

# M40ST

**MDOS (Modular Disk Operating System)**

## **Benutzerhandbuch**

**olivetti L1**



# **M40ST**

**MDOS (Modular Disk Operating System)**

## **Benutzerhandbuch**

**olivetti L1**





## EINLEITUNG

Das vorliegende Handbuch beschreibt die Bedienung des Computersystems M40 ST und das Arbeiten mit Standardsoftware.

Für Selbstprogrammierer ist zusätzlich das "System und Programmierhandbuch MDOS/BASIC M40 ST" erforderlich.

Die nachfolgenden Ausführungen beziehen sich auf das Betriebssystem MDOS ab Releasestufe 2.0.



## INHALT

### Überblick über Kapitel

1. Das Computersystem M40 ST
2. Bedienung des Systems M40 ST
3. Arbeiten mit Anwenderprogrammen
4. Arbeitsweise des Betriebssystems MDOS
5. Bibliotheken und Files
6. Systembefehle
7. Dienstprogramme
8. Calculator- und Debugging-Mode

## INHALTSVERZEICHNIS

## SEITE

1.	<u>DAS COMPUTERSYSTEM M40 ST</u>	1.1
1.1	Überblick	1.1
1.2	Standardkonfiguration	1.2
1.2.1	Zentraleinheit	1.2
1.2.2	Tastatur	1.3
	Alphanumerische Tastatur	1.5
	Numerische Tastatur	1.8
	Funktionstastatur	1.9
	Konsole	1.13
1.2.3	Bildschirm	1.14
1.2.4	Disketteneinheiten	1.16
1.2.5	Integrierte Festplatteneinheit	1.18
1.2.6	Drucker	1.18
1.2.7	Option	1.18
1.3	Begriffsdefinition	1.19
2.	<u>BEDIENUNG DES SYSTEMS M40 ST</u>	2.1
2.1	Einschalten des Systems	2.1
2.2	Einlegen und Entnehmen von Disketten	2.2
2.3	Laden des Betriebssystems	2.4
2.3.1	Laden eines Floppy-Disk-Betriebssystems	2.4
2.3.2	Laden eines Plattenbetriebssystems	2.5
2.3.3	Meldungen nach dem Laden des Systems	2.6
2.3.3.1	Fehlermeldungen	2.6
2.3.3.2	Warnmeldungen	2.7
2.4	Datierung	2.7
2.5	Eingaben über Tastatur	2.84

## INHALTSVERZEICHNIS

## SEITE

3.	<u>ARBEITEN MIT ANWENDERPROGRAMMEN</u>	3.1
3.1	Vorbereitende Arbeiten vor der Programmausführung	3.1
3.2	Programmaufruf	3.3
3.3	Programmausführung	3.4
3.3.1	Fehlermeldungen, die unmittelbar nach einem gültigen Aufruf erfolgen können	3.4
3.3.2	Nach dem "Melden" des Programmes auftretende Fehlermeldungen	3.5
3.4	Programmunterbrechungen und Programmabbruch	3.6
3.5	Sichern von Datenbeständen	3.6
3.5.1	Betriebsbereit stellen von Disketten	3.6
3.5.2	Kopieren von Datenbeständen	3.7
3.6	Manuelles Rechnen	3.8
4.	<u>ARBEITSWEISE DES BETRIEBSSYSTEMS</u>	4.1
4.1	Systemprädisposition	4.1
4.1.1	Festlegung bestimmter Hardwarekonfigurationen	4.1
4.1.2	Festlegung von bestimmten Standardinhalten	4.2
4.2	Betriebsarten des Systems	4.3
4.2.1	Command-Mode	4.4
4.2.2	Running-Mode	4.4
4.2.3	Debugging-Mode	4.7
4.2.4	Calculator-Mode	4.7
4.3	Zustandsdiagramm	4.7

## INHALTSVERZEICHNIS

## SEITE

5.	<u>BIBLIOTHEKEN UND FILES</u>	5.1
5.1	Bibliotheken	5.1
5.5.1	Bibliotheken auf Disketten	5.1
5.1.2	Bibliotheken auf Festplatten	5.1
5.1.3	Anlegen von Bibliotheken	5.2
5.1.4	Verwaltung der Bibliotheken	5.4
5.1.5	Arbeiten mit Bibliotheken	5.5
5.1.6	Zugriff auf Bibliotheken und Files	5.6
5.2	Allgemeine Informationen über externe Files	5.7
5.2.1	Allgemeines	5.7
5.2.2	Filenamen	5.8
5.2.3	Erstellen von Files	5.8
5.2.4	Öffnen von Datenfiles	5.11
5.2.5	Schließen eines Files	5.12
5.2.6	Schützen von Files	5.12
6.	<u>SYSTEMBEFEHLE</u>	6.1
6.1	Eingabe von Befehlen über die Tastatur	6.2
6.2	Ausführung von Befehlen mit Prozeduren	6.3
6.2.1	Erstellen einer Prozedur	6.3
6.2.1.1	Eingabe von Systembefehlen und Dienstprogrammen	6.3
6.2.1.2	Eingabe von BASIC-Statements	6.3
6.2.1.3	Spezielle Befehle für Prozeduren	6.4
6.2.1.4	Prozedur als Datenfile	6.4
6.2.2	Aufruf einer Prozedur	6.4

## INHALTSVERZEICHNIS

## SEITE

6.3	Liste und Funktionen der Systembefehle	6.5
	CALCULATOR	6.9
	CATALOG	6.11
	CONFIGURE	6.15
	CREATE	6.19
	DATE	6.21
	DCHANGE	6.23
	DRAW	6.25
	ENVIRONMENT	6.27
	ERASE	6.29
	EXEC	6.31
	LBCLOSE	6.33
	LBOPEN	6.35
	LBSTORE	6.37
	LVTOC	6.39
	MODIFY	6.41
	OPTIONS	6.43
	PROCEDURE	6.45
	PURGE	6.51
	REVERSE	6.53
	SPACE	6.59
	VALIDATE	6.61
7.	<u>DIENSTPROGRAMME</u>	7.1
7.1	Allgemeine Hinweise	7.1
7.2	Liste der Dienstprogramme	7.1
	DCOPY	7.3
	DINIT	7.5
	FLCOPY	7.7
	FLPRINT	7.11
	LBCREATE	7.13
	LBEMPTY	7.15
	LBPROTECT	7.17
	LBRENAME	7.19
	LBSCRATCH	7.21
	LIBCOPY	7.23
	RESTRUCT	7.27
	VOLLAB	7.29

## INHALTSVERZEICHNIS

## SEITE

8.	<u>CALCULATOR- UND DEBUGGING-MODE</u>	8.1
8.1	Calculator-Mode	8.1
8.2	Einsetzen des Calculator-Modes	8.1
8.3	Festlegen der Art des Argumentes trigonometrischer Funktionen	8.2
8.4	Befehl zur Belegung der Funktionstasten	8.3
8.5	Lokale Variable und numerische Ausdrücke	8.4
8.5.1	Zahlendarstellung	8.5
8.5.2	Standardfunktionen	8.5
8.6	Verarbeitung der Anweisungen	8.5
8.7	Ausgabe und Ausgabeformat der Ergebnisse	8.6
8.8	Debugging-Mode	8.7
8.8.1	Erreichen und Verlassen des Debugging-Modes	8.7
8.8.2	Operationen im Debugging-Mode	8.8
	Behebbarer Fehler	8.8
	Abfrage von Variablenwerten	8.10
	Wertzuweisung an Variable	8.11
8.8.3	Rechnen im Debugging-Mode	8.12
8.8.4	START-, STOP-Befehle im Debugging-Mode	8.13



1.	<u>DAS COMPUTERSYSTEM M40 ST</u>	1.1
1.1	Überblick	1.1
1.2	Standardkonfiguration	1.2
1.2.1	Zentraleinheit	1.2
1.2.2	Tastatur	1.3
	Alphanumerische Tastatur	1.5
	Numerische Tastatur	1.8
	Funktionstastatur	1.9
	Konsole	1.13
1.2.3	Bildschirm	1.14
1.2.4	Disketteneinheiten	1.16
1.2.5	Integrierte Festplatteneinheit	1.18
1.2.6	Drucker	1.18
1.2.7	Option	1.18
1.3	Begriffsdefinition	1.19



## 1. DAS COMPUTERSYSTEM M40 ST

### 1.1 Überblick

Das System M40 ST wurde speziell für technisch-wissenschaftliche Anwendungen entwickelt. Besonderer Wert wird auf die Problemlösung des Anwenders gelegt, die nur durch die Abstimmung von Hardware und Anwendersoftware erzielt werden kann.

Das Computersystem M40 ST besteht aus folgenden Hardwarekomponenten:

- Zentraleinheit (mit integriertem Massenspeicher)
- Arbeitsplatz (bestehend aus Tastatur und grafischem Bildschirm)
- Massenspeicher (Floppy-Disk, Festplatte)
- Drucker (Thermodrucker, Nadeldrucker)
- Tisch



Abb. 1.1 - M40 ST

## 1.2 Standardkonfiguration

### 1.2.1 Zentraleinheit

Die Zentraleinheit des Systems M40 ST enthält als Kern eine CPU, die auf dem 16-bit Mikroprozessor ZILOG Z 8001 aufbaut. Die Memory Management Unit ZILOG Z 8010 sorgt für die dynamische Ansteuerung der segmentierten Speicherbereiche. Auf dem CPU-Board befindet sich außerdem eine RS 232 C-Schnittstelle (V24), die für den Anschluß eines Druckers vorgesehen ist. Der Arbeitsspeicher (RAM, Random Access Memory) hat eine Kapazität von 256 KB. Davon stehen 64 KB für Programm-Code und Daten zur Verfügung. Eine für die Datensicherheit wichtige Eigenschaft des Speichers ist der Parity-Check. Im Gehäuse der Zentraleinheit sind neben der CPU auch die Controller für die externen Datenträger (Megafloppy, Festplatte) und den Arbeitsplatz (Tastatur, grafischer Bildschirm) untergebracht.

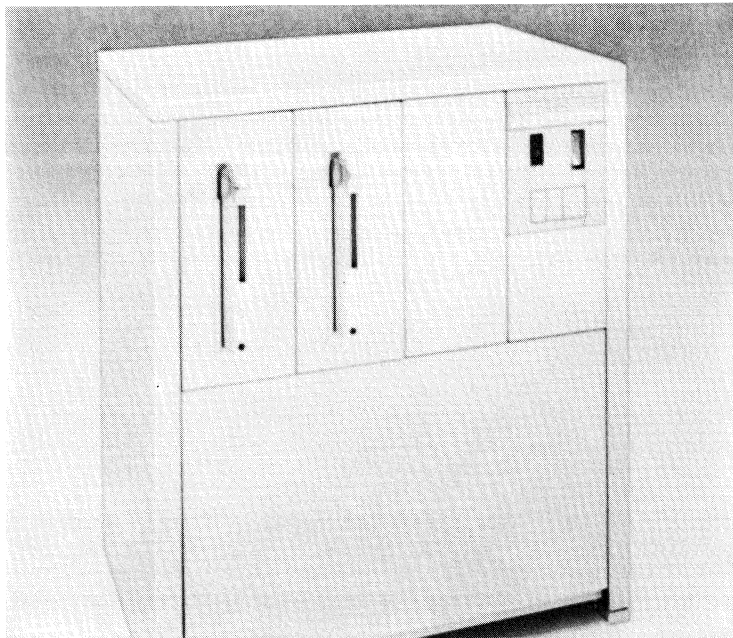


Abb. 1.2.1 – CPU-Gehäuse mit 2 Floppy-Disk-Laufwerken

### 1.2.2 Tastatur



Das System M40 ST ist mit einer superflachen elektronischen Kontaktastatur ausgestattet. Sie ist frei beweglich und in den Neigungen dreifach verstellbar.

Die alphabetischen Tasten verfügen über leichte Tastmulden und sind stufenartig unter optimalem Winkel angeordnet. Die Anschläge der Tasten ermöglichen sowohl ein akustisches als auch taktiler Feedback. Neben dem ASCII-Zeichensatz sind nationale Zeichensätze verfügbar.

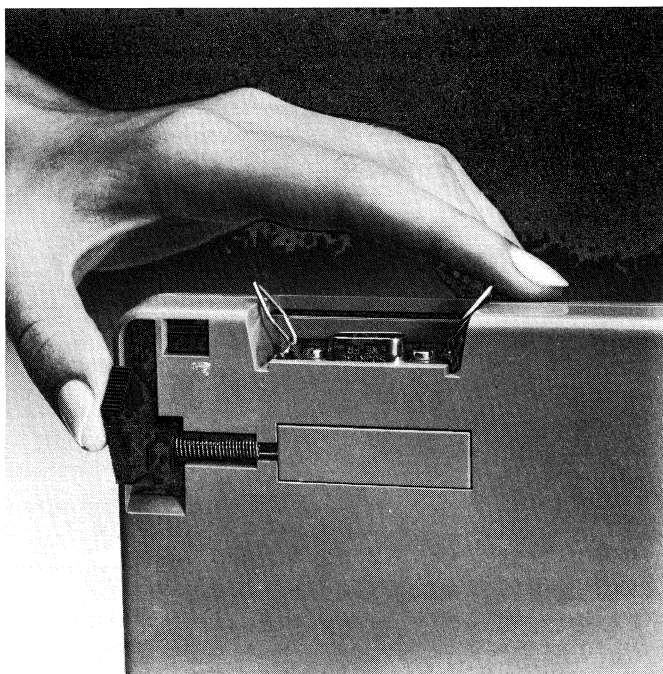
Die Tastatur ist in drei Bereiche gegliedert:

- Alphanumerisches Tastenfeld  
57 Tasten mit 26 BASIC-Ganzworten  
Groß- und Kleinschreibung
- Numerisches Tastenfeld  
19 Tasten für Ziffern und  
algebraische Operationen
- Funktionstastatur  
20 Tasten für Systembefehle und Editing  
16 frei programmierbare Funktionstasten

Vier Konsollampen zeigen den jeweiligen Systemstatus an.

Bevor die einzelnen Tastenfelder beschrieben werden, sollen einige Hinweise zur Höhenverstellung der Tastatur gegeben werden. Wie bereits erwähnt, ist die extrem flache Tastatur infolge eines beweglichen Fußes dreifach in der Höhe verstellbar. Die Höhe der H-Taste beträgt:

bei Neigung von $5^{\circ}$ :	31 mm
bei Neigung von $8^{\circ}30'$ :	35 mm
bei Neigung von $12^{\circ}$ :	38 mm



Wie in obiger Abbildung angedeutet, wird die unterschiedliche Tastaturneigung erreicht, indem die Füße, die sich an der Unterseite der Tastatur befinden, nach innen gedrückt und in die gewünschte Stellung gebracht werden. Die beiden nachfolgenden Abbildungen zeigen die Tastaturneigung in der flachen und in der steilsten Neigung.

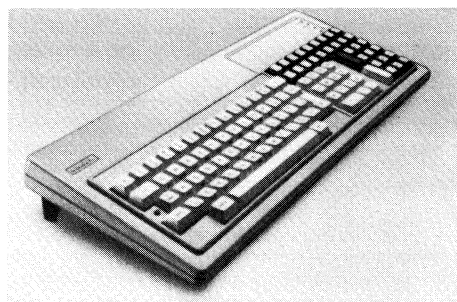
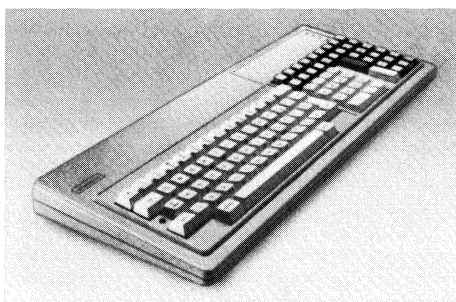
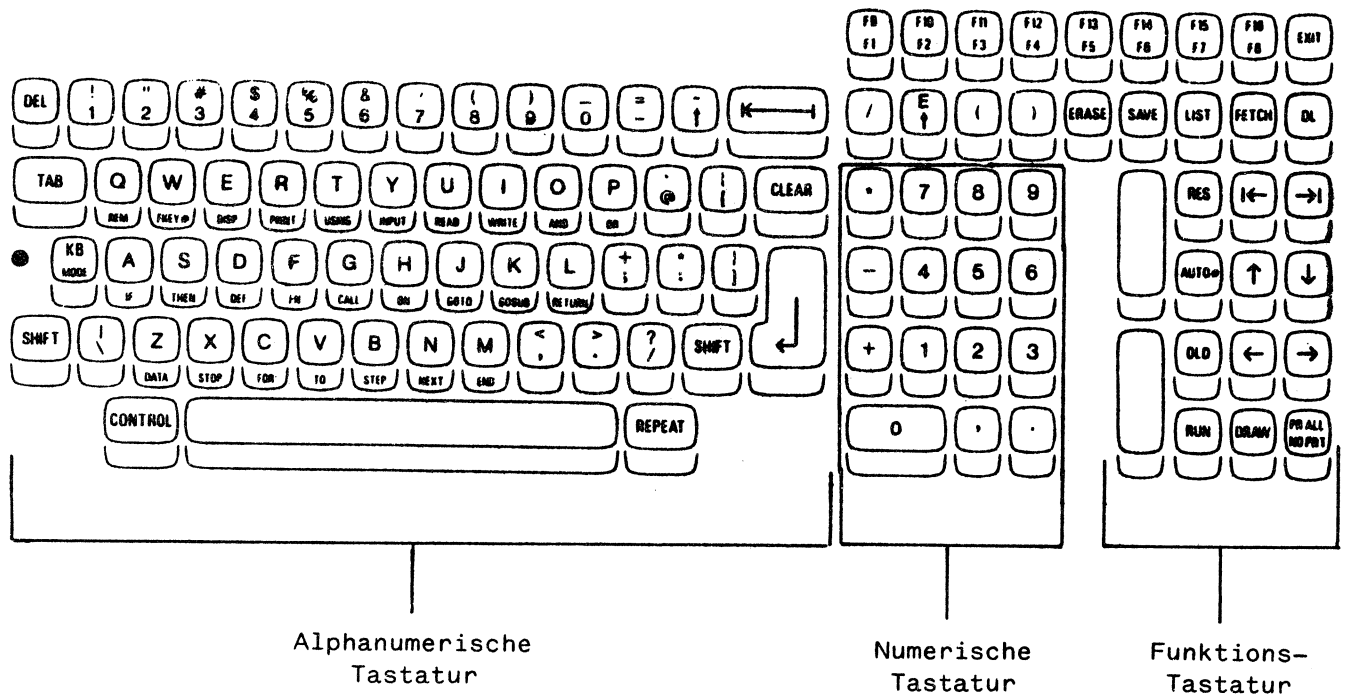


Abb. -Tastatur in flacher Position und steilster Neigung-

Die folgende Abbildung zeigt die Tastenbelegung der Gesamttastatur. Im vorliegenden Fall wird die ASCII-Version beschrieben. Für die Tastatur mit deutschem Zeichensatz gilt Analoges.

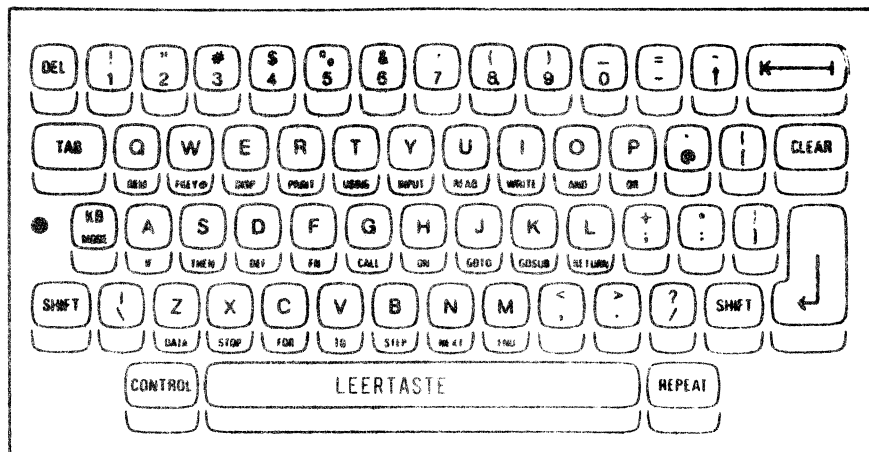


### Alphanumerische Tastatur

Die Anordnung der alphanumerischen Tastatur entspricht der einer Schreibmaschine. In der obersten Reihe sind die Ziffern und Sonderzeichen angeordnet, die untere breite Taste erzeugt ein Leerzeichen (Blank).

Die alphanumerische Tastatur ermöglicht neben Groß- und Kleinschreibung auch die Eingabe von BASIC-Worten durch einen einzigen Tastendruck.





## Alphanumerische Tastatur

Drückt man irgendeine Taste länger als 2 Sekunden, so wird der Buchstabe, der der Taste zugeordnet ist, so lange wiederholt, bis man die Taste wieder losläßt.

Neben den standardmäßigen alphabetischen Zeichen, Ziffern und Sonderzeichen verfügt die alphanumerische Tastatur noch über folgende Tasten:

**DEL**

### DELETE

Erzeugt das im Terminal-Betrieb benötigte ISO-Zeichen DEL. Im Ausdruck oder auf dem Bildschirm wird es durch das Symbol dargestellt.

**TAB**

### TAB

Für eine spätere Anwendung vorsorglich eingeplant.

**KB  
MODE**

### KEYBOARD MODE

Durch Drücken dieser Taste wird die Nutzung der Tastatur als Schreibmaschine ermöglicht, d.h. es können Groß- und Kleinbuchstaben erzeugt werden. Die SHIFT-Taste wird in diesem Falle wie bei einer normalen Schreibmaschine zur Erzeugung der Großbuchstaben benutzt.

Die Anzeige links neben der Taste leuchtet, wenn der KB-Mode eingeschaltet (aktiv) ist. Um ihn wieder auszuschalten, ist die Taste KB-Mode erneut zu drücken. Ist der KB-Mode nicht aktiv, so erhält man über SHIFT die BASIC-Worte (anstelle der Großbuchstaben).



SHIFT**SHIFT**

Wird SHIFT gleichzeitig mit einer Taste mit zwei Bedeutungen gedrückt, so wird das obere Zeichen dargestellt. Die SHIFT-Taste entspricht also der Umschalttaste einer Schreibmaschine. Auf gleiche Art erfolgt auch der Zugriff auf die BASIC-Schlüsselwörter (im KB-Mode).

CONTROL**CONTROL**

Wird CONTROL zusammen mit der Taste EXIT (obere rechte Taste der Gesamttastatur) gedrückt, so wird ein laufendes Programm, die Ausgabe aufgrund eines LIST oder CATALOG-Befehls oder eine Prozedur (siehe Systembefehl PRO) unterbrochen.

Die Taste CONTROL wird außerdem im Terminal-Betrieb und bei Anschluß von peripheren Geräten benötigt. Zusammen mit einer alphanumerischen Taste erzeugt sie die ersten 32 ISO-Zeichen (Spalte 0 und 1 der ISO-Code-Tabelle).

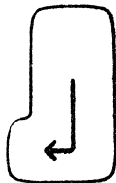
BS**BACK SPACE**

Löscht das Zeichen links vom Cursor und verschiebt etwaige Zeichen rechts davon und eine Stelle nach links.

CLEAR**CLEAR**

Löscht die Fehler- oder sonstige Meldung, die aufgrund einer falschen Eingabe oder Operation auf dem Bildschirm ausgegeben wurde.

Wird gleichzeitig SHIFT und CLEAR gedrückt, so wird sowohl die Meldung als auch die ungültige Eingabe gelöscht.

**END OF LINE oder ENTER**

Beendet eine Eingabe-Zeile oder beliebige eingegebene Daten. Im vorliegenden Handbuch wird diese Taste mit "END OF LINE"-Taste bezeichnet.

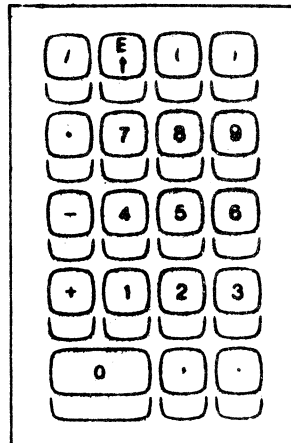
REPEAT**REPEAT**

Zusammen mit einer anderen Taste, bewirkt sie die Wiederholung des entsprechenden Zeichens, bis eine der beiden Tasten gelöscht wird.

Der alphanumerische Bereich der Tastatur verfügt über folgende Sonderzeichen:

- Ziffern: 0 bis 9
- Interpunktionszeichen: , . ; : ! ? "
- Klammern: ( ) [ ] { }
- Akzente: ' ` ^ ~
- Vergleichs- und Zuweisungszeichen: > < =
- arithmetische Operationszeichen: + - \* / ↑
- spezielle Zeichen: \$ % & \_ | \ # &

## Numerische Tastatur



Die Tasten haben folgende Bedeutung:

NUMERISCHE  
Tasten

Sie erzeugen die Dezimalziffern 0 bis 9. Der Dezimalpunkt steht bei Dezimaldarstellung einer Zahl für das Komma.



Diese Taste dient zur Eingabe von Zahlen in Exponential-Darstellung. Die nach E eingegebene Zahl wird vom System als Exponent von 10 interpretiert. Dagegen gilt bei Verwendung des Zeichens  $\uparrow$ : die auf  $\uparrow$  folgende Zahl wird als Exponent interpretiert.

Werden nacheinander die Tasten 1, 2, E und 5 gedrückt, so wird die Zahl  $12 \times 10^5$  gebildet. Durch Drücken der Tasten 1, 2,  $\uparrow$  und 5 hingegen wird die Zahl  $12^5$  generiert.

ARITHMETISCHE  
Tasten

Das System führt folgende arithmetische Rechenoperationen aus:

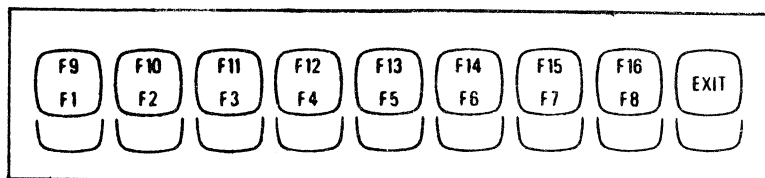
Addition  
Subtraktion  
Multiplikation  
Division  
Potenzierung  
Zuweisung





## ① Funktionstasten

Dieses Feld besteht aus den acht in der Abbildung dargestellten Tasten.



Funktionstasten

Den 16 Funktionstasten F1 bis F16 können selbstgewählte Zeichenfolgen zugeordnet werden. Jedesmal, wenn danach eine solche Taste gedrückt wird, lädt das System die entsprechende Zeichenfolge in den Tastaturbuffer. Jeder Taste sind zwei Funktionen zugeordnet. Die obere Funktion wird mit der Taste SHIFT aktiviert.

## ② Steuertaste EXIT



Durch Drücken der Taste EXIT wird ein laufendes Programm unterbrochen (Funktion STEP, Abarbeitung STEP-BY-STEP). Das System geht vom sogenannten "Runing-Mode" in den "Debugging-Mode" über (siehe Kapitel 4.3 – Betriebsarten des Systems). Dies erlaubt das Abfragen und Testen von Programmvariablen. Durch wiederholtes Drücken wird jedesmal eine weitere Programmzeile ausgeführt.

Wird EXIT im Zusammenhang mit der SHIFT-Taste gedrückt, so wird die Programmausführung fortgesetzt (Funktion CONTINUE).

Wird EXIT im Zusammenhang mit der CONTROL-Taste gedrückt, so wird – wie bereits zuvor erwähnt – die Programmausführung abgebrochen (Funktion BREAK). Das System geht in den "Command-Mode" über (siehe Kapitel 4.3 – Betriebsarten des Systems).

Bemerkung:

Die oben beschriebenen Programm-Steuerungen sind nachstehend tabellarisch zusammengefaßt:

TASTE	FUNKTION
EXIT	STEP
SHIFT EXIT	CONTINUE
CONTROL EXIT	BREAK

Tabelle: Steuertasten

Die in der Spalte 1 und 2 der Tabelle aufgeführten Begriffe werden im vorliegenden Handbuch synonym benutzt.

### ③ Systembefehle

Die Tasten für Systembefehle ermöglichen ein sofortiges Ausführen auf Tastendruck.

Nach dem Drücken erscheint der entsprechende Befehl auf dem Bildschirm. Durch Betätigen der "END-OF-LINE"-Taste wird der Befehl ausgeführt.

Die Tasten für Systembefehle sind die folgenden:







- ERASE**        löscht den Bildschirminhalt.
- SAVE**         speichert ein sich im Arbeitsspeicher befindliches Programm oder Text-File unter einem Namen in einer Bibliothek.
- LIST**         listet eine oder mehrere Zeilen eines sich im Arbeitsspeicher befindlichen Programms oder Textfiles.
- FETCH**        überträgt ein Programm oder eine Textfile-Zeile aus dem Arbeitsspeicher auf den Bildschirm zum Editieren (Verändern).
- DEL**          löscht eine oder mehrere Zeilen eines Programmes oder eines Textfiles.
- RES**          ermöglicht die Neu- bzw. Umnummerierung eines Programmes oder Textes im Arbeitsspeicher.

<b>AUTO</b>	bewirkt die automatische Vorgabe von Zeilennummern bei der Eingabe von Text- oder Programmzeilen.
<b>OLD</b>	lädt ein Programm- oder Textfile aus einer Bibliothek in den Arbeitsspeicher.
<b>RUN</b>	startet die Ausführung eines Programmes.
<b>DRAW</b>	erstellt eine Hardcopy des Bildschirminhaltes auf dem Thermodrucker PR 2400.

Eine ausführliche Beschreibung dieser Befehle finden Sie in Kapitel 6 oder im "System- und Programmierhandbuch MDOS/BASIC, M40 ST".

#### ④ Editing-Tasten

Für das Einfügen von Zeichen und Verändern von Programm- oder Textzeilen stehen die 6 folgenden Tasten zur Verfügung:

	verschiebt den Cursor an den Zeilenanfang
	verschiebt den Cursor an das Zeilenende
	Anzeige der nächstfolgenden Programm- oder Textzeile
	Anzeige der vorangehenden Programm- oder Textzeile
	verschiebt den Cursor um eine Stelle nach links
	verschiebt den Cursor um eine Stelle nach rechts

#### ⑤ Tasten für Calculator-Mode



Die blaue Taste links von der "RES"- und "AUTO"-Taste wird für das Arbeiten im Calculator-Mode benötigt. Mit Hilfe dieser Taste kann man Rechen-ergebnisse einem der Register im Calculator-Mode zuweisen (siehe Kapitel 8, Calculator- und Debugging-Mode).

#### ⑥ Steuertaste für Ausgabe



Das Drücken dieser Taste zusammen mit SHIFT bewirkt die Aktivierung der "PRINT ALL"-Funktion.

