv.

COPY (laufw:)(kopienname.typ) (laufw:)dateiname.typ

Kopieren der Datei "dateiname.typ" in die Datei "kopienname.typ". Falls die beiden Namen gleich sind, kann "kopienname.typ" weggelassen werden, dann muß in jedem Fall laufwerk angegeben werden. Es ist möglich mit nur einem Laufwerk zu kopieren (Pause zum Diskettenwechsel vor dem Zurückschreiben)

DUMPLD (laufwerk:)dateiname.(lod)

Ausdrucken einer Datei vom Typ ".LOD"

DO name parameterliste

Ausführen der Prozedur "name" mit den Werten aus der Parameterliste

FILCOM (laufw:)datei1.typ (laufw:)datei2.typ

Vergleichen der Dateien datei1 und datei2

Unterschiede werden als HEX-Werte ausgegeben

FILES (laufw:)dateiname.typ

Ausgabe eines ausführlichen Inhaltsverzeichnisses aller zu dateiname.typ passenden Dateien; ferner Angabe des belegten und des freien Platzes der Diskette

FILES DSK:

Ausgabe der (vereinbarten) Daten des Betriebslaufwerkes

FILES (laufw:)dateiname.typ =attr

Setzen von Attributen für Dateien. Es gibt die Attributpaare: SYS/DIR und R/O/R/W.

FORMAT

Formatieren einer Diskette. FORMAT ist ein menu-gesteuertes interaktives Programm

GENCOM (laufw:)dateiname \$adresse

Generieren eines ausführbaren Programmes (Typ .COM) für die Startadresse adresse aus einer .LOD-Datei

IO

Ausgabe der Einstellung des IO-Bytes (Zuordnung zwischen logischen und physikalischen Peripheriegeräten)

IO ldev:=pdev:

Verändern des IO-Bytes. Das logische Gerät ldev wird dem physikalischen Gerät pdev zugewiesen

MTEST

ist ein Speichertestprogramm und wird nur benötigt, falls das Computersystem getestet werden muß

PSP (laufw:)dateiname.typ

Aufruf des Maschinensprachenentwicklungs- und Test-Paketes. Falls dateiname.typ angegeben ist, wird die entsprechende Datei gleichzeitig geladen. PSP ist ein interaktives Programm

SYSGEN

Systemerstellungs-, Systemkopier- und allg. Kopierprogramm. SYSGEN ist interaktiv und menugesteuert

USER nummer (name)

Wechseln des Benutzerbereiches. Programme sind nur dem jeweiligen Benutzer zugänglich. nummer und name müssen zusammengehören, sonst wird der Wechsel nicht gestattet. (Ausnahme ist USER 0, er ist immer und für alle anderen Benutzer ohne besonderen Schutz erreichbar.)

Dies sind die mit dem ZDOS-System mitgelieferten Dienstprogramme, diese Liste kann durch Zukauf von weiteren Programmen fast beliebig verlängert werden.

CONFIG ist ein Dienstprogramm zur Einstellung der normalerweise vorhandenen Schnittstellen und Peripheriegeräte. **CONFIG** basiert auf Menüs, und ist daher weitgehend selbsterklärend. Beim Aufruf erscheint folgendes Menü:

H.K.M. CONFIGURATOR PROGRAM VERSION 2.6 (23.09.82)

SELECT I/O-PORT

- (A) FOR SERIAL I/O PORT A, NORMALLY READER/PUNCH
- (B) FOR SERIAL I/O-PORT B, NORMALLY CONSOLE
- (P) FOR PRINTER (EPSON MX 80)
- (X) FOR EXIT TO ZDOS

Jetzt kann durch Eingabe von **A**, **B** oder **P** die gewünschte Funktion ausgewählt werden. **X** beendet **CONFIG** und kehrt zum Betriebssystem zurück.

Bei Eingabe von **A** erscheint folgendes Menü:

SELECT BAUDRATE AND HANDSHAKE SIGNALS FOR PORT A

- | | |
|-------------------|-----------------------|
| (A) FOR 9600 BAUD | (J) FOR HANDSHAKE ON |
| (B) FOR 4800 BAUD | (K) FOR HANDSHAKE OFF |
| (C) FOR 2400 BAUD | (L) FOR NO PARITY |
| (D) FOR 1200 BAUD | (M) FOR PARITY ODD |
| (E) FOR 600 BAUD | (N) FOR PARITY EVEN |
| (F) FOR 300 BAUD | (O) FOR 7 BITS |
| (G) FOR 150 BAUD | (P) FOR 8 BITS |
| (H) FOR 110 BAUD | (Q) FOR 1 STOP BIT |
| (I) FOR 75 BAUD | (R) FOR 2 STOP BITS |
| (X) FOR EXECUTION | |

Hierbei wird bei **HANDSHAKE ON** nur auf Peripheriegeräte gewartet, d.h. **CTS** und **DCD** werden beachtet. **RTS** ist konstant aktiv, **DTR** ist konstant passiv.

Durch Eingabe von **P** im Hauptmenü können folgende Einstellungen an einen angeschlossenen EPSON MX80 übergeben werden:

Achtung: **CONFIG** kann nur richtig arbeiten, wenn vom **BOOT** beim Kaltstart die Baudrate des Terminal übergeben worden ist (CTC-Time-Constant)

MX-80 CONTROL CODES

- (B) FOR HORIZONTAL TAB EVERY 8-TH COLUMN
- (C) FOR 132 CHARACTERS/LINE
- (D) FOR 80 CHARACTERS/LINE
- (E) FOR PRINT EMPHASIZED
- (F) FOR PRINT NOT EMPHASIZED
- (G) FOR DOUBLE PRINTING
- (H) FOR SINGLE PRINTING
- (X) FOR EXIT

(vgl. EPSON MX80 Manual)

Die gewählten Einstellungen in den Untermenüs werden erst bei Verlassen des jeweiligen Menüs durch X ausgeführt. Im Untermenü B ist zu beachten, das bei Veränderungen der Baudrate auch die Baudrate des an Port B angeschlossenen Terminals umzustellen ist!

Zur richtigen Einstellung der Baudrate der seriellen Schnittstellen ist die Kenntnis des Systemtaktes erforderlich. Dieser wird indirekt vom Urlader BOOT641 oder einer neueren Version an HEAS übergeben und in der CONFIG-Tabelle abgespeichert. Bei Modifikation des Bootstraploaders ist hierauf zu achten!

Eine einmal mit CONFIG gewählte Einstellung bleibt bis zum nächsten Kaltstart des Systems bestehen. Nach jedem Kaltstart gelten wieder die Standardeinstellungen. CONFIG muß also nach jedem Kaltstart neu aufgerufen werden, um andere als die Standardeinstellungen zu erhalten. (dies ist möglich in der Prozedur INIT.DO)

CONFIG verarbeitet die Eingabezeile, d.h. die sonst in den Menüs selektierten Einstellungen können auch hinter dem Aufruf stehen. Beispiel:

0A>CONFIG AAJMRXX

CONFIGURATION FOR PORT A IS:

9600 BAUD, HANDSHAKE: YES PARITY: ODD, 8 BIT, 2 STOPBIT

0A>

wählt das Menu für Schnittstelle A, stellt 9600 baud (A) ein, schaltet Handshake auf on (J), setzt Parity odd (M) und definiert 2 Stopbits (R); dann wird zuerst das Schnittstellenmenu (X) und dann CONFIG verlassen (X).

Die Eingabe der Schnittstellenparameter mittels der Eingabezeile ist die einzig sichere Version, die Parameter der Terminal-Schnittstelle zu verändern; alle interaktiven Versuche können zu Störungen des Computersystems führen.

Das Programm COPY ermöglicht es, einzelne Dateien zu kopieren. Die Dateilänge ist durch den zur Verfügung stehenden Speicher beschränkt. (ca 56KBytes bei 64K-Systemen.) COPY wird folgenderweise aufgerufen:

```
COPY zieldatei quelldatei bzw.
COPY laufwerk: quelldatei.
```

COPY erlaubt es, Dateien mit nur einem Laufwerk zu kopieren. Nach Einlesen der Quelldatei ist die Diskette zu wechseln und RETURN einzugeben; erst danach wird die Datei auf die jetzt neu eingelegte Diskette kopiert. Die bei COPY verwendeten Dateinamen entsprechen der unter ZDOS üblichen Schreibweise:
laufwerk:dateiname.dateityp

Ferner besteht die Möglichkeit, mit SYSGEN ganze Disketten zu duplizieren (mit 2 Laufwerken); COPY kann nur einzelne Dateien kopieren. COPY kopiert bei der Eingabe: OA)COPY A: B:*. * nicht etwa alle Dateien von B nach A, sondern nur die erste Datei von B nach A.

COPY-Fehlermeldungen

NO SOURCE FILES	Die Quell-Datei kann nicht gefunden werden; falsche Eingabe
NO DIRECTORY SPACE	Inhaltsverzeichnis der Zieldiskette ist voll, Kopie kann nicht zurückgeschrieben werden
OUT OF DATA SPACE	Zieldiskette ist voll, Kopie kann nicht zurückgeschrieben werden
CAN'T CLOSE DESTINATION	Betriebssystemfehler, kann eigentlich nie vorkommen
SORRY FILE OVERLAPS SYSTEM	-- NO COPY -- Quelldatei zu lang, kann mit COPY nicht kopiert werden, da sie nicht vollständig in den verfügbaren Speicher passt

COPY-Meldungen und Anweisungen

INSERT DESTINATION DISK AND TYPE <CR>

Quelldatei ist eingelesen,
vor dem Schreiben ist die
Zieldiskette einzulegen; nur
falls Quell- und Zieldatei
auf dem selben Laufwerk sind.

INSERT SYSTEM DISK AND TYPE <CR>

Betriebssystemdiskette einle-
gen, anschließend erfolgt
Rückkehr ins Betriebssystem.

