

M20

PCOS

*(Professional-Computer
Operations System)*

Betriebssystem-Benutzer-Handbuch



olivetti

Das Handbuch dient der Information, sein Inhalt ist ohne ausdrückliche schriftliche Vereinbarung nicht Vertragsgegenstand. Technische Änderungen behalten wir uns vor. Die angegebenen Daten sind lediglich Nominalwerte.

© Copyright 1984 by Deutsche Olivetti DTS GmbH.

VORWORT

Das vorliegende Handbuch beschreibt das eigens für den M20 entwickelte Betriebssystem **PCOS** (Professional Computer Operating System). Zugrunde liegt der Release-Stand PCOS 2.0.

Das Release 2.0 ist - bezogen auf die Systemebene - nicht kompatibel mit älteren Ständen (z.B. 1.3), jedoch aufwärts kompatibel mit dem Release 4.0.

BASIC 8000-Programme, die unter Rel. 1.3 erstellt wurden, sind auch kompatibel mit unter Release 2.0 erstellten. Ausnahmen: Über **GET%** gespeicherte Bildschirm-Arrays, die mit **PUT%** wieder abgebildet werden. Zur Konvertierung von solchen unter Rel. 1.3 erstellten Arrays in den unter Rel. 2.0 benötigten Aufbau existiert ein Konvertierungsprogramm (getconv.bas).

Mit ISAM, Release 1.3, erstellte Filestrukturen sind kompatibel mit solchen unter Rel. 2.0. Unter Rel. 2.0 muß aber das Assemblerfile isam.sav im Rel. 2.0 zur Verarbeitung verwendet werden.

Durch PCOS 2.0 werden folgende Programmiersprachen gestützt:

- * BASIC 8000
- * Z 8001-Assembler

Rel. 2.0 stützt eine integrierte HDU, die nur für PCOS genutzt werden kann.

Für das Release 4.0 sind einige wenige Erweiterungen bzw. Änderungen vorgenommen worden, die in einem Ergänzungsband beschrieben sind. Betroffen sind davon:

- die PCOS-Befehle: **f1, la, pk, sf, sp** und **ss**
- die Fehlercodes: Bedeutungen und zugeordnete Meldungen; dies hat somit evtl. auch Auswirkungen auf BASIC 8000-Fehlerbehandlungsroutinen unter Rel. 2.0 bzw. älter
- Außerdem existieren einige zusätzliche PCOS-Befehle.

Durch PCOS 4.0 wird unter anderem zusätzlich gestützt:

- * LAN (Local Area Network, Verbund von mehreren M20, ausschließlich unter PCOS)
- * Belegung der HDU auch von mehreren Betriebssystemen (PCOS, MS/DOS, CP/M-86, USCD)

VORWORT

Die vorliegende Ergänzung zum Handbuch
PCOS Betriebssystem
beschreibt die Erweiterungen, die unter Release 4.0
vorgenommen wurden.

Die PCOS-Befehle von Release 4.0 sind kompatibel
mit denen von Release 2.0. Ausnahmen:

- pk,
- ck,
- rs, sc und ci,
- vf und vn sowie
- vv.

Die genannten Befehle (sowie die Befehle sp, sf
und ss, falls mit zusätzlichen, unter Rel. 4.0
möglichen Parametern verwendet) müssen beim Arbeiten
mit Release 4.0 in diesem Releasestand eingesetzt
werden.

Alle mit ISAM, Release 2.0 erstellten Filestrukturen
sind kompatibel mit Release 4.0.

BASIC ist im wesentlichen kompatibel, für das
Arbeiten mit Random-Files ist eine zusätzliche
Zugriffsart (Random, nur Lesen) eingeführt. Leichte
Überarbeitungen der Fehlerbehandlungsroutinen
könnten nötig werden, da einige BASIC-Fehlercodes
umgestellt wurden und neue hinzugekommen sind.

PCOS 4.0 unterstützt u.a. folgende zusätzliche Möglichkeiten:

- * weitere Drucker, u.a. Farbausgaben an PRI481
- * eine unter die Betriebssysteme PCOS, MS-DOS, CP/M-86 und UCSD aufgeteilte HDU
- * eine externe HDU (anstelle einer integrierten). Die externe HDU ist genauso anzusprechen wie die integrierte. (Stationsname IO:)
- * LAN (Local Area Network) unter PCOS:
Ein File-Server mit HDU, zusammengeschlossen mit mehreren M20 mit oder ohne HDU und/oder Drucker als Satelliten.
Gemeinsame Nutzung von Files durch Satelliten ist nur auf der Hard-Disk des File-Servers möglich. Die einzelnen Satelliten können jedoch lokale Files (für sich alleine) bearbeiten.
Nach Installation von LAN kann im File-Server kein alternatives Processor-Board (APB) mehr installiert werden, um die 3 anderen möglichen Betriebssysteme einzusetzen. An den anderen Satelliten ist ein APB möglich, sofern sie keine HDU haben.
- * Kopieren von Files auf MS-DOS-formatierten Disketten nach PCOS-formatierten Disketten und umgekehrt.

GLIEDERUNG

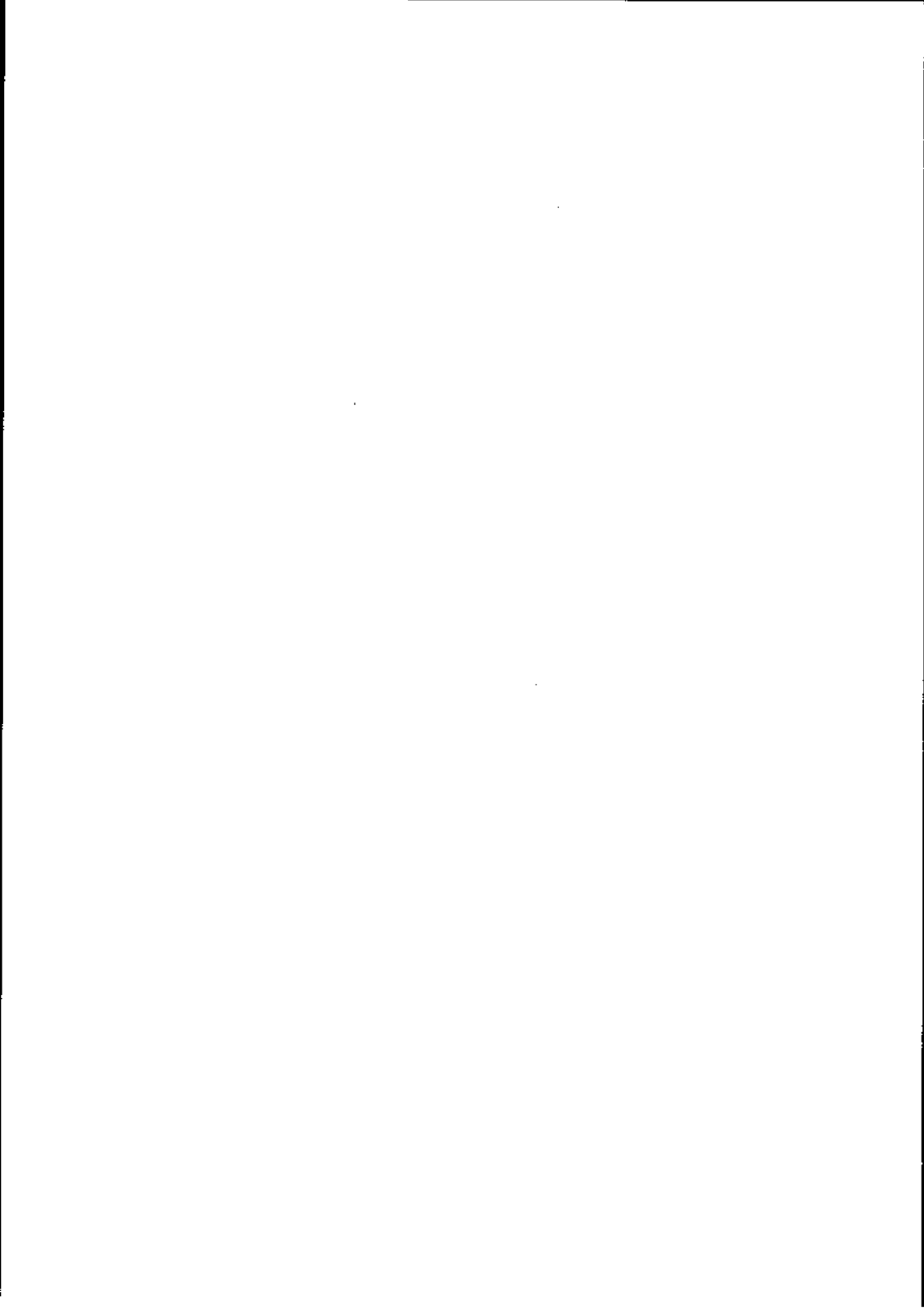
	Seite
0. Begriffserläuterungen, Vorbemerkungen zu PCOS	0.1
0.1 Begriffserläuterungen	0.1
0.2 Kurzgefaßte Vorbemerkungen zu PCOS	0.7
0.2.1 Gliederung des Betriebssystems PCOS	0.7
0.2.2 Speicheraufteilung	0.8
0.2.3 Systemebenen und Übergänge	0.9
0.2.4 Diskettenstationen, Diskettentypen und -verwaltung	0.10
0.2.5 Filetypen und -verwaltung	0.14
0.2.6 mögliche Peripherien	0.17
0.2.7 Besonderheiten bei der HDU-Version	0.19
0.2.7.1 Initialisieren der Hard-Disk-Unit (HDU) bei Erstinstallation	0.20
0.2.7.2 Übertragen zusätzlicher Files von MFD auf HD	0.22
0.2.7.3 Sicherung von auf HD gespeicherten Daten	0.23
0.2.7.4 Übertragen einzelner FD-Files auf Hard-Disk	0.24