Es werden

- . alle Output-Daten (DISP, PRINT) eines Programmes,
- . alle durch Systembefehle erzeugten Listings,
- . alle Zeilen, die durch "End of Line" abgeschlossen werden und
- . alle Systemmeldungen

auf dem Systemdrucker ausgegeben.

Das Drücken der Taste ohne SHIFT bewirkt die Aktivierung der "NO PRINT"- Funktion:

Die gesamte Ausgabe auf den Systemdrucker wird unterdrückt und erscheint nur auf dem Bildschirm.

Ist "PRINT ALL" aktiv, so leuchtet die Konsollampe L1, ist "NO PRINT" aktiv, so leuchtet die Konsollampe L2.

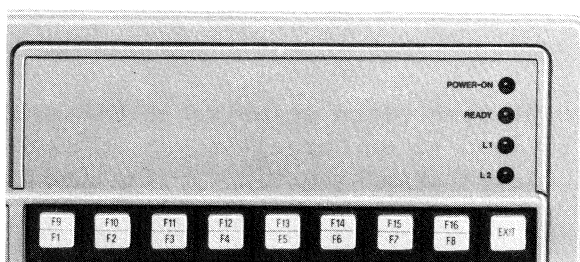
#### Bemerkung:

Sind beide Funktionen aktiv, werden die oben angeführten Arten von Ausgaben nur auf dem Bildschirm und nicht auf dem Systemdrucker ausgegeben.

Um die Aktivierung der Funktionen "PRINT ALL" bzw. "NO PRINT" aufzuheben, ist ein erneutes Drücken der Taste mit bzw. ohne SHIFT notwendig.

#### Konsole

Die Konsole befindet sich oberhalb der Tastatur. Sie besteht aus vier Signallampen:



- - POWER-ON
- - READY
- - L1
- - L2

Das Aufleuchten der Lampen hat folgende Bedeutung:

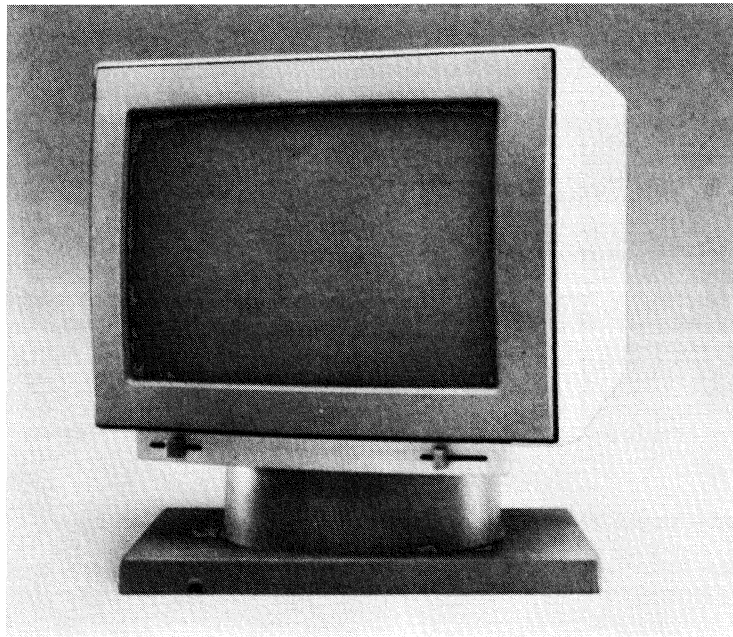
POWER-ON: Das System ist eingeschaltet.

READY: Das System ist bereit Daten anzunehmen.  
Die Signallampe erlischt, falls das System aktiv ist, d.h. wenn z.B. ein Programm läuft, ein Listing erzeugt wird u.a..

L1: Die "PRINT ALL"-Funktion ist aktiv.

L2: Die "NO PRINT"-Funktion ist aktiv.

### 1.2.3 Bildschirm



Das System verfügt über einen graphischen Bildschirm von 15 Zoll Größe und einer Kapazität von 560x392 adressierbaren Elementarpunkten.

Der Schirm ist dreh- und neigbar und verfügt über einen Helligkeits- und Kontrastregler. Der Zeichenvorrat entspricht dem der Tastatur und ist entweder ASCII oder der nationale Zeichensatz. Es werden sowohl Groß- und Kleinbuchstaben als auch Ziffern, Sonder- und Steuerzeichen angezeigt.



Alphanumerisch lassen sich

25 Zeilen à 80 Zeichen oder

41 Zeilen à 80 Zeichen

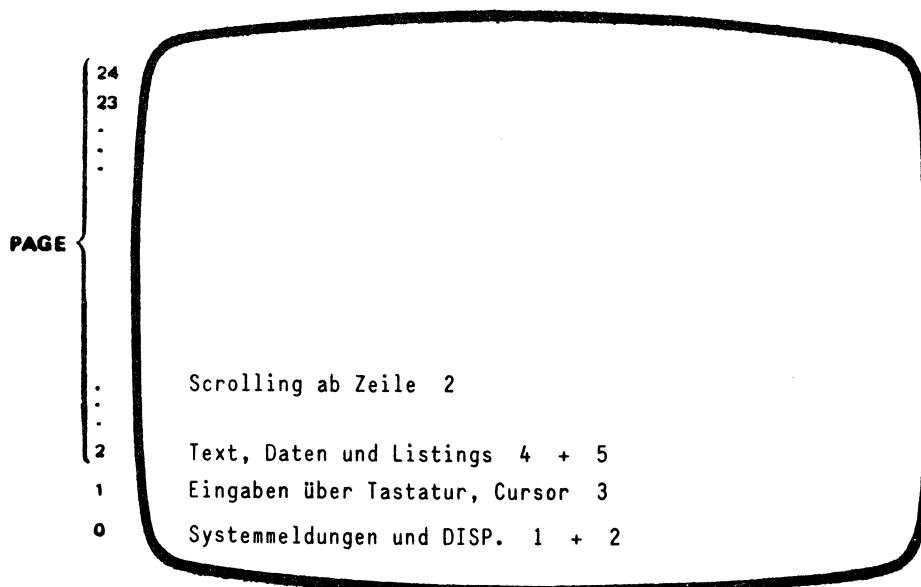
auf dem Schirm darstellen.

Programmgesteuert kann der Schirm beliebige Texte durch Blinken, doppelte Helligkeit, Umkehrung und Einrahmen bzw. Unterstreichen hervorheben.

Auf dem Bildschirm werden folgende Informationen ausgegeben:

1. Systemmeldungen (Hinweise, Fehlermeldungen)
2. Text oder Daten die durch (DISP) USING) hervorgerufen werden.
3. Eingaben über Tastatur (Eingabezeile)
4. Text oder Daten die durch PRINT (USING) oder MAT PRINT (USING) hervorgerufen werden.
5. Listing die durch Systembefehle oder Dienstprogramme erzeugt werden.

Welche Daten und Meldungen wo erscheinen, soll die nachfolgende Skizze verdeutlichen.



Bemerkung: Zeile 1 des Bildschirms wird im vorliegenden Handbuch auch Inputzeile genannt.

Zeilenaufbau im Bildschirm

### Paging

Eine sogenannte "Seite" wird durch die Zeilen 2 bis 24 gebildet. Sollen vom Programm oder System mehr als 23 Zeilen ausgegeben werden, so kann ein seitenweises Anhalten der Ausgabe auf dem Bildschirm durch Aktivierung der "NO PRINT"-Funktion erreicht werden: Nachdem 23 Zeilen ausgegeben sind, wird die Ausgabe unterbrochen. Dabei leuchtet die Signallampe "READY" auf und L2 blinkt. Zum Fortsetzen der Ausgabe ist die "END-OF-LINE"-Taste zu drücken. Dadurch werden die nächsten 23 Zeilen auf dem Bildschirm ausgegeben. Das "Paging", d.h. das Anhalten nach 23 Ausgabezeilen) kann durch nochmaliges Drücken der "NO-PRINT"-Taste aufgehoben werden.

#### Bemerkungen:

Ist das "PAGING" aktiviert und die Ausgabe wartet auf das Fortsetzen (L2 blinkt), so kann in diesem Zustand weder das Paging aufgehoben werden (Drücken von "NO PRINT") noch die Ausgabe ganz abgebrochen werden (CONTROL EXIT). Vor dem Ausführen einer dieser Funktionen muß erst die "END-OF-LINE"-Taste betätigt werden.

Ist die "PRINT-ALL"-Funktion aktiv, so werden alle in Zeile 0 und 1 des Bildschirms erscheinenden Daten oder Meldungen, auch in der Zeile 2 des Bildschirms und folgende (Scrolling), ausgegeben. Dabei verschwindet jeweils die Zeile 24 des Bildschirms.

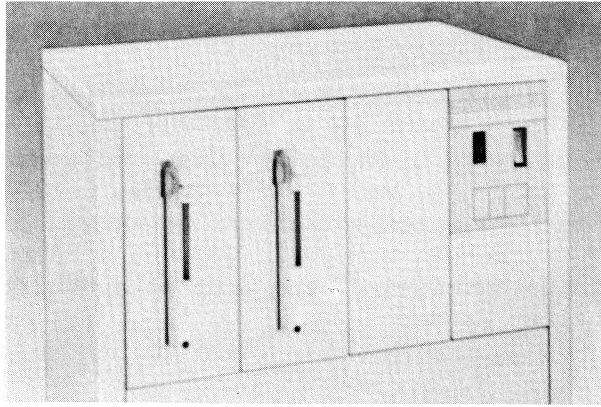
Für den Bildschirm gibt es drei Systembefehle, die es erlauben, den Inhalt des Schirms zu löschen (Befehl ERASE), eine Kopie des Schirminhaltes auf dem Drucker (PR 2400) ausgeben (Befehl DRAW) oder die Umkehrdarstellung des Bildschirminhaltes erzeugen (REVERSE).

Die Programmierung des Bildschirm in BASIC inklusive Graphik ist im "System- und Programmierhandbuch MDOS/BASIC", Druck-Nr. ... beschrieben.

### 1.2.4

#### Disketteneinheiten

Die Disketteneinheiten des Systems M40 ST sind im freistehenden CPU-Gehäuse untergebracht (siehe Abb. 1.1). Die Standardkonfiguration besitzt zwei integrierte Laufwerke.



#### Disketteneinheiten M40 ST

Die Konfiguration M40 ST/HD mit integrierter Festplatte besitzt ein Diskettenlaufwerk. Eine Kurzbeschreibung der Festplatte erfolgt im nachfolgenden Kapitel 1.2.5.

In der Standardversion mit zwei Disketteneinheiten muß während des Arbeitens mit der Anlage immer die sogenannte "Systemdiskette" eingelegt sein. Auf der Systemdiskette befindet sich das Betriebssystem, welches Mikroprogramme der Firmware und andere Basissoftware-Programme inklusive den (inkrementellen) BASIC-Compiler enthält.

Der noch freie Platz auf der Systemdiskette und der Platz auf der Diskette, die in den zweiten Schacht eingelegt wird – wir nennen sie "User-Diskette" – steht für Olivetti-Programme, Anwenderprogramme und Dateien zur Verfügung.

Der freie Platz auf der Diskette wird durch Eingabe des Befehls LVT F1 (Diskette in Laufwerk 1) oder LVT F2 (Diskette in Laufwerk 2) ausgegeben.

Man kann davon ausgehen, daß ein Buchstabe, der auf der Diskette abgespeichert wird, dort 1 Byte der Speicherkapazität belegt.

Die Diskette ist ein Speichermedium mit Direktzugriff. Bei einer bestimmten Rotationsgeschwindigkeit ist ein Leser oder Schreiber von Informationen durch einen magnetischen Schreib-/Lesekopf möglich.

Die Informationen werden auf konzentrischen Kreisen, sogenannten Spuren abgespeichert.

Jede Oberfläche der Diskette enthält 74 Spuren. Jede Spur ist in 26 Sektoren zu je 256 Bytes unterteilt.

Somit besitzt die Diskette (Megafloppy) für den Anwender eine Kapazität von 985.088 Bytes (962-KB). Neben dieser Megafloppy können auch Disketten im Standardformat 8 Zoll, 256 KB im System M40 ST verarbeitet werden. Das Format dieser Floppies entspricht genau dem Format der Disketten der Vorgängermodelle P6060 und P6066. Dadurch können Datenträger dieser Systeme auch im M40 ST verarbeitet werden.

#### 1.2.5 Integrierte Festplatteneinheit

Die Konfiguration M40 ST/HD besitzt neben einer Disketten-einheit eine in das CPU-Gehäuse (siehe Abb...) integrierte Festplatte. Sie ist dort untergebracht, wo sich bei der Standardkonfiguration das zweite Floppylaufwerk befindet.

Die Winchester-Festplatte hat eine Größe von 5 1/4 Zoll und eine Kapazität von 18 MB (unformatiert).

Bei der M40 ST/HD-Version befindet sich das Betriebssystem nicht auf der Diskette, sondern auf der Festplatte.

#### 1.2.6 Drucker

Für das Computersystem M40 ST sind eine Reihe von Druckern mit den verschiedensten Eigenschaften geplant (Graphik, Hardcopy, Schönschrift u.a.).

Das Betriebssystem MDOS unterstützt unter anderem die drei folgenden Drucker:

- |         |  |
|---------|--|
| PR 2400 | Thermodrucker, 260 Zeichen/Sek., 10 Z/Zoll, Breite 8 Zoll                      |
| PR 1450 | Nadeldrucker mit Stachelbandantrieb 10 Z/Zoll und 16,6 Z/Zoll, Breite 8 Zoll   |
| PR 1472 | Nadeldrucker mit Stachelbandantrieb 10 Z, 12 Z und 16,6 Z/Zoll, Breite 13 Zoll |

Eine Beschreibung der Drucker ist den einzelnen Bedienungsanleitungen zu entnehmen.

#### 1.2.7 Option

Optional kann das System M40 ST mit einem seriellen Interface-Controller SIC 3332 ausgerüstet werden. Dieser Controller verfügt wahlweise über zwei V24-, zwei Current-Loop- oder eine V24- und eine Current-Loop--Schnittstelle.

Das System ist mit einem oder zwei SIC 3332 ausrüstbar und kann somit maximal 4 serielle Schnittstellen unterstützen.

### 1.3

#### Begriffsdefinition

In diesem Abschnitt werden die Zeichen und Begriffe erläutert, die für die Anwendung von Systembefehlen und Dienstprogrammen relevant sind.

#### Zeichenerklärung

{ }	Einer der angeführten Parameter muß verwendet werden.
[ ]	Die angegebenen Parameter können verwendet werden oder fehlen.
<u>S</u>	Unterstrichene Parameter werden vom System eingefügt, falls keiner der Parameter angegeben wird.
.....	Der in der letzten Klammer angeführte Parameter kann beliebig oft wiederholt werden.
NAME	In diesem Schrifttyp angeführte Wörter und Buchstaben sowie alle Sonderzeichen wie: ( ) , ; * + - müssen in dieser Form eingegeben werden.
Name	Alle in dieser Schrift geschriebenen Wörter sind in den Formaten Platzhalter und werden in der Anwendung der Befehle entsprechend ersetzt.

#### Begriffserklärung

##### AKTUELLE

- LÄNGE EINES STRINGS	Länge eines Strings (einer Stringvariablen, eines Stringausdruckes) nach dem zuletzt ausgeführten Statement (i.a. ist die aktuelle Länge kleiner als die deklarierte Länge).
- DIMENSION EINER MATRIX	Dimension einer Matrix nach dem zuletzt ausgeführten MAT... Statement des Types MATREAD (m,n) etc.
- FILEGRÖSSE	Der Platz in einem Datenfile, der von Daten belegt ist

ARGUMENT	Einzusetzende Größe in einer Funktion
BIBLIOTHEK	Logische Zusammenfassung von Files unter einem Namen durch Platzzuweisung auf einer Disk. Das Inhaltsverzeichnis einer Bibliothek (Katalog) kann in maximal drei Teilbibliotheken unterteilt werden.
DELIMITER	Positive ganze Zahl zwischen 0 und 255. Das zugehörige ISO-Zeichen wird als Trennzeichen verwendet (siehe BUILD, ASSIGN...).
DISK	Der Begriff "Disk" steht allgemein für externe Datenträger (z.B. Floppy-Disk, Festplatte).
FELD	Einfach- oder doppelt indizierte numerische oder alphanumerische Variable - num. Feld:           Feld, dessen Elemente numerische Variable sind (Vektor, Matrix) - alphanum. Feld: Feld, dessen Elemente Stringvariable sind.
FILEDESIGNATOR	Positive ganze Zahl, die in einem File als Identifikationsnummer zugeordnet wird.
FILENAME	Name eines Files mit max. 6 Zeichen. Beginnt mit einem Großbuchstaben und enthält kein Sonderzeichen. Gehört das File zur Package- bzw. Common-Teilbibliothek, ist das Zeichen "*" bzw. "+" Bestandteil des Namens.
IMAGEFELD	"Druckmaske", folgende Zeichen sind möglich: #            Zeichen für Ziffer (für das Vorzeichen ist ein zusätzliches Ziffernzeichen zu reservieren) .            Dezimalpunkt \$            Dollarzeichen ↑↑↑↑        Exponentenfeld (immer 4 Zeichen) 'LLL..L     linksbündig            alphanumerische 'RRR..R     rechtsbündig           Ausgabe 'CCC..C     zentrierte Alle anderen Zeichen, die im IMAGEFELD verwendet werden, werden unverändert übernommen.

INIZIIERTE VARIABLE	Variable der Form A (I), B (I,J), C\$ (I), D\$ (I,J), I,J: numerische Ausdrücke.
KONSTANTE	
- NUM. KONSTANTE	reelle Zahl
- STRINGKONSTANTE	in Anführungszeichen eingeschlossene Folge von ISO-Zeichen
LAUFVARIABLE	numerische, nicht indizierte Variable
LIB.-NAME	Name einer Bibliothek mit max. 6 alphanumerischen Zeichen. Beginnt mit einem Großbuchstaben und enthält keine Kleinbuchstaben und Sonderzeichen (LIB.- $\hat{=}$ LIBRARY)
LIB.-REF.	Identifikation einer Bibliothek durch Angabe des Bibliotheknamens und des Namens der Einheit, die die Bibliothek enthält (LIB.-REF. $\hat{=}$ LIBRARY REFERENCE)  Lib.-Ref. kann folgende Formate annehmen: (Lib.-Name, Unit-Name) Lib-Name (,Unit-Name)
MATRIX	Zweifach indizierte Variable
NUMERISCHER AUSDRUCK	Ein aus numerischen Konstanten oder Variablen und den arithmetischen Operationen sowie Klammern, Standard- funktionen oder vom Benutzer definierten Funktionen gebildeter Ausdruck, der als Ergebnis einen numerischen Wert liefert. Erfordert die Ausführung eines Statements einen ganzzahligen Wert (z.B. TAB (X), A(U)...), wird immer gerundet.
PASSWORD	Beliebige Folge von max. 6 Zeichen außer Blank und Komma zum Schutze von Bibliotheken.
STRING	Folge von Zeichen
STRING- AUSDRUCK	Ausdruck, der aus Stringkonstanten, Stringvariablen und Standard- oder definierten Funktionen, die als Ergebnis einen String liefern, bestehen kann. Die einzige erlaubte Operation ist die Stringaddition (+).
STRINGVARIABLE	Variable, in der eine Folge von Zeichen gespeichert werden kann. Falls nicht anders definiert, beträgt die Länge 16 Zeichen (max. 1023 Zeichen).

TEILBIBLIOTHEK	Teil eines Inhaltsverzeichnisses einer Bibliothek mit folgenden Bezeichnungen: Package-, Common- und User-Teilbibliothek. Die Teilbibliotheken unterscheiden sich durch unterschiedliche Schutzmöglichkeiten.
UNIT-NAME	Logischer Name einer Diskstation (Einheit)
VEKTOR	Einfach indizierte Variable
VOLUME	Kennzeichnung einer Disk durch max.
LABEL	6 alphanumerische Zeichen. Beginnt mit einem Großbuchstaben und enthält keine Kleinbuchstaben und Sonderzeichen.



2.	<u>BEDIENUNG DES SYSTEMS M40 ST</u>	2.1
2.1	Einschalten des Systems	2.1
2.2	Einlegen und Entnehmen von Disketten	2.2
2.3	Laden des Betriebssystems	2.4
2.3.1	Laden eines Floppy-Disk-Betriebssystems	2.4
2.3.2	Laden eines Plattenbetriebssystems	2.5
2.3.3	Meldungen nach dem Laden des Systems	2.6
2.3.3.1	Fehlermeldungen	2.6
2.3.3.2	Warnmeldungen	2.7
2.4	Datierung	2.7
2.5	Eingaben über Tastatur	2.8



## 2. BEDIENUNG DES SYSTEMS M40 ST

### 2.1. Einschalten des Systems

Zum Einschalten des Systems ist der Hauptschalter OFF/ON des CPU-Gehäuses auf ON zu stellen.



Unmittelbar nach dem Einschalten wird automatisch die Funktionstüchtigkeit der Hardware überprüft. Diese sogenannte "Power-On-Autodiagnose" zeigt dem Benutzer eventuelle Störungen von Zentraleinheit, Bildschirm ,Floppy-Disk-Controller u.a. an.

Die Anzeige erfolgt über ein Statusdisplay, das sich links neben dem Hauptschalter befindet und zwar oberhalb der RESET-Taste.

Die Anzeigen bestehen aus einem Zeichen, nämlich einer der Ziffern 0, 1, ..., 9 oder einer der Buchstaben A, B, C,..., F.

Als Beispiel sind nachfolgend drei Anzeigen erläutert:

Anzeige im Statusdisplay	Bedeutung
C	Aufforderung zum Laden des Betriebssystems (bei M40 Floppy-Version)
E	eingelegte Diskette oder Festplatte enthält keine Betriebssysteme.
.	"Alles in Ordnung", Normaler Zustand

Tabelle 2.1: Anzeigen im Statusdisplay

Zum Laden des Betriebssystems siehe Abschnitt 2.3.

**Einlegen von Disketten**

1. System einschalten
2. Diskette aus der Schutzhülle (siehe Abb. 2.2.1)



Abbildung 2.2.1

3. Diskette in gewünschtes Laufwerk einführen.  
Dabei muß sich wie aus Abb. 2.2.2 ersichtlich,  
der Verschußhebel oben und der Diskettenaufkleber  
auf der linken Seite befinden.



Abbildung 2.2.2

4. Nachdem die Diskette bis zum Anschlag eingeführt wurde (sie rastet hörbar ein), ist der Verschußhebel nach unten zu schieben um das Laufwerk zu schließen. Siehe dazu Abb. 2.2.3



Abbildung 2.2.3

#### Entnehmen von Disketten

1. Verschußhebel der gewünschten Diskettenstation nach oben führen bis zum Anschlag. Die Diskette springt anschließend einige Zentimeter aus dem Laufwerk (siehe Abb. 2.2.4)

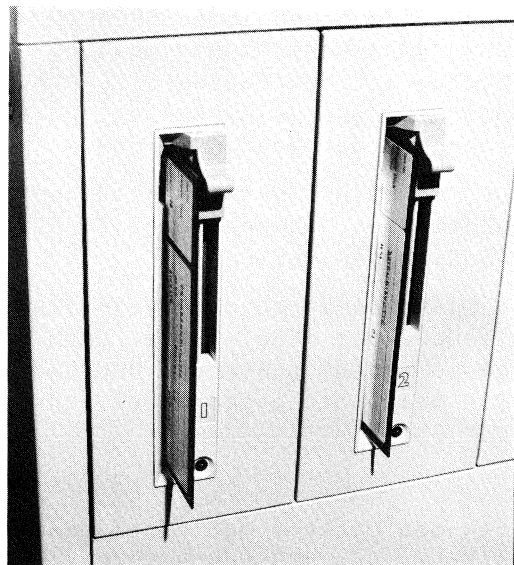


Abbildung 2.2.4

2. Diskette entnehmen und in die Schutzhülle stecken.

Achtung:

Disketten sind staub- und druckempfindlich. Daher bitte beachten:

- Disketten niemals ohne Schutzhülle aufbewahren.
- Disketten senkrecht in der mitgelieferten Schachtel lagern.
- Disketten-Etiketten vor dem Aufkleben beschriften.
- Die sichtbaren Teile der Diskette dürfen nicht berührt werden.

2.3 Laden des Betriebssystems

2.3.1 Laden eines Floppy-Disk-Betriebssystems

Wie unter Abschnitt 2.1 beschrieben, wird das System eingeschaltet. Danach erscheint im Statusdisplay die Meldung **"A"** und die Autodiagnose wird durchgeführt. Anschließend erscheint als Aufforderung zum Einlegen der Systemdiskette, sowohl im Statusdisplay als auch auf dem Bildschirm die Meldung **"C"**.

Nach der Beschreibung von Abschnitt 2.2 ist in das linke Laufwerk (Floppy F1) die Systemdiskette und (falls erforderlich) in das rechte Laufwerk (Floppy F2) eine User-Diskette einzulegen.

Nach dem Schließen der Laufwerke erscheint sofort ein Punkt **"."** im Statusdisplay und es wird ein Teil des Betriebssystems (residenter Teil) von der Systemdiskette in den Arbeitsspeicher geladen. Anschließend leuchtet die Signallampe **"READY"** auf. Auf dem Bildschirm erscheint die Meldung **"READY"** und das System ist betriebsbereit, d.h. es können z.B. sofort Programme gestartet, Dateien aufgelegt oder Programme aufgerufen werden.

Legt man im Status **"C"** keine Systemdiskette sondern z.B. eine Userdiskette ein, so erscheint im Statusdisplay und auf dem Schirm die Meldung **"E"**.

Man ersetzt diese Diskette durch die Systemdiskette und drückt anschließend die RESET-Taste. Dann wird die Autodiagnose erneut gestartet und das System geladen. Die Meldungen gehen dabei von **"A"** über **"C"** zu **"."** über:

Bemerkung:

Nach dem Anschalten des Systems muß nicht gewartet werden bis die Aufforderung **"C"** erscheint, sondern es können die System- und gegebenenfalls Userdiskette sofort eingelegt und die Diskettenschächte geschlossen werden.

### 2.3.2

#### Laden eines Plattenbetriebssystems

##### Vorbemerkung:

Das Plattenbetriebssystem wird durch Einlegen einer sogenannten "Bootstrap-Diskette" in den Arbeitsspeicher übertragen.

Die Erstellung dieser "Bootstrap-Diskette" geschieht bei der Systeminstallation und Initialisierung der Festplatte. Dabei wird mit Hilfe von zwei Disketten (Original Bootstrap- und Software-Diskette) das Plattenbetriebssystem auf die Festplatte geschrieben. Dieser Vorgang wird nur bei der Installation des Systems oder beim Übertragen einer neuen Ausgabe (Release) des Betriebssystems durchgeführt. Auf eine Beschreibung der Prozedur wird hier verzichtet.

Das Plattenbetriebssystem ist nur auf einer M40 Konfiguration die eine Festplatte enthält lauffähig. Nach dem Einschalten des Systems (M40 ST/HD) legt man sofort die "Bootstrap-Diskette" in das Diskettenlaufwerk ein und schließt den Schacht. Danach wird die Autodiagnose durchgeführt. Nach einer knappen Minute erscheint im Statusdisplay (neben dem Hauptschalter) die Meldung "." und das Betriebssystem wird von der Festplatte in den Arbeitsspeicher geladen.

Erscheint fortlaufend die Zeichenfolge "8", "3", "1" und "." im Statusdisplay, so befindet sich entweder eine falsche (d.h. keine Bootstrap-Diskette) oder keine Diskette im Schacht. Es ist dann die Bootstrap-Diskette einzulegen und die RESET-Taste zu drücken. Danach erfolgt das Laden wie oben beschrieben.

##### Bemerkungen:

1. Auf dem System M40 mit Festplatte kann auch das Floppy-Disk-Betriebssystem laufen. Dazu lädt man wie in Abschnitt 2.3.1 beschrieben.
2. Die Länge der Autodiagnose hängt von der Kapazität (Anzahl KB) des Arbeitsspeichers ab.
3. Die Bootstrap-Diskette und andere Disketten können, falls nur eine Diskettenstation vorhanden ist, mit den Dienstprogrammen EXE DCO, F1, F1 kopiert werden, wobei F1 der Name der Floppy-Station ist.

### 2.3.3 Meldungen nach dem Laden des Systems

Beim Laden des Betriebssystems können einzelne Fehler- oder Warnmeldungen ausgegeben werden. In diesem Abschnitt sind nur die wichtigsten davon aufgeführt, die oft auf falsche Disketten oder andere Bedienungsfehler zurückzuführen sind. Falls möglich, wird die Behebung der Fehlerursache angegeben.

#### 2.3.3.1 Fehlermeldungen

<u>ERROR</u>	<u>Bedeutung</u>
171 172 175 176	Über den Befehl CONFIGURE wurde eine Hardware-konfiguration definiert, die im aktuellen System nicht vorhanden ist ( siehe Fehlerliste). Es kann normal weitergearbeitet, ein neuer CON-Befehl eingeschaltet werden.
151 152 157-160	Auf einer Diskette oder Platte wurde ein Lesefehler festgestellt (siehe Fehlerliste).
154	Bei einem Platten-System mit konfigurier-ten Floppy-Laufwerken ist ein Laufwerk leer. Es kann normal weitergearbeitet oder eine zweite Diskette eingelegt werden.
156	Die Platten-Konfiguration enthält kein Betriebssystem. Auf die Festplatte ist daher ein Betriebssystem zu überspielen.
173	Es ist keine Bibliothek offen. Die erforder-lichen Bibliotheken sind mit LBO bzw. DCH zu öffnen.

Treten andere als die oben erwähnten Fehlermeldungen auf, so ist zu prüfen, ob diese nicht im Zusammenhang mit einem Autostart (siehe Kapitel 6.2.2) über die Prozedur \*SETUP stehen. Zum Beispiel kann nach dem Laden des Systems der Fehler 187 (File nicht gefunden) auftreten, wenn die das File \*SETUP beinhaltende Biblio-thek nicht offen ist (siehe auch Kapitel 3.3).



#### 2.3.3.2 Warnmeldungen

Als Warnmeldung können ausgegeben werden:

##### **ABN unit name**

Die Festplatte befindet sich noch nicht im Betriebszustand READY. Entweder nochmals aus- und einschalten oder den Betriebszustand READY abwarten, die Meldung mit CLEAR löschen und die Taste SHIFT EXIT (Funktion CONTINUE) drücken.

##### **ABN PRT**

Der angeschlossene Systemdrucker befindet sich in einem nicht regulären Zustand (z.B. Status LOCAL). LOCAL-Taste betätigen.

##### **NO LIBRARY ... ON UNIT unit name**

Das System versucht die über LBSTORE festgelegten Standardbibliotheken zu öffnen, die in obiger Meldung genannten Bibliotheken sind jedoch auf der entsprechenden Einheit nicht vorhanden.

#### 2.4 Datierung

Bei der Erstellung von Programmen bzw. Files ist es nach dem Laden des Betriebssystems sinnvoll, das aktuelle Datum mit dem Befehl DATE einzugeben. Das allgemeine Format lautet:

##### **DATE Datum**

"Datum" umfaßt 6 beliebige Zeichen, doch darf kein Leerzeichen verwendet werden.