COPY COMPLETE

Kopie ist erstellt

Beispiele:

1. Das Programm COPY.COM soll von der Diskette in Laufwerk A auf die in B befindliche Diskette kopiert werden, und dort wieder COPY.COM heißen.

OA>COPY B: A:COPY.COM

COPY COMPLETE

OA>

2. Wie bei 1., nur gleichzeitiges Umbenennen der neuen Datei in TEST.

OA>COPY B:TEST A:COPY.COM

COPY COMPLETE

OA>

3. Kopieren mit nur einem Laufwerk.
Die Datei TEST soll auf eine neue Diskette kopiert werden. Die neue Diskette ist bereits formatiert.

OA>COPY A: A:TEST

INSERT DESTINATION DISK AND TYPE <CR>

(neue Diskette einlegen und
RETURN drücken)

COPY COMPLETE

INSERT SYSTEM DISK AND TYPE <CR>

(Diskette mit ZDOS einlegen
und RETURN drücken)

OA>

Achtung

Bei COPY muß die richtige Diskette angegeben werden; es sucht nicht auf Laufwerk A weiter.

DUMPL0D erstellt auf dem Bildschirm einen formatierten Ausdruck (Dump) einer .LOD-Datei. Aufruf:

0A>DUMPL0D (laufwerk:)dateiname

Beispiel:

0A>DUMPL0D PSP
DUMPL0D VERS 1.2

```
0000 24 0000 C3 03 00 31 5A 0C 2A 06
0009 12 0008 00 22 01 00 21 00 00 22
0012 00 0010 06 00 21 38 00 36 C3 23
001B 09 0018 36 50 23 36 00 21 6C 0C
0024 08 0020 36 00 21 7C 0C 36 50 11
002D 40 0028 DC 06 0E 09 CD 05 00 01
0036 48 0030 72 0C CD 93 02 3A 80 00
003F 10 0038 B7 CA C2 00 11 5C 00 32
0048 40 0040 6C 0C 0E 0F CD 05 00 3C
0051 24 0048 CA 25 05 C3 BB 04 18 72
005A 04 0050 33 33 ED 73 6A 0C 31 6A
```

Eine Zeile besteht aus folgenden Feldern:

1. relative Adresse in der .LOD-Datei
2. 8 Bit Relokatierinformation
3. relative Adresse in der zugehörigen .COM-Datei
4. 8 Byte Maschinencode

Die Relokatierbits geben an, ob das zugehörige Byte Maschinencode absolut (Bit = 0) oder seitenabhängig (Bit = 1) ist. Das höchstwertige Bit der Relokatierinformation gehört zum ersten, das niedrigstwertige Bit zum achten der nachfolgenden Bytes.

In .LOD-Dateien sind nur die Felder 2 und 4 enthalten; die Adressen werden von DUMPL0D beim Ausdrucken berechnet. Die Aufzeichnung erfolgt in 8-Bit Worten. Aufgezeichnet wird in der Reihenfolge:

1 Rel.-Byte, 8 Byte Maschinencode, 1 Rel.-Byte, 8 Byte Maschinencode, usw.

Innerhalb einer .LOD-Datei gilt eine Zeile, d.h. Rel-Byte und die folgenden 8 Byte Maschinencode, die vollständig aus FF besteht, als Marker für das Ende des Programms; sonst gilt auch das Ende der Datei.

DUMPL0D-Fehlermeldungen

NO LOD FILE	die zu druckende Datei kann nicht gefunden werden (auch nicht auf Laufwerk A)
MEMORY OVERFLOW	Datei ist zu lang (länger als ca. 56KByte)

Das Programm **DO** ermöglicht die Verarbeitung von Prozeduren (vgl. auch Kap. 2.0). Eine Prozedur ist eine Kette von Befehlen, die unter einem Namen (Dateityp = **.DO**) auf der Diskette abgelegt werden, und so immer wieder durch Aufruf mit diesem Namen abgearbeitet werden können. Eine Prozedur wird durch folgende Eingabe gestartet:

```
OA>DO prozedur par1 par2 par3 ...
```

Die Parameter **par1** bis maximal **par9** ersetzen die formalen Platzhalter innerhalb der Befehlskette.

Beispiel: Die Befehlskette:

```
DIR $1:
DIR $2:
TYPE $3
```

wird unter dem Namen **INFO.DO** auf der Diskette abgelegt.

Mit dem Aufruf: **DO INFO A B A:SCRIPT.TXT** wird die Prozedur **INFO** ausgeführt. Die Platzhalter **\$1**, **\$2** und **\$3** werden durch **A**, **B** und **A:SCRIPT.TXT** ersetzt. Somit werden jetzt die Befehle:

```
DIR A:
DIR B:
TYPE A:SCRIPT.TXT
```

im KI ausgeführt.

Für das Erstellen von Prozeduren gelten folgende Vereinbarungen:

^P und ^S sind nicht möglich.

\$1 ... \$9 stehen für die Parameter 1 bis 9

%1 ... %9 stehen für evtl. benötigte Parameter

Achtung: Die Parameter mit \$ sollten kleinere Nummern haben als die mit %. So ist z.B. \$1 \$2 %3 möglich, %1 \$2 hingegen nicht sinnvoll.

\$\$ ergibt das Dollarzeichen \$

^char (d.h. (nur hier) zuerst ^ und dann char) ergibt das zugehörige Kontrollzeichen.

^Z (gleichzeitig) beendet die Befehlskette

CARRIAGE RETURN und ein evtl. folgendes LINE FEED beenden eine Prozedurzeile.

^M ergibt Carriage-Return

^C veranlasst ein Warmstart

Eine Schachtelung von Prozeduren ist nicht erlaubt.

DO-Fehlermeldungen

NO PROCEDURE	Die Prozedur existiert nicht
BUFFER OVERFLOW	Befehlskette zu lang
PARAMETER ERROR	Parameterfehler innerhalb der Befehlskette
PARAMETER MISSING	beim Aufruf wurde ein \$-Parameter vergessen

Abbruchkriterien während der Laufzeit einer Prozedur

Alle Dienstprogramme brechen bei schwerwiegenden Fehlern jede Prozedur ab. Bei Eingabe von ^S, ^C am Terminal (STOP und Abbruch-Funktion) wird jede Prozedur abgebrochen. Die Antwort ^C bei einer LEAS oder HEAS-Fehlermeldung bricht Prozeduren ab.

Eine Prozedur kann durch Eingabe von ^S, ^C am Terminal während ihrer Ausführung abgebrochen werden. Ferner können Programme bei Fehlern die Prozedur durch HEAS-Aufruf abbrechen.

Prozeduren können aneinandergehängt werden. So ist z.B. eine endlose Prozedur möglich, die sich selbst immer wieder aufruft:

```

SYSGEN
V                               Vergleich der Disketten in A und B
A
B
^M                             zurück zu ZDOS
DO TEST                        Wiederaufruf der Prozedur TEST.DO

```

FILCOM dient dazu, 2 Dateien zu vergleichen. Mit der Eingabe: `FILCOM datei1 datei2` werden die Dateien `datei1` und `datei2` miteinander verglichen. Als Beispiel sollen die beiden Assembler Quelltexte `TEST.MAC` und `TEST.BAK` miteinander verglichen werden. (`TEST.BAK` ist die alte Version von `TEST.MAC`)

Listing von `TEST.BAK`:

```

        TITLE TEST
;
        LD      DE, MESS
        LD      C, 9                ; PRINT STRING
        CALL    8                    ; ZDOS SERVICE CALL
        RET
;
MESS:    DEFB    0DH, 0AH, 'TEST LAEUFT$'
;
        END

```

Listing von `TEST.MAC`:

```

        TITLE TEST
;
        LD      DE, MESS
        LD      C, 9                ; PRINT STRING
        CALL    5                    ; ZDOS SERVICE CALL
        RET
;
MESS:    DEFB    0DH, 0AH, 'TEST LAEUFT$'
;
        END

```

Vergleich der beiden Dateien:

`0A>FILCOM B:TEST.BAK B:TEST.MAC`

H.K.M. FILE COMPARE PROGRAM

TYPE CR TO EXECUTE

OR ^C TO ABORT

(RETURN eingeben)

(Anfang des Vergleichs)

** 003C: 38 35

(Unterschied gefunden)

** FINISHED **

(Ende des Vergleichs)

TYPE CR TO RETURN TO ZDOS

(RETURN eingeben)

0A>

FILCOM-Fehlermeldungen

** NO FILE **

keine Datei mit diesem Namen

** FILELENGTH NOT EQUAL **

unterschiedliche Längen

** ABORTED **

Vergleich abgebrochen

und

** FINISHED **

normal beendet

FILES ist ein Programm mit folgenden Funktionen:

1. erweitertes Directory-Listing mit Angabe von Dateiname, Lade- bzw. Startadresse, Attributen, Dateilänge in Blocks und Anzahl der Sektoren und Directory-Einträge. Die Dateilänge bezieht sich auf den wirklich belegten Platz auf der Diskette, die Anzahl der Sektoren stimmt nur für sequentielle Dateien, bei RANDOM-Dateien ist RECS die Nummer des letzten Sektors. Beispiel:

OA>FILES B:*.*

FILE	ADR	ACC	SYS/DIR	RECS	LENGTH	ENTRIES
DO .COM	0100	R/W	DIR	5	2K	1
FILCOM .COM	0100	R/W	DIR	7	2K	1
GENCOM .COM	0100	R/W	DIR	4	2K	1
PSP .LOD	0100	R/W	DIR	28	4K	1

BYTES ALLOCATED : 16K BYTES FREE : 372K

2. Ausgabe der Diskparameter des Betriebslaufwerks:
Angezeigt werden die Kapazität, die Anzahl der Einträge im Inhaltsverzeichnis, die Extentlänge, die Blocklänge, die Sektorgröße und die Anzahl der Systemspuren. Beispiel:

OA>FILES DSK:

CAPACITY:	388 KBYTE
DIRECTORY ENTRIES:	64
EXTENT=	32 KBYTE
BLOCK=	2 KBYTE
RECORD=	128 BYTE
RECORDS/TRACK:	40
RESERVED TRACKS:	2

3. Setzen der Datei-Attribute

Dateiattribute sind SYS bzw. DIR und R/W bzw. R/O. SYS bedeutet, daß die Datei nicht im Inhaltsverzeichnis erscheint; DIR hebt SYS wieder auf. R/O bedeutet Nur-Lese-Datei; R/W heißt Schreib/Lese-Datei. FILES Fehlermeldungen

SYNTAX ERROR	falsches Attribut, es gibt nur SYS DIR R/W R/O
NO SUCH FILE	die zu verändernde Datei existiert nicht

Beispiel:

OA>FILES B:ZDOS.TXT =R/O

FILES SET TO R/O

FILE	ADR	ACC	SYS/DIR
ZDOS .TXT	4D00	R/O	DIR

BYTES ALLOCATED : 246K BYTES FREE : 186K

FILES sucht nicht automatisch auf Laufwerk A.