1. PCOS im Überblick	1.1
1.1 Bootstrap, Laden des Betriebssystems	1.1
1.2 physischer und logischer Reset	1.5
1.3 Eingabe, Aufruf und Abarbeitung von PCOS-Befehlen	1.6
1.4 residente und nicht-residente PCOS-Befehle	1.8
1.5 Ändern von Default-Werten	1.9
1.6 Tastatur, Belegung und Sonderfunktionen	1.10
1.7 Erstellen von neuen Betriebssystem-Konfigurationen	1.16
1.8 Bildschirmorganisation, Farbbildschirme	1.18
1.9 besondere Graphik-Befehle	1.31
1.10 besondere Hilfen, Tools	1.33
1.11 Verbindungen zu BASIC	1.35

2.	PCOS-Befehle	2.1
2.1	Formatbeschreibung	2.1
2.2	Bildung von Diskettenspezifikationen	2.6
2.3	Bildung von Filenamen	2.11
2.4	Besonderheiten bei Filenamen	2.12
2.5	besondere Syntaxelemente	2.16
2.5.1	Unterdrückung von zusätzlichen Ausgaben; automatische Eingabe	2.16
2.5.2	Filenamenauswahl	2.18
2.5.3	Wechsel des Ein/Ausgabe-Mediums; Prozeduren	2.20
2.6	alphabetisches Verzeichnis der PCOS-Befehle (Seiten im einzelnen siehe nächstes Blatt dieser Gliederung)	2.27
2.7	gegliederter Überblick	2.217

Befehl	Seite	Befehl	Seite
ba	2.27	ps	2.123
bv	2.29	pu	2.127
ci	2.35	rf	2.129
ck	2.39	rk	2.132
dc	2.45	rs	2.133
ed	2.49	sa	2.135
ep	2.69	sb	2.149
fc	2.71	sc	2.153
fd	2.75	sd	2.157
ff	2.77	se	2.159
fk	2.79	sf	2.161
fl	2.81	sl	2.167
fm	2.83	sp	2.171
fn	2.85	ss	2.175
fp	2.89	va	2.179
fr	2.91	vc	2.181
fu	2.93	vd	2.185
fw	2.95	vf	2.187
ie	2.97	vl	2.191
kb	2.99	vm	2.195
la	2.105	vn	2.199
ls	2.111	vp	2.203
lt	2.113	vq	2.205
pk	2.115	vr	2.207
pl	2.119	vv	2.209
pr	2.121	wf	2.213

3. Tastaturen und Zeichensätze	3.1
3.1 Tabelle der Tastencodes und Sonderfunktionen	3.1
3.2 Aufstellung der den Tastencodes zugeordneten Original-ISO-Codes und Sonderfunktionen - unterschieden nach Tastaturen	3.3
3.3 Tastatur-File	3.13
4. PCOS-Fehlermeldungen	4.1
5. Fehlermeldungen während der Autodiagnose und vom Bootstrap	5.1
6. Tabelle zur Umrechnung dezimal/hexadezimal/dual	6.1



0.1 Begriffserläuterungen

Anweisung	BASIC-Sprachelement, bestehend aus einem oder mehreren Schlüsselworten und Parametern; s. Programm- bzw. Direkt-Anweisung
ASCII-File	File aus mit BASIC-Zeilennummern versehenen Programm- oder Textzeilen; die Zeichen werden im ASCII-Format auf das File geschrieben. Die Zeilen sind durch besondere Trennzeichen (dezimaler Code 13 und dezimaler Code 10) getrennt
Assembler-File	File, das ein Assembler-Programm enthält. Auch auf Diskette ausgelagerte PCOS-Befehle sind Assembler-Files
BASIC-Befehl	BASIC-Sprachelement, das üblicherweise außerhalb von Programmen verwendet wird, z.B. zum Erstellen von Programmen oder Dokumentationen; üblicherweise ohne Zeilennummern; einige Befehle können sinnvoll mit Zeilennummer im Programm verwendet werden
BASIC-Ebene	Command- bzw. Direkt-Mode erkenntlich am Ok; nur BASIC-Sprachelemente werden akzeptiert; Edit-Mode erkenntlich an ausgegebener Zeilennummer; Running-Mode zur Abarbeitung

BASIC-Zeilenummer	Steht am Anfang einer BASIC-Programmzeile und zeigt an, wo die Anweisung(sfolge) im Arbeitsspeicher einsortiert wurde; kann zwischen 0 und 65529 einschließlich liegen
Blank	Leertaste, Space (␣)
Bootstrap	Programm im ROM, das den ladbaren Teil des Betriebssystems lädt
Default-Wert	Anfangswert, der vom Betriebssystem selbständig für einen Parameter gesetzt wird
File-Nr.	Lfd. Nummer eines Datenfiles bzw. des zugeordneten File-Puffers (1 bis 15), die beim Öffnen des Files vergeben wird und unter der das File von da an anzusprechen ist
File-Puffer	Von PCOS werden automatisch File-Puffer (Zwischenspeicher) angelegt (physische File-Puffer). In BASIC können diese auf eine gewünschte Länge (logische Länge = Record-Länge) reduziert werden
Full-Screen-Editor	Programm zur Bearbeitung (Erstellen und Ändern) von File-Inhalten am Gesamtschirm
Integer-Wert	Ganzzahl zwischen -32768 und 32767; Typkennzeichen: %

ISAM	Index-sequential access-method; Programmpaket in Assembler zur Verarbeitung von index-sequentiellen Filestrukturen
Key	Schlüssel; Zugriffskriterium in eine index-sequentielle Filestruktur
Mode	Betriebsart in der BASIC-Ebene
Overlay	Programm(teil), das über einem anderen Programm abgearbeitet wird, ohne dieses aus dem Speicher zu löschen
Parameter	Konstante oder variable Größe
Password	Kennwort, ohne dessen Kenntnis ein Zugriff auf die damit geschützte Diskette bzw. das damit geschützte File nicht möglich ist
PCOS-Befehl	Assembler-Routine, utility, Dienstprogramm, von PCOS-Ebene her oder als Overlay in BASIC aufrufbar
PCOS-Ebene	Erkennlich am > -Zeichen; nur PCOS-Befehle werden akzeptiert
PCOS-Textfile	mit Hilfe des Full-Screen-Editors erstelltes File (ohne Parameter %c bei ed-Aufruf)

Programm-Anweisung	BASIC-Anweisung, die im Arbeitsspeicher gespeichert wird und erst im Gesamtprogramm abgearbeitet wird (mit Zeilennummer)
Programmzeile	Zeile mit max. 255 Zeichen mit einer Zeilennummer am Anfang, es folgt eine BASIC-Anweisung bzw. durch : verbundene BASIC-Anweisungen
Puffer	Zwischenspeicher
Random-File	Mit fester Recordlänge (Pufferlänge: max. 4096 Bytes); die Records sind über Ordnungsnummer (Recordnummer) anzusprechen; alle Daten sind in Form von Strings zu organisieren; numerische Werte müssen konvertiert werden; zwischen den Records werden keine automatischen Trennzeichen gesetzt (Byte-Stream); es existiert ein physischer Puffer, dessen Größe beim Öffnen auf gewünschte Länge (logische Größe) reduziert werden kann
Record	Datensatz

Residenter PCOS-Befehl	Ständig im RAM befindlich; zu unterscheiden sind PCOS-Befehle, die vom Laden des Betriebssystems an grundsätzlich im Systemteil des Speichers vorhanden sind, solche, die vom Laden eines (besonders konfigurierten) Betriebssystems an ständig zusätzlich im Anwenderspeicher vorhanden sind und solche, die nur bis zum nächsten Laden des Systems vorhanden sind
Schreibschutz	Verhindert das Verändern des Inhalts einer Diskette oder eines Files
Sequentielles File	Rein sequentielles File, nur Schreib- oder nur Lese-Modus; Anhängen ans Ende möglich; ein n-ter Record des Files kann nur angesprochen werden, wenn die n-1 Records vorher angesprochen worden sind; die Informationen sind im ASCII-Format gespeichert, die Records sind durch besondere Trennzeichen (dezimaler Code 13 und evtl. dezimaler Code 10) getrennt
System-File	Enthält den direkt ladbaren Teil des Betriebssystems oder eine besondere Konfiguration desselben und wird nach Einschalten vom Bootstrap geladen

Variable	BASIC-Speicherplatz, der mit einem symbolischen Namen identifiziert wird; Variablenname: maximal 39 Zeichen lang, mit einem Buchstaben beginnend und keine Sonderzeichen (außer dem Punkt) enthaltend; am Ende (ggf. auch als 40. Zeichen) darf zusätzlich noch ein Typkennzeichen stehen (% = Integer, ! = einfache Genauigkeit, # = doppelte Genauigkeit, \$ = String)
Window	Logisch unabhängiger Teilbildschirm (Fenster); nur in BASIC-Ebene anlegbar
Window-Nummer	Lfd. Nummer des Windows, in dem gearbeitet werden soll; wird beim Eröffnen des Windows vergeben

0.2 Kurzgefaßte Vorbemerkungen zu PCOS

0.2.1 Gliederung des Betriebssystems PCOS

Das Betriebssystem des M20 befindet sich auf Diskette(n).

