

\*\*\*\*\*  
\*  
\*                   E O S                   \*  
\*  
\*    Betriebssystem für Microcomputer    \*  
\*                   mit Z80-CPU           \*  
\*  
\*                   Version 3             \*  
\*  
\*                   Anwenderhandbuch     \*  
\*  
\*\*\*\*\*

Copyright (C) 1983 Däumling & Zimmermann

Die Vervielfältigung auch von Teilen dieses Handbuchs bedarf unserer schriftlichen Genehmigung.

EOS ist ein eingetragenes Warenzeichen der Firma Däumling & Zimmermann, Seevetal, Deutschland.

CP/M ist ein eingetragenes Warenzeichen von Digital Research, Pacific Grove, California.

Abschnitt A: Allgemeine Einführung .....	1
A.1.    Wer oder was ist eigentlich EOS? .....	3
A.2.    EOS und seine Geschwister .....	4
A.3.    Äußerlichkeiten .....	5
A.3.1.    Der Bildschirm .....	5
A.3.2.    Die Tastatur .....	5
A.3.3.    Der Drucker .....	6
A.3.4.    Das Zusatzgerät .....	6
A.3.5.    Massenspeicher .....	6
A.3.6.    Das Behandeln von Disketten .....	8
A.3.7.    Die Zentraleinheit .....	9
A.4.    Innere Werte .....	10
A.4.1.    Das Bit .....	10
A.4.2.    Das Byte .....	10
A.4.3.    Das Feld .....	11
A.4.4.    Der Satz .....	11
A.4.5.    Dateien .....	11
A.5.    Die Bedienung des Rechners .....	13
A.5.1.    Der Kaltstart .....	13
A.5.1.1.    Das Autostart - Kommando .....	15
A.5.2.    Der Warmstart .....	16
A.5.3.    Das Wechseln von Disketten .....	17
A.5.4.    Tastatureingaben .....	18
A.5.5.    Stopzeichen .....	19
A.5.6.    Anhalten der Bildschirmausgabe .....	19
A.6.    Dateinamen .....	20
A.6.1.    Eindeutige Dateinamen .....	20
A.6.1.1.    Der Dateityp .....	21
A.6.2.    Mehrdeutige Dateinamen .....	23
A.6.2.1.    Das Fragezeichen .....	23
A.6.2.2.    Das Sternzeichen .....	24
A.6.3.    Das Laufwerk .....	24
A.6.4.    Dateiattribute .....	25
A.7.    Benutzerbereiche .....	26
A.8.    Die Statuszeile des MZ-3500 .....	27
A.9.    Die Tastatur des MZ-3500 .....	28
A.9.1.    Die Definition von Sondertasten .....	29
A.9.2.    Vordefinition der Funktionstasten .....	30

Abschnitt B: Eingabe von Kommandos .....	32
B.1.  Der Kommando - Interpreter .....	33
B.1.1.  Der Bildschirm - Editor .....	33
B.1.2.  Die Interpretation der Eingabe .....	34
B.1.3.  Die Eingabe von Kleinbuchstaben .....	35
B.1.4.  Sonderzeichen am Zeilenanfang .....	36
B.1.4.1.  Das Semikolon .....	36
B.1.4.2.  Der Doppelpunkt .....	36
B.1.4.3.  Das Sternchen .....	36
B.1.5.  Die Suche nach den Dateien .....	37
B.1.6.  Eingebaute Kommandos .....	38
B.1.6.1.  BEN - Wechsel des Benutzerbereichs .....	39
B.1.6.2.  DRUCKER - Drucker ein- und ausschalten .....	39
B.2.  Das Menüprogramm .....	40
B.2.1.  Mögliche Eingaben beim Menüprogramm .....	41
B.2.2.  Die Programmierung des Menüprogramms .....	42
B.2.2.1.  Beispiel einer Menüdatei .....	43

Abschnitt C: Dienstprogramme .....	45
C.1.    Menüprogramme .....	45
C.2.    Programme mit Kommandozeile .....	46
C.2.1.  Optionen .....	46
C.3.    \$CONFIG - Konfiguration des Betriebssystems ....	47
C.4.    \$COPY - Der Dateikopierer .....	48
C.4.1.  Aufruf und Kommandoeingabe .....	48
C.4.2.  Direktaufruf .....	51
C.4.3.  Der Kopiervorgang und das Fehlerverhalten .....	51
C.4.4.  Verhalten bei Platzmangel .....	53
C.4.5.  Optionen .....	53
C.4.5.1. Kopieren mit Rückfrage .....	54
C.4.5.2. Sparsames Kopieren .....	54
C.4.5.3. Überspringen von temporären Dateien .....	55
C.4.5.4. Quellen nach dem Kopiervorgang löschen .....	55
C.4.5.5. Kopieren mit Prüflesen .....	55
C.4.5.6. Löschen des höchstwertigen Bits .....	56
C.4.5.7. Protokoll ausschalten .....	56
C.4.5.8. Datensicherung .....	56
C.4.5.9. Kopieren zwischen Benutzerbereichen .....	57
C.4.6.  Kopieren über Geräte .....	57
C.4.6.1. Datenkommunikation .....	58
C.4.6.2. Das Senden des Dateiendekennzeichens .....	59
C.4.6.3. Direktschaltung des Bildschirms .....	59
C.4.7.  Das Unterbrechen der Kopie .....	59
C.4.8.  Fehlermeldungen .....	60
C.5.    \$COPYDSK - Kopie von Disketten .....	62
C.6.    \$DATE - Setzen des Tagesdatums .....	63
C.7.    \$DEVICE - Definieren der externen Geräte .....	64
C.8.    \$DIR - Inhaltsverzeichnis eines Laufwerks .....	66
C.8.1.  \$DIR - Optionen .....	67
C.8.1.1. SYSTEM - Anzeige versteckter Dateien .....	68
C.8.1.2. ALLE - Alle Dateien anzeigen .....	68
C.8.1.3. VORSCHUB - Seitenweise Anzeige .....	69
C.8.1.4. ZEIT - Anzeige der Zeitmarken .....	69
C.8.1.5. BEN - Anzeige über alle Benutzerbereiche .....	70
C.8.1.6. DRUCKER - Ausgabe auf dem Drucker .....	70
C.9.    \$DISKINF - Plattenformate anzeigen .....	71
C.10.   \$DO - Programmierbares Abarbeiten von Kommandos ..	72
C.10.1.  Aufruf .....	72
C.10.2.  Eingabe von Steuerzeichen .....	74
C.10.3.  Weitere Sonderzeichen .....	74
C.10.4.  Die Anzeige der Abarbeitung .....	75
C.10.5.  Beispiele .....	76
C.10.6.  Fehlermeldungen .....	77

C.11.	\$ERASE - Löschen von Dateien .....	78
C.11.1.	SYSTEM - Nur versteckte Dateien löschen .....	79
C.11.2.	ALLE - Alle Dateien eines Benutzerbereichs löschen .....	79
C.11.3.	BEN - Dateien in allen Benutzerbereichen löschen .....	79
C.11.4.	R/O - Auch schreibgeschützte Dateien löschen ...	79
C.11.5.	FRAGEN - Jede Datei bestätigen lassen .....	80
C.11.6.	DIREKT - sofortiges Löschen .....	80
C.11.7.	Englische Optionen .....	80
C.12.	\$INIT - Initialisierung von Disketten .....	81
C.12.1.	Minidiskettenformate .....	82
C.12.2.	Standarddisketten .....	82
C.12.3.	Die Hard Disk .....	83
C.13.	\$INITDIR - Steuern von Zeiteintragen .....	84
C.14.	\$KEYDEFS - Speichern und Laden der Tastaturdefinitionen .....	86
C.15.	\$MOD - Arbeiten mit Funktionsmodulen .....	87
C.15.1.	Das Anfügen von Funktionsmodulen .....	88
C.15.2.	Das Entfernen von Funktionsmodulen .....	88
C.15.3.	LADER - Verbleiben des Programmladers .....	89
C.15.4.	COM - Funktionsmodule zusammenfassen .....	89
C.15.5.	PROT - Protokollierung unterdrücken .....	89
C.15.6.	MODULE - Liste aller aktiven Module .....	90
C.15.7.	Englische Optionen .....	90
C.16.	\$RENAME - Umbenennen von Dateien .....	91
C.16.1.	SYSTEM - Nur versteckte Dateien umbenennen .....	92
C.16.2.	ALLE - Alle Dateien eines Benutzerbereichs umbenennen .....	92
C.16.3.	BEN - Dateien in allen Benutzerbereichen umbenennen .....	93
C.16.4.	R/O - Auch schreibgeschützte Dateien umbenennen	93
C.16.5.	FRAGEN - Jede Datei bestätigen lassen .....	93
C.16.6.	DIREKT - sofortiges Umbenennen .....	93
C.17.	\$SAVE - Abspeichern des Arbeitsspeichers .....	94
C.18.	\$SET - Setzen und Löschen von Dateiattributen ..	96
C.19.	\$SPOOL - Dateien im Hintergrund drucken .....	97
C.19.1.	Optionen .....	98
C.19.1.1.	STOP - Druck anhalten und freigeben .....	98
C.19.1.2.	RESTART - Druck neu starten .....	98
C.19.1.3.	ABBRUCH - Druck abbrechen .....	99
C.19.1.4.	VORSCHUB - Seitenvorschub unterdrücken .....	99
C.19.2.	Zusätzliches für Eingeweihte .....	99
C.20.	\$TYPE - Ausdrucken von Dateiinhalten .....	100
C.20.1.	VORSCHUB - Seitenweise Ausgabe .....	100
C.20.2.	SCHRITT - langsame Bildschirmausgabe .....	100
C.20.3.	CONTROL - Ausgabe von Steuerzeichen .....	100
C.20.4.	DRUCKER - Ausgabe auf dem Drucker .....	101
C.20.5.	Englische Optionen .....	101

Abschnitt D: der Grafik - Interpreter .....	103
D.1.    Die Aktivierung des Grafik - Interpreters .....	103
D.2.    Das Befehlsformat .....	104
D.3.    Die Definition von Bit Images .....	105
D.4.    Ein Programmierbeispiel .....	106
D.5.    Liste der Grafikkommandos .....	107
D.6.    Die einzelnen Befehle .....	108
D.6.1.  INIT - Initialisieren der Grafik .....	108
D.6.2.  CRT - Definition des Bildschirms .....	109
D.6.3.  BKGR - Hintergrundfarbe definieren .....	110
D.6.4.  COLR - Wahl der Zeichenfarbe .....	110
D.6.5.  MASK - Definition der Zeichenmaske .....	110
D.6.6.  MODE - Zeichenmodus setzen .....	111
D.6.7.  IMG - Bitmuster definieren .....	111
D.6.8.  CLR - Grafik löschen .....	112
D.6.9.  SET - Bildpunkt schreiben .....	112
D.6.10.  RES - Bildpunkt löschen .....	112
D.6.11.  LINE - Linie zeichnen .....	113
D.6.12.  TO - Linie fortschreiben .....	113
D.6.13.  RECT - Rechteck zeichnen .....	113
D.6.14.  FILL - Rechteckfläche füllen .....	113
D.6.15.  CIRC - Kreis zeichnen .....	114
D.6.16.  ARC - Kreisabschnitt zeichnen .....	114
D.6.17.  TEXT - Texte ausgeben .....	115
D.6.18.  COPY - Hard Copy auf dem Tintenstrahldrucker ...	116
D.6.19.  END - Grafik beenden .....	116

Herzlich willkommen in der Familie der Computerbenutzer!

---

Sie sind also dazu auserwählt worden, sich in die Bedienung und Benutzung Ihres neuen Computers einzuarbeiten. Bevor Sie die folgenden Seiten durcharbeiten, sollten Sie sich mit der Betriebsanleitung Ihres Computersystems vertraut machen, die Ihnen Aufschluß darüber gibt, welche Stecker in welche Steckdose gehören, wo die verschiedenen Geräte eingeschaltet werden und was Sie beachten müssen, um Schaden an Ihren neuen Geräten zu vermeiden.

So, nun haben Sie einige -zig Seiten unseres Benutzerhandbuchs vor sich, die Ihnen in Fleisch und Blut übergehen müssen, damit Sie in Zukunft zugig mit Ihrem neuen Werkzeug umgehen können.

Keine Angst! - Sie werden es ganz bestimmt schaffen. Die beste Methode, sich an die neue Umgebung zu gewöhnen, ist die praktische Arbeit. Scheuen Sie sich nicht, dieses Handbuch neben Ihren Bildschirm zu legen und nach und nach alles auszuprobieren. Vorher sollten Sie jedoch das ganze Buch einmal durchlesen, damit Sie einen Gesamtüberblick erhalten.

Abschnitt A  
Allgemeine Einführung

\*\*\*\*\*  
\* was ist EOS eigentlich? \*  
\*\*\*\*\*

### A.1. Wer oder was ist eigentlich EOS?

Hinter der Bezeichnung EOS verbirgt sich das Betriebssystem Ihres Rechners. Es besteht aus einem Paket von Programmen, die es Ihnen ermöglichen, mit dem Computer sinnvolle Dinge zu tun. Ohne diese Programme wäre ein Computer nicht mehr als eine nutzlose Anhäufung von allerlei elektronischen und mechanischen Geräten, die bestenfalls das Herz eines Bastlers hoher schlagen lassen.

Diese Betriebssystem - Programme erledigen Dinge von grundsätzlicher Bedeutung. Einige von ihnen sind immer im Einsatz, ohne daß Sie sich dessen bewußt zu sein brauchen. Wenn Sie eine Taste auf Ihrer Bildschirm-Tastatur drücken, wenn ein Zeichen auf dem Bildschirm ausgegeben wird, wenn Sie den Rechner einschalten und auch wenn Sie meinen, daß gar nichts geschieht, immer ist das Betriebssystem im Einsatz.

Einige dieser Programme haben Namen. Sie werden die Namen und Funktionen dieser Programme noch kennenlernen. Andere wiederum verrichten tief im Inneren des Rechners ihre Arbeit und nur der Programmierer weiß, wie sie heißen.

Kurz und gut, alle jene Programme, die nicht unmittelbar mit dem Problem zu tun haben, das Sie bearbeiten, ohne die Sie aber nicht auskommen können, sind dem Betriebssystem zuzuordnen.

EOS ist nun in vielerlei Hinsicht ein besonderes Betriebssystem. Es bemüht sich, alle Anweisungen und Meldungen in einem möglichst verständlichen Deutsch auszugeben. Es verfügt über besondere Funktionen, die dem Techniker die Fehlersuche erleichtern. Es ist schneller, gründlicher und sicherer als andere vergleichbare Betriebssysteme.

EOS verfügt über Dienstprogramme, die seinen Einsatz komfortabler, sicherer und praxisnaher gestalten.

EOS kann Ihren Drucker bedienen und gleichzeitig andere Programme verwalten. Vorbei die Zeit des Wartens, bis endlich eine Liste gedruckt war! Sie arbeiten mit Ihrem Textprogramm, Ihrer Lager- oder Finanzbuchhaltung, während auf dem Drucker die Auswertungen ausgegeben werden.

\*\*\*\*\*  
\*vergleichbare Betriebssysteme\*  
\*\*\*\*\*

## A.2. EOS und seine Geschwister

Es ist kein Geheimnis: Bei der Konstruktion von EOS hat ein anderes Betriebssystem Pate gestanden - das System CP/M der kalifornischen Firma Digital Research.

Diese Patenschaft hat ihren guten Grund. CP/M hat eine Normung für Programme geschaffen, die sich weitweit bewahrt und behauptet hat. An diese Norm halten sich einige tausend Programme, die alle auf allen Computern funktionieren, auf denen ein zu CP/M kompatibles Betriebssystem installiert ist.

Die Norm funktioniert hier ähnlich wie die genormte Verbindung eines Plattenspielers mit Ihrer HiFi-Anlage. Sie müssen sich nur vorstellen, daß die Steckverbindung ungleich mehr Pole besitzt.

Als Benutzer einer Computeranlage, auf der EOS installiert ist, befinden Sie sich in der Gesellschaft von - grob geschätzt - 400.000 Computerbenutzern in aller Welt. Täglich werden es mehr und täglich wächst die Zahl von guten Anwendungsprogrammen, die alle auf diesen Rechnern eingesetzt werden können.

```
*****  
*      Außerlichkeiten      *  
*****
```

### A.3. Außerlichkeiten

Es ist schwierig, über das äußere Erscheinungsbild und die Funktion einzelner Geräte etwas zu sagen, wenn man nicht voraussagen kann, was den Konstrukteuren neuer Computersysteme bei der Gestaltung noch alles einfallen kann. Wir wollen hier jedoch nach besten Kräften versuchen, die Funktion jener Geräte zu beschreiben, die in aller Regel Bestandteil eines Computersystems sind, auf dem EOS installiert ist. Ja, einige dieser Geräte sind zwingender Bestandteil und unbedingt erforderlich, damit EOS funktionieren kann.

#### A.3.1. Der Bildschirm

Seine Anwesenheit ist zwingend erforderlich. Ohne ihn könnten Sie nicht mit dem Computer kommunizieren. Er macht Zeichen sicht- und lesbar, die der Computer ( oder besser das Programm, welches in dem Computer gerade läuft ) an ihn schickt. Da jedes Zeichen einzeln übertragen wird, kann man den Bildschirm als zeichenorientiertes Gerät klassifizieren. Wir werden diesem Begriff noch bei einigen anderen Geräten begegnen.

Der Bildschirm hat aus der Sicht des Betriebssystems den symbolischen Namen CONOUT:, der aus der englischen Bezeichnung CONsole OUTput abgeleitet ist. Da die Console als klassische Ein- und Ausgabeeinheit auch Eingaben des Bedieners an den Computer übermittelt, kann der Bildschirm nicht ohne seinen ständigen Begleiter auskommen:

#### A.3.2. Die Tastatur

Auch sie ist ein für die Funktion Ihres Rechners und seines EOS - Betriebssystems unbedingt erforderlicher Bestandteil. Wegen ihrer starken Bindung zum Bildschirm ist sie häufig mit ihm durch Kabel verbunden oder sogar im gleichen Gehäuse untergebracht.

Über die Tastatur können Sie dem Computer Ihren Willen aufzwingen. Hier geben Sie Ihre Kommandos ein. Dies geschieht ebenfalls Zeichen für Zeichen, mit jedem Tastendruck, den Ihre Finger ausführen. Auch die Tastatur ist daher ein zeichenorientiertes Gerät.

Die Tastatur wird aus der Sicht des Betriebssystems mit dem Namen CONIN: (vom englischen CONsole INput) angesprochen und vertritt den Eingabekanal der Console.

```
*****
*      Außerlichkeiten      *
*****
```

### A.3.3. Der Drucker

Er ist ein Vertreter jener Geräte, die nur Nachrichten des Computers empfangen können. Seine Anwesenheit ist zwar nicht unbedingt erforderlich - es gibt zahlreiche Programme, die vollkommen ohne Drucker auskommen und dennoch sehr nützliche Dienste verrichten - aber ansonsten obligatorisch.

Der Drucker empfängt die auszugebenden Texte ebenfalls Zeichen für Zeichen. Er wird mit dem Namen LSTOUT:, welcher vom englischen LIST OUTPUT stammt, angesprochen.

### A.3.4. Das Zusatzgerät

Neben diesen Geräten kennt EOS noch ein frei definierbares Zusatzgerät. Dieses kann beispielsweise für eine Telefon - Verbindung zu einem anderen Rechner benutzt werden. Das Zusatzgerät ist in zwei Kanäle unterteilt:

```
AUXIN: - der Eingabekanal und
AUXOUT: - der Ausgabekanal
```

Sie werden diesen Namen bei der Benutzung von => \$DEVICE begegnen. Als Anwender kommen Sie nur selten oder nie damit in Berührung.

### A.3.5. Massenspeicher

Unter diesem Begriff fassen wir jene Geräte zusammen, die es Ihnen erlauben, einmal eingegebene Daten, Programme, Texte usw. aufzubewahren. Dieses Speichern ist eine Funktion, das Lesen der gespeicherten Daten die andere Funktion dieser Geräteklasse.

Die Computersysteme, auf denen EOS als Betriebssystem eingesetzt ist, verfügen über mindestens eine solche Speichereinheit.

Physisch sind diese Speichereinheiten als Disketten- oder Winchesterlaufwerke in Ihrem Rechner eingebaut oder durch Kabel mit ihm verbunden. Bis zu 16 solcher Geräte können von EOS verwaltet werden.

Allerdings wollen wir in diesem Zusammenhang nicht mehr von Geräten sprechen. Es kann nämlich durchaus passieren, daß eine große Festplatteneinheit der Übersichtlichkeit wegen in mehrere logische Speichereinheiten aufgeteilt ist, die sich optisch in einem Gerät befinden. Bei Diskettenlaufwerken sind meist zwei in einem Gerät oder im Gehäuse des Rechners untergebracht. Wir werden daher in Zukunft immer den Begriff Laufwerk verwenden.

```
*****
*      Außerlichkeiten      *
*****
```

Die Laufwerke haben keinen besonderen Namen, sondern werden durch einen Buchstaben, gefolgt von einem Doppelpunkt, angewählt. Das erste Laufwerk heißt demnach A:, das zweite mit B: bis hin zu P:.

Auf Ihrem MZ-3500 können je nach Ausbaustufe bis zu 11 Laufwerke definiert sein. Allgemein gilt jedoch der Grundsatz, daß die schnellste Speichereinheit zuerst kommt. Sind Sie Besitzer eines Winchester - Laufwerks, wird dieses als Laufwerk A: und B: angesprochen. Standard - Floppylaufwerke folgen den Winchesterlaufwerken. Zuletzt kommen die Mini - Floppylaufwerke. Die eingebaute RAM - Disk ist immer fest als Laufwerk P: installiert.

Als Beispiel sei hier die Konfiguration eines Rechners mit zwei Mini - Diskettenlaufwerken gezeigt:

```
A: rechtes Diskettenlaufwerk
B: linkes Diskettenlaufwerk
P: RAM - Disk
```

Ein System mit einem Standard - Doppelfloppylaufwerk und zwei Mini - Floppylaufwerken würde so aussehen:

```
A: Standardfloppy, rechtes Laufwerk
B: Standardfloppy, linkes Laufwerk
C: Minifloppy, rechtes Laufwerk
D: Minifloppy, linkes Laufwerk
P: RAM - Disk
```

Ein System mit einer Winchesterplatte, einem Standard - Doppelfloppylaufwerk und zwei Mini - Floppylaufwerken hatte die folgende Konfiguration:

```
A: Winchesterlaufwerk
B: Winchesterlaufwerk
C: Standardfloppy, rechtes Laufwerk
D: Standardfloppy, linkes Laufwerk
E: Minifloppy, rechtes Laufwerk
F: Minifloppy, linkes Laufwerk
P: RAM - Disk
```

Diskettenlaufwerke und Winchesterplatten haben eines gemeinsam: die Daten werden blockweise auf ihnen gespeichert. Mit jedem Zugriff auf diese Speichereinheiten wird ein Block von Daten geschrieben bzw. gelesen. Solche Einheiten werden daher auch als blockorientierte Geräte klassifiziert.

\*\*\*\*\*  
\*           Äußerlichkeiten           \*  
\*\*\*\*\*

### A.3.6. Das Behandeln von Disketten

Hier nun einige Dinge bezüglich Ihrer Disketten, die es zu beachten gilt: Sie sollten immer sehr vorsichtig mit der Bedienung des Netzschalters sein. Es kann durchaus passieren, daß der Computer durch das Ausschalten desselben in einen unkontrollierten Zustand gerät und irgendwelchen Murks auf die Disketten oder die Winchesterplatten schreibt. Dieser Murks ist für die dort gespeicherten Dateien und Programme mit Sicherheit tödlich. Davon einmal abgesehen kann es für die Elektronik und Mechanik von Winchesterlaufwerken sehr schädlich sein, wenn der Strom während eines Schreib- oder Lesevorgangs abgeschaltet wird. Wir wollen uns solche Schwierigkeiten lieber für die Fälle aufheben, in denen so etwas durch höhere Gewalt geschieht.

Immer die Disketten aus den Laufwerken nehmen, bevor abgeschaltet wird! Niemals abschalten, wenn eine Kontrolleuchte an Disketten- oder Winchesterlaufwerken brennt!

Disketten sind sehr empfindlich. Versuchen Sie bitte nicht, herauszufinden, wie empfindlich sie sind, indem Sie nachlässig mit ihnen umgehen.

Niemals mit den Fingern auf die Magnetfolien fassen!

Die Disketten-Aufkleber niemals mit einem harten Stift beschriften. Am besten die Aufkleber vor dem Aufkleben beschriften!

Disketten immer in der Schutzhülle aufbewahren!

Vorsicht vor magnetischen Einflüssen durch Transformatoren, Motoren, Lautsprecher, Fernseher usw.!

Nicht in die Sonne legen, sondern kühl aufbewahren!

Beschädigte Disketten nicht weiterverwenden!

Es gibt gute und schlechte Disketten. Billige Disketten müssen nicht schlecht sein. Sehr gute Disketten können nicht billig sein. Sie sollten sich für sehr gute Disketten entscheiden. Spätestens wenn Ihnen durch einen Diskettenfehler die Arbeit mehrerer Tage oder Wochen zerstört wurde, wissen Sie warum. Außerdem schonen die sehr guten die Schreib-Lese-Köpfe Ihrer Diskettenlaufwerke. Sie werden auf die Dauer billiger!

```
*****  
*      Außerlichkeiten      *  
*****
```

### A.3.7. Die Zentraleinheit

Unter diesem Namen fassen wir jenen geheimnisumwitterten Teil Ihres Computers zusammen, in dem alle Fäden zusammenlaufen. Dies ist durchaus bildlich gesprochen. Dorthin gehen die Kabel vom Bildschirm, der Tastatur, dem Drucker und den Massenspeichern. Es gibt natürlich auch Rechner, bei denen die Zentraleinheit im gleichen Gehäuse wie die Tastatur untergebracht ist oder wo die Größe der Diskettenlaufwerke kaum vermuten läßt, daß dort auch das elektronische Herz schlägt. Bei der sturmischen Entwicklung der Mikroelektronik wollen wir uns an dieser Stelle über die Unterbringung dieses wichtigen Teiles lieber nicht weiter auslassen.

Die Zentraleinheit besteht aus folgenden Funktionseinheiten:

- die CPU

Hier handelt es sich immer um einen Baustein des Typs Z80, wenn EOS funktionieren soll. Sie setzt die in den Programmen vorhandenen Anweisungen in die gewünschten Aktionen um und verwaltet den Hauptspeicher. Mehr sollten Sie sich an dieser Stelle nicht merken.

- der Hauptspeicher

Merken Sie sich bitte nur, daß hier die Programme ablaufen. Auch EOS hat dort seine Residenz. Seine Größe ist seine Leistung. Die Z80 - CPU kann bis zu 64 KBytes adressieren. Um auch größere Hauptspeicher ansprechen zu können, enthält EOS verschiedene Speicherwaltungs - Mechanismen. Somit können die 128 KBytes Ihres MZ-3500 voll angesprochen werden. Im Hauptspeicher werden Daten und Programme gespeichert. Jedes gespeicherte Zeichen kann von der CPU direkt angesprochen, gelesen, verändert und wieder gespeichert werden. Sein Nachteil ist seine Abhängigkeit von der Energie - er verliert sofort sein Gedächtnis, wenn der Strom abgeschaltet wird.

- die Steuereinheiten

Diese, zum Teil sehr intelligenten, Bausteine übernehmen die Aufgabe, alle angeschlossenen Geräte zu steuern, oder besser, die Datenübertragung mit ihnen abzuwickeln.

```

*****
*           Innere Werte           *
*****

```

#### A.4. Innere Werte

Hatten wir es eben noch mit faßbaren Dingen zu tun, wenden wir uns nunmehr Begriffen zu, die zwar untrennbar mit der Datenverarbeitung verbunden, aber eben leider nur schwer zu erklären sind, weil man sie nicht sehen kann. Dem interessierten Leser sei an dieser Stelle entsprechend ausführliche Spezialliteratur empfohlen, da wir hier wirklich nur oberflächliche Aufklärung bieten können.

##### A.4.1. Das Bit

Computer sind dumm, aber schnell. Einer, der es wissen muß, hat den Begriff vom schnellen Idioten geprägt. Ein Computer kennt nur zwei logische Zustände. Er kann, wenn man so will, noch nicht einmal bis zwei zählen. Seine Grenze liegt bei Eins. Damit hat er immer zwei Möglichkeiten:

0 oder 1  
ein oder aus

das ist hier die Frage. Diese beiden Zustände lassen sich in einem Bit darstellen. Es ist die kleinste logische Größe, mit der der Computer agiert. Da dem Menschen das Bit zu klein war, faßte er mehr oder weniger willkürlich einige Bits zusammen und nannte diese neue Größe

##### A.4.2. Das Byte

Genau acht Bits machen ein Byte aus. Ein Byte ist die kleinste Größe, auf die ein Computer speichertechnisch Bezug nehmen kann. Es kennt, wenn man die Null mitrechnet, genau 256 Zustände - mit 8 Bit kann man bis 255 zählen.

Auf den ersten Blick ist das nicht viel. Aber überlegen Sie mal, was man in 256 alles unterbringen kann. 256 verschiedene Zeichen zum Beispiel brauchen wir gar nicht, um unser Alphabet darzustellen, denn unser Alphabet hat weit weniger, selbst wenn wir die Ziffern und die Sonderzeichen hinzunehmen.

Wir merken uns einfach, daß ein Zeichen immer ein Byte belegt, dann liegen wir immer richtig.

Um mit größeren Zeichenmengen elegant umgehen zu können, wurden neue Begriffe geschaffen:

KB = Kilobyte	=	1024 Bytes
MB = Megabyte	=	1024 KB = 1.048.576 Bytes

```

*****
*           Innere Werte           *
*****

```

#### A.4.3. Das Feld

Dieser Begriff geht mehr in die Richtung der Anwendungsprogrammierung und hat eigentlich nichts mit der Beschreibung eines Betriebssystems zu tun.