FORMAT ist ein Programm zur Formatierung von soft-sektorierten Disketten; es ist abhängig von der verwendeten Hardware (Controller uPD 765)

Die in dem hier verwendeten System benutzten Disketten sind soft-sektoriert und müssen daher vor Benutzung formatiert werden. Als Format wird **MFM** mit der Sektorlänge von 512 Byte / Sektor verwendet. Auf 5,25" Disketten werden 10, auf 8" Disketten 15 Sektoren pro Spur angeordnet. Die zweite Seite der Diskette wird hinter der ersten angehängt. Die Daten eines jeden Sektors werden invertiert aufgezeichnet. Sektoren werden ab 1, Spuren ab 0 gezählt. Füllerbyte für invertierte Aufzeichnung ist 1A, für normale Aufzeichnung wird E5 verwendet (8" Single Density). Seiten sind 0 (auch für einseitige Disketten) und 1 (zweite Seite).

Ferner wird für 8" Disketten zusätzlich das "Standard CP/M Format" unterstützt, d.h. **FM**-Aufzeichnung (Single Density) mit 26 Sektoren / Spur, 128 Byte Sektorlänge einseitig auf 77 Spuren und **nichtinvertierter** Aufzeichnung.

Da das Format-Programm für alle Versionen gültig sein soll, ist es für die jeweilige Hardware installierbar. Dies geschieht beim ersten Aufruf, ist aber jederzeit später noch zu verändern. Es gibt ein Menu für von ZDOS verwendete Diskettenformate und eine Möglichkeit, ein USER-Diskettenformat zu vereinbaren. Aus dem Menu kann ein Diskettenformat durch Angabe des Kennbuchstabens (A..K) gewählt werden; für das USER-Format werden die benötigten Parameter interaktiv abgefragt (hierbei muß der Benutzer genau über das gewünschte Format Bescheid wissen). Zur Wahl des richtigen Kennbuchstabens sollte man sich an die Meldung des **HEAS** erinnern. Mögliche Meldungen sind:

HEAS VERSION 5.5 DMA-VERSION 5,25"	A oder B
(80 Track)	C oder D
HEAS VERSION 5.5 NON-DMA-VERSION 5,25"	H oder I
HEAS VERSION 5.5 DMA-VERSION 8"	E oder F
HEAS VERSION 5.5 EPC 5,25"	J
(80 Track)	K

Der jeweils erste fett gedruckte Buchstabe gilt für Single-Sided-Laufwerke, der zweite für Double-Sided-Laufwerke. (Dies kann ausprobiert werden: Installiert wird die Double-Sided-Version, falls dann das **FORMAT**-Programm am Anfang der zweiten Seite hängen bleibt, war das Laufwerk Single-Sided. In jedem Fall muß die erste Seite einwandfrei formatierbar sein.)

Den Unterschied zwischen 80 und 40 Track-Laufwerken kann man notfalls folgendermaßen bestimmen: Es wird eine Seite mit 80 Spuren formatiert; anschließend wird mit SYSGEN (siehe dort) versucht, die Spuren 40 bis 80 zu lesen. Wenn dies fehlerfrei möglich ist, handelt es sich um ein 80 Spur Laufwerk, sonst wahrscheinlich um ein 40 Spur-Laufwerk. Das USER-Diskettenformat ist in den Grenzen des uPD 765-Floppy-Controller-ICs frei wählbar (siehe techn. Unterlagen); leichte Modifikationen können durch Auswählen eines Standardformates (A..K) und anschließendes Ändern der eingestellten Parameter erfolgen. (Ein gleichbleibender Wert muß hier neu eingegeben werden, Carriage-Return setzt eine 0 ein.) Das Installieren von FORMAT bezieht sich nur auf das FORMAT-Programm und nicht auf SYSGEN oder HEAS.

Einige Beispiele zur Benutzung von FORMAT:

Die Diskette in Laufwerk B soll formatiert werden. FORMAT ist bereits installiert; es soll das normale Format verwendet werden. Die Hardware besteht aus Double-Sided-40-Track-Laufwerken. Eingaben sind die fett geschriebenen Buchstaben.

```
OA>FORMAT
FORMAT V. 3.3 (01.02.83)
(Type X for Exit)
(B) 5,25", MFM, SS, 40 TRK - FDC at 00
FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = B
FORMATTING TRACK 00 bis 39 für Seite 0
FORMATTING TRACK 00 bis 39 für Seite 1
hit CARRIAGE RETURN to return to ZDOS
OA>
```

Formatieren von nur Spur 5 einer Diskette (FORMAT ist bereits richtig installiert) -- gezählt wird ab 0 --

```
OA>FORMAT
FORMAT V. 3.3 (01.02.83)
(Type X for Exit)
(B) 5,25", MFM, SS, 40 TRK - FDC at 00
FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = T
START FORMATTING AT SIDE (0..1): 0
STARTING TRACK (0..79): 5
STOP FORMATTING AT SIDE (0..1): 0
LAST TRACK (0..79): 5
FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = B
FORMATTING TRACK 05
hit CARRIAGE RETURN to return to ZDOS
OA>
```

Mit dem FORMAT aus dem vorigen Beispiel soll eine einseitige Diskette formatiert werden. Damit ist FORMAT vor der Ausführung umzuinstallieren:

0A>FORMAT

FORMAT V. 3.3 (01.02.83)

(Type X for Exit)

(B) 5,25", MFM, SS, 40 TRK - FDC at 00

FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = I

TYPE A...K TO SELECT, U FOR YOUR OWN LAYOUT

OR X TO CHANGE FDC-BASE-ADDRESS

(A) 5,25", MFM, SS, 40 TRK

(B) 5,25", MFM, DS, 40 TRK

(C) 5,25", MFM, SS, 80 TRK

(D) 5,25", MFM, DS, 80 TRK

(E) 8", MFM, SS, 77 TRK

(F) 8", MFM, DS, 77 TRK

(G) 8" STANDARD CP/M

(H) NON-DMA, MFM, SS, 40 TRK

(I) NON-DMA, MFM, DS, 40 TRK

(J) EPC, MFM, DS, 40 TRK

(K) EPC, MFM, DS, 80 TRK

A

FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = B

FORMATTING TRACK 00 bis 39 für Seite 0

hit CARRIAGE RETURN to return to ZDOS

Es soll ein FORMAT für zweiseitige 35-Track-Mini-Laufwerke erstellt werden. Es gibt 3 Laufwerke im Computersystem: Zuerst ist (B) zu wählen, danach kann die Anzahl der Spuren geändert werden.

0A>FORMAT

FORMAT V. 3.3 (01.02.83)

(Type X for Exit)

not installed, please install - FDC at 00

TYPE A...K TO SELECT, U FOR YOUR OWN LAYOUT

OR X TO CHANGE FDC-BASE-ADDRESS

(A) 5,25", MFM, SS, 40 TRK

(B) 5,25", MFM, DS, 40 TRK

(C) 5,25", MFM, SS, 80 TRK

(D) 5,25", MFM, DS, 80 TRK

(E) 8", MFM, SS, 77 TRK

(F) 8", MFM, DS, 77 TRK

(G) 8" STANDARD CP/M

(H) NON-DMA, MFM, SS, 40 TRK

(I) NON-DMA, MFM, DS, 40 TRK

(J) EPC, MFM, DS, 40 TRK

(K) EPC, MFM, DS, 80 TRK

B

FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = I

TYPE A...I TO SELECT, U FOR YOUR OWN LAYOUT

OR X TO CHANGE FDC-BASE-ADDRESS

(A) 5,25", MFM, SS, 40 TRK
 (B) 5,25", MFM, DS, 40 TRK
 (C) 5,25", MFM, SS, 80 TRK
 (D) 5,25", MFM, DS, 80 TRK
 (E) 8", MFM, SS, 77 TRK
 (F) 8", MFM, DS, 77 TRK
 (G) 8" STANDARD CP/M
 (H) NON-DMA, MFM, SS, 40 TRK
 (I) NON-DMA, MFM, DS, 40 TRK
 (J) EPC, MFM, DS, 40 TRK
 (K) EPC, MFM, DS, 80 TRK

U

(H) = HEX, (D) = DEC, (S) = STRING INPUT WANTED !

RETRY-COUNTER (D) = 3 = 3

FM (O) OR MFM (I) : 01 = 1 (MFM)

SINGLE (O) OR DOUBLE (I) SIDED : 01 = 1 (zweiseitig)

TRACKS / SIDE (D) = 40 = 35 (35 Spuren pro Seite)

FORMAT GAP LENGTH (H) = 1E = 1E (1EH für 5,25")

FILLER BYTE (H) = 1A = 1A (invertierte Aufzeichnung)

N (SECTOR-SIZE) = 02 = 2 (512 Byte Sektorlänge)

(0=128, 1=256, 2=512, 3=1024 Bytes Sektorlänge)

SECTORS / TRACK (D) = 10 = 10 (10 Sektoren pro Spur)

R/W-GAP-LENGTH (H) = 10 = 10 (10H für 5,25")

NUMBER OF DISKS = 2 = 3 (3 Laufwerke im System)

1 STEP (O) or 2 STEPS (I) = 0 = 0 (normal)

(eine 1 hier emuliert ein 40 Spur-Laufwerk auf einem 80 Spur-Laufwerk)

NAME FOR THIS FORMAT = 5,25", DOUBLE SIDED, 35 TRACK

FORMAT DISK (A..D or I for INSTALL or T for SETTRACK) = B

FORMATTING TRACK 00 bis 34 für Seite 0

FORMATTING TRACK 00 bis 34 für Seite 1

hit CARRIAGE RETURN to return to ZDOS

to save this Version, type: SAVE 15 FORMAT.COM

0A>SAVE 15 FORMAT35.COM

ab jetzt ist die so installierte Version FORMAT35 jederzeit wieder aufrufbar und ohne weitere Installation benutzbar.