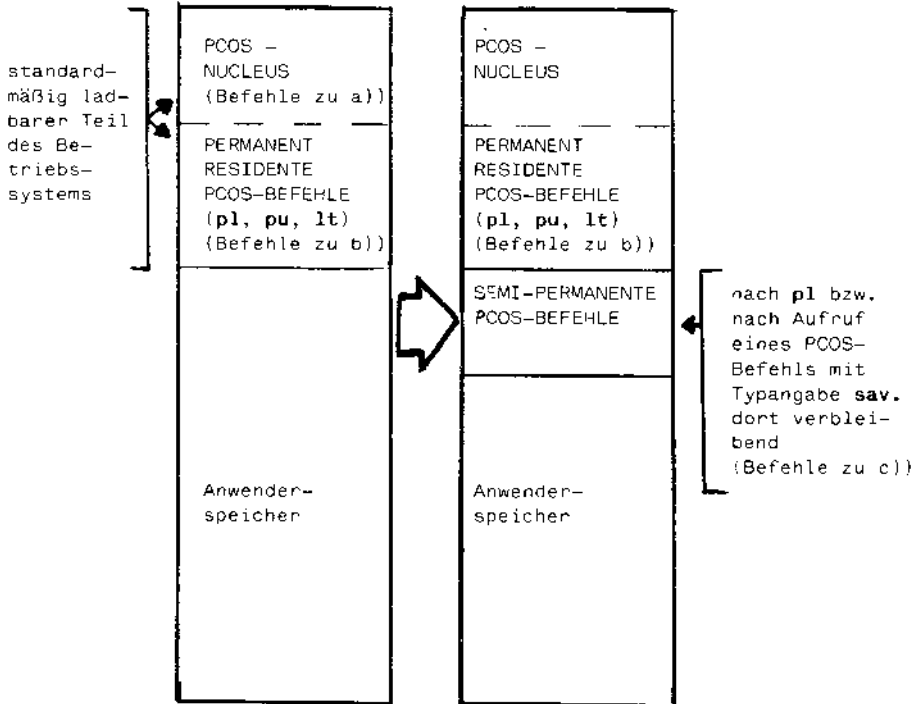
Es ist unterteilt in

- a) den Teil, der nach dem Einschalten des M20 sofort von Systemdiskette in einen dafür reservierten Teil des Speichers geladen wird und von da an bis zum Ausschalten bzw. Neuladen dort verbleibt; dieser Teil heißt im allgemeinen PCOS.SAV und ist nicht als Einzelprogramm aufrufbar.
- b) einzelne Programme (residente PCOS-Befehle), die in a) enthalten sind - entweder standardmäßig oder aufgrund Resident-Machung (vgl. Kapitel 1.4 und 1.7) - und einzeln direkt aus dem Speicher abgerufen werden können, wenn benötigt.
- c) einzelne Programme, die von der Systemdiskette einzeln aufgerufen, in den Speicher geladen und abgearbeitet werden können, wenn sie benötigt werden (nicht residente PCOS-Befehle). Die auf einer Systemdiskette vorhandenen PCOS-Befehle können dem Inhaltsverzeichnis der Systemdiskette (über PCOS-Befehle `vq` bzw. `vl` aufzurufen) entnommen werden. Die Namen solcher PCOS-Befehle enden auf `.cmd` oder `.sav` (Typangabe). Es hängt von der Typangabe ab, ob der betreffende PCOS-Befehl danach im Speicher verbleibt oder nicht (vgl. Kapitel 1.4 und 2.4).

0.2.2 Speicheraufteilung

Die Speichergröße ist abhängig von der Hardware-Ausstattung des M20; die aktuelle Größe kann mit dem PCOS-Befehl **dc** abgefragt werden.

Der Speicher ist gegliedert in:



Der Arbeitsspeicher ist in Segmente unterteilt; Segmentierung und Belegung kann ebenfalls mit **dc** abgefragt werden.

0.2.3 Systemebenen und Übergänge

Zu unterscheiden sind drei Systemebenen:

- a) PCOS-Ebene (erkennlich am >; vor > steht entweder Ø, 1 oder 1Ø; diese Zahl gibt an, auf welcher Disk(etten)-Station zuletzt zugegriffen wurde).

In dieser Ebene können nur PCOS-Befehle eingegeben werden; diese werden dann von der Command-Control-Sprache auf syntaktische Richtigkeit untersucht, gesucht, ggf. geladen und schließlich abgearbeitet. Danach wird wieder > angezeigt (vgl. Kapitel 1.3).

Nach dem Laden des Betriebssystems befindet sich der M20 zuerst immer in der PCOS-Ebene, wenn kein Selbststartprogramm vorhanden ist (vgl. Punkt b)).

- b) BASIC-Ebene (im allgemeinen erkennlich an der Meldung Ok).

In dieser Ebene hat der BASIC-Interpreter Kontrolle (BASIC 8000 wird am M20 ausschließlich interpretativ abgearbeitet); nur BASIC-Sprachelemente dürfen eingegeben werden (vgl. BASIC-Handbuch "BASIC 8000"). BASIC-Programme sind in dieser Ebene zu laden und abzuarbeiten. PCOS-Befehle können hier (auch in laufenden Programmen) in Form von Overlays aufgerufen werden (BASIC-Anweisungen **CALL** und **EXEC**).

Der Aufruf der BASIC-Ebene geschieht durch den PCOS-Befehl **ba** oder automatisch nach dem Laden des Betriebssystems, wenn auf einer der einliegenden Disketten ein Selbststartprogramm namens **INIT.BAS** (BASIC-Programm-File) vom Betriebssystem gefunden wird (vgl. Kapitel 1.1 und 2.4).

Der Übergang von der BASIC- in die PCOS-Ebene kann mit Hilfe des BASIC-Befehls **SYSTEM** erreicht werden.

c) Ebene des Full-Screen-Editors

In dieser Ebene ist fast der gesamte Arbeitsspeicher zur Verwendung des PCOS-Befehls **ed** reserviert. Dieser ruft den Full-Screen-Editor auf, mit dessen Hilfe Files (PCOS-Textfiles oder BASIC-Files in ASCII- oder sequentiellm Format) über den gesamten Bildschirm mit Hilfe von Cursorsteuerung und besonderen Funktionen editiert werden können.

0.2.4 Diskettenstationen, Diskettentypen und -verwaltung

Die rechte Diskettenstation hat den Namen **0:**, die linke (falls vorhanden) den Namen **1:**. Ist rechts eine Hard-Disk installiert, hat diese Station den Namen **10:** und die Diskettenstation den Namen **0:**. (Genauerer vgl. Kapitel 0.2.7, Kapitel 2.2)

Folgende Diskettentypen stehen zur Verfügung

- a) 320-KB-Disketten (doppelseitig, doppelte Dichte). Diese können nur in Stationen mit zwei Schreib/Lese-Köpfen eingesetzt werden.

Die Kapazität für Anwenderdaten und -Programme beträgt nach Formatieren mit **vf** 282.634 Bytes. Aufbau: 2 Seiten zu je 40 Spuren (davon verwendet: 35) zu je 16 Sektoren zu je 256 Bytes (Ausnahme: Spur 0, Seite 0 mit je 128 Bytes/Sektor).

- b) 160-KB-Disketten (einseitig, doppelte Dichte). Diese können sowohl in Stationen mit zwei Schreib/Lese-Köpfen als auch in solchen mit nur einem Schreib/Lese-Kopf eingesetzt werden.

Die Kapazität für Anwenderdaten und -Programme beträgt nach Formatieren mit **vf** 139.264 Bytes. Aufbau: 40 Spuren (davon verwendet: 35) zu je 16 Sektoren zu je 256 Bytes (Ausnahme: Spur 0 mit je 128 Bytes/Sektor).

- c) 640-KB-Disketten (doppelseitig, doppelte Dichte). Diese können nur in Stationen für 640-KB-Disketten eingesetzt werden.

Die Kapazität für Anwenderdaten und -Programme beträgt nach Formatieren mit **vf** 638.976 Bytes. Aufbau: 2 Seiten zu je 80 Spuren zu je 16 Sektoren zu je 256 Bytes (Ausnahme: Spur 0, Seite 0 mit je 128 Bytes/Sektor).

In 640-KB-Stationen kann auf 640-KB-Disketten sowohl geschrieben als auch gelesen werden; 320- und 160-KB-Disketten können dort nur gelesen werden.

In 320-KB-Stationen können sowohl 320- als auch 160-KB-Disketten geschrieben und gelesen werden, aber keine 640-KB-Disketten.

Bei **vf** bzw. **vn** für 160-KB-Disketten muß in 320-KB-Stationen ein besonderer Parameter (**%s**) angegeben werden.

In 160-KB-Stationen können nur 160-KB-Disketten gelesen und geschrieben werden (Kein **%s** bei **vf** bzw. **vn**).

Für alle drei Diskettentypen gelten die folgenden technischen Daten:

Durchmesser:	5 1/4 Zoll
durchschnittliche Zugriffszeit:	303 ms
Übertragungsrate:	250 KBit/Sek.

d) Weiterhin steht eine integrierte Winchester-Festplatte ("Hard-Disk") mit 11,25 MB (unformatiert) zur Verfügung (nur bei HDU-Version):

6 Schreib/Leseköpfe (1 pro Oberfläche)

Kapazität nach **vf**: 8,844 MB

6 Oberflächen zu je 180 Spuren zu je 32 Sektoren zu je 256 Bytes

Durchmesser: 5 1/4 Zoll

Durchschnittliche Zugriffszeit: 66 ms

Auf allen Disketten bzw. der Hard-Disk wird von PCOS ein Inhaltsverzeichnis und eine Bit-Map geführt, in der vermerkt ist, welche Sektoren der Disk(ette) aktuell von Files belegt sind. Spur 0 auf Seite 0 wird bei **vf** mit Standard-Kontroll-Daten

beschrieben und von PCOS nicht benutzt.

Zur Verwaltung von Disketten bzw. Hard-Disk vgl. Kapitel 2.2 und alle PCOS-Befehle, die mit **v** beginnen:

1. Formatieren (physisches Löschen bzw. Vorbereiten einer Diskette): **vf**
2. Inhaltsverzeichnis löschen: **vn**
3. Kopieren: **vc**, **vm**, aber auch **fc**
4. Inhaltsverzeichnis: **va**, **vq**, **vl** (s. auch BASIC-Befehl FILES)
5. Vergabe/Änderung Diskettenname: **vr**
6. Disketten-Passwords: **vp**, **vd**
7. diverse in BASIC-Programmen benötigte Informationen: **bv**
8. Ansprechen gewünschte Diskettenstation: **se**
9. Überprüfen Hard-Disk: **vv**
10. Kopieren von Hard-Disk auf Diskette(n) + umgekehrt: **sa**

0.2.5 Filetypen und -verwaltung

PCOS kann folgende Filetypen verwalten:

- | | |
|-----------------------------|---|
| a) Systemfile | direkt ladbarer Betriebssystemteil auf Diskette |
| b) Assemblerfile | PCOS-Befehle; Assembler-Programme |
| c) BASIC-Programm-Files | Komprimiert; Schutzmöglichkeit gegen Einsicht und Änderung (Protect) |
| d) ASCII-Files | im ASCII-Format mit Zeilennummern; Textzeilen oder Programm-Moduln |
| e) Sequentielle Files | im ASCII-Format ohne Zeilennummern; Datenfiles |
| f) Random-Files | im ASCII- und binär gepackten Format; Datenfiles |
| g) PCOS-Textfiles | mit Hilfe des Full-Screen-Editors (ohne Parameter %c) erstelltes File |
| h) Index-sequentielle Files | (nur unter Zuhilfenahme des Zusatzpaketes ISAM); Kombination von 2 Files: File für Datensätze = normales Random-File + File für Keys = von ISAM verwaltet |

Alle Files sind ursprünglich als reine Folge von Bytes (Byte-Stream) zu betrachten, da PCOS selbst keine automatischen Trennzeichen setzt (Ausnahme: PCOS-Textfiles).