Beispiel: 060780

falsch: AUG 80

Beim Ausdruck des Kataloges der Bibliotheken und Files wird das eingegebene Datum in 3 Gruppen zu je 2 Zeichen, getrennt durch "-", angegeben.

Beispiel: 06-06-80

Jede generierte Bibliothek bzw. jedes File wird dann mit dem jeweiligen Datum versehen. Dadurch wird das Erstellungs- bzw. Modifikationsdatum jeder Bibliothek bzw. jedes Files ersichtlich.

Das eingegebene Datum wird zudem auf der Systemdisk festgehalten und nur durch einen neuen DATE-Befehl verändert.

Beim Ändern von Programm- oder Textfiles (REPLACE) bzw. Datenfiles (MODIFY) wird das aktuelle Datum in der Spalte LAST MOD (letzte Modifikation) des Kataloges ausgegeben. In der Spalte CREATE bleibt das ursprüngliche Datum erhalten.

## 2.5 Eingaben über die Tastatur

Über die Tastatur können eingegeben werden:

- Systembefehle
- BASIC-Anweisungen
- Strings
- numerische Daten
- Textzeilen
- Berechnungen im Calculator-Mode

Die über die Tastatur eingegebenen Zeichen werden in einem Register, dem sogenannten Tastaturbuffer, abgespeichert. Dieser Buffer hat eine Kapazität von 80 Zeichen.

Jedes eingetastete Zeichen erscheint sofort im Bildschirm und zwar an der durch den Pointer bezeichneten Stelle. Der Pointer selbst wird um eine Stelle nach rechts verschoben.

Numerische Eingaben können in Fest- oder Gleitkomma (exponentielle Darstellung) erfolgen. Maximal werden 13 signifikante Stellen und, bei Gleitkommadarstellung, 2 Stellen für den Exponenten sowie negative Vorzeichen akzeptiert.

Beispiel: .99999999999999E99 / -127.5698 / -31.55E-12

Das Drücken der Taste EOL bewirkt, daß alle Zeichen außer dem Pointer vom Tastaturbuffer in den Hauptspeicher übertragen werden. Die Zeichen auf der Input-Zeile (zweitunterste Zeile) des Schirms werden gelöscht und der Pointer weist auf dessen erste Stelle. Eingaben über die Tastatur können auch überlappend mit den vom System ausgeführten Arbeiten erfolgen. Der Befehl EOL wird jedoch ignoriert und erzeugt ein akustisches Signal. END OF LINE wird erst angenommen, wenn die Signallampe READY nicht mehr leuchtet.

### Bemerkung:

Wird versucht mehr als 80 Zeichen über die Tastatur einzugeben, so wird das letzte Zeichen nicht akzeptiert, sondern ein akustisches Signal ausgegeben.

Werden zwei Zeichen gleichzeitig eingegeben, so werden beide ignoriert und es ertönt ein Signal.

### Korrektur von Eingaben über die Tastatur

Vor dem Drücken der Taste EOL können Zeichen im Tastaturbuffer (zweite Bildschirm-Zeile von unten)

- gelöscht
- modifiziert oder
- eingefügt werden.

Dazu dienen neben dem Pointer (Punkt in der Inputzeile) die in der nachfolgenden Tabelle aufgeführten Tasten der Funktionstastatur. Sie werden Editing-Tasten genannt. Der Pointer kann innerhalb der Input-Zeile auf dem Schirm auf folgende drei Arten bewegt werden:







Tasten	Wirkung
 	Versetzen des Pointers um eine Stelle nach links bzw. rechts
REPEAT  REPEAT 	Wiederholtes Versetzen des Pointers um eine Stelle nach links bzw. rechts und zwar solange bis die REPEAT-Taste nicht mehr betätigt wird.
 	Springen des Pointers an den Zeilenanfang (vor das erste Zeichen) bzw. an das Zeilenende (nach dem letzten Zeichen)

Tabelle: Bewegungen des Pointers der Input-Zeile durch Editing-Tasten.

### Löschen

- Löschen eines Zeichens:

Der Pointer muß an die Stelle unmittelbar rechts des zu löschenden Zeichens gesetzt werden. Dazu dienen die Editing-Tasten der obigen Tabelle. Anschließend ist die Taste BS (Back Space) zu drücken.

- Löschen aller Zeichen:

Unabhängig von der Position des Pointers sind die Tasten SHIFT und CLEAR zu drücken. Alle im Buffer, d.h. der Inputzeile, vorhandenen Zeichen werden dadurch gelöscht. Der Pointer weist auf die erste Stelle in der Inputzeile.

Modifizieren

- Ändern eines Zeichens:

Das zu ändernde Zeichen wird wie oben beschrieben gelöscht und das neue Zeichen eingetastet. Das eingegebene Zeichen ersetzt das gelöschte Zeichen im Buffer. Der Pointer zeigt auf die erste Stelle rechts vom neuen Zeichen.

Einfügen

- Einfügen eines Zeichens:

Der Pointer muß an der für das Einfügen des Zeichens vorgesehenen Stelle stehen. Das nun eingetastete Zeichen steht in der Inputzeile an der Stelle, an der es eingefügt wurde. Der Pointer zeigt auf die erste Stelle rechts vom neuen Zeichen. Die rechts vom eingefügten Zeichen liegenden Zeichen werden um eine Stelle nach rechts verschoben.

3.	<u>ARBEITEN MIT ANWENDERPROGRAMMEN</u>	3.1
3.1	Vorbereitende Arbeiten vor der Programmausführung	3.1
3.2	Programmaufruf	3.3
3.3	Programmausführung	3.4
3.3.1	Fehlermeldungen, die unmittelbar nach einem gültigen Aufruf erfolgen können	3.4
3.3.2	Nach dem "Melden" des Programmes auftretende Fehlermeldungen	3.5
3.4	Programmunterbrechungen und Programmabbruch	3.6
3.5	Sichern von Datenbeständen	3.6
3.5.1	Vorbereiten von Disketten	3.6
3.5.2	Kopieren von Datenbeständen	3.7
3.6	Manuelles Rechnen	3.8



### 3. ARBEITEN MIT ANWENDERPROGRAMMEN

Dieses Kapitel richtet sich vorwiegend an Anwender, die mit bestehenden Programmen arbeiten, die Programmierung jedoch nicht kennen. Es vermittelt die grundsätzlichen, zur Verarbeitung von Programmen und zur Sicherung von Datenbeständen notwendigen Kenntnisse. Dennoch wird empfohlen, die restlichen Kapitel des Handbuchs zum besseren Verständnis zu studieren.

#### 3.1 Vorbereitende Arbeiten vor der Programmausführung

Das System ist gemäß der Beschreibung im Kapitel 2 einzuschalten.

Damit ein Programm ausgeführt werden kann, müssen die das Programm und die erforderlichen Datenbestände enthaltenden Bibliotheken geöffnet werden, und zwar auf eine der folgenden Arten:

1. Automatisch, wenn die Bibliotheken mit Hilfe des Systembefehls LBS als "Standardbibliotheken" gespeichert sind.
2. Durch Ausführung einer Prozedur.
3. Durch Öffnen mit dem Systembefehl LBO.

Man beachte, daß im Normalfall Punkt 1 realisiert ist. Ansonsten werden Hinweise in den einzelnen Programmbeschreibungen gegeben.

Zu 1. Bei kontinuierlichem Arbeiten in unabhängigen Anwendungsbereichen werden die Bibliotheken meistens automatisch beim Einschalten der Maschine geöffnet. Fehlermeldungen treten in diesem Falle normalerweise nur dann auf, wenn falsche Disketten oder die Disketten in die falschen Stationen eingelegt wurden. Die einfachste Behebung besteht im korrekten Einlegen der Disketten und dem Neustart des Systems.

Zu 2. Bei ineinandergreifenden Anwendungen hängt die Folge der zu öffnenden Bibliotheken von der Entscheidung des Anwenders ab. Durch den Aufruf einer Prozedur erlaubt das System die Ausführung der dazu erforderlichen Befehle.

Der Aufruf hat folgendes Format:

**PRO name,** (Bibliotheksangabe)

Das jeweils gültige Format wird in der entsprechenden Bedienungsanleitung beschrieben.

Fehlermöglichkeiten:

- Fehlerhafte Eingaben
- falsche Disketten oder Disketten falsch eingelegt.

Bemerkung:

Auch die Prozedur ist in einer Bibliothek enthalten. Diese wird jedoch üblicherweise automatisch geöffnet.

Zu 3. Öffnen mit dem Systembefehl LBO

Ist im Programmpaket kein automatisches Öffnen der Bibliotheken vorgesehen, so hat dies mit dem Systembefehl LBO zu erfolgen.

**FORMAT: LBO** (Bibliotheksbezeichnung, Einheit)

Das korrekte Format und die erforderlichen Bezeichnungen sind den Programmbeschreibungen zu entnehmen. Der Befehl ist für alle für die entsprechenden Arbeiten erforderlichen Bibliotheken auszuführen.

Die wichtigsten beim Öffnen von Bibliotheken auftretenden Fehlermeldungen und ihre häufigsten Ursachen sind:

ERROR 196: Format der Eingaben überprüfen.

ERROR 190: Fehlerhafte Eingabe oder nach vorangehendem Kopiervorgang oder Diskettenwechsel wurde nicht mit CONTINUE (SHIFT EXIT) fortgesetzt.

ERROR 177, Diese Fehlermeldungen sind meistens auf  
215, fehlerhafte Eingabe der erforderlichen  
216, Bezeichnungen zurückzuführen. Ist diese  
217, korrekt, so können falsche Disketten oder  
218: vertauschte Disketten die Ursache sein,  
oder es wurde bei der Installation des  
Systems eine andere Bezeichnung als in  
der Programmbeschreibung gewählt.

Für alle angeführten Fehlermeldungen gilt:

Fehlermeldung durch Drücken der Taste CLEAR löschen und gegebenenfalls das Format des Befehls überprüfen.

Ist die Fehlerursache gefunden (z.B. fehlerhafte Eingabe, SHIFT EXIT vergessen), so ist sie zu beheben und END OF LINE zu drücken. Ist der Befehl jetzt korrekt, wird er ausgeführt.



Nach dem Öffnen der erforderlichen Bibliotheken kann das gewünschte Anwenderprogramm ausgeführt werden. Für die Form des Programmaufrufes gelten wieder unterschiedliche Möglichkeiten, deren wichtigste folgende sind:

1. Aufruf mit "RUN filename"

Wird das Programm mit diesem Befehl gestartet, so ist das Format der Bedienungsanleitung zu entnehmen.

2. Aufruf mit "PRO filename"

Der so gestartete Ablauf ist aus Programmen und der Ausführung von Systembefehlen kombiniert. Das Format des Aufrufes ist in der Bedienungsanleitung des Programmes beschrieben.

3. Das Programm wird nach dem Einschalten des Systems automatisch über eine Prozedur gestartet. In diesem Falle werden die erforderlichen Eingaben zur Auswahl eines bestimmten Programmes über den Bildschirm verlangt.

4. Über Funktionstasten

Der Aufruf einer in 1. oder 2. angegebenen Form wurde einer Funktionstaste zugewiesen. Das Aktivieren der entsprechenden Taste hat die gleiche Wirkung wie die direkte Eingabe des Befehls.

Die wichtigsten beim Aufruf eines Programmes auftretenden Fehlermeldungen haben meistens einfache Ursachen:

ERROR 196: Falsches Format der Eingabe.

ERROR 190: Falsches Format der Eingabe oder CONTINUE nach einem vorangehenden Kopiervorgang oder Diskettenwechsel nicht ausführen.

ERROR 173: Bibliotheken nicht geöffnet.

ERROR 187: Meistens fehlerhafte Eingabe des Programmnamens, falsche Disketten eingelegt oder mindestens eine der erforderlichen Bibliotheken nicht geöffnet.

ERROR 178: Entweder falscher Bibliotheksname oder

ERROR 180: Die Bibliothek ist noch nicht offen.

#### Bemerkung:

Beim Auftreten einer Fehlermeldung ist zunächst CLEAR zu drücken. Ist zur Behebung der Fehlerursache die Ausführung anderer Befehle notwendig, so müssen die Zeile mit SHIFT und CLEAR gelöscht und diese Befehle ausgeführt werden. Sonst genügt die Korrektur der Eingabezeile. Nach dem Drücken der Taste END OF LINE wird, falls die Fehlerursache behoben und das Eingabeformat korrekt ist, der Befehl ausgeführt.

### 3.3 Programmausführung

Ist der Aufruf des Programmes korrekt, so wird das Programm geladen und die erforderlichen Datenfiles werden gesucht. Anschließend übernimmt das Programm die Ausführungskontrolle und führt den Benutzer durch das Programm.

Maßgebend für die Ausführung des Programmes sind die im Bildschirm erscheinenden Angaben und die Erläuterungen in den Bedienungsanleitungen. In diesem Abschnitt werden einige Möglichkeiten angeführt, wie Fehlermeldungen des Programmes durch Richtigstellung der erforderlichen Bedienung vermieden bzw. behoben werden können.

#### 3.3.1 Fehlermeldungen, die unmittelbar nach einem gültigen Aufruf erfolgen können

ERROR 187: Ein verlangtes Datenfile kann nicht gefunden werden, weil:

- entweder die das File enthaltende Bibliothek nicht offen ist oder
- die erforderlichen Datenfiles nicht angegeben sind (siehe Bedienungsanleitung).

ERROR 76: Die letzte Ausführung dieses oder eines anderen, die gleichen Datenfiles verwendenden Programmes wurde nicht regulär beendet, wodurch in den Datenfiles fehlerhafte Konditionen vorliegen.

- Entweder sieht das Programmpaket ein "Notprogramm" vor (z.B. \*SOS); dann ist im allgemeinen dieses auszuführen (siehe Bedienungsanleitung)
- oder der Systembefehl VALIDATE ist auf einzelne Datenfiles anzuwenden (siehe Beschreibung in Kapitel 6).

ERROR 117, 181 Der Anwenderspeicher der Maschine ist zu klein. Falls das Programm auf dem System bereits eingesetzt wurde bzw. für die installierte Konfiguration geeignet ist, kann die Fehlerquelle bei den geladenen Options (siehe Systembefehl OPTIONS) oder einer Hardwarekonfiguration liegen, die die Erfordernisse des Programmes übersteigt (siehe Systembefehl CONFIGURE).

ERROR 63: Das Betriebssystem ist so konfiguriert, daß nur mit Programmen gearbeitet werden kann, mit denen bereits eine fehlerfreie Preexecution durchgeführt wurde. Dies geschieht mit der unter dem Systembefehl RUN oder der auf Seite 4.2 angegebenen Befehlsfolge.

### 3.3.2

#### Nach dem "Melden" des Programmes auftretende Fehlermeldungen:

Error 75: Das Programm benötigt zu seiner Ausführung OPTIONS, die nicht geladen sind (siehe Bedienungsanleitung bezüglich der erforderlichen Options und Systembefehl OPTIONS im Kapitel 6).

ERROR 65: Siehe Beschreibung von ERROR 117 und 181 unter 3.3.1.

ERROR 14: Meistens ist der angeschlossene Systemdrucker nicht eingeschaltet. Diesen einschalten und SHIFT EXIT (Funktion CONTINUE) drücken.

#### Bemerkungen:

- Die hier angeführten Fehlermeldungen geben nur einen groben Überblick und sind weder hinsichtlich ihres Auftretens noch bezüglich ihrer möglichen Ursachen vollständig beschrieben. Meistens läßt sich die Fehlerursache jedoch aufgrund der Beschreibung in der Liste der Fehlercodes und etwas Überlegung leicht finden und beheben.
- Fortsetzung des Programmes nach einer Fehlermeldung:
  - . Die Lampe READY leuchtet:  
Vor jeder weiteren Tätigkeit (außer einer Fehleranalyse, die nur unter Anleitung des Erstellers erfolgen sollte) muß CLEAR gedrückt und anschließend die Funktion BREAK ausgeführt werden.
  - . Die Lampe READY leuchtet nicht:  
Nach CLEAR kann der Fehler behoben und das Programm neu gestartet werden.

Treten Fehlermeldungen auf, deren Ursache nicht erklärbar und nicht behebbbar erscheint, so ist die Programmbeschreibung nochmals auf alle erforderlichen Bedienungselemente und Anwendungsgrenzen zu prüfen. Führt auch diese Kontrolle zu keinem Ergebnis, so sind die genaue Form der Fehlermeldung und alle bis zum Auftreten des Fehlers durchgeführten Bedienungsschritte zu notieren, bevor der Lieferant des Programmes avisiert wird.

### 3.4 Programmunterbrechungen und Programmabbruch

Soll die Ausführung eines Programmes kurzfristig unterbrochen werden, zum Beispiel um neues Papier einzulegen, so ist die Taste EXIT zu drücken. Die Ausführung kann danach mit SHIFT EXIT (CONTINUE) wieder aufgenommen werden.

Soll ein Programm vorzeitig abgebrochen werden, so ist

- falls das Programm eine End-Routine enthält, diese unbedingt auszuführen;
- das Programm durch Drücken von CONTROL EXIT (BREAK) abzubrechen.

### 3.5 Sichern von Datenbeständen

Magnetische Datenträger bilden ein zuverlässiges und preisgünstiges Speichermedium. Dennoch kann ein Datenträger durch Abnutzung oder Beschädigung zerstört werden. Demzufolge ist die Sicherung der Datenbestände durch Anlegen von Duplikaten unerlässlich, da dies die einzige Möglichkeit ist, das zuverlässige Weiterarbeiten und das Vermeiden von Stillstandzeiten und -kosten im Falle von Beschädigungen zu garantieren.

Duplikate werden durch das Kopieren der Datenbestände erstellt. Für das Computer-System stehen dazu mehrere Dienstprogramme zur Verfügung.

#### 3.5.1 Vorbereiten von Disketten

Vor der Verwendung fabrikneuer Disks oder vor ihrer Wiederverwendung sind einige Operationen durchzuführen, die im folgenden beschrieben werden.

#### 1. Vorbereiten neuer Disketten:

- Neue Disketten können als Empfängerdisketten beim Kopieren mit dem Dienstprogramm DCOPY direkt verwendet werden.
- Soll die Diskette kein Betriebssystem enthalten, kann nach Ausführung des Dienstprogrammes DINIT mit dem Dienstprogramm LBCREATE eine Bibliothek erstellt werden.
- Soll die Diskette ein Betriebssystem enthalten, so muß dieses von einer anderen Systemdiskette mit dem Dienstprogramm DCOPY kopiert werden.

#### 2. Wiederverwendung bereits gebrauchter Disketten:

Gebrauchte Disketten können beliebig oft wiederverwendet werden. Dabei gilt:

- Sie können als Empfängerdiskette beim Kopieren mit DCOPY sofort verwendet werden.
- Soll der gesamte Anwenderbereich oder Teile einer System- oder Userdiskette (Diskette ohne Betriebssystem) erneut verwendet werden, bestehen folgende Möglichkeiten:
  - . Löschen des Inhaltes der Bibliothek mit dem Dienstprogramm LBEMPTY.
  - . Löschen der Bibliothek(en) mit dem Dienstprogramm LBSCRATCH und anschließend Neuerstellen von Bibliothek(en) mit dem Dienstprogramm LBCREATE.
- Soll eine Systemdiskette in eine Userdiskette umgewandelt werden, so kann dies erreicht werden durch:
  - . Verwendung als Empfängerdiskette beim Kopieren mit DCOPY;
  - . Ausführung des Dienstprogrammes DINIT mit Parameter L, wobei die umzuwandelnde Diskette im Laufwerk 2 (F2) liegen muß. Im Laufwerk 1 (F1) muß ein gültiges Betriebssystem liegen.

### 3.5.2

#### Kopieren von Datenbeständen

##### 1. Kopieren von Disketten:

Am einfachsten und schnellsten mit dem Dienstprogramm DCOPY (siehe Kapitel 7).

## 2. Kopieren von Bibliotheken

Neben dem physischen Kopieren von Disketten (DCOPY), können auch einzelne Bibliotheken (logischer Teilbereich einer Diskette) kopiert werden. Dafür kann das Dienstprogramm LIBCOPY benutzt werden. Voraussetzung ist, daß eine Zielbibliothek (Kopie) zuvor mit LBCREATE angelegt wurde.

## 3. Kopieren von einzelnen Files

Es können einzelne Files sowohl von einer Bibliothek in eine andere als auch unter anderem File-name in dieselbe Bibliothek kopiert werden. Dafür kann das Dienstprogramm FLCOPY benutzt werden.

### Bemerkung:

Näheres zur Organisation der Datenbestände auf Diskette siehe unter Kapitel 5: Bibliotheken und Files.

## 3.6

### Manuelles Rechnen

Das System M40 ST erlaubt auch manuelles Rechnen wie mit einem Tischrechner.

Diese Möglichkeiten sind in Kapitel 8 ausführlich beschrieben.

4.	<u>ARBEITSWEISE DES BETRIEBSSYSTEMS</u>	4.1
4.1	Systemprädisposition	4.1
4.1.1	Festlegung bestimmter Hardwarekonfigurationen	4.1
4.1.2	Festlegung von bestimmten Standardinhalten	4.2
4.2	Betriebsarten des Systems	4.3
4.2.1	Command-Mode	4.4
4.2.2	Running-Mode	4.4
4.2.3	Debugging-Mode	4.7
4.2.4	Calculator-Mode	4.7
4.3	Zustandsdiagramm	4.7





#### 4. ARBEITSWEISE DES BETRIEBSSYSTEMS

##### 4.1 Systemprädisposition

Der Anwender kann jederzeit bestimmte Parameter des Betriebssystems ändern. Diese bleiben solange erhalten (sie werden automatisch auf der Systemdisk gespeichert) bis sie vom Anwender selbst geändert werden. Dies betrifft zum einen die Festlegung bestimmter Hardwarekomponenten und die softwaremäßige Festlegung bestimmter Standardinhalte des Systems.

##### 4.1.1 Festlegung bestimmter Hardwarekonfigurationen

Diese Festlegung wird durch Angabe von Parametern der Systembefehle CON und OPT durchgeführt.

##### Bestimmung der Übertragungsparameter des angeschlossenen Druckers

Das Betriebssystem hat gewisse Parameter für die Übertragung von Ausgabedaten an den Drucker gespeichert, wie z.B. Übertragungsgeschwindigkeit, ParityCheck und Anzahl Stopbits. Diese Standardwerte (auch Defaultwerte genannt) sind auf den Thermodrucker PR 2400 ausgerichtet. Sie werden im Bedarfsfall über den Systembefehl CONFIGURE, siehe Kapitel 6, Seite 6-15, eingegeben.

Als Faustregel gilt, soweit in der jeweiligen Programmbeschreibung nichts anderes ausgesagt wird:

Vor dem ersten Arbeiten mit Programmen, ist je nach angeschlossenem Drucker folgendes über Tastatur einzugeben:

Drucker	Befehl
PR 2400	CON <b>EOL</b>
PR 1450 PR 1472	CON SP=6F7YE2 <b>EOL</b>

Danach arbeitet der angeschlossene Drucker ordnungsgemäß.

### Systemmodul für graphisches Arbeiten

Um auf dem System graphische Anwendungen fahren zu können, müssen über den Systembefehl OPT (OPTION) optionale Teile des Betriebssystems geladen werden, die die Graphik steuern. Bei Olivetti-Standardsoftware sind diese Module grundsätzlich schon geladen.

Für das graphische Arbeiten gibt es zwei Möglichkeiten:

OPTION PLO	Der angeschlossene Thermodrucker wird als Plotter benutzt. Die Graphik kann länger als DIN A4 sein.
OPTION GDI (Graphikdisplay)	der Bildschirm arbeitet als graphischer Schirm. Es kann eine Hard-Copy auf dem Thermodrucker gemacht werden.

Näheres siehe unter Systembefehl OPT, Kapitel 6.

#### 4.1.2

### Festlegung von bestimmten Standardinhalten

Vom Anwender können softwareseitig z.B. das Datum oder die Funktionstasten belegt werden. Diese Daten werden auf der Systemdisk gespeichert und bleiben solange erhalten bis sie vom Programm oder dem Anwender selbst wieder geändert werden.

### Datierung

Mit dem Systembefehl DATE kann das aktuelle Datum eingegeben werden. Dieses Datum wird dann bei der Erstellung von Files und Bibliotheken gelesen und mit abgespeichert.

### Standardbelegung von Funktionstasten

Die Funktionstasten können in verschiedenen Betriebszuständen (z.B. direkt vom Anwender oder vom Programm) mit einem Inhalt belegt werden. Nach dem Ausschalten des Systems ist dieser Inhalt gelöscht. Soll der Inhalt jedoch erhalten bleiben, so kann er mit dem Systembefehl STKEYS auf der Systemdisk gespeichert werden. Bei jedem Neuladen des Systems oder bei Ausführung des Befehles LDKEYS wird dann dieser Inhalt den Funktionstasten erneut zugewiesen.

### Festlegung von Standardbibliotheken

Für das Arbeiten mit Files muß die Bibliothek, in der sie enthalten sind, geöffnet sein (siehe Kapitel 5). Damit nun beim Einschalten des Systems oder beim Neuladen des Betriebssystems die benötigten Bibliotheken nicht jedesmal neu geöffnet werden müssen, besteht die Möglichkeit, diese Bibliotheken vom System automatisch öffnen zu lassen.

Soll eine bestimmte Folge von Bibliotheken immer automatisch geöffnet werden, so sind diese einmal zu öffnen (es sei denn, sie sind schon offen) und die so eingegebenen Informationen durch Ausführung des Befehles LBSTORE auf der Systemdisk zu speichern. Die so definierten Bibliotheken können – außer beim Neuladen des Systems – auch durch Ausführung des Befehles LBRESTORE geöffnet werden.

Die Liste der automatisch zu öffnenden Bibliotheken kann durch Ausführung der Systembefehle ENVIRONMENT oder LVT\* erhalten werden.

### Automatischer Start einer Prozedur

Das System kann so festgelegt werden, daß nach dem Laden automatisch mit der Verarbeitung einer Prozedur begonnen wird (siehe Abschnitt 6.2). Dazu muß in einer automatisch geöffneten Bibliothek eine Prozedur namens \*SETUP existieren und der Systembefehl CONFIGURE mit dem Parameter PRO angegeben werden.

## 4.2

### Betriebsarten des Systems

Das Computersystem M40 ST arbeitet auf verschiedenen Systemebenen, die im folgenden Betriebsarten genannt werden.

Dieser Abschnitt zeigt diese unterschiedlichen Zustände in ihrer Bedeutung und ihren Möglichkeiten auf.

Das System M40 ST kennt folgende Betriebsarten:

- COMMAND-MODE
- RUNNING-MODE
- DEBUGGING-MODE
- CALCULATOR-MODE

Der Zusammenhang der einzelnen Betriebsarten wird am Ende dieses Kapitels in einer Graphik verdeutlicht.

#### 4.2.1 Command-Mode

Der Command-Mode erlaubt:

- die Eingabe und Ausführung von Systembefehlen (Kapitel 6),
- Den Aufruf von Dienstprogrammen (Abschnitt 6.4 und Kapitel 7),
- die Eingabe von Programmen und Texten sowie Editing-Operationen,
- den Übergang in den Calculator-Mode (Kapitel 8).

Das System ist im Command-Mode:

- nach dem Laden
- nach der Ausführung eines Anwenderprogrammes,
- nach der Eingabe von Systembefehlen im Calculator-Mode oder nach seiner expliziten Aufhebung durch Drücken der Konsoltaste,
- nach dem Drücken der Tasten CTRL EXIT (BREAK) im Running- oder Debugging-Mode.

Im Command-Mode leuchtet die Signallampe READY konstant. Nach dem Laden des Systems bzw. nach einer Rückkehr in den Command-Mode aus dem Running- oder Debugging-Mode erscheint im Schirm zusätzlich die Meldung "READY" (falls im Befehl SAVE des laufenden Programms der Parameter MSG=0 nicht angegeben wurde).

#### 4.2.2 Running-Mode

Führt das System Dienst- oder Anwenderprogramme aus, so befindet es sich im Running-Mode. Bei der Ausführung von Dienstprogrammen richten sich die Eingriffsmöglichkeiten nach der Art und den Erfordernissen des jeweils aufgerufenen Dienstprogrammes. Die folgende Beschreibung des Running-Modus befaßt sich nur mit der Ausführung von Anwenderprogrammen.

Der Aufruf zur Ausführung eines Programmes erfolgt mit einem zulässigen Format des Befehles RUN.

Wird die Ausführung eines Programmes mit RUN gestartet, so wird, falls erforderlich, eine globale Syntaxprüfung durchgeführt und mit der Verarbeitung des Programmes begonnen. Die Funktionen des Befehles RUN sind in Kapitel 6 detailliert beschrieben.

Bei der Ausführung von Anwenderprogrammen können zwischenzeitlich andere Betriebszustände erreicht oder die Programmausführung durch ein reguläres oder durch Fehler verursachtes Programmende abgebrochen werden.

Möglichkeiten des Verlassens der Programmausführung:

#### Rückkehr in den Command-Mode:

- Wird die Ausführung eines Programmes durch Erreichen der END-Anweisung regulär beendet, so erfolgt die Rückkehr in den Command-Mode.
- Nach dem Drücken der Tasten CTRL EXIT (BREAK) wird die Verarbeitung des laufenden Programms nach regulärer Beendigung des sich in Verarbeitung befindlichen Statements abgebrochen, eventuell geöffnete Files geschlossen und das Programm für Editing-Operationen wieder zur Verfügung gestellt.

#### Übergang in den Debugging-Mode

Der Übergang aus dem Running-Mode in den Debugging-Mode erfolgt:

- durch Drücken der Taste EXIT (STEP),
- durch einen STOP-Befehl im Programm,
- durch Ausführung eines im Debugging-Mode eingegebenen STOP-Befehles,
- durch einen behebbaren Fehler
- nach nicht behebbaren Fehlern ohne Möglichkeit der Programmfortsetzung.

Wird während der Ausführung eines Programmes die Taste EXIT (STEP) gedrückt, so wird die Programmverarbeitung am Ende der laufenden Operation unterbrochen. Danach stehen alle Möglichkeiten des Debugging-Modus zur Verfügung (siehe Kapitel 8). Tritt bei der Verarbeitung eines Programmes ein Fehler auf, so meldet das System den Fehlercode und bleibt im Debugging-Mode stehen.

Mit den im Debugging-Mode möglichen Operationen können die Ergebnisse des den Fehler verursachenden Statements erzeugt, die Werte aller im Programm aufgeführten Variablen abgefragt, Rechenoperationen ausgeführt und die Verarbeitung mit dem nächsten oder einem beliebigen anderen Befehl fortgesetzt werden. Die Rückkehr in den Running-Mode erfolgt entweder mit den Tasten SHIFT EXIT (CONTINUE) oder mit dem Befehl START. Die Anweisung "START Zeilennummer" wird vom System nicht akzeptiert, wenn der Programmstart mit "RUN program name" und die Abspeicherung des Programms ohne vorherige Preexecution (d.h. vollständige Syntaxprüfung) erfolgte (siehe Beschreibung der Befehle RUN und PREPARE in Kapitel 6).