Ein falsch installiertes FORMAT hängt das System auf GENCOM ist ein Lader für Page-relokätierbare Dateien (Dateityp = .LOD). Da das Betriebssystem seitenweise (d.h. in Schritten zu 256 Byte) relokätierbar ist, muß vor dem Kopieren auf die Systemspuren aus der .LOD-Datei eine .COM-Datei erstellt werden. Hierzu wird GENCOM aufgerufen:

GENCOM laufwerk:dateiname \$adresse

GENCOM erstellt eine Datei mit dem Namen dateiname.COM auf der durch laufwerk selektierten Diskette, bzw. der Diskette A, falls die .LOD-Datei auf laufwerk nicht gefunden werden konnte.

Folgende Fehler können gemeldet werden:

** NO MEMORY **	oberhalb der Adresse OFFFHH
** CAN'T SAVE COM-FILE **	
	die Diskette ist voll
** NO LOD-FILE **	dateiname.LOD existiert nicht
** NO ADDRESS **	die Angabe \$adresse fehlt
** MEMORY OVERFLOW **	die Datei ist zu lang

Das in LOD-Dateien verwendete Format wird bei DUMPLOD (vgl. Kap. 2.3) näher beschrieben.

Prinzipiell kann Software im LOD-Format nicht direkt ausgeführt werden; sie muß in jedem Fall erst mittels GENCOM in ein ausführbares Format (COM-Datei) umgeformt werden.

Achtung: die Meldung ** NO MEMORY ** erscheint auch bei falschem .LOD-Code. Die GENCOM-Version von ZDOS 5.a kann LOD-Files der ZDOS-Version 5.b nicht richtig verarbeiten (Ende-Kennung ist ab jetzt neu in LOD-Dateien).

Die Ende-Kennung besteht aus einem Rel-Byte FF und 8 folgenden Maschinencode-Bytes FF. (vergleiche DUMPLOD).

Achtung beim Generieren von Dienstprogrammen

Die höchste Anfangsadresse für ZDOS, d.h. KI, LEAS und HEAS ist 0E000H, da sich alle Teile auf den KI-Anfang beziehen. PSP kann dann maximal an Adresse 0D300H beginnen.

IO ist ein Hilfsprogramm zur Zuordnung der logischen zu den physikalischen Geräten. Folgende Eingaben sind möglich: (vgl. Die IOBYTE-Funktion)

OA> IO erstellt eine Liste der aktuellen Zuordnungen.

OA> IO ldev:=pdev: ordnet dem logischen Gerät ldev: das physikalische Gerät pdev: zu.

Beispiele:

OA> IO (Zuordnungsliste)

```
CON: IS CRT: (TTY:, CRT:, BAT:, UC1:)
RDR: IS TTY: (TTY:, PTR:, UR1:, UR2:)
PUN: IS TTY: (TTY:, PTP:, UP1:, UP2:)
LST: IS LPT: (TTY:, CRT:, LPT:, UL1:)
```

OA> IO LST:=TTY: (Zuweisung)

```
CON: IS CRT: (TTY:, CRT:, BAT:, UC1:)
RDR: IS TTY: (TTY:, PTR:, UR1:, UR2:)
PUN: IS TTY: (TTY:, PTP:, UP1:, UP2:)
LST: IS TTY: (TTY:, CRT:, LPT:, UL1:)
```

Logische Geräte ldev: sind: CON:, RDR:, PUN:, LST:.

IO-Fehlermeldungen

```
END OF LINE : xx?       Zeile ist unvollständig
xx?                      Gerätenamen sind nicht zulässig
```

Im HEAS verwendete Zuordnung zwischen physikalischen Geräten pdev: und Schnittstellen auf der CPU-Karte:

```
TTY   =   SIO Port A
CRT   =   SIO Port B
PTR   =   .....      (frei)
PTP   =   .....      (frei)
LPT   =   PIO Centronics (8 bit)
```

```
UC1   =   SIO Port B
```

```

UR1   =   SIO Port B
UR2   =   SIO Port B   (8 bit)
UP1   =   SIO Port B
UP2   =   .....      (frei)
UL1   =   .....      (frei)

```

Speichertestprogramm

MTEST ist ein Speichertestprogramm, das den Speicher ab Adresse 2000H bis zu Adresse FFFFH testet. Hierzu werden quasizufällige Muster in den Speicher geschrieben, und nach Beschreiben des gesamten Speichers wieder gelesen und verglichen. MTEST findet fast alle in Speichern vorkommenden Fehler. MTEST überschreibt beim Testen des Speichers das Betriebssystem und muß daher eigene Ein-Ausgabe-Routinen benutzen. MTEST arbeitet nur mit dem Terminal, das an Port B der SIO angeschlossen ist. Nach dem Aufrufen meldet sich das Programm mit:

MEMORY TEST PROGRAM

Anschließend wird ein statischer RAM-Test durchgeführt.

STATIC TEST RUNNING

Nach Beendigung des statischen Speichertest:

STATIC TEST COMPLETE

beginnt der dynamische Speichertest mit insgesamt 256 Durchläufen. Jeder erfolgte Durchlauf wird durch einen Punkt auf dem Terminal dokumentiert.

DYNAMIC TEST RUNNING

Nach Beendigung des letzten der 256 Durchläufe erfolgt die Meldung:

DYNAMIC TEST COMPLETE

Anschließend geht das Testprogramm in eine Endlosschleife. Zum Weiterarbeiten mit dem System ist jetzt der RESET-Taster zu betätigen und damit das System neu zu starten.

Von MTEST gefundene Fehler im Speicher werden folgenderweise angezeigt:

*** ERROR *** adresse SHOULD BE data1 IS data2 OR data3

D.h. an der angegebenen Adresse wurde statt data1 ein anderer Wert data2 oder data3 gelesen. Unterschiede zwischen data2 und data3 deuten auf fehlende bzw. nicht ansprechbare Speicherchips hin. Zum Deuten anderer Fehler sei hier der Algorithmus zur Erzeugung der Quasizufallsfolge angegeben:

```

;
;***** DYNAMIC TEST *****
;
DYNTEST: LD      B,0          ; Laufzähler
STRT:    LD      HL,2000H     ; Startadresse
FILL:    LD      A,L          ; Berechnung Quasizufalls-
        XOR      H           ; folge L xor H xor B
        XOR      B           ; mit HL = Adresse und
        LD       (HL),A      ; B = Durchlaufzähler
        INC      HL
        LD       A,H
        CP       0           ; High(Endadresse+1)
        JR       NZ,FILL

```

0. PSP-Aufruf

Es gibt 2 Möglichkeiten, den PSP aufzurufen:

```
OA>PSP
OA>PSP (laufwerk:)dateiname.typ
```

Beide Aufrufe starten den PSP, der zweite Aufruf lädt vorher noch die Datei (laufwerk:)Dateiname.typ an die in ihrem FCB vereinbarte Adresse im Speicher. Diese Eingabe ersetzt die Befehle:

```
>I(laufwerk:)dateiname.typ
>R.
```

Es ist darauf zu achten den Dateityp .COM bei ausführbaren Programmen beim Aufruf mitanzugeben. Die Eingabezeile:

```
OA>PSP__
veranlasst die Meldung: FILE NOT FOUND, da nach einer Datei mit dem Namen blank gesucht wird, die es unter ZDOS nicht geben kann!
```

1. Einleitung

PSP stellt ein Hilfsmittel dar, um interaktiv Maschinenprogramme sowie den Zustand des Rechners zu testen und zu kontrollieren. Nach der Anmeldung (SIGN ON MESSAGE) verlangt PSP Eingaben mit '>', falls das zuletzt eingegebene Kommando erfolgreich ausgeführt wurde und mit '?', falls das zuletzt eingegebene Kommando wegen syntaktisch falscher Eingabe nicht ausgeführt werden konnte.

Jedes Kommando kann bis zu 80 Zeichen lang sein (das Zeilenende wird als 81-tes Zeichen automatisch hinzugefügt), wobei das erste Zeichen das Kommando darstellt. Es gibt folgende Kommandos:

Ybyte1 byte2 ...
Byte-String im Speicher lokalisieren

Danfang ende
Speicherausdruck (DUMP) in hex und ASCII

Fanfang ende wert
Füllen des Speichers mit konstanten Werten

Gstart stop
Ausführung (GO) eines Maschinenprogramms

Lanfang ende
Disassemblierung (LIST) des Speicherinhaltes

Mvon bis nach
Verschieben (MOVE) von Speicherinhalten

Xregister
Register-Anzeige und Veränderung

Sadresse
Ersetzen (SUBSTITUTE) von Speicherinhalten

Tvon bis mode
Schrittweise Überwachung (TRACE) von Programmen

Vanfang1 anfang2
Vergleichen (VERIFY) zweier Speicherbereiche

Itext
Eingabe von Dateinamen

Radresse
Datei einlesen

Wvon bis
Datei abspeichern

Das Kommandozeichen benötigt, je nach Kommando, bis zu drei Hexadezimalzahlen, die mit einem beliebigen Trennzeichen (alle Zeichen außer 0..9 und A...F) abgeschlossen werden können.