Zum Zugriff müssen über PCOS-File-Puffer angelegt werden (vgl. PCOS-Befehl **sb**).

Files werden auf der Diskette dynamisch verwaltet; d.h., ein File kann per Programm angelegt und vergrößert, über den PCOS-Befehl **ff** aber auch auf die logische Länge reduziert werden (auf soviel Sektoren, wie durch den Record im höchsten beschriebenen Sektor festgelegt sind).

Minimum für ein File: 2 Sektoren (erster Sektor wird zur Adressierung benötigt)

Freie Sektoren 320-KB-Diskette: doppelseitig,
nach **vf** doppelte Dichte, 1088 Sektoren frei
 (35 Spuren je Seite, je 16 Sektoren,
 je 256 Bytes).

160-KB-Diskette: einseitig, doppelte Dichte, (35 Spuren, je 16 Sektoren, je 256 Bytes)

Hard-Disk: 34.296 Sektoren zu je 256 Bytes frei

Ein File kann von PCOS selbständig angelegt werden; es wird im Vielfachen des im PCOS-Befehl **ss** gesetzten Parameters 'anzulegen- de Sektoren' angelegt. Man kann mit dem PCOS-Befehl **fn** gezielt eine bestimmte Anzahl aufeinander folgender Sektoren reservieren. Muß ein File vergrößert werden, wird es analog zu oben automatisch vergrößert.

Ist die Position hinter dem letzten Sektor des Files bereits belegt, wird es im nächsten als frei gekennzeichneten Sektor von PCOS selbständig fortgesetzt (sog. 'Extent-Bildung').

Zugriffe auf Files in mehreren Extents dauern länger.

Mit Hilfe des PCOS-Befehls **fc** werden Files so kopiert, daß sie danach mit Sicherheit nur aufeinanderfolgende Sektoren belegen.

Beim Anlegen und Löschen von Files werden nur im Inhaltsverzeichnis die betreffenden Sektoren als belegt bzw. wieder frei vermerkt (Bit-Map). Die Vorbelegung des Files selbst mit **CHR\$(Ø)**, wie durch das Programm zur physischen Formatierung (**vf**) erreicht, findet dabei nicht statt.

Multi-Volume-Files sind nicht möglich.

Zur genaueren Beschreibung vgl.:

- | | |
|--|---|
| 1. Anlegen eines Files: | fn, ed (s. auch BASIC-Anweisung OPEN und BASIC-Befehl SAVE) |
| 2. Namen ändern: | fr (s. auch BASIC-Befehl NAME) |
| 3. Kopieren eines oder mehrerer Files: | fc, fm , bei HD auch sa |
| 4. Schreibschutz: | fw, fu |
| 5. File-Password: | fp, fd |
| 6. Verkürzen: | ff |
| 7. Löschen, Reaktivieren: | fk, rk (s. auch BASIC-Befehl KILL) |
| 8. Listen ASCII-Inhalt: | fl |

0.2.6 mögliche Peripherien

Neben den in der Basis fest integrierten Peripherien Bildschirm und Tastatur unterstützt PCOS folgende Schnittstellen:

a) eine Schnittstelle für Parallel-Anschluß (Centronics-like)

Dort werden die für den M20 angebotenen Drucker von Olivetti standardmäßig angeschlossen. Mit dem PCOS-Befehl `sf` kann die Parallel-Schnittstelle definiert werden. Die Ausgabe an die parallele Schnittstelle kann entweder über die BASIC-Anweisung `LPRINT` oder über Wechsel des Ein/Ausgabe-Mediums (vgl. Kapitel 2.5.3) erfolgen.

b) in der Basis-Ausstattung standardmäßig enthalten:

eine Schnittstelle für seriellen Anschluß (nach V.24- bzw. RS232C-Norm). Dort können auf Wunsch alle von Olivetti angebotenen Drucker auch seriell angeschlossen werden. Selbstverständlich können über diese Schnittstelle alle Peripherien bedient werden, die mit einer gemäß RS232C genormten Schnittstelle versehen sind (vgl. die PCOS-Befehle `sf`, `sd`, `rs`, `sc`, und `ci` sowie das Handbuch "V24.-Programmierung"). Die Ein- oder Ausgabe über eine RS232C-Schnittstelle kann auch über Wechsel des Ein/Ausgabe-Mediums (vgl. Kapitel 2.5.3) erfolgen.

c) Erweiterungen:

1. mit einer zusätzlichen Platte:

- entweder zwei weitere V.24-Schnittstellen
- oder eine weitere V.24-Schnittstelle und eine Current-loop-(20mA-)Schnittstelle
- oder zwei Current-loop-Schnittstellen (20-mA-Schnittstellen)

2. mit einer anderen zusätzlichen Platte:

eine IEC-(IEEE488)-Schnittstelle.

Zur Programmierung dieser Schnittstelle vgl. PCOS-Befehl `ie` sowie das Handbuch "IEC-Programmierung". Die Ein- oder Ausgabe über die IEEE488-Schnittstelle kann auch über Wechsel des Ein/Ausgabe-Mediums (vgl. Kapitel 2.5.3) erfolgen.

Es ist möglich, alle Schnittstellen gleichzeitig unabhängig voneinander zu verwalten.

0.2.7 Besonderheiten bei der HDU-Version des M20

Alle für M20 gültigen PCOS-Befehle sind auch in der HDU-Version möglich.

Für Diskettenspezifikationen und Filenamen gelten - abweichend von der FDU-Version - folgende Regeln:

1. Die FD-Station **0**: (rechts) ist durch die HDU ersetzt und hat den Namen **10**:.
2. Die FD-Station **1**: (links) hat in der HDU-Version den Namen **0**:.

Es gibt einen besonderen zusätzlichen PCOS-Befehl **vv**, mit dem sich die Organisation der HD durch PCOS beeinflussen läßt.

Beim Laden des Betriebssystems von HD lädt der Bootstrap grundsätzlich die zuletzt mit **ps** auf HD abgespeicherte Betriebssystem-Konfiguration. Davon abweichende Konfigurationen können mit **pr** aufgerufen werden.

0.2.7.1 Initialisieren der Hard-Disk-Unit (HDU) bei Erstin- stallation

1. PCOS von der Systemdiskette laden:

Diskette PCOS-Rel. 2.0 in das Floppy-Disk-Laufwerk links (Name der Station ist hier 10:) einlegen. System einschalten und F-Taste gedrückt halten, bevor das akustische Signal ertönt. Hierdurch wird vom Bootstrap PCOS von der MFD geladen.

Achtung: Soll das Betriebssystem von der FD geladen werden, muß vorher ein physischer Reset erfolgen. Ein logischer Reset (Drücken von CTRL RESET) reicht dazu nicht aus.

2. Danach muß die HD (Stationsname: 10:) mit den drei folgenden PCOS-Befehlen formatiert und auf die Aufnahme von Files vorbereitet werden:

vf 10: (Der : wird hier noch durch Drücken der Ä-Taste erzeugt.)

vv 10:

vf 10:

vn 10:

Achtung: **vn** muß erfolgen!

3. Mit dem PCOS-Befehl **sl 1** ist danach die deutsche Tastatur DIN 2137 zu aktivieren.

4. Mit dem PCOS-Befehl **ps 10:** ist das Betriebssystem unter dem Namen PCOS.SAV auf die HD abzuspeichern. Von da an lädt der Bootstrap PCOS stets von HDU, sofern nicht während des Ladens die F-Taste gedrückt gehalten wird.

5. Mit dem PCOS-Befehl **fc 0:* 10:** sind alle PCOS-Befehle von der Systemdiskette auf die HD zu kopieren. Beim Kopierversuch des Files PCOS.SAV muß die Frage **'File already exists. Do you wish to overwrite (y or n)?'** mit **n** beantwortet werden.

6. Updates von neuen PCOS-Versionen auf eine bereits bespielte HD erfolgen mit dem PCOS-Befehl **fc %f 0:* 10:.**

0.2.7.2 Übertragen zusätzlicher Files von MFD auf HD

1. gewünschte MFD in Station \emptyset : einlegen
2. Eingabe von `fc \emptyset :* l \emptyset :`
3. Existiert ein bestimmtes File schon auf HD, wird 'File already exists. Do you wish to overwrite (y or n)?' gemeldet. Soll das File auf HD durch das FD-File ersetzt werden, ist `y` einzugeben, andernfalls `n`.
4. Sollen Files grundsätzlich übertragen werden - also auch wenn sie bereits vorhanden sind (sog. "force-copy") - muß nach `fc %f` eingegeben werden.

Beispiel: `fc %f \emptyset :* l \emptyset :`

0.2.7.3 Sicherung von auf HD gespeicherten Daten

Die auf HD gespeicherten Daten sind in Form von Files organisiert. Für das Sicherungsverfahren müssen nur diejenigen (Daten-)Files auf eine Sicherungs-MFD kopiert werden, die Änderungen unterliegen. Da eine 320-KB-MFD maximal 280 KB Anwenderdaten aufnehmen kann und PCOS keine Multi-Volume-Files unterstützt, sollte kein solches File auf HD größer als 280 K sein. Ist eine 640-KB-FD-Station vorhanden, gilt das Ebengesagte analog.