Gehören mehrere Zeichen logisch zusammen, so bilden sie eine Menge, der man einen Namen geben kann. Angenommen, Sie haben die Absicht, die Adressen Ihrer zu Kunden speichern. Die Adresse setzt sich aus folgenden Daten zusammen:

```

Name
Vorname
Strasse
Ort

```

Hierbei kann das Feld "Name" jeden beliebigen Inhalt annehmen. Ob dort Meier, Schulze oder Müller gespeichert ist, bestimmen Sie bei der Eingabe dieser Namen.

#### A.4.4. Der Satz

Auf Neudeutsch auch Record genannt. Er ist eine logisch zusammengehörende Menge von Feldern. Knüpfen wir an das obige Beispiel an, so bilden die Felder Name, Vorname, Strasse und Ort den Satz Adresse. Wann immer Ihr Programm einen solchen Satz speichert, ist der logische Inhalt durch die Reihenfolge der darin enthaltenen Felder festgelegt.

Zeichen bilden Felder, die Sätze bilden, und diese wiederum bilden:

#### A.4.5. Dateien

Einfach. nicht wahr? Alle Adressen einmal zusammengefaßt hatten früher unter Verwendung von Karteikarten in einem Karteikasten Platz gefunden und waren Kundenkartei genannt worden. Dank des Fortschrittes der Technik dürfen wir heute die Bezeichnung "Kundendatei" verwenden.

Weil wir so flott vorangekommen sind, wollen wir an dieser Stelle noch ein wenig verweilen.

Es sind Dateien, auf die ein Programm zurückgreift. Sie werden von ihm gelesen, geschrieben, verändert, umbenannt oder gelöscht. Es gibt Programme, die alle diese Funktionen gleichzeitig und mit mehreren Dateien vornehmen. Eine Finanzbuchhaltung gehört mit Sicherheit zu dieser Familie von Programmen.

```
*****  
*           Innere Werte           *  
*****
```

Das Betriebssystem stellt solchen Programmen die entsprechenden Möglichkeiten zur Verfügung. Wir haben vorher die internationale Normung von EOS besprochen. Zu dieser Norm gehört auch die Schreibweise der Dateinamen - doch davon später.

Merken Sie sich an dieser Stelle bitte noch eines: Auch Programme sind Dateien. Dateien, die im Gegensatz zu Ihren Adressdaten Anweisungen für die CPU enthalten. Eine besondere Funktion des Betriebssystems lädt diese Programmdateien an eine genau bestimmte Stelle des Hauptspeichers und veranlaßt die CPU, die dort gespeicherten Anweisungen zu befolgen.

```
*****
*           der Kaltstart           *
*****
```

### A.5. Die Bedienung des Rechners

Jetzt wird es ernst! Spätestens jetzt sollten Sie die Bedienungsanleitung Ihres neuen Computers lesen oder sich vom Verkäufer die Inbetriebnahme zeigen lassen. Gehen wir davon aus, daß alles ordnungsgemäß verdrahtet ist, dann suchen Sie jetzt bitte nach dem Netzschalter, weil wir den ersten Schritt tun:

#### A.5.1. Der Kaltstart

Öffnen Sie die Diskettenschachtel und schalten Sie den Rechner ein. Es erscheint die englische Meldung "SYSTEM LOADING" und nach einer kleinen Weile die Meldung "NO SYSTEM MEDIA". Jetzt legen Sie Ihre EOS - Systemdiskette in den Schacht des rechten Mini - Diskettenlaufwerks ein und schließen Sie die Tür.

Na, hat sich was getan? Nein - dann war das nicht das richtige Diskettenlaufwerk oder die Diskette war nicht richtigerum eingelegt - oder es war nicht die Systemdiskette. Versuchen Sie ein paar Variationen. Wenn das immer noch nicht zum Erfolg führt, sollten Sie vielleicht noch einmal in der Betriebsanleitung nachlesen. 80 % der an dieser Stelle auftretenden Schwierigkeiten sind durch mangelnde Vorabinformation begründet.

Wenn alles in Ordnung ist, erscheint auf dem Bildschirm die Meldung:

```
EOS Version 3
Copyright (C) 1983 Daeumling & Zimmermann
```

Modul	Bnk	Lade	Strt	Große
SUBCPU	1	4000	4000	2008
SYSVAR	0	FE00	FE00	0200
SYSTEM	1	2000	3788	1FFA
MZ3500	2	2000	2000	160B
SHELL	0	8000	0100	2062
BANKMGR	0	F100	F500	0500

>

```

*****
*           der Kaltstart           *
*****

```

Diese Zeilen sind das sogenannte Ladeprotokoll von EOS. Das Betriebssystem besteht aus einer Reihe von Dateien, die hintereinander weg in die verschiedenen Speicherbanke Ihres Rechners geladen werden. Das Protokoll selbst hat mehr optischen Wert. Eingeweihten sagt es die Bank, die Ladeadresse, die Startadresse und die Größe eines jeden Moduls. Wichtig ist hingegen folgendes: Die obigen sechs Module werden aus ebensoviele Dateien geladen. Diese Dateien haben die folgenden Namen:

```

SUBCPU.SYS      SYSVAR.SYS      SYSTEM.SYS
MZ3500.SYS     SHELL.SYS       BANKMGR.SYS

```

Diese Dateien müssen alle auf der Systemdiskette vorhanden sein, damit EOS korrekt geladen werden kann. Fehlt auch nur eine dieser Dateien, kann EOS nicht geladen werden. Damit Ihnen niemals eine Datei verloren geht, sollten Sie sich so bald wie möglich mit Hilfe des Programms => \$COPYDSK eine Kopie Ihrer Systemdiskette anfertigen.

Das Zeichen ">" ist im Gegensatz zu der obigen Meldung nicht typisch für einen Kaltstart, sondern vielmehr eine Bereitschaftsanzeige Ihres EOS - Betriebssystems. Sie besagt, daß EOS bereit ist, Ihre Anweisungen entgegenzunehmen. Im Grunde gibt nicht EOS selbst dieses Zeichen aus, sondern vielmehr der Kommando - Interpreter SHELL. Doch mehr darüber später.

Was ist nun geschehen, als der Kaltstart durchgeführt wurde?

Ihr Computer ist für diese Funktion vorprogrammiert. Im Gegensatz zu den Programmen, die sich auf einem der Laufwerke befinden können, ist dieses Programm jedoch im wahrsten Sinne des Wortes fest in den Speicher der Zentraleinheit eingebrannt und kann daher auch nicht verloren gehen, wenn der Strom abgeschaltet wird. Das Programm ist sehr kurz und hat im wesentlichen die Aufgabe, vom Laufwerk A: die einzelnen EOS - Bausteine einzulesen und in den Hauptspeicher zu schreiben. Dann wird die Steuerung des weiteren Ablaufs EOS übertragen. EOS setzt die angeschlossenen Geräte und Steuereinheiten in ihre Grundstellung, zeigt die hübsche Meldung auf dem Schirm an und lädt jenen Teil des Betriebssystems (namlich das Programm SHELL), der die Bereitschaftsanzeige ausgibt und auf Ihre Eingaben wartet.

```
*****  
*           der Kaltstart           *  
*****
```

#### A.5.1.1. Das Autostart - Kommando

Dieser Absatz behandelt eine Fähigkeit von EOS, deren Verständnis bereits voraussetzt, wie Kommandos und Dateinamen aussehen. Da es sich jedoch dabei um eine Sache handelt, die nur beim Kaltstart von EOS auftritt, muß sie hier bereits beschrieben werden. Lesen Sie diesen Abschnitt daher erst genau durch, wenn Sie bereits etwas mehr über EOS wissen.

Es ist möglich, eine Reihe von Kommandos automatisch beim Start von EOS ausführen zu lassen. Derartige Kommandos werden in eine Datei namens "PROFILE.SUB" auf dem Laufwerk A eingetragen. Ist diese vorhanden, wird beim Start von EOS automatisch das Kommando

```
$DO A:PROFILE
```

ausgeführt. Die Struktur einer ".SUB" - Datei ist unter der Beschreibung des Dienstprogramms \$DO nachzulesen. Hier nur ein kurzes Beispiel: Sie möchten, daß alle Menüdateien (was das ist, werden Sie später zu wissen bekommen) automatisch auf die RAM - Disk P: kopiert werden. Deshalb erstellen Sie die Datei "A:PROFILE.SUB" mit folgendem Inhalt:

```
$COPY *.MNU P:
```

Dadurch wird das Programm \$COPY automatisch geladen und ausgeführt.

```
*****  
*           der Warmstart           *  
*****
```

### A.5.2. Der Warmstart

Der Computer läuft also. Auf dem Bildschirm sehen wir die Bereitschaftsanzeige und konnten mit der Arbeit beginnen. Das ist richtig - der Vollständigkeit halber wollen wir uns dennoch an dieser Stelle mit dem Warmstart des Betriebssystems beschäftigen.

Wie bereits gesagt, ist die Bereitschaftsanzeige das Lebenszeichen eines Teiles des Betriebssystems. Dieser Programmteil von DOS nennt sich SHELL, weil er sozusagen die "äußere Schale" des Betriebssystems (wie bei einer Muschel, engl.: shell) darstellt. Er hat die Aufgabe, Ihnen zu Diensten zu sein und Ihre Eingaben in sinnvolle Aktionen umzusetzen.

Wenn er einmal ein Programm geladen hat, ist seine Anwesenheit nicht mehr erforderlich, ja sogar unerwünscht, damit Hauptspeicherplatz für die nützlicheren Programme frei wird. Es ist daher üblich, daß Programme den SHELL einfach ignorieren und ihn überschreiben. Wenn sie dann Ihre Arbeit beendet haben, fordern sie den immer noch im Speicher vorhandenen und lebensnotwendigen Teil von DOS auf, den SHELL vom Laufwerk P: (der RAM - Disk) nachzuladen und die Steuerung wieder an ihn zu übergeben. So geschieht es, daß nach jedem ordentlichen Abschluß eines Programmes wieder die Bereitschaftsanzeige von SHELL auf dem Bildschirm erscheint.

Diesen Vorgang des Nachladens nennt man Warmstart. Weil SHELL ein ziemlich kleines Programm ist, geht das sehr flott. Wenn Sie durchaus darauf aus sind, Ihr System zum Absturz zu bringen, dann löschen Sie doch einfach mal die Datei "SHELL.SYS" auf der RAM - Disk P: Sie ist allerdings durch das Setzen der Datei - Attribute SYSTEM und R/O vor solchen Albernheiten geschützt. Was Attribute sind, lesen Sie bitte im Kapitel über Dateinamen nach.

```
*****  
*           Disketten wechseln           *  
*****
```

### A.5.3. Das Wechseln von Disketten

Als unvorbereiteter Computerbenutzer konnten Sie davon ausgehen, daß gerade dieser Vorgang die einfachste Sache überhaupt sei. Sicher, man braucht ja nur den Diskettenschacht zu öffnen - ein Griff und - die Sache ist erledigt. Im Prinzip ja, aber...

Wenn Sie ein Anwendungsprogramm zum Beispiel zur Textverarbeitung oder zur Finanzbuchhaltung benutzen, sollten Sie nur dann eine Diskette wechseln, wenn Sie dazu aufgefordert werden.

Warum das Ganze?

Aus Gründen der Datensicherheit prüft EOS bei jedem Zugriff auf eine Speichereinheit, ob dort immer noch die gleiche Diskette vorhanden ist, wie beim letzten Zugriff. Stellt sich heraus, daß die Diskette gewechselt wurde, ohne EOS Bescheid zu sagen, weigert es sich, auf diese Diskette zu schreiben und tut seinen Unwillen durch eine Systemfehlermeldung kund.

Diese Mimik ist bei der Vielzahl der unter EOS lauffähigen Programme eine notwendige Sicherheitsmaßnahme. Wenn Sie hingegen Disketten wechseln, während EOS auf ein Kommando wartet, kann Ihnen nichts passieren. Da ja jetzt ein neues Kommando kommt, rechnet EOS damit, daß zu diesem neuen Kommando auch neue Disketten benötigt werden. Also: keine Angst vor dem Diskettenwechsel, ehe Sie ein EOS - Kommando eingeben!

```

*****
*           Tastatur-Eingaben           *
*****

```

#### A.5.4. Tastatureingaben

Hinter der Bereitschaftsanzeige ">" sehen Sie einen hellen kleinen Block. Diesem Block kommt die größte Wichtigkeit zu. Er zeigt an, wo auf dem Bildschirm das nächste Zeichen, das Sie eintippen, angezeigt wird. Probieren Sie es einmal aus: Wenn Sie irgend ein Zeichen eingetippt haben, erscheint dieses genau an der Stelle, wo sich der Block befindet, und dieser wandert eine Stelle weiter. Dieser kleine Block, der gelegentlich auch mal blinken kann, nennt sich Bildschirmpositionsanzeiger. Dieser Wort - Lindwurm wird im allgemeinen Sprachgebrauch zum englischen Wort "Cursor" abgekürzt. Merken Sie sich diesen Begriff gut; er wird Ihnen später noch oft genug über den Weg laufen.

Wenn Sie eine Buchstaben-Taste drücken, erscheint dieser Buchstabe auf dem Bildschirm. Wie mit einer Schreibmaschine können Sie jetzt Ihre Anweisungen an EOS übermitteln. Sie schreiben ihm eine Mitteilung und schicken sie durch das Niederdrücken der ENTER - Taste ab. Als Darstellung wählen wir ab jetzt die Zeichenfolge <ENTER>. Immer, wenn Sie diese Zeichenfolge sehen, wird von Ihnen erwartet, daß Sie dort die ENTER - Taste drücken.

Neben den normalen (und teilweise auch etwas extravaganten) Schriftzeichen gibt es einen ganzen Block von nicht darstellbaren Zeichen, die Steuer- oder Controlzeichen. Diese steuern die Bildschirmdarstellung. Der Wagenrücklauf ist beispielsweise ein derartiges Steuerzeichen. Es ist gelegentlich nötig, derartige Steuerzeichen einzugeben. Daher hat man sich einfallen lassen, diese Steuerzeichen nach dem Alphabet durchzumerieren. Zusätzlich haben Sie an der linken Seite der Tastatur eine Taste namens "CTRL". Wenn Sie nun diese Taste gleichzeitig mit einer Buchstabentaste drücken - wobei es egal ist, ob Sie Groß- oder Kleinschreibung verwenden -, wird ein Steuerzeichen generiert.

Als Beispiel generieren Sie mal das Steuerzeichen BEL ( Glocke ). Im allgemeinen Sprachgebrauch heißt dieses Zeichen Control-G. Drücken Sie mal die Control - Taste gleichzeitig mit dem "G". Wie Sie hören, piept's (nicht bei Ihnen, sondern im Rechner!).

Als Darstellung für CONTROL-Zeichen wählen wir den Aufwärts-Pfeil. In anderen Dokumentationen wird die Abkürzung CTRL- verwendet:

^G oder CTRL-G bedeutet: geben Sie an dieser Stelle ein CONTROL-G ein.

```
*****
*       Tastatur-Eingaben       *
*****
```

### A.5.5. Stopzeichen

Diese Zeichen bewirken einen Abbruch des laufenden Programms. Sie können als erstes Zeichen einer Eingabezeile eingegeben werden. Standard-Stopzeichen ist das ^C (Control-C). Darüberhinaus wird auch gern ^A oder <ESCAPE> als Stopzeichen verwendet. Sie können solche Stopzeichen mit Hilfe von => \$CONFIG frei definieren. Immer, wenn ein Stopzeichen ordnungsgemäß von EOS erkannt wurde, erscheint die Meldung

<Restart>

auf dem Bildschirm.

EOS bietet Anwenderprogrammen die Möglichkeit, die Erkennung von Stopzeichen zu unterbinden; wundern Sie sich daher nicht, wenn die Eingabe eines Stopzeichens unter Umständen keine Wirkung hat! Die meisten EOS - Dienstprogramme ignorieren beispielsweise das Stopzeichen oder lassen es nur an bestimmten Stellen zu.

### A.5.6. Anhalten der Bildschirmausgabe

Viele Programme geben ihre Bildschirmausgaben in einem Tempo aus, daß kein normaler Mensch mehr mit dem Lesen nachkommt. Daher schien es nötig, ein spezielles Zeichen zu definieren, das den Bildschirm anhält. Dieses Zeichen ist das

^S (Control-S).

Geben Sie dieses ein, halt die Bildschirmausgabe sofort an. Nach dem Anhalten haben Sie dann verschiedene Möglichkeiten der Eingabe:

^Q (Control-Q) läßt die Ausgabe weiterlaufen.

^C (Control-C) oder ein anderes Stopzeichen bricht das Programm ab.

^P (Control-P) schaltet den Drucker ein. Der Drucker läuft dann so lange parallel zum Bildschirm mit, bis er durch ein erneutes ^P oder durch das SHELL - Kommando => DRUCKER AUS wieder angehalten wird.

Auch hier sei nicht verschwiegen, daß manche Programme die Eingabe von Control-S ganz einfach ignorieren. Das Dienstprogramm => \$TYPE beispielsweise ist nicht durch Control-S anzuhalten, aus dem einfachen Grunde, daß es sowieso nach der Ausgabe einer Bildschirmseite von selbst anhält.

```
*****
*       Dateinamen       *
*****
```

## A.6. Dateinamen

Bereits in der allgemeinen Einführung hatten wir erwähnt, daß die Namen der Dateien Teil der Normung von EOS und anderen CP/M-kompatiblen Betriebssystemen sind. Da nun nahezu alle Kommandos direkt oder indirekt mit Dateien zu tun haben, ist jetzt der Augenblick gekommen, wo wir über den Gebrauch der Dateinamen sprechen müssen.

### A.6.1. Eindeutige Dateinamen

Der Name einer Datei besteht aus zwei Teilen, dem Dateinamen an sich und dem Dateityp.

```
KUNDEN.DTA   FIBU.DTA   D12345
```

Das sind einige typische Dateinamen. Links vom Punkt steht der Dateiname, rechts davon der Dateityp. Für den Dateinamen haben wir 8 Zeichen zur Verfügung. Der wahlweise Dateityp kann bis zu 3 Zeichen lang sein. Es gibt Fälle, in denen der Dateityp unbedingt erforderlich ist. Wir kommen noch darauf zu sprechen.

Bei der Formulierung eines Dateinamens können mit einigen Ausnahmen alle darstellbaren Zeichen verwendet werden. Steuerzeichen scheiden aus. Ebenso sind die Zeichen

```
= (Gleichheitszeichen)
< (Kleiner - Zeichen)
> (Großer - Zeichen)
: (Doppelpunkt)
. (Punkt)
, (Komma)
Leerstelle ( blank )
A
U (aus Kompatibilität zu CP/M)
```

nicht zulässig, weil diese eine besondere Bedeutung bei der Eingabe von Kommandos haben.

Darüberhinaus verlangen einige Programme, daß der Dateiname mit einem Buchstaben beginnt und nur Buchstaben und Zahlen enthält. Auch können bestimmte Dateitypen vorgeschrieben sein. Mehr über diese Vorschriften erfahren Sie im Allgemeinen in den entsprechenden Programmbeschreibungen.

Auf der Kommandoebene von EOS werden bei allen Ihren Eingaben die Kleinbuchstaben in Großbuchstaben verwandelt. Sie können auf keine Datei gezielt Bezug nehmen, deren Name von einem Programm erzeugt wurde, das Kleinbuchstaben verwendet hat.

```
*****
*           Dateinamen           *
*****
```

Hier nun einige Beispiele für gültige Dateinamen:

```
ADRESSEN.STM   DATEI1   DATEI2.BAK
```

Und einige fehlerhafte Dateinamen:

```
ADRESSEN1.STM   = der Dateiname ist länger als
                  8 Zeichen. Der Rest wird ignoriert

DATEI:1.BAK     = enthält als unzulässiges Zeichen
                  einen Doppelpunkt

KUNDEN. 01     = enthält als unzulässiges Zeichen
                  eine Leerstelle ( blank ) im
                  Datei-Typ
```

#### A.6.1.1. Der Dateityp

Lassen Sie uns auf den Dateityp noch etwas näher eingehen, da ihm in vielen Fällen eine besondere Bedeutung zukommt.

Mit Hilfe der drei Zeichen, die uns für den Dateitypen zur Verfügung stehen, läßt sich naturgemäß eine Fülle von Kombinationen bilden. Zunächst einige Dateitypen, die im Zusammen mit EOS und seiner internationalen Normung sehr wichtig sind und von Ihnen niemals am falschen Ort verwendet werden sollten:

- .COM = bezeichnet eine Datei, deren Inhalt ein direkt ausführbares Programm ist. Solche Programme sind aus der Kommando-Ebene von EOS direkt aufrufbar. Wir werden jene Programme, die zum Lieferumfang von EOS gehören, unter dem Abschnitt Dienstprogramme eingehender besprechen.
- .SUB = Dateien dieses Typs enthalten Kommandofolgen, die vom Dienstprogramm => \$DO in Kommandos umgesetzt werden können.
- .SYS = diese Dateien enthalten Ihr EOS - Betriebssystem.
- .RSX = EOS - Funktionsmodul (siehe auch das Systemhandbuch, Abschnitt A).
- .BAK = kennzeichnet eine Back-Up-Datei, in der alte Versionen von Daten aufbewahrt werden. Textbearbeitungsprogramme erzeugen solche Dateitypen automatisch. Eine besondere Funktion des Dienstprogramms => \$COPY berücksichtigt diesen Umstand. Sie sollten diesen Dateityp deshalb immer im Sinne des Erfinders benutzen.

```
*****
*           Dateinamen           *
*****
```

**.\$\$\$** = ein weiterer Dateityp, der temporäre Hilfsdateien kennzeichnet. Auch dieser Typ ist bei der Benutzung von => \$COPY von besonderer Bedeutung. Alle EOS - Dienstprogramme, die mit temporären Dateien arbeiten, erzeugen Dateien dieses Typs.

**.MNU** = Hilfsdateien für das EOS - Menuprogramm.

**.KEY** = Dateien, die Tastatur - Definitionen enthalten.

Es folgen nun einige Dateitypen, die aus der Sicht von EOS keine besondere Berücksichtigung finden, aber international einheitlich verwendet werden. Sicher ist es eine gute Angewohnheit, sich daran zu halten:

**.DTA** = diese Datei enthält Daten. Welche das sind, sollte der Dateiname näher erläutern.

**.NDX** = diese Datei enthält einen Index auf Datensätze. Es ist ein guter Brauch, NDX und DTA mit dem gleichen Dateinamen zu versehen, damit auf dem ersten Blick die Zusammenhänge klar sind.

**.OVR**

**.OVL** = in diesen Dateien befinden sich Programmsegmente bzw. Overlays, die von einem anderen Programm während der Ausführung nachgeladen werden. Auch hier ist es üblich, den Bezug zwischen dem Hauptprogramm und den Overlays im Dateinamen deutlich zu machen.

**.REL**

**.CRL**

**.IRL** = kennzeichnen ein Objektmodul in einem besonderen verschiebbaren Format. Solche Dateien werden von ASSEMBLERN oder COMPILERN erzeugt, um im nächsten Arbeitsgang, dem Linken (binden) zu einer .COM-Datei, also einem direkt ausführbaren Programm verarbeitet zu werden.

**.TXT**

**.DOC** = solche Dateien enthalten Erläuterungen, also Texte, welche im Allgemeinen Programmfunktionen oder Disketteninhalte beschreiben.

Darüberhinaus ist es auch üblich, den Quelltext von Programmiersprachen im Dateityp zu kennzeichnen. Einige Übersetzer (Compiler) und Interpreter (wie soll man das einigermaßen verständlich übersetzen?) verlangen die Einhaltung des Dateityps und erzeugen auch ganz gezielt bestimmte Dateitypen:

```
*****
*           Dateinamen           *
*****
```

```
.C   = C (die Sprache, in der EOS geschrieben wurde)
.BAS = BASIC           .COB = COBOL
.PLI = PL/I           .PAS = PASCAL
.ASM = ASSEMBLER     .MAC = ASSEMBLER
.ALG = ALGOL          .FOR = FORTRAN
.MMP = MUMPS          .LSP = LISP
```

### A.6.2. Mehrdeutige Dateinamen

Jetzt wird es ein wenig kompliziert. Bei der Eingabe von Kommandos können nämlich auch mehrdeutige Dateinamen verwendet werden. Diese Technik ist, wie Sie bald feststellen werden, sehr nützlich und erlaubt es Ihnen, auf eine ganze Gruppe von Dateien Bezug zu nehmen.

Diese Mehrdeutigkeit ist allerdings nur im Umgang mit System-Kommandos, Dienstprogrammen und speziell darauf eingerichteten Programmen von Nutzen. Sie wird durch die Verwendung von sogenannten Blindzeichen (wildcards) bei der Vorgabe von Dateinamen erreicht. Diese Blindzeichen sind

- ? - das Fragezeichen und
- \* - das Sternzeichen

Wir empfehlen Ihnen, die Funktion mit Hilfe des => \$DIR - Kommandos zu trainieren.

#### A.6.2.1. Das Fragezeichen

Es steht, wenn es bei der Vorgabe eines Kommandos im Dateinamen oder Dateityp verwendet wird, für ein beliebiges Zeichen an dieser Stelle des Dateinamens.

Nehmen wir einmal an, auf der aktuellen Speichereinheit befinden sich folgende Dateien:

```
KUNDEN1.DTA      KUNDEN2.DTA      KUNDEN3.DTA
KUNDEN1.BAK      KUNDEN2.BAK
```

so können wir durch Vorgabe des Dateinamens "KUNDEN?.DTA" die Dateien KUNDEN1.DTA, KUNDEN2.DTA und KUNDEN3.DTA ansprechen. Andersherum wäre es möglich, durch "KUNDEN1.???" die Dateien KUNDEN1.DTA und KUNDEN1.BAK auszuwählen.

```
*****
*           Dateinamen           *
*****
```

#### A.6.2.2. Das Sternzeichen

Es ersetzt eine beliebige Folge von Zeichen und erubrigt das Eintippen von mehreren Fragezeichen.

\*.\*

bedeutet, daß alle Dateien auf der Speichereinheit gemeint sind. Wenn wir bei dem Beispiel bleiben und annehmen, daß sich die Dateien

```
KUNDEN1.DTA      KUNDEN2.DTA      KUNDEN3.DTA
KUNDEN1.BAK      KUNDEN2.BAK
```

— auf der Speichereinheit befinden, können wir durch Vorgabe von

```
*.DTA           die Dateien
```

```
KUNDEN1.DTA     KUNDEN2.DTA     KUNDEN3.DTA
```

ansprechen und durch

```
KUNDEN1.*       die Dateien
```

```
KUNDEN1.DTA     KUNDEN1.BAK
```

in unser Kommando aufnehmen, ohne daß acht Fragezeichen einzugeben waren. Es lohnt sich aber schon, wenn man zwei Fragezeichen sparen kann - wie das folgende Beispiel für die Verwendung von "\*" im Dateityp zeigt:

```
K*.D*           ergibt
```

```
KUNDEN1.DTA     KUNDEN2.DTA     KUNDEN3.DTA
```

#### A.6.3. Das Laufwerk

Bei den bisherigen Beispielen sind wir immer davon ausgegangen, daß sich die betroffenen Dateien auf dem aktuellen Laufwerk befinden, jenem Laufwerk also, die uns EOS in seiner Statuszeile mitteilt. Wie man das aktuelle Laufwerk wechselt, erfahren Sie auch bald. Hier geht es darum, Dateien anzusprechen, die nicht auf dem aktuellen Laufwerk befinden.

```
*****
*           Dateinamen           *
*****
```

Das Laufwerk wird durch den Buchstaben A bis P symbolisiert. Sie erinnern sich doch? - je nach der Ausbaustufe Ihres Systems kommen für die 16 möglichen Laufwerke die Buchstaben A: bis P: in Frage. Dieser Buchstabe kann in den Dateinamen mit aufgenommen werden. Er steht an erster Stelle im Dateinamen und wird, damit EOS ihn nicht zum Namen hinzurechnet, durch einen Doppelpunkt vom Rest des Namens abgetrennt:

```
A:KUNDEN1.DTA   B:KUNDEN1.BAK   C:*.BAK   A:KUNDEN1.*
```

Die obigen Beispiele zeigen die korrekte Verwendung des Laufwerks im Dateinamen. Nun wird auch klar, warum im Namen selbst kein Doppelpunkt verwendet werden darf.

#### A.6.4. Dateiattribute

Jede Datei kann mit einem oder mehreren Attributen versehen werden. Diese Attribute sagen neben dem Namen und dem Typ einer Datei noch etwas über die Verwendungsart einer Datei aus. Attribute sind normalerweise nicht sichtbar. Sie können sich diese jedoch durch eine Sonderform des => \$DIR - Kommandos anzeigen lassen. Die Attribute selbst werden durch das Dienstprogramm => \$SET gesetzt bzw. gelöscht.