2. Die PSP-Kommandos

Die einzelnen Kommandos werden nachfolgend beschrieben. Bei der Beschreibung der Kommandos werden die hexadezimalen Zahlen durch 'hhhh' mit Leerzeichen getrennt dargestellt. Sie dürfen von einem bis zu vier hexadezimale Zeichen enthalten (Längere Zahlen werden automatisch auf die vier letzten gekürzt).

2.1 Byte-String im Speicher lokalisieren

Mit dem Y-Kommando wird der gesamte Speicher nach dem Vorkommen eines Bytestrings durchsucht. Ausgegeben wird die Adresse des Stringanfangs.
Das Format ist:

```
>Yhh hh hh hh .....
```

2.2 Ausdrucken des Speicherinhaltes

Mit dem D-Kommando kann man Speicherinhalte in hexadezimaler und zeichenweiser (ASCII) Darstellung ausdrucken. Die möglichen Formate sind:

```
>Dhhhh hhhh
>Dhhhh
```

Im ersten Fall wird der Speicher zeilenweise von der Anfangs- bis zur Endadresse ausgegeben. Im zweiten Fall werden ab hhhh 16 Zeilen dargestellt.

```
>D1005 1009
      0 1  2 3  4 5  6 7  8 9  A B  C D  E F
1000 1234 1235 1236 1237 1238 1239 1241 1242 #.4.5.6.7.8.9.A.B
```

In ASCII nicht darstellbare Speicherinhalte werden als '.' dargestellt. Das höchstwertige Bit wird bei der ASCII-Darstellung nicht beachtet.

2.3 Das Laden des Speichers mit Konstanten

Das F-Kommando hat die Form:

>Fhhhh hhhh hh

wobei die erste Hexadezimalzahl die Anfangsadresse, die zweite die Endadresse und die dritte eine Bytekonstante, mit der der Speicherbereich beschrieben werden soll, ist. Man beachte:

- Falls die zweite Adresse kleiner als die erste ist, wird über FFFF hinaus bis zu der angegebenen Adresse der Speicher gelöscht.
- Das ZDOS-Betriebssystem wird zwar von Platte geladen, jedoch ist nach Löschen des Bereichs 0000-0100 bzw. E800-FFFF das System nur noch mit Hardware-RESET zu laden.

2.4 Das Starten von Benutzerprogrammen

Mit dem Kommando G wird ein Programm gestartet. Das Kommando kann folgende Gestalt annehmen:

>Ghhhh
>Ghhhh hhhh
>G, hhhh

Die erste Form startet ein Benutzerprogramm, wobei die Kontrolle dem Monitor entzogen wird. Die zweite Form startet ein Benutzerprogramm an der angegebenen Adresse und führt es bis zur zweiten aus, wonach die Kontrolle wieder an den Monitor gelangt. Man stelle sicher, daß die zweite Adresse auch erreicht wird. Die dritte Form dient dazu, fortlaufend von der zuletzt erreichten Adresse bis zur durch hhhh angegebenen auszuführen.

2.5 Die Disassemblierung von Programmen

Mit dem L-Kommando lassen sich Programme im Speicher disassemblieren. Die Form ist:

>Lhhhh
>Lhhhh hhhh

Die erste Zahl gibt die Anfangsadresse, die zweite (falls vorhanden) die Endadresse an. Jederzeit kann die Disassemblierung durch Eingabe eines beliebigen Zeichens abgebrochen werden. Bei fehlender Endadresse werden 20 Bytes disassembliert.

2.6 Das Verschieben von Speicherinhalten

Mit dem M-Kommando werden Speicherinhalte verschoben. Die Form ist:

```
>Mhhhh hhhh hhhh
```

Die drei angegebenen Adressen sind: von, bis, nach. Man beachte, daß im Gegensatz zu den meisten Implementierungen sich die Bereiche ohne Schaden überlappen dürfen.

2.7 Registerausgabe und Änderung

Mit dem Kommando X werden die Register angezeigt und bei Hinzufügung eines Registernamens verändert. Registernamen sind: S, A, B, D, H, X und Y. Als Register werden Doppelregister (16 bit) betrachtet.

Ein Beispiel zur Anzeige:

```
>X
PC (PC) SP AF BC DE HL IX IY (BC DE HL IX IY)
1234 00 2345 3456 4567 5678 6789 7890 8901 00 00 00 00 00
>
```

Ein Beispiel zur Veränderung von Registern:

```
>XA
0000 1234 (d.h. A=12, F=34)
>
```

2.8 Das Verändern von Speicherinhalten

Mit dem S-Kommando werden Speicherinhalte einzeln angezeigt und möglicherweise geändert. Das Kommando hat die Form:

```
>S1FFC
1FFC 21=
1FFD 00=11
1FFE 11=00
1FFF C3=
2000 FF=00
?
```

Folgendes ist geschehen:

Der Inhalt von 1FFC wird als 21 angezeigt. Das eingegebene RETURN läßt ihn unverändert. Der Inhalt von 1FFD war 00 und wird zu 11 geändert. Der Inhalt von 1FFE war 11 und wird zu 00 geändert. Der Inhalt von 1FFF bleibt unverändert C3. Der Inhalt von 2000 wird als FF angegeben, jedoch war der Versuch ihn zu 00 zu ändern nicht erfolgreich, da der Monitor sich mit '?' meldet. Der Grund ist einfach: Der Speicherbereich ab 2000 war schreibgeschützt. Verlassen wird das S-Kommando durch Eingabe von Leerzeichen und dann Carriage Return.

2.9 Das schrittweise Überwachen von Programmen

Mit dem T-Kommando kann auf schnellste Weise die Ausführung eines Maschinenprogrammes verfolgt werden. TRACE überwacht nur Hauptprogramme; Unterprogramme werden immer in Echtzeit ohne Überwachung durchgeführt. Das Kommando hat die Gestalt:

>Thhhh hhhh m

Dabei bedeuten die ersten beiden Hexadezimalzahlen die Adressen von bis, das m den Modus. Es gibt drei verschiedene Betriebsarten (Modi):

m=2 Es wird ununterbrochen der angegebene Bereich ausgeführt und überwacht

m=1 Nach jeder Seite (12 Zeilen) wird die überwachte Ausführung unterbrochen. Sie kann mit RETURN fortgeführt werden oder mit 'M' beendet werden.

m=0 (oder andere Eingabe)
Es wird ähnlich wie bei der seitenweisen Überwachung nach jedem Befehl mit der Anforderung einer Eingabe unterbrochen.

>T1000 1011 0

1000	LD	HL,2121	FF80 0000 0000 0000 2121 0000 0000
1003	JP	NZ,100A	FF80 0000 0000 0000 2121 0000 0000
M			
>			

Erläuterungen zum Ausdruck:

Die erste Zahl stellt die Adresse des auszuführenden Befehls dar; darauf folgt die mnemotechnische Darstellung des Befehls. Die nachfolgende Kolonne gibt die Registerinhalte in der Reihenfolge: SP, AF, BC, DE, HL, IX, IY nach der Ausführung des Befehls wieder.

Man beachte:

- Da die Befehle nicht simuliert sind, sondern von der Z80-CPU durchgeführt werden und mit Stolperstein die Analyse anschliessend durchgeführt wird, sind einige Punkte zu beachten.
- Unterprogrammaufrufe werden nicht verfolgt, sie sind also gegebenenfalls getrennt zu prüfen.
- Die Gruppe der RESTARTS kann nicht überwacht werden und führt zu einem Abbruch mit Rückkehr in den Monitor.
- Die Gruppe der RETURNS wird nicht ausgeführt, im bedingten Fall führt eine zutreffende Bedingung wie bei einem unbedingten RETURN zum Abbruch.
- Die Bereiche des PSP können nicht überwacht werden.
- Programme im ROM oder EPROM können nicht überwacht werden.
- Befehle, die den Stackpointer umladen, sind nicht zulässig. (z.B. EX IY,SP)

2.10 Der Vergleich von Speicherinhalten

Mit dem V-Kommado werden zwei angegebene Speicherbereiche miteinander verglichen. Die Form ist:

>V8000 9000

Die angegebenen Bereiche werden byteweise miteinander verglichen. Dies geschieht solange, bis sich die Inhalte unterscheiden. Darauf werden die Inhalte ausgegeben und auf eine Eingabe gewartet. Mit RETURN wird der Vergleich fortgesetzt, mit jeder anderen Eingabe (vornehmlich Zwischenraum) gelangt man wieder in den Monitor. Falls die beiden Adressen gleich sind entspricht dies einer unendlichen Schleife!

2.11 Das I-Kommando

Das I-Kommando erfüllt zwei Aufgaben.

1. Vorbereitung für Lesen oder Schreiben.

```
> ITEST.COM  
> R
```

bewirkt, daß die Datei TEST.COM geladen wird.

2. Setzen der Speicherbereiche 005C ... 007F (vor-
eingestellter FCB) und 0080 ... 00FF (Eingabe-
zeile). In dieser Form ersetzt das I-Kommando die
Eingabe im KI. Dies ermöglicht es, Programme zu
testen, die die KI-Eingabezeile benutzen.

```
> ITEST.COM  
> R  
> I DATEI1 DATEI2 A B C  
> Gstartadresse
```

ersetzt die KI-Eingabe:

```
OA> TEST DATEI1 DATEI2 A B C
```

2.12 Einlesen einer Datei

Mit dem R-Kommando wird eine mittels I-Kommando
angegebene Datei geladen. Der PSP überwacht hierbei
den verfügbaren Speicher. Das R-Kommando hat die
Form:

```
> R  
> R hhhh
```

hhhh ist die Adresse, ab der die Datei geladen
werden soll. Achtung: Die beiden letzten Stellen von
hhhh werden als 00 angenommen. R allein lädt die
Datei an die im FCB vereinbarte Adresse. (siehe auch
unten).

2.13 Abspeichern einer Datei

Das W-Kommando lädt einen Speicherbereich auf die
Diskette. Es hat die Form:

```
> W h h 00 h h F F
```

Unter dem mittels I-Kommando vereinbarten Dateinamen
wird der Bereich zwischen Start- und Endadresse auf
Diskette abgelegt.