Die Sicherung eines Files geschieht durch Eingabe des PCOS-Befehls **fc lØ:'Filebezeichnung' Ø:**, wobei für 'Filebezeichnung' der Name des zu kopierenden Files anzugeben ist.

Beispiel: Das File ARlØl soll von HD auf FD gesichert werden. Eingabe von **fc lØ:ARlØl Ø:**. Die Frage 'File already exists...' wird mit **y** beantwortet.

Sind die Filenamen so aufgebaut, daß sie gemeinsame, die Zieldiskette bezeichnende Bestandteile am Namen haben (Filenamenauswahl; vgl. PCOS-Handbuch), kann der PCOS-Befehl **fc** in der Form **fc lØ:'Filenamenauswahl' Ø:** eingesetzt werden. Zur Meldung 'File already exists...' vgl. Kapitel 0.2.7.2.

Beispiele: **fc lØ:??PAKET.s Ø:**
fc %f lØ:*b Ø:

Mit Hilfe des von der Deutschen Olivetti entwickelten besonderen PCOS-Befehls **save.cmd** können auch Files von HD auf FD und zurück übertragen werden, die größer als 280 KB bzw. 620 KB - je nach Art der FD-Station - sind.

0.2.7.4 Übertragen einzelner FD-Files auf Hard-Disk

Um ein bestimmtes File von FD auf HD zurückzuübertragen, ist folgendermaßen vorzugehen:

1. FD links einlegen
2. Eingabe von `fc %f Ø:'Filebezeichnung' 1Ø:`, wobei für 'Filebezeichnung' der Name des zu übertragenden Files anzugeben ist.

Beispiel: `fc %f Ø:AR1Ø1 1Ø:`

1.1 Bootstrap, Laden des Betriebssystems

Im ROM (EPROM) des M20 befinden sich folgende stets vorhandene Firmware-Programme:

- Autodiagnose, zur Selbstüberprüfung nach physischem Reset (vgl. Kapitel 1.2);
- Bootstrap, zum Laden des direkt ladbaren Teils des Betriebssystems;
- Verwaltung der Echtzeituhr.

Nach physischem Reset (vorher Drucker einschalten, darf nicht auf LOCAL stehen) wird zuerst die Autodiagnose durchgeführt. In dieser Phase sollten keine Disketten in den Schächten liegen. Bei Fehlermeldungen während der Autodiagnose vgl. Kapitel 5.

Nach erfolgreicher Autodiagnose meldet der Bootstrap **Insert diskette and type any key**. Jetzt ist die Systemdiskette einzulegen (möglichst Station 0: = rechts) und, falls später die andere Station benötigt wird, dort eine beliebige (initialisierte) Diskette. Anschließend ist eine beliebige Taste zu betätigen. Wird danach erneut **Insert diskette and type any key** - zusammen mit einer anderen Meldung - ausgegeben, kann dies folgende Ursachen haben:

- a) es liegt keine oder eine schadhafte Systemdiskette ein;
- b) die Systemdiskette liegt links (Station 1:) ein, rechts liegt keine Diskette ein, der Schacht ist aber geschlossen;

c) eine Diskettenstation oder beide sind schlecht justiert.

Im Normalfall wird der direkt ladbare Teil von PCOS in den permanenten Bereich des Arbeitsspeichers geladen, so daß danach beide Diskettenstationen für Anwenderprogramme und -Daten frei werden. (Ausnahme: Es werden zwischendurch nichtresidente PCOS-Befehle aufgerufen oder andere Betriebssystem-Konfigurationen neu geladen). Der Bootstrap sucht erst auf der Diskette in Station 0: nach einem File, das vom Inhalt her den ladbaren Teil beinhaltet. Üblicherweise hat dieses den Namen PCOS.SAV, dieser ist jedoch nicht zwingend. Wird in Station 0: kein solches File gefunden, wird die Suche auf Station 1: fortgesetzt.

Nach dem Laden des Betriebssystems bestehen folgende Möglichkeiten:

a) das System geht sofort in die PCOS-Ebene über und meldet sich mit 0> bzw. mit 1> als bereit zum Entgegennehmen von PCOS-Befehlen. Vor dem >-Zeichen (Prompt) wird immer angegeben, auf welcher Station zuletzt zugegriffen wurde. Dies ist wichtig für den Zugriff auf Files, bei deren Namen am Anfang nicht der Disketten-Identifizier angegeben ist (vgl. hierzu Kapitel 2.4).

b) Es wird ein automatisches Selbststartprogramm abgearbeitet. Dafür gibt es drei Möglichkeiten:

1. Auf einer der einliegenden Disketten befindet sich ein Assembler-File namens INIT.CMD.
2. Auf einer der Disketten befindet sich ein Assembler-File namens INIT.SAV.

3. Auf einer der Disketten befindet sich ein BASIC-Programm-File oder ASCII-File namens INIT.BAS.

Das betreffende Programm wird dann sofort geladen und zur Ausführung gebracht. Bei INIT.BAS wird vorher noch automatisch der BASIC-Interpreter geladen (vgl. PCOS-Befehl `ba`) und in die BASIC-Ebene verzweigt. Die Files `basic.cmd`, `basic.abs` und `INIT.BAS` müssen auf der gleichen Diskette sein.

Die Suche nach einem Selbststartprogramm geschieht in folgender Reihenfolge:

1. Ist auf Station 0: INIT.CMD?

wenn ja, Laden, Übergang in die PCOS-Ebene und Abarbeiten. INIT.CMD wird danach wieder aus dem Arbeitsspeicher entfernt; das System befindet sich in der PCOS-Ebene.

wenn nein, Suche auf Station 1:.

2. Ist kein INIT.CMD vorhanden:

Ist auf Station 0: INIT.SAV?

wenn ja, Laden, Übergang in die PCOS-Ebene und Abarbeiten. Danach bleibt INIT.SAV im Speicher resident; das System befindet sich in der PCOS-Ebene.

wenn nein, Suche auf Station 1:.

3. Ist auch kein INIT.SAV vorhanden:

Suche nach INIT.BAS (erst auf 0:, dann auf 1:). Wird INIT.BAS gefunden, wird auf derselben Diskette nach `basic.cmd` und `basic.abs` gesucht. Wird eines dieser Files nicht gefunden,

erfolgt der Übergang in die PCOS-Ebene ohne Laden von INIT.BAS. Andernfalls wird **ba** geladen, die BASIC-Ebene aufgerufen und INIT.BAS abgearbeitet. Das System verbleibt danach in der BASIC-Ebene, es sei denn, dort wird der BASIC-Befehl **SYSTEM** gegeben.

Selbstverständlich können alle drei Programmfiles vom Anwender selbst erstellt und dem jeweiligen Programmpaket angepaßt werden. (z.B. Residentmachen ständig benötigter PCOS-Befehle, Setzen der Echtzeituhr, Ändern von Systemparametern etc.).

Wird während des Ladens des Betriebssystems die Taste **B** gedrückt gehalten, wird sofort der BASIC-Interpreter geladen (sofern auf der Diskette mit dem ladbaren Teil gespeichert) und in die BASIC-Ebene verzweigt, unabhängig vom Vorhandensein eines Startprogramms.

Wird während des Ladens die Taste **S** gedrückt gehalten, wird immer in die PCOS-Ebene verzweigt.

In beiden Fällen werden alle evtl. vorhandenen Selbststartprogramme unterdrückt.

1.2 Physischer und logischer Reset

Der physische Reset wird erreicht durch Aus- und Wiedereinschalten (ON/OFF-Schalter) oder - besser - durch Druck auf den Kontakt im Loch an der Basis des M20 (rechts außen, von vorne gesehen). Das Betätigen der Tastenkombination CONTROL (blaue Taste) und gleichzeitig RESET bewirkt den logischen Reset. Auch danach muß das Betriebssystem neu geladen werden, aber Autodiagnose und Rücksetzen der Echtzeituhr auf Defaultwert werden nicht durchgeführt (vgl. zum Setzen der Echtzeituhr den PCOS-Befehl `ss` sowie im BASIC-Handbuch die speziellen Variablen `DATE$` und `TIME$`).

Sowohl beim logischen als auch beim physischen Reset wird der gesamte Speicher gelöscht. Nach Laden des Betriebssystems sind diejenigen Default-Werte gesetzt, die im neugeladenen Betriebssystem Gültigkeit haben.

1.3 Eingabe, Aufruf und Abarbeitung von PCOS-Befehlen

PCOS-Befehle sind Assembler-Programme, die entweder von Diskette oder direkt aus dem Arbeitsspeicher aufgerufen werden können und Bestandteile des Betriebssystems sind. Zur Unterscheidung der Typen vgl. Kapitel 1.4 und 2.4, zur Konfiguration eines Betriebssystems vgl. Kapitel 1.7. Sie bestehen aus dem Namen des Files als Schlüsselwort (z.B. basic.cmd) - auf Diskette oder im Speicher - und bis zu 18 Parametern. Insgesamt können maximal 256 Zeichen eingegeben werden, zur Trennung von Parametern ist das Komma vorgesehen; stattdessen kann aber auch Blank gesetzt werden.

Zum Aufruf reichen die ersten beiden Buchstaben des PCOS-Befehls. Groß- und Kleinbuchstaben werden dabei nicht unterschieden.

Beginnen zwei Files mit den gleichen zwei Buchstaben und haben als Typangabe entweder `.cmd`, `.sav` oder `.bas` am Ende des Namens, wird stets derjenige aufgerufen, der entweder bereits im Arbeitsspeicher ist oder auf Diskette weiter vorne im Inhaltsverzeichnis steht. Alle PCOS-Befehle können im allgemeinen ohne Diskettenspezifikation aufgerufen werden.

Die Suche findet in folgender Reihenfolge statt:

1. Ist der PCOS-Befehl im Arbeitsspeicher? (Reihenfolge der Überprüfung der Typangabe: erst `.cmd`, dann `.sav`, dann `.bas`);
2. Ist der PCOS-Befehl auf der zuletzt angesprochenen Diskette? (Reihenfolge nach Typangabe wie bei 1.);

3. Ist der PCOS-Befehl auf der anderen Diskette?;
4. Wird kein PCOS-Befehl mit dem angegebenen Schlüsselwort gefunden, wird Error 92 gemeldet.
5. Ist der betreffende Befehl nicht mit dem Release kompatibel, das beim Laden zugrunde lag, wird Error 99 gemeldet.