Durch das Drücken der Taste EXIT (STEP) wird jeweils eine BASIC-Anweisung ausgeführt und dann in den Debugging-Mode zurückgekehrt.

### Inputanforderung

Tritt bei der Programmausführung eine der Anweisungen RKB, INPUT, MAT INPUT oder RECEIVE #n auf ("n" = Adresse des I/O-Kanales Tastatur), so wartet das System auf die Eingabe der Daten in richtiger Anzahl und korrektem Format.

- Nach Abschluß der Eingabe kehrt das System in den Running-Mode zurück.
- Werden die Tasten CTRL EXIT (BREAK) gedrückt, während das System auf eine Eingabe wartet, wird der Programmablauf abgebrochen.
- Wird die Taste EXIT (STEP) gedrückt, so geht das System in den Debugging-Mode. Nach Drücken der Tasten SHIFT EXIT (CONTINUE) oder erneutem Drücken der Taste EXIT (STEP), erscheint im Bildschirm ein Fragezeichen; die Eingabe kann nun erfolgen. Vor der Dateneingabe können zum Beispiel Zwischenrechnungen manuell durchgeführt werden.

Wartet das System auf eine Eingabe, so hört die Signallampe READY mit Blinken auf und leuchtet konstant.

Erfolgt die Programmausführung innerhalb einer Prozedur mit automatischer Inputanforderung, so werden die Eingabedaten bei der Inputanforderung dem Commandfile entnommen und das Programm ohne Unterbrechung fortgesetzt.

### Operator Call

Einzelne Dienstprogramme verlangen bestimmte Interventionen des Benutzers, wie etwa den Wechsel von Disks.

Die Aufforderung zum Wechseln einer Disk erfolgt im Bildschirm. Nach Durchführung des Disktausches wird die Verarbeitung des Dienstprogrammes durch Drücken der Taste EXIT (CONTINUE) fortgesetzt.

Das Drücken der Tasten CTRL EXIT (BREAK) wird nur akzeptiert, wenn eine arbeitsfähige Konstellation vorhanden ist.

### Behebbarer Fehler

Tritt während eines Programmlaufes ein behebbarer Fehler auf (z.B. fehlende Wertzuweisung), so geht das System unter Meldung des Fehlercodes in den Debugging-Mode. Eine Abfrage von Variablen und gegebenenfalls eine Wertzuweisung sind möglich. Der Programmablauf kann durch Drücken der Taste EXIT (CONTINUE) fortgesetzt werden.

### Nicht behebbarer Fehler in der Ausführungsphase

(Fehlercodes 65 – 97, 162 – 170)

Tritt während eines Programmlaufes ein nicht behebbarer Fehler auf (z.B. RETURN ohne GOSUB), so geht das System unter Meldung des Fehlercodes in einen Quasi-Debugging-Mode, das heißt, eine Abfrage von Variablen ist möglich, der Programmablauf kann aber nicht fortgesetzt werden. Die Tasten CTRL EXIT (BREAK) müssen gedrückt werden, wonach das System in den Command-Mode übergeht.

#### 4.2.3 Debugging-Mode

Das System befindet sich im Debugging-Mode:

- nach Drücken der Taste EXIT (STEP),
- nach Ausführung der Anweisung PREPARE,
- bei einer STOP-Anweisung im Programmablauf,
- nach Meldung eines behebbarer Fehlers,
- nach Auftreten eines nicht behebbarer Fehlers.

Der Debugging-Mode erlaubt das Austesten eines Programmes und die Behebung von Fehlern. Die Arbeitsweise wird in Kapitel 8 beschrieben.

#### 4.2.4 Calculator-Mode

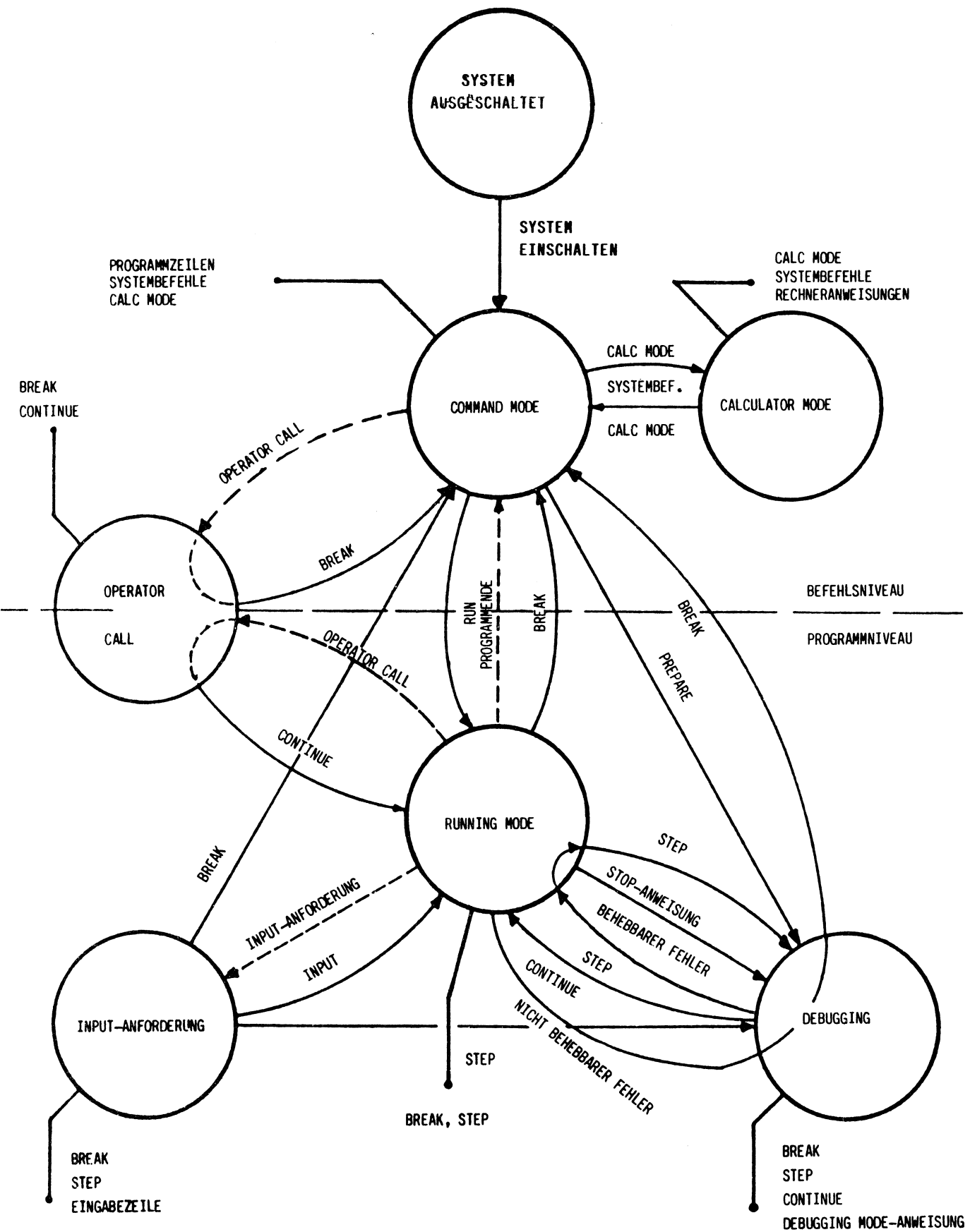
Durch Aktivierung des Calculator-Modes mit Eingabe von CAL, ist das System als Tischrechner zu verwenden. Der Debugging-Mode enthält auch den Calculator-Mode. Die Beschreibung der Möglichkeiten im Calculator-Mode erfolgt in Kapitel 8.

#### 4.3 Zustandsdiagramm

Zum besseren Verständnis des folgenden Zustandsdiagrammes des Betriebssystems beachte man folgende Analogie:

Funktion	Tastendruck
BREAK	CONTROL u. EXIT
STEP	SHIFT u. EXIT
CONTINUE	EXIT

### Zustandsdiagramm





5.	<u>BIBLIOTHEKEN UND FILES</u>	5.1
5.1	Bibliotheken	5.1
5.5.1	Bibliotheken auf Disketten	5.1
5.1.2	Bibliotheken auf Festplatten	5.1
5.1.3	Anlegen von Bibliotheken	5.2
5.1.4	Verwaltung der Bibliotheken	5.4
5.1.5	Arbeiten mit Bibliotheken	5.5
5.1.6	Zugriff auf Bibliotheken und Files	5.6
5.2	Allgemeine Informationen über externe Files	5.7
5.2.1	Allgemeines	5.7
5.2.2	Filenamen	5.8
5.2.3	Erstellen von Files	5.8
5.2.4	Öffnen von Datenfiles	5.11
5.2.5	Schließen eines Files	5.12
5.2.6	Schützen von Files	5.12



## 5. Bibliotheken und Files

Für die permanente Speicherung von Programmen und Datenbeständen stehen bezüglich Charakteristik und Kapazität unterschiedliche Speichermedien zur Verfügung, nämlich:

- zwei integrierte Diskettenstationen oder
- eine integrierte Diskettenstation und eine integrierte Festplatteneinheit.

Um ein zuverlässiges und rasches Wiederauffinden gespeicherter Datenbestände zu erreichen, ist eine logische Organisation dieser Speichermedien erforderlich. Diese Organisation erfolgt in mehreren Stufen:

- Die Datenbestände werden mit einem Namen versehen (Filename).
- Für die Filenamen werden Inhaltsverzeichnisse angelegt (Teilbibliotheken).
- Die Files werden auf spezifizierten Speichermedien an bestimmten Stellen der Bibliotheken gespeichert, in denen auch die Inhaltsverzeichnisse verwaltet werden.

### 5.1 Bibliotheken

Eine Bibliothek ist ein reservierter, mit einem Namen versehener Speicherbereich auf einem Speichermedium.

#### 5.1.1 Bibliotheken auf Disketten

Eine Standard-Diskette mit 256 KB kann nur eine einzige, dem Anwender zugängliche Bibliothek enthalten. Eine Megafloppy kann zwischen einer und maximal 14 Bibliotheken umfassen.

#### 5.1.2 Bibliotheken auf Festplatte

Die integrierte Festplatte kann in mehrere "logische Platten", sogenannte Daten-Sets eingeteilt werden. Jedes Daten-Set kann mehrere Bibliotheken umfassen.

Die Summe der Bibliotheksgrößen ist durch die Kapazität des Daten-Sets beschränkt. Die Summe der Größen der Daten-Sets ist durch die Kapazität der Festplatte beschränkt.

### 5.1.3 Anlegen von Bibliotheken

Bibliotheken dienen zur Verwaltung von Files. Allen Files einer Bibliothek steht ein gemeinsamer Speicherbereich zur Verfügung. Die Filenamen aller Files einer Bibliothek werden in bis zu drei getrennten Inhaltsverzeichnissen verwaltet.

Die Inhaltsverzeichnisse einer Bibliothek werden "Teilbibliotheken" genannt. Sie unterscheiden sich bezüglich des für sie möglichen Schutzes. Folgende Arten von Teilbibliotheken sind vorhanden:

- Package-Teilbibliothek
- Common-Teilbibliothek,
- User-Teilbibliothek.

Der Schutz einer Teilbibliothek wird durch die Ausführung des Dienstprogrammes LBPROTECT erreicht und bezieht sich immer auf die Gesamtheit der in ihr enthaltenen Files. Der Schutz einzelner Files vor Editing kann über den Systembefehl SECURE erreicht werden.

Für das Anlegen einer Bibliothek sind folgende Angaben erforderlich:

1. Vergabe eines Bibliotheksnamens,
2. Festlegung der Einheit, die die Bibliothek aufnehmen soll,
3. Eventuell Vergabe eines Passwords,
4. Festlegung der Größe der Bibliothek,
5. Festlegung der Teilbibliotheken dieser Bibliothek und der Anzahl der in jede Teilbibliothek aufnehmbaren Filenamen.

#### zu 1. Vergabe eines Bibliotheksnamens

Ein Bibliotheksname kann aus ein bis sechs Großbuchstaben oder Ziffern bestehen. Das erste Zeichen muß ein Buchstabe sein. Bibliotheken, die auf derselben Einheit gespeichert sind, müssen verschiedene Namen tragen.

Beispiele für gültige Bibliotheksnamen:

- BIBL1
- GEO
- ARCHIV

zu 2. Festlegung der Einheit, auf welcher die Bibliothek angelegt wird.

Die Festlegung der für die Speicherung der Bibliothek vorgesehenen Einheit erfolgt durch Angabe des Namens "unit name" dieser Einheit.

Bibliotheksname und "unit name" bilden gemeinsam die "lib ref", die im weiteren die Identifikation der Bibliothek ermöglicht. Das vollständige Format von "lib ref" lautet:

(Bibliotheksname, unit name)

Sie kann auch andere Formen annehmen, deren Format und Gültigkeit in der Beschreibung der Systembe-  
fehle und Dienstprogramme jeweils angeführt ist.

zu 3. Vergabe eines Passwords

Bibliotheken können mit einem Password versehen werden. Der Zugriff auf Files einer so geschützten Bibliothek ist nur möglich, wenn das Password angegeben wird. Ohne Kenntnis ihres Passwords kann mit einer Bibliothek nicht gearbeitet werden.

Wurde bei der Installation des Betriebssystems in einer Plattenkonfiguration ein Systempassword vergeben, so ist hat dieses anstelle des Bibliotheks-  
passwords ebenfalls Gültigkeit.

Als "password" ist eine beliebige Folge von maximal sechs Zeichen möglich. Wird kein Password vergeben, so ist der Zugriff auf die Files dieser Bibliothek ohne Angabe eines Passwords möglich.

zu 4. Festlegung der Größe der Bibliothek

Die gewünschte Größe der Bibliothek ist in K-Bytes anzugeben. Dabei ist zu berücksichtigen, daß der für Inhaltsverzeichnisse und Leerplatzverwaltung erforderliche Platzbedarf bei der Angabe der Bibliotheksgröße mit einbezogen werden muß. Wird für die Bibliothek keine Größe angegeben, so wird vom System der Default-Wert genommen. Vergl. dazu das Dienstprogramm LBC (Library Create) in Kapitel 7.

## zu 5. Festlegung der Teilbibliotheken

Jede Bibliothek kann in bis zu drei Teilbibliotheken aufgeteilt werden. Jede Teilbibliothek besteht aus dem Inhaltsverzeichnis der in ihr enthaltenen Files.

Die Definition der Größe der Teilbibliothek erfolgt durch Festlegung der Anzahl für das Inhaltsverzeichnis zur Verfügung gestellten Sektoren.

Jeder Sektor kann die Informationen von 27 Files aufnehmen.

Erfolgt keine explizite Festlegung der Größe der Teilbibliotheken, so werden vom Betriebssystem Standardannahmen getroffen:

- Bei Bibliotheken auf Standard-Diskette (256 KB)
  - für Package-Teilbibliothek 5 Sektoren (65 Files),
  - für Common-Teilbibliothek 4 Sektoren (52 Files),
  - Für User-Teilbibliothek 5 Sektoren (65 Files).
- Bei Bibliotheken auf Megafloppies und Festplatten:
  - für Package-Teilbibliothek 3 Sektoren (81 Files),
  - für Common-Teilbibliothek 2 Sektoren (54 Files),
  - für User-Teilbibliothek 3 Sektoren (81 Files).

Die Kapazität einer Bibliothek hängt somit von zwei Vereinbarungen ab:

- Die Summe der einzelnen Filegrößen darf mit dem für die Verwaltung erforderlichen Platz die Gesamtkapazität nicht übersteigen.
- Die Anzahl der in einer Bibliothek speicherbaren Files ist begrenzt durch die Anzahl der für Teilbibliotheken vereinbarten Sektoren.

Das Anlegen einer Bibliothek erfolgt mit dem Dienstprogramm LBCREATE (siehe Kapitel 7).

### 5.1.4 Verwaltung der Bibliotheken

Bibliotheken können wie in 5.1.3 beschrieben, vereinbart werden. Inhaltsverzeichnisse, die Informationen über die vorhandenen Bibliotheken ausgeben, sind über den Systembefehl LVTOC (Library Volume Table of Contents) erhältlich.

Sollen Bibliotheken oder ihr Inhalt als Gesamtheit gelöscht werden, so stehen mehrere Dienstprogramme zur Verfügung:

LBEMPTY        löscht alle in einer Bibliothek gespeicherten Files. Die Bibliothek bleibt jedoch bestehen.

LBSCRATCH    löscht eine Bibliothek (und damit auch ihren Inhalt). Um den dadurch frei werdenden Platz wieder verwenden zu können, muß das Dienstprogramm RESTRUCT ausgeführt werden.

DINIT        erlaubt das Löschen aller Bibliotheken einer Diskette oder Platte.

Soll der Inhalt einer Bibliothek oder Teilbibliothek in eine andere, bestehende Bibliothek kopiert werden, so ist das Dienstprogramm LIBCOPY auszuführen. Einzelne Files können mit FLCOPY von einer (Teil-)Bibliothek in eine andere (Teil-)Bibliothek kopiert werden.

Soll der Name oder das Password einer Bibliothek geändert werden, steht das Dienstprogramm LBRENAME zur Verfügung.

#### 5.1.5        Arbeiten mit Bibliotheken

Im vorangehenden Abschnitt wurde auf die Möglichkeiten zur Verwaltung von Bibliotheken hingewiesen.

Im folgenden werden nun die zum Arbeiten mit Bibliotheken erforderlichen Bedienungselemente beschrieben.

Um auf in einer Bibliothek enthaltene Files zugreifen zu können, muß dem System bekannt sein, mit welchen Bibliotheken gearbeitet werden soll. Dieser Vorgang wird als "Öffnen einer Bibliothek" bezeichnet und mit dem Systembefehl LBOPEN durchgeführt. Maximal sechs Bibliotheken können gleichzeitig geöffnet sein.

Mit dem Systembefehl LBSTORE wird erreicht, daß bestimmte Bibliotheken bei jedem Laden des Betriebssystems automatisch geöffnet werden. Sie werden "Standardbibliotheken" genannt. Sind keine Bibliotheken als Standardbibliotheken festgelegt, so wird die erste Bibliothek der Systemdisk automatisch geöffnet, sofern sie nicht durch ein Password geschützt ist.

#### Bemerkungen:

- Das Öffnen bewirkt keine Markierung in der entsprechenden Bibliothek. Im Arbeitsspeicher der Basiseinheit werden Informationen über die momentan verwendeten Bibliotheken gespeichert. Diese Informationen gehen daher beim Ausschalten der Maschine oder beim Neuladen des Betriebssystems verloren.

- Ist die Arbeit mit einer oder allen geöffneten Bibliotheken beendet, so können sie mit dem Befehl LBCLOSE geschlossen werden. Der Zugriff auf den Inhalt dieser Bibliothek(en) ist danach nicht mehr möglich. Sollen anstelle der momentan geöffneten Bibliotheken wieder die Standard-Bibliotheken (LBSTORE) geöffnet werden, steht dafür der Systembefehl LBRESTORE zur Verfügung.
- Für die Anwendung aller Systembefehle und der meisten Dienstprogramme, die sich auf Files oder den Inhalt einer Bibliothek beziehen, muß die Bibliothek geöffnet sein. Für die geöffneten Bibliotheken oder Teilbibliotheken kann mit CATALOG ein Inhaltsverzeichnis ausgegeben und mit SPACE der verfügbare Platz ermittelt werden.

#### 5.1.6 Zugriff auf Bibliotheken und Files

Bibliotheken werden über "lib ref" identifiziert. Fehlt in "lib ref" die Angabe "unit name" und soll auf eine vorhandene, aber nicht geöffnete Bibliothek zugegriffen werden, so wird diese Bibliothek auf der Systemdisk gesucht. Ist in "lib ref" nur "unit name" angegeben, so wird die erste Anwenderbibliothek auf der Einheit dieses Namens angesprochen.

Bei offenen Bibliotheken wird der angegebene Bibliotheksname in der Reihenfolge des Öffnens der Bibliotheken gesucht. Dabei wird immer nach der ersten Übereinstimmung der in "lib ref" vorhandenen Angaben mit der Liste der geöffneten Bibliotheken (siehe LVT\*) gesucht.

Bei sich auf Files beziehenden Systembefehlen kann "lib ref" entfallen.

Wird ein bestehendes File verlangt, so werden die Bibliotheken in der Reihenfolge ihres Öffnens durchsucht.

Wird ein neues File erstellt, ohne daß "lib ref" angegeben wird, so wird es in der ersten geöffneten Bibliothek angelegt.

Werden in Programmen Datenfiles verwendet, wo werden die Bibliotheken beim Programmaufruf in der Reihenfolge ihres Öffnens nach diesen Files durchsucht.

#### Bemerkungen:

- Enthalten verschiedene (offene) Bibliotheken Files mit gleichem Namen, so wird auf das als erstes gefundene File zugegriffen. Bei der Anwendung von Systembefehlen kann durch Angabe der entsprechenden "lib ref" auf jedes gewünschte File zugegriffen werden.



- Die Systembefehle PREPARE und RUN brauchen zur Ausführung Platz in der ersten offenen Bibliothek, wenn sie ohne Angabe eines Filenamens aufgerufen werden. Steht dieser Platz nicht zur Verfügung (ERROR 188), so kann gegebenenfalls durch eine andere Reihenfolge des Öffnens der Bibliotheken dieser Platz geschaffen werden.

## 5.2 Allgemeine Informationen über externe Files

### 5.2.1 Allgemeines

Ein File ist eine problemorientierte, organisatorische Zusammenfassung gleichartiger oder gleichbehandelter Daten in einem Speichermedium.

Es werden vier Filetypen unterschieden, und zwar:

1. Programmfiles
2. Textfiles
3. Datenfiles
4. Objectfiles

Die Files heißen "extern", da sie in einer Bibliothek auf einem externen Speicher enthalten sind.

#### Programm- und Textfiles

Programm- bzw. Textfiles erlauben es, Programme bzw. Texte unter einem Namen zu speichern.

- Ein Programmfile enthält eine mit Zeilennummern versehene Folge ausführbarer BASIC-Anweisungen.
- Ein Textfile besteht aus einer Folge numerierter Zeilen, die beliebige Folgen von ISO-Zeichen enthalten.

Ein Programmfile kann somit nur ein Programm, ein Textfile nur einen Text enthalten. Die Begriffe "Programm" und "Programmfile" (bzw. "Text" und "Textfile") haben also die gleiche Bedeutung.

Für die Behandlung von Programm- und Textfiles ist eine Reihe von Systembefehlen vorgesehen. Textfiles können außerdem in einem BASIC-Programm wie sequentielle Datenfiles gelesen werden.

#### Datenfiles

Datenfiles werden für die programmunabhängige, permanente Speicherung numerischer und alphanumerischer Daten verwendet.

## Objectfiles

Objectfiles enthalten eine ausführbare Assembleroutine, die von einem BASIC-Programm aufgerufen werden kann.

### 5.2.2 Filenamen

Jedes File in einer Bibliothek wird durch seinen Namen identifiziert. Über diesen Namen kann das File gesucht oder vor unerlaubtem Zugriff geschützt werden.

Der Filename setzt sich aus einem Kennzeichen der Teilbibliothek (wenn das File einer Package- oder Common-Teilbibliothek angehört) und dem eigentlichen Namen zusammen.

- Der eigentliche Name besteht aus maximal 6 Großbuchstaben oder Ziffern, wobei das erste Zeichen ein Buchstabe sein muß.

- Für das Kennzeichen der Teilbibliothek gilt:

- "\*": File einer Package-Teilbibliothek

- "+": File einer Common-Teilbibliothek

Files, die zum Bestand einer User-Teilbibliothek gehören, haben kein Kennzeichen, d.h., ihre Namen beginnen mit einem Großbuchstaben.

	richtige Namen	falsche Namen	
Package-Teilbibliothek	*SINES	*	(kein Alphazeichen)
Common-Teilbibliothek	+G	+8G	(das 1. Zeichen ist kein Buchstabe)
User-Teilbibliothek	GRAPH2	GRAPH66	(mehr als 6 Zeichen)

### 5.2.3 Erstellen von Files

#### Erstellen von Programmen und Textfiles

Programme und Texte werden entweder unmittelbar im Arbeitsspeicher kreiert oder sie ergeben sich aus der Umwandlung eines Files mit anderem Typ.

Soll ein Programm über die Tastatur im Arbeitsspeicher erstellt werden, so ist vor der Eingabe der Befehl "NEW" einzugeben (alle in der Folge erwähnten Systembefehle sind in Kapitel 6 beschrieben).

Soll ein Textfile erstellt werden, so lautet der entsprechende Befehl "TEXT".

Wird nach der Erstellung eines Programmes oder Textes der Befehl "SAVE" gegeben, so wird das sich im Arbeitsspeicher befindliche Programm (bzw. der Text) unter seinem Namen in der ebenfalls namentlich angegebenen Teilbibliothek einer Bibliothek gespeichert.

Beispiel:

```
TEXT
AUTO#
10 DAS IST EIN TEXT
20 DER IN DER COMMON-TEILBIBLIOTHEK
30 DER OBEREN FLOPPY-DISK-STATION
40 GESPEICHERT WERDEN SOLL.
```

Nach der Ausführung des Befehles "SAVE +TEXT1,(,F1)" ist der Text in der Common-Teilbibliothek der Diskette der oberen Floppy-Disk-Station enthalten und kann über seinen Namen wieder angesprochen werden (z.B. mit OLD +TEXT1).

#### Erstellen von Datenfiles

Für die vollständige Vereinbarung eines Datenfiles sind folgende Angaben erforderlich:

1. Name (der auch die Teilbibliothek festlegt).
2. Angabe der für die Erstellung des Files vorgesehenen Bibliothek.
3. Zugriffsart
4. Maximaler Speicherbedarf

Ein Datenfile wird mit dem Befehl "CREATE" erstellt.

#### Zugriffsarten

Ein Datenfile kann sequentiellen oder random (direkten) Zugriff erlauben.

##### - Sequentielle Datenfiles:

Die Daten sind im File in unmittelbarer Folge gespeichert. Das Schreiben von Daten in ein solches File erzeugt eine physische Reihenfolge, die nicht unterbrochen werden kann, und die auch die Reihenfolge des Lesens der Daten aus dem File festlegt.

Sequentielle Datenfiles können nur auf zwei Arten beschrieben werden:

- Ab Beginn des File. Der vorherige Inhalt wird dabei vollständig zerstört.

- Durch Anhängen der Daten an den bestehenden Inhalt, der dabei erhalten bleibt. Es ist nicht möglich, ein Datenelement direkt anzusprechen und gegen ein anderes auszutauschen.
- Randomfiles
  - Wird für ein Datenfile die Zugriffsart "random" festgelegt, so kann auf jedes Datenelement des Files direkt zugegriffen werden, das heißt, die Daten können an jede beliebige Stelle im File geschrieben bzw. ab jeder Position im File gelesen werden. Die Festlegung der Position des Datenelementes kann unabhängig von der Schreib- oder Leseanweisung in einer eigenen Anweisung erfolgen.

#### Platzbedarf

Die Daten eines Files werden in binärem Format gespeichert. Die kleinste Einheit, auf die in einem Datenfile zugegriffen werden kann, ist das aus 4 Bytes bestehende Wort.

Eine numerische Information belegt im Datenfile in einfacher Genauigkeit ein Wort, in doppelter Genauigkeit zwei Worte.

Für alphanumerische Daten (Strings) errechnet sich der Platzbedarf wie folgt:

- Die Anzahl der Worte ist gleich  $\text{INT}((n-1)/4+2)$ , wenn "n" die Anzahl der Zeichen im String angibt.

Um im Befehl CREATE die Anzahl der für ein Datenfile zu reservierenden Bytes festzulegen, muß entsprechend der Art und der maximalen Anzahl der im File aufzunehmenden Daten zunächst die Anzahl der Worte und daraus die erforderliche Anzahl Bytes errechnet werden.

#### Bemerkungen:

- Das Wort ist die kleinste Einheit, auf die in einem Randomfile zugegriffen werden kann.
- Ist die Anzahl der im CREATE-Befehl angegebenen Bytes kein Vielfaches von 128 bzw. 256, so wird diese Anzahl automatisch auf das nächstgrössere Vielfache von 128 bzw. 256 erhöht. Erfolgt keine Längenangabe, so werden 4096 Bytes reserviert.
- Wird das File in der Bibliothek einer Standard-Diskette (256 KB) angelegt, so ist seine Größe immer ein Vielfaches von 128 Bytes. Wird es hingegen in einer Bibliothek einer Megafloppy oder Festplatte angelegt, so ist seine Größe immer ein Vielfaches von 256 Bytes.
- Bei Randomfiles ist die aktuelle Länge, unabhängig von der tatsächlichen Belegung mit Daten, immer gleich der maximalen Länge.

#### Beispiel:

In einer Bibliothek namens "ARCHIV" soll unter dem Namen "SFILE" ein File mit sequentielltem Zugriff erstellt werden. 1000 Strings mit je zehn Zeichen sollen darin Platz finden. Der erforderliche Platzbedarf in Bytes beträgt demzufolge:

$$1000*4*INT((10-1)/4+2)=4000*4=16000$$

Der CREATE-Befehl zur Erstellung dieses Files lautet:

CRE SFILE, ARCHIV,S,16000

#### 5.2.4 Öffnen von Daten-Files

Bevor ein Programm auf ein externes Datenfile zugreifen kann, muß dieses durch Angabe seines Namens in einer FILES- oder FILE:-Anweisung geöffnet werden. Jedem angeführten File wird ein logischer Kommunikationskanal zwischen Hauptspeicher und Diskstation zugeordnet. Jeder dieser Kanäle trägt eine Nummer, die das ihm durch seinen Namen zugeordnete File im weiteren Programmablauf identifiziert. Die Zuordnung der Kanäle zu den Files wird durch die Position des Filenamens in der FILES-Anweisung bestimmt. Die daraus entstehende Ordnungszahl heißt "File-designator".

Nach dem Öffnen sind Ein- und Ausgabeoperationen mit diesem File möglich, doch muß zwischen Random- und sequentielltem File unterschieden werden.

#### Sequentielle Files

Nach dem Öffnen eines sequentiellen Files in einer FILES- oder FILE:-Anweisung kann das File ab dessen Beginn gelesen werden. Ist eine Aufzeichnung vorgesehen, so ist dies erst nach der Anweisung SCRATCH: möglich, falls ab Beginn des Files geschrieben werden soll, oder nach APPEND:, falls an ein bestehendes File Daten angefügt werden sollen.

Soll aus dem Schreib-Mode wieder in den Lese-Mode übergegangen bzw. mit dem Lesen erneut begonnen werden, so ist dies ab Filebeginn nach der Anweisung RESTORE: möglich.