Das Programm SYSGEN erfüllt 3 Funktionen:

- Betriebssystem kopieren (Spur 0 und 1)
- ganze Disketten (oder einzelne Spuren) kopieren
- Vergleich von Disketten (nach dem Kopieren)

Da die Systemspuren unter dem ZDOS-Dateisystem nicht erreichbar sind, ist zum Kopieren des Betriebssystems ein besonderes Kopierprogramm erforderlich. Diese Aufgabe wird von SYSGEN übernommen. SYSGEN ist ein Kopierprogramm, das sowohl das Betriebssystem, wie auch ganze Disketten oder nur bestimmte Spuren kopieren oder vergleichen kann. Es ist genauso wie FORMAT (vgl. dort) installierbar. Folgende Beispiele verdeutlichen die Benutzung von SYSGEN:

Installierung I für einseitige Mini-Disketten A; nur Betriebssystem kopieren S von A nach B; das anschließende RETURN übergibt die Kontrolle wieder an ZDOS.

0A>SYSGEN

SYSGEN VERS. 3.2 (26.10.82)

(TYPE X FOR EXIT)

(B) 5,25", MFM, DS, 40 TRK

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
SETTRACK (T) OR INSTALL (I) I

TYPE A...I TO SELECT, U FOR YOUR OWN LAYOUT
OR X TO CHANGE FDC-ADDRESS

(A) 5,25", MFM, SS, 40 TRK

(B) 5,25", MFM, DS, 40 TRK

(C) 5,25", MFM, SS, 80 TRK

(D) 5,25", MFM, DS, 80 TRK

(E) 8", MFM, SS, 77 TRK

(F) 8", MFM, DS, 77 TRK

(G) 8" STANDARD CP/M

(H) 5,25", NON-DMA, SS

(I) 5,25", NON-DMA, DS

A

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
SETTRACK (T) OR INSTALL (I) S

SOURCE DISK (OR CR TO COPY FROM MEMORY, STARTING AT 1000H) = A

DESTINATION DISK (OR X TO SKIP WRITING) = B

HIT CARRIAGE RETURN TO RETURN TO ZDOS

TO SAVE SELECTED VERSION TYPE: SAVE 15 SYSGEN.COM

Kopieren von nur Spur 2; SYSGEN ist richtig installiert:
 -- gezählt wird bekanntlich ab Spur 0 --

0A>SYSGEN

SYSGEN VERS. 3.2 (26.10.82)

(TYPE X FOR EXIT)

(B) 5,25", MFM, DS, 40 TRK

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
 SETTRACK (T) OR INSTALL (I) T

START COPYING AT SIDE : 0

STARTING TRACK : 2

STOP COPYING AT SIDE: 0

LAST TRACK : 2

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
 SETTRACK (T) OR INSTALL (I) C

SOURCE DISK = A

DESTINATION DISK = B

COPYING TRACK 02

HIT CARRIAGE RETURN TO RETURN TO ZDOS

TO SAVE SELECTED VERSION TYPE: SAVE 15 SYSGEN.COM

Ein gültiges Betriebssystem ist bereits im Speicher ab Adresse 1000H geladen und soll auf die Diskette B geschrieben werden.

0A>SYSGEN

SYSGEN VERS. 3.2 (26.10.82)

(TYPE X FOR EXIT)

(B) 5,25", MFM, DS, 40 TRK

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
 SETTRACK (T) OR INSTALL (I) S

SOURCE DISK (OR CR TO COPY FROM MEMORY, STARTING AT 1000H) =

DESTINATION DISK (OR X TO SKIP WRITING) = B

HIT CARRIAGE RETURN TO RETURN TO ZDOS

TO SAVE SELECTED VERSION TYPE: SAVE 15 SYSGEN.COM

0A>

Das Betriebssystem kann auf mehrere Methoden in den Speicher gelangen.

- mit der Prozedur GEN
- mit SYSGEN von einer Diskette
- Mit dem PSP mit dem R1000-Befehl
- Im KI durch LOAD ZDOS.COM \$1000

Eine einseitige Mini-Diskette soll kopiert werden und anschließend geprüft werden. Also: Kopieren auswählen C, von A nach B und Rückkehr ins Betriebssystem RETURN. Danach wieder SYSGEN aufrufen, Verify wählen V von A nach B.

OA>SYSGEN (kopieren)

SYSGEN VERS. 3.2 (26.10.82)

(TYPE X FOR EXIT)

(A) 5,25", MFM, SS, 40 TRK

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
SETTRACK (T) OR INSTALL (I) C

SOURCE DISK FOR COPY = A

DESTINATION DISK = B

COPYING TRACK 00 (zählt bis 39 hoch)

HIT ANY KEY TO RETURN TO ZDOS

TO SAVE SELECTED VERSION TYPE: SAVE 15 SYSGEN.COM

OA>SYSGEN (vergleichen)

SYSGEN VERS. 3.2 (26.10.82)

(TYPE X FOR EXIT)

(A) 5,25", MFM, SS, 40 TRK

SELECT: SYSGEN (S), COPY (C), VERIFY (V),
SETTRACK (T) OR INSTALL (I) V

SOURCE DISK = A

DESTINATION DISK = B

VERIFY TRACK 00 (zählt bis 39 hoch)

HIT ANY KEY TO RETURN TO ZDOS

TO SAVE SELECTED VERSION TYPE: SAVE 15 SYSGEN.COM

OA>

Unter ZDOS sind wie unter CP/M 16 Benutzer (USER) zugelassen. Das heißt nicht, daß ZDOS oder CP/M ein Multi-User-Betriebssystem ist. Die unterschiedlichen Benutzer können nie gleichzeitig arbeiten. Um einen gewissen Schutz zwischen ihnen zu gewährleisten, existieren die 16 User-Areas. In ZDOS sind Systemprogramme als Dateien des Benutzers 0 gespeichert. Diese sind allgemein benutzbar, können aber nur vom Benutzer 0 gelöscht oder beschrieben bzw. verändert werden. Der normale Anwender arbeitet als Benutzer 1 bis 15. Seine Dateien sind nur für ihn zugänglich. Als Schutz muß ein Password eingegeben werden. Hierzu dient das Programm USER:

OA>USER nummer password

Das Setzen der Benutzernummer "nummer" wird nur dann ausgeführt, wenn das richtige "password" miteingegeben wurde.

Beim Ausliefern des Systems existieren keine "passwords". Sie können beim Wechsel der Benutzernummer vereinbart werden. (Dazu muß die Systemdiskette beschreibbar sein; WRITE-PROTECT ist also vorher zu entfernen). Ein beim Wechsel in einen Benutzerbereich benutztes "password" gilt als gesetzt, und muß ab jetzt immer zum Wechsel benutzt werden.

Ein Beispiel:

Beginnend mit der Originaldiskette (keine "passwords") wird

OA>USER 2

eingegeben. Das System meldet sich jetzt mit:

2A>USER 4 otto

Ab jetzt ist als "password" für USER 4 "otto" vereinbart; d.h. wenn jetzt nach einem Kaltstart der Befehl:

OA>USER 4

eingegeben wird, erscheint folgende Fehlermeldung:

** WRONG PASSWORD **

OA>

Der Benutzerbereich wird nicht geändert. Es bleibt der Benutzerbereich 0 erhalten.

Der Befehl USER 0 ist immer möglich. Er wird nicht geschützt.

-- Achtung --

Ein einmal gesetztes password kann nicht wieder gelöscht oder geändert werden. Die einzige Möglichkeit besteht darin, das unveränderte Programm USER wieder neu von der ZDOS-Systemdiskette zu holen (ohne passwords)

Eine Prozedur ist eine Kette von Befehlen, die unter einem Namen auf der Diskette abgelegt ist. Sie kann durch DO name jederzeit aufgerufen werden. DO ist da Programm, das sie von der Diskette liest und für die Verarbeitung durch den KI vorbereitet. (vgl. Kap. 1.7)
Mit ZDOS werden 2 Prozeduren mitgeliefert:

Die Systemgenerierungsprozedur GEN.DO

Die Prozedur GEN.DO dient zur Systemgenerierung. Sie erstellt ein Betriebssystem und schreibt es auf die Diskette in Laufwerk B.

Aufruf:

```
OA>DO GEN version $adresse side
```

```
mit:  version =  ZDOSxx mit xx wie unten be-
          adresse = Anfangsadresse des Kommandointer-
          und: side = S für single-sided-Laufwerke
                   bei double-sided entfällt die
                   Angabe S
```

Erläuterung von xx:

xx	Diskettenversion		
5N4	2 * 5"	NON-DMA	DD, DS 40 TRACK
5N8	2 * 5"	NON-DMA	DD, DS 80 TRACK
5N84	2 * 5"	NON-DMA	DD, DS 80/40 TRACK
5D4	2 * 5"	DMA	DD, DS 40 TRACK
5D8	2 * 5"	DMA	DD, DS 80 TRACK
5D84	2 * 5"	DMA	DD, DS 80/40 TRACK
5E4	2 * 5"	EPC	DD, DS 40 TRACK
5E8	2 * 5"	EPC	DD, DS 80 TRACK
5E84	2 * 5"	EPC	DD, DS 80/40 TRACK
8	2 * 8"	DD, DS	
8S	1 * 8"	DD, DS + 1 * 8"	SD, SS

Die normale Adresse fuer ZDOS ist \$E000 (bei Optionen verschiebt sie sich nach vorne)! Wenn ZDOS ab Adresse 1000H in den Speicher geladen wird, gelten folgende Adressen:

KI	1000 bis 17FF
LEAS	1800 bis 23FF
HEAS	2400 bis 2DFF

Achtung! Der HEAS-Datenbereich und die Interrupt-Tabelle werden nicht gespeichert. Geladen werden von ZDOS nur 7,5 KByte.

Es gibt keine reine Single-Density Version!