Sind optionale Parameter nicht angegeben, wird der bis zum Aufruf gültige Wert dafür angenommen. Für Parameter, deren Werte unverändert übernommen werden sollen, ist nur der Platzhalter ,(Komma) zu setzen.

PCOS-Befehle werden im allgemeinen in der PCOS-Ebene aufgerufen; mittels der BASIC-Anweisungen **CALL** bzw. **EXEC** können sie auch in der BASIC-Ebene aufgerufen und abgearbeitet werden, ohne laufende BASIC-Programme oder vorhandene BASIC-Daten aus dem Arbeitsspeicher zu löschen. **CALL** enthält alle Möglichkeiten von **EXEC**, ist aber leistungsfähiger, da damit auch Werte von PCOS an den BASIC-Interpreter übergeben werden können (vgl. z.B. PCOS-Befehl **It**).

1.4 Residente und nicht-residente PCOS-Befehle

In dem standardmäßig von OLIVETTI ausgelieferten Systemfile PCOS.SAV sind neben dem Nukleus des Betriebssystems folgende einzeln aufrufbare PCOS-Befehle von Anfang an resident (im Arbeitsspeicher vorhanden); diese können also nie aus dem Arbeitsspeicher entfernt werden:

lt, **pl** und **pu**.

Alle anderen PCOS-Befehle sind standardmäßig von Systemdiskette zu laden und abzuarbeiten (nicht-resident). Einige werden nach Abarbeitung automatisch wieder aus dem Speicher entfernt, so daß beim nächsten Aufruf wieder die Systemdiskette einliegen muß (Typangabe am Ende des Filenamens: **.cmd**). Andere bleiben nach erstmaligem Laden bis zum nächsten Reset resident (Typangabe: **.sav**).

Es ist mit Hilfe des PCOS-Befehls **pl** möglich, einen PCOS-Befehl vom Typ **.cmd** bis zum nächsten Reset resident zu machen. Für PCOS-Befehle vom Typ **.sav** ist dies nicht nötig.

Mit Hilfe des PCOS-Befehls **pu** ist es möglich, solche zusätzlich resident gemachten PCOS-Befehle wieder aus dem Speicher zu entfernen. (Ausnahme: Die bei **pu** gesondert aufgeführten PCOS-Befehle, da diese die Adressierung im Arbeitsspeicher verändern.)

1.5 Ändern von Default-Werten

Für die Drucker, den BASIC-Interpreter, PCOS selbst, evtl. vorhandene RS232C-(V24-)Schnittstellen, die Namen von angeschlossenen Einheiten, zugrundeliegende nationale Tastatur und Bildschirmzeichensatz sind nach Laden des Betriebssystems automatisch bestimmte Default-Werte gesetzt. Diese können mit folgenden PCOS-Befehlen geändert werden:

ss: PCOS selbst (Echtzeit, Bildschirmformat, Filesystem, Verifikation)

sb: BASIC-Interpreter (max. Recordlänge, max. File-Anzahl, max. Window-Anzahl, Arbeitsspeichergröße)

sf: Drucker (PCOS-gesteuerte Druckparameter, Anschlußart)

sc: RS232C-(V24-)Schnittstelle(n) (Übertragungseigenschaften)

sl: Tastatur (zugrundeliegender Zeichensatz)

sd: angeschlossene Einheiten (Namen)

wf: Bildschirmzeichensatz

ck: Tastencodes und zugeordnete Tastaturfunktionen bzw. ISO-Codes

pk: geänderte Tastaturbelegungen (Zeichen(folgen))

Zur Arbeit mit Files in der BASIC-Ebene sind die PCOS-Befehle **ss** und vor allem **sb** von besonderer Bedeutung.

1.6 Tastatur, Belegung und Sonderfunktionen

Alle Tasten(kombinationen) haben stets einen bestimmten Tastencode. Diese Tastencodes sind unabhängig von der nationalen Tastatur (vgl. Kapitel 3.1).

Die den Tastencodes zugeordneten Sonderfunktionen bzw. (Original-) ISO-Codes sind durch die nationale Tastatur festgelegt (vgl. Kapitel 3.2). Die Zuordnung kann mit Hilfe des PCOS-Befehls **ck** geändert werden. Sonderfunktionscodes haben Vorrang.

Jedem ISO-Code ist im Arbeitsspeicher eine Punktmatrix zugeordnet, in der festgehalten ist, welches Punktraster am Bildschirm (und nur dort) auszugeben ist, wenn der betreffende ISO-Code angefordert wird. Die zugeordnete Punktmatrix kann mit Hilfe der PCOS-Befehle **rf** und **wf** unter Zuhilfenahme des Full-Screen-Editors (PCOS-Befehl **ed**) geändert werden (vgl. zum Aufbau des Tastaturfiles Kapitel 3.3).

Wird eine Tastenbelegung mit Hilfe des PCOS-Befehls **pk** geändert, bewirkt dies einen Verweis in eine weitere interne Tabelle im Arbeitsspeicher. In dieser ist festgehalten, welches Zeichen bzw. welche Zeichenfolge anstelle des durch den Original-ISO-Code definierten Zeichens erzeugt werden soll. Dies gilt auch für andere Ausgabeeinheiten.

Mit **ps** können beliebige Zwischenstände abgespeichert werden, die danach sofort nach Laden des betreffenden Betriebssystems gültig sind.

Die Cursor-Steuerung ist nur in der Ebene des Full-Screen-Editors oder in BASIC-Programmen aktiv, die dies vorsehen, also auf keinen Fall innerhalb der PCOS-Ebene oder innerhalb der BASIC-Ebene bei der Eingabe von BASIC-Befehlen oder -Programmzeilen.

Bestimmten CONTROL- bzw. COMMAND-Kombinationen können in bestimmten Programmen besondere Funktionen zugeordnet sein (z.B. im Full-Screen-Editor; vgl. PCOS-Befehl **ed**).

Standardmäßig gelten folgende Sonderfunktionen:

SHIFT (rechts oder links) :	1	<table border="1"><tr><td>CONTROL =</td></tr><tr><td>blaue Taste</td></tr><tr><td>COMMAND =</td></tr><tr><td>gelbe bzw. rote Taste</td></tr></table>	CONTROL =	blaue Taste	COMMAND =	gelbe bzw. rote Taste
CONTROL =						
blaue Taste						
COMMAND =						
gelbe bzw. rote Taste						
CONTROL + C :	2					
CONTROL + H :	3					
CONTROL + I :	4					
CONTROL + G :	5					
CONTROL + M :	6					
CONTROL + S :	7					
CONTROL + RESET :	8					
CONTROL + - :	9					
COMMAND + - :	10					

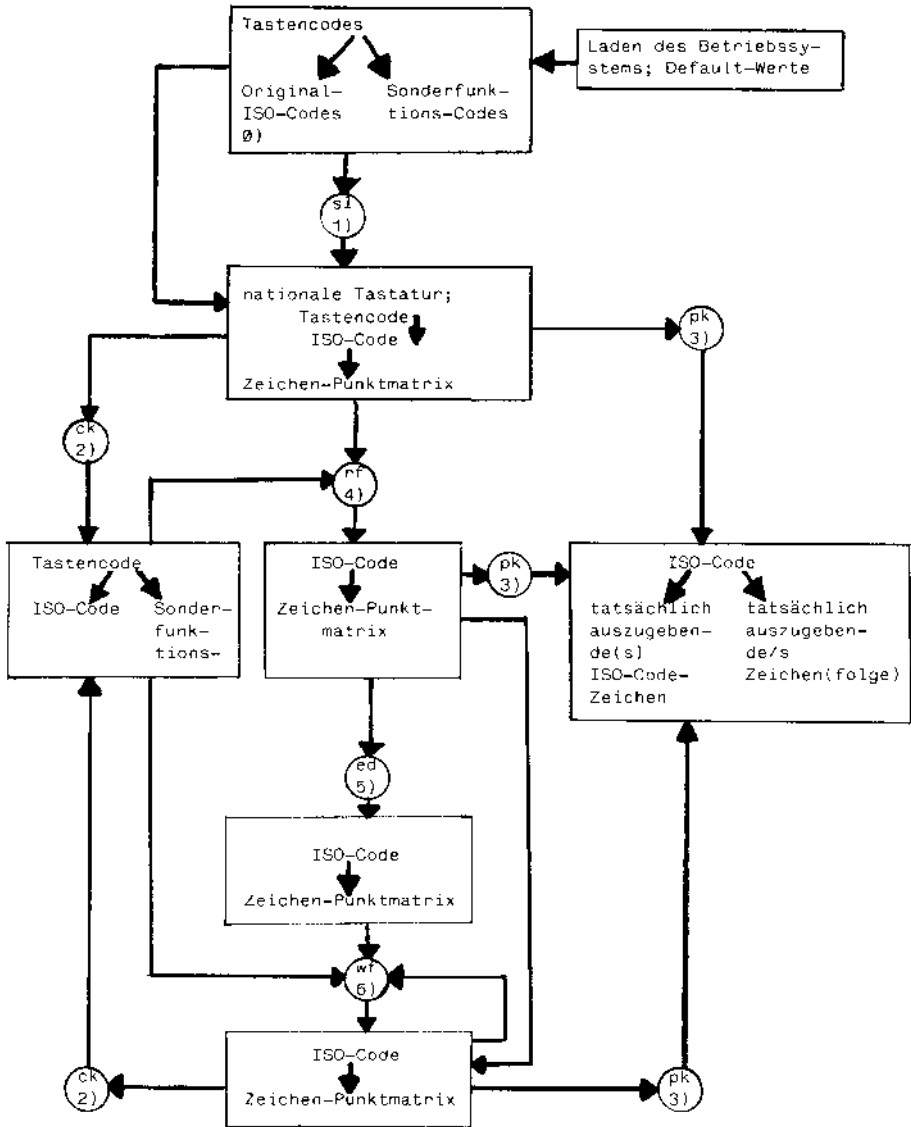
} - unten rechts im alpha-numerischen Feld

Die genannten Tasten sind gleichzeitig zu drücken.