#### Randomfiles

Neben allen von sequentiellen Files in einem Programm gebotenen Einsatzbereichen gibt es bei Files mit Randomzugriff folgende Möglichkeiten:

Nach dem Öffnen des Files in einer FILES- oder FILE:-Anweisung kann sowohl gelesen als auch geschrieben werden. Die Positionierung auf ein bestimmtes Wort im File erfolgt mit einer SETW:-Anweisung.

Es gibt zwei Arten von Randomfiles:

- R-File:

Wird das File mit Parameter R vereinbart, so enthält es nach der Ausführung von CREATE numerische Daten in einfacher Genauigkeit, die den Wert "nicht initialisiert" haben.

- Z-File:

Wird das File mit Parameter Z vereinbart, so enthält es nach der Ausführung von CREATE numerische Daten in einfacher Genauigkeit, die den Wert 0 aufweisen.

### Textfile

Textfiles können in einem Programm vom Lese-Modus wie sequentielle Datenfiles behandelt werden. Nach dem Öffnen oder nach Ausführung der Anweisung RESTORE: können Textfiles mit READ: gelesen werden. Jede Zeile des Textfiles entspricht einem String, wobei die Zeilennummer Bestandteil dieses Strings ist.

#### 5.2.5

##### Schließen eines Files

Werden in Programmen bestimmte Operationen mit Datenfiles ausgeführt, so werden diese Files in ihrer Bibliothek als geöffnet gekennzeichnet. Diese Markierung wird nach Beendigung der laufenden Operation oder bei einem regulären Programmende (END oder Funktion BREAK) gelöscht. Bei einem irregulären Programmende (z.B. bei Stromausfall bleibt die Markierung erhalten.

Bei der nächsten Ausführung eines auf ein offen gebliebenes Datenfile zugreifenden Programmes wird ERROR 76 gemeldet. Mit Hilfe des Systembefehls CATALOG können dann die offen gebliebenen Files ermittelt werden. Solche Datenfiles sind vor dem neuerlichen Programmstart mit dem Befehl VALIDATE zu schließen, wobei jedoch zu beachten ist, daß durch ein irreguläres Programmende Informationen verloren gehen können.

#### 5.2.6

##### Schützen von Files

Das System M40 ST ermöglicht das Schützen von Programm-, Text- und Datenfiles vor bestimmten Operationen. Es ist dadurch zum Beispiel möglich, Programme zu erstellen, die zwar ausgeführt, aber nicht gedruckt oder modifiziert werden können. Der Schutz einzelner Files erfolgt mit dem Befehl SECURE.

6.	<u>SYSTEMBEFEHLE</u>	6.1
6.1	Eingabe von Befehlen über die Tastatur	6.2
6.2	Ausführung von Befehlen mit Prozeduren	6.3
6.2.1	Erstellen einer Prozedur	6.3
6.2.1.1	Eingabe von Systembefehlen und Dienstprogrammen	6.3
6.2.1.2	Eingabe von BASIC-Statements	6.3
6.2.1.3	Spezielle Befehle für Prozeduren	6.4
6.2.1.4	Prozedur als Datenfile	6.4
6.2.2	Aufruf einer Prozedur	6.4
6.3	Liste und Funktionen der Systembefehle	6.5
	CALCULATOR	6.9
	CATALOG	6.11
	CONFIGURE	6.15
	CREATE	6.19
	DATE	6.21
	DCHANGE	6.23
	DRAW	6.25
	ENVIRONMENT	6.27
	ERASE	6.29
	EXEC	6.31
	LBCLOSE	6.33
	LBOPEN	6.35
	LBSTORE	6.37
	LVTOC	6.39
	MODIFY	6.41
	OPTIONS	6.43
	PROCEDURE	6.45
	PURGE	6.51
	REVERSE	6.53
	SPACE	6.59
	VALIDATE	6.61





Das Computersystem M40 ST erlaubt die Lösung technischer und wissenschaftlicher Probleme mit Programmen in der problemorientierten Sprache BASIC. Auch die Kommunikation mit dem System erfordert eine Sprache. Sie besteht aus Systembefehlen, die das Erstellen und Ausführen eines BASIC-Programmes rasch und einfach gestalten. Mit den verfügbaren Befehlen kann man unter anderem:

- Programme erstellen
- Programme ausführen
- Programme modifizieren
- Programme speichern
- Daten- und Textfiles erstellen und abspeichern
- Bibliotheken verwalten

Die Systembefehle ermöglichen außerdem den Austausch von Informationen zwischen dem Hauptspeicher und den externen Einheiten des Systems, nämlich der Floppy-Disk- (FDU) oder einer Platten-Einheit sowie dem Drucker.

In diesem Kapitel wird der Aufbau und die Anwendung dieser Befehle besprochen. Vor der Erläuterung der einzelnen Systembefehle ist es aber notwendig, die allgemeinen Regeln und Elemente dieser Befehle kennenzulernen.

Die folgenden Symbole werden nur verwendet, um das Format eines Befehles zu definieren. Sie sind nicht Bestandteil des Befehles:

-	Bindestrich
<u></u>	Unterstreichung
{ }	geschweifte Klammer
[ ]	eckige Klammer
...	Folge von Punkten

Diese Symbole bedeuten:

{ }	schließt eine Folge von Parametern ein, von denen mindestens einer vorkommen muß.
[ ]	schließt eine Folge von optionalen Parametern ein. Diese Parameter können (müssen aber nicht) angegeben werden.
-	Stehen mehrere Parameter zur Auswahl und wird keiner davon angegeben, so setzt das System den unterstrichenen Parameter ein. Ein unterstrichener Parameter muß also nicht explizit eingegeben werden. Kommata, die am Ende von Eingaben trotz Weglassen von Parametern aufgrund der Formate erforderlich wären, entfallen.

- ... der vorherige Operand kann mehrmals wiederholt werden.
- , trennt die Operanden eines Systembefehles voneinander.

Zu beachten ist, daß das Format eines Befehles, insbesondere die Stellung eines Parameters innerhalb desselben, genau eingehalten werden muß.

#### Beispiel

Richtig:	Falsch:
CAT ,,,F	CAT F

6.1

### Eingabe von Befehlen über die Tastatur

Ein Befehl besteht aus einem Schlüsselwort, dem meistens ein oder mehrere Operanden folgen. Die Schlüsselwörter sind englische Begriffe, die die Funktion des Befehles beschreiben.

Die Operanden liefern die zusätzlichen Informationen zur Ausführung der Operation. Für die Schlüsselwörter der am häufigsten verwendeten Befehle sind eigene Tasten vorhanden. Das beschleunigt die Eingabe der Befehle und hilft, Eingabefehler zu vermeiden. Alle Schlüsselwörter können auf die ersten 3 Buchstaben abgekürzt werden, da das System nur die ersten 3 Zeichen zur Bestimmung eines Befehles analysiert.

Wird nach dem dritten Zeichen eines Schlüsselwortes ein falscher Buchstabe eingegeben, wird dieser Name trotzdem richtig interpretiert.

Nach Eingabe des Schlüsselwortes sind die Operanden zeichenweise einzutasten. Die Anweisung wird danach durch Drücken der Taste END OF LINE abgeschlossen. Zwischen dem Schlüsselwort und dem ersten Operanden müssen eine oder mehrere Leerstellen stehen.

Nach dem Drücken der Taste END OF LINE wird der Befehl analysiert. Wenn der Befehl vom System interpretiert werden kann, wird er unmittelbar ausgeführt. Andernfalls erscheint eine Fehlermeldung. Der Befehl kann dann mit den Möglichkeiten der Editing-Operationen im Display korrigiert werden.

Die Eingabe der Systembefehle kann auch mit Hilfe der Funktionstasten F1-F16 erfolgen (siehe FKEY in den Kapiteln 6.5 und 9.1)

6.2

## 6.2

### Ausführung von Befehlen mit Prozeduren

Um eine Vielzahl von Systembefehlen oder andere manuelle Eingaben ohne Eingriff des Benützers ablaufen lassen zu können, bietet das System dem Anwender die Möglichkeit, Katalog-Prozeduren zu definieren.

Zu diesem Zweck werden die entsprechenden Anweisungen und Befehle in einem File - dem COMMAND FILE - gespeichert. Die Ausführung der Befehlsfolge wird durch den Befehl PRO (PROCEDURE) gestartet. Dies ermöglicht dem Benutzer, "Programme" auf Systemebene zu erstellen. Häufig verwendete Folgen von Systemanweisungen können also in Commandfiles abgelegt und bei Bedarf abgerufen werden.

#### 6.2.1

##### Erstellen einer Prozedur

Eine Prozedur wird am einfachsten als Textfile definiert und anschliessend mit TRANSCODE in ein Datenfile umgewandelt.

Zuerst sind die für das Erstellen eines Textfiles erforderlichen Systembefehle einzugeben, also TEXT und normalerweise auch AUTO .

##### 6.2.1.1

##### Eingabe von Systembefehlen und Dienstprogrammen

Systembefehle und Aufrufe von Dienstprogrammen sind zeilenweise einzugeben.

Beispiel:

```
30 CRE *DFILE, (MAXL, D1), R, 50000
40 EXE LBC,(MORITZ,D2),,SIZE=100,*=2,+=3,NP=5
```

##### 6.2.1.2

##### Eingabe von BASIC-Statements

Innerhalb einer Prozedur können auch Programme neu erstellt oder vorhandene Programme modifiziert werden. Bei den einzelnen Programmen ist zusätzlich zur Zeilennummer des Textfiles auch die Zeilennummer des BASIC-Statements einzugeben. Wird ein Programm in einer Prozedur neu erstellt, muß der Systembefehl NEW eingetastet und das Programm mit END beendet werden.

Beispiel:

```
10 NEW
20 10 INPUT A
30 20 B=A*PI
40 30 PRINT A,B
50 40 END
60 RUN
```

#### Bemerkung:

Die Syntax-Kontrolle der BASIC-Anweisungen erfolgt erst bei der Ausführung der Prozedur. Für die Verarbeitung eines so definierten Programmes gelten die üblichen Regeln.

#### 6.2.1.3

##### Spezielle Befehle für Prozeduren

Zusätzlich zu den allgemeinen Systembefehlen bestehen spezielle Befehle, die nur innerhalb einer Prozedur verwendet werden können. Ihre Anwendung ist in Kapitel 6.4 ausführlicher beschrieben.

IN=                    für automatischen Input bei der Verarbeitung von Programmen durch Prozeduren.

IF CC=                für bedingte Sprünge innerhalb einer Prozedur.

TEST,PRO             für spezielle Behandlung von Fehlermeldungen.

#### 6.2.1.4

##### Prozedur als Datenfile

Das Commandfile einer Prozedur kann direkt als Datenfile erstellt werden. Folgender Aufbau ist zu berücksichtigen:

- Jede Prozedurzeile wird als String abgespeichert (max. 80 Zeichen).
- Die ersten 4 Zeichen jedes Strings enthalten eine 4-stellige Zeilennummer, eventuell mit führenden Nullen.

#### 6.2.2

##### Aufruf einer Prozedur

Eine Prozedur kann auf mehrere Arten zur Verarbeitung aufgerufen werden:

- Manueller Aufruf mit dem Systembefehl PROCEDURE (siehe Kapitel 6.3).
- Automatischer Aufruf im Anschluss an eine Initialisierungsphase des Systems. In diesem Falle muss die Prozedur unter dem Namen \*SETUP abgespeichert sein.
- Automatischer Aufruf durch eine andere Prozedur.

Liste und Funktion der Systembefehle

Die Systembefehle und ihre Funktionen sind nachstehend in alphabetischer Reihenfolge aufgeführt.

<u>Name</u>	<u>Funktion</u>
CALCULATOR	ruft den "Calculator-Mode auf.
CATALOG	druckt das Inhaltsverzeichnis einer offenen Bibliothek.
CONFIGURE	erlaubt die Festlegung einer bestimmten Systemkonfiguration.
CREATE	reserviert den Speicherplatz für ein Datenfile in einer Bibliothek.
DATE	speichert das Datum auf der Systemdisk oder -diskette.
DCHANGE	erlaubt den Wechsel einer Diskette. Das Programm im Hauptspeicher wird dadurch nicht gelöscht.
DRAW	erstellt eine Hardcopy des Bildschirminhaltes auf dem Thermodrucker.
ENVIRONMENT	beschreibt die aktuelle Systemkonfiguration.
ERASE	löscht den Bildschirminhalt.
EXEC	lädt ein Dienstprogramm (siehe Kapitel 7) in den Arbeitsspeicher und verarbeitet es.
LBCLOSE	schließt Bibliotheken.
LBOPEN	öffnet Bibliotheken.
LBSTORE	definiert die momentan offenen Bibliotheken als "Standard" für automatisches Öffnen bei jedem Laden des Systems.
LVTOC	druckt das Verzeichnis der Bibliotheken einer Einheit oder aller offenen Bibliotheken.

MODIFY	ändert den Namen und/oder den reservierten Speicherplatz eines Files.
OPTIONS	lädt zusätzliche Teile des Betriebssystems in den Hauptspeicher. Der für den Anwender verfügbare Teil des Hauptspeichers wird dadurch verkleinert.
PROCEDURE	führt eine Katalogprozedur aus.
PURGE	löscht ein File einer nicht geschützten Bibliothek.
REVERSE	Umkehrung des Bildes auf dem Bildschirm.
RUN	startet die Ausführung eines Programmes.
SPACE	zeigt den verfügbaren Speicherplatz einer Bibliothek im Display an.
VALIDATE	schließt ein offengebliebenes File.

# CALCULATOR



BEFEHL: CALCULATOR

FUNKTION: Aufruf des Calculator-Mode um direkte Rechenoperationen auszuführen.

FORMAT:  $\text{CAL}[\text{CULATOR}] \left[ \begin{matrix} n \\ \text{FL} \end{matrix} \right]$

n ist eine natürliche Zahl zwischen 0 und 13.

FL Ergebnisse sollen in Gleitkommadarstellung angezeigt werden.

WIRKUNG: Das System wird vom Command-Mode in den Calculator-Mode umgeschaltet. Der Operand (n) bestimmt die Anzahl Nachkommastellen, mit denen das Ergebnis einer Rechenoperation dargestellt wird. Der Operand (FL) bestimmt, daß die Darstellung in Gleitkommaformat erfolgen soll. Fehlt der Operand, so wird das Ergebnis im Standardformat angezeigt.

BEMERKUNGEN:

- Durch Eingabe irgendeines gültigen Systembefehles (außer CALCULATOR) kehrt das System vom Calculator-Mode in den Command-Mode zurück.
- Nähere Beschreibung des Calculator Modes erfolgt in Kapitel 8.

BEISPIELE: siehe Kapitel 8





BEFEHL: CATALOG

FUNKTION: Ausdruck des Inhaltsverzeichnisses von Bibliotheken.

FORMAT: 
$$\text{CAT[ALOG]} \left[ \left( \begin{array}{c} \text{filename[,lib ref]} \\ \left[ \begin{array}{c} * \\ + \\ : \end{array} \right] , [\text{lib ref}], \left[ \begin{array}{c} P \\ T \\ D \\ O \end{array} \right] [,F] \end{array} \right) \right]$$

"lib ref" kann folgende Formen annehmen:

- (lib name,unit name)
- lib name
- (,unit name)

Als "lib name" ist der Bibliotheksname, als "unit name" der Name der Diskstation anzugeben.

"\*" = Package-Teilbibliothek.

"+" = Common-Teilbibliothek.

":" = ganze Bibliothek.

"filename" Werden nur Informationen über ein bestimmtes File verlangt, ist "filename" durch den effektiven Namen des Files zu ersetzen.

"P" = Druck des Kataloges der Programmfiles.

"T" = Druck des Kataloges der Textfiles.

"D" = Druck des Kataloges der Datenfiles.

"O" = Druck des Kataloges der Objektfiles (Assembler-Routinen).

"F" = Druck aller Informationen über die Files.

WIRKUNG: Der vollständige Befehl mit allen Operanden (z.B. CATALOG :,(LIB5,D1),P,F) bewirkt den Ausdruck der Informationen über alle durch die ersten 3 Operanden spezifizierten Files:

- Filename
- Art des Files
- Erstellungsdatum des Files
- Datum der letzten File-Modifikation
- Für das File reservierter Speicherplatz in Bytes

- Aktuelle Länge des Files in Bytes
- Identifikationscode des Files
- Anzahl der Abschnitte (max. 4), in die das File innerhalb einer Bibliothek zerlegt ist.

Offen gebliebene Datenfiles werden durch das Wort OPEN gekennzeichnet (siehe Systembefehl VALIDATE).

Für die einen Common-Bereich verwendenden Programme wird die dafür eingesetzte Anzahl Worte mit C=...W angegeben.

Fehlt der 4. Operand (F), werden folgende, gekürzte Informationen über die durch die anderen Operanden spezifizierten Files ausgegeben:

- Filename (ohne Bibliotheksangabe \* oder +)
- Art des Files

Fehlt der 3. Operand, erfolgt der Ausdruck der Informationen über Programm-, Text- und Datenfiles (sowie Objektfiles).

Fehlt der 2. Operand, erfolgt der Ausdruck der Informationen über die erste offene Bibliothek (siehe LVT\*).

Fehlt der 1. Operand, erfolgt der Ausdruck der Informationen über die Files der User-Teilbibliothek.

- BEMERKUNGEN:
- Steht als 1. Operand "filename", ist der 3. und 4. Operand bedeutungslos.
  - Die Ausführung des Befehles kann durch Drücken der Tasten CTRL + EXIT abgebrochen werden.
  - Ein mit DEAD gekennzeichnetes File ist nicht mehr verwendbar und kann mit PURGE nicht gelöscht werden. Damit der Speicherplatz dieses Files wieder verfügbar wird, sind alle übrigen Files dieser Bibliothek mit LIBCOPY in eine "neue" Bibliothek zu kopieren. Anschließend kann mit LBSCRATCH die "alte" Bibliothek gelöscht werden.
  - Die durch "lib ref" angegebene Bibliothek muß offen sein (siehe LBOPEN).
  - Beinhaltet der Operand "lib ref" nur den Bibliotheks-namen, so wird der Katalog der ersten offenen Bibliothek gleichen Namens gedruckt.
  - Besteht der Operand "lib ref" nur aus dem Namen der Einheit, so wird der Katalog der ersten offenen, auf dieser Einheit gespeicherten Bibliothek ausgedruckt.
  - Für jedes File belegt das Betriebssystem einen zusätzlichen Sektor mit Kontrollinformationen, der in der angegebenen Filelänge nicht enthalten ist (siehe SPACE).

BEISPIELE:    CAT :                    Druck einer Liste der Namen und Arten  
    aller Files der ersten offenen Biblio-  
    thek (LVT\*).

CAT +,(LIB1,F1),T,F    Druck eines Kataloges der Textfiles  
    der Common-Teilbibliothek von LIB1  
    auf der Disk F1.

CAT \*,,,F                Ausdruck eines vollständigen Kataloges  
    aller Files der Package-Teilbibliothek  
    der ersten offenen Bibliothek.

Ausdruck aller Teilbibliotheken der oberen Diskettenstation  
 mit vollständiger Information.

CAT :,(F1),,F

MDOSC -R 2.0    \* VOLLABEL = INTERN    \* LIBRARY = P6FSYS    \* DATE:22-05-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
*STEUER	R	27-06-77	27-06-77	4096	4096		1
*OK	P	23-11-79	20-06-78	3712	3712	EDOCEDOC	1
*UOL	P	27-06-77	20-06-78	0640	0640		1
*SEC	P	23-11-79	20-06-78	3712	3712	EDOCEDOC	1
*CODE	P	23-11-79	20-06-78	3968	3968	EDOCEDOC	1
*TEXT	P	11-12-78	11-12-78	0768	0768		1
+READHX	P						DEAD
+SEQU	S	22-05-80	22-05-80	1024	0000		OPEN
+RANDOM	R	22-05-80	22-05-80	1152	1152		1
+ZERO	Z	22-05-80	22-05-80	0896	0896		1
COPY	P	20-06-78	20-06-78	0384	0384		1
D	P	20-06-78	11-12-78	10240	10240		1
MIDOC	P	07-12-78	07-12-78	9984	9984		1
HEXA	P	11-12-78	11-12-78	1152	1152		1
FNK	T	11-12-78	11-12-78	0896	0896		1

Ausdruck der Liste der Files der User-Teilbibliothek der  
 ersten offenen Bibliothek.

CAT

MDOSC -R 2.0    \* VOLLABEL = SYSBTS    \* LIBRARY = P6066    \* DATE:22-05-80

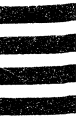
FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
FIGUR1	P						
PROG1	P						
FIGUR2	P						
START	P						
MATRIX	R						
MORGEN	P						
XX	S						

Ausdruck aller Informationen zum File \*STEUER.

CAT \*STEUER, C.F1)

\*STEUER R 27-06-77 27-06-77 4096 4096

1



BEFEHL: CONFIGURE

FUNKTION: Festlegung einer bestimmten Systemkonfiguration

FORMAT:  $\text{CON}[\text{FIGURE}][\text{MS}=\text{n}][, \text{PRO}] \left[ \text{SP} = \begin{Bmatrix} 0 \\ \vdots \\ 9 \end{Bmatrix} F \begin{Bmatrix} 7 \\ 8 \end{Bmatrix} \begin{Bmatrix} Y \\ N \end{Bmatrix} \begin{Bmatrix} E \\ 0 \end{Bmatrix} \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \right]$

n Kapazität des Anwenderspeichers in K-Bytes  
( $15 \leq n \leq 192$ )

Pro Automatischer Start der Katalogprozedur \*SETUP  
bei jedem Laden des Betriebssystems.

SP=... Bestimmung der Übertragungsparameter des angeschlossenen Systemdruckers.

WIRKUNG: Der Inhalt des Arbeitsspeichers wird gelöscht und das System neu initialisiert. Die Parameter werden auf der Systemdisk gespeichert und bestimmen die Konfiguration so lange, bis sie durch einen neuen CONFIGURE-Befehl geändert wird.

- Der Parameter MS (Memory-Size) ist angegeben:

Die Größe des Anwenderspeichers wird auf n K-Bytes reduziert.

- Der Parameter PRO ist angegeben:

Beim Laden des Betriebssystems wird automatisch die Prozedur mit dem Namen \*SETUP gestartet.

- Der Parameter SP=... ist angegeben:

Der angeschlossene Systemdrucker wird mit den unter SP angegebenen Übertragungsparametern angesteuert. Im folgenden werden die einzelnen Übertragungsparameter erläutert (V24-Schnittstelle)

$\begin{Bmatrix} 0 \\ : \\ 9 \end{Bmatrix}$  Durch eine ganze Zahl zwischen 0 und 9 wird die Übertragungsgeschwindigkeit (Band-Rate) bestimmt. Die folgende Tabelle gibt die zum jeweiligen Zahlen-Code korrespondierende Band-Rate an:

<u>Zahlen-Code</u>	<u>Band-Rate</u>
0	50
1	110
2	300
3	600
4	1200
5	2400
6	4800
7	9600
8	19200
9	38400

F Parameter F (Full Duplex) ist obligatorisch und immer anzugeben.

$\begin{Bmatrix} 7 \\ 8 \end{Bmatrix}$  Definiert die Anzahl bits pro übertragener Bytes.

$\begin{Bmatrix} Y \\ N \end{Bmatrix}$  bestimmt, wo ein Parity-Check zu machen ist:

Y (YES) für "Ja",  
N (No) für "Nein"

$\begin{Bmatrix} E \\ O \end{Bmatrix}$  definiert die Art des Parity-Checks:

E (Even) für gerade Parität  
O (Odd) für ungerade Parität

$\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$  bestimmt die Anzahl Stop-Bits nach jedem Zeichen.

BEMERKUNGEN: - Wird der Parameter MS nicht angegeben, so wird vom System die volle Anwenderkapazität benutzt.

- Wird der Parameter PRO nicht angegeben, so wird eine (vorhandene) Prozedur \*SETUP nicht ausgeführt.
- Wird der Parameter SP nicht angegeben, so werden die Defaultwerte SP=8F7YE2 angenommen. Diese entsprechen den Übertragungsparametern des Thermo-  
druckers PR 2400 als Systemdrucker. Bei Nadeldruckern sollte SP=6F7YE2 angegeben werden.
- Wird nur CON (ohne Parameter) eingegeben, so wird das System neu geladen und die folgenden Defaultwerte angenommen:
  - . Volle Anwenderkapazität
  - . Keine automatische Startprozedur
  - . Übertragungsparameter für Thermodrucker PR 2400.

BEISPIELE:

CON PRO

CON SP=6F7YE2





BEFEHL: CREATE

FUNKTION: Reservieren des für ein Datenfile in einer Bibliothek notwendigen Speicherplatzes auf einer Disk.

FORMAT: CRE[ATE] filename, [lib ref], [ $\begin{pmatrix} S \\ R \\ Z \end{pmatrix}$ ][,n]

"filename" Name des zu erstellenden Files.

"lib ref" kann folgende Formen annehmen:

- (lib name, unit name)
- lib name
- (, unit name)

Als "lib name" ist der Bibliotheksname, als "unit name" der Name der Disketten-Diskstation anzugeben.

"S" definiert, daß das File "sequentiell" ist.

"R" gibt an, daß das File "random" ist. Es enthält numerische Daten in einfacher Genauigkeit, die den Wert "nicht initialisiert" aufweisen.

"Z" bezeichnet das File als "random". Es beinhaltet numerische Daten in einfacher Genauigkeit mit dem Wert 0 (Null).

"n" ganze positive Zahl.  
Sie bestimmt die Anzahl der vom System für das File zu reservierenden Bytes.

WIRKUNG: Der alle Parameter aufweisende Befehl reserviert n Bytes für das durch "filename" bezeichnete File in der durch "lib ref" angegebenen Bibliothek.

- Fehlt der 3. Operand, nimmt das System "S" an.
- Fehlt der 4. Operand, werden 4096 Bytes für das File auf der angegebenen Disk reserviert.

BEMERKUNGEN: - Die Anzahl Bytes (n) wird vom System automatisch auf ein Vielfaches von 256 aufgerundet.  
- Die angesprochene Bibliothek muß offen sein.

- In einer mit LBPROTECT geschützten Package-Teilbibliothek kann kein neues File angelegt werden.
- Ein und derselbe Filename kann auf verschiedenen Disks und auch in mehreren Bibliotheken verwendet werden. Wird jedoch ein Filename ein zweites Mal in der gleichen Bibliothek angegeben, erscheint die Fehlermeldung ERROR 186.
- Eine Bibliothek (bzw. Teilbibliothek) kann maximal die bei der Erstellung des Dienstprogrammes LBCREATE festgelegte Anzahl Files aufnehmen (ERROR 183).

#### BEISPIELE:

```
CRE *NAME1,(,F1),R,8192
```

In der Package-Teilbibliothek auf der Disk in der linken Station wird ein Randomfile mit 8 K-Bytes erstellt.

```
CRE SFILE
```

In der ersten offenen Bibliothek wird ein sequentielles File (SFILE) mit 4 K-Bytes erstellt.

BEFEHL: DATE

FUNKTION: Speicherung des eingetasteten Datums auf der Systemdisk zwecks automatischer Datierung von Operationen mit Files auf einer Disk.

FORMAT: DAT[E] Datum

"Datum" Folge von 6 beliebigen ISO-Zeichen, mit Ausnahme des Leerzeichens.

WIRKUNG: Bei jedem Laden des Systems (nach dem Einschalten oder nach Ausführung eines OPTIONS- oder CONFIGURE-Befehles) wird auch die letzte, als Datum auf der Systemdisk gespeicherte Zeichenfolge in den Hauptspeicher geladen.

BEMERKUNG: Die normalerweise verwendete Zeichenfolge lautet:

TTMMJJ (= Tag, Monat, Jahr)

Die Ausgabe der Zeichen erfolgt in drei Gruppen zu je zwei durch "-" getrennte Zeichen (z.B. "120379" = 12-03-79 oder "Juli 78" = Ju-li-78)

BEISPIELE:

DATE JULI78						
FILE	P	JU-LI-78	JU-LI-78	0256	0256	1

DATE 120379						
FILE	P	JU-LI-78	12-03-79	0256	0256	1





BEFEHL: DCHANGE (disk change)

FUNKTION: Austauschen von Disks bei geladenem System, ohne dadurch den Inhalt des Arbeitsspeichers zu löschen.

FORMAT: DCH[ANGE] unit name

"unit name" Name der Diskeinheit.

WIRKUNG: Die Bibliotheken in der durch "unit name" angegebenen Einheit werden geschlossen. Das System gibt die Display-Meldung "LOAD DISK ON unit name" aus und erlaubt damit den Austausch der Disk.

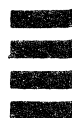
Nach dem Austausch ist die Funktion CONTINUE zu betätigen.

Sind weniger als sechs Bibliotheken offen und ist die erste Bibliothek auf der neuen Disk nicht durch ein Passwort geschützt, wird diese geöffnet und die Display-Meldung "LIBR lib name OPEN ON unit name" ausgegeben.

BEMERKUNGEN:

- Der Systembefehl DCHANGE muß
  - . vor jedem Austausch einer Disk
  - . und vor jedem Einsatz bisher nicht belegter Stationen verwendet werden. Andernfalls erscheint die Fehlermeldung "ABN unit name - DCH OMITTED". Diese Meldung ist mit CLEAR und Funktion CONTINUE zu löschen und DCHANGE nachzuholen.
- Befindet sich das Betriebssystem auf der Disk so ist nur ein Wechsel gleicher Release-Stufe und, bei reduziertem System, mit gleichem Inhalt möglich. Andernfalls muß das System neu eingeschaltet bzw. mit dem entsprechenden BOOTSTRAP neu geladen werden.





BEFEHL: DRAW

FUNKTION: Hardcopy des Bildschirminhaltes auf Thermodrucker.

FORMAT: DRA[W] [A]

WIRKUNG: A gibt an, daß vom Inhalt des Bildschirms, der durch den alphanumerischen Controller dargestellt wird, eine Hardcopy erstellt werden soll. Der aktuelle Inhalt des Bildschirms wird über den Thermodrucker ausgegeben. Der Ausdruck erfolgt im Verhältnis 1:1.12. Ist der Operand A nicht angegeben, so wird der Inhalt des graphischen Controllers ausgegeben.

BEMERKUNGEN: - Der Ausdruck kann mit der Funktion BREAK oder EXIT abgebrochen werden.  
- Das System muß für den Befehl DRAW ohne Operand A mit der Option GDI initialisiert sein.





BEFEHL: ENVIRONMENT

FUNKTION: Beschreibung der im Einsatz stehenden Konfiguration.