Die Initialisierungsprozedur INIT.DO

INIT.DO ist die Prozedur, die bei jedem Kaltstart des Systems automatisch aufgerufen wird. Hierin können beliebige Programme aufgerufen werden; z.B. CONFIG, IO und auch andere Benutzerprogramme.

INIT.DO wird als leere Prozedur geliefert. Sie ist vom Benutzer an die jeweiligen Gegebenheiten anzupassen.

Einige Beispiele:

1. Serieller Drucker mit Handshake bei 9600 Baud:

```
>> START INIT <<  
ID LST:=TTY:  
CONFIG AAJXX  
>> STOP INIT <<
```

2. Maschinensprachenentwicklungssystem, direkter Aufruf von PSP:

PSP	Monitor Aufruf
XS	SP auf 3000H setzen
3000	
X	Register anzeigen

Zur Ausführung der Prozedur INIT.DO sind auf der Diskette in Laufwerk A folgende Dateien erforderlich:

```
DO.COM  
INIT.DO  
und die in INIT.DO aufgerufenen Programme
```

Im HEAS ist die wahlweise mögliche IOBYTE-Funktion implementiert. Das IOBYTE gibt die Zuordnungen zwischen logischen und physikalischen Geräten an. Es betrifft die Konsole (CON), den Drucker (LST) und den Lochstreifenleser (RDR) und -stanzer (PUN). Als IOBYTE wird der Inhalt der Adresse 0003 interpretiert. Das IOBYTE besteht aus 4 Feldern zu je 2 Bit in folgender Reihenfolge:

	LIST		PUNCH		READER		CONSOLE	
Bit	7	6	5	4	3	2	1	0
	MSB				LSB			

Es gelten folgende Zuordnungen:

CON:	00	TTY:	als Konsole
	01	CRT:	als Konsole
	10	BAT:	Batch Betriebsart, d.h. RDR: als Eingabegerät und LST: als Ausgabegerät
RDR:	11	UC1:	als Konsole
	00	TTY:	als Streifenleser
	01	PTR:	als Streifenleser
	10	UR1:	als Streifenleser
PUN:	11	UR2:	als Streifenleser
	00	TTY:	als Streifenstanzer
	01	PTP:	als Streifenstanzer
	10	UP1:	als Streifenstanzer
LST:	11	UP2:	als Streifenstanzer
	00	TTY:	als Drucker
	01	CRT:	als Drucker
	10	LPT:	als Drucker
	11	UL1:	als Drucker

hierbei ist TTY eine Teletype
 CRT ein Datensichtgerät
 PTR ein schneller Streifenleser
 PTP ein schneller Streifenstanzer
 LPT ein Zeilendrucker

dagegen sind UC1, UR1, UR2, UP1, UP2 und UL1 vom Anwender jeweils zu implementierende Geräte, deren Namen festliegen.

Ein Beispiel:

Beim Kaltstart des Systems wird das IOBYTE = 81H gesetzt, d.h.

LST: = LPT: (10)
 PUN: = TTY: (00)
 RDR: = TTY: (00)
 CON: = CRT: (01)

Die einfachste Möglichkeit, eine ZDOS-Diskette zu erstellen, ist eine vorhandene ZDOS-Systemdiskette der gleichen Version zu duplizieren. Damit werden aber teilweise zu viele Dateien übertragen. Deshalb sollten die überflüssigen (d.h. die in einem speziellen Anwendungsfall nicht unbedingt benötigten) Dateien anschließend gelöscht werden.

Folgende Dateien werden fast immer benötigt:

SYSTEM.HLP	Hilfsteixe des Systems
USER.COM	Benutzerbereich wechseln
DO.COM	Prozeduren starten
+ die benötigten Prozeduren	?????????.DO

empfohlen wird dann noch:

FILES.COM	
CONFIG.COM	je nach System
IO.COM	je nach System

Die restlichen Dateien hängen vom Anwendungsfall ab.

Ein Beispiel:

neue Diskette formatieren (FORMAT)
System von vorhandener Systemdiskette kopieren
(SYSGEN S)
Dateien kopieren (entweder alle mit SYSGEN C oder
nur bestimmte mit COPY b: a:dateiname)
anschließend neu erstellte Diskette in Laufwerk A
prüfen (Kaltstart!)

Folgende Schritte sind zur Inbetriebnahme von ZDOS erforderlich:

1. Es ist sicherzustellen, daß die verwendete Hardware, also der Computer funktioniert.

Benötigt wird zusätzlich zum Rechner ein Terminal mit V24-Schnittstelle, das an das Port B der SIO bzw. des DART angeschlossen werden muß. Seine Baudrate kann 1200, 2400, 4800 oder 9600 Baud betragen. Diese 4 Baudraten werden vom Bootstraploader erkannt. Im BOOT werden die Handshake-Leitungen des Ports nicht beachtet.

2. Das mit ZDOS zusammen gelieferte EPROM vom Typ 2716 mit dem Aufkleber **BOOT641** oder einer höheren Versionsnummer ist auf Adresse 0 in den Computer einzusetzen.

3. Der Computer ist einzuschalten. Jetzt meldet sich nach Eingeben von mehreren Leerzeichen am Terminal der Urlader **BOOT641**; die Baudrate des Terminals wird angezeigt, nach Durchführung eines einfachen Speichertests wird die vorhandene Speichergröße in 16k Schritten angegeben und anschließend der Mode des Floppycontrollers bestimmt (DMA oder NON-DMA). Anhand dieser Angaben ist zuerst zu prüfen, ob die ausgelieferte Version von ZDOS auf diesem System lauffähig ist, d.h. die Adresse von ZDOS muß mindestens 2000H kleiner sein als die Speichergröße und bei der NON-DMA-Meldung darf kein ZDOSxD geliefert sein.

Ist dieses sichergestellt, kann ZDOS prinzipiell auf diesem Computer laufen.

4. Jetzt wartet der Computer noch immer darauf, daß eine Diskette in Laufwerk A (das ist das Laufwerk mit der Laufwerksadresse 0) eingelegt wird. Unter der Voraussetzung, daß bereits sichergestellt wurde, daß keinesfalls auf eine schreibgeschützte Diskette geschrieben werden kann, und der Prüfung, daß die ZDOS-Diskette wirklich schreibgeschützt ist (bei 5,25" Disketten schützt der Aufkleber über dem Write-Protect-Loch, bei 8" Disketten darf kein Aufkleber über dem Write-Protect-Loch vorhanden sein), kann jetzt die ZDOS-Diskette in Laufwerk A eingelegt werden.

5. Nach Schließen der Laufwerkstür greift der Urlader auf die Diskette zu und lädt das Betriebssystem. Anschließend meldet sich HEAS mit seiner Versionsnummer. Nach weiteren Diskettenzugriffen meldet sich der KI mit 0A).

Jetzt muß die evtl. vorhandene Boot-LED aus sein; d.h. das EPROM mit dem Urlader wird ab jetzt nicht mehr selektiert.

6. Jetzt ist prinzipiell sichergestellt, daß ZDOS auf diesem Computer lauffähig ist. Als erstes ist jetzt von der Original-ZDOS-Diskette eine Kopie zu machen. (Programm FORMAT und SYSGEN).

7. **Keinesfalls darf jemals auf die Original-ZDOS-Diskette geschrieben werden.** Zum Arbeiten ist immer eine Kopie der Originaldiskette zu verwenden.
8. Die jetzt erstellte Kopie von ZDOS sollte nun auf das jeweilige System angepaßt werden. Hierzu ist erforderlich:

Generierung eines passenden ZDOS mittels der Prozedur GEN.

Erstellung eines zu diesem ZDOS passenden PSP. Der PSP kann als höchste Adresse bei einem \$E000-ZDOS maximal für die Adresse D300 erstellt werden. Eingaben hierzu:

GENCOM PSP \$D300

bzw. eine andere Adresse bei anderem ZDOS-Anfang.

Installierung von FORMAT und SYSGEN und anschließendes SAVE der generierten Versionen auf der Diskette.

Nun sollte das so erhaltene Betriebssystem ausgiebig getestet werden. Erst nach diesen Tests sollten Änderungen an HEAS erfolgen (falls erforderlich).

9. Beim Betrieb von ZDOS auftretende Fehler sollten so genau wie möglich spezifiziert werden und zusammen mit einem einfachen Beispiel auf dem mitgelieferten Formular an den Lieferanten eingeschickt werden. Bei wirklichen Fehlern wird Ihnen gegen den Preis einer Diskette nach Beseitigung des Fehlers eine neue Version zugeschickt. Dieses bezieht sich selbstverständlich nur auf Fehler, die durch ein standardmäßiges ZDOS-Betriebssystem verursacht werden. **Bitte bei Reklamationen immer die Originaldiskette mit Aufkleber einsenden.**

HEAS ist über eine definierte Sprungleiste aufrufbar. Diese Sprungleiste beginnt immer mit einer Seitengrenze. Es gibt zwei Möglichkeiten, ihre Lage im Speicher zu lokalisieren:

1. Die Speicherzellen 0001 und 0002 enthalten den Adressteil des Befehls: JP WBOOT. Sie zeigen also auf Sprungleiste + 3. Damit ergibt sich folgende Aufrufmöglichkeit:

```
LD    HL, (1)
LD    L, functionsnummer * 3
LD    DE, RETADR
PUSH DE
JP    (HL)
```

RETADR:

2. Die Sprungleiste enthält 17 mal den Befehl JP adresse. Damit kann nach 17 mal C3 in Abständen von 3 Byte gesucht werden. Diese Methode ist sicherer, aber auch wesentlich aufwendiger. Die erste Lösung versagt nur bei speziellen Debuggern oder ähnlichen Programmen, die einen Warmstart resident überdauern.