- 1 Umschaltung von Klein- auf Großschreibung bzw. vom unteren auf das obere Zeichen
- 2 Programmabbruch; es kann damit auch ein fälschlich eingegebener Befehl (vor Auslösen durch CR) neutralisiert werden
- 3 Löschen des zuletzt eingegebenen Zeichens und Rückpositionierung des Cursors
- 4 Tabulation auf die nächste Spalte mit der Nr. 1, 9, 17, 25 etc.
- 5 verdeckte Eingabe; Zeichen wird zwar angenommen, nicht aber am Bildschirm ausgegeben (besondere Form des Cursors); Aufhebung durch erneut **G** oder Eingabeabschlußtaste (CR, S1 oder S2)
- 6 Wirkung wie CR; der von lt gelieferte Wert bleibt unverändert
- 7 Pausenfunktion (z.B. zum Anhalten eines Bildschirmlistings); Aufhebung durch beliebigen anderen Tastendruck
- 8 Neuladen des Betriebssystems, logischer Reset (vgl. Kapitel 1.2)

- 9 Aktivieren bzw. Aufheben Cursor-Feststell-Funktion (Cursor-Steuerung statt Ziffern; bei SHIFT: Ziffern statt Cursor-Steuerung) (vgl. dazu auch PCOS-Befehl **ck**)
- 10 Aktivieren bzw. Aufheben Buchstaben-Feststell-Funktion (Groß- statt Kleinbuchstaben; bei SHIFT Klein- statt Großbuchstaben) (vgl. dazu auch PCOS-Befehl **ck**)

Zusammenhang zwischen **sl**, **ck**, **pk**, **rf**, **ed** und **wf**



- Ø) automatische Zuordnung nach Laden des Betriebssystems
- 1) Zuordnung von bestimmten Zeichen (8 x 10-Punkt-Matrizen) zu den Original-ISO-Codes
 - 2) Zuordnung eines neuen ISO-Codes bzw. einer Sonderfunktion wie **C** anstelle des Original-ISO-Codes bzw. der Original-Sonderfunktion. Bei Abfrage wird Sonderfunktions-Code ausgegeben - wenn vorhanden - anstelle des ISO-Zeichen-Codes
 - 3) Verweis auf andere Zeichen(folge), die beim ISO-Code statt des Original-ISO-Zeichens erzeugt wird.
 - 4) Abspeichern der 8 x 10-Punktmatrizen, die bei den einzelnen ISO-Codes erzeugt werden, in ein 'Tastaturfile'
 - 5) Anzeigen, Ändern oder Neuaufnahme von einzelnen im 'Tastaturfile' gespeicherten 8 x 10-Punkt-Matrizen in Abhängigkeit vom ISO-Code
 - 6) Aktivieren der im 'Tastaturfile' beschriebenen 8 x 10 Punkt-Matrizen oder der beim Laden des Betriebssystems gültigen Punkt-Matrizen

1.7 Erstellen von neuen Betriebssystem-Konfigurationen

Das Vorhandensein von zusätzlich residenten PCOS-Befehlen (vgl. Kapitel 1.4), geänderte Systemparameter, Tastaturbelegungen bzw. Bildschirmzeichensätze (vgl. Kapitel 1.6) haben grundsätzlich nur Gültigkeit bis zum nächsten logischen oder physischen Reset (Ausnahme: Echtzeituhr wird vom logischen Reset nicht betroffen). Nach dem Laden des Betriebssystems müßten normalerweise alle von der Original-Betriebssystem-Konfiguration abweichenden Bedingungen wiederhergestellt werden (z.B. durch erneute Anwendung der PCOS-Befehle **pl**, **sf**, **sc**, **ch**, **pk** etc.).

Um dies zu vermeiden, kann zu jedem Zeitpunkt während der Arbeit mit dem System vom Anwender entschieden werden, daß die gerade gültigen Bedingungen ("Systemkonfiguration") als neuer ladbarer Teil des Betriebssystems auf Diskette abgespeichert werden sollen. Dies geschieht mit Hilfe des PCOS-Befehls **ps**. Die mit **ps** gewonnene, auf Diskette abgespeicherte Betriebssystem-Konfiguration kann dann

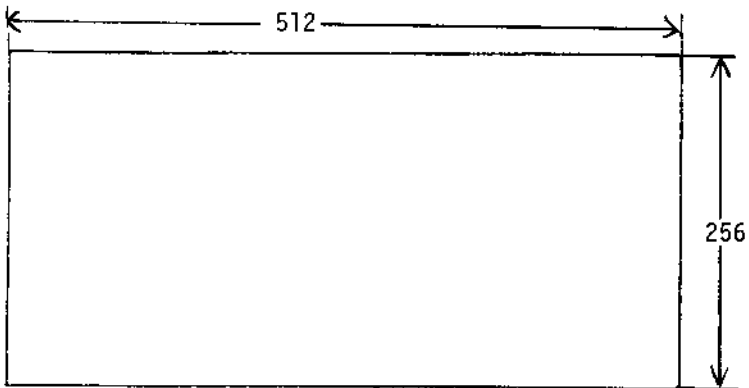
- a) zum Laden des Betriebssystems nach logischem oder physischem Reset anstelle der Original-Betriebssystem-Konfiguration verwendet werden;
- b) während der Arbeit mit dem System als neues zu ladendes Betriebssystem direkt aufgerufen werden (PCOS-Befehl **pr**), z.B., weil ein anderes Anwendungspaket zum Einsatz kommt, für das andere vorher per **ps** abgesicherte Parameter benötigt werden.

pr kann in der PCOS-Ebene oder (via **CALL** bzw. **EXEC**) in der BASIC-Ebene aufgerufen werden, ohne daß ein Reset nötig wird. Die betreffende Systemdiskette muß dazu natürlich einliegen. Das so geladene Systemfile braucht nicht an erster Stelle im Inhaltsverzeichnis zu stehen.

1.8 Bildschirmorganisation, Farbbildschirme

Punktraster und Bitmap

Der M20-Bildschirm ist physisch eingeteilt in ein Raster von 512 x 256 Elementarpunkten (= 131072 Elementarpunkte)



Sowohl die graphische als auch die alphanumerische Darstellung basiert auf dem Leuchten oder Nichtleuchten von einzelnen Elementarpunkten innerhalb der 512x256-Punkte-Matrix. Für die Darstellung des Bildschirminhaltes steht beim Schwarz-Weiß-Bildschirm ein Speicherbereich von 16 KB zur Verfügung. Für den Vier-Farbschirm werden 32 KB benötigt. Für den Acht-Farb-Schirm werden weitere 16 KB belegt. Für die 16 zusätzlichen KB des Vier-Farb-Schirms muß eine RAM-Erweiterung (32 oder 128 KB) vorhanden sein, für die weiteren 16 KB des Acht-Farb-Schirms noch eine Erweiterung. 32- und 128-KB-Erweiterungen können nicht vermischt eingesetzt werden.

Die zur Verwaltung der Elementarpunkte benötigten Speicherbereiche werden als Bitmap bezeichnet.

Beim Schwarz-Weiß-Schirm ist ein Bit innerhalb der Bitmap einem Elementarpunkt auf dem Bildschirm zugeordnet:

Bit	Darstellung auf dem Schirm
0	Hintergrundfarbe (leuchtet nicht)
1	Vordergrundfarbe (leuchtet)

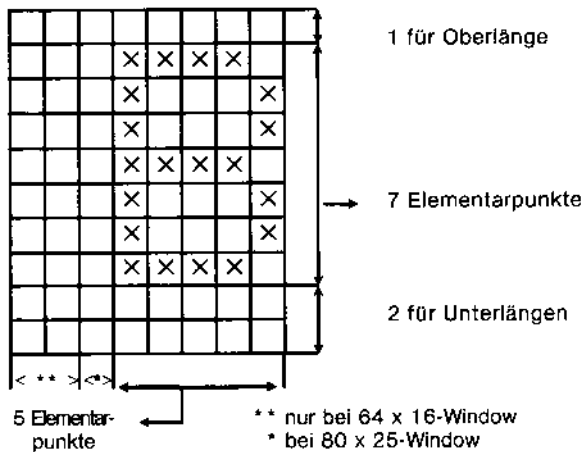
Beim Vierfarb-Schirm sind jeweils zwei Bit zur Darstellung eines Punktes in einer der vier Farben nötig (Kombination von zwei Bit-Maps), beim Achtfarb-Schirm werden für jede Farbe drei Bit erforderlich.

Zeichendarstellung

Die Zeichen sind in einer 8 x 10-Punktmatrix definiert, von der im allgemeinen nur 5 x 7 Punkte belegt werden.

Sonderfälle bilden z.B. Buchstaben mit Oberlänge wie g, q und j oder mit Unterlänge wie die Umlaute Ä, Ü und Ö.