Die meisten möglichen Attribute sind fest für EOS reserviert. Drei Attribute können jedoch von Ihnen nach Belieben gesetzt werden:

- SYSTEM - Dieses Attribut macht eine Datei global zugänglich. Sie kann auch von anderen Benutzerbereichen aus gelesen werden. Außerdem erscheint eine Datei mit dem SYSTEM - Attribut nicht im normalen Inhaltsverzeichnis eines Laufwerks, was unter Umständen erheblich zur Übersichtlichkeit desselben beiträgt.
- BACKUP - Das Hilfsprogramm => \$COPY setzt bei jeder Datei, die gesichert wurde, dieses Attribut. EOS selbst löscht dieses Attribut automatisch, wenn die Datei geändert wurde. Somit ist eine Datei, die noch nicht gesichert wurde, schnell zu erkennen. Eine Sonderfunktion von \$COPY kopiert nur Dateien, die dieses Attribut nicht besitzen.
- R/O - Dateien mit dem R/O (Read Only) - Attribut können weder geändert noch gelöscht werden.

```
*****  
*      Benutzerbereiche      *  
*****
```

### A.7. Benutzerbereiche

Jedes Laufwerk kann in 16 sogenannte Benutzerbereiche unterteilt werden, die von 0 bis 15 durchnummeriert werden. Sie befinden sich normalerweise im Benutzerbereich 0. Wenn Sie nun beispielsweise auf der Winchesterplatte die Übersicht zu verlieren drohen, können Sie mit Hilfe des in SHELL eingebauten Kommandos => BEN den Benutzerbereich wechseln. Alle Programme können nur auf Dateien zugreifen, die sich innerhalb des selben Benutzerbereichs befinden. Eine Ausnahme bildet der Benutzerbereich 0. Dateien, die sich in diesem Benutzerbereich befinden, und mit dem SYSTEM - Attribut versehen sind, gelten als global. Sie können von jedem anderen Benutzerbereich aus gelesen, aber nicht geändert werden. Somit können Sie vermeiden, daß Sie in jedem Benutzerbereich Kopien aller Programme, die Sie benötigen sollten, vorrätig halten müssen.

Wenn Sie als SHELL das mitgelieferte Menuprogramm benutzen, haben Sie Pech; mit diesem Programm ist ein Wechsel des Benutzerbereichs nicht möglich. Ihnen bleibt daher nur der Benutzerbereich 0 für Ihre Aktivitäten.

```

*****
*           Die Statuszeile           *
*****

```

#### A.8. Die Statuszeile des MZ-3500

Die 25. Zeile Ihres Bildschirms hat eine besondere Funktion, die sich bereits durch die Invers - Darstellung ankündigt: Sie ist als Informations-Zeile definiert und von den normalen Bildschirm - Funktionen abgetrennt. In dieser Zeile kann entweder eine System - Statuszeile oder eine frei definierbare Benutzer - Statuszeile angezeigt werden. Es kann jederzeit zwischen der System- und der Benutzer - Statuszeile umgeschaltet werden. Beide Zeilen sind völlig voneinander unabhängig.

Im Nachfolgenden wollen wir uns mal ansehen, was uns EOS in der System - Statuszeile alles erzählt. Wie Sie sehen, ist diese Zeile in 8 Felder unterteilt. Diese Felder wollen wir einmal als Felder 1 bis 8 bezeichnen, wobei das Feld 1 das Feld ganz links und Feld 8 das Feld ganz rechts bezeichnet. Diese Felder haben im Einzelnen die folgende Bedeutung:

Feld 1 : Hier wird der Name des laufenden Programms angezeigt. Wenn das SHELL - Programm aktiv ist und Sie somit Kommandos eingeben können, steht hier der Schriftzug "E O S".

Feld 2 : Anzeige des aktuell angewählten Laufwerks.

Feld 3 : Anzeige des aktuell angewählten Benutzerbereichs. Wenn Sie Programme benutzen, die zwischen Laufwerken und Benutzerbereichen hin- und herschalten, können Sie das in der Statuszeile verfolgen.

Feld 4 : Aktueller Zustand der Tastatur und des Bildschirms:

'gesperrt'	-	alle Tastatureingaben gesperrt
'Monitor'	-	Bildschirm ist im Monitor - Mode
'Key Def'	-	Eingabe von Tastendefinitionen
'BenStat'	-	Setzen der Benutzer - Statuszeile
'SysStat'	-	Setzen der System - Statuszeile
'Graphik'	-	Bildschirm ist im Pseudo - Grafikmode
'Editieren'	-	Bildschirm - Editor aktiv
'Hard Copy'	-	Hardcopy des Bildschirms wird ausgegeben

Genauere Informationen finden sich im Abschnitt D des Systemhandbuchs. Dort ist der Bildschirm im Einzelnen beschrieben.

Feld 5 : frei für den Anwender.

Feld 6 : Hier erfolgt die Anzeige 'Execute', wenn EOS gerade eine ".SUB" - Datei abarbeitet.

Feld 7 : Anzeige des Datums.

Feld 8 : Anzeige der Uhrzeit.

```
*****
*           die Tastatur           *
*****
```

### A.9. Die Tastatur des MZ-3500

Ihr MZ-3500 hat neben der normalen Tastatur eine Reihe von Sondertasten. Diese sind frei programmierbar und teilweise auch schon von EOS vordefiniert. Dem Zehnerblock kommt in Verbindung mit der CONTROL - Taste eine besondere Bedeutung zu. Drücken Sie eine Taste des Zehnerblocks zusammen mit der CONTROL - Taste, können Sie verschiedene Funktionen auslösen. Merke: diese Funktionen stehen nur auf dem Zehnerblock zur Verfügung; die in der Normaltastatur enthaltenen Zahlen haben in Verbindung mit der CONTROL - Taste keine Wirkung!

Im Nachfolgenden wollen wir uns mit diesen Funktionen befassen:

- CTRL-0 : Ihr MZ-3500 hat einen internen Tastatur - Eingabepuffer von 140 Zeichen Länge, so daß Sie bereits Eingaben tippen können, ehe der Rechner seine Eingabebereitschaft anzeigt. Diese Fähigkeit wird "Type Ahead" genannt. Wenn Sie einmal der Meinung sein sollten, daß Sie Unsinn eingetippt haben, drücken Sie CTRL-0. Dadurch wird der Eingabepuffer vollständig gelöscht.
- CTRL-1 : Die Wirkung der SHIFT - und LOCK - Tasten wird umgekehrt. Wenn Sie mittels => \$DEVICE Ihren Rechner auf Kleinschreibung konfiguriert haben, können Sie mit Hilfe dieser Funktion eine normale Großschreibung erreichen (die LOCK - Taste schaltet leider die gesamte Tastatur einschließlich der Sonderzeichen auf Großschreibung um).
- CTRL-2 : Umschaltung zwischen DIN - Zeichensatz und ASCII - Zeichensatz. Diese Funktion ist nur in Verbindung mit dem speziellen EOS - Zeichengenerator verfügbar.
- CTRL-3 : Die Uhr in der Statuszeile ein- bzw. ausschalten.
- CTRL-4 : Der Bildschirminhalt wird über die Parallelschnittstelle auf einen Drucker ausgegeben (Hard Copy).
- CTRL-5 : Der Graphik - Bildschirm wird auf den SHARP - Tintenstrahldrucker IO-700 ausgegeben.
- CTRL-6 : Die Anzeige der hochauflösenden Graphik wird ein- bzw. ausgeschaltet.
- CTRL-7 : Beginn einer Tastendefinition.
- CTRL-8 : Ende einer Tastendefinition.
- CTRL-9 : Bildschirm und Tastatur neu initialisieren. Die Standard - Tastendefinitionen werden wieder eingesetzt.

```
*****
*           die Tastatur           *
*****
```

### A.9.1. Die Definition von Sondertasten

Sie können mit Hilfe der Tastatur - Funktionen CTRL-7 und CTRL-8 selbst Ihre Sondertasten definieren. Das geht wie folgt vor sich:

1. Sie drücken CTRL-7. Es erscheint ein Großer - Zeichen ">" auf dem Bildschirm.
2. Sie drücken die Taste, die Sie neu definieren wollen. Die Taste wird im Klartext angezeigt. Wenn Sie beispielsweise die Funktionstaste I neu definieren wollen, erscheint der Text "I=". In der Statuszeile erscheint jetzt im Feld 4 der Text "Key Def". Falls die Taste, die Sie definieren wollen, nicht neu definierbar ist, hören Sie ein Piepen.
3. Sie geben Ihren Text ein. Steuerzeichen werden als auf der Seite liegende Buchstaben dargestellt.
4. Sie beenden Ihre Definition durch Drücken der Taste CTRL-8. Es erscheint das Kleiner - Zeichen "<".

Wenn Sie sich vertippt haben sollten, können Sie die Definition jederzeit durch die Eingabe von CTRL-7 beenden.

Lassen Sie uns ein Beispiel durchgehen. Wir wollen die Taste "INS" in der oberen rechten Bildschirmcke mit dem Text "Einfugen!" belegen. Der Vorgang sieht dann so aus:

Sie	der Rechner
CTRL-7	>
INS	INS=
Einfugen!	
CTRL-8	<

Wenn Sie nun die INS - Taste drücken, erscheint jedesmal der Text "Einfugen!"

Ein kleiner Zusatz: wenn Sie eine Taste gezielt auf den EOS - Standardwert zurücksetzen wollen, drücken Sie an Stelle des Textes die CMD - Taste. Jede Taste kann mit insgesamt 128 Zeichen belegt werden. Für alle Tasten insgesamt stehen etwa 300 Zeichen zur Verfügung.

Die Funktionstasten können auch von einem Programm aus programmiert werden. Näheres ist im Abschnitt D des Systemhandbuchs zu finden.

Und noch etwas: es gibt das Dienstprogramm => \$KEYDEFS, um Ihre persönlichen Definitionen in einer Datei zu speichern und sie aus dieser Datei wieder zu laden. Wie das gemacht wird, lesen Sie bitte bei der Beschreibung von \$KEYDEFS nach.

```
*****
*           die Tastatur           *
*****
```

### A.9.2. Vordefinition der Funktionstasten

Die meisten Funktionstasten werden vom System vordefiniert:

Taste	Definition	Funktion
-----	-----	-----
fest definiert:		
CMD	ESC	ESCAPE - Taste
ENTER	CTRL-M	Zeile an EOS übergeben
umprogrammierbar:		
TAB	TAB	horizontaler Tabulator
INS	CTRL-O	Zeichen einfügen
DEL	RUBOUT	Zeichen löschen
DEB	CTRL-Underline	Monitor Mode ein/aus
BRK	CTRL-S	Bildschirmausgabe anhalten
CTRL-BRK	CTRL-C	Programm abbrechen
UP	CTRL-K	Cursor eine Zeile rauf
DOWN	CTRL-V	Cursor eine Zeile runter
LEFT	CTRL-H	Cursor ein Zeichen links
RIGHT	CTRL-L	Cursor ein Zeichen rechts
HOME	CTRL-^	Cursor nach links oben (Homeposition)
CTRL-HOME	CTRL-Z	Bildschirm löschen
CL	CTRL-X	Eingabezeile löschen
-ENTER	ESC 'T'	Bis Zeilenende löschen
OO	'O' 'O'	Doppel-Null
Dezimalpunkt	','	Dezimalkomma

Sie sollten bei der Umprogrammierung der Cursor - Pfeiltasten ein wenig daran denken, daß der eingebaute Bildschirm - Editor von EOS genau die vordefinierten Steuerzeichen benötigt, um zu funktionieren.



**Abschnitt B**  
**Die Eingabe von Kommandos**

```

*****
*   Kommando - Eingaben   *
*****

```

Die vornehmste Aufgabe von EOS ist es, Eingaben eines Benutzers als sinnvolle Kommandos zu erkennen und diese auszuführen. Diese Aufgabe wird von einem Spezialprogramm namens SHELL durchgeführt. Jedesmal, wenn Sie in der Lage sind, ein Kommando einzugeben oder ein Programm auszuwählen, kommunizieren Sie mit diesem Programm. Sagen wir es einmal anders: Wenn kein weiteres Programm aktiv ist, ist SHELL aktiv.

Im Normalfall gibt SHELL ein bestimmtes Zeichen aus, um anzuzeigen, daß nun ein Kommando eingegeben werden kann, und wartet dann auf diese Eingabe. Eine derartige Eingabe wird dann nach bestimmten Regeln interpretiert. Ist die Zeile als Kommando erkannt worden, wird das entsprechende Programm gesucht und geladen und die Kontrolle an selbiges übergeben.

Diese Art des Startens von Kommandos setzt allerdings doch schon einige Kenntnisse der System - Philosophie von EOS voraus. Um jedoch allen Benutzern gerecht zu werden, vor allem denjenigen, die nicht mit EOS als Betriebssystem, sondern mit bereits fertigen Programmen arbeiten wollen, wurden zwei verschiedene SHELL - Programme entwickelt. Das Programm, mit dem Ihre Diskette ausgeliefert wird, nennt sich SHELL.SYS und führt eine normale Anforderung eines Kommandos und dessen Interpretation durch. Das andere jedoch, USRSHELL.SYS genannt, bietet eine ganz andere Möglichkeit der Programmwahl: es stellt Menus zur Verfügung, aus denen per Tastendruck das gewünschte Programm ausgewählt wird. Einfacher geht es nicht. Um dieses Programm als SHELL zu aktivieren, muß die Datei SHELL.SYS irgendeinen anderen Namen bekommen (sie kann auch gelöscht werden; jedoch tun Sie das besser nicht auf Ihrer Original - Systemdiskette!). Danach wird USRSHELL.SYS in SHELL.SYS umbenannt. Beim nächsten Kaltstart bekommen Sie statt des Kommando - Interpreters das Menuprogramm zu sehen.

Das Menuprogramm muß vor Gebrauch an die zu startenden Programme angepaßt werden. Diese Arbeit wird jedoch meist von demjenigen übernommen, der Ihre Programme installiert. Dieser Mensch wird dann auch die entsprechende Installation des Menuprogramms vornehmen. Sie als "Normalanwender" werden meist mit diesen Arbeiten nicht konfrontiert werden. Sie dürfen daher gleich mit dem nächsten Abschnitt beginnen. Anwender, die erst einmal mit dem Kommando - Interpreter arbeiten wollen, sei die Lektüre des nachfolgenden Abschnitts empfohlen.

```
*****
*   Kommando - Interpreter   *
*****
```

### B.1. Der Kommando - Interpreter

Wenn Sie Ihre EOS - Systemdiskette zu ersten Mal in den Diskettenschacht einlegen und die Maschine einschalten, bekommen Sie irgendwann ein Großes - Zeichen ">" auf dem Bildschirm zu sehen, ehe sich der Rechner nach dem Ladevorgang beruhigt und die Lichter an Ihrem Diskettenlaufwerk erloschen. Dieses Zeichen ist die Bereitschaftsanzeige von EOS. Es zeigt Ihnen an, daß der Kommando - Interpreter auf ein Kommando von Ihnen wartet. Um es noch deutlicher zu machen, erscheint in der Statuszeile ganz links die Schriftfolge "E O S". Sie können jetzt einen beliebigen Text eingeben. Der Interpreter versucht, diesen als Kommando zu interpretieren. Wie er das macht, wollen wir uns im Nachfolgenden anschauen. Eine Zusatzbemerkung: Falls Sie Kommandos ausführen, erscheinen je nach Verschachtelung der \$DO - Laufe mehrere ">" - Zeichen hintereinander. Genaueres steht bei der Beschreibung des Programms \$DO.

#### B.1.1. Der Bildschirm - Editor

Ehe wir auf die internen Dinge des Kommando - Interpreters eingehen, drücken Sie doch einmal eine der Pfeiltasten in der rechten Hälfte der Tastatur. Sie werden sehen, daß man damit den Cursor in beliebige Richtungen bewegen kann. Wenn Sie wollen, können Sie irgend etwas nettes irgendwo auf dem Bildschirm verewigen.

Die ganze Sache hat natürlich einen handfesten Hintergrund: Wenn Sie einmal ein Kommando eingegeben haben, sich aber vertippt haben sollten, können Sie den Tippfehler mit Hilfe der Pfeiltasten wieder korrigieren. Sie haben zusätzlich noch zwei Tasten für derartige Änderungen zur Verfügung. Beide sind in der rechten oberen Ecke des Bildschirms untergebracht. Die eine, die "DEL" - Taste, löscht das Zeichen, das sich gerade an der Cursor - Position befindet. Die andere, die "INS" - Taste, macht genau das umgekehrte: sie fügt an der Cursor - Position ein Leerzeichen ein, so daß Sie sich für Eingabekorrekturen Platz schaffen können.

Wenn Sie einmal einen ganzen Haufen Zeichen wahllos eintippen, werden Sie merken, daß Sie die erste Spalte des Bildschirms nicht beschreiben können. Das hat den einfachen Grund, daß dort ja immer die Bereitschaftsanzeige ">" erscheint. Diese Anzeige soll auch stehen bleiben und darf deshalb nicht überschrieben werden.

Eine Sonderfunktion hat die "ENTER" - Taste. Wenn Sie diese drücken, hat der ganze Spuk ein Ende. Der Kommando - Interpreter nimmt dann die Zeile, in der Sie gerade Ihren Cursor stehen hatten. Er ignoriert die erste Spalte der Zeile (die Sie ja sowieso nicht ändern konnten), und versucht, die ganze Zeile als Kommando aufzufassen.

```
*****
* Kommando - Interpretation *
*****
```

### B.1.2. Die Interpretation der Eingabe

Wenn Sie jetzt eine Zeile eingegeben und anschließend "ENTER" gedrückt haben, haben Sie ein Kommando eingegeben (egal, wie unsinnig diese Zeile auch ist). Der Interpretier nimmt sich das erste Wort vor; dieses ist für ihn immer das Kommandowort (es gibt noch Sonderfälle, auf die wir aber erst später eingehen wollen). Nehmen wir einmal an, Sie haben eingegeben:

```
>Das ist aber ein schöner Rechner!
```

Das erste Wort - das Kommandowort - wäre somit das Wortchen "Das". Was macht der Rechner nun mit der Eingabe? Zuerst wird die gesamte Zeile in Großbuchstaben umgesetzt:

```
DAS IST ABER EIN SCHONER RECHNER!
```

Dann wird das erste Wort näher besehen. Der Interpretier nimmt einfach an, daß dieses Wort der Name einer Programmdatei ist. Da ihm der Dateityp fehlt, wird automatisch der Typ ".COM" angefügt:

```
DAS.COM
```

Nun wird nach dieser Datei gesucht. Wird diese gefunden, so wird sie geladen und ausgeführt. Der Rest der Kommandozeile wird dem Programm "DAS.COM" auf den Weg mitgegeben:

```
IST ABER EIN SCHONER RECHNER!
```

"DAS.COM" kann diesen Zeilenrest selbst interpretieren; es könnte beispielsweise die Datei "IST" einrichten und mit lauter "ABER"s vollschreiben oder einen ähnlichen Unsinn anstellen.

Was aber, wenn es die Datei "DAS.COM" gar nicht gibt? Dann ist der Interpretier noch lange nicht am Ende. Er versucht es einfach mit einem neuen Dateityp, nämlich ".SUB". Falls diese Datei gefunden werden konnte, nimmt der Interpretier an, daß sie eine Reihe von Kommandos enthält, die alle auszuführen sind. Deshalb startet er das Programm "\$DO.COM", das diese Datei für die Interpretation aufbereitet:

```
$DO DAS IST ABER EIN SCHONER RECHNER!
```

Was das Programm "\$DO.COM" genau alles macht, ist unter dem entsprechenden Kapitel näher nachzulesen. In diesem Zusammenhang interessiert uns nur, daß durch die Eingabe des Wortchens "Das" nach zwei Dateien gesucht wird:

```
erstens nach DAS.COM und zweitens nach DAS.SUB.
```

```
*****
* Kommando - Interpretation *
*****
```

Wenn beide Dateien nicht vorhanden sind (was nicht gerade unwahrscheinlich ist), ist der Interpreter endlich mit seinem Latein am Ende; er gibt aus:

Programm nicht gefunden: DAS

Es kann noch anders kommen: stellen Sie sich vor, die Datei "DAS.SUB" existiert und das Programm \$DO nicht, das ja dazu nötig wäre, um die Datei "DAS.SUB" zu interpretieren. Dann kommt die Meldung:

Datei nicht ausführbar: DAS

Fassen wir zusammen: das erste Wort einer Zeile wird als Dateiname interpretiert. Falls kein Typ vorhanden ist, wird automatisch nach bestimmten Dateitypen gesucht. Eine Erweiterung dieser Weisheit ist natürlich die, daß Sie als erstes Wort auch einen vollständigen Dateinamen angeben können: Geben Sie zum Beispiel ein:

B:DAS.NIX IST ABER EIN SCHONER RECHNER!

Wird die Datei "B:DAS.NIX" ganz gezielt gesucht (und auch geladen, falls sie gefunden wurde).

### B.1.3. Die Eingabe von Kleinbuchstaben

Natürlich gibt es einige Tricks und Kniffe, die Sie bei der Kommando - Eingabe anwenden können. Sie könnten sich zum Beispiel darüber ärgern, daß Ihre Eingaben prinzipiell in Großbuchstaben umgewandelt werden. Wenn Sie nun durch irgend einem dummen Zufall eine Datei erzeugt haben, die Kleinbuchstaben enthält, können Sie diese nicht einmal löschen! Keine Sorge; selbstverständlich können Sie das. Schließen Sie einfach alles, was nicht in Großbuchstaben umgewandelt werden soll, in Anführungszeichen ein. Ein Beispiel: Sie haben die Datei "Schrott.DAT" erzeugt und wollen diese löschen. Mit dem Kommando

LOSCHE Schrott.DAT

werden Sie mit Sicherheit keinen Erfolg haben; die Datei "SCHROTT.DAT" gibt es (hoffentlich) nicht. Sie müssen den Dateinamen in Anführungszeichen setzen:

LOSCHE "Schrott.DAT"

was zum Erfolg führt. Was das "LOSCHE" genau bewirkt, wird später erklärt.

```
*****  
* Kommando - Interpretation *  
*****
```

#### B.1.4. Sonderzeichen am Zeilenanfang

Es gibt eine Reihe von Sonderzeichen, mit denen eine Kommando - Eingabe beginnen kann. Diese Zeichen müssen an erster Stelle der Zeile, ohne führende Leerzeichen, stehen. Die folgenden Sonderzeichen sind zulässig:

##### B.1.4.1. Das Semikolon

Durch ein Semikolon wird eine Kommentarzeile eingeleitet. Der Rest der Zeile wird ignoriert. Dieses Feature hat im praktischen Betrieb wenig Bedeutung. Kommentarzeilen sind jedoch innerhalb von ausführbaren Dateien, d.h. Dateien, die ihrerseits Kommandos enthalten, sinnvoll.

##### B.1.4.2. Der Doppelpunkt

Eine Zeile, die mit einem Doppelpunkt beginnt, ist eine sogenannte bedingt ausführbare Zeile. Sie wird nur interpretiert, wenn das vorangegangene Programm fehlerfrei beendet wurde. Ist hingegen das vorangegangene Programm mit einem Fehler beendet worden, beispielsweise durch einen EOS - Systemfehler, wird eine derartige Zeile nicht interpretiert, sondern als Kommentar aufgefaßt.

Der Programmfehler - Code wird durch eine interne Systemfunktion gesetzt. Viele der zur Zeit käuflichen Programme kennen diese Funktion noch nicht; es kann daher sein, daß, obwohl ein Programm durch einen (vom Programm selbst erkannten) Fehler beendet wurde, eine mit einem Doppelpunkt beginnende Zeile dennoch interpretiert wird.

##### B.1.4.3. Das Sternchen

Eine mit einem Sternchen beginnende Zeile wird direkt an den eingebauten Grafik - Interpreter weitergegeben. Somit ist es möglich, Grafik - Kommandos direkt als Kommando einzugeben und interaktiv kleine Bildchen zu zeichnen. Die möglichen Grafik - Kommandos sind in einem eigenen Kapitel beschrieben.

```
*****  
*      Suche nach Dateien      *  
*****
```

### B.1.5. Die Suche nach den Dateien

Normalerweise wird nur auf dem aktuellen Laufwerk und im aktuellen Benutzerbereich nach den Kommando - Dateien gesucht. Dies kann natürlich auf die Dauer lastig sein, da es nötig wäre, auf jedem Laufwerk und in jedem Benutzerbereich Kopien der benötigten Programme zur Verfügung zu halten. EOS hat deshalb eine Erweiterung des Such - Algorithmus eingebaut bekommen.

Sprechen wir zuerst über die Benutzerbereiche. Befinden Sie sich in einem anderen Benutzerbereich als 0, sind alle Kommandos nicht mehr auffindbar. Um diese dennoch laden zu können, können Dateien mit einem speziellen Attribut versehen werden. Was Datei - Attribute sind, haben Sie ja inzwischen erfahren. Das Attribut, was hierzu benötigt wird, ist das "SYSTEM" - Attribut. Auf Dateien, die ein derartiges Attribut besitzen, kann auch von anderen Benutzerbereichen aus zugegriffen werden. Ein weiterer Vorteil: Dateien mit dem "SYSTEM" - Attribut erscheinen nicht mehr im Inhaltsverzeichnis, was erheblich zur Übersichtlichkeit desselben beitragen kann. Attribute werden mit dem Dienstprogramm => \$SET gesetzt und gelöscht.

Desweiteren wurde die Kommandosuche über mehrere Laufwerke ausgedehnt. Sie können mit Hilfe des Programms => \$CONFIG bis zu 4 Laufwerke definieren, auf denen nach der Kommandodatei gesucht werden soll. Bei Auslieferung von EOS wird automatisch erst auf der RAM - Disk "P:" gesucht, dann auf dem aktuellen Laufwerk. Das kann jedoch beliebig umdefiniert und erweitert werden.

Für die Suche muß nicht unbedingt in jedem Laufwerk auch eine Diskette liegen. Wenn mal ein zu durchsuchendes Laufwerk keine Diskette enthält, wird es automatisch übergangen.

```
*****
*   Eingebaute Kommandos   *
*****
```

### B.1.6. Eingebaute Kommandos

In den Kommando - Interpreter sind eine Reihe von Kommandos fest eingebaut. Einige Kommandonamen werden zwecks der Kompatibilität zu CP/M in andere Kommandonamen übersetzt. Dieses Kapitel soll Auskunft über derartige Konventionen geben.

Eine Sonderform des eingebauten Kommandos ist das Wechseln des aktuellen Laufwerks. Zu diesem Zweck geben Sie den Namen des Laufwerks, gefolgt von einem Doppelpunkt, ein:

B:

definiert beispielsweise das Laufwerk B: als das aktuelle Laufwerk. Das aktuelle Laufwerk wird in der Statuszeile an zweiter Stelle angezeigt.

Die folgenden beiden Kommandos sind fest eingebaut:

BEN - Wechsel des Benutzerbereichs  
 DRUCKER - Drucker ein- und ausschalten

Folgende Kommando werden übersetzt:

USE oder USER	- in BEN
PRI oder PRINTER	- in DRUCKER
INH, INHALT, DIR, DIRS oder DIRSYS	- in \$DIR
LOESCHE, LOSCHE, ERA oder ERASE	- in \$ERASE
LISTE, TYP oder TYPE	- in \$TYPE
UMB, UMBENENNE, REN oder RENAME	- in \$RENAME

Die obigen Dienstprogramme haben die folgende Wirkung:

INH, INHALT, DIR - Ausgabe des Inhaltsverzeichnisses

DIRS, DIRSYS - Ausgabe des Inhaltsverzeichnisses für Systemdateien, hat die selbe Wirkung wie "INHALT +SYSTEM"

LOESCHE,  
 LOESCHE, ERA, ERASE - Löschen von Dateien

UMB, UMBENENNE,  
 REN, RENAME - Umbenennen von Dateien

Die genaue Aufrufart dieser Kommandos ist bei der Beschreibung der Dienstprogramme nachzulesen. Wenn Sie beispielsweise genau wissen wollen, wie das Inhaltsverzeichnis aufrufen ist, schlagen Sie bei der Beschreibung von \$DIR nach. An dieser Stelle seien hier nur kurz die beiden eingebauten Kommandos besprochen.

```
*****
*   Eingebaute Kommandos   *
*****
```

#### B.1.6.1. BEN - Wechsel des Benutzerbereichs

Der aktuelle Benutzerbereich wird in der Statuszeile an dritter Stelle angezeigt. Er kann auch durch die einfache Eingabe von "BEN" erfragt werden. Ein Beispiel:

```
BEN
```

```
Aktueller Benutzerbereich: 0
```

Wenn Sie eine Zahl zwischen 0 und 15 an BEN anfügen, wird der Benutzerbereich auf diesen Wert umgestellt:

```
BEN 15
```

```
Aktueller Benutzerbereich: 15
```

Geben Sie irgendeinen Unsinn ein, erscheint:

```
Nur Benutzernummern zwischen 0 und 15 zulässig!
```

Wegen der Kompatibilität zu CP/M haben die Kommandos "USER" bzw. "USE" die selbe Wirkung.

#### B.1.6.2. DRUCKER - Drucker ein- und ausschalten

Sie können den Drucker parallel zum Bildschirm betreiben. Zu diesem Zweck wurde dieses Kommando eingebaut. Zum Einschalten des Druckers geben Sie ein:

```
DRUCKER EIN
Drucker ist eingeschaltet.
```

Das Ausschalten passiert mit:

```
DRUCKER AUS
Drucker ist ausgeschaltet.
```

Falls Sie sich vertippt haben sollten, erfolgt die Meldung:

```
EIN oder AUS erwartet.
```

Der Kompatibilität zum Englischen halber haben die Kommandos "PRINTER" bzw. "PRI" die selbe Wirkung. Desgleichen können Sie statt "EIN" auch "ON" und statt "AUS" auch "OFF" verwenden.

```

*****
*           Menuprogramm           *
*****

```

## B.2. Das Menuprogramm

Ein ganz anderes Prinzip der Kommandoeingabe wird mit dem Menuprogramm verfolgt. Wenn Sie dieses Programm benutzen, erscheint ein zweigeteilter Bildschirm. Im linken Teil sind eine Reihe von Wahlmöglichkeiten untereinander aufgelistet. Die rechte Hälfte des Bildschirms bleibt vorerst leer. Eine der linken Zeilen ist invers auf dem Bildschirm dargestellt.

Drucken Sie nun die Pfeil-aufwärts- oder die Pfeil-abwärts - Taste. Sie werden merken, daß Sie mit diesen beiden Tasten das inverse Feld auf und ab bewegen können. Wenn Sie mal eine Weile gar nichts tun, erscheint in der rechten Hälfte des Bildschirms ein Zusatztext, der Sie über Sinn und Zweck des gerade invers erscheinenden Menüpunkts aufklärt.