FORMAT: ENV[IRONMENT]

WIRKUNG: Folgende Informationen über die System-Konfiguration werden ausgegeben:

- Release-Code der Systemsoftware
- zuletzt mit DATE gespeichertes Datum
- physische Größe des Anwenderspeichers
- Art des Systemdruckers (alphanumerisch/graphisch)
- vorhandene Controller
- Spezifikationen gemäß zuletzt verwendetem CONFIGURE-Befehl (logische Größe des Arbeitsspeichers)
- momentan geladene Optionen
- Liste der für Disk-Einheiten zulässigen Namen
- Name der Systemdisk
- Liste der zur Zeit offenen Bibliotheken
- Aktuelle Liste der beim Einschalten der Anlage vom Betriebssystem automatisch geöffneten Bibliotheken (siehe auch LBSTORE, LBRESTORE)
- Art des Bildschirms (graphisch)

BEISPIEL:

```

ENV      MDOSC -R 2.0 *          SYSTEM ENVIRONMENT      * DATE:03-01-83
      .
      MEMORY PHYSICAL SIZE : 64 K
      SYSTEM PRINTER : GRAPHIC
      VIDEO DISPLAY : GRAPHIC
      I/O INTERFACES :

      MEMORY LOGICAL SIZE: 64 K
      SOFTWARE OPTIONS : GDISP

      DISK UNITS: NAME          PERIPH CODE      PERIPH TYPE
                  F1           C0              FDU
                  F2           C1              FDU

      OPERATING SYSTEM ON DISK UNIT F1

      OPEN LIBRARIES :   NAME          DISK UNIT
                        DATEN          F1
                        P6441          F2
                        HELP           F1
                        D6411          F2

      STORED LIBRARIES : NAME          DISK UNIT
                        DATEN          F1
                        P6441          F2
  
```



BEFEHL: ERASE

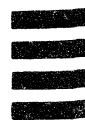
FUNKTION: Löschen des Bildschirminhaltes

FORMAT: ERA[SE]  $\left[ \begin{matrix} A \\ B \end{matrix} \right]$

WIRKUNG: A bestimmt den alphanumerischen Controller  
G bestimmt den graphischen Controller.

BEMERKUNG: Ist kein Operand angegeben, so wird der gesamte Bildschirm-  
inhalt gelöscht. Bei Angabe des Operanden A wird der Inhalt  
des alphanumerischen Controllers gelöscht. Bei Angabe des  
Operanden G wird der Inhalt des graphischen Controllers  
gelöscht.





BEFEHL: EXEC (execute)

FUNKTION: Laden eines Dienstprogrammes in den Arbeitsspeicher und dessen Ausführung.

FORMAT: EXE[C] Dienstprogramm[,Parameter[,Parameter]...]

"Dienst-  
programm" Name des Dienstprogrammes.

"Parameter" Definition der für das jeweilige Dienstpro-  
gramm spezifischen Operanden.

WIRKUNG: EXEC verlangt vom System das durch seinen Namen spezifizier-  
te Dienstprogramm. Dieses wird in den Arbeitsspeicher gela-  
den und ausgeführt.

BEMERKUNG: Durch den Befehl EXEC wird der Inhalt des Arbeitsspeichers  
gelöscht.

Liste der verfügbaren Dienstprogramme:

DCOPY  
DINIT  
FLCOPY  
FLPRINT  
LBCREATE  
LBEMPTY  
LBPROTECT  
LBRENAME  
LBSCRATCH  
LIBCOPY  
RESTRUCT  
VOLLABEL





BEFEHL: LBCLOSE (library close)

FUNKTION: Schließen von offenen Bibliotheken.

FORMAT:  $LBC[LOSE] \left\{ \begin{array}{c} \text{lib ref} \\ * \end{array} \right\}$

"lib ref" kann folgende Formen annehmen:  
 - (lib name,unit name)  
 - lib name  
 - (,unit name)  
 Als "lib name" ist der Bibliotheksname,  
 als "unit name" der Name der Diskstation  
 anzugeben.

"\*" Schließen aller offenen Bibliotheken.

WIRKUNG: Die durch "lib ref" bezeichnete Bibliothek wird geschlossen. Die Verwendung des Parameters "\*" bewirkt die Schließung aller momentan auf irgendeiner Disk offenen Bibliotheken.

BEMERKUNG: Ist die angegebene Bibliothek auf der spezifizierten Platteneinheit schon geschlossen oder nicht vorhanden, erfolgt eine entsprechende Fehlermeldung.

BEISPIELE:

LBC (CH2,D2)

Die Bibliothek CH2 auf der Disk F2 wird geschlossen.

LBC (CH3,D1)  
ERROR 178

Auf der Disk F1 ist keine offene Bibliothek des Namens CH3 vorhanden.





BEFEHL: LBOPEN (library open)

FUNKTION: Öffnen einer Bibliothek.

FORMAT: LBO[PEN] lib ref[,password]

"lib ref" kann folgende Formen annehmen:

- (lib name,unit name)
- lib name
- (,unit name)

Als "lib name" ist der Bibliotheksname, als "unit name" der Name Diskstation anzugeben.

"password" Ist die Bibliothek mit einem Password versehen, kann sie nur mit dem entsprechenden Kennwort geöffnet werden.

WIRKUNG: Die angegebene Bibliothek wird geöffnet und für die Bearbeitung freigegeben.

- BEMERKUNGEN:
- Gleichzeitig können maximal sechs Bibliotheken geöffnet sein. Wird der Befehl LBOPEN gegeben, obwohl schon sechs Bibliotheken offen sind oder die verlangte Bibliothek auf der genannten Platteneinheit nicht vorhanden ist, erfolgt eine Fehlermeldung.
  - Befindet sich die Bibliothek auf der Systemdisk, muß die Platteneinheit im Parameter "lib ref" nicht angegeben werden.
  - Für alle sich auf ein oder mehrere Files beziehenden Systembefehle muß die entsprechende Bibliothek offen sein (z.B. RUN, SAVE, CATALOG).
  - Mit dem Befehl LVT\* kann festgestellt werden, welche Bibliotheken offen sind.

BEISPIELE:

```

LBO (HELP,F1)

LBO (HELP1.D1),PASS2
ERROR 217

LBO FRANC
ERROR 218
  
```

falsches Password

Bibliothek "FRANC" befindet sich nicht auf der Systemdisk



- BEFEHL:** LBSTORE (library store)
- FUNKTION:** Die Namen der im Moment offenen Bibliotheken werden auf der Systemdisk gespeichert und diese in Zukunft automatisch geöffnet.
- FORMAT:** LBS[TORE]
- WIRKUNG:** Die Namen der zur Zeit offenen Bibliotheken sind nach Ausführung dieses Befehles auf der Systemdisk gespeichert. Bei jeder Initialisierung des Systems werden dann die in dieser Liste enthaltenen Bibliotheken automatisch geöffnet. Die Liste der Bibliotheksnamen wird ausgedruckt. Sie werden als Standard-Bibliotheken bezeichnet.
- BEMERKUNGEN:**
- Das System wird initialisiert durch
    - . das Einschalten der Anlage oder Drücken der RESET-Taste,
    - . die Ausführung des Befehles CONFIGURE oder OPTIONS.
  - Ist bei Ausführung von LBS keine Bibliothek offen, wird auch bei der Initialisierung des Systems keine Bibliothek geöffnet. Wurde LBS noch nie verwendet, so öffnet das System nach einer Initialisierung die erste nicht mit einem Password geschützte Bibliothek.
  - Ist eine Bibliothek mit einem Password versehen und wird sie mit LBSTORE als "Standard" definiert, so wird sie in Zukunft ebenfalls automatisch geöffnet, das heißt, die Absicherung mittels Password entfällt.
  - Durch den Befehl LVT\* kann die Liste der offenen Bibliotheken ausgedruckt werden.

BEISPIEL:

LUT*										
OPEN LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
STORED LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
LBO (.F1)										
LUT*										
OPEN LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
P6FSYS	F1									
STORED LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
<table border="1"> <tr> <td colspan="2">LBS</td> </tr> <tr> <td>NAME</td> <td>PERIPHERAL</td> </tr> <tr> <td>P6FSYS</td> <td>F2</td> </tr> <tr> <td>P6FSYS</td> <td>F1</td> </tr> </table>			LBS		NAME	PERIPHERAL	P6FSYS	F2	P6FSYS	F1
LBS										
NAME	PERIPHERAL									
P6FSYS	F2									
P6FSYS	F1									
LUT*										
OPEN LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
P6FSYS	F1									
STORED LIBRARIES :										
NAME	DISK UNIT									
P6FSYS	F2									
P6FSYS	F1									

BEFEHL: LVTOC (library volume table of contents)

FUNKTION: Druck eines Verzeichnisses der Bibliotheken einer Einheit oder aller offenen Bibliotheken.

FORMAT: LVT[OC]  $\left[ \begin{array}{c} \{ \text{unit name} \} \\ * \end{array} \right]$

"unit name" Name der entsprechenden Disk-Einheit.

"\*" Ausgabe der Liste aller offenen sowie der Standard-Bibliotheken.

WIRKUNG: Durch die Angabe des Parameters "unit name" werden die vorhandenen Bibliotheken mit folgenden Angaben gelistet:

- Überschrift mit Release-Code des Betriebssystems, Volume-Label, Sektorlänge und Datum (Befehl DATE).
- Für jede Bibliothek:
  - . Name der Bibliothek (LIBRARY)
  - . Erstellungsdatum (CREATE)
  - . Physische Anfangs- und Endadresse durch Angabe von Zylinder (erste 3 Ziffern) und Sektor (letzte 3 Ziffern)
  - . Anzahl der reservierten Sektoren (SECTORS)
  - . Anzahl der noch freien Sektoren innerhalb der Bibliothek (EMPTY SECTORS)
- Gesamtzahl der noch freien Sektoren dieser Einheit (FREE SPACE) nach Ausführung des Dienstprogrammes RESTRUCT.

Bei Verwendung des Parameters "\*" werden alle im Moment offenen und die Liste der mit LBS registrierten Standard-Bibliotheken ausgedruckt.

BEMERKUNG: - Wird kein Parameter angegeben, erfolgt der Druck des Bibliothekskataloges der Systemdisk.

BEISPIEL:

```

LUT F1
MDOSC -R 2.0 * VOLLABEL=          * TRACK FORMAT=256 BYTE * DATE:03-01-83

LIBRARY  CREAT      BEG OF EXT      END OF EXT      SECTORS      EMPTY SECTORS
OSLIB    27-10-82    01001          16052          832
L1ESEP    27-10-82    17001          17014          14
L1ESES    27-10-82    17015          32024          790
HELP      03-01-83    32025          52008          1024          34
DATEN     03-01-83    52009          59044          400           388
FREE SPACE (TOTAL) = 788 SECTORS

LUT F2
MDOSC -R 2.0 * VOLLABEL=L1/ESE * TRACK FORMAT=256 BYTE * DATE:03-01-83

LIBRARY  CREAT      BEG OF EXT      END OF EXT      SECTORS      EMPTY SECTORS
D6435    07-06-82    01001          15044          772          15
D6432    07-06-82    15045          32032          872          23
P6441    07-06-82    32033          41012          448          11
D64351   07-06-82    41013          56024          792          40
D64321   07-06-82    56025          73012          872          23
D6411    07-10-82    73013          74040          80           0
FREE SPACE (TOTAL) = 12 SECTORS

LUT*
OPEN LIBRARIES :
NAME
DATEN
P6441
HELP
D6411
DISK UNIT
F1
F2
F1
F2

STORED LIBRARIES :
NAME
DATEN
P6441
DISK UNIT
F1
F2

```

BEFEHL:           MODIFY

FUNKTION:        Änderung eines Filenamens und/oder des für ein Datenfile reservierten Speicherplatzes in einer Bibliothek.

FORMAT:           MOD[IFY] filename<sub>1</sub>,[lib ref],[filename<sub>2</sub>][,n]

"lib ref"           kann folgende Formen annehmen:  
                     - (lib name,unit name)  
                     - lib name  
                     - (,unit name)  
                     Als "lib name" ist der Bibliotheksname,  
                     als "unit name" der Name der Diskstation  
                     anzugeben.

"filename<sub>1</sub>"       Name des zu modifizierenden Files.

"filename<sub>2</sub>"       neuer Filename.

"n"                 positive ganze Zahl, die den für das File  
                     in der Bibliothek neu zu reservierenden  
                     Speicherplatz festlegt.

WIRKUNG:         Der vollständige Befehl mit allen vier Operanden bewirkt,  
                     daß der alte Name des Files in der Bibliothek durch den  
                     neuen Namen ersetzt und der Speicherbereich auf der Disk  
                     neu auf n Bytes geändert wird.

- Fehlt der Parameter "n", wird nur der Name des Programm-,  
     Text- oder Datenfiles geändert, nicht aber seine Kapazität.
- Fehlt der Parameter "filename 2", behält das File seinen  
     Namen, doch werden neu n Bytes auf der Disk für das File  
     reserviert.

BEMERKUNGEN:    - Mit MODIFY geänderte Filenamen behalten ihre Zugehörigkeit  
                     zu einer bestimmten Teilbibliothek. Daher muß die Kenn-  
                     zeichnung + oder \* für "filename 2" nicht eingegeben  
                     werden.

- Um zu verhindern, daß Daten verloren gehen, kann der  
     für sequentielle Datenfiles reservierte Platz durch MODIFY  
     nur bis zum bereits belegten Speicherplatz verkleinert  
     werden.

- Da das logische und das physische Ende von Random-Files zusammenfallen, kann der für ein Random-File reservierte Speicherplatz nicht verkleinert werden.
- Zwei Files einer Bibliothek dürfen nie den gleichen Namen tragen.
- Bei mit dem Dienstprogramm LBPROTECT geschützten Files von Package- und Common-Teilbibliotheken kann nur die Größe modifiziert werden.

BEISPIEL: Änderung des Namens und der reservierten Länge eines Random-Files.

MDOSC -R 2.0 * VOLLABEL = * LIBRARY = P6FSYS * DATE:07-05-80								
FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT	
QUATSH	P	13-03-80	13-03-80	0384	0384		1	
TEST	P	13-03-80	13-03-80	0384	0384		1	
TEST1	P	13-03-80	13-03-80	0384	0384		1	
SYS	P	15-12-78	20-06-79	2432	2432		1	
YYY	P	13-03-80	13-03-80	0256	0256		1	C=1000
TEXT1	P	13-03-80	13-03-80	0256	0256		1	
FUNK1	T	07-05-80	07-05-80	0512	0512		1	
→ DATEN	R	07-05-80	07-05-80	0512	0512		1	

MOD DATEN, C,F2),DATE11,1000

MDOSC -R 2.0 * VOLLABEL = * LIBRARY = P6FSYS * DATE:07-05-80								
FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT	
QUATSH	P	13-03-80	13-03-80	0384	0384		1	
TEST	P	13-03-80	13-03-80	0384	0384		1	
TEST1	P	13-03-80	13-03-80	0384	0384		1	
SYS	P	15-12-78	20-06-79	2432	2432		1	
YYY	P	13-03-80	13-03-80	0256	0256		1	C=1000
TEXT1	P	13-03-80	13-03-80	0256	0256		1	
FUNK1	T	07-05-80	07-05-80	0512	0512		1	
→ DATE11	R	07-05-80	07-05-80	1024	1024		1	





BEFEHL: OPTIONS

FUNKTION: Vorwahlbefehl für die Aktivierung bestimmter Teile des Betriebssystems und Neuladen des Systems.

FORMAT: 
$$\text{OPT}[\text{IONS}] \left[ \left\{ \begin{array}{c} \text{GDI} \\ \text{PLO} \end{array} \right\} \right]$$

WIRKUNG: Die im Befehl aufgeführten Module (= Teile) des Betriebssystems werden in den Anwenderteil des Hauptspeichers geladen.

- Parameter PLO:  
wird zur Ausführung von Plotanweisungen für den Systemdrucker benötigt (Thermodrucker).
- Parameter GDI:  
erlaubt die Ausgabe von graphischen Darstellungen auf dem Bildschirm.

Die Eingabe des Befehles ohne Parameter bewirkt, daß die Markierung der bisher geladenen Options auf der Systemdisk gelöscht wird. Das System wird ohne Options neu initialisiert.

- BEMERKUNGEN:
- Die bei der Initialisierung des Systems mitzuladenden Options sind auf der Systemdisk gespeichert. Bei jeder Neuinitialisierung, das heißt nach dem Einschalten sowie nach CON, OPT oder einem Fehler mit anschließendem Systemabbruch (ERROR n\*A), konfiguriert sich das System entsprechend dem zuletzt ausgeführten OPTIONS-Befehl.
  - Durch einen OPTIONS-Befehl wird der Inhalt des Arbeitsspeichers gelöscht.
  - Die Options werden nur bei der Ausführung von Anweisungen benötigt. Die Eingabe und die syntaktische Kontrolle kann auch bei nicht geladenen Options erfolgen.

- Die Options benötigen zusätzlichen Platz im Anwenderspeicher gemäß nachfolgender Tabelle:

OPTIONS	PLATZBEDARF 1 K = 1024 bytes
PLO	2 K
GDI	2,5 K

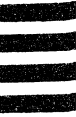
- Die folgende Aufstellung gibt Auskunft, wann die beiden Options benötigt werden:

PLO - Plot:

Alle BASIC-Anweisungen zur Verwendung des Thermodruckers als Plotter.

GDI - Graphic Display:

Alle BASIC-Anweisungen und Befehle für den Bildschirm DSM, mit Ausnahme der BASIC-Anweisungen DISP, ERASE, REVERSE sowie der Debugging- und Systembefehle ERASE, REVERSE, DRAW.



BEFEHL: PROCEDURE

FUNKTION: Abruf einer Katalog-Prozedur zur Verarbeitung.

FORMAT: PRO[CEDURE] [filename,[lib ref],[A],[Zeilennummer<sub>1</sub>][,Zeilennummer<sub>2</sub>]]

"filename" Name des Commandfiles.

"lib ref" kann folgende Formen annehmen:  
 - (lib name,unit name)  
 - lib name  
 - (,unit name)  
 Als "lib name" ist der Bibliotheksname,  
 als "unit name" der Name der Diskstation  
 anzugeben.

"A" Definition, daß mit automatischem Input  
gearbeitet wird.

"Zeilennummer<sub>1</sub>" erste zu verarbeitende Zeile der Prozedur.

"Zeilennummer<sub>2</sub>" letzte zu verarbeitende Zeile der Prozedur.

WIRKUNG: Sind alle Parameter angegeben, wird das durch "filename"  
 und "lib ref" definierte Commandfile verarbeitet. Die Ausführung  
 beginnt mit "Zeilennummer1" und endet nach der Bearbeitung  
 von "Zeilennummer2". Die zur Ausführung von INPUT-  
 bzw. RKB-Statements notwendigen Eingabewerte werden den  
 entsprechenden "IN=" -Zeilen der Prozedur entnommen.

Ist "Zeilennummer1" nicht angegeben, so beginnt die Verarbeitung  
 mit der ersten Zeile der Prozedur.

Fehlt "Zeilennummer2", so endet die Arbeit nach der letzten  
 Zeile der Prozedur.

Fehlt die Angabe "A", so hat die Dateneingabe manuell zu  
 erfolgen. In der Prozedur enthaltene "IN=" -Zeilen werden  
 ignoriert.

Ist kein Parameter angeführt, so wird die Verarbeitung  
 einer abgebrochenen Prozedur wieder aufgenommen.

Aufbau und spezielle Befehle einer Prozedur werden in Kapitel  
 6.2 und 6.4 ausführlich beschrieben.

- BEMERKUNGEN:
- Jede Zeile der Prozedur wird im Bildschirm angezeigt und, wenn PRINT ALL aktiviert ist, ausgedruckt.
  - Nach Beendigung der Prozedur befindet sich das System im Command-Mode.
  - Mit der Funktion BREAK kann die Verarbeitung der Prozedur abgebrochen werden. Das System befindet sich danach im Command-Mode.
  - Enthält eine Zeile der Prozedur einen syntaktischen Fehler, erscheint eine Fehlermeldung und die Verarbeitung wird abgebrochen.
    - . Ist der Fehler behebbar, kann dieser wie üblich im Debugging-Mode behoben und die Verarbeitung der Prozedur danach mit der Funktion CONTINUE fortgesetzt werden.
    - . Ist der Fehler nicht behebbar, kann der Debugging-Mode nur mit der Funktion BREAK verlassen werden. Die Prozedur ist damit vorzeitig beendet und das System befindet sich im Command-Mode.
  - Befindet sich das System aufgrund eines bei der Ausführung der Prozedur aufgetretenen Fehlers im Command-Mode, kann die Verarbeitung der Prozedur durch Eingabe des Befehles PROCEDURE ohne Operanden wieder aufgenommen werden. Begonnen wird in diesem Falle mit der fehlerhaften Zeile unmittelbar folgenden Zeile.
  - Beinhaltet die Prozedur einen OPTIONS- oder CONFIGURE-Befehl, wird sie nach Ausführung des Befehles nicht fortgesetzt.
  - Eine Prozedur kann mit PRO-Befehlen andere Prozeduren aufrufen.
  - Enthält die Prozedur eine Zeile mit "IF CC = X, Zeilennummer", worin als "Zeilennummer" ein höherer Wert, als der im Aufruf für "Zeilennummer2" definierte, angegeben ist, wird die Prozedur bei dieser Zeile trotzdem fortgesetzt und beendet, wenn entweder die letzte Zeile der Prozedur oder "Zeilennummer2" erreicht ist.
  - Tritt bei der Verarbeitung einer Prozedur ein Fehler auf, kann das Datenfile im Arbeitsspeicher mit
 

```
TRA T, filename, [lib ref],
```

 in ein Textfile umgewandelt werden. Nach der Korrektur mit den Editing-Möglichkeiten des Textfiles ist es möglich, das Datenfile nach
 

```
PUR filename, [lib ref]
```

 mit
 

```
TRA D, filename, [lib ref],
```

 in der korrigierten Fassung wieder zu speichern. Die Prozedur kann mit PRO fortgesetzt werden.

- Der Befehl TEST innerhalb einer Prozedur ermöglicht das automatische Übergehen fehlerhafter Zeilen (siehe Kapitel 6.4).
- Wird eine Prozedur unter dem Filenamen \*SETUP abgespeichert, kann sie bei der Initialisierung des Systems automatisch gestartet werden (siehe Befehl CONFIGURE PRO).

BEISPIELE:     Commandfile CMD 1

```
FILE

0010 OLD PYT
0020 LIST
0030 RUN PYT
0040 IN=2.5
0050 IN=7.4
0060 45PRINT"A=";A,"B=";B
0070 RUN
0080 IN=3.5
0090 IN=5.5
0100 LIST

END OF LISTING
```

### Beispiel 1:

Aufruf ohne automatischen Input (PRINT ALL aktiviert)

```
PRO CMD1

OLD PYT
LIST
FILE      PYT

0010 DISP "EINGABE VON A";
0020 INPUT A
0030 DISP "EINGABE VON B"
0040 INPUT B
0050 LET C=SQR(A*A+B*B)
0060 PRINT "C=";C
0070 PRINT
0080 PRINT
0090 END

END OF LISTING

RUN PYT
**** FORMALLY CORRECT PROGRAM ****
EINGABE VON A?
2
EINGABE VON B
?
3
C= 3.6055513

45PRINT"A=";A,"B=";B
RUN
**** FORMALLY CORRECT PROGRAM ****
EINGABE VON A?
2
EINGABE VON B
?
3
A= 2          B= 3
C= 3.6055513

LIST
FILE      PYT

0010 DISP "EINGABE VON A";
0020 INPUT A
0030 DISP "EINGABE VON B"
0040 INPUT B
0045 PRINT "A=";A,"B=";B
0050 LET C=SQR(A*A+B*B)
0060 PRINT "C=";C
0070 PRINT
0080 PRINT
0090 END

END OF LISTING
```

## Beispiel 2:

Aufruf mit automatischem Input

```
PRO CMD1,,A
OLD PYT
LIST
FILE      PYT

0010 DISP "EINGABE VON A";
0020 INPUT A
0030 DISP "EINGABE VON B"
0040 INPUT B
0050 LET C=SQR(A+A+B*B)
0060 PRINT "C=";C
0070 PRINT
0080 PRINT
0090 END

END OF LISTING

RUN PYT
**** FORMALLY CORRECT PROGRAM ****
EINGABE VON A?
2.5
EINGABE VON B
?
7.4
C= 7.8108898

45PRINT"A=";A,"B=";B
RUN
**** FORMALLY CORRECT PROGRAM ****
EINGABE VON A?
3.5
EINGABE VON B
?
5.5
A= 3.5      B= 5.5
C= 6.5192024

LIST
FILE      PYT

0010 DISP "EINGABE VON A";
0020 INPUT A
0030 DISP "EINGABE VON B"
0040 INPUT B
0045 PRINT "A=";A,"B=";B
0050 LET C=SQR(A+A+B*B)
0060 PRINT "C=";C
0070 PRINT
0080 PRINT
0090 END

END OF LISTING
```





BEFEHL: PURGE

FUNKTION: Löschen eines Files in einer Bibliothek.

FORMAT: PUR[GE] filename [,lib ref]

"filename" Name des zu löschenden Files.  
 "lib ref" kann folgende Formen annehmen:  
 - (lib name,unit name)  
 - lib name  
 - (,unit name)  
 Als "lib name" ist der Bibliotheksname,  
 als "unit name" der Name der Diskstation  
 anzugeben.

WIRKUNG: Das System prüft, ob in der durch "lib ref" definierten Bibliothek ein File namens "filename" vorhanden ist. Wenn ja, wird dieses File gelöscht. Der dadurch frei gewordene Platz kann für die Speicherung weiterer Files wieder verwendet werden.

BEMERKUNGEN: - Ist der Filename in verschiedenen Bibliotheken vorhanden und der Parameter "lib ref" nicht definiert, so wird das File mit dem entsprechenden Namen in der ersten geöffneten Bibliothek (LVT\*) gelöscht.  
 - Für Files von mit dem Dienstprogramm LBPROTECT geschützten Package- oder Common-Teilbibliotheken ist der Befehl unwirksam.  
 - Die mit "lib ref" angesprochene Bibliothek muß offen sein.

BEISPIEL:

```
PUR TEST1
PUR FUNK1,(F2)
```



BEFEHL: REVERSE

FUNKTION: Umkehrung des Bildes auf dem Bildschirm von positiv in negativ und umgekehrt.

FORMAT: REV[ERSE]

WIRKUNG: Darstellungen auf dem Bildschirm werden von positiv in negativ umgewandelt und umgekehrt.

BEMERKUNGEN: 

- Die Hardcopy mit DRAW auf dem Thermoprinter erscheint immer als positiv.
- Der Befehl hat immer eine Umkehrwirkung, gleichgültig, ob von einem Programm ein positives oder negatives Bild erstellt wurde.



BEFEHL: RUN

FUNKTION: Start der Programmausführung.

FORMAT: RUN  $\left[ \begin{array}{l} \text{filename[,lib ref]} \\ \text{Zeilennummer} \end{array} \right]$

"filename" Name des zu verarbeitenden Files.

"lib ref" kann folgende Formen annehmen:

- (lib name,unit name)
- lib name
- (,unit name)

Als "lib name" ist der Name der Bibliothek, als "unit name" der Name der Diskstation anzugeben.

"Zeilennummer" Zeile, mit welcher die Verarbeitung beginnen soll.

- WIRKUNG:
- Wird der Operand "filename" eingegeben, überprüft das System, ob ein File dieses Namens in der durch "lib ref" angesprochenen Bibliothek gespeichert ist. Die Suche nach diesem File beginnt immer in der ersten offenen Bibliothek (LVT\*), sofern der Parameter "lib ref" nicht definiert ist. Ist das File ein Programm oder Textfile, wird es in den Arbeitsspeicher geladen. Handelt es sich um ein Programm, wird dieses ausgeführt. Andernfalls erscheint eine Fehlermeldung.
  - Besteht die Eingabe aus dem Befehl "RUN" oder "RUN Zeilennummer", wird geprüft, ob sich ein Programm im Arbeitsspeicher befindet. Wenn nicht, erfolgt eine Fehlermeldung. Wenn ja, wird das Programm ab der definierten Zeilennummer ausgeführt.

- BEMERKUNGEN:
- Wird "RUN Zeilennummer" eingegeben, beginnt die Ausführung mit der angegebenen Zeile. Die Zeilennummer darf sich jedoch nicht auf eine Anweisung innerhalb einer FOR/NEXT-Schleife oder einer Multi-Line-Funktion beziehen.
  - Nach Eingabe eines RUN-Befehles prüft das System, ob eine Preexecution durchzuführen ist. Unter "Preexecution" versteht man eine Prüfung des Programmes durch das System

Eine Preexecution wird nur durchgeführt, wenn:

- . ein neu eingegebenes Programm zum ersten Mal ausgeführt werden soll;
  - . bei der letzten Preexecution ein Fehler festgestellt wurde;
  - . ein Programm nach dem letzten RUN- oder PREPARE-Befehl geändert worden ist;
  - . das nach "RUN filename" gestartete Programm ohne vorgängige, fehlerfreie Preexecution gespeichert worden ist.
- Eine erneute Preexecution eines Programmes beim Aufruf durch "RUN filename" unterbleibt, wenn folgende Befehle ausgeführt werden:
    - . OLD filename [,lib ref]
    - . PREPARE der Funktion
    - . Ausführung der Funktion BREAK nach fehlerfreier Preexecution
    - . REPLACE [,lib ref]
  - Tritt während der Preexecution ein Fehler auf, wird dieser gemeldet und das Programm nicht ausgeführt. Fehlermeldungen, die sich auf fehlende Files, Platzprobleme bezüglich der Disk oder den nötigen Platzbedarf im Anwenderspeicher beziehen, erfolgen erst nach Abschluß der Preexecution, das heißt, die Rückspeicherung nach BREAK ist trotzdem möglich.
  - Ist die Taste PRINT ALL aktiviert, wird nach der fehlerfreien Preexecution die Meldung "\*\*\*\*\* FORMALLY CORRECT PROGRAM\*\*\*\*\*" ausgegeben.
  - Wird ein durch die Preexecution bereits als fehlerfrei bezeichnetes Programm erneut gestartet, unterbleibt eine erneute Preexecution.
  - Zwecks Optimierung wird das Programm vor der eigentlichen Verarbeitung in eine Form gebracht, die die Anwendung von Befehlen, wie LIST, FETCH und ähnliche, nicht erlaubt. Daher wird vom System eine Kopie der ursprünglichen Version in der ersten offenen Bibliothek zwischengespeichert und nach der Ausführung oder einem Abbruch mit der Funktion BREAK wieder in den Arbeitsspeicher geladen. Ist in der entsprechenden Bibliothek nicht genügend Platz dafür vorhanden oder keine Bibliothek offen, erscheint die Fehlermeldung ERROR 188.

Die Zwischenspeicherung unterbleibt, wenn der Befehl in Form von "RUN filename" eingegeben wird. In diesem Falle lädt das System nach Ausführung des Programmes die gespeicherte Version erneut in den Arbeitsspeicher.

- Enthält das Programm eine COMMON-Anweisung, kann mit RUN ohne Parameter nur gestartet werden, sofern das Programm im Arbeitsspeicher schon einen Namen hat.
- Die Meldung "PROGRAM filename RUNNING" kann durch die Eingabe des Parameters "MSG" im Befehl SAVE unterdrückt werden.