HEAS-Sprungleiste

00	JP	BOOT	Einsprung nach Kaltstart
03	JP	WBOOT	Warmstart
06	JP	CST	Konsolzeichen eingegeben ?
09	JP	CIN	Konsoleingabe
0C	JP	COUT	Konsolenausgabe
0F	JP	LIST	Druckerausgabe
12	JP	PNCH	Lochstreifenausgabe
15	JP	RDR	Lochstreifeneingabe
18	JP	HOME	Kopf auf Spur 0
1B	JP	SELDSK	Laufwerk selektieren
1E	JP	SETTRK	Spur einstellen
21	JP	SETSEC	Sektor auswählen
24	JP	SETDMA	DMA-Puffer festlegen
27	JP	READ	Sektor lesen
2A	JP	WRITE	Sektor schreiben
2D	JP	LST	Drucker fertig ?
30	JP	SECTRA	Sektorverschränkung berechnen
33	JP	CBUF	Pufferadresse übergeben
36	JP	CONFIG	Konfiguration übergeben
39	00 oder FF		schnelle oder langsame Laufwerke

HEAS-Aufrufe verändern im Wesentlichen alle Register. HEAS verfügt nicht wie LEAS über seinen eigenen Stack.

Systemkaltstart

Beim Einschalten des Systems und bei jedem RESET führt das System einen Kaltstart aus. Hierbei kopiert der Bootstraploader das Betriebssystem (KI, LEAS und HEAS) von den Systemspuren der Diskette in Laufwerk A in den Arbeitsspeicher. Gleichzeitig werden alle implementierten Peripherieschnittstellen initialisiert. Der KI meldet sich auf der Konsole und erwartet die Eingabe eines Kommandos. Benutzt wird immer nach einem Kaltstart der Bereich des Benutzers 0. Ein anderer muß durch USER selektiert werden. Nur nach einem Kaltstart läuft die INIT-Prozedur ab. Sie ist durch Eingabe von ^S, ^C abbrechbar, falls dies erforderlich ist (z.B. der Speicherinhalt soll nicht verändert werden, da noch "gesaved" werden soll).

Systemwarmstart

Ein Systemwarmstart wird entweder durch Eingabe von ^C oder als Ende eines Anwenderprogramms bei dem LEAS-Aufruf SYSTEM RESET oder bei Sprung auf Adresse 0000H ausgeführt. Hierbei werden nur KI und LEAS neu von der Diskette geladen. KI meldet sich wie bei einem Kaltstart, ändert aber weder das Betriebslaufwerk noch den Benutzer. Die CONSOLE-CHECK-Option wird nach jedem Warmstart wieder aktiviert. Ein Warmstart setzt alle Disketten zurück, d.h. anschließend sind alle Disketten wieder im R/W-Zustand, d.h. beschreibbar (unter ZDOS).

***** Ab Version 5.b1 löscht HEAS den jetzt ausge-
***** lagerten LEAS-Datenbereich bei jedem Warmstart.

LEAS-Aufrufe sind Befehle von Anwendungsprogrammen an das Betriebssystem. Damit kann dem Anwendungsprogrammierer die Arbeit stark vereinfacht werden. Er kann alle Funktionen des Disketten-Betriebssystems benutzen. LEAS-CALLS benutzen folgende formalisierte Übergabe:

Jeder Aufruf geht auf Adresse 0005. Dort wird von LEAS ein Sprung in LEAS eingetragen. Damit ist eine Unabhängigkeit von sowohl der LEAS-Version wie auch der Lage im Speicher erreicht. Die gewünschte Funktion wird LEAS vom Anwendungsprogramm in Register C mitgeteilt. Es gibt die Funktionen 0 bis 40 und 250 bis 255. LEAS stellt Unterprogramme für folgende Tätigkeiten zur Verfügung:

1. Zeichen-Ein- und Ausgabe an Peripheriegeräte

- Konsoleingabe
- Konsolausgabe
- Lochstreifen einlesen
- Lochstreifen stanzen
- Ausgabe an den Drucker
- direkte Konsol-E/A
- Setzen von E/A-Zuordnungen
- zeilenweise Ein- und Ausgabe
- Abfrage des Konsolstatus

2. Dateimanipulationen (Disketten-E/A)

- Rücksetzen des Disksystems
- Laufwerksauswahl
- Datei erstellen
- Datei eröffnen
- Datei schließen
- Datei umbenennen
- Inhaltsverzeichnis durchsuchen
- sequentielles oder wahlfreies Lesen und Schreiben

3. Druckaufrufe

- Datei auf den Drucker ausgeben
- Druckausgabe abbrechen

4. Systemwarmstart

Benötigte Werte werden an LEAS in Register E oder DE übergeben. Antworten von LEAS erscheinen in A oder HL. Auf der nächsten Seite ist eine Funktionsübersicht aufgeführt. Im einzelnen sind die Funktionen im LEAS-Benutzer-Handbuch erklärt.

Nummer	Funktion	Eingabe	Ausgabe
00H	0 SYSTEM RESET	-	-
01H	1 CONSOLE INPUT	-	A = Zeichen
02H	2 CONSOLE OUTPUT	E = Zeichen	-
03H	3 READER INPUT	-	A = Zeichen
04H	4 PUNCH OUTPUT	E = Zeichen	-
05H	5 LIST OUTPUT	E = Zeichen	-
06H	6 DIRECT CONSOLE INPUT/OUTPUT	in: E = OFFH oder Zeichen	in: A = Zeichen oder 0
07H	7 GET I/O-BYTE	-	A = IOBYTE
08H	8 SET I/O-BYTE	E = IOBYTE	-
09H	9 PRINT STRING	DE = .Text	-
0AH	10 READ CONSOLE BUFFER	DE = .Puffer	-
0BH	11 GET CONSOLE STATUS	-	A = 0 oder OFFH
0CH	12 RETURN VERSION NO	-	HL = Version
0DH	13 RESET DISK SYSTEM	-	-
0EH	14 SELECT DISK	E = Diskno.	-
0FH	15 OPEN FILE	DE = .FCB	A = Suchcode
10H	16 CLOSE FILE	DE = .FCB	A = Suchcode
11H	17 SEARCH FIRST	DE = .FCB	A = Suchcode
12H	18 SEARCH NEXT	-	A = Suchcode
13H	19 DELETE FILE	DE = .FCB	A = Suchcode
14H	20 READ SEQUENTIAL	DE = .FCB	A = Fehlercode
15H	21 WRITE SEQUENTIAL	DE = .FCB	A = Fehlercode
16H	22 MAKE FILE	DE = .FCB	A = Suchcode
17H	23 RENAME FILE	DE = .FCB	A = Suchcode
18H	24 RETURN LOGIN VECTOR	-	HL = Loginvect
19H	25 RETURN CURRENT DISK	-	A = Disknummer
1AH	26 SET DMA ADDRESS	DE = .DMA	-
1BH	27 GET ADDR(ALLOC)	-	HL = .Alloc
1CH	28 WRITE PROTECT DISK	-	-
1DH	29 GET R/O-VECTOR	-	HL = R/O-Vector
1EH	30 SET ATTRIBUTES	DE = .FCB	A = Suchcode
1FH	31 GET ADDR(DSK PARMS)	-	HL = .DPB
20H	32 SET/GET USER CODE	get: E = FFH set: E = user	get: A = User set: -
21H	33 READ RANDOM	DE = .FCB	A = Fehlercode
22H	34 WRITE RANDOM	DE = .FCB	A = Fehlercode
23H	35 COMPUTE FILE SIZE	DE = .FCB	-
24H	36 SET RANDOM RECORD	DE = .FCB	-
25H	37 LOGOUT DISKS	DE = Vector	-
28H	40 WRITE RANDOM II	DE = .FCB	A = Fehlercode
FAH	250 DISABLE CONSOLE CHECK	-	-
FBH	251 OPEN FILE II	DE = .FCB	A = Suchcode
FCH	252 RETURN NDI3KS	-	A = Zahl der Disks
FDH	253 GET SPOOL STATUS	-	A = Status
FEH	254 STOP SPOOLING	-	-
FFH	255 SPOOL FILE	DE = .FCB	A = Suchcode

hierbei bedeutet: .xxxx Adresse von xxxx

Achtung: Aus Kompatibilitätsgründen ist BA gleich HL, falls in HL Werte übergeben werden.

PSP-Kommandos

Ybyte1 byte2 ...	Byte-String im Speicher lokalisieren
Danfang ende	Speicherausdruck (DUMP) in hex und ASCII
Fanfang ende wert	Füllen des Speichers mit konstanten Werten
Gstart stop	Ausführung (GO) eines Maschinenprogramms
Lanfang ende	Disassemblierung (LIST) des Speicherinhaltes
Mvon bis nach	Verschieben (MOVE) von Speicherinhalten
Xregister	Register-Anzeige und Veränderung
Sadresse	Ersetzen (SUBSTITUTE) von Speicherinhalten
Tvon bis mode	Schrittweise Überwachung (TRACE) von Programmen
Vanfang1 anfang2	Vergleichen (VERIFY) zweier Speicherbereiche
Itext	Eingabe von Dateinamen
Radresse	Datei einlesen
Wvon bis	Datei abspeichern

ASCII-Tabelle (deutsch)

	0	1	2	3	4	5	6	7	msd
0	NUL	DLE	SP	0	§	P	`	p	
1	SOH	DC1	!	1	A	Q	a	q	
2	STX	DC2	"	2	B	R	b	r	
3	ETX	DC3	#	3	C	S	c	s	
4	EOT	DC4	\$	4	D	T	d	t	
5	ENQ	NAK	%	5	E	U	e	u	
6	ACK	SYN	&	6	F	V	f	v	
7	BEL	ETB	'	7	G	W	g	w	
8	BS	CAN	(8	H	X	h	x	
9	HT	EM)	9	I	Y	i	y	
A	LF	SUB	*	:	J	Z	j	z	
B	VT	ESC	+	;	K	Ä	k	ä	
C	FF	FS	,	<	L	ö	l	ö	
D	CR	GS	-	=	M	Ü	m	ü	
E	SO	RS	.	>	N	^	n	ß	
F	SI	US	/	?	O	_	o	DEL	

lsd