Haben Sie Interesse an einem Punkt gefunden, drucken Sie "ENTER". Diese Taste kann mehrere Effekte zur Folge haben. Im Normalfall starten Sie dadurch ein Programm. Es kann auch sein, daß Sie ein weiteres Menu aktivieren. Diese Verschachtelung von Menus kann sehr weit gehen. Um ein Menu wieder zu verlassen, drucken Sie die "HOME" - Taste. Das Menuprogramm kehrt dann sofort in das vorhergehende Menu zurück.

Wenn Sie ein Programm starten, kann es sein, daß noch weitere Angaben für den Start notwendig sein sollten. In diesem Fall erscheint in der rechten Hälfte eine Reihe von Feldern. Jedes Feld enthält eine Überschrift und eventuell bereits einen Text, den Sie stehen lassen oder abändern können. Die Eingabefelder sind mit Unterstrichen gekennzeichnet. Sie können mit Hilfe der Cursortasten alle diese Felder anspringen. Die "DEL" - Taste löscht ein Zeichen. Wenn Sie ein Feld vollständig ausgefüllt haben, drucken Sie "ENTER". Wenn Sie auf diese Weise das letzte Feld verlassen haben, wird das Programm gestartet. In diesem Moment wird der Bildschirm gelöscht und die Meldung

```
Laden von XXX.COM
```

Dies ist nur eine "protokollarische" Meldung. Sie sagt Ihnen, daß auf Grund Ihrer Angaben ein Programm gefunden werden konnte und geladen wird. Falls durch Ihre Eingabe eine Datei aktiviert wurde, die selbst Kommandos enthält, erfolgt die Meldung:

```
Laden von $DG.COM
```

Anschließend werden die Kommandos, die gerade ausgeführt werden, wie folgt angezeigt:

```
Kommando: $DIR *.COM
```

Hier wird zum Beispiel angezeigt, daß nun gerade das Kommando "\$DIR \*.COM" ausgeführt wird.

```

*****
*           Menüprogramm           *
*****

```

Wenn das Menüprogramm zusätzliche Eingabe von Ihnen fordert, Sie aber diese für den Augenblick verweigern wollen, drücken Sie einfach die "HOME" - Taste. Diese befördert Sie wieder in die Auswahl der Menüpunkte zurück.

### B.2.1. Mögliche Eingaben beim Menüprogramm

Zusammengefaßt haben Sie die folgenden Eingabemöglichkeiten:

Pfeil auf	- inverses "fenster" um eine Zeile nach oben
Pfeil ab	- inverses "fenster" um eine Zeile nach unten
ENTER	- durch "fenster" hervorgehobenen Menüpunkt anwählen
HOME	- aktuelles Menü wieder verlassen

Zusätzlich gibt es bei der Eingabe von Zusatzwerten noch die folgenden Tasten:

Pfeil links	- Cursor um eine Stelle nach links. Wenn sich der Cursor auf der ersten Stelle eines Feldes befindet, wird auf die letzte Stelle des vorangehenden Feldes gesprungen.
Pfeil rechts	- Cursor um eine Stelle nach rechts. Wenn sich der Cursor ganz rechts in einem Feld befindet, wird auf die erste Stelle des nachfolgenden Feldes gesprungen.
DEL	- das Zeichen, das sich unter dem Cursor befindet, wird gelöscht. Falls sich der Cursor am Ende einer Eingabe befindet, wird das links vom Cursor stehende Zeichen gelöscht.
HOME	- Die Eingabe von Zusatzwerten wird abgebrochen. Sie kehren in die Auswahl der Menüpunkte zurück.
ENTER	- Verlassen des aktuellen Eingabefeldes und Sprung in das nächste Feld. Befanden Sie sich mit dem Cursor im letzten Feld, wird das Programm gestartet.

```
*****
*           Menu - Dateien           *
*****
```

### B.2.2. Die Programmierung des Menuprogramms

An diesem Punkt können Sie, verehrter Anwender, getrost ein laßchen Kaffee einnehmen und zum nächsten Kapitel übergehen. Sie, lieber Programmierer, verweilen besser noch ein wenig.

Das Menuprogramm wird über Menudateien gesteuert. Diese sind normale Textdateien vom Typ ".MNU". Beim Kaltstart von EOS wird die Datei "EOS.MNU" als Hauptmenu geladen.

Eine Menudatei hat die folgende Struktur: die erste Spalte jeder Zeile enthält ein Steuerzeichen, das die Art der Zeile kennzeichnet. Von diesen Steuerzeichen gibt es eine ganze Reihe:

- H - Header. Der nachfolgende Text wird zentriert im linken oberen Feld ausgegeben. Es werden maximal 25 Zeichen übernommen.
- M - Menüpunkt. Der nachfolgende Text wird als Menüpunkt übernommen. Es werden maximal 25 Zeichen übernommen. Alle weiteren Angaben bis zum nächsten Menüpunkt beziehen sich auf diesen Menüpunkt.
- : - Hilfstext. Zu jedem Menüpunkt können bis zu 14 Zeilen Hilfstext eingegeben werden. Es werden pro Zeile maximal 50 Zeichen übernommen. Der Hilfstext erscheint automatisch, wenn etwa 3 Sekunden lang keine Benutzereingabe erfolgte.
- S - Submenu. Wenn der Menüpunkt angewählt wird, soll ein Sub - Menu gestartet werden. Der nachfolgende Text wird als Dateiname interpretiert. Fehlt der Dateityp, wird der Typ ".MNU" angenommen.
- P - Programm starten. Bei Anwahl des Menüpunktes soll das Programm gestartet werden, das als Folgetext angegeben ist. Die Suche nach dem Programm folgt dem selben Schema wie bei Kommando - Interpreter beschrieben. Der Folgetext kann auch aus einer kompletten Kommandozeile bestehen. Es werden maximal 80 Zeichen übernommen.
- E - Eingabe von Programm - Parametern. Falls die Kommandozeile, die bei der Option "P" angegeben wird, unvollständig ist, kann sie vom Benutzer ergänzt werden. Die Benutzereingaben werden in der Reihenfolge der "E" - Zeilen an die "P" - Zeile vor dem Programmstart angehängt. Der Zeilenrest der "E" - Zeile ist durch ein Komma zweigeteilt. Der erste Teil wird als Prompt ausgegeben. Der zweite Teil dient zur Vorbesetzung der Eingabe. Dieser Teil darf auch fehlen.

```
*****
*           Menü Dateien           *
*****
```

C - Confirm. Nach dem Ende des Programmlaufs erscheint in der Statuszeile die Aufforderung, die Leertaste zu drücken. Diese Option dient dazu, eventuelle Bildschirmausgaben des Programms zu erhalten, ehe das Menüprogramm seine Maske ausgibt.

; - Kommentarzeile.

Alle anderen Zeichen am Zeilenanfang bewirken das Überlesen der entsprechenden Zeile. Somit können auch Leerzeilen eingefügt werden.

Die Syntax - Checks der Menüdatei wurden bewußt so klein wie möglich gehalten, um das Programm nicht unnötig groß werden zu lassen. Sie als Programmierer dürfen sich daher bitte nicht wundern, wenn das Menüprogramm u.U. abstürzt, weil Sie sich in der Struktur der Menüdatei vertan haben!

#### B.2.2.1. Beispiel einer Menüdatei

Als Beispiel schauen wir uns eine Menüdatei an, die ein Testmenü von drei Punkten erstellt. Der erste Punkt hat den Start eines Sub - Menüs zur Folge. Der zweite Punkt startet ein Programm. Der dritte startet ebenfalls ein Programm, jedoch mit Eingabe von Zusatzparametern. Weitere Beispiele sind auf der EOS - Systemdiskette zu finden.

```
; Menü - Datei eines Testmenüs
HTest - Menü
; erster Menüpunkt: Start eines Submenüs
MSubmenü laden
; Help - Text:
: Durch Anwahl dieses Punkt starten Sie
: ein tolles Submenü namens "NEUMENU.MNU"
; Angabe des Submenüs:
SNEUMENU
; zweiter Menüpunkt: Start eines Programms
MEditieren von TESTDAT
: Damit können Sie die Datei "TESTDAT" editieren.
; Startangabe:
PEDIT TESTDAT
; dritter Menüpunkt: Start eines Programms mit Zusatzparametern
MEditieren einer Datei
: Damit können Sie eine Datei Ihrer Wahl
: editieren.
; Startangabe:
PEDIT
; Eingabe des Dateinamens mit Vorbesetzung "TESTDAT":
EBitte den Namen der Datei,TESTDAT
; Eingabe von irgendwelchen obskuren Optionen:
EBitte hier mögliche Optionen eingeben:
```

Abschnitt C  
Dienstprogramme

```
*****
*      Dienstprogramme      *
*****
```

### C. Dienstprogramme

Um DOS wirklich nutzen zu können, brauchen Sie mehr als das eigentliche Betriebssystem. Sie brauchen eine Reihe von kleinen Hilfsroutinen, die Ihnen die kleinen Arbeiten abnehmen. Zu diesen Arbeiten gehören beispielsweise das Löschen und Umbenennen von Dateien, das Initialisieren von Disketten und vieles andere mehr. Alle diese Programme fallen unter den Begriff Dienstprogramme.

Alle DOS - Dienstprogramme haben eines gemeinsam: ihr Name beginnt mit einem Dollarzeichen "\$". Aber da hat schon die Gemeinsamkeit ein Ende: die DOS - Dienstprogramme lassen sich in zwei Gruppen einordnen: die einen werden einfach durch Angabe ihres Namens aufgerufen. Diese Programme erscheinen dann mit einer hübschen Bildschirmmaske und einem Menu, aus denen dann die gewünschte Funktion ausgewählt wird. Die anderen hingegen werden mit einer kompletten Kommandozeile aufgerufen. Fehlt diese oder ist sie fehlerhaft, wird kurz auf dem Bildschirm ausgegeben, wie eine derartige Zeile aussehen sollte. Im nachfolgenden wollen wir uns diese beide Arten von Programmen mal näher ansehen.

#### C.1. Menuprogramme

Die erste Klasse von Dienstprogrammen sind die sogenannten Menuprogramme. Diese Programme werden einfach durch das Eintippen des Programmnamens aufgerufen. Sie erscheinen dann mit einem Hilfsmenu auf dem Bildschirm. Die wesentliche Eigenschaft eines Menuprogramms ist die, daß Sie alle Ihre Eingaben mit den Cursorstasten zwischen der alphanumerischen Tastatur und dem Zehnerblock erledigen können. Sie sehen, daß meist eine Zeile in dem Menu invertiert dargestellt ist. Diese Zeile wollen wir in Zukunft als Fenster bezeichnen. Sie können dieses Fenster mit Hilfe der Pfeiltasten auf jede gewünschte Zeile bewegen. Probieren Sie es ruhig mal aus!

Wenn Sie nun eine Zeile erleuchtet haben, die Ihr Interesse findet, so drücken Sie einfach die "ENTER" - Taste. Dadurch wird diese Zeile aktiviert. Meist wählt das Drücken der "ENTER" - Taste einen bestimmten Punkt aus einer Liste von Punkten aus. Sie können auf diese Weise auch ein Kommando aktivieren. Kommandos erscheinen meist ganz rechts in der Bildschirmmaske und sind groß geschrieben. Das meistverwendete Kommando ist wohl das "ENDE" - Kommando, durch das Sie das Programm wieder verlassen können.

Sie haben eine weitere Möglichkeit, das Programm zu verlassen: drücken Sie einfach CONTROL-C oder das von Ihnen mittels => \$CONFIG definierte zweite Stopzeichen. Dieses führt (fast) immer zum Abbruch des Programms.

```
*****
*      Dienstprogramme      *
*****
```

## C.2. Programme mit Kommandozeile

Die zweite Klasse von Dienstprogrammen sieht etwas anders aus: diese werden durch Eingabe einer kompletten Kommandozeile gestartet. Falls Sie das Menuprogramm SHELL benutzen, sind dies die Programme, die neben der Auswahl der Menüpunkte auch noch die Eingabe von Zusatzparametern verlangen. Eine typische Kommandozeile würde wie folgt aussehen:

```
$SET *.* +SYSTEM
```

### C.2.1. Optionen

Viele der so zu startenden Dienstprogramme sind durch zusätzliche Optionen steuerbar. Optionen sind spezielle Kommandoworte, die entweder mit einem Plus "+" oder mit einem Minus "-" beginnen. Optionen brauchen nicht angegeben werden. Falls eine Option fehlen sollte, tritt automatisch eine Vorbesetzung in Kraft.

Optionen können an jeder Stelle einer Kommandozeile erscheinen (außer vor dem Kommandonamen, also als erstes Wort). Sie brauchen nicht den vollen Namen einer Option anzugeben; der erste Buchstabe genügt völlig. Ein Beispiel: Das Kommando

```
$SET *.* +SYSTEM      hat den selben Effekt wie
$SET +S *.*
```

Was bedeutet nun das Plus- bzw. Minuszeichen? Nun, Optionen können ein- oder ausgeschaltet werden. Geben Sie ein Plus ein, wird die Option eingeschaltet. Geben Sie ein Minus ein, wird die Option ausgeschaltet. Programme haben für ihre Optionen die unterschiedlichsten Vorbesetzungen. Eine Option kann per Vorbesetzung ein- oder ausgeschaltet sein. Auf die einzelnen Möglichkeiten und Vorbesetzungen wird in den nachfolgenden Kapiteln noch näher eingegangen werden.

```
*****  
*   $CONFIG   *  
*****
```

### C.3. \$CONFIG - Konfiguration des Betriebssystems

Mit \$CONFIG können Sie die folgenden Parameter Ihres EOS - Betriebssystems fest einstellen:

- die Anzahl der Minidiskettenlaufwerke
- die Anzahl der Standarddiskettenlaufwerke
- die Suchfolge des Kommandointerpreters (siehe Abschnitt B)
- das für temporäre Dateien zu verwendende Laufwerk
- ein zusätzliche Stopzeichen (siehe Abschnitt A)
- den Tabulatorstop des Systems
- die Version des Betriebssystem - Kerns
- das Kalenderjahr, um die Uhr richtig zu stellen

Das Programm selbst gehört weder zu den Menuprogrammen, noch ist es durch eine komplexe Kommandozeile aufrufbar. Vielmehr erfolgt nach dem Aufruf

\$CONFIG

ein Frage- und Antwortspiel, das Sie durchgehen können.

\$CONFIG benötigt zum einwandfreien Funktionieren die Systemdatei "SYSVAR.SYS". Diese Datei wird immer auf der Diskette gesucht, die sich im rechten Minidiskettenlaufwerk befindet. \$CONFIG wartet so lange, bis Sie die richtige Diskette (normalerweise die Systemladediskette) eingelegt haben oder aber das Programm durch ein Stopzeichen beendet haben. Diese Datei wird von \$CONFIG geändert und, falls Sie es wünschen, zurückgeschrieben. Die Änderungen treten dann nach dem nächsten Ladevorgang des Systems (also nach den nächsten Einschalten des Rechners) in Kraft.

```
*****
*   $COPY   *
*****
```

#### C.4. \$COPY - Der Dateikopierer

So unscheinbar es auch ist, so ist dieses Programm doch mit das wichtigste Programm Ihres Betriebssystems. Es ist imstande, Ihre Dateien hin und her zu schaufeln sowie Dateien zu sichern. Desweiteren können die logischen Zeichengeräte Ihres Rechners angesprochen werden. Sie können \$COPY dazu benutzen, Ihren Rechner als Bildschirm eines anderen Rechners zu benutzen. Falls nötig, können Sie sogar mit anderen Rechnern kommunizieren!

All dieses wird von einem Programm durchgeführt, das noch nicht einmal ein Menu besitzt. Es ist im Gegenteil nur durch die Eingabe von Kommandozeilen zu aktivieren. Das ist keine Schikane; \$COPY besitzt derart viele Optionen, daß die Steuerung dieser Optionen per Menu fast unmöglich wäre. Es sind so viele Optionen, daß die Übersetzung in deutsche Optionsnamen mangels verfügbarer Buchstaben des Alphabets unterlassen wurde. Sie müssen in diesem Fall mit englischen Optionen vorliebnehmen.

##### C.4.1. Aufruf und Kommandoeingabe

Kommen wir zum Start von \$COPY. Der ist ganz einfach: Sie geben ein:

```
$COPY
```

Das Programm antwortet prompt mit:

```
EOS V3 Dateikopierer (? = Hilfe)
```

```
*
```

Das Sternchen ist die Bereitschaftsanzeige von \$COPY. Nun können Sie nach Herzenslust Ihre Kommandos eingeben. Fangen wir mit dem einfachsten an: In der ersten Zeile, die \$COPY ausgibt, wird dezent darauf hingewiesen, daß ein wenig Hilfe zur Verfügung steht. Geben Sie also mal ein Fragezeichen ein und drücken Sie die ENTER - Taste. \$COPY antwortet Ihnen prompt mit der Liste der verfügbaren Optionen:

```
*****
*   $COPY   *
*****
```

```
Kommando:  Quelle Ziel (Optionen)
-ASCII     kein Übertragungsende bei Ctrl-Z
+BACKUP    Backup - Modus
+CONFIRM   bei jeder Datei zurückfragen
+DEST n    Kopie in den Benutzerbereich n
+ERASE     Quelle nach Kopiervorgang löschen
-INTERRUPT Tastatur-Abfrage unterdrücken
-PROT      kein Protokoll
+REMOVE    Ziel vor dem Kopiervorgang löschen
+SOURCE n  Kopie aus Benutzerbereich n
-TEMP      Dateitypen .$$$ und .BAK nicht mitkopieren
+VERIFY    prüflesen
+ZERO      höchstwertiges Bit ausblenden
```

Das ist furwahr eine lange Liste. Uns soll zuerst die oberste Zeile interessieren:

```
Kommando:  Quelle Ziel (Optionen)
```

Das ist in etwa das Format einer Kommandozeile. Eine derartige Zeile muß mindestens zwei Dateibezeichnungen enthalten: die erste bezeichnet dabei die Quelle, die zweite das Ziel. Ein Beispiel: wollen Sie die Datei "\$COPY.COM" von A: nach B: kopieren, so sieht die Kommandozeile wie folgt aus:

```
* A:$COPY.COM B:$COPY.COM
```

Wenn die Datei auf dem Ziellaufwerk den selben Namen haben soll wie auf dem Quelllaufwerk, so brauchen Sie den zweiten Namen nicht voll auszusprechen. Es reicht, wenn Sie das Laufwerk angeben:

```
* A:$COPY.COM B:
```

Falls Sie sich sowieso auf dem Laufwerk A: befinden (was bedeutet, daß das Laufwerk A: in der Statuszeile des Bildschirms angezeigt wird), können Sie die Laufwerksangabe auch weglassen:

```
* $COPY.COM B:
```

Kürzer geht es allerdings nicht. Sie konnten auf den Gedanken verfallen, daß, wenn Sie sich auf dem Laufwerk B: befinden sollten, obige Kommandozeile auch so aussehen könnte:

```
* A:$COPY.COM
```

Das klappt allerdings nicht, da immer zwei Dateibezeichnungen angegeben werden müssen.

```
*****
*   $COPY   *
*****
```

Gehen wir einen Schritt weiter. Sie wollen sicherlich auch mal mehrere Dateien auf einmal kopieren. Kein Problem. Sie sind frei, sowohl bei der Quelle als auch beim Ziel mehrdeutige Dateinamen anzugeben. Wollen Sie beispielsweise alle ".COM" - Dateien von A: nach B: kopieren, geben Sie ein:

```
* A:*.COM B:
```

Ganz einfach sieht die Kopie eines kompletten Laufwerks aus:

```
* A: B:
```

Da Sie beim Ziel auch einen mehrdeutigen Dateinamen angeben können, ergibt sich ein kurioser Nebeneffekt: die Dateien werden einzeln umbenannt. Wenn Sie sich zum Beispiel entschließen sollten, Ihre ".COM" - Dateien auf dem Ziellaufwerk mit dem Typ ".OLD" zu versehen (aus welchen Gründen auch immer), sieht die Kommandozeile so aus:

```
* A:*.COM B:*.OLD
```

Selbstverständlich können Sie auch einzelne Dateien so kopieren, daß sie auf dem Ziellaufwerk einen neuen Namen bekommen, wie zum Beispiel:

```
* A:$COPY.COM B:$COPY.OLD
```

Wieder mal folgt jetzt eine Einschränkung: Wenn Sie als Quelle einen mehrdeutigen Dateinamen und als Ziel einen eindeutigen Dateinamen angeben, meint \$COPY. Das würde nämlich bedeuten, daß alle Dateien auf dem Ziellaufwerk den selben Namen bekommen, was wiederum bedeutet, daß nur die zuletzt kopierte Datei nachher in einsamer Größe dasteht.

Fassen wir einmal die Bedingungen zusammen, die eine Kommandozeile erfüllen muß, damit \$COPY sie akzeptiert:

- immer zwei Datei-Bezeichnungen angeben
- Quelle und/oder Ziel können mehrdeutig sein
- wenn alle Dateien eines Laufwerks gemeint sind, genügt die Angabe des Laufwerks
- ein mehrdeutiger Dateiname als Quelle und ein eindeutiger Name als Ziel ist Unsinn und daher unzulässig

```
*****
*   $COPY   *
*****
```

#### C.4.2. Direktaufruf

Wenn Sie nur einen Kopiervorgang machen wollen, dann können Sie Ihre Kommandozeile direkt an den Programmnamen anfügen. Wollen Sie beispielsweise alle ".COM" - Dateien von A: nach B: kopieren, können Sie einfach eingeben:

```
$COPY A:*.COM B:
```

\$COPY antwortet dann mit:

```
EOS V3 Dateikopierer
```

```
A:DATEI1 .COM ...
```

Die drei Pünktchen deuten das Kopierprotokoll an, auf das wir gleich eingehen wollen.

#### C.4.3. Der Kopiervorgang und das Fehlerverhalten

Lassen Sie uns bei dem Beispiel bleiben, daß Sie alle ".COM" - Dateien von A: nach B: kopieren wollen. \$COPY wird Ihnen ein Kopierprotokoll ausgeben, das in etwa so aussehen konnte:

```
$COPY A:*.COM B:
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
A:DATEI1 .COM 10 KB kopiert nach B:DATEI1 .COM
A:DATEI2 .COM 8 KB kopiert nach B:DATEI2 .COM
A:DATEI3 .COM 19 KB kopiert nach B:DATEI3 .COM
A:DATEI4 .COM 2 KB kopiert nach B:DATEI4 .COM
```

Wie Sie sehen, gibt Ihnen \$COPY bei jeder Datei Auskunft über

- die gerade bearbeitete Datei
- die Größe dieser Datei
- der Name der Zieldatei

```
*****
*   $COPY   *
*****
```

Gerade der letzte Punkt ist beim gleichzeitigen Umbenennen von Dateien von Vorteil, gibt er Ihnen doch Auskunft über die derart entstandene Zieldatei. Wenn Sie mit Umbenennen kopieren, sehen Sie den Effekt:

```
$COPY A:*.COM B:$*.OLD
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
A:DATEI1 .COM 10 KB kopiert nach B:$ATEI1 .OLD
A:DATEI2 .COM 8 KB kopiert nach B:$ATEI2 .OLD
A:DATEI3 .COM 19 KB kopiert nach B:$ATEI3 .OLD
A:DATEI4 .COM 2 KB kopiert nach B:$ATEI4 .OLD
```

Wenn keine Datei gefunden werden konnte, antwortet \$COPY mit

```
Keine Datei gefunden.
```

Kommen wir zu einem unangenehmen Kapitel: den Fehlern. Es gibt mannigfaltige Möglichkeiten, Fehler zu machen. Ein Fehler beim Kopieren wird wie folgt angezeigt:

```
$COPY A:*.COM B:$*.OLD
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
A:DATEI1 .COM 10 KB kopiert nach B:$ATEI1 .OLD
A:DATEI2 .COM 8 KB kopiert nach B:$ATEI2 .OLD
A:DATEI3 .COM 19 KB Schreib/Lesefehler.
```

Ein Fehler während eines Kopiervorgangs wird also hinter dem Namen der Quelldatei angezeigt. Das muß nicht bedeuten, daß der Fehler im Zusammenhang mit der Quelldatei aufgetreten ist. Es kann ebenso die Zieldatei betroffen sein. Ein Fehler führt zumeist zum Abbruch des Kopiervorgangs.

```
*****
*   $COPY   *
*****
```

#### C.4.4. Verhalten bei Platzmangel

Es kommt oft vor, als einem lieb ist, daß die Diskette beim Kopieren plötzlich voll ist. Stellen Sie sich nur vor, daß Sie eine Hard Disk mit 5 Megabytes auf Disketten zu je 390 KiloBytes sichern wollen. Schon haben Sie den Salat. \$COPY versucht, in einem derartigen Fall besonders behutsam vorzugehen. Ehe \$COPY eine Datei kopiert, wird erst nachgesehen, ob diese Datei überhaupt noch Platz auf der Zielplatte hat. Ist dies nicht der Fall, wird nachgesehen, ob eine Datei gleichen Namens bereits existiert, und ob genug Platz wäre, wenn diese Datei gelöscht wurde. Ist dies der Fall, wird bei Ihnen nachgefragt, ob es so genehm wäre:

```
A:DATEI3 .COM 19 KB Platte voll. Ziel löschen? (J/N)
```

Geben Sie jetzt ein "J" ein, erfolgt die Meldung:

```
B:DATEI3 .COM          gelöscht.
A:DATEI3 .COM 19 KB kopiert nach B:DATEI3.COM
```

Ist hingegen wirklich kein Platz mehr auf der Platte, erfolgt die Frage

```
A:DATEI3 .COM 19 KB Platte voll. Diskettenwechsel? (J/N)
```

Sie haben jetzt die Möglichkeit, eine neue Diskette einzulegen und weiterzumachen. Falls Sie auf einer der beiden Fragen mit "N" antworten, wird der Kopiervorgang abgebrochen. Es wird ebenfalls abgebrochen, wenn sich die Frage nach dem Diskettenwechsel erubrigt, da es sich bei der vollen Platte um die Hard Disk oder um die RAM - Floppy handelt. In diesem Fall erfolgt lapidar die Fehlermeldung

```
A:DATEI3 .COM 19 KB Platte voll.
```

#### C.4.5 Optionen

Nun wird es interessant, denn jetzt kommen wir zu den Optionen von \$COPY. Wir wollen diese der besseren Übersichtlichkeit halber in mehrere Themenkreise zusammenfassen. Wie Optionen eingegeben werden, wissen Sie ja bereits, aber eine kleine Auffrischung kann nie schaden: Eine Option fängt immer mit "+" oder "-" an. Sie kann an jeder beliebigen Stelle der Zeile stehen. Es genügt die Angabe des ersten Buchstabens der Option. Die meisten Optionen können frei miteinander kombiniert werden. Falls eine Option unsinnig ist, wird sie einfach ignoriert.

```
*****
*   $COPY   *
*****
```

#### C.4.5.1. Kopieren mit Rückfrage

Wenn Sie sich nicht sicher sind, welche Dateien Sie kopieren wollen, geben Sie die Option "+CONFIRM" ein. Dann wird bei jeder Datei zurückgefragt:

```
$COPY A:*.COM B: +C
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
A:DATEI1 .COM 10 KB Kopieren? (J/N) J
A:DATEI1 .COM 10 KB kopiert nach B:DATEI1 .COM
A:DATEI2 .COM 8 KB Kopieren? (J/N) J
A:DATEI2 .COM 8 KB kopiert nach B:DATEI2 .COM
A:DATEI3 .COM 19 KB Kopieren? (J/N) N
A:DATEI4 .COM 2 KB Kopieren? (J/N) J
A:DATEI4 .COM 2 KB kopiert nach B:DATEI4 .COM
```

#### C.4.5.2. Sparsames Kopieren

Normalerweise kopiert \$COPY mit größter Rücksichtnahme. Es wird zuerst eine temporäre Datei auf dem Ziellaufwerk eingerichtet. Ist diese temporäre Datei glücklich gefüllt worden, wird erst dann die alte Zieldatei gelöscht. Das kann zur Folge haben, daß der Platz auf der Zieldiskette knapp wird, da eine Datei unter Umständen zeitweilig in zweifacher Ausfertigung vorhanden ist. Wenn Sie das umgehen wollen, geben Sie die Option "+REMOVE" ein. In diesem Fall wird erst die alte Zieldatei gelöscht und dann kopiert:

```
$COPY A:*.COM B: +R
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
B:DATEI1 .COM 10 KB gelöscht.
A:DATEI1 .COM 10 KB kopiert nach B:DATEI1 .COM
B:DATEI2 .COM 8 KB gelöscht.
A:DATEI2 .COM 8 KB kopiert nach B:DATEI2 .COM
B:DATEI3 .COM 19 KB gelöscht.
A:DATEI3 .COM 19 KB kopiert nach B:DATEI3 .COM
B:DATEI4 .COM 2 KB gelöscht.
A:DATEI4 .COM 2 KB kopiert nach B:DATEI4 .COM
```

```
*****
*   $COPY   *
*****
```

#### C.4.5.3. Überspringen von temporären Dateien

Normalerweise wird alles kopiert, was vorgefunden wird, also auch temporäre Dateien und alte Dateiversionen. Diese Dateien mit den Typen "\$\$\$" und ".BAK" können mit der Option "-TEMP" übersprungen werden:

```
$COPY A:*.COM B: -T
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
B:DATEI1 .COM 10 KB gelöscht.
A:DATEI1 .COM 10 KB kopiert nach B:DATEI1 .COM
B:DATEI2 .$$$ 8 KB temporäre Datei, nicht kopiert.
B:DATEI3 .BAK 19 KB temporäre Datei, nicht kopiert.
A:DATEI4 .COM 2 KB kopiert nach B:DATEI4 .COM
```

#### C.4.5.4. Quellen nach dem Kopiervorgang löschen

Wenn Sie die Dateien, die Sie kopiert haben, zwecks Aufräumungsarbeiten anschließend löschen wollen, geben Sie die Option "+ERASE" ein:

```
$COPY A:*.COM B: +R
```

```
EOS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
A:DATEI1 .COM 10 KB kopiert nach B:DATEI1 .COM
A:DATEI1 .COM 10 KB gelöscht.
A:DATEI2 .COM 8 KB kopiert nach B:DATEI2 .COM
A:DATEI2 .COM 8 KB gelöscht.
A:DATEI3 .COM 19 KB kopiert nach B:DATEI3 .COM
A:DATEI3 .COM 19 KB gelöscht.
A:DATEI4 .COM 2 KB kopiert nach B:DATEI4 .COM
A:DATEI1 .COM 2 KB gelöscht.
```

#### C.4.5.5. Kopieren mit Prüfllesen

Für die Leute, die großen Wert auf Datensicherheit legen, gibt es die Option "+VERIFY". geben Sie diese an, wird jede Datei, nachdem sie geschrieben wurde, noch einmal prüfgelesen.

```
*****
*   $COPY   *
*****
```

#### C.4.5.6. Löschen des höchstwertigen Bits

Einige Textverarbeitungsprogramme erzeugen Dateien, in denen teilweise das achte Bit eines Zeichens gesetzt ist. Diese Dateien können unter Umständen nicht von anderen Programmen verarbeitet werden. Geben Sie beim Kopieren einer derartigen Datei die Option "+ZERO" an, wird das achte Bits jedes Zeichens automatisch gelöscht.