- Die Programmausführung kann durch Ausführung der Funktion BREAK abgebrochen werden. Begonnene In-/Outputoperationen werden noch regulär beendet und eventuell geöffnete Files geschlossen. Das System geht in den Command-Mode.
- Mit der Taste EXIT kann die Programmausführung unterbrochen und das System in den Debugging-Mode versetzt werden.

#### BEISPIEL:

```

OLD TEST
RUN
**** FORMALLY CORRECT PROGRAM ****

RUN

OLD TEST
PRE
**** FORMALLY CORRECT PROGRAM ****
:: ROOM=15942 W ::
REP

RUN TEST

```

Bei der ersten Ausführung wird die Preexecution durchgeführt.

Bei der zweiten Ausführung entfällt die Preexecution.

Abspeichern des Programmes mit durchgeführter Preexecution (d.h., daß diese bei weiteren Programmläufen entfällt).





BEFEHL: SPACE

FUNKTION: Abfrage des freien Platzes einer Bibliothek.

FORMAT: SPA[CE] [lib ref]

"lib ref" kann folgende Formen annehmen:

- (lib name,unit name)
- lib name
- (,unit name)

Als "lib name" ist der Bibliotheksname, als "unit name" der Name der Diskstation anzugeben.

WIRKUNG: Der freie Platz in der mit "lib ref" angegebenen Bibliothek wird im Bildschirm angezeigt. Die Angabe erfolgt in Bytes.

BEMERKUNG: Wird in einer Bibliothek ein File angelegt (z.B. durch SAVE, CREATE), so wird vom Betriebssystem außer dem im Katalog ausgewiesenen Platz ein Sektor (256 Bytes) für Kontrollinformationen belegt.

BEISPIEL:

```
SPA (,F1)
SPACE = 142336
```





BEFEHL: VALIDATE

FUNKTION: Schließen eines offen gebliebenen Datenfiles.

FORMAT: VAL[IDATE] ,filename[,lib ref]

"filename" Name des zu schließenden Files.

"lib ref" kann folgende Formen annehmen:

- (lib name,unit name)
- lib name
- (,unit name)

Als "lib name" ist der Bibliotheksname,  
als "unit name" der Name der Diskstation  
anzugeben.

WIRKUNG: Das aufgrund eines nicht regulären Programmendes (z.B. Stromausfall) offen gebliebene Datenfile namens "filename" wird geschlossen.

BEMERKUNGEN: - Wurde das File beschrieben, so können einige der zuletzt eingegebenen Daten eines Randomfiles zerstört sein. Bei sequentiellen Files ist nach dem Schließen die vor dem Öffnen vorhandene aktuelle Länge gespeichert. Die Daten des Files sind nur entsprechend der gespeicherten, effektiven Länge des Files verfügbar, das heißt, daß alle Werte verloren sind, die zwischen dem letzten Programmstart und dessen irregulärem Abbruch eingegeben wurden.

- Offen gebliebene Files werden im Katalog mit "OPEN" gekennzeichnet.

BEISPIEL:

CAT TEST								
TEST	S	07-05-80	07-05-80	1024	0000		1	OPEN
VAL TEST								
CAT TEST								
TEST	S	07-05-80	07-05-80	1024	0000		1	



7.	<u>Dienstprogramme</u>	7.1
7.1	Allgemeine Hinweise	7.1
7.2	Liste der Dienstprogramme	7.1
	DCOPY	7.3
	DINIT	7.5
	FLCOPY	7.7
	FLPRINT	7.11
	LBCREATE	7.13
	LBEMPTY	7.15
	LBPROTECT	7.17
	LBRENAME	7.19
	LBSCRATCH	7.21
	LIBCOPY	7.23
	RESTRUCT	7.27
	VOLLAB	7.29



## 7. Dienstprogramme

### 7.1 Allgemeine Hinweise

Im folgenden Abschnitt wird die Funktion der im Betriebssystem enthaltenen Dienstprogramme beschrieben.

Der Aufruf eines Dienstprogrammes erfolgt unter Angabe des Namens und der erforderlichen Parameter mit dem Befehl EXEC.

Im Gegensatz zur Ausführung von Systembefehlen wird bei der Ausführung von Dienstprogrammen der Arbeitsspeicher benötigt. Soll der bisherige Inhalt des Arbeitsspeichers erhalten bleiben, ist vor Aufruf eines Dienstprogrammes der Inhalt des Arbeitsspeichers mit SAVE oder REPLACE abzuspeichern.

#### Bemerkung:

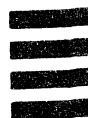
Der Begriff "Systemdisk" steht im folgenden für diejenige Disk, die das aktuelle Betriebssystem enthält.

### 7.2 Liste der Dienstprogramme

DCOPY	Kopieren von Disks.
DINIT	Initialisieren einer Disk.
FLCOPY	Kopieren von Files.
FLPRINT	Ausdruck eines Datenfiles.
LBCREATE	Anlegen einer Bibliothek auf einer Disk.
LBEMPTY	Löschen aller Files einer Bibliothek.
LBPROTECT	Schützen von Teilbibliotheken.
LBRENAME	Ändern eines Bibliotheksnamens.
LBSCRATCH	Löschen einer Bibliothek.
LIBCOPY	Kopieren einer Bibliothek.
RESTRUCT	Neuorganisation des Diskinhaltes nach dem Löschen einer Bibliothek.
VOLLABEL	Kennzeichnung einer Disk mit dem Volume-Label.







DIENST-  
PROGRAMM: DCOPY (disk copy)

FUNKTION: Kopieren von Disks.

FORMAT: EXE[C] DCO[PY], unit name<sub>1</sub>, unit name<sub>2</sub>[[volume label][,V]]

"unit name<sub>1</sub>" Input-Einheit (Original).  
 "unit name<sub>2</sub>" Output Einheit (Kopie).  
 "volume label" Kennzeichnung der Disk.  
 "V" "volume label" ebenfalls kopieren.

WIRKUNG:

- Der gesamte Inhalt der Disk wird ohne Änderung der logischen Struktur kopiert. Die Kopie kann nur auf artgleiche Platten erfolgen.
- Ist der Parameter "volume label" angegeben, so wird geprüft, ob die Output-Disk den im Aufruf angegebenen Volume-Label enthält. Bei Nichtübereinstimmung wird keine Kopie erstellt.
- Ist der Parameter "V" spezifiziert, so wird der Volume-Label des Originals übernommen.

MELDUNGEN:

- Nach dem Aufruf des Dienstprogrammes "DCOPY" erscheint im Display die Meldung "DISK ON unit name1 unit name2". Sie erlaubt eine nochmalige Kontrolle, ob die Wahl der Platteneinheiten richtig war:
  - . richtig: CONTINUE
  - . falsch: BREAK
- Nach Beendigung des Kopiervorgangs erscheint die Anzeige "RESTORE DISKS". Bei Bedarf können nun die am Kopiervorgang beteiligten Disks ausgewechselt werden. Mit CONTINUE wird der Command-Mode wieder erreicht.
- Die Meldung "LOAD DISK ON unit name" zeigt an, daß die Disk zur Aufnahme der Kopie in der genannten Station fehlt. Die Disk kann eingelegt und das Dienstprogramm mit CONTINUE fortgesetzt werden.

- ERROR n [-RESTORE DISK]

Der Kopiervorgang wurde aufgrund des Fehlers n abgebrochen (siehe Liste der Fehlermeldungen). Erscheint gleichzeitig die Meldung "RESTORE DISKS", so können die für den Kopiervorgang verwendeten Disks auf ihre Funktionsfähigkeit geprüft und allenfalls ausgetauscht werden. Der Kopiervorgang ist nicht durchgeführt.

ACHTUNG:

Betriebssysteme werden immer mitkopiert. Erfolgt eine Kopie auf eine Disk, die das aktuelle Betriebssystem enthält, so wird dieses gelöscht. Es muß dann entweder eine andere Systemdiskette eingelegt oder das System neu geladen werden.

BEISPIELE:

Kopieren von Diskettenstation F2 auf F1.

```
EXE DCO,F2,F1
DISK ON F1    F1
RESTORE DISKS
END OF LISTING
```

Kopieren bei Monofloppyeinheit (im Beispiel mit Stationsbezeichnung F1)

```
EXE DCO,F1,F1
DISK ON F1    F1
RESTORE DISKS
END OF LISTING
```

DIENST-  
PROGRAMM: DINIT

FUNKTION: Initialisieren einer Disk.

FORMAT: EXE[C] DIN[IT], unit name  $\left[ , \left[ \text{volume label} \right] \left[ , L \right] \right]$

"unit name" Name der Einheit, die die zu initialisierende Disk enthält.

"volume label" gewünschte Kennzeichnung der zu initialisierenden Disk.

"L" logische Initialisierung.

WIRKUNG:

- Ist der Parameter "L" angegeben, werden alle Bibliotheken der Disk gelöscht.
- Ist der Parameter "volume label" angegeben, so wird dieser auf die entsprechende Disk geschrieben.
- Fehlt der Operand "volume label", erhält die entsprechende Disk den Standardnamen "MDOS".

MELDUNG: Nach dem Abruf des Dienstprogrammes, jedoch vor Beginn der Initialisierung, erscheint die Display-Meldung "ACTION ON UNIT unit name". Die Initialisierung kann darauf mit CONTINUE gestartet oder mit BREAK verhindert werden. Das Ende der Initialisierung wird durch "READY" angezeigt.

BEMERKUNGEN:

- Die aktuelle Systemdisk kann mit DINIT nicht gelöscht werden.
- Jede neue Disk muß vor Ihrer Verwendung auf dem System mit DINIT initialisiert werden.

- Mit DINIT kann ein nicht mehr benötigtes Betriebssystem auf einer Disk gelöscht werden.

BEISPIEL:

```
EXE DIN,F1,DISK1,L
ACTION ON UNIT F1    ?
READY
```

LUT F1

ND05C -R 2.0 \* VOLLABEL=DISK1 \* TRACK FORMAT=256 BYTE \* DATE:07-05-80

LIBRARY	CREAT	BEG OF EXT	END OF EXT	SECTORS	EMPTY SECTORS
FREE SPACE (TOTAL) = 1898 SECTORS					
MAXIMUM BLOCK OF FREE SPACE = 1898 SECTORS					



DIENST-  
PROGRAMM: FLCOPY (file copy)

FUNKTION: Kopieren eines Files aus einer (Teil-)Bibliothek in eine andere (oder in die gleiche) Bibliothek.

FORMAT: EXE[C] FLC[OPY], IN=[lib ref<sub>1</sub>],[password<sub>1</sub>],filename<sub>1</sub>,  
OUT=[lib ref<sub>2</sub>],[password<sub>2</sub>] $\left[ \left\{ \begin{array}{c} \text{filename}_2 \\ * \\ + \end{array} \right\} \right]$

"lib ref" kann folgende Formen annehmen:  
- (lib name,unit name)  
- lib name  
- (,unit name)  
Als "lib name" ist der Bibliotheksname,  
als "unit name" der Name der Disk anzugeben.

"IN=lib ref<sub>1</sub>" gibt die das zu kopierende File enthaltende (Teil-)Bibliothek an.

"OUT=lib ref<sub>2</sub>" bestimmt die (Teil-)Bibliothek, in die das File kopiert werden soll.

"filename<sub>1</sub>" Name des zu kopierenden Files.

"filename<sub>2</sub>" Name, den die Kopie erhalten soll.

"\*" spezifiziert die Package-Teilbibliothek.

"+" spezifiziert die Common-Teilbibliothek.

"password<sub>1</sub>" Password für die Bibliothek mit dem Original-file.

"password<sub>2</sub>" Password für die Bibliothek, die die Kopie aufnimmt.

WIRKUNG: - Werden alle Parameter angegeben, so gelten für den letzten Parameter folgende Regeln:

- Ist "filename2" spezifiziert, so wird geprüft, ob in der Empfängerbibliothek bereits ein File gleichen Namens existiert. Wenn ja, erscheint eine Fehlermeldung. Andernfalls wird kopiert und die Kopie erhält den "filename2".

- . Bei Angabe von "\*" wird das File unter seinem bestehenden Namen in die Package-Teilbibliothek kopiert.
- . Durch Eingabe von "+" wird das File unter seinem Originalnamen in die Common-Teilbibliothek kopiert.
- Fehlt der letzte Parameter, wird das File unter seinem alten Namen in die User-Teilbibliothek aufgenommen.
- Ist "lib ref1"="lib ref2", wird das File in die Bibliothek kopiert, die auch das Original enthält.
- Die übrigen Wirkungen von "lib ref" sind aus den nachstehenden Tabellen ersichtlich:

IN=	"filename" wird gesucht in:	
ohne lib-ref	1. Bibliothek der Systemdisk	1)
lib name	der entsprechenden Bibliothek der Systemdisk	
,unit name	allen offenen Bibliotheken in der Einheit "unit name"	1)
lib name, unit name	der angegebenen Bibliothek in der Einheit "unit name"	

OUT=	File wird kopiert in:	
ohne lib ref	1. Bibliothek der Systemdisk	1)
lib name	die entsprechende Bibliothek auf der Systemdisk	
,unit name	1. Bibliothek in der Einheit "unit name"	1)
lib name, unit name	die angegebene Bibliothek in der Einheit "unit name"	

1) Die Suche erfolgt in der mit LTV\* angegebenen Reihenfolge.

- BEMERKUNGEN:
- Die spezifizierten Bibliotheken müssen nicht offen sein.
  - Aus oder in eine geschützte Package-Teilbibliothek (LBPROTECT) kann nicht kopiert werden.

- Ist das Originalfile mit SECURE geschützt, wird dieser Schutz mitkopiert.
- Soll ein offenes File (siehe CATALOG) kopiert werden, wird es zuerst geschlossen und erst dann dupliziert.
- Soll ein mit DEAD gekennzeichnetes File (siehe CATALOG) kopiert werden, so erfolgt eine Fehlermeldung.





DIENST-  
PROGRAMM: FLPRINT (file print)

FUNKTION: Ausdruck des Inhaltes eines Datenfiles.

FORMAT: EXE[C] FLP[RINT], filename[,lib ref]

"filename" Name des Datenfiles.  
 "lib ref" kann folgende Formen annehmen:  
 - (lib name,unit name)  
 - lib name  
 - (,unit name)  
 Als "lib name" ist der Bibliotheksname,  
 als "unit name" der Name der Disk anzugeben.

WIRKUNG: Es wird kontrolliert, ob ein ungeschütztes Datenfile mit dem Namen "filename" in der mit "lib ref" angegebenen Bibliothek existiert. Wenn ja, wird sein Inhalt über den Drucker im Standardformat ausgegeben. Der Ausdruck kann mit BREAK abgebrochen werden. Die angegebene Bibliothek muß offen sein.

BEISPIEL:

CAT DATA7		R 07-05-80 07-05-80		2048	2048	1
DATA7						
EXE FLP,DATA7						
0	TEXT1	.41839390	0	TEXT2		
.88649157	ABCDEFGHIJKLMNP		-6.0511000E+62	0		
.98707879	0	.95457924	0	.55713896		
0	.81827105	0	4.7693357E-02	0		
.68675523	0	.17611750	0	3.5848356E-02		
0	.50296996	0	.62662024	0		
.88834013	0	.85254312	0	.73719855		
0	.70058045	0	.16484721	0		
.23854311	0	.35330382	0	.43174324		
0	.95315603	0	.49297749	0		
8.9233064E-02	0	.19927847	0	.45305855		
0	.87681069	0	.96187680	0		
.41639730	0	.28043841	0	.86194278		
READY						





DIENST-  
PROGRAMM: LBCREATE (library create)

FUNKTION: Anlegen einer Bibliothek auf einer Disk.

FORMAT: EXE[C] LBC[REATE],lib ref,[password][,SIZE=Z][,\*=n1]  
[,+=n2][,NP=n3]

"lib ref"	kann folgende Formen annehmen: - (lib name,unit name) - lib name Als "lib name" ist der Bibliotheksname, als "unit name" der Name der Disk anzugeben.
"SIZE=Z"	gewünschte Kapazität (Z) in K-Bytes.
"password"	Kennwort mit maximal 6 Zeichen.
"n1, n2, n3"	positive ganze Zahlen $\leq 99$ zur Platzreservierung für die Filenamen in den Teilbibliotheken.
"*"	Package-Teilbibliothek.
"+"	Common-Teilbibliothek.
"NP"	User-Teilbibliothek.

WIRKUNG:

- Auf der mit "lib ref" angegebenen Disk oder Diskette wird die spezifizierte Bibliothek mit den gewünschten Teilbibliotheken eingerichtet. Die Zahlen n1, n2 und n3 legen fest, wieviele Directory-Sektoren den einzelnen Teilbibliotheken zur Verfügung gestellt werden. Da in jedem Sektor 27 Filenameneinträge möglich sind, wird die Zahl der in den einzelnen Teilbibliotheken speicherbaren Files implizit festgelegt.
- Die gewünschte Größe der Bibliothek wird durch den Parameter "SIZE=Z" in Kilobytes angegeben.
- Die Größe "Z" einer Bibliothek errechnet sich aus der für die Datenspeicherung benötigten Kapazität, den Sektoren für die Eintragung der Teilbibliotheken (n1+n2+n3) und 4 Sektoren (1 K-Byte) für Kontrolleintragungen des Systems.

Somit läßt sich für die Größe "Z" (in K-Bytes) folgende Formel anwenden:

$$Z = K + \text{INT}[(n1 + n2 + n3) / 4] + 2$$

worin "K" die gewünschte Kapazität in der Bibliothek für Datenspeicherung (in K-Bytes) ist und für n1, n2, n3 die im Befehlsformat verwendeten Werte bzw. deren Angaben aus der Default-Tabelle einzutragen sind.

BEMERKUNGEN: - Fehlen die Parameter für die Kapazität und/oder die Teilbibliotheken, werden folgende Default-Werte angenommen:

Typ des Datenträgers	SIZE (Kapazität der Bibliothek)	Parameter der Teilbibliotheken		
		*	+	NP
Megafloppy (256 Bytes pro Sektor)	262144 Bytes (1024 Sektoren)	3	2	3
Standardfloppy (256 KB, 128 Bytes pro Sektor)	Gesamtkapazität der Diskette (ca. 240 KB)	5	4	5

- Soll eine neue Bibliothek mit LBCREATE auf einer bereits benutzten Diskette eingerichtet werden, muß die Diskette vorher mit DINIT oder LBSCRATCH gelöscht werden.
- Neue Disketten müssen mit DINIT initialisiert werden. Um auf eine neu angelegte Bibliothek zugreifen zu können, muß sie mit LBO geöffnet werden.
- Die maximale Anzahl der Bibliotheken beträgt für die Megafloppy 14 und für die Standard-Diskette 1.

BEISPIEL:

```

LUT F1
  MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:07-05-80

  LIBRARY CREAT      BEG OF EXT      END OF EXT      SECTORS      EMPTY SECTORS
  FREE SPACE (TOTAL) = 1898 SECTORS
  MAXIMUM BLOCK OF FREE SPACE = 1898 SECTORS

  EXE LBC, (P6FSYS,F1),,NP=5,*2,+=0
  READY

LUT F1
  MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:07-05-80

  LIBRARY CREAT      BEG OF EXT      END OF EXT      SECTORS      EMPTY SECTORS
  P6FSYS 07-05-80    001001          073026          1898          1887
  FREE SPACE (TOTAL) = 0 SECTORS
  MAXIMUM BLOCK OF FREE SPACE = 0 SECTORS

```



DIENST-  
PROGRAMM: LBEMPTY (library empty)

FUNKTION: Löschen aller Files einer Bibliothek.

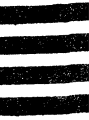
FORMAT: EXE[C] LBE[MPTY], lib ref[,password]

"lib ref" kann folgende Formen annehmen:  
- (lib name,unit name)  
- lib name  
Als "lib name" ist der Bibliotheksname,  
als "unit name" der Name der Disk anzugeben.  
"password" Password der entsprechenden Bibliothek.

WIRKUNG: Alle in der mit "lib ref" spezifizierten Bibliothek enthaltenen Files werden gelöscht. Name und Größe der Bibliothek bleiben jedoch erhalten. Der frei gewordene Speicherplatz kann wieder verwendet werden.

BEMERKUNGEN: - Ist unter "lib ref" nur "lib name" angegeben, wird die entsprechende Bibliothek auf der Systemdisk angesprochen.  
- Ist die durch "lib ref" spezifizierte Bibliothek mit einem "password" geschützt, muß dieses im Aufruf angegeben werden.





Dienst-  
PROGRAMM: LBPROTECT (library protect)

FUNKTION: Schützen von Teilbibliotheken vor der Anwendung bestimmter Systembefehle.

FORMAT: EXE[C]LBP[ROTECT],[lib ref],[password]  $\left[ \begin{matrix} * \\ + \end{matrix} \right]$

"lib ref" kann folgende Formen annehmen:  
 - (lib name,unit name)  
 - lib name  
 - (,unit name)  
 Als "lib name" ist der Bibliotheksname,  
 als "unit name" der Name der Disk anzugeben.

"password" Passwort der entsprechenden Bibliothek.

"\*" Eine Package-Teilbibliothek soll geschützt werden.

"+" Die Common-Teilbibliothek soll geschützt werden.

WIRKUNG: Wird als letzter Parameter "\*" angegeben, so werden für die Package-Teilbibliothek der mit "lib ref" angegebenen Bibliothek folgende Systembefehle und Dienstprogramme nicht ausgeführt:

CREATE  
 MODIFY  
 PURGE  
 SAVE  
 TRANSCODE  
 TRUNCATE  
 EXE FLCOPY

Für das Dienstprogramm LIBCOPY gilt:  
 Es wird nur kopiert, wenn die Package-Teilbibliothek der Empfängerbibliothek leer ist. Wird das Dienstprogramm ausgeführt, so ist auch die Kopie geschützt.

Wird als letzter Parameter "+" angegeben, so werden für die Common-Teilbibliothek der durch "lib ref" bezeichneten Bibliothek folgende Systembefehle nicht ausgeführt:

MODIFY  
 PURGE

Wird als letzter Parameter weder "\*" noch "+" angegeben, so werden sowohl die Package-Teilbibliothek als auch die Common-Teilbibliothek geschützt.

- BEMERKUNGEN:
- Wird im Parameter "lib ref" nur "lib name" angegeben, so wird die Bibliothek mit dem Namen "lib name" auf der Systemdisk geschützt.
  - Wird im Parameter "lib ref" nur "unit name" angegeben, so wird die erste Bibliothek (LVT\*) der Einheit mit dem Namen "unit name" geschützt.
  - Fehlt der Parameter "lib ref", so wird die erste Bibliothek (LVT\*) der Systemdisk geschützt.
  - Ist die durch "lib ref" bestimmte Bibliothek mit einem Password versehen, so muß es beim Aufruf des Dienstprogrammes angegeben werden.
  - Der durch dieses Dienstprogramm erreichte Schutz kann nicht mehr aufgehoben werden.





DIENST-  
PROGRAMM: LBRENAME (library rename)

FUNKTION: Änderung des Namens einer Bibliothek und/oder des Passwords.

FORMAT: EXE[C] LBR[ENAME],lib ref,[passw.<sub>1</sub>],{new lib name[,passw.<sub>2</sub>]}

"lib ref" kann folgende Formen annehmen:  
- (lib name,unit name)  
- lib name  
Als "lib name" ist der Bibliotheksname,  
als "unit name" der Name der Disk anzugeben.

"password<sub>1</sub>" altes Password.

"password<sub>2</sub>" neues Password.

"new lib name" neuer Name für die Bibliothek.

WIRKUNG: Werden alle Parameter angegeben, so wird der durch "lib ref" bestimmten Bibliothek der Name "new lib name" und das neue "password2" zugewiesen.

Wird "password1" angegeben, aber "password2" nicht, so ist die Bibliothek in Zukunft nicht mehr durch ein Password geschützt.

Fehlt der Parameter "new lib name", so wird nur das Password der Bibliothek geändert.

BEMERKUNGEN:

- Ein Password ist ein String mit 1 bis 6 beliebigen Zeichen (außer Leerzeichen).
- Wird für "lib ref" nur "lib name" angegeben, so wird die Bibliothek auf der Systemdisk gesucht.
- Eine offene Bibliothek ist auch nach Ausführung des Dienstprogrammes noch offen.
- Die Liste der bei jeder Systeminitialisierung automatisch geöffneten Bibliotheken wird nicht geändert und muß allenfalls mit LBSTORE korrigiert werden.

- Soll eine bisher nicht mit einem Password geschützte Bibliothek ein Password erhalten, so muß als "password1" das Systempassword angegeben werden. Wurde kein Systempassword vorgegeben, so kann eine Bibliothek nachträglich kein Password mehr erhalten.
- Da bei EFDOS-Systemen kein Systempassword möglich ist, ist die nachträgliche Vergabe eines Passwords an eine Bibliothek nur durch Kopieren mit LIBCOPY zu erreichen.

BEISPIEL:

```

LUT F1
  MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:01-08-80

  LIBRARY CREAT      BEG OF EXT  END OF EXT  SECTORS  EMPTY SECTORS
  P6FSYS  07-05-80  001001    073026    1898    1887
FREE SPACE (TOTAL) = 0      SECTORS
MAXIMUM BLOCK OF FREE SPACE = 0      SECTORS

EXE LBR,(P6FSYS,F1),,LIB1
READY

LUT F1
  MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:01-08-80

  LIBRARY CREAT      BEG OF EXT  END OF EXT  SECTORS  EMPTY SECTORS
  LIB1    07-05-80  001001    073026    1898    1887
FREE SPACE (TOTAL) = 0      SECTORS
MAXIMUM BLOCK OF FREE SPACE = 0      SECTORS

```



DIENST-  
PROGRAMM: LBSCRATCH (library scratch)

FUNKTION: Löschen einer Bibliothek.

FORMAT: EXE[C] LBS[CRATCH], lib ref[,password]

"lib ref" kann folgende Formen annehmen:  
- (lib name,unit name)  
- lib name  
Als "lib name" ist der Bibliotheksname,  
als "unit name" der Name der Disk anzugeben.  
"password" Password der zu löschenden Bibliothek.

WIRKUNG: Die mit "lib ref" bestimmte Bibliothek wird gelöscht.

BEMERKUNGEN:

- Ist die zu löschende Bibliothek mit einem Password versehen, so muß dieses angegeben werden.
- Ist in "lib ref" nur "lib name" angegeben, so wird die entsprechende Bibliothek auf der Systemdisk gesucht.
- Wird eine Bibliothek auf einer Disk gelöscht, so ist der freigewordene Speicherplatz erst nach Ausführung von RESTRUCT wieder verwendbar.

Bei der Ausgabe der Bibliotheksübersicht mittels LVT erfolgt deshalb die Ausgabe des freien Speicherplatzes getrennt nach gelöschten und nicht gelöschten Bibliotheken.

- Ist eine mit diesem Dienstprogramm gelöschte Bibliothek in der Liste der bei jeder Systeminitialisierung automatisch geöffneten Bibliotheken, so wird diese Liste nicht geändert. Sie ist gegebenenfalls mit LBSTORE neu festzulegen.

BEISPIEL:

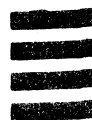
```
LUT F1
MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:01-00-80

LIBRARY CREAT      BEG OF EXT  END OF EXT  SECTORS  EMPTY SECTORS
LIB1      07-05-80  001001    073026    1898     1887
FREE SPACE (TOTAL) = 0      SECTORS
MAXIMUM BLOCK OF FREE SPACE = 0      SECTORS

EXE LBS, (LIB1,F1)
READY

LUT F1
MDOSC -R 2.0 * VOLLABEL=DISK1 * TRACK FORMAT=256 BYTE * DATE:01-00-80

LIBRARY CREAT      BEG OF EXT  END OF EXT  SECTORS  EMPTY SECTORS
FREE SPACE (TOTAL) = 1898 SECTORS
MAXIMUM BLOCK OF FREE SPACE = 1898 SECTORS
```



DIENST-  
PROGRAMM: LIBCOPY (library copy)

FUNKTION: Kopieren einer (Teil-)Bibliothek in eine andere Bibliothek.

FORMAT:

$$\text{EXE}[C] \text{ LIB}[\text{COPY}], \text{ IN}=[\text{lib ref}], [\text{password}_1], \left[ \begin{matrix} * \\ + \\ : \end{matrix} \right],$$

$$\text{OUT}=[\text{lib ref}], [\text{password}_2], [\text{filename}]$$

"lib ref" kann folgende Formen annehmen:

- (lib name, unit name)
- lib name
- (, unit name)

Als "lib name" ist der Name der Bibliothek, als "unit name" der Name Diskstation anzugeben.

"IN=lib ref" bestimmt die Bibliothek, aus der eine oder mehrere Teilbibliotheken kopiert werden.

"password<sub>1</sub>" Passwort der mit "IN=lib ref" bestimmten Bibliothek.

"\*" Eine Package-Teilbibliothek soll kopiert werden.

"+" Eine Common-Teilbibliothek soll kopiert werden.

":" Alle Teilbibliotheken sollen kopiert werden.

"OUT=lib ref" bestimmt die Empfängerbibliothek.

"password<sub>2</sub>" Passwort der Empfängerbibliothek.

"filename" bestimmt, ab welchem File mit Kopieren begonnen werden soll (siehe CATALOG).

WIRKUNG: Die spezifizierten Teilbibliotheken der mit "IN=lib ref" bestimmten Bibliothek werden in die mit "OUT=lib ref" definierte Bibliothek kopiert. Die Files der Teilbibliotheken werden in der gleichen Reihenfolge kopiert, wie sie in der durch CATALOG erzeugten Liste aufgeführt sind. Ist der Parameter "filename" angeführt, so wird der Kopiervorgang erst mit diesem File gestartet.

- BEMERKUNGEN:
- Die angesprochenen Bibliotheken müssen nicht offen sein.
  - Wird in einem Operand für "lib ref" nur "lib name" gesetzt, bezieht sich das Dienstprogramm auf die entsprechende Bibliothek der Systemdisk.
  - Wird in einem Operand für "lib ref" nur "unit name" spezifiziert, so bezieht sich das Dienstprogramm auf die erste Bibliothek (siehe LVT) der angegebenen Einheit.
  - Fehlt "lib ref" im Parameter "IN=", so wird aus der ersten Bibliothek der Systemdisk kopiert.  
Fehlt "lib ref" im Parameter "OUT=", so erfolgt die Kopie in die erste Bibliothek der Systemdisk.
  - Sind die angesprochenen Bibliotheken mit einem Passwort versehen, so ist dieses anzugeben.
  - Fehlt der dritte Operand (\*/+/:), wird die User-Teilbibliothek kopiert.
  - Durch diesen Kopiervorgang werden alle Files in einem Abschnitt abgespeichert (siehe CATALOG, Kolonne EXT). Eine Ausnahme bildet das Kopieren auf User-Disketten, wo allenfalls ein File in maximal zwei Teilen gespeichert sein kann.
  - Eine mit LBPROTECT geschützte Package-Teilbibliothek kann nur kopiert werden, wenn die Empfängerbibliothek noch keine Files enthält. Ein nur teilweises Kopieren unter Verwendung des Parameters "filename" ist nicht möglich.