#### C.4.5.7. Protokoll ausschalten

Diese Option ist für die etwas ruhigeren Menschen, denen \$COPY zu viel erzählt. Geben Sie die Option "-PROT" an, erfolgt keine Protokollierung des Kopiervorgangs mehr. Im Fehlerfall wird die aktuelle Datei wie oben beschrieben zusammen mit der Fehlermeldung angezeigt.

#### C.4.5.8. Datensicherung

Diese Option ist eine sehr interessante Option. Sie ermöglicht es Ihnen nämlich, gezielt diejenigen Dateien zu sichern, die seit dem letzten Sichern geändert worden sind. Dazu bedient sich \$COPY des BACKUP - Attributs einer Datei (siehe auch => \$SET). Findet \$COPY eine Datei, wo dieses Attribut fehlt, wird diese Datei gesichert und anschließend mit dem BACKUP - Attribut versehen. Wird diese Datei später einmal verändert, löscht EDS automatisch das BACKUP - Attribut, so daß die Datei bei der nächsten Datensicherung mit erfaßt wird. Geben Sie nun die Option "+BACKUP" ein, wird auf Datensicherung umgeschaltet:

```
$COPY A:*.COM B: +B
```

```
EDS V3 Dateikopierer
```

```
4 Dateien gefunden.
```

```
A:DATEI1 .COM    10 KB kopiert nach B:DATEI1 .COM
B:DATEI2 .COM     8 KB bereits gesichert, nicht kopiert.
B:DATEI3 .COM    19 KB bereits gesichert, nicht kopiert.
A:DATEI4 .COM     2 KB kopiert nach B:DATEI4 .COM
```

Nach der Kopie haben alle vier Dateien das BACKUP - Attribut gesetzt.

```
*****
*   $COPY   *
*****
```

#### C.4.5.9. Kopieren zwischen Benutzerbereichen

Wenn Sie mit Benutzerbereichen arbeiten wollen, müssen Sie auch in der Lage sein, Dateien zwischen verschiedenen Benutzerbereichen hin- und herzukopieren. Dazu gibt es zwei Optionen: "+SOURCE" und "+DEST". Beide Optionen werden von einer Zahl gefolgt, von der Option durch ein Leerzeichen getrennt, die den Benutzerbereich angibt. "+SOURCE" gibt den Benutzerbereich der Quelle an, und "+DEST" den Benutzerbereich des Ziels. Wollen Sie alle Ihre ".COM" - Dateien beispielsweise vom Benutzerbereich 9 in den Benutzerbereich 12 verlagern, geben Sie das Kommando:

```
$COPY A:*.COM A: +SOURCE 9 +DEST 12
```

#### C.4.6. Kopieren über Geräte

Neben den üblichen Dateien und Laufwerken können Sie mit \$COPY auch die logischen Ein/Ausgabegeräte ansprechen. Diese Geräte haben einen festen Namen und sind bei \$COPY dem Pseudo - "Laufwerk" X: zugeordnet. Es gibt die folgenden Geräte:

```
X:CONIN   - die Tastatur
X:CONOUT  - der Bildschirm
X:LSTOUT  - der Drucker
X:AUXIN   - der Eingabekanal des Zusatzgerats
X:AUXOUT  - der Ausgabekanal des Zusatzgerats
X:EOF     - sendet ein Ctrl-Z als Dateiendezeichen
```

Diese Geräte können an Stelle eines Dateinamens angegeben werden, mit der Einschränkung, daß der zweite Dateiname eindeutig sein muß. Sie können also

- von einem Eingabegerät in eine Datei kopieren
- aus einer Datei auf ein Ausgabegerät kopieren
- von einem Eingabegerät auf ein Ausgabegerät kopieren.

Alle Übertragungen werden mit einem Ctrl-Z als Dateiendezeichen abgeschlossen. Falls es unbedingt gewünscht ist, kann die Ausgabe über ein Ctrl-Z hinaus weitergeführt werden. Zu diesem Zweck gibt es die Option "-ASCII". geben Sie diese Option an, wird beim Übertragen eines Ctrl-Z nicht aufgehört, was für die Übertragung von Programmdateien sehr nützlich ist.

```
*****
*   $COPY   *
*****
```

Ein paar Beispiele: Wenn Sie die Datei "TEXT.DAT" direkt auf dem Drucker ausgeben wollen, geben Sie ein:

```
$COPY TEXT.DAT X:LSTOUT
```

Wenn Sie die Lust verspüren, mal per Hand eine Datei zu erstellen, geben Sie ein:

```
$COPY X:CONIN DATEI
```

Jetzt gehen alle Tastatureingaben direkt in die Datei "DATEI", und zwar solange, bis Sie ein Ctrl-Z eingeben.

#### C.4.6.1. Datenkommunikation

Dies ist ein sehr wichtiges Kapitel. Mit \$COPY haben Sie die Möglichkeit, Dateien über eine längere Leitung zu übertragen. Schalten Sie beide Rechner über ein Modem über die RS232 - Schnittstelle zusammen. Ordnen Sie mit Hilfe von => \$DEVICE den Geräten AUXIN und AUXOUT die serielle Schnittstelle zu. Sie können dann mit dem Kommando

```
$COPY dateiname X:AUXOUT
```

eine Datei senden, und Ihr Gegenüber empfängt sie mit dem Kommando

```
$COPY X:AUXIN dateiname
```

Für Eingeweihte: das Einlesen über das Gerät AUXIN erfolgt mit Hilfe des XON/XOFF - Protokolls. Wenn der interne Puffer voll ist (meist nach 32 KBytes, wenn keine Funktionsmodule aktiv sind), schickt \$COPY das Steuerzeichen Ctrl-S (XOFF) über das Gerät AUXOUT. Wenn der Puffer geschrieben ist und \$COPY bereit ist, neue Daten zu empfangen, schickt \$COPY das Zeichen Ctrl-Q (XON). Wenn der Sender seine serielle Schnittstelle auf das XON/XOFF - Protokoll konfiguriert hat, wird dadurch automatisch der Sendevorgang unterbrochen bzw. fortgesetzt.

```
*****
*   $COPY   *
*****
```

#### C.4.6.2. Das Senden des Dateiendekennzeichens

Falls die Kommunikation unterbrochen sein sollte, fehlt dem Empfänger das Dateiendekennzeichen, um den Empfangsvorgang korrekt beenden zu können. Dieses Zeichen können Sie explizit mit dem Kommando

```
$COPY X:EOF X:AUXOUT
```

senden. Natürlich können Sie auch eine Leerdatei mit dem Kommando

```
$COPY X:EOF LEERDAT
```

erzeugen - falls Sie es wünschen.

#### C.4.6.3. Direktschaltung des Bildschirms

Sie können beispielsweise Ihren Rechner als grafisches Ausgabegerät für einen anderen Rechner benutzen. Belegen Sie AUXIN mit der RS232 - Schnittstelle (=> \$DEVICE) und geben Sie ein:

```
$COPY X:AUXIN X:CONOUT -PROT -ASCII
```

Durch "-PROT" wird das Kopierprotokoll ausgeschaltet, und durch die Option "-ASCII" wird dem Rechner die Möglichkeit genommen, \$COPY durch ein Ctrl-Z abzubrechen.

#### C.4.7. Das Unterbrechen der Kopie

Sie werden sich gelegentlich auch mal vertippen und aus diesem Grunde den Kopiervorgang abbrechen wollen. Das können Sie jederzeit, indem Sie ein Stopzeichen eingeben. Der Kopiervorgang wird sofort mit (beispielsweise) der Meldung

```
A:DATEI1 .COM 10 KB *** abgebrochen ***
```

abgebrochen.

Die Geräte AUXOUT und AUXIN können ebenfalls durch Eingabe eines Stopzeichens unterbrochen werden, was besonders wichtig ist, wenn die Kommunikation zusammenbrechen sollte. Falls Sie ein Stopzeichen eingeben sollten, wird automatisch ein Ctrl-Z gesendet bzw. in die Empfangsdatei eingetragen.

```
*****
*   $COPY   *
*****
```

Das führt uns zur letzten Option von \$COPY: die Abfrage der Tastatur kostet zwangsläufig Zeit. Wenn Sie sicher sind, daß die Kommunikation einwandfrei läuft, können Sie unter Umständen eine höhere Übertragungsgeschwindigkeit erreichen, indem Sie die Möglichkeit der Unterbrechung durch die Option "-INTERRUPT" ausschalten. Diese Option gilt auch für die Kopie von Dateien. Der Kopiervorgang kann dann nicht mehr unterbrochen werden.

#### C.4.8. Fehlermeldungen

+SOURCE: Ungültige Benutzernummer (0-15)  
+DEST: Ungültige Benutzernummer (0-15)

Sie haben bei den o.a. Optionen eine falsche Benutzernummer eingegeben oder sie schlicht vergessen.

Kein Ziel angegeben.

Sie haben nur eine Dateibezeichnung eingegeben.

Fehler beim Prüfllesen.

Es trat ein Vergleichsfehler beim Prüfllesen der Datei auf. Tritt nur auf, wenn Sie die Option "+VERIFY" angegeben haben.

Inhaltsverzeichnis voll.

Es kann keine neue Datei eingerichtet werden, da keine Namen mehr ins Inhaltsverzeichnis passen. Löschen Sie einige Dateien oder entfernen Sie mittels => \$INITDIR die Zeitmarken, falls Sie Zeiteinträge definiert haben.

\*\*\* abgebrochen \*\*\*

Der Kopiervorgang wurde abgebrochen, da Sie ein Stopzeichen eingegeben haben.

Name nicht eindeutig.

Hier wird ein eindeutiger Dateiname gefordert.

```
*****
*   $COPY   *
*****
```

Unbekanntes Gerät.

Sie haben mit der Laufwerksbezeichnung "X:" einen ungültigen Geratenamen verwendet.

Quelle muß ein Eingabegerät sein.

Sie haben als Quelle ein Ausgabegerät angegeben.

Ziel muß ein Ausgabegerät sein.

Sie haben als Ziel ein Eingabegerät angegeben.

Mehrere Dateien als Quelle und eine Datei als Ziel nicht zulässig.

Sie haben als Quelle einen mehrdeutigen Dateinamen und als Ziel einen eindeutigen Dateinamen angegeben.

Quelle und Ziel sind gleich, ignoriert.

Im Lauf eines Kopiervorgangs kann es vorkommen, daß eine Datei auf sich selbst kopiert werden soll. \$COPY ignoriert diese Datei einfach.

Schreib/Lesefehler.

Es trat ein physikalischer Schreib/Lesefehler auf.

Laufwerk ist schreibgeschützt.

Das Ziellaufwerk ist schreibgeschützt. Entfernen Sie den Schreibschutz von Ihrer Diskette und versuchen Sie es noch einmal.

Datei ist schreibgeschützt.

Sie haben mit Hilfe von => \$SET das R/O - Attribut der Zieldatei gesetzt. Benutzen Sie das gleiche Programm, um das Attribut wieder zu entfernen.

Laufwerk ungültig.

Sie haben ein nicht existentes Laufwerk angegeben.

Datei existiert bereits.

Ein seltener Fehler, der auf interne Verwirrungen von \$COPY zurückzuführen ist. Löschen Sie die Zieldatei per Hand und versuchen Sie es noch einmal.

```
*****
*   $COPYDSK   *
*****
```

### C.5. \$COPYDSK - Kopie von Disketten

Dieses Programm dient dazu, eine schnelle Kopie von Disketten herzustellen. Die Disketten werden physikalisch, Spur für Spur, kopiert. Das Programm gehört zur Klasse der menugesteuerten Programme und ist daher über das einfache Kommando

#### \$COPYDSK

aufzurufen. Die Bildschirmmaske hat drei Spalten: die linke Spalte enthält eine Liste der verfügbaren Laufwerke. Aus dieser Liste ist das Laufwerk auszuwählen, in dem die zu kopierende Diskette liegt. Die mittlere Spalte enthält die selbe Liste. Aus dieser Liste ist das Laufwerk auszuwählen, in dem die Leerdiskette liegt. Die rechte Spalte enthält die verfügbaren Kommandos.

Grundsätzlich ist nur die Kopie von zwei Disketten gleichen Formats möglich. Deshalb ist auch die Kopie beispielweise von einer Minidiskette auf eine Standarddiskette von vorneherein nicht anwählbar. Entdeckt das Programm beim Start der Kopie, daß die Zieldiskette ein anderes Format hat als die Quelldiskette, wird der Kopiervorgang ebenfalls abgebrochen.

Nach dem Start wird eine Art Meßplatte auf dem Bildschirm ausgegeben, bestehend aus Punkten, wobei jeder Punkt für eine zu kopierende Spur steht. Es werden zwei Meßplatten ausgegeben, je eine für die Quell- sowie für die Zieldiskette. Je nach Fortschritt der Kopie werden diese Punkte durch Buchstaben ersetzt:

- X - Systemspur. Diese Spuren enthalten unter Umständen Teile des Betriebssystems und werden nicht mit kopiert.
- L - Lesen. Nach dem Lesen jeder Spur erscheint in der oberen Meßplatte dieser Buchstabe.
- S - Schreiben. Nach dem Schreiben jeder Spur erscheint auf der unteren Meßplatte dieser Buchstabe.

```
*****  
*   $DATE   *  
*****
```

### C.6. \$DATE - Setzen des Tagesdatums

Diese kleine Hilfsroutine setzt Ihnen das aktuelle Tagesdatum neu. Diese Routine werden Sie höchst selten benötigen, da Ihr Rechner eine batteriegepufferte Uhr besitzt, die nur einmal neu gestellt werden muß. Sie rufen \$DATE auf mit:

```
$DATE tt.mm.jj hh:mm:ss
```

wobei "tt.mm.jj" dem Datum und "hh:mm:ss" der Uhrzeit entspricht. Beide Daten müssen in der oben angegebenen Form eingegeben werden. Das Datum hat demnach die Form "Tag, Monat, Jahr", durch Punkte getrennt und die Uhrzeit die Form "Stunde, Minute, Sekunde", durch Doppelpunkte getrennt. Fehlt eine dieser Angaben oder ist sie fehlerhaft, wird sie interaktiv neu angefordert. Sie können das Datum also auch nur durch die Eingabe von

```
$DATE
```

setzen.

Wenn das Datum korrekt eingegeben ist, wird eine beliebige Tastatureingabe angefordert, ehe die Uhr neu gesetzt wird. Somit ist es möglich, die Uhr auf die Sekunde genau zu stellen.

Ein Beispiel: Setzen des 30. Juni 1983, 12 Uhr 15:

```
$DATE 30.06.83 12:15:00
```

Oder das selbe einmal interaktiv mit einem Tippfehler (EOS - Ausgaben in Fettdruck):

```
$DATE
```

```
Bitte das Datum in der Form tt.mm.jj : 30.06.83
```

```
Bitte die Uhrzeit in der Form hh:mm:ss : 12.15.00
```

```
Bitte die Uhrzeit in der Form hh:mm:ss : 12:15:00
```

Eine beliebige Eingabe stellt die Uhr: <beliebige Taste>

```
*****
* $DEVICE *
*****
```

### C.7. \$DEVICE - Definieren der externen Geräte

Dieses Programm dient zur Konfiguration von Bildschirm, Tastatur und Druckern. Es bietet für verschiedene Schnittstellen verschiedene Wahlmöglichkeiten und weist den fünf logischen Geräten, die unter EOS definiert sind, entsprechende physikalische Geräte zu.

Ihr MZ-3500 enthält eine Reihe von physikalischen Geräten, die beim Programmstart von \$DEVICE in der linken Spalte angezeigt werden. Diese Geräte sind:

SCREEN	- der Bildschirm
KEYBD	- die Tastatur
RS232	- die serielle Schnittstelle
CENTRO	- die parallele Centronics - Schnittstelle
CE332P	- die Parallelschnittstelle für den Drucker CE-332P
GRAPH	- der Graphik - Interpreter
DUMMY	- ein Leergerät, auf das unerwünschte Ausgaben umgelegt werden können. Bei Eingaben liefert DUMMY ein Ctrl-Z zurück.

In der zweiten Spalte ist die Art der Geräte aufgeführt. Diese sind:

ein	- Eingabegerät
aus	- Ausgabegerät
e/a	- Ein/Ausgabegerät

Die dritte Spalte ist bereits von Ihnen anderbar. Die ersten drei Zeilen sind mit den Cursorstasten anwählbar. Wenn Sie eine dieser Zeilen anwählen und ENTER drücken, werden Sie sehen, daß der Text sich laufend verändert. Jedesmal, wenn Sie ENTER drücken, wird ein neuer Text angezeigt, der jeweils eine neue Option für das entsprechende Gerät anbietet. Die möglichen Optionen im Einzelnen:

**SCREEN:** Hier können Sie zwischen dem deutschen und dem englischen Zeichensatz wählen (falls Sie den EOS - Zeichengenerator besitzen). Diese Umschaltung legt fest, wie der Bildschirm beim Kaltstart aussehen soll.

**KEYBD:** Wählen Sie zwischen Kleinbuchstaben und Großbuchstaben. Im Normalfall schreiben Sie in Kleinbuchstaben und drücken die SHIFT - Taste, wenn Sie Großbuchstaben haben wollen. Das können Sie mit dieser Option umkehren.

**RS232:** Hier wählen Sie zwischen verschiedenen Übertragungsgeschwindigkeiten und -protokollen. Eingebaut sind die Protokolle CTS und XON/XOFF.

```
*****  
*   $DEVICE   *  
*****
```

Die vierte Spalte dient zur Zuweisung von physikalischen zu logischen Geräten. Unter EOS sind fünf logische Geräte definiert:

```
CONIN:    - Tastatur  
CONOUT:   - Bildschirm  
LSTOUT:   - Drucker  
AUXIN:    - Zusatzgerät, Eingabe  
AUXOUT:   - Zusatzgerät, Ausgabe
```

Das Zusatzgerät soll Sie in diesem Zusammenhang nicht interessieren. Sie ordnen einem Gerät (z. B. dem Drucker) ein physikalisches Gerät zu, indem Sie das inverse "Fenster" mit Hilfe der Pfeiltasten auf das jeweilige Feld bewegen und dann so lange "ENTER" drucken, bis die richtige Zuordnung erscheint.

Zu guter Letzt können Sie die derartig erfolgte Zuordnung der Geräte in das Betriebssystem übertragen. Dazu können Sie in der rechten Spalte zwischen drei Kommandos wählen:

SETZEN überträgt die Zuordnung in das Betriebssystem. Die Zuordnung gilt bis zum nächsten System - Kaltstart.

SCHREIBEN überträgt die Zuordnung auf Dauer in das Betriebssystem. Zu diesem Zweck muß die System - Ladediskette in das rechte Diskettenlaufwerk eingelegt werden. \$DEVICE modifiziert dann die auf dieser Diskette befindliche Datei "SYSVAR.SYS". Somit steht Ihnen die erfolgte Konfiguration sofort nach dem Systemkaltstart zur Verfügung.

ENDE beendet das Programm, ohne die aktuelle Zuweisung zu ändern.

```
*****
*   $DIR   *
*****
```

### C.8. \$DIR - Inhaltsverzeichnis eines Laufwerks

Es gibt immer wieder Momente, in denen Sie sehen wollen, was sich so alles auf Ihren Disketten bzw. Ihrer Festplatte tummelt. Für diese Zwecke wurde das Programm \$DIR entwickelt. Es gibt Ihnen Auskunft darüber, wo sich welche Dateien befinden, wie groß diese sind und welche Attribute diesen Dateien zugeordnet sind. Auf Wunsch wird Ihnen auch noch erzählt, wann jede Datei eingerichtet wurde und wann zuletzt auf sie zugegriffen wurde.

Um das Programm zu starten, gibt es verschiedene Möglichkeiten. benutzen Sie den Kommando - Interpreter für Programmierer, geben Sie ein:

```
INH      oder
INHALT   oder
DIR      oder
DIRSYS   oder
DIRS
```

Die ersten beiden Kommandos sind an die deutsche Sprache angelehnt. Die letzten drei sind der englischen Sprache entnommen, wobei die letzten beiden Möglichkeiten wiederum Sonderformen des 'Normalkommandos' sind. Benutzen Sie den Interpreter für Anwender, wählen Sie einfach den Menüpunkt "Inhaltsverzeichnis" (oder ähnlich) an.

Was bekommen Sie nun, wenn Sie "DIR" oder etwas ähnliches eingeben? Nun, ein Inhaltsverzeichnis Ihres Laufwerks, alphabetisch geordnet und mit den Dateigrößen versehen. Das Schirmbild könnte in etwa so aussehen:

```
Gerat A:TEST .DSK          15.06.83  15:30      Zugriffe/Anderungen
Name   Typ  Blcks KBytes
TEST1  .DAT   10    2
TEST2  .DAT   200   26
2 Datei(en), 28 KBytes, 334 KBytes frei. Systemdateien vorhanden.
```

Die Kopfzeile zeigt Ihnen den Namen des Laufwerks, das Datum und die Uhrzeit sowie die Art der Zeiteinträge. Wie Sie EOS dazu bringen können, Zeiteinträge in das Inhaltsverzeichnis aufzunehmen, lesen Sie bitte beim Dienstprogramm => \$INITDIR nach. Zur Linken sehen Sie eine vierspaltige Anzeige. Die ersten beiden Spalten zeigen Ihnen den Dateinamen sowie den Dateitypen. Die dritte Spalte sagt schon etwas über die Größe der Datei: diese Zahl gibt an, wieviele Blöcke zu je 128 Bytes Ihre Datei enthält. Wenn Sie diese Zahl mit 128 malnehmen, erhalten Sie die Größe Ihrer Datei in Bytes. Die vierte Zahl gibt die Anzahl KBytes an, die die Datei tatsächlich belegt.

```
*****
*   $DIR   *
*****
```

Wozu, werden Sie fragen, brauche ich die Anzahl der Blöcke, wenn ich bereits die Größe in KBytes habe? Ja, so einfach ist das nicht. Eine Datei kann im ungünstigsten Fall bis zu 16 KBytes wertvollen Speicherplatz belegen, obwohl sie nur aus einem einzigen Block besteht. Um derartige Dateien erkennen zu können, werden eben beide Zahlen ausgegeben.

Kommen wir zur letzten Zeile: Dort wird Ihnen gezeigt, wie viele Dateien gefunden wurden, wie viel Platz diese Dateien belegen und wie viel Platz noch insgesamt auf dem Laufwerk zur Verfügung steht. Desweiteren wird Ihnen gesagt, daß eigentlich noch mehr Dateien anzuzeigen gewesen wären, daß diese Dateien aber durch das Setzen des "SYSTEM" - Attributs (siehe => \$SET) versteckt wurden. Durch Eingabe einer Option können Sie auch diese Dateien sichtbar machen.

Falls Sie das komplette Inhaltsverzeichnis Ihres Laufwerks nicht interessiert, können Sie die Anzahl der angezeigten Dateien einschränken, indem Sie angeben, was genau angezeigt werden soll. Dieses tun Sie im Allgemeinen, indem Sie noch einen mehrdeutigen Dateinamen angeben. Dadurch werden nur diejenigen Dateien angezeigt, die auch diesem Namen entsprechen. Ein Beispiel:

```
DIR *.COM
```

zeigt Ihnen alle Dateien des Typs "COM".

```
DIR *.*
```

zeigt Ihnen alle Dateien an, hat also den selben Effekt wie

```
. DIR
```

#### C.8.1. \$DIR - Optionen

Das soeben gezeigte Anzeigeformat hat den Nachteil, daß das Inhaltsverzeichnis seitenweise angezeigt wird und der Bildschirm jedesmal zwischendurch gelöscht wird. Außerdem wollen Sie ja sicherlich auch mal die Zeitmarken sehen. Für diese und andere Wünsche sind Optionen eingerichtet worden. Was Optionen sind, haben Sie ja bereits erfahren.

Die nachfolgenden Optionen haben allesamt eine englische Entsprechung, damit auch Benutzer, die an anderssprachige Versionen von EOS gewohnt sind, ohne Umlernen mit der deutschen Version umgehen können:

ALLE	entspricht dem englischen	ALL
BEN	entspricht dem englischen	USERS
DRUCKER	entspricht dem englischen	LIST
ZEIT	entspricht dem englischen	TIME

```
*****
*       $DIR       *
*****
```

### C.8.1.1. SYSTEM - Anzeige versteckter Dateien

Wenn Sie auch Dateien sehen wollen, die mit Hilfe von => \$SET versteckt wurden (die also das "SYSTEM" - Attribut gesetzt haben), geben Sie die Option "+SYSTEM" an. Dann bekämen Sie beispielsweise als Ausgabe:

```
Gerät A:TEST   .DSK           15.06.83  15:30      Zugriffe/Änderungen
Name   Typ  Blcks KBytes
SYSDAT1 .COM   23     4
SYSDAT2 .COM    8     2
2 Datei(en), 6 KBytes, 334 KBytes frei. Normaldateien vorhanden.
```

Wie Sie sehen, bekommen Sie ein ganz anderes Inhaltsverzeichnis als weiter oben gezeigt, und statt "Systemdateien vorhanden" wird nun "Normaldateien vorhanden" angezeigt. Diese beiden Dateien sind also im Normalfall versteckt.

Falls Sie den SHELL - Kommandointerpreter benutzen, hat die Eingabe von

```
DIRS      oder      DIRSYS
```

den selben Effekt wie die Eingabe

```
DIR +SYSTEM
```

### C.8.1.2. ALLE - Alle Dateien anzeigen

Nun haben wir gesehen, wie Sie entweder die normalen Dateien oder aber die versteckten Dateien anzeigen lassen können. Wenn Sie beide Arten von Dateien gleichzeitig sehen wollen, geben Sie die Option "+ALLE" ein. Sie erhalten dann das folgende Bild:

```
Gerät A:TEST   .DSK           15.06.83  15:30      Zugriffe/Änderungen
Name   Typ  Blcks KBytes
SYSDAT1 .COM   23     4
SYSDAT2 .COM    8     2
TEST1   .DAT   10     2
TEST2   .DAT  200    26
4 Datei(en), 34 KBytes, 334 KBytes frei.
```

```
*****
*      $DIR      *
*****
```

### C.8.1.3. VORSCHUB - Seitenweise Anzeige

Diese Option ist eigentlich eine "Negativ - Option", was heißen will, daß normalerweise Seitenweise angezeigt wird. Diese Art der Anzeige kann aber sehr stören, wenn Sie beispielsweise zwei Inhaltsverzeichnisse gleichzeitig auf dem Bildschirm sehen möchten. Deshalb wurde auch eine Kurzversion des Inhaltsverzeichnisses eingeführt. Dieses Inhaltsverzeichnis wird mit der Option "-VORSCHUB" angefordert. Das Schirmbild sahe dann so aus:

```
DIR -VORSCHUB
```

```
TEST1 .DAT 2K TEST2 .DAT 26K
```

```
2 Datei(en), 28 KBytes, 340 KBytes frei. Systemdateien vorhanden.
```

Zusätzlich wird also der tatsächlich belegte Platz für jede Datei angezeigt.

### C.8.1.4. ZEIT - Anzeige der Zeitmarken

Bislang haben wir nur Ausgaben gehabt, die etwas über die Größe einer Datei ausgesagt haben. EOS bietet, wie Sie sicher bereits wissen, die Möglichkeit, den Zeitpunkt von Dateizugriffen und -änderungen zu protokollieren (näheres siehe auch => \$INITDIR). Um sich diese Zeitmarken näher anzusehen, gibt es die Option "+ZEIT". Sehen wir uns doch einmal ein Beispiel an:

```
DIR +ALLE +ZEIT
```

```
Gerat A:TEST .DSK 15.06.83 15:30 Zugriffe/Änderungen
Name Typ Blcks KBytes Letzter Zugriff Änderung Attribute
SYSDAT1 .COM 23 4 14.06.83 20:30 14.06.83 14:22 sys
SYSDAT2 .COM 8 2 12.06.83 10:23 sys
TEST1 .DAT 10 2 14.06.83 15:55 14.06.83 15:55
TEST2 .DAT 200 26 13.06.83 19:41
4 Datei(en), 34 KBytes, 334 KBytes frei.
```

Sie haben jetzt drei zusätzliche Spalten. Zwei davon sind Zeiteinträge, die Ihnen hier zum Beispiel sagen, daß die Datei "SYSDAT1.COM" am 14. Juni 1983 um 14 Uhr 22 zuletzt geändert wurde und am selben Tag um 20 Uhr 30 zuletzt gelesen wurde. Die letzte Spalte zeigt Ihnen die Attribute für jede Datei. Sie sehen, daß die ersten beiden das "SYSTEM" - Attribut besitzen und daher normalerweise unsichtbar sind.

An Stelle der Spalte "Letzter Zugriff" kann auch die Spalte "Eingerichtet" auftreten, je nach der Definition des Inhaltsverzeichnisses (=> \$INITDIR). Diese Spalte gibt Ihnen dann den Zeitpunkt an, als die Datei neu eingerichtet wurde.

```
*****
*   $DIR   *
*****
```

## C.8.1.5. BEN - Anzeige über alle Benutzerbereiche

Gehen wir noch einen Schritt weiter. Bislang haben Sie sich nur im aktuellen Benutzerbereich umgesehen. Wollen Sie sich einmal über alle Dateien in allen Benutzerbereichen informieren, geben Sie die Option "+BEN" an. Sie bekommen dann das folgende Bild:

```
DIR +ALLE +BEN *
Gerat A:TEST   .DSK           15.06.83  15:30           Zugriffe/Anderungen
Ben Name      Typ  Blcks  KBytes  Letzter Zugriff  Änderung  Attribute
0  SYSDAT1    .COM    23     4  14.06.83  20:30  14.06.83  14:22  sys
0  SYSDAT2    .COM     8     2  12.06.83  10:23           sys
11 BENBER11.  .   0     0           14.06.83  11:11
0  TEST1     .DAT    10     2  14.06.83  15:55  14.06.83  15:55
0  TEST2     .DAT   200     0           14.06.83  11:11
0  TEST1     .DAT    10     2  14.06.83  15:55  14.06.83  15:55
0  TEST2     .DAT   200    26           13.06.83  19:41
5 Datei(en), 34 KBytes, 334 KBytes frei.
```

Sie sehen, daß plötzlich eine Datei mehr im Inhaltsverzeichnis erscheint. Diese Datei BENBER11 ist leer und liegt im Benutzerbereich 11.

## C.8.1.6. DRUCKER - Ausgabe auf dem Drucker

Wenn Sie eine hubsche Druckerausgabe haben wollen, geben Sie als einzige Option "+DRUCKER" an. Sie erhalten dann eine Liste, als ob Sie die Optionen "+ALLE" und "+BEN" eingegeben hatten, auf dem Drucker, mit einem Seitenkopf und Seitenvorschub alle 64 Zeilen. Auf dem Bildschirm erscheint die Meldung:

```
Drucken einer beliebigen Taste bricht den Ausdruck ab...
```

Wenn Sie desgleichen tun, erscheint sowohl auf dem Ausdruck als auch auf dem Bildschirm die Meldung

```
Ausdruck abgebrochen.
```

und der Drucker stellt seine Tätigkeit wieder ein.

```
*****
* $DISKINF *
*****
```

## C.9. \$DISKINF - Plattenformate anzeigen

Da EOS insgesamt 12 verschiedene Diskettenformate kennt, erwies es sich als notwendig, ein kleines Programm beizufügen, das über das Format der jeweils eingelegten Disketten Auskunft gibt. Desweiteren gibt dieses Programm Auskunft über die Belegung der Platten sowie über die verbleibende freie Speicherkapazität. Der Aufruf erfolgt einfach mit

```
$DISKINF
```

Die Antwort konnte dann so aussehen:

```
Laufwerk A: Hard Disk D510, 5048 KB Kapazität, 2000 KB frei
Laufwerk B: Hard Disk D510, 5048 KB Kapazität, 5016 KB frei
Laufwerk C: EOS-Format, 390 KB Kapazität, 286 KB frei
Laufwerk D: nicht bereit
Laufwerk P: RAM-Floppy, 48 KB Kapazität, 19 KB frei
```

Sie sehen, daß immer alle definierten Laufwerke auch angezeigt werden, egal, ob sich eine Diskette dort befindet oder nicht. In unserem obigen Beispiel wäre das Laufwerk D: ohne Diskette. Falls eine Diskette vorgefunden wird, die EOS nicht lesen kann, erfolgt beispielsweise die Meldung:

```
Laufwerk D: nicht bestimmbar
```

Wenn Sie die komplette Liste nicht interessiert, können Sie auch beim Kommandoaufruf ganz gezielt das Laufwerk angeben, das Sie sehen mochten:

```
$DISKINF C:
```

```
Laufwerk C: EOS-Format, 390 KB Kapazität, 286 KB frei
```

In diesem Fall werden nicht definierte Laufwerke wie Laufwerke behandelt, die nicht bereit sind. Wenn Sie beispielsweise die Daten des nicht definierten Laufwerks 0: sehen wollen, passiert folgendes:

```
$DISKINF 0:
```

```
Laufwerk 0: nicht bereit
```

```
*****
*   $DO   *
*****
```

### C.10. \$DO - Programmierbares Abarbeiten von Kommandos

Dieses Programm ist weniger für den unbedarften Anwender als viel mehr für die Arbeitsvorbereitung und die Ablaufsteuerung gedacht. Nichtsdestoweniger ist es auch für jeden, der sich über sich ständig wiederholende Kommandofolgen ärgert, eine wertvolle Hilfe.

\$DO räumt Ihnen die Möglichkeit ein, Ihre Kommandofolgen so, wie Sie sie sonst immer eintippen würden, in einer Datei abzulegen und sie von EOS direkt ausführen zu lassen. Das geschieht in der Weise, daß Ihre Tastatur auf eine temporär eingerichtete Datei umgeschaltet wird und alle Eingaben aus dieser Datei gelesen werden. Ist die Datei abgearbeitet, wird sie gelöscht und Ihre Tastatur wieder aktiviert.

Um ein wenig mehr Komfort in die Sache hereinzubringen, wurde \$DO geschrieben. \$DO nimmt sich eine beliebige Datei vor, die allerdings vom Typ ".SUB" sein muß, bereitet diese auf, indem er verschiedene Texte ersetzt, Steuerzeichen einfügt usw., trägt die derart aufbereitete Datei in eine temporäre Datei ein und schaltet anschließend die Tastatur um.

Der Aufruf von \$DO kann auch verschachtelt sein. Sie können also aus einer \$DO - Datei heraus \$DO noch einmal aufrufen. Die Verschachtelung wird nur vom verfügbaren Speicherplatz begrenzt.

Es sei noch kurz erwähnt, daß die vorbereitenden Arbeiten von \$DO selbst nicht, die Ausführung der Anweisungen jedoch wie gewohnt durch Eingabe eines Stopzeichens abbrechen läßt.

#### C.10.1. Aufruf

Um \$DO auf eine Datei loszulassen, geben Sie ein:

```
dateiname parameter1 parameter2 ...
```

Dadurch wird \$DO automatisch gestartet, wenn die Datei "dateiname" vom Typ ".SUB" ist. \$DO sucht sich dann die Datei "dateiname.SUB" und bereitet diese auf.

Wie Sie sehen, können Sie hinter dem Dateinamen Parameter angeben. Das sind normalerweise beliebige Texte, die durch Leerstellen oder Kommata voneinander getrennt sind. Falls Sie einen Text übergeben wollen, der Leerstellen oder Kommata enthält, schließen Sie ihn einfach in Hochkommata ein:

```
$DO name Wort1 Wort2 'Dies ist Wort 3' Wort4
```

```
*****
*   $DO   *
*****
```

Diese Worte kann \$DO nun an bestimmten Stellen im Text der Datei einfügen. Der Mechanismus ist relativ einfach: \$DO numeriert alle Worte, die in der Kommandozeile hinter dem Programmnamen stehen, von 0 an durch. Der Name der Datei wäre also das Wort 0, das Wort "Wort1" das Wort 1 usw. An der Stelle, wo eines dieser Worte im Text stehen soll, setzen Sie ein Dollarzeichen "\$" ein, gefolgt von der Nummer des Wortes, das dort hinein soll. Ein Beispiel: Sie haben eine Datei namens "TEST.SUB" mit dem Inhalt:

```
Dies ist das 1. Wort: $1
Dies ist das 2. Wort: $2, und das 3. Wort: $3
```

Falls Sie nun \$DO wie folgt auf die Datei loslassen:

```
$DO TEST WORT1 WORT2 'DIES IST WORT 3'
```

erzeugt \$DO eine temporäre Datei mit folgendem Inhalt:

```
Dies ist das 1. Wort: WORT1
Dies ist das 2. Wort: WORT2, und das 3. Wort: DIES IST WORT 3
```

Auf diese Weise können Sie bis zu 9 Worte durch die Angabe von "\$1" bis "\$9" aus der Kommandozeile in Ihren Text übernehmen. Zudem haben Sie durch die Zeichenfolge "\$0" Zugriff auf den Namen der Datei, die gerade abgearbeitet wird.

Falls Sie das Dollarzeichen im Text als solches verwenden wollen, steht dem nichts entgegen. Sie sollten es hingegen zweimal hintereinander schreiben, falls das Dollarzeichen vor einer Zahl zu stehen kommen sollte. Es ist überhaupt ein guter Brauch, Dollarzeichen im Text doppelt anzugeben, um sie von Parametern unterscheiden zu können.

Schauen wir uns ein paar Beispiele an:

```
; das klappt gut:
$$SET DATEI +SYS
; das klappt ebenfalls noch:
$SET DATEI +SYS
; da muß das Dollarzeichen doppelt angegeben werden:
$$12345678
; in der nächsten Zeile wird '$1' durch einen Parameter ersetzt:
$12345678
```

```
*****
*      $DO      *
*****
```

### C.10.2. Eingabe von Steuerzeichen

Falls Sie Steuerzeichen in den Text einfügen wollen, benutzen Sie einfach die Notation, die EOS bei der Ausgabe von Steuerzeichen verwendet: tragen Sie den Hochpfeil "^", gefolgt von dem Buchstaben des Steuerzeichens ein. Ein Control-C wird dann durch die Zeichenfolge "^C" im Text dargestellt. Zulässige Steuerzeichen sind "^A" bis "^\_":

^\$ - NUL	^P - DLE = Drucker ein/aus
^A - SOH	^Q - DC1
^B - STX	^R - DC2
^C - ETX = Programmabbruch	^S - DC3
^D - EOT	^T - DC4
^E - ENQ	^U - NAK
^F - ACK	^V - SYN
^G - BEL = Piepen	^W - ETB
^H - BS = Backspace	^X - CAN
^I - HT = Tabulator	^Y - EM
^K - VT	^A - ESC = Escape
^L - FF = Seitenvorschub	^O - FS
^M - CR = Wagenrücklauf	^U - GS
^N - SO	^_ - US
^O - SI	

Ausgenommen sind folgende Steuerzeichen:

^J - Line feed	= wird einfach ignoriert
^Z - End of File	= Reserviert für das Dateiendekennzeichen
^^ - RS	= Reserviert für die Erzeugung von ""

Die letzte Angabe dient wie beim Dollarzeichen dazu, das Zeichen "" dadurch in den Text einzufügen, indem man es doppelt angibt.

### C.10.3. Weitere Sonderzeichen

\$DO kennt noch weitere Sonderzeichen. Diese sind allerdings nur am Anfang einer Zeile zulässig. Es gibt insgesamt vier davon:

#### 1) Der Doppelpunkt.

Alle Zeilen, die mit einem Doppelpunkt beginnen, werden vom Kommando - Interpreter nur bedingt akzeptiert. Sie werden nicht ausgeführt, wenn das zuletzt gelaufene Programm mit einem Fehler beendet wurde. Der Fehler kann ein von diesem Programm gesetzter Rückkehrcode sein oder aber auch durch einen EOS - Systemfehler bzw. einen Programmabbruch durch Eingabe eines Stopzeichens entstanden sein.

```
*****
*      $DD      *
*****
```

## 2) Das Kleiner - Zeichen "<".

Mit diesem Zeichen ist die explizite Deklaration einer Zeile als Programm - Eingabe möglich. Normalerweise übergibt \$DD seine Zeilen nur dem Kommando - Interpreter. Alle Programme, die während ihres Laufs Eingaben benötigen, müssen wie bisher von der Tastatur bedient werden. Zeilen, die mit einem Kleiner - Zeichen beginnen, werden dann als Programm - Eingabe herangezogen.

Es gelten dabei die folgenden Sonderbestimmungen: Benötigt ein Programm mehr Eingaben, als Programmzeilen da sind, wird bis zum Ende dieses Programms auf die Tastatur umgeschaltet. Sind hingegen mehr Programmzeilen da, als das Programm benötigt, wird vom Kommando - Interpreter eine Warnung ausgegeben, daß überflüssige Eingaben ignoriert wurden. Der Kommando - Interpreter setzt dann beim ersten Systemkommando wieder auf.

## 3) Das Fragezeichen.

Steht an erster Stelle einer Zeile ein Fragezeichen, wird dort eine Zeile interaktiv vom Bildschirm angefordert. Diese Eingabemöglichkeit funktioniert sowohl im System - Modus als auch innerhalb eines Programms.

## 4) Das Semikolon.

Dieses Zeichen kann zur Markierung eines Kommentars verwendet werden. Eine derartige Markierung ist allerdings nur im System - Modus möglich. Als Programm - Eingabe ist eine derartige Zeile nicht zulässig.

### C.10.4. Die Anzeige der Abarbeitung

Damit Sie immer wissen, wann \$DD aktiv ist, erscheinen zwei Arten von Anzeigen: zum Ersten erscheint in der Statuszeile im fünften Feld von links der Text "Execute". Desweiteren wird die Bereitschaftsanzeige beim Programm SHELL (aber nicht beim Menüprogramm USRSHELL) modifiziert. Für jeden aktiven \$DD - Lauf wird ein zusätzliches Großes - Zeichen ">" ausgegeben. Bei einem normalen \$DD - Lauf haben Sie also die folgende Bereitschaftsanzeige:

```
>>
```

Wenn Sie innerhalb des \$DD - Laufs einen weiteren \$DD - Lauf starten, erscheint die Bereitschaftsanzeige

```
>>>
```

und so weiter.

```
*****
*      $DO      *
*****
```

### C.10.5. Beispiele

Schauen wir uns ein Beispiel an. Nehmen wir an, wir haben ein Programm namens TESTPROG, das interaktiv mehrere Zeilen als Eingabe erwartet. Eine leere Zeile beendet das Programm. Nach dem Lauf von TESTPROG wollen wir ein beliebiges Kommando frei eingeben. Anschließend soll das Programm noch einmal gestartet werden, aber nur, wenn das Kommando, was wir eingegeben haben, einwandfrei beendet werden konnte. Eine \$DO - Datei würde dann wie folgt aussehen (in diesem Fall sind die Zeilen nummeriert, um darauf Bezug nehmen zu können):

```
1: ; Beispiel von Sonderzeichen - Zeilen:
2: TESTPROG
3: <Eingabe Nummer 1
4: <Eingabe Nummer 2
5: <
6: ? Bitte etwas eingeben
7: :TESTPROG
8: <Eingabe
```

Wenden wir uns den einzelnen Zeilen zu:

- 1: Dies ist ein Kommentar, der direkt ausgegeben wird.
- 2: Aufruf des Programms TESTPROG.
- 3: Beliebige Programm - Eingabe für TESTPROG.
- 4: Noch eine beliebige Programm - Eingabe für TESTPROG.
- 5: Eine leere Zeile, um TESTPROG zu beenden.
- 6: Hier wird eine Zeile vom Bildschirm angefordert. Der Text hinter dem Fragezeichen wird nicht interpretiert. Dort kann also ein Kommentar eingetragen werden.
- 7: Bedingtes Starten von TESTPROG
- 8: Eingabe für den bedingten Start von TESTPROG
- 9: Nach der ersten Eingabe fordert der bedingte Lauf von TESTPROG weitere Eingaben an, bis eine Leereingabe erfolgt. Erst dann wird vom Kommando - Interpreter dieser Text ausgegeben.

```
*****
*      $DD      *
*****
```

## C.10.6. Fehlermeldungen

Folgende Fehlermeldungen sind definiert:

Datei name nicht gefunden

Die Datei "name.SUB" wurde nicht g

Folgende Fehlermeldungen sind definiert:

Datei name nicht gefunden

Die Datei "name.SUB" wurde nicht gefunden.

Datei name: Inhaltsverzeichnis voll

Das Inhaltsverzeichnis des Laufwerks für temporäre Dateien ist voll.  
Die temporäre Datei konnte nicht eingerichtet werden.

Datei name: Schreibfehler

Die Platte ist voll. Die temporäre Datei konnte nicht vollständig geschrieben werden.

<Programmeingaben ignoriert>

Diese Warnung wird vom Kommando - Interpreter ausgegeben, wenn noch Programmeingaben von letzten Programmlauf übrig sind.

<Ausführung abgebrochen>

Wenn Sie ein Stopzeichen eingeben, wenn sich \$DD die nächste Eingabezeile holt, wird \$DD abgebrochen. Diese Meldung erscheint in diesem Fall auf dem Bildschirm.

```
*****
*   $ERASE   *
*****
```

## C.11. \$ERASE - Löschen von Dateien

Das Programm \$ERASE hilft Ihnen, Dateien wieder zu löschen. Es ist mit allen Feinheiten ausgestattet und wie die anderen Dienstprogramme durch Optionen erweiterbar. Falls Sie Benutzer des Kommando - Interpreters sind, haben Sie die folgenden Kommandos zur Verfügung:

```
LOESCHE   oder
LÖSCHE   oder
ERA       oder
ERASE
```

Alle vier Kommandos haben den selben Effekt: Geben Sie nur das Kommando ein, erklärt Ihnen \$ERASE, wie es genau aufzurufen ist. Geben Sie einen Dateinamen an, wie zum Beispiel:

```
LÖSCHE DATEI1.COM
```

wird \$ERASE versuchen, diese Datei zu löschen. Konnte die Datei gelöscht werden, erfolgt die Meldung:

```
A:DATEI1 .COM ( 2 KB)   gelöscht.
1 Datei(en) gelöscht, 2 KBytes freigegeben, 346 KBytes frei.
```

Sie sehen, daß Ihnen gesagt wird, wie viel Platz die Datei vor der Löschung beanspruchte, und wie viel Platz durch die Löschung freigegeben wurde. Wurde die Datei nicht gefunden, erfolgt die Meldung

```
0 Datei(en) gefunden.
```

Sie können selbstverständlich auch mehrere Dateien löschen, indem Sie einen mehrdeutigen Dateinamen angeben. Dies wurde zum Beispiel wie folgt aussehen:

```
LÖSCHE *.COM
3 Datei(en) gefunden. Löschen? (J/N) J
A:$DIR .COM ( 10 KB)   gelöscht.
A:$SET .COM ( 8 KB) r/o nicht gelöscht.
A:PROGRAMM.COM ( 20 KB) gelöscht.
2 Datei(en) gelöscht, 30 KBytes freigegeben, 376 KBytes frei.
```

\$ERASE fragt immer vorsichtshalber noch einmal nach, ob die Dateien wirklich gelöscht werden sollen, falls mehrere Dateien gefunden wurden. Die Datei "\$SET.COM" konnte nicht gelöscht werden, da sie mittels => \$SET mit einem Schreibschutz versehen wurde. Allerdings wäre sie durch Angabe einer Option löscher gewesen, was uns zu den Optionen von \$ERASE bringt.

```
*****
*   $ERASE   *
*****
```

### C.11.1. SYSTEM - Nur versteckte Dateien löschen

Im Normalfall werden von \$ERASE nur sichtbare Dateien erfaßt. Dateien, die durch => \$SET mit dem System - Attribut versehen wurden, werden nicht gelöscht. Sie können durch die Eingabe "+SYSTEM" auf alle derartige Dateien zugreifen.

### C.11.2. ALLE - Alle Dateien eines Benutzerbereichs löschen

Wenn Sie sowohl Normaldateien als auch Systemdateien löschen wollen, geben Sie die Option "+ALLE" an.

### C.11.3. BEN - Dateien in allen Benutzerbereichen löschen

Mit der Option "+BEN" können Sie den Löschvorgang auf sämtliche Benutzerbereiche erweitern. Diese Option ist mit den Optionen "+SYSTEM" und "+ALLE" kombinierbar. \$ERASE zeigt Ihnen dann an Stelle des Laufwerks den Benutzerbereich der gelöschten Datei an:

```
LÖSCHE *.COM +BEN
3 Datei(en) gefunden. Löschen? (J/N) J
0:$DIR .COM ( 10 KB) gelöscht.
0:$SET .COM ( 8 KB) r/o nicht gelöscht.
1:PROGRAMM.COM ( 20 KB) gelöscht.
2 Datei(en) gelöscht, 30 KBytes freigegeben, 376 KBytes frei.
```

In diesem Fall wäre die Datei "\$DIR.COM" im Benutzerbereich 0 und die Datei "PROGRAMM.COM" im Benutzerbereich 1 gelöscht worden.

### C.11.4. R/O - Auch schreibgeschützte Dateien löschen

In den bisherigen Beispielen haben wir gesehen, daß schreibgeschützte Dateien vom Löschen ausgenommen wurden. Wenn Sie die Option "+R/O" angeben, wird der Schreibschutz beim Löschen ignoriert. Das sähe dann so aus:

```
LÖSCHE *.COM +R/O
3 Datei(en) gefunden. Löschen? (J/N) J
A:$DIR .COM ( 10 KB) gelöscht.
A:$SET .COM ( 8 KB) r/o gelöscht.
A:PROGRAMM.COM ( 20 KB) gelöscht.
3 Datei(en) gelöscht, 38 KBytes freigegeben, 368 KBytes frei.
```

```
*****
*   $ERASE   *
*****
```

#### C.11.5. FRAGEN - Jede Datei bestätigen lassen

Wenn Sie selektiv löschen wollen, geben Sie die Option "+FRAGEN" ein. Dann wird bei jeder Datei einzeln gefragt, ob diese gelöscht werden soll:

```
LÖSCHE *.COM +FRAGEN
3 Datei(en) gefunden.
A:$DIR .COM ( 10 KB)   Löschen? (J/N) J
A:$SET .COM (  8 KB) r/o nicht gelöscht.
A:PROGRAMM.COM ( 20 KB)   Löschen? (J/N) J
2 Datei(en) gelöscht, 30 KBytes freigegeben, 376 KBytes frei.
```

#### C.11.6. DIREKT - sofortiges Löschen

In den obigen Beispielen haben Sie gesehen, daß jedesmal noch einmal zurückgefragt wird, wenn mehrere Dateien gefunden wurden und gelöscht werden sollen. Das kann insbesondere bei der Abarbeitung von Kommandodateien hinderlich sein. Die Rückfrage kann jedoch mit der Option "+DIREKT" unterdrückt werden. In diesem Fall werden alle gefundenen Dateien gelöscht, ohne daß noch einmal gefragt wird, ob es auch seine Richtigkeit damit hat.

#### C.11.7. Englische Optionen

Die nachfolgenden Optionen haben allesamt eine englische Entsprechung, damit auch Benutzer, die an anderssprachige Versionen von EOS gewöhnt sind, ohne Umlernen mit der deutschen Version umgehen können:

ALLE	entspricht dem englischen	ALL
BEN	entspricht dem englischen	USERS
FRAGEN	entspricht dem englischen	CONFIRM

```
*****
*   $INIT   *
*****
```

### C.12. \$INIT - Initialisieren von Disketten

Fabrikneue Disketten werden je nach Fabrikat entweder unformatiert oder aber mit einem Standardformat beschrieben geliefert. Disketten müssen in irgendeiner Weise formatiert sein, damit sie überhaupt verwendbar sind. Somit ist es nötig, Disketten neu zu formatieren, ehe sie auf Ihrem Rechner verwenden können.

\$INIT gehört zu den menugesteuerten Programmen. Es wird aufgerufen durch die einfache Eingabe von

```
$INIT5 für Minidisketten,
$INIT8 für Standarddisketten und
$INITHD für die Hard Disk.
```

Das Menu selbst hat drei Spalten: die linke Spalte dient zum Anwählen des Laufwerks, in dem die zu initialisierende Diskette liegt. Die mittlere Spalte enthält eine Tabelle der möglichen Diskettenformate. Die rechte enthält letztendlich die Kommandoliste.

Für ganz eilige ist bereits ein Format vordefiniert worden. Sie merken das daran, daß das Fenster bereits über dem "START" - Kommando steht. Das bedeutet, daß Sie nur durch Drücken der "ENTER" - Taste bereits den Formatiervorgang einleiten können.

Wenn Sie "START" angewählt haben, erscheint in der Schirmmitte eine Art Meßplatte, die durch Punktchen gekennzeichnet ist. Diese Meßplatte enthält einen Punkt für jede zu initialisierende Spur auf Ihrer Diskette. Sie ist natürlich für Minidisketten und Standarddisketten unterschiedlich lang. Für jede formatierte Spur wird ein Punkt durch ein "I" ersetzt. Wenn alle Spuren initialisiert sind, wird jede Spur noch einmal prüfgelesen. Bei jeder erfolgreich prüfgelesenen Spur wird das "I" in der Meßplatte durch ein "P" ersetzt.

Falls die Diskette erfolgreich initialisiert und prüfgelesen werden konnte, tut das Programm dies in der untersten Zeile kund. Falls ein Fehler auftrat, wird dies ebenfalls dort gemeldet.

Was ist, wenn Sie den Initialisierungsvorgang abbrechen wollen? Nun, das geht wie immer: geben Sie CONTROL-C oder ein anderes => Stopzeichen ein. Der Formatiervorgang wird sofort abgebrochen.

```
*****
*                                                                 *
*   Die Initialisierung von Disketten zerstört die Daten       *
*                   VOLLSTÄNDIG !                               *
*                   Achten Sie daher darauf,                   *
*                   daß Sie die richtige Diskette eingelegt   *
*                   ehe Sie "START" anwählen!                  *
*                                                                 *
*****
```

```
*****
*   $INIT   *
*****
```

### C.12.1. Minidiskettenformate

Für Minidisketten sind die folgenden Formate definiert:

- FDOS/CPM - Dieses Format wird von SHARP für das FDOS - Betriebssystem. Desweiteren gibt es eine Implementation von CP/M 2.2, die dieses Format verwendet.
- MZ-80B - Das Format des SHARP - Rechners MZ-80B. Dieses und die Implementation des Bildschirmes des MZ-80B gewährleistet die volle Kompatibilität der unter CP/M auf dem MZ-80B entwickelten Software.
- ~ PC-3201 - Das DiCOS - Format auf dem Rechner PC-3201 der Firma SHARP. Wenn Sie glücklicher Besitzer eines PC-3201 unter DiCOS sind, werden Sie dieses Format zu schätzen wissen.
- MZ-35xx - Das EOS - Standardformat für Ihren Rechner. Dieses Format vereint die größte Packungsdichte mit dem schnellsten Zugriff. Dies ist auch die Voreinstellung für das Programm \$INIT5.

### C.12.2. Standarddisketten

Hier stehen ebenfalls vier verschiedene Formate zur Verfügung. Im Grunde sind es sogar acht Formate, da die Formate für einseitig und doppelseitig beschreibbare Disketten unterschiedlich sind. Die Formate im Einzelnen:

- ~ 128 Bytes - Die einseitige Version dieses Formats hat sich als Standardformat zum Austausch von Software weltweit durchgesetzt. Dieses Format ist bei Software - Versendern als "A1" - Format bekannt. Zu empfehlen beim Austausch von Daten und Programmen.
- 256 Bytes - Die einseitige Version hat wiederum eine Besonderheit: Sie ist kompatibel zum Aufzeichnungsformat des Rechners HAYAC-2900 von SHARP. Die doppelseitige Version ist kompatibel zum Aufzeichnungsformat der IMS-800 - Rechnerfamilie.
- 512 Bytes - Nur der Vollständigkeit halber mit in der Liste. Zu nichts kompatibel.
- 1024 Bytes - Das EOS - Standardformat für Standarddisketten. Kompatibel zum EOS-Format der IMS-8000 - Rechnerfamilie. Dies ist die Vorbesetzung für das Programm \$INIT8.

```
*****  
*   $INIT   *  
*****
```

### C.12.3. Die Hard Disk

Das \$INIT - Programm für die Hard Disk verhält sich etwas anders als die entsprechenden Diskettenprogramme. Zum ersten fällt die mittlere Spalte weg, da das Format der Hard Disk nicht von Ihnen bestimmt werden kann.

Eine weitere Änderung betrifft die Behandlung von Lesefehlern beim Prüfllesen. Die Disketten - Formatierprogramme brechen die Formatierung bei einem Lesefehler einfach ab und fordern Sie auf, die defekte Diskette wegzuworfen. Diese Methode ist nach langem Nachdenken und Beraten bezüglich einer Hard Disk für unrentabel erklärt worden. Deshalb werden defekte Blöcke auf einer Hard Disk in einer speziellen Datei zusammengefaßt. Diese Datei belegt dann genau die Stellen auf der Hard Disk, bei denen Lesefehler auftraten, so daß die Stellen für andere Dateien gesperrt sind.

Für Leute, die es wissen wollen: die Datei heißt "BadBlock.SYS", ist im Benutzerbereich 15 zu finden und hat die Attribute "R/O", "SYSTEM" und "BACKUP" gesetzt.

```
*****
* $INITDIR *
*****
```

### C.13. \$INITDIR - Steuern von Zeiteinträgen

EOS hat die segensreiche Fähigkeit, den Zeitpunkt von Dateizugriffen festhalten zu können. Im Einzelnen können die folgenden Zeiten für jede Datei festgehalten werden:

1. Der Zeitpunkt des letztes Zugriffs,
2. Der Zeitpunkt der letzten Änderung oder
3. Der Zeitpunkt der Neueinrichtung der Datei.

Es können allerdings maximal zwei dieser sogenannten Zeitmarken pro Datei geschrieben werden. Das bedeutet, daß Sie sich zwischen den beiden Zeitmarken für den letzten Zugriff und für die Neueinrichtung entscheiden müssen, da die eine Marke die andere ausschließt.

Um diese Zeitmarken auch sehen zu können, gibt es die Option "ZEIT" beim INHALT - Kommando. Genauereres steht in der Beschreibung des entsprechenden Dienstprogramms \$DIR.