MELDUNGEN: RESTORE DISKS  
Der Kopiervorgang ist abgeschlossen. Disketten können gewechselt werden. Die Taste CONTINUE führt in den Command-Mode zurück.

LOAD DISK ON unit name1 [unit name2]  
Die angegebene Bibliothek (ev. beide) ist in der Einheit "unit name" nicht vorhanden. Der Kopiervorgang ist aufgeschoben. Die entsprechenden Disketten können gewechselt werden. Der Kopiervorgang wird mit CONTINUE aktiviert oder mit BREAK endgültig abgebrochen (z.B. falscher Bibliotheksname).

ERROR n - RESTORE DISKS  
Der Kopiervorgang ist aufgrund der Fehlermeldung n abgebrochen worden. Je nach Fehlercode (siehe Anhang) sind Disketten auszuwechseln. Mit CONTINUE wird der Command-Mode wieder erreicht.

ERROR n ON SUBLIB  $\left\{ \begin{array}{c} * \\ + \\ NP \end{array} \right\}$

Der Kopiervorgang in die angegebene Teilbibliothek (NP steht für User-Teilbibliothek) mußte abgebrochen werden, da in der Directory nicht genügend Platz für die Eintragung aller Filenamen vorhanden war. Werden im Aufruf des Dienstprogrammes noch Kopien weiterer Teilbibliotheken gefordert, so wird die Arbeit mit der nächsten Teilbibliothek fortgesetzt.

filename NOT COPIED

Der Kopiervorgang wurde für die das File "filename" enthaltende Teilbibliothek abgebrochen, da in der Empfängerbibliothek kein Platz mehr vorhanden war. Das File "filename" und alle Files, die gemäß CATALOG in der Fileliste der entsprechenden Teilbibliothek nachfolgend angeführt sind, wurden nicht kopiert.

NO LIBRARY lib name ON UNIT unit name

Die Liste der Bibliotheken, die bei jeder Systeminitialisierung automatisch geöffnet werden, stimmt mit den vorhandenen Bibliotheken nicht mehr überein, da das Dienstprogramm für den Kopiervorgang einen Diskettenwechsel verlangt hatte.

BEISPIEL:

Inhalt der Bibliothek LIB1:

CAT :,(F1),,F

EFDOSC-R 3.1 \* VOLLABEL = DISK1 \* LIBRARY = LIB1 \* DATE:01-08-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
------	------	-------	----------	----------	-----------	------	-----

FILE1	S	01-08-80	01-08-80	50048	0000		1
FILE2	R	01-08-80	01-08-80	150016	150016		1
FILE77	S	01-08-80	01-08-80	35072	0000		2

Inhalt der Bibliothek P6FSYS:

CAT :,(F2),,F

EFDOSC-R 3.1 \* VOLLABEL = \* LIBRARY = P6FSYS \* DATE:01-08-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
------	------	-------	----------	----------	-----------	------	-----

TEST	S	07-05-80	07-05-80	1024	0000		1
TEST2	P	07-05-80	07-05-80	0384	0384		1
TEST3	P	07-05-80	07-05-80	0384	0384		1
DATA7	R	07-05-80	01-08-80	5120	5120		1

EXE LIB,IN=(F2),,OUT=(F1)  
DATA7 NOT COPIED  
ERROR 188

Kopieren der Bibliothek P6FSYS (alle Teilbibliotheken) in die Bibliothek LIB1. Das File DATA7 hat keinen Platz in LIB1!

Neuer Inhalt der Bibliothek LIB1:

CAT :,(F1),,F

EFDOSC-R 3.1 \* VOLLABEL = DISK1 \* LIBRARY = LIB1 \* DATE:01-08-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
------	------	-------	----------	----------	-----------	------	-----

FILE1	S	01-08-80	01-08-80	50048	0000		1
FILE2	R	01-08-80	01-08-80	150016	150016		1
FILE77	S	01-08-80	01-08-80	35072	0000		2
TEST	S	07-05-80	07-05-80	1024	0000		1
TEST2	P	07-05-80	07-05-80	0384	0384		1
TEST3	P	07-05-80	07-05-80	0384	0384		1

SFA (F1)  
SPACE = 2944

Kontrolle: Freier Platz in LIB1 kleiner als File DATA7.





DIENST-  
PROGRAMM: RESTRUCT

FUNKTION: Neuorganisation eines Diskinhaltes nach dem Löschen von Bibliotheken.

FORMAT: EXE[C] RES[TRUCT][,unit name]

"unit name" Name der Disk.

WIRKUNG: Der Speicherplatz von mit LBSCRATCH gelöschten Bibliotheken einer Disk wird wieder freigegeben. Dabei entsteht eine neue, lückenlose Bibliotheksorganisation.

BEMERKUNGENÄ

- Fehlt der Parameter "unit name", so ist die Systemdisk angesprochen.
- Offene Bibliotheken bleiben offen, jedoch wird die Liste der bei einer Initialisierung des Systems automatisch geöffneten Bibliotheken gelöscht. Sie muß mit LBSTORE neu definiert werden.

MELDUNGEN: LIBRARY lib name OK  
Der Name jeder bei der Neuorganisation korrekt übernommenen Bibliothek wird in obiger Form aufgelistet.

ERROR n LIBRARY lib name DELETED  
Die Ausführung des Dienstprogrammes wurde aufgrund des Fehlercodes n (siehe Anhang) gestoppt. Die mit "lib name" angegebene Bibliothek ist zerstört. RESTRUCT kann mit CONTINUE fortgesetzt oder mit BREAK endgültig abgebrochen werden.





DIENST-  
PROGRAMM: VOLLABEL (volume label)

FUNKTION: Kennzeichnung einer Disk.

FORMAT: EXE[C] VOL[LABEL], unit name, volume label

"unit name" Name der Einheit, deren Disk oder Disketten einen "volume label" erhalten soll.

"volume label" Name aus 1 bis 6 Großbuchstaben oder Ziffern. Das erste Zeichen muß ein Buchstabe sein.

WIRKUNG: Die in der mit "unit name" angesprochenen Einheit enthaltene Disk erhält den spezifizierten Volume-Label.

BEISPIEL:

CAT

MDOSC -R 2.0 \* VOLLABEL = \* LIBRARY = P6FSYS \* DATE:01-08-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
------	------	-------	----------	----------	-----------	------	-----

EXE VOL,F2,DISK2  
READY

CAT

MDOSC -R 2.0 \* VOLLABEL = DISK2 \* LIBRARY = P6FSYS \* DATE:01-08-80

FILE	TYPE	CREAT	LAST MOD	MAX SIZE	USED SIZE	CODE	EXT
------	------	-------	----------	----------	-----------	------	-----



8.	<u>CALCULATOR- UND DEBUGGING-MODE</u>	8.1
8.1	Calculator-Mode	8.1
8.2	Einsetzen des Calculator-Modes	8.1
8.3	Festlegen der Art des Argumentes trigonometrischer Funktionen	8.2
8.4	Befehl zur Belegung der Funktionstasten	8.3
8.5	Lokale Variable und numerische Ausdrücke	8.4
8.5.1	Zahlendarstellung	8.5
8.5.2	Standardfunktionen	8.5
8.6	Verarbeitung der Anweisungen	8.5
8.7	Ausgabe und Ausgabeformat der Ergebnisse	8.6
8.8	Debugging-Mode	8.7
8.8.1	Erreichen und Verlassen des Debugging-Modes	8.7
8.8.2	Operationen im Debugging-Mode	8.8
	Behebbarer Fehler	8.8
	Abfrage von Variablenwerten	8.10
	Wertzuweisung an Variable	8.11
8.8.3	Rechnen im Debugging-Mode	8.12
8.8.4	START-, STOP-Befehle im Debugging-Mode	8.13



## 8. CALCULATOR- UND DEBUGGING-MODE

### 8.1 Calculator-Mode

In der Betriebsart Calculator-Mode kann das System als Tischrechner verwendet werden, ohne daß die sich im Arbeitsspeicher befindlichen Programme und Textfiles dadurch beeinflußt werden. Im einzelnen lassen sich numerische Ausdrücke berechnen, die aus numerischen Konstanten, Standardfunktionen und/oder gespeicherten Werten zuvor durchgeführter Rechenoperationen bestehen.

Als Standardfunktionen stehen alle auch in einem BASIC-Programm verwendbaren Funktionen zur Verfügung. Für die trigonometrischen Funktionen kann zusätzlich festgelegt werden, ob mit Bogenmaß, Altgrad oder Neugrad gerechnet werden soll.

Das Ergebnis einer Operation läßt sich in einer von 4 speziellen Variablen abspeichern. Es wird am Bildschirm angezeigt, kann aber auch gedruckt werden. Dazu ist zuvor die Taste PRINT ALL zu betätigen. Die Funktionstasten lassen sich im Calculator-Mode wie durch ein BASIC-Programm belegen. Die Standardbelegung ist danach durch den Befehl LDKEYS wieder einsetzbar.

### 8.2 Einsetzen des Calculator-Modes

Der Calculator-Mode wird durch Eingabe des Befehls CAL eingesetzt und kann durch erneute Eingabe dieses oder eines beliebigen anderen Systembefehles wieder verlassen werden.

Zusätzlich kann beim Aufruf ein anderes Format als das Standardformat gewählt werden. Siehe dazu Abschnitt 8.7.

Im einzelnen sind folgende Eingaben zulässig:

- Eingabe des Befehles CAL: der Calculator-Mode wird verlassen.
- Eingabe eines System-Befehles: der Calculator-Mode wird verlassen und der Befehl ausgeführt.
- Eingabe einer Calculator-Mode-Zeile:
  - . Befehl, daß bei der Berechnung trigonometrischer Funktionen das Argument als Bogenmaß, Alt- oder Neugrad interpretiert werden soll.
  - . Befehl zur Belegung der Funktionstasten.
  - . Anweisung zur Berechnung eines numerischen Ausdruckes.

Nicht akzeptiert werden:

- BASIC-Statements;
- Zeilennummern vor einem der oben angeführten Befehle;
- Variablennamen, die von  $\Phi$ ,  $\Phi 0$ ,  $\Phi 1$  oder  $\Phi 2$  verschieden sind.

Alle Berechnungen im Calculator-Mode werden durch Drücken von EOL abgeschlossen.

### 8.3

#### Festlegen der Art des Argumentes trigonometrischer Funktionen

Die Festlegung der Arbeitsweise im Bogenmaß, bzw. in Altgrad oder Neugrad erfolgt durch einen der folgenden Befehle:

- SRAD                      Berechnung im Bogenmaß
- SDEG                     Berechnung in Altgrad
- SGRAD                    Berechnung in Neugrad

Nach dem Einsetzen des Calculator-Modes wird "SRAD" angenommen, das heißt, die Argumente trigonometrischer Funktionen werden als Bogenmaß interpretiert.

Nach dem Befehl "SDEG" wird in Altgrad gerechnet. Dabei sind die Winkel in dezimaler Form (z.B. 25.7 Grad), nicht in der Form "Grad, Minute, Sekunde", einzugeben.

Nach dem Befehl "SGRAD" wird in Neugrad gerechnet. Auch hier müssen die Winkel in dezimaler Form eingegeben werden.

Die Verwendung von Alt- oder Neugrad im Calculator-Mode hat keinen Einfluß auf die Interpretation der Argumente trigonometrischer Funktionen in einem BASIC-Programm, dort wird nur im Bogenmaß gerechnet.

Für den Zusammenhang von Altgrad, Neugrad und Bogenmaß gilt:

$$100 \text{ Neugrad} \hat{=} 90 \text{ Altgrad} \hat{=} \pi/2$$

$$\pi = 3,1415\ldots$$

#### Beispiel:

```
CAL
COS (PI)
-1.0000000000000000

SDEG
COS (180)
-1.0000000000000000

SGRAD
COS (200)
-1.0000000000000000

SRAD
COS (PI)
-1.0000000000000000
```



Befehl zur Belegung der Funktionstasten

FORMAT: FKEY n, String-Konstante

"n" positive ganze Zahl,  $1 \leq n \leq 16$ .

"String-Konstante" beliebige Folge von ISO-Zeichen.

Wie in der BASIC-Anweisung "FKEY" kann ein EOL durch einen Doppelpunkt am Ende des Strings codiert werden, so daß das Drücken der EOL-Taste entfällt, wenn die Funktionstaste gedrückt ist.

Bemerkung:

Dem FKEY-Befehl darf keine Zeilennummer vorangehen. Die ursprüngliche Standardbelegung kann durch Eingabe des Befehls LDKEYS erreicht werden. Dabei wird der Calculator-Mode verlassen und der Command-Mode eingesetzt.

Beispiel:

Standardbelegung der Funktionstasten:

F1: STANDARD	
F2: BELEGUNG	
CALC-MODE	Calculator-Mode einschalten.
FKEY 1, CALCULATOR-MODE-	Funktionstaste 1 belegen.
F1 F2	Tasten F1 und F2 drücken.
CALCULATOR-MODE-BELEGUNG	Die modifizierte Belegung wird angezeigt.
FKEY 1, 3*5:	Nach Drücken der Funktionstaste
F1	wird der String 3*5 als numerischer Ausdruck interpretiert, berechnet und
15.00000000000000	das Ergebnis angezeigt.
LDKEYS	
F1 F2	
STANDARDBELEGUNG	Die Standardbelegung wird wieder hergestellt und angezeigt. Der Calculator-Mode wird verlassen.

Bemerkung:

Die Summe der Länge der Strings, mit welchen die Funktionstasten belegt sind, darf 238 nicht überschreiten.

Eine Anweisung zur Berechnung numerischer Ausdrücke hat folgende Form:

```
num.Ausdruck
 $\Phi[n] = \text{num.Ausdruck}$ 
```

" $\Phi$ " Dieses Symbol ist der Taste RESULT zugeordnet. Die Taste Result ist die zweite Taste links von der Taste RES bzw. AUTO.

"num.Ausdruck" numerischer Ausdruck.

"n" eine der Ziffern 0, 1 oder 2.

$\Phi$ ,  $\Phi 0$ ,  $\Phi 1$ ,  $\Phi 2$  sind die Namen lokaler Variablen. Diesen Variablen können die Resultate der im Calculator-Mode durchgeführten Berechnungen als Wert zugewiesen werden.

Eine Anweisung im Calculator-Mode kann durch die Taste EOL oder SUM abgeschlossen werden. Dadurch ergeben sich vier mögliche Formate:

- num.expr. EOL  
Der Wert des Ausdruckes "num.expr." wird berechnet und der lokalen Variablen  $\Phi$  zugewiesen.
- RESULT n = num.expr. EOL  
Der Wert des numerischen Ausdruckes wird berechnet und der Variablen  $\Phi n$  zugewiesen.

Die Variablen werden durch eine der folgenden Anweisungen neu initialisiert, das heißt, es wird ihnen der Wert 0 zugewiesen:

- 0 EOL es wird  $\Phi = 0$  gesetzt.
- RESULT n = 0 EOL es wird die Variable  $\Phi n = 0$  gesetzt.

Numerische Ausdrücke im Calculator-Mode können enthalten:

- numerische Konstante
- lokale Variable
- numerische Standardfunktionen (SIN, .....
- Operationszeichen (+, \*, -, /, ^)

Reihenfolge und Priorität der Berechnung sind die gleichen wie in einem BASIC-Programm, sie können durch Setzen von Klammern beeinflußt werden.

### 8.5.1

#### Zahlendarstellung

Numerische Werte werden, wie in einem BASIC-Programm, auch im Calculator-Mode intern im Gleitkomma-Format (halblogarithmische Darstellung) dargestellt. Die Berechnung und Darstellung erfolgt in doppelter Genauigkeit. Jeder Zahl  $Z$  wird eine Mantisse  $m$  und ein Exponent  $n$  zugeordnet:

$$Z = m \cdot 10^n$$

"m" 13-stellige Mantisse der Form

X.YYYYYYYYYYYYY

$1 \leq X \leq 9, 0 \leq Y \leq 9$

"n" ganze Zahl,  $-99 < n < 99$

Eine Zahl kann als ganze Zahl oder als Dezimalzahl eingegeben werden; sie darf bis zu 13 Ziffern (ohne Dezimalpunkt und Vorzeichen) enthalten:

248 -5 1234.567890123

Eine weitere Form der Eingabe ist das Gleitkomma-Format:

nEa

"n" Dezimalzahl mit maximal 13 Ziffern

"a" ganze Zahl zwischen -99 und +99

5,698E2 -678.3456E4 2E-27

Da alle eingegebenen Zahlen durch das System in die halblogarithmische Darstellung übersetzt werden, ist die gewählte Eingabeform für die Berechnung unerheblich.

### 8.5.2

#### Standardfunktionen

Die im Calculator-Mode verwendbaren numerischen Standardfunktionen sind die gleichen wie in der Programmierung mit BASIC. Die Liste der Funktionen ist in Kapitel 9 enthalten.

### 8.6

#### Verarbeitung der Anweisungen

Erfolgt der Abschluß der Eingabe einer Anweisung durch EOL, wird die Anweisung syntaktisch überprüft. Ein eventueller Syntax-Fehler wird angezeigt und die Tastatur mit Ausnahme der Tasten RECALL und CLEAR gesperrt.

Durch Drücken der Taste RECALL wird die fehlerhafte Anweisung im Display sichtbar. Die Stellung des Pointers gibt eine der möglichen Fehlerpositionen an. Die Fehler können mit den Editing-Routinen korrigiert und die korrigierte Anweisung danach durch EOL wieder eingegeben werden. Durch Drücken von CLEAR wird die fehlerhafte Anweisung gelöscht.

Im Display sind sichtbar:

- alle Eingaben
- alle Fehlermeldungen
- die Ergebnisse

Wird RESULT gedrückt, erscheint im Display das Zeichen " $\Phi$ " als Symbol der lokalen Variablen  $\Phi$ .

Die Werte der lokalen Variablen können durch folgende Anweisungen abgefragt werden:

- |                           |  |
|---------------------------|--|
| - RESULT EOL:             | der aktuelle Wert der Variablen $\Phi$ wird ausgegeben, bleibt aber unverändert.   |
| - RESULT n EOL            | der aktuelle Wert der Variablen $\Phi$ n wird ausgegeben und der Wert von $\Phi$ durch diesen überschrieben; der Wert von $\Phi$ n bleibt unverändert. |
| - RESULT n = RESULT n EOL | der Wert von $\Phi$ n wird ausgegeben; der Wert aller Variablen (inkl. $\Phi$ ) bleibt unverändert.  |

Üblicherweise erfolgt die Ausgabe von Ergebnissen im Standardformat. Es kann jedoch beim Aufruf des Calculator-Modes in ein anderes Ausgabeformat angegeben werden.

Erfolgt der Aufruf mit

CAL n

mit  $0 \leq n \leq 13$ , so erfolgt die Ausgabe der Ergebnisse grundsätzlich mit n Nachkommastellen.

Erfolgt der Aufruf mit

CAL FL

so werden alle Ergebnisse in Gleitkommadarstellung mit allen intern dargestellten Ziffern angezeigt.

Der Debugging-Mode dient zum Testen und Prüfen von Programmen und Programmteilen. Treten während der Ausführung eines Programmes nicht behebbare Fehler auf, so ermöglicht der Übergang in den Debugging-Mode die Abfrage von Variablenwerten sowie Berechnungen zur Ermittlung der Fehlerursache. Der Debugging-Mode kann in diesem Falle nur durch Drücken der Taste BREAK und somit durch Übergang in den Command-Mode verlassen werden. In allen anderen Fällen sind folgende Operationen verfügbar:

- Korrektur behebbarer Fehler
- Abfrage von Variablenwerten
- Zuweisung neuer Werte an Variable
- Zeilenweise Verarbeitung eines Programmes
- Neubelegung der Funktionstasten
- Fortsetzung des Programmlaufes in einer beliebigen Zeile
- Anweisung zum Stoppen des Programmlaufes in einer beliebigen Zeile

Außerdem besteht die Möglichkeit, mit vom Benutzer definierten, ein- oder mehrzeiligen Funktionen in gleicher Weise wie mit Standardfunktionen zu arbeiten.

Befindet sich das System im Command- oder Calculator-Mode, so wird der Debugging-Mode durch Ausführung des Befehles PREPARE erreicht.

Befindet sich das System im Running-Mode, wird der Debugging-Mode erreicht durch:

- Ausführung eines (programmierten) STOP-Statements.
- Ausführen der Funktion STEP (Taste EXIT).
- Auffinden eines behebbaren Fehlers durch das System während eines Programmlaufes.
- Ausführung eines zuvor eingegebenen "STOP Zeilennummer"-Befehles.
- Auftreten eines nicht behebbaren Fehlers.

Erwartet das System eine Tastatureingabe, wird der Debugging-Mode durch Ausführen der Funktion STEP (Taste EXIT) erreicht.

Der Debugging-Mode wird verlassen durch:

- Ausführen der Funktion BREAK (Ctrl. EXIT). In diesem Falle wird der Command-Mode erreicht.
- Ausführen der Funktion CONTINUE (EXIT). In diesem Falle wird die Ausführung des Programmes fortgesetzt.
- Eingabe des Befehles "START Zeilennummer". In diesem Falle wird die Ausführung des Programmes bei der angegebenen Zeilennummer fortgesetzt. Es ist vom Anwender sicherzustellen, daß eine Ausführung des Programmes ab der angegebenen Zeilennummer sinnvoll ist.
- Ausführen der Funktion STEP. Es wird nur ein Statement ausgeführt und dann wieder in den Debugging-Mode zurückgekehrt. Durch dieses Vorgehen wird die Verarbeitung des Programmes in Einzelschritten ermöglicht.
- Ausführen der Funktion BREAK nach Auftreten eines nicht behebbaren Fehlers.

#### 8.8.2

#### Operationen im Debugging-Mode

##### Behebbarer Fehler

Behebbarer Fehler sind solche, für die im Betriebssystem eine Standardlösung vorgesehen ist.

Beispiele solcher Fehler sind unter anderem:

- Eine Variable soll verarbeitet werden, ohne daß ihr zuvor ein Wert zugewiesen wurde.
- Die Quadratwurzel aus einer negativen Zahl soll gezogen werden.
- Eine singuläre Matrix soll invertiert werden (d.h. Determinante = Null).
- Die angesprochene periphere Einheit ist nicht eingeschaltet.
- Der für die Erzeugung eines Bildes beim Plotten vereinbarte Buffer kann keine weiteren Punkte aufnehmen.

Der Benutzer hat die Möglichkeit, die Fehler durch explizite Wertzuweisungen oder Änderungen zu beheben oder nur die Taste EXIT zu drücken. Im letzteren Falle werden die Werte vom System zugewiesen.

Ist die Fehlerbehebung im Programm durch die Kontrolle von internen Interrupts vorgesehen, so erfolgt keine Fehlermeldung und der Debugging-Mode wird nicht erreicht.

Die behebbaren Fehler sind in der Liste der Fehlercodes gesondert aufgeführt. Zusätzlich sind dort Lösungen angegeben, die vom System als Standardwerte angenommen werden.

### Beispiel:

```
NEW
10 X = Y+2
20 PRINT
30 PRINT X
40 END

RUN
ERROR 1 IN LINE 10
```

Eingabe eines neuen Programmes.

Im Display erscheint die Meldung, daß ein behebbarer Fehler aufgetreten ist. Die Variable Y ist nicht definiert.

### Möglichkeiten:

```
1. CONTINUE
   0
```

Der Variablen Y wird der Wert 0 zugewiesen und  $0+2 = 0$  ausgedruckt.

```
2. Wertzuweisung durch den
   Benutzer
   Y = 2
   START 10
   4
```

Direkte Wertzuweisung an Y. Fortsetzung des Programmlaufes. Der Wert  $2+2 = 4$  wird ausgegeben.

### Bemerkung:

Wird ein behebbarer Fehler gemeldet, so ist die den Fehler auslösende Anweisung bereits ausgeführt. Wird eine von der Standardlösung verschiedene Lösung gewünscht, so sind entweder alle von dieser Anweisung betroffenen Ergebnisse zu korrigieren oder die Programmausführung muß mit START sinnvoll fortgesetzt werden.

### Abfrage von Variablenwerten

Werte von Variablen können durch Eingabe des Namens der Variablen und anschließendem Drücken der Taste EOL abgefragt werden.

#### Beispiel:

NEW	Eingabe eines neuen Programmes.
10 A = 3	
20 B = A+2	
30 C = B+2	
40 STOP	
50 END	
RUN	Start des Programmlaufes.
STOP IN LINE 40	Durch das STOP-Statement wird das System in den Debugging-Mode versetzt.
A EOL	Der Wert von A wird abgefragt.
3	Der Wert wird ausgegeben.
C EOL	Der Wert der Variablen C wird abgefragt
81	und ausgegeben.

Wie im Calculator-Mode erfolgt die Ausgabe der Werte über das Display. Ein Ausdruck erfolgt nur, wenn die Taste PRINT ALL aktiviert ist.

Die Ausgabe erfolgt im Standardformat.

Die Werte alphanumerischer Variablen können ebenfalls abgefragt werden.

#### Beispiel:

NEW	Eingabe eines neuen Programmes.
10 A\$ = "TEST"	
20 B\$ = "AUSGABE"	
30 C\$ = A\$+B\$	
40 STOP	
50 END	
RUN	Start des Programmlaufes.
STOP IN LINE 40	STOP-Anweisung in Zeile 40. Das System geht in den Debugging-Mode.
A\$ EOL	Die Variablenwerte werden abgefragt
TEST	und ausgegeben.
C\$ EOL	
TESTAUSGABE	



Zu beachten ist jedoch: Ein Programm wird zeilenweise verarbeitet. Daher können nur Variable abgefragt werden, die zu diesem Zeitpunkt bereits einen definierten Wert aufweisen. Andernfalls erfolgt eine Fehlermeldung. Ein Beispiel soll diesen Sachverhalt erläutern:

NEW	Eingabe eines neuen Programmes.
10 STOP	
20 A = 5	
30 END	
RUN	
STOP IN LINE 10	Ausführung des STOP-Statements. Übergang in den Debugging-Mode.
A EOL	Abfrage der Variablen A:
0	Auf dem Display erscheint kurz die Zahl 0 und anschließend die Fehlermeldung ERROR 1, da die Variable A zur Zeit der Ausführung des STOP-Statements noch keinen Wert zugewiesen bekommen hat. Bevor die Wertzuweisung erfolgen oder die Taste CONTINUE gedrückt werden kann, ist die Fehlermeldung ERROR 1 durch Drücken der Tasten SHIFT und CLEAR zu löschen.
ERROR 1	

Nach Abfrage der Variablenwerte wird der Programmablauf durch Drücken der Taste CONTINUE fortgesetzt usw.

#### Wertzuweisung an Variable

Die Wertzuweisung an Variable geschieht durch folgende Anweisung:

```
numerische Variable = num.Ausdruck EOL
String-Variable      = Stringausdruck EOL
```

Wertzuweisungen können an numerische und alphanumerische Variable erfolgen. Diese Variablen können globale, im Programm verwendete Variable oder die lokalen Variablen  $\Phi$ ,  $\Phi 0 - \Phi 2$  des Calculator-Modes sein. Im Debugging-Mode an globale Variable zugewiesene Werte bleiben nach Wiederaufnahme des Programmlaufes durch CONTINUE usw. erhalten. Jede im Programm bereits definierte Variable sowie alle Funktionen können verwendet werden.

### Beispiel:

```
NEW                               Eingabe eines neuen Programmes.
10 A = 3
20 B$ = "TEST"
30 PRINT A; B$
40 STOP
50 PRINT A; B$
60 END

RUN
3 TEST
STOP IN LINE 40
A = 100 EOL
B$ = "PROGRAMM" EOL
CONTINUE
100 PROGRAMM
```

#### 8.8.3

### Rechnen im Debugging-Mode

Der Debugging-Mode umfaßt alle auch im Calculator-Mode zur Verfügung stehenden Möglichkeiten. Auch das Eingabeformat einer Anweisung ist das gleiche (siehe Kapitel 12).

Insbesondere gilt:

- Es kann in Neugrad, Altgrad oder im Bogenmaß gerechnet werden.
- Die Funktionstasten können belegt werden.
- Die Variablen  $\Phi$ ,  $\Phi 0$ ,  $\Phi 1$  und  $\Phi 2$  stehen zur Verfügung.

Die entsprechenden Befehle sind die gleichen wie im Calculator-Mode, es ist aber zu beachten, daß bei Fortsetzung des Programmlaufes nur noch im Bogenmaß gerechnet wird! Zusätzlich zu den Möglichkeiten für das manuelle Rechnen sind folgende Operationen ausführbar:

- Verwendung aller im BASIC-Programm angeführten Variablen und nicht nur der Variablen  $\Phi$ ,  $\Phi 0$ ,  $\Phi 1$  und  $\Phi 2$ . Beide Arten können gemischt verwendet werden.
- Verwendung von im Programm definierten Variablenwerten.
- Verwendung von im Programm durch DEFFN oder DEFFN/FNEND definierten Funktionen.

#### Beispiel 1: Bestimmung des Maximums zweier Zahlen

NEW	Eingabe eines neuen Programmes.
10 DEFFNA (X,Y)	
20 IF X > Y THEN 50	
30 FN* = Y	
40 GOTO 60	
50 FN* = X	
60 FNEND	
70 END	
PRE	Die Preexecution wird durchgeführt. Das System befindet sich im Debugging-Mode.
FNA(50,20) EOL	Die Funktion FNA wird als Standardfunktion verwendet.
50	Der lokalen Variablen $\Phi$ wird der Wert 100,
100 EOL	der lokalen Variablen $\Phi 0$ der Wert 200 zugewiesen.
$\Phi 0 = 200$ EOL	Der Wert der Funktion FNA wird ermittelt.
FNA( $\Phi$ , $\Phi 0$ ) EOL	
200	

#### 8.8.4

##### START-, STOP-Befehle im Debugging-Mode

Der Befehl "STOP ln EOL", worin "ln" die Zeilennummer ist, bewirkt, daß der Programmlauf unmittelbar vor Ausführung des Statements mit der Zeilennummer "ln" unterbrochen wird. Durch diesen Befehl wird der Debugging-Mode aber nicht verlassen. Dazu muß die Taste "SHIFT EXIT" oder "EXIT" wieder gedrückt oder ein Befehl "START ln" eingegeben werden.

##### Bemerkung:

Der Befehl "STOP ln" bleibt gültig, bis entweder ein anderer STOP-Befehl eingegeben oder das Programmende erreicht wird.

Der Debugging-Befehl "STOP ln" hat die gleiche Wirkung wie das BASIC-Statement "ln STOP".

Durch die Kombination von START- und STOP-Befehlen ist es möglich, einzelne Programmteile isoliert auszuführen:

```
STOP ln 1 EOL
START ln 2 EOL
```

Das Programmstück zwischen den Zeilennummern ln 2 (inklusive) und ln 1 (exklusive) wird ausgeführt.