Die Steuerung dieser Zeiteinträge wird vom Programm \$INITDIR wahrgenommen. Dieses Programm ist menügesteuert. Das bedeutet in diesem Fall, daß auf genaue Erklärungen verzichtet werden kann, da das Programm selbsterklärend ist. Der Aufruf ist ganz einfach:

`$INITDIR`

Wenn Zeitmarken eingetragen werden sollen, muß dafür im Inhaltsverzeichnis Platz geschaffen werden. Daher werden Sie beim Start des Programms \$INITDIR den Punkt "MARKEN SCHREIBEN" finden. Wenn Sie diesen Punkt anwählen, wird im Inhaltsverzeichnis Platz für Zeitmarken geschaffen. Dieser Platz beansprucht etwa ein Viertel des Inhaltsverzeichnisses, was bedeutet, daß Sie bei aktivierten Zeitmarken nur etwa 75 Prozent der sonst möglichen Dateieinträge erzeugen können. Im Normalfall wird es Ihnen sowieso nie passieren, daß Ihr Inhaltsverzeichnis überläuft. Falls es dennoch passieren sollte, können Sie die Zeiteinträge wieder durch das Kommando "MARKEN LÖSCHEN" aus dem Inhaltsverzeichnis entfernen.

\$INITDIR bietet Ihnen noch zusätzlich die Möglichkeit, Ihren Disketten Namen zu verleihen. Diese Namen tauchen bei der Ausgabe des Inhaltsverzeichnisses in der linken oberen Bildschirmecke auf. Diese Namen nennt man im Neudeutschen "Label". Um einen derartigen Label einzutragen, wählen Sie einfach den Punkt "Plattenlabel eintragen" an. Sie springen dann in das untere Feld, wo der Label ausgegeben wird. Dort können Sie Ihren neuen Label eintragen. Wenn Sie dieses Feld verlassen wollen, ohne den Label zu ändern, drücken Sie einfach die Taste HOME oder die Pfeil auf- bzw. Pfeil - Abwärtstaste. Sie müssen wie üblich Ihre Eingabe mit ENTER abschließen, damit der Label neu geschrieben wird.

```
*****  
* $INITDIR *  
*****
```

Der Label kann auch separat durch Anwahl des Kommandos "LABEL SCHREIBEN" neu geschrieben werden. Dies sollten Sie nach der Selektion der Art der Zeitmarkenbehandlung immer machen, damit die entsprechenden Anweisungen zusammen mit dem Label auf der Platte festgehalten werden.

```
*****
*   $KEYDEFS   *
*****
```

#### C.14. \$KEYDEFS - Speichern und Laden der Tastaturdefinitionen

Wenn Sie sich Ihre Tastatur mit Hilfe der Tasten CTRL-7 und CTRL-8 (siehe Abschnitt A) programmiert haben, werden Sie sicher den Wunsch haben, Ihre Definitionen der Nachwelt zu erhalten. Zu diesem Zweck geben Sie ein:

```
$KEYDEFS +SPEICHERN      oder      $KEYDEFS +S
```

Dieses Programm speichert Ihre Tastaturdefinitionen in die Datei "KEYDEFS.KEY" ab. Wenn Sie Ihre Definitionen wiederhaben wollen, geschieht dies ganz einfach wie folgt:

```
- $KEYDEFS
```

Das "+SPEICHERN" kann, wie bei jeder anderen Option, auch durch "+S" abgekürzt werden.

Sie haben die Möglichkeit, in der Kommandozeile den Namen einer Datei anzugeben, die erzeugt bzw. aus der geladen werden soll. Fehlt der Dateityp, wird automatisch der Typ ".KEY" generiert. Ein Beispiel: Das Speichern und das Laden über die Datei "KEY1.KEY" sieht so aus:

```
Speichern: $KEYDEFS KEY1 +S
Laden:     $KEYDEFS KEY1
```

```
*****  
*      $MOD      *  
*****
```

### C.15. \$MOD - Arbeiten mit Funktionsmodulen

Dieses Programm ist als Hilfsmittel für Entwickler gedacht. Sie als "Normalverbraucher" werden dieses Programm nie benötigen. Falls Sie jedoch an der Technik und der Handhabung von Funktionsmodulen interessiert sind, sollten Sie diesen Abschnitt lesen.

Um eine größtmögliche Flexibilität von EOS zu gewährleisten, kann EOS von Programmierern selbst erweitert werden. Dies wird durch die sogenannten Funktionsmodule ermöglicht. Ein Funktionsmodul ist ein spezielles Assemblerprogramm, das mittels des Programms \$MOD an ein normales Programm "angehängt" wird. Wenn EOS beim Laden eines derart erweiterten Programms auf ein Funktionsmodul stößt, wird es an die oberste Stelle des verfügbaren Arbeitsspeichers geladen. Dort fängt es alle Systemaufrufe auf, die das geladene Programm benutzt. Diese Aufrufe werden dann zum Teil von diesem Funktionsmodul direkt interpretiert. Ein Beispiel eines Funktionsmoduls ist das Programm \$DO. Das Programm selbst tut nichts anderes, als die angegebene Datei zu konvertieren und eine temporäre Datei zu erzeugen. Diese Datei wird dann von einem ubergelagerten Funktionsmodul abgearbeitet, das die EOS - Funktion 68 abfängt und selbst interpretiert. Wenn also ein anderes Programm die EOS - Funktion 68 benutzen will, muß gewährleistet sein, daß das entsprechende Funktionsmodul auch verfügbar ist.

Ein Funktionsmodul bestimmt meist selbst, wie lange es im Arbeitsspeicher verbleibt. Ist es mit seiner Arbeit fertig, setzt es ein bestimmtes Flag. EOS wird dann dieses Modul nach dem Ende des aktuellen Programmlaufs aus dem Speicher entfernen.

Der Typ eines Funktionsmoduls ist "RSX" vom englischen "Resident System Extension". Der interne Aufbau eines derartigen Moduls ist genauestens im Systemhandbuch, Abschnitt A, beschrieben.

\$MOD ist nun dasjenige Programm, mit dem sich ein oder auch mehrere Funktionsmodule an ein Programm anfügen lassen. Desweiteren kann die Entfernung und Generierung von Modulen durch eine Reihe von Optionen gesteuert werden. Ohne Optionen gibt es zwei Aufrufmöglichkeiten, die wir gleich behandeln wollen.

Eine Besonderheit ist noch zu erwähnen: Der Ladevorgang selbst wird in EOS ebenfalls durch ein Funktionsmodul erledigt. Dieses Modul hat den Namen "\$PGMLDR" und wird bei jedem Laden des SHELL - Programms mitgeladen, da der SHELL selbst nicht fähig ist, Programme zu laden. Dieses Lademodul wird im Normalfall vor dem Start des geladenen Programms aus dem Speicher entfernt. Werden jedoch weitere Module geladen, bleibt es aktiv im Speicher und kann über die EOS - Funktion 59 benutzt werden. Mit einer speziellen Option kann ein Programm so modifiziert werden, daß der Lader im Speicher verbleibt, ohne daß dazu ein zusätzliches Modul geladen werden muß.

```
*****
*      $MOD      *
*****
```

### C.15.1. Das Anfügen von Funktionsmodulen

Dieser Aufruf sieht wie folgt aus:

```
$MOD comfile rsx1 rsx2 ...rsx15
```

Der erste Parameter "comfile" bezeichnet hier die ".COM" - Datei, an die Module angefügt werden sollen. Danach folgt die Liste der RSX - Dateien, die als Funktionsmodul angefügt werden sollen. Eine ".COM" - Datei kann mit maximal 15 Modulen erweitert werden. Diese Module werden in der Reihenfolge ihres Auftretens in der Kommandozeile geladen, was bedeutet, daß bei zwei Modulen, die die selbe Funktion abfangen, das zuletzt erwähnte und somit geladene Modul den Aufruf abfangt.

Falls die ".COM" - Datei bereits angefügte Module besitzt, werden die neuen Module hinten an diese angefügt. Ist ein in der Liste erwähntes Modul bereits in der ".COM" - Datei vorhanden, wird es durch das neue Modul ersetzt.

Ein Beispiel: Sie haben eine Datei namens "PROG.COM". Diese Datei enthält bereits die Module "MOD1", "MOD2" und "MOD3". Wenn Sie nun eingeben:

```
$MOD PROG MOD1 MOD4
```

wird MOD1 in der Datei PROG.COM durch das neue Modul MOD1 ersetzt. Das Modul MOD4 wird hinten an die Liste angefügt. Wenn Sie das Programm nun starten, fangt zuerst das Modul MOD4 die EOS - Aufrufe ab, dann MOD3, MOD2 und zuletzt MOD1.

### C.15.2. Das Entfernen von Funktionsmodulen

Um alle Funktionsmodule aus einer ".COM" - Datei zu entfernen (was eine normale ".COM" - Datei zur Folge hat), geben Sie ein:

```
$MOD progname
```

Dadurch werden aus der Datei "progname.COM" alle Module wieder entfernt.

```
*****
*   $MOD   *
*****
```

### C.15.3. LADER - Verbleiben des Programmladers

Wenn Sie den Programmlader aus Ihrem Programm heraus benutzen mochten, ohne Funktionsmodule laden zu müssen, geben Sie ein:

```
$MOD progname +LADER
```

Dadurch wird die Datei "progname.COM" mit einem speziellen Vorsatz versehen, der dem Lader signalisiert, daß er sich nicht selbst entfernen darf. Somit steht dem Programm der Systemaufruf 59 zur Verfügung.

### C.15.4. COM - Funktionsmodule zusammenfassen

Es kann gelegentlich vorkommen, daß Sie ein oder mehrere Module für mehrere nachfolgende Programme benötigen. Um zu vermeiden, daß Sie das Modul an jedes Programm binden müssen, können Sie mit Hilfe dieser Option eine Datei erzeugen, die nur Funktionsmodule enthält. Diese Module können bereits das Flag gesetzt haben, das ihre Inaktivität anzeigt; sie werden dennoch frühestens erst nach dem nächsten Programmaufstart entfernt. Wenn Sie zum Beispiel eingeben:

```
$MOD MOD1 MOD2 MOD3 +COM
```

werden die Module MOD1.RSX, MOD2.RSX und MOD3.RSX zu einer Datei namens "MOD1.COM" zusammengefaßt. Diese Datei kann wie jede andere Programmdatei aufgerufen werden und tut nichts anderes, als die darin enthaltenen Module zu aktivieren.

### C.15.5. PROT - Protokollierung unterdrücken

Im Normalfall erhalten Sie ein Ladeprotokoll auf dem Bildschirm, so daß Sie über die Struktur der entstandenen Datei informiert werden. Dieses Protokoll könnte so aussehen:

Name	Typ	Offset	Große	
PROG	.COM	0100	1000	
MOD1	.RSX	1100	02E0	alt
MOD2	.RSX	1480	00F0	ersetzt
MOD3	.RSX	1600	0300	neu

Die Spalte "Offset" gibt an, ab welcher Adresse relativ zum Dateianfang das jeweilige Modul beginnt. Die Größe gibt die Größe des Modul - Codes in Bytes an. Schließlich wird noch erwähnt, ob das angefügte Modul aus der alten ".COM" - Datei übernommen wurde ("alt"), durch ein neues Modul ersetzt wurde ("ersetzt") oder neu angefügt wurde ("neu"). Wenn Sie das Protokoll nicht haben mochten, geben Sie die Option "-PROT" an.

```
*****
*      $MOD      *
*****
```

### C.15.6. MODULE - Liste aller aktiven Module

Der Vollständigkeit halber wurde diese Option auch hier implementiert. Wenn Sie eingeben:

```
$MOD +MODULE
```

bekommen Sie eine Liste aller zur Zeit aktiven Funktionsmodule auf dem Bildschirm:

Aktive Funktionsmodule:

Name	Strt	Größe
\$DO	E706	0500
\$DO	EC06	0500
\$PGMLDR	F106	0300

In diesem Beispiel konnte gerade ein zweifach verschachtelter \$DO - Lauf stattfinden. Der Programmlader verbleibt dadurch aktiv im Speicher.

Die Option kann auch in Verbindung mit einer normalen Kommandozeile eingegeben werden. In diesem Fall wird vor der Bearbeitung der Kommandozeile diese Liste ausgegeben. Ist zur Zeit kein Modul aktiv, erfolgt die Meldung:

Keine Funktionsmodule aktiv.

### C.15.7. Englische Optionen

Die folgenden englischen Optionen sind definiert:

+COM	entspricht dem englischen	+COM
+LADER	entspricht dem englischen	+LOADER
+MODULE	entspricht dem englischen	+MODULES
-PROT	entspricht dem englischen	-PROT

```
*****
* $RENAME *
*****
```

## C.16. \$RENAME - Umbenennen von Dateien

Das Programm \$RENAME hilft Ihnen, Dateien umzubenennen. Es ist mit allen Feinessen ausgestattet und wie die anderen Dienstprogramme durch Optionen erweiterbar. Falls Sie Benutzer des Kommando - Interpreters sind, haben Sie die folgenden Kommandos zur Verfügung:

```
UMB      oder
UMBENENNE oder
REN      oder
RENAME
```

Alle vier Kommandos haben den selben Effekt: Geben Sie nur das Kommando ein, erklärt Ihnen \$RENAME, wie es genau aufzurufen ist.

Es gibt zwei Aufrufarten des \$RENAME - Kommandos. Die erste ist CP/M - kompatibel:

```
UMBENENNE date1.neu = date1.alt
```

Hier geben Sie zuerst den neuen Dateinamen an, gefolgt von einem Gleichheitszeichen und dem alten Dateinamen. Vielen Leuten erscheint diese Schreibweise etwas umständlich. Deshalb erlaubt \$RENAME noch eine andere Schreibweise:

```
UMBENENNE date1.alt date1.neu
```

wo Sie zuerst den alten Dateinamen und dann den neuen eingeben, ohne ein Gleichheitszeichen dazwischen zu stellen. Suchen Sie sich Ihre private Aufrufart aus.

Das ist aber noch nicht alles. \$RENAME erlaubt Ihnen nämlich, auch mehrdeutige Dateinamen anzugeben. Das Kommando

```
UMBENENNE *.COM *.OLD
```

benennt beispielsweise alle Ihre ".COM" - Dateien in Dateien des Typs ".OLD" um.

Diese Umbenennung von mehrdeutigen Dateinamen verläuft in etwa so: Wird eine Datei gefunden, auf die der Name paßt, werden in diesem Namen alle Zeichen in den neuen Namen übernommen, die in beiden angegebenen Dateinamen durch Fragezeichen, respektive Sternchen, ersetzt sind. Falls dabei Unsinn zustande kommt, wird die Ersetzung nicht durchgeführt. Ein Beispiel: Sie haben die Dateien

```
LANG1111.DAT   KURZ.DAT   LANG2222.DAT
```

```
*****
*   $RENAME   *
*****
```

und geben das Kommando

```
UMBENENNE *.DAT ??????X.DAT
```

so wurden die folgenden Namen entstehen:

```
LANG111X.DAT   KURZ   X.DAT   LANG222X.DAT
```

Sie werden unschwer erkennen, daß der zweite Name "KURZ X.DAT" wegen darin enthaltener Leerstellen nicht möglich ist. Die Umbenennung wurde in diesem Fall so aussehen:

```
LANG111X.DAT   KURZ.DAT   LANG222X.DAT
```

Das "X" wird im zweiten Dateinamen einfach ignoriert.

Sie sollten, falls Sie mehrdeutige Dateinamen verwenden, noch darauf achten, daß keine zwei Dateien gleichen Namens entstehen; \$RENAME lehnt sonst die Umbenennung der zweiten Datei ab.

Ein Beispiel:

```
UMBENENNE *.COM *.CIM
3 Datei(en) gefunden. Umbenennen? (J/N) j
A:$DIR   .COM ( 11 KB)   umbenannt in $DIR   .CIM
A:$SET   .COM ( 8 KB) r/o nicht umbenannt
A:$ERASE .COM ( 8 KB)   umbenannt in $ERASE .CIM
```

\$RENAME fragt immer vorsichtshalber noch einmal nach, ob die Dateien wirklich gelöscht werden sollen, falls mehrere Dateien gefunden wurden. Die Datei "\$SET.COM" konnte nicht umbenannt werden, da sie mittels => \$SET mit einem Schreibschutz versehen wurde. Allerdings wäre sie durch Angabe einer Option löscherbar gewesen - was uns zu den Optionen von \$RENAME bringt.

#### C.16.1. SYSTEM - Nur versteckte Dateien umbenennen

Im Normalfall werden von \$RENAME nur sichtbare Dateien erfaßt. Dateien, die durch => \$SET mit dem System - Attribut versehen wurden, werden nicht umbenannt. Sie können durch die Eingabe "+SYSTEM" auf alle derartige Dateien zugreifen.

#### C.16.2. ALLE - Alle Dateien eines Benutzerbereichs umbenennen

Wenn Sie sowohl Normaldateien als auch Systemdateien umbenennen wollen, geben Sie die Option "+ALLE" an.

```
*****
*   $RENAME   *
*****
```

### C.16.3. BEN - Dateien in allen Benutzerbereichen umbenennen

Mit der Option "+BEN" könne Sie das Umbenennen auf sämtliche Benutzerbereiche erweitern. Diese Option ist mit den Optionen "+SYSTEM" und "+ALLE" kombinierbar. \$RENAME zeigt Ihnen dann an Stelle des Laufwerks den Benutzerbereich der umbenannten Datei an:

```
UMBENENNE *.COM *.CIM +BEN
3 Datei(en) gefunden. Umbenennen? (J/N) j
0:$DIR    .COM ( 11 KB)    umbenannt in $DIR    .CIM
1:$SET    .COM (   8 KB) r/o nicht umbenannt
0:$ERASE  .COM (   8 KB)    umbenannt in $ERASE  .CIM
```

### C.16.4. R/O - Auch schreibgeschützte Dateien umbenennen

In den bisherigen Beispielen haben wir gesehen, daß schreibgeschützte Dateien vom Umbenennen ausgenommen wurden. Wenn Sie die Option "+R/O" angeben, wird trotz Schreibschutz umbenannt. Das sahe dann so aus:

```
UMBENENNE *.COM *.CIM +BEN
3 Datei(en) gefunden. Umbenennen? (J/N) j
0:$DIR    .COM ( 11 KB)    umbenannt in $DIR    .CIM
1:$SET    .COM (   8 KB) r/o umbenannt in $SET    .CIM
0:$ERASE  .COM (   8 KB)    umbenannt in $ERASE  .CIM
```

### C.16.5. FRAGEN - Jede Datei bestätigen lassen

Wenn Sie selektiv umbenennen wollen, geben Sie "+FRAGEN" an. Dann wird bei jeder Datei einzeln gefragt, ob diese umbenannt werden soll:

```
UMBENENNE *.COM *.CIM +FRAGEN
3 Datei(en) gefunden.
A:$DIR    .COM ( 11 KB)    Umbenennen? (J/N) j
A:$SET    .COM (   8 KB) r/o nicht umbenannt
A:$ERASE  .COM (   8 KB)    Umbenennen? (J/N) j
```

### C.16.6. DIREKT - sofortiges Umbenennen

In den obigen Beispielen haben Sie gesehen, daß jedesmal noch einmal zurückgefragt wird, wenn mehrere Dateien gefunden wurden und umbenannt werden sollen. Das kann insbesondere bei der Abarbeitung von Kommandodateien hinderlich sein. Die Rückfrage kann jedoch mit der Option "+DIREKT" unterdrückt werden. In diesem Fall werden alle gefundenen Dateien umbenannt, ohne daß noch einmal gefragt wird, ob das auch so stimmt.

```
*****
*   $SAVE   *
*****
```

### C.17. \$SAVE - Abspeichern des Arbeitsspeichers

Dieses Programm erfüllt eine sehr spezielle Funktion, die Sie, lieber Anwender, selten benötigen werden. Sie ist nur für Programmierer von Interesse.

Unter CP/M 2.2 bzw. DiCOS war das SAVE - Kommando fest in den Kommandointerpreter eingebaut. Mit diesem Kommando konnte man Teile des Arbeitsspeichers in eine Datei übertragen. Somit war es möglich, ein Programm mit Hilfe eines geeigneten Testprogramms (Debugger) zu laden und zu ändern. nach dem Lauf des testprogramms konnte dann das geänderte Programm mittels des SAVE - Kommandos auf die Platte übertragen werden.

- EOS kennt dieses Kommando nicht, da der Kommandointerpreter ein Programm wie jedes andere ist und somit auch den selben Arbeitsspeicherbereich belegt. Aus diesem Grunde mußte ein anderer Mechanismus implementiert werden, um das Abspeichern eines Programms auf die Platte zu ermöglichen, ehe der Speicher vom Kommandointerpreter neu beschrieben wurde.

\$SAVE besteht im Grunde nur aus einem Funktionsmodul. Wenn Sie das Kommando

```
$SAVE
```

eingeben, wird dieses Modul an eine Stelle direkt unterhalb des EOS - Systemeintrags geladen. dort bleibt es bis zum nächsten Programmende inaktiv. Nach Ende des nachfolgenden Programms (das kann beispielsweise ein Debugger sein), meldet sich \$SAVE mit

```
EOS V3 - Abspeichern des Arbeitsspeichers
```

- Dateiname :

Hier können Sie den Namen der Datei angeben, in die der Hauptspeicher abgelegt werden soll. Drücken Sie nur die "ENTER" - Taste, bricht \$SAVE ab. Ist die angegebene Datei vorhanden, wird noch einmal zurückgefragt:

```
datei.typ loeschen? (J/N)
```

Geben Sie etwas anderes als "J" ein, wird ein neuer Dateiname angefordert. Die nächste Frage ist die nach der Start- sowie der Endadresse des abzulegenden Bereichs. Diese beiden Werte sind im Sedezimalcode anzugeben. Nach dem Schreiben der Datei deaktiviert sich \$SAVE selbst.

```
*****  
*   $SAVE   *  
*****
```

Ein Beispiel: Sie wollen den Speicherbereich von 100H bis 3FFH in die Datei TEST.COM ablegen:

```
>$SAVE
```

```
><Hier Aufruf eines Debuggers o. a.>
```

```
EOS V3 - Abspeichern des Arbeitsspeichers
```

```
Dateiname   : test.com  
TEST.COM loeschen? (J/N) j  
Startadresse: 0100  
Endadresse  : 03FF
```

```
Ende <Debugger> 1:00  
>
```

Im EOS - Lieferumfang ist auch das Funktionsmodul \$SAVE.RSX enthalten. Dieses Modul kann mit Hilfe von => \$MOD direkt an einen Debugger o.a. angebunden werden. Der explizite Start von \$SAVE erubrigt sich somit.

Der Aufbau und die Wirkungsweise eines Funktionsmoduls ist im Systemhandbuch im Abschnitt A ausfuhrlich beschrieben.

```
*****
*   $SET   *
*****
```

### C.18. \$SET - Setzen und Löschen von Dateiattributen

Ihnen ist sicherlich bereits an verschiedenen Stellen das Dateiattribut "SYSTEM" über den Weg gelaufen. Dieses und andere Attribute können mit diesem Programm gesetzt und gelöscht werden.

Die meisten möglichen Attribute sind fest für EOS reserviert. Drei Attribute können jedoch von Ihnen nach Belieben gesetzt werden:

SYSTEM - Dieses Attribut macht eine Datei global zugänglich. Sie kann auch von anderen Benutzerbereichen aus gelesen werden. Außerdem erscheint eine Datei mit dem SYSTEM - Attribut nicht im normalen Inhaltsverzeichnis eines Laufwerks, was unter Umständen erheblich zur Übersichtlichkeit desselben beiträgt.

BACKUP - Das Hilfsprogramm => \$COPY setzt bei jeder Datei, die gesichert wurde, dieses Attribut. EOS selbst löscht dieses Attribut automatisch, wenn die Datei geändert wurde. Somit ist eine Datei, die noch nicht gesichert wurde, schnell zu erkennen. Eine Sonderfunktion von \$COPY kopiert nur Dateien, die dieses Attribut nicht besitzen.

R/O - Dateien mit dem R/O (Read Only) - Attribut können weder geändert noch gelöscht werden.

Wie setzen Sie nun diese Attribute? Ganz einfach, indem Sie den Namen des jeweiligen Attributs, angeführt von einem Plus- oder einem Minuszeichen, angeben. Geben Sie kein Attribut an, wird das nicht angegebene Attribut nicht verändert.

Sehen wir uns ein paar Beispiele an: Erst setzen Sie das SYSTEM - Attribut für die Dateien \*.COM:

```
$SET *.COM +SYSTEM
PROG1 .COM ( 4 KB) gesetzt auf sys
PROG2 .COM ( 10 KB) gesetzt auf sys
PROG3 .COM ( 1 KB) gesetzt auf sys
```

Jetzt setzen Sie noch den Schreibschutz auf die Datei PROG2.COM und nehmen statt dessen das SYSTEM - Attribut weg:

```
$SET PROG2.COM +R/O -SYSTEM
PROG2 .COM ( 10 KB) gesetzt auf r/o
```

Und jetzt löschen Sie das SYSTEM - Attribut wieder:

```
$SET *.COM -SYSTEM
PROG1 .COM ( 4 KB) zurückgesetzt
PROG2 .COM ( 10 KB) gesetzt auf r/o
PROG3 .COM ( 1 KB) zurückgesetzt
```

```
*****
*   $SPOOL   *
*****
```

### C.19. \$SPOOL - Dateien im Hintergrund drucken

Das Dienstprogramm \$SPOOL ist wohl eines der angenehmsten Hilfsprogramme Ihres EOS - Betriebssystems. Der Zweck dieses Programms ist es, Ihre Dateien auszudrucken, während Sie normal weiterarbeiten. Das Programm selbst gehört zu den Programmen, die mit einer Kommandozeile gestartet werden. Um eine Datei im Hintergrund zu drucken, geben Sie ein:

```
$SPOOL dateiname
```

Wenn \$SPOOL die Datei starten konnte, erfolgt die Meldung

```
Datei wird ausgedruckt.
```

Wenn der der Hintergrunddruck bereits aktiv ist, erfolgt die Meldung

```
Hintergrunddruck bereits aktiv.
```

Wenn die zu druckende Datei nicht gefunden wurde, heißt es:

```
Datei nicht gefunden.
```

Damit die zu druckende Datei nicht unnötig aufbereitet werden muß, tut \$SPOOL noch weitere Dinge. Die Tabulatorstops, die sich in der Datei befinden, werden automatisch auf den zur Zeit gültigen System - Tabulatorstop erweitert. Das bedeutet, daß Sie alle acht Stellen einen Tabulatorstop haben, sofern Sie diesen Wert nicht mit Hilfe von => \$CONFIG verändert haben. Desweiteren fügt \$SPOOL alle 64 Zeilen einen Seitenvorschub ein. Diese Vorschube können allerdings durch eine Option unterdrückt werden.

Kommen wir zu den Nachteilen von \$SPOOL: durch das Bedienen des Druckers wird die Bildschirmausgabe geringfügig langsamer. Ein weiterer Nachteil ist der, daß der Drucker selbstverständlich nicht zur Verfügung steht, wenn im Hintergrund gedruckt wird. Wenn Sie den Drucker direkt ansprechen wollen, entweder durch ein Programm oder durch das Kommando "DRUCKER EIN" (siehe Abschnitt B), tut sich gar nichts; Ihre Druckausgaben werden einfach ignoriert.

Ein derartiges Druckprogramm kommt natürlich nicht ohne gewisse Steuermöglichkeiten aus. Diese Steuerung erfolgt hier durch die Angabe von folgenden Optionen:

```
*****  
*   $SPOOL   *  
*****
```

### C.19.1. Optionen

Ein paar kurze Worte zu den Optionen von \$SPOOL: bis auf die Option "VORSCHUB" sind alle ohne Dateinamen anzugeben. Falls \$SPOOL mit einer Option aufgerufen wird, ohne daß der Hintergrunddruck aktiv ist, erfolgt die Meldung

Hintergrunddruck ist nicht aktiv.

#### C.19.1.1. STOP - Druck anhalten und freigeben

Wenn Sie den Druck einmal anhalten wollen, geben Sie das Kommando

```
$SPOOL +STOP
```

\$SPOOL antwortet dann mit

Ausdruck angehalten.

Wenn Sie dann weiterzudrucken wünschen, geben Sie sinnigerweise ein:

```
$SPOOL -STOP
```

Die Antwort von \$SPOOL lautet dann

Ausdruck wird fortgesetzt.

#### C.19.1.2. RESTART - Druck neu starten

Wenn Sie es sich überlegt haben und den Druck noch einmal von vorne beginnen lassen wollen, geben Sie ein:

```
$SPOOL +RESTART
```

\$SPOOL setzt dann die Datei neu auf und antwortet mit

Ausdruck neu gestartet.

```
*****
*   $SPOOL   *
*****
```

### C.19.1.3. ABBRUCH - Druck abbrechen

Diese Option gilt als Notbremse, wenn Ihnen der Ausdruck gar nicht gefallen sollte. Sie geben ein:

```
$SPOOL +ABBRUCH
```

Der Ausdruck wird sofort abgebrochen. Das Programm quittiert dieses mit der Meldung

```
Ausdruck abgebrochen.
```

### C.19.1.4. VORSCHUB - Seitenvorschub unterdrücken

Wie schon erwähnt, besitzt \$SPOOL die Freundlichkeit, alle 64 Zeilen automatisch einen Seitenvorschub einzufügen. Dies ist meist recht hübsch anzuschauen, aber unter Umständen auch sehr störend. Wenn Sie auf die Vorschube verzichten wollen, geben Sie nach dem Dateinamen die folgende Option an:

```
$SPOOL dateiname -VORSCHUB
```

Dann werden eben keine Seitenvorschube generiert. Die Option VORSCHUB ist für Fremdsprachler auch als Option "-PAGE" implementiert.

### C.19.2. Zusätzliches für Eingeweihte

Für alle, die es genauer wissen wollen: Das Funktionsmodul \$SPOOL.RSX fängt die EOS - Systemaufrufe 1, 2, 5, 9, 10, 11 und 111 ab. Desweiteren werden die Vektoren 3, 4, 5 und 6 in der BIOS - Sprungleiste abgeändert, damit Programme, die Console I/O direkt über den BIOS - Sprungvektor machen, ebenfalls von \$SPOOL erfaßt werden. Alle EOS - Aufrufe außer dem Aufruf 10 sind in der Priorität niedriger gewichtet als der Bildschirm, was konkret heißt, daß der Drucker nur bei jeder achten Bildschirmausgabe bzw. Statusabfrage überprüft wird, um die Bildschirmausgabe nicht unnötig zu verlangsamen. Serielle Drucker sollten mit einer Geschwindigkeit von maximal 1200 Baud angeschlossen werden. Diese Geschwindigkeit bringt die geringste Einbuße an Bildschirm - Ausgabegeschwindigkeit.

```
*****
*   $TYPE   *
*****
```

## C.20. \$TYPE - Ausdrucken von Dateiinhalten

Dieses Hilfsprogramm erlaubt es Ihnen, Dateien auf dem Bildschirm anzuschauen bzw. auf dem Drucker auszugeben. Auch dieses Programm hat eine Reihe von Optionen, die Ihnen zur Hand gehen können. Der Aufruf des Programms sieht vom Interpreter für Programmierer aus wie folgt aus:

```
TYPE dateiname optionen
```

wobei "dateiname" den Namen der Datei bezeichnet, die ausgegeben werden soll. Die Angabe von Optionen ist, wie der Name schon sagt, optional. Der Dateiname muß eindeutig sein. Fehlt der Dateiname, wird dieses beanstandet mit der Meldung:

```
Welche Datei soll ausgegeben werden?
```

Auch dieses Programm hat eine Reihe von hubschen Optionen:

### C.20.1. VORSCHUB - Seitenweise Ausgabe

Dies ist der Normalfall, was bedeutet, daß immer nur eine Seite der Datei auf einmal auf dem Bildschirm angezeigt wird. Nach dem Anzeigen dieser Seite geht es nur dann weiter, wenn Sie die Leertaste gedrückt haben. Wenn Sie stattdessen eine Stoptaste drücken, wird die Ausgabe abgebrochen. Diese Option können Sie durch die Angabe von "-VORSCHUB" ausschalten. Die Ausgabe läuft dann kontinuierlich auf dem Bildschirm durch. Sie kann durch die Eingabe von Ctrl-S angehalten und dann durch die Eingabe von Ctrl-Q fortgesetzt bzw. durch die Eingabe eines Stopzeichens abgebrochen werden.

### C.20.2. SCHRITT - langsame Bildschirmausgabe

Geben Sie die Option "+SCHRITT" an, wird die Datei durchgehend ausgegeben, jedoch verlangsamt, so daß Sie Zeit genug haben, die Ausgabe an einer bestimmten Stelle anzuhalten.

### C.20.3. CONTROL - Ausgabe von Steuerzeichen

Normalerweise werden im Text enthaltene Steuerzeichen direkt ausgegeben, so daß Sie beispielsweise bestimmte Bildschirm- oder Druckersteuerzeichen in den Text mit ablegen können. Für Testzwecke ist die Ausgabe der Steuerzeichen mit Hilfe der Option "-CONTROL" abschaltbar. In diesem Falle werden Steuerzeichen durch die Zeichenfolge "^x", wobei x das entsprechende Steuerzeichen, als Buchstabe dargestellt, bezeichnet.

```
*****
*   $TYPE   *
*****
```

#### C.20.4. DRUCKER - Ausgabe auf dem Drucker

Wenn Sie eine hubsche Drucker - Ausgabe haben wollen, geben Sie als einzige Option "+DRUCKER" an. Sie erhalten dann eine Ausgabe der Datei auf dem Drucker, mit einem Seitenkopf, der den Dateinamen, das Tagesdatum und die Seite anzeigt, und Seitenvorschüben alle 64 Zeilen. Auf dem Bildschirm erscheint die Meldung:

Drücken einer beliebigen Taste bricht den Ausdruck ab...

Wenn Sie desgleichen tun, erscheint sowohl auf dem Ausdruck als auch auf dem Bildschirm die Meldung

Ausdruck abgebrochen.

und der Drucker stellt seine Tätigkeit wieder ein.

#### C.20.5. Englische Optionen

Die nachfolgenden Optionen haben allesamt eine englische Entsprechung, damit auch Benutzer, die an anderssprachige Versionen von EOS gewöhnt sind, ohne Umlernen mit der deutschen Version umgehen können:

SCHRITT	entspricht dem englischen	SLOW
DRUCKER	entspricht dem englischen	LIST
VORSCHUB	entspricht dem englischen	PAGE

Abschnitt D  
Der Grafikinterpret

```
*****
*      Grafikinterpreter      *
*****
```

## D. Der Grafik - Interpreter

Wie Sie wissen, hat Ihr Rechner die Möglichkeit, eine phantastische Farbgraphik auszugeben. Diese Grafik mußte selbstverständlich auch in EOS eingebunden werden. Nach langem Überlegen haben wir uns für zwei Möglichkeiten entschieden, die Grafik anzusprechen: die erste ist für Systemprogrammierer interessant. Sie führt über die EOS - Systemfunktion 115 direkt in den Systemkern von EOS hinein. Diese EOS - Systemfunktion ist eingehend im Abschnitt C des Systemhandbuchs beschrieben. In diesem Abschnitt soll die zweite, für jeden Programmierer durchführbare Art der Grafikprogrammierung besprochen werden.

Zu Beginn der Implementation stellte sich die Frage, wie die Grafik möglichst von jeder Programmiersprache aus anzusprechen sei. Mitgelieferte Unterprogramme scheiden aus, da sonst ein Einbinden von Assembler-routinen in das jeweilige Programm unumgänglich gewesen wäre. Da blieb nur der Weg über die Standard - Schnittstellen. Es wurde daher eine Grafik - Sprache entwickelt, die über die Bildschirmausgabe angesprochen werden kann. Nach Ausgabe eines bestimmten Bildschirm - Steuercode ist der Interpreter aktiv.

### D.1. Die Aktivierung des Grafik - Interpreters

Um diesen Interpreter anzusprechen, sind drei Wahlmöglichkeiten vorgesehen:

1. Über die Bildschirm - Schnittstelle. Wenn Sie die Zeichenfolge ESCAPE,I (großes I) ausgeben, wird die normale Bildschirmausgabe deaktiviert. Alle Ausgaben auf den Bildschirm werden auf den Grafik - Interpreter umgeleitet. Dieser Zustand endet erst mit dem Ende des Programmlaufs oder durch das Grafikkommando "END".
2. Über das Konfigurationsprogramm \$DEVICE. Der Grafikinterpreter ist als physikalisches Gerät "GRAPH" im Betriebssystem installiert. Sie haben die Möglichkeit, dieses Gerät dem Bildschirm, dem Drucker oder dem Zusatzgerät fest zuzuordnen. Alle Ausgaben, die dann auf dem jeweiligen Gerät ausgegeben werden sollen, landen im Grafikinterpreter.
3. Per Sonderkommando im SHELL - Kommandointerpreter. Dies ist mehr als Spielerei und zum Testen der Kommandos gedacht. Alle Kommandoeingaben, die an der ersten Stelle ein Sternchen "\*" enthalten, werden direkt an den Grafikinterpreter weitergeleitet. Somit können Sie interaktiv kleine Grafiken erstellen. Im SHELL - Menüprogramm ist dieses selbstverständlich nicht möglich.

```
*****
*      Grafikinterpreter      *
*****
```

## D.2. Das Befehlsformat

Jeder Grafikbefehl besteht aus einer ganz normalen Textzeile, abgeschlossen durch das Steuerzeichenpaar <CR><LF>. Das bedeutet, daß Sie diese Befehle durch ganz normale Ausgabe - Anweisungen aus Ihrem Programm heraus generieren können. Am Anfang jeder Zeile steht ein Grafikbefehl in Klartext. Dieser ist meist drei bis vier Zeichen lang. Die Schreibart ist egal, Großbuchstaben werden genau wie Kleinbuchstaben akzeptiert. Nach dem Befehlswort folgen eventuelle Parameter, durch eine Leerstelle abgesetzt. Die Parameter selbst sind durch Kommata getrennt. Leerstellen dürfen an jeder Stelle der Zeile auftreten (außer, natürlich, innerhalb eines Befehlswords oder eines Parameters).

- Jeder Parameter hat seine Grenzwerte. Werden diese überschritten, so wird zumeist der maximal mögliche Wert eingesetzt. Die Grafik hat die folgenden Grenzwerte:

```
X - Achse : 0 bis 639
Y - Achse : 0 bis 399
Farbe     : 0 bis 7
Bitmuster : 0 bis 126
```

Wenn Sie einmal einen oder mehrere Parameter weggelassen haben sollten, werden diese durch den Wert 0 ersetzt. Wundern Sie sich also nicht, wenn Sie auf dem Bildschirm gar nicht sehen, weil Sie aus Versehen in der Farbe schwarz (0) gezeichnet haben!

- Lassen wir uns die Parameter etwas näher anschauen: Die Grafik selbst bewegt sich im sogenannten zweiten Quadranten. Das bedeutet, daß sich der Koordinatenpunkt 0,0 in der linken oberen Bildschirmcke befindet. Die X - Achse wächst zur rechten Schirmseite hin, die Y - Achse zur Bildschirmunterkante hin. Eine Linie ist aus einem Vielfachen von 16 Bildpunkten zusammengesetzt. Diese bilden im Normalfall einen durchgehenden Strich. Sie können diese 16 Bildpunkte allerdings mit dem Grafikbefehl "MASK" neu definieren, so daß Sie auch strichpunktierte etc. Linien schreiben können.

Farben gibt es in acht Variationen. Hier die Liste der möglichen Farben zusammen mit ihren Farbcodes, die bei einigen Befehlen angegeben werden müssen:

```
0 - schwarz
1 - blau
2 - rot
3 - magenta
4 - grün
5 - cyan
6 - gelb
7 - weiß
```

Alle anderen Werte haben die Ausgabe der Farbe weiß zur Folge.

```
*****
*      Grafikinterpreter      *
*****
```

### D.3. Die Definition von Bit Images

Der Grafikinterpreter hat die Möglichkeit, daß Sie sich Ihren eigenen Zeichenvorrat definieren. Mit dem Befehl "TEXT" können Sie beliebige Texte auf dem Bildschirm ausgeben. Diese Texte können natürlich Ihre eigenen Zeichen enthalten. Der Befehl "FILL", der vorgegebene Bereiche mit einem Muster füllt, greift ebenfalls auf diese Zeichentabelle zurück, so daß Sie sich Ihre eigenen Füllmuster definieren können.

Jedes Zeichen ist als eine Matrix von 8 mal 8 Bits zu sehen. Mit dem Befehl "IMG" haben Sie die Möglichkeit, bis zu 8 Bytes zu je 8 Bits als Matrix zu definieren. Als Beispiel wollen wir uns mal das Bitmuster für den Buchstaben "A" ansehen:

```
Byte 1 :    0 0 0 1 1 0 0 0    = 18 Hex
Byte 2 :    0 0 1 0 0 1 0 0    = 24 Hex
Byte 3 :    0 1 0 0 0 0 1 0    = 42 Hex
Byte 4 :    0 1 1 1 1 1 1 0    = 7E Hex
Byte 5 :    0 1 0 0 0 0 1 0    = 42 Hex
Byte 6 :    0 1 0 0 0 0 1 0    = 42 Hex
Byte 7 :    0 1 0 0 0 0 1 0    = 42 Hex
Byte 8 :    0 0 0 0 0 0 0 0    = 00 Hex
```

Eine andere Art der Definition kann für das Füllen von Flächen verwendet werden:

```
Byte 1 :    1 0 0 0 1 0 0 0    = 88 Hex
Byte 2 :    0 1 0 0 0 1 0 0    = 44 Hex
Byte 3 :    0 0 1 0 0 0 1 0    = 22 Hex
Byte 4 :    0 0 0 1 0 0 0 1    = 11 Hex
Byte 5 :    1 0 0 0 1 0 0 0    = 88 Hex
Byte 6 :    0 1 0 0 0 1 0 0    = 44 Hex
Byte 7 :    0 0 1 0 0 0 1 0    = 22 Hex
Byte 8 :    0 0 0 1 0 0 0 1    = 11 Hex
```

Bei einem Füllbefehl mit diesem Muster wird die Fläche schraffiert.

```
*****
*      Grafikinterpreter      *
*****
```

Für Markierungen, Schraffuren etc. sind die Bit - Images 0 bis 12 bereits fest vordefiniert. Sie können allerdings beliebig undefiniert werden:

- 0 - 100 prozentige Füllung
- 1 - 75 prozentige Füllung
- 2 - 50 prozentige Füllung
- 3 - 25 prozentige Füllung
- 4 - 10 prozentige Füllung
- 5 - Schraffur schräg nach rechts
- 6 - Schraffur schräg nach links
- 7 - gekreuzte Schraffur
- 8 - senkrechte Striche
- 9 - waagerechte Striche
- 10 - ein stilisiertes "D&Z" (naja!)
- 11 - senkrechte Schlangenlinien
- 12 - waagerechte Schlangenlinien

#### D.4. Ein Programmierbeispiel

Ganz kurz hier nun ein Programmierbeispiel in der Sprache BASIC: Nehmen wir mal an, Sie haben 100 Werte von in einem Feld namens Y. Der Höchstwert von Y ist ebenfalls 100. Um nun einen Kurvenzug zu zeichnen, benötigen Sie das folgende Programm:

```
1 REM Umrechnungs - Funktion fuer Y:
2 DEF FNY (Y) = 400 - (4 * Y)
10 DIFF = 640 / 100 : REM Inkrement von X
20 REM Initialisieren des Interpreters:
30 PRINT CHR$(33); "I" : REM Start des Interpreters
40 PRINT "INIT" : PRINT "CRT 1,1,0,1" : REM Definition des Schirms
50 REM Positionieren des Grafik - Cursors auf Kurvenanfang:
60 PRINT "SET 0,"; FNY (Y (0))
70 REM Ausgabeschleife:
80 I = 1 : FOR X = DIFF TO 640 STEP DIFF
90 PRINT "TO " ; X ; "," ; FNY (Y (I))
100 I = I + 1 : NEXT X
110 REM Abschalten des Interpreters:
120 PRINT "END"
110 END
```

Wie die einzelnen Grafikbefehle funktionieren, lesen Sie bitte bei den jeweiligen Absätzen nach.

```
*****  
*      Grafikinterpreter      *  
*****
```

## D.5. Liste der Grafikkommandos

Einschalten der Grafik:           ESC I (Escape, großes I)

Kommandos:

INIT - Initialisieren der Grafik  
CRT - Definieren der Bildschirmausgabe  
BKGR - Hintergrundfarbe setzen  
COLR - Zeichenfarbe setzen  
MASK - Strichmaske definieren  
MODE - Zeichenmodus definieren  
IMG - neue Zeichen definieren  
CLR - Grafik löschen  
SET - Bildpunkt setzen  
RES - Bildpunkt löschen  
LINE - Linie zeichnen  
TO - Linie weiterzeichnen  
RECT - Rechteck zeichnen  
FILL - Rechteck füllen  
CIRC - Kreis zeichnen  
ARC - Kreisbogen zeichnen  
TEXT - Text ausgeben  
COPY - Hard Copy durchführen  
END - Ende der Grafik

```
*****  
*      Grafikinterpreter      *  
*****
```

## D.6. Die einzelnen Befehle

Im Nachfolgenden wird nun jeder mögliche Grafikbefehl genauestens beschrieben. Noch einige Worte vorweg: Sie können Ihre Grafikprogramme wundervoll testen, indem Sie die Zeichenfolge "ESC,I" am Anfang weglassen. So bekommen Sie die Befehle ganz normal auf den Bildschirm. Desweiteren empfiehlt es sich, die Grafikbefehle erst einmal interaktiv im Kommando - Interpreter durchzuspielen.

### D.6.1. INIT - Initialisieren der Grafik

Aufruf:     INIT

Mit diesem Befehl wird die gesamte Grafik in den Grundzustand versetzt. Sie sollten diesem Befehl am Anfang jeder Grafik - Ausgabe geben.

```
*****
*      Grafikinterpreter      *
*****
```

### D.6.2. CRT - Definition des Bildschirms

Aufruf: CRT nr,text,screen,color

Da Ihr Rechner zwei verschiedene Videoausgänge für Grafikbildschirme hat, ist es nötig, den Bildschirm etwas näher zu definieren. Es sind vier Parameter mitzuliefern. Die Parameter im Einzelnen:

- nr - Die Nummer des Bildschirms. Es sind die Zahlen 1 oder 2 zulässig. Andere Werte führen zur Ausgabe auf dem Bildschirm 1.
- text - Wenn Sie hier eine 0 eintragen, werden die normalen Textausgaben nicht mit angezeigt. Eine 1 an dieser Stelle bewirkt, daß der Bildschirm alle normalen Textausgaben weiter anzeigt.
- screen - Hier geben Sie an, ob Sie einen Farbbildschirm angeschlossen haben. Eine 0 bedeutet, daß Sie mit dem normalen grünen Bildschirm arbeiten, eine 1, daß Sie mit einem Farbbildschirm arbeiten.
- color - Hier muß ein Wert von 0 bis 7 angegeben werden. Dieser Wert bestimmt, welche Farbanteile auf dem Bildschirm ausgegeben werden sollen. Die möglichen Werte:

- 0 - nichts ausgeben
- 1 - Blauanteile ausgeben
- 2 - Rotanteile ausgeben
- 3 - Blau- und Rotanteile ausgeben
- 4 - Grünanteile ausgeben
- 5 - Grün- und Blauanteile ausgeben
- 6 - Grün- und Rotanteile ausgeben
- 7 - alles ausgeben

Andere Werte haben eine Ausgabe von allen Farbanteilen zur Folge.

```
*****
*      Grafikinterpreter      *
*****
```

### D.6.3. BKGR - Hintergrundfarbe definieren

Aufruf: BKGR n

Die Hintergrundfarbe ist von der jeweils gezeichneten Farbe vollkommen unabhängig. Sie ist rein hardwaremäßig definiert und belegt keinen Speicherplatz. N darf einen Wert zwischen 0 und 7 annehmen. Andere Werte werden auf den Wert 7 (weiß) umgesetzt.

### D.6.4. COLR - Wahl der Zeichenfarbe

Aufruf: COLR n

Die Zeichenfarbe bestimmt die Farbe fast aller folgenden Grafikkommandos. Die einzige Ausnahme bildet das "FILL" - Kommando. N darf Werte zwischen 0 und 7 annehmen. Andere Werte werden auf den Wert 7 (weiß) umgesetzt.

### D.6.5. MASK - Definition der Zeichenmaske

Aufruf: MASK bitmuster

Jede Linie, die Sie zeichnen, setzt sich bekanntlich aus Bildpunkten zusammen. Der Grafikinterpreter geht da noch etwas weiter: Jede Linie wird aus einem Vielfachen von einem Vektor zu 16 Bits zusammengesetzt. Normalerweise sind alle Bits in diesem Vektor auf 1 gesetzt, was bedeutet, daß Sie ununterbrochene Linien ziehen. Wenn Sie diesen Vektor aber undefinieren, können Sie auch gestrichelte oder punktierte Linien zeichnen. Der Vektor wird als Argument als eine Kette von Nullen und Einsen angegeben. Falls Zeichen fehlen sollten, wird der Bitvektor rechtsbündig mit führenden Nullen übergeben. Bei einem zu langen Vektor (mehr als 16 Einsen und Nullen) werden die letzten 16 Zeichen übernommen.

Schauen wir uns zwei Beispiele an: Wenn Sie eine gestrichelte Linie zeichnen wollen, geben Sie den Befehl

```
MASK 1111000011110000
```

Wenn Sie eine punktierte Linie zeichnen wollen, geben Sie ein:

```
MASK 1000100010001000
```

```
*****
*      Grafikinterpreter      *
*****
```

## D.6.6. MODE - Zeichenmodus setzen

Aufruf:       MODE n

Sie können, wenn Ihnen danach ist, auch eine andere Art des Zeichnens als die normale, überschreibende Art wählen. Sie haben die Wahl zwischen vier Möglichkeiten:

- 0 - Überschreiben. Jede Ausgabe überschreibt die alte Grafik. Dies ist der Normalmodus.
- 1 - Farbe invertieren. Es wird mit der invertierten Zeichenfarbe gezeichnet.
- 2 - Farbe löschen. Die dem eingestellten Farbcode entsprechenden Farbanteile werden gelöscht. Der Farbcode richtet sich hier nach den Farbanteilen, wie bei dem CRT - Kommando beschrieben.
- 3 - Mischen. Die eingestellte Zeichenfarbe wird den bereits vorhandenen Farbinformationen hinzugefügt.

Alle anderen Werte haben die Einstellung des Modus 0 zur Folge.

## D.6.7. IMG - Bitmuster definieren

Aufruf: IMG n,bit1,bit2,bit3,bit4,bit5,bit6,bit7,bit8

Dieser Befehl ist weiter oben schon ausführlich erläutert worden. "n" nimmt hier eine Zahl zwischen 0 und 126 ein. Andere Werte bewirken das Einsetzen der Zahl 126. Diese Zahl entspricht einem Buchstaben des ASCII - Alphabets. Wollen Sie beispielsweise den Buchstaben "A" neu definieren, so setzen Sie hier den Wert 65 ein, da "A" das 65. Zeichen des ASCII - Alphabets ist. Die Werte "bit1" bis "bit8" stellen acht Bitvektoren zu je acht Bits dar. Ein gesetztes Bit wird durch eine 1 markiert, ein gelöscht Bit durch eine 0. Fehlende Bits, wird jeder Vektor rechtsbündig übergeben. Werden mehr als 8 Bits pro Vektor übergeben, werden die letzten 8 Bits übernommen.

Als Beispiel wollen wir uns die unter D.3 beschriebene Definition des Buchstabens A ansehen:

```
IMG 65,00011000,00100100,01000010,01111110,01000010,01000010,01000010
```

Wie Sie sehen, sind hier nur 7 Bitvektoren mitgegeben worden, da der achte Vektor sowieso den Wert 0 hat.

```
*****
*   Grafikinterpreter   *
*****
```

## D.6.8. CLR - Grafik löschen

```
Aufruf:   CLR           oder
          CLR  x1,y1,x2,y2,color
```

Dieser Befehl löscht die gesamte Grafik oder Teile davon. Wenn Sie den Befehl ohne Parameter verwenden:

```
CLR
```

wird die gesamte Grafik gelöscht. Wenn Sie hingegen die Parameterliste mitliefern:

```
CLR x1,y1,x2,y2,color
```

dann passiert folgendes:  $x1,y1$  bezeichnet die linke obere Ecke eines Rechtecks,  $x2,y2$  die rechte untere Ecke. Innerhalb dieses Rechtecks werden alle Farbanteile gelöscht, die dem Wert von "color" entsprechen. "color" selbst kann folgende Werte annehmen:

- 0 - nichts löschen
- 1 - Blauanteile löschen
- 2 - Rotanteile löschen
- 3 - Blau- und Rotanteile löschen
- 4 - Grünanteile löschen
- 5 - Grün- und Blauanteile löschen
- 6 - Grün- und Rotanteile löschen
- 7 - alles löschen

Alle anderen Werte haben eine komplette Löschung der Grafik zur Folge.

## D.6.9. SET - Bildpunkt schreiben

```
Aufruf:   SET x,y
```

An dem Koordinatenpunkt  $X,Y$  wird ein Bildpunkt gesetzt.

## D.6.10. RES - Bildpunkt löschen

```
Aufruf:   RES x,y
```

An den Koordinatenpunkt  $X,Y$  wird ein Bildpunkt gelöscht.

```
*****
*      Grafikinterpreter      *
*****
```

## D.6.11. LINE - Linie zeichnen

Aufruf:     LINE  x1,y1,x2,y2

Vom Koordinatenpunkt x1,y1 bis zum Koordinatenpunkt x2,y2 wird eine Linie gezeichnet. Die Art der Linie kann vorher durch den Befehl "MASK" definiert werden.

## D.6.12. TO - Linie fortschreiben

Aufruf:     TO    x,y

Es wird eine Linie vom aktuellen Standpunkt des (unsichtbaren) Grafikcursor bis hin um Koordinatenpunkt x,y gezeichnet. Dieser Befehl dient in erster Linie dazu, Kurvenzüge zu zeichnen, wie das Programmierbeispiel zeigt.

## D.6.13. RECT - Rechteck zeichnen

Aufruf:     RECT  x1,y1,x2,y2

Es wird ein Rechteck gezeichnet, dessen linke obere Ecke durch die Koordinaten x1,y1 und dessen rechte untere Ecke durch die Koordinaten x2,y2 bestimmt wird.

## D.6.14. FILL - Rechteckfläche füllen

Aufruf:     FILL  x1,y1,x2,y2,image,color

x1,y1 bezeichnet die linke obere, x2,y2 die rechte untere Ecke einer rechteckigen Fläche. Diese wird mit einem Bitmuster gefüllt. Der Parameter "image" bezeichnet dabei eine Zahl zwischen 0 und 126, entsprechend dem ASCII - Alphabet. Sie können die Fläche mit den vordefinierten Füllmuster 0 bis 12, mit druckbaren ASCII - Zeichen mit den Werten 33 bis 126 oder aber mit einem Muster Ihrer Wahl füllen, das Sie vorher mit dem Befehl "IMG" definiert haben. Die Farbe der Füllung kann mit dem Parameter "color" bestimmt werden.

Als Beispiel füllen wir mal den halben Bildschirm mit unserem Firmenlogo (D&Z) in rot:

```
FILL 0,0,200,399,10,2
```

```
*****  
*      Grafikinterpreter      *  
*****
```

## D.6.15. CIRC - Kreis zeichnen

Aufruf: CIRC x,y,r

Es wird ein Kreis um den Mittelpunkt x,y mit dem Radius r gezeichnet.

## D.6.16. ARC - Kreisabschnitt zeichnen

Aufruf: ARC x,y,r,phi1,phi2

Es wird ein Kreisabschnitt gezeichnet. Der Mittelpunkt wird durch x,y angegeben, der Radius durch r. Phi1 und phi2 geben jeweils den Startwinkel und den Endwinkel an. Diese Winkel können Werte zwischen 0 und 359 annehmen. Null Grad entsprechen der Richtung rechter Bildschirmrand, 90 Grad der Richtung oberer Bildschirmrand.

Beispiel: Zeichnen eines Viertelkreises mit dem Mittelpunkt 100,100 und dem Durchmesser 50 im ersten Quadranten:

ARC 100,100,50,0,90

```
*****
*       Grafikinterpreter       *
*****
```

## D.6.17. TEXT - Texte ausgeben

Aufruf:       TEXT x,y,rot,size,text

Diese Funktion bietet die generelle Möglichkeit, Texte auszugeben. Diese können in Größe und Schreibrichtung beeinflußt werden. Desweiteren ist die Ausgabe in Kursivschrift möglich. Es sei noch darauf hingewiesen, daß die Größenangabe eine reine in-etwa-Angabe ist. Die absolute Größe des Textes kann je nach Schreibrichtung variieren. Und nun die Parameter im Einzelnen:

x,y - Startpunkt des Textes. Dieser Punkt bezeichnet die linke untere Ecke der 8 x 8 - Matrix des ersten Buchstabens.

rot - Schreibrichtung. Mögliche Werte sind von 0 bis 7, wobei sich die Richtung aus  $(45 * \text{rot})$  errechnet. Das bedeutet, daß eine 0 die normale Schreibweise bewirkt, bei einer 1 wird der Text schrag nach oben geschrieben, bei einer 2 senkrecht nach oben etc.

Wird auf diesen Wert der Wert 10 addiert, bewirkt dies eine Ausgabe als Kursivschrift. Werte von 10 bis 17 haben demnach den selben Effekt wie Werte von 0 bis 7, mit der Ausnahme, daß der Text kursiv erscheint.

size - Größe des Textes. Es sind Werte von 1 bis 16 zulässig. Size entspricht einem Vergrößerungsfaktor. Eine 1 entspricht dem Maßstab 1:1, eine 2 dem Maßstab 1:2 etc. Die kleinste Schrift wird demnach durch den Wert 1 erreicht, die größte durch den Wert 16.

Der Text selbst besteht aus dem gesamten Zeilenrest, der unmittelbar hinter dem Komma nach "size" folgt inklusive führender Leerstellen.

Beispiel: Ausgabe des Textes "Hallo, Du!" an den Koordinaten 100,100 auf dem Kopf stehend und in doppelter Größe:

```
TEXT 100,100,4,2,Hallo, Du!
```

```
*****  
*      Grafikinterpreter      *  
*****
```

## D.6.18. COPY - Hard Copy auf dem Tintenstrahldrucker

Aufruf: COPY color

Der Grafik - Bildschirm wird auf dem Tintenstrahldrucker IO-700 von SHARP ausgegeben. Die auszugebenden Farbanteile können als Parameter mitgegeben werden:

- 0 - nichts ausgeben
- 1 - Blauanteile ausgeben
- 2 - Rotanteile ausgeben
- 3 - Blau- und Rotanteile ausgeben
- 4 - Grünanteile ausgeben
- 5 - Grün- und Blauanteile ausgeben
- 6 - Grün- und Rotanteile ausgeben
- 7 - alles ausgeben

Alle anderen Werte haben eine komplette Ausgabe der Grafik zur Folge.

## D.6.19. END - Grafik beenden

Aufruf: END

Dieser Befehl bewirkt nur das Zurückschalten des Bildschirms aus dem Grafikmodus, falls die Grafikausgabe mit der Steuerzeichenfolge "ESC,I" gestartet wurde. Ansonsten ist er ohne Funktion. Allerdings sollte es zum guten Ton gehören, jedes Grafikprogramm mit diesem Befehl abzuschließen.