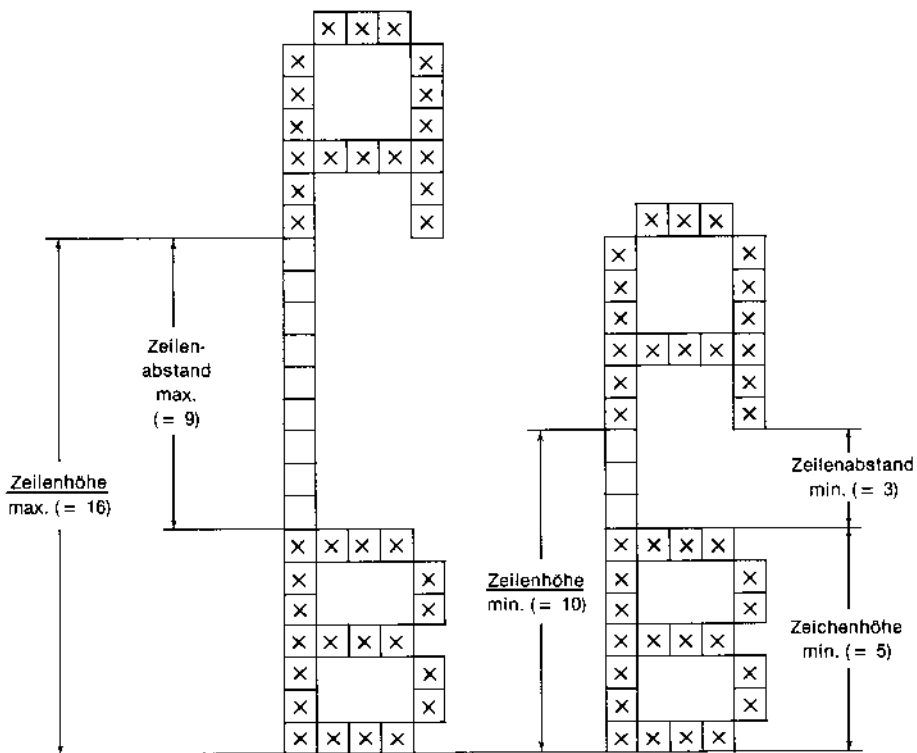
Beispiel: Buchstabe B



Bestimmung der Anzahl Zeilen im Bildschirm

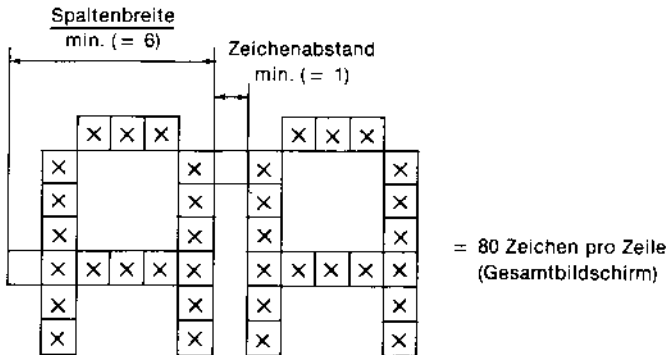
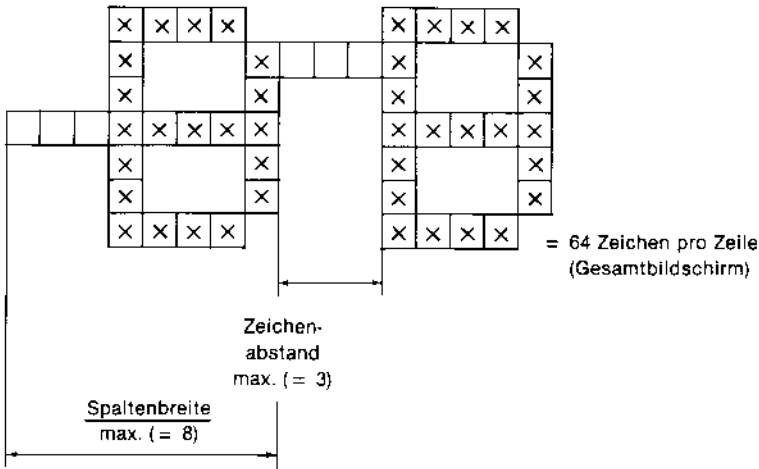
Die Anzahl Textzeilen im Bildschirm kann zwischen 16 und 25 variiert werden. Die Zeilenhöhe setzt sich zusammen aus der Höhe eines Zeichens (5 bis maximal 8 Elementarpunkte, Ausnahme: Ober-/Unterlänge) plus einem zusätzlich möglichen Zeilenabstand (Ø - 6 Elementarpunkte bei maximaler Zeichenhöhe von 8 Elementarpunkten).

Beispiel:

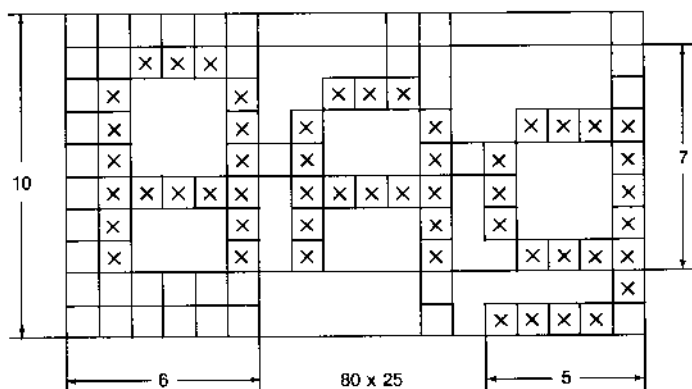
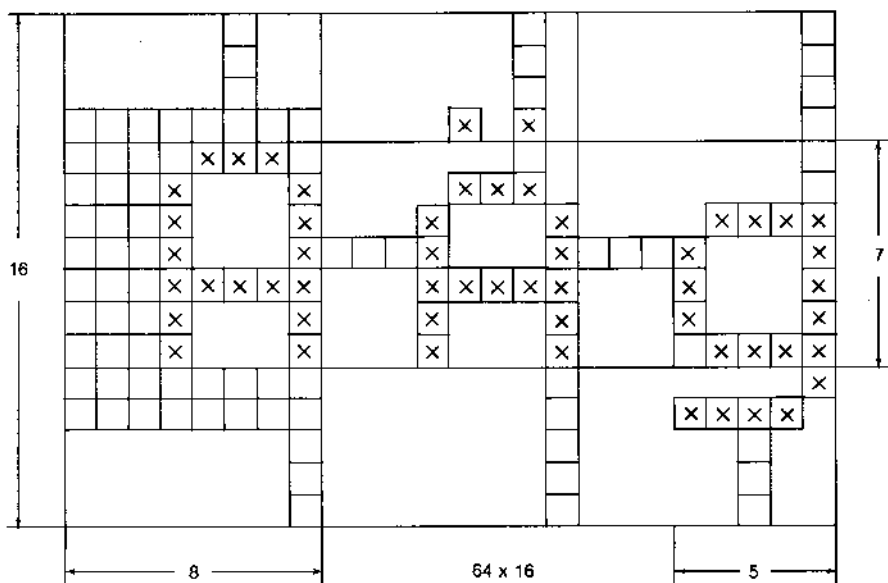


Bestimmung der Anzahl Zeichen pro Zeile

Die Anzahl Textzeichen pro Zeile ist festgelegt durch die Anzahl Elementarpunkte einer Textspalte. Dabei kann die Spaltenbreite entweder 6 Elementarpunkte (80-Zeichen-Window) oder 8 Elementarpunkte (64-Zeichen-Window) umfassen. Von diesen werden normalerweise nur die 5 am weitesten rechts liegenden zur Punktdarstellung verwendet; beim 80-Zeichen-Window werden die zwei Punktspalten ganz links außen nicht abgebildet.



Zusammenfassung der Zeichendarstellung bei 64 x 16- und 80 x 25-
Bildschirmformat



Arbeit mit Windows

Das Bildschirm-Handling des M20 zeichnet sich durch die Möglichkeit der Window-Technik aus. Per Definition ist der M20-Bildschirm als ein Window zu betrachten, aus dem bis zu 16 unabhängige rechteckige Windows durch Teilung gebildet werden können. Diese Windows können als "eigene Bildschirme" betrachtet werden, die unabhängig voneinander angesprochen werden können. Das Anlegen von Windows geschieht ausschließlich innerhalb der BASIC-Ebene mit der BASIC-Funktion **WINDOW**. Innerhalb der Windows kann all das gemacht werden, was auf dem Gesamtbildschirm möglich ist; d.h., es existiert in jedem Window ein Text- und ein Graphik-Cursor. Das Window kann einzeln aufgelöst werden (BASIC-Anweisung **CLOSE WINDOW**). Es ist möglich, eine Hard-Copy für ein einzelnes Window oder den Gesamtschirm zu erstellen (PCOS-Befehl **sp**), den Textinhalt eines Windows auszugeben (PCOS-Befehl **ls**), ein eigenes Scrolling durchzuführen, die Anzahl Zeilen bzw. Anzahl Zeichen pro Zeile, eine eigene Vorder- und Hintergrundfarbe sowie eine eigene Grafik-Skalierung für jedes Window festzulegen.

Anlegen von Windows

Nach dem Laden des Betriebssystems ist der gesamte Bildschirm als ein Window zu betrachten, das die Window-Nr. 1 hat.

Das Anlegen ("Eröffnen") eines neuen Windows erfolgt durch Teilung eines bestehenden Windows innerhalb der BASIC-Ebene. Dabei wird festgelegt, in welcher Form die Teilung erfolgt und wie groß das neue Window sein soll.

Das ursprüngliche Window bleibt mit seiner Nummer erhalten und wird in seiner Größe so reduziert, daß es den nach der Abspaltung des neuen Windows verbleibenden Rest umfaßt.

Auflösen von Windows

Windows können einzeln oder insgesamt wieder aufgelöst werden (BASIC-Anweisungen **CLOSE WINDOW** und **CLEAR**).

Aktivieren eines Windows

Jedes Window stellt einen logisch selbständigen Bildschirm dar. Da sinnvoll nur mit jeweils einem Bildschirm gearbeitet werden kann, muß jeweils eines der eröffneten Windows als aktives Window vereinbart werden (BASIC-Anweisung **WINDOW%**).

Alle PCOS-Befehle, die innerhalb der BASIC-Ebene aufgerufen werden und nicht einen eigenen Parameter 'Window-Nr.' haben, beziehen sich immer auf das aktive Window.

Die dabei geltenden Ausgabeformate sind durch die Grenzen des aktiven Windows bestimmt.

Die Inhalte der nicht aktiven Windows bleiben unverändert bestehen. Das System merkt sich die Größe (Format), die gültige Skalierung und die Position des Cursors in jedem einzelnen Window.

Wird ein Window aktiviert, können die zugeordneten Werte sofort weiterverwendet werden.

Die genaue Beschreibung der Regeln über das Anlegen, Auflösen und Aktivieren von Windows findet sich im Handbuch "BASIC 8000" bei den BASIC-Anweisungen **WINDOW%** und **CLOSE WINDOW** sowie der Funktion **WINDOW**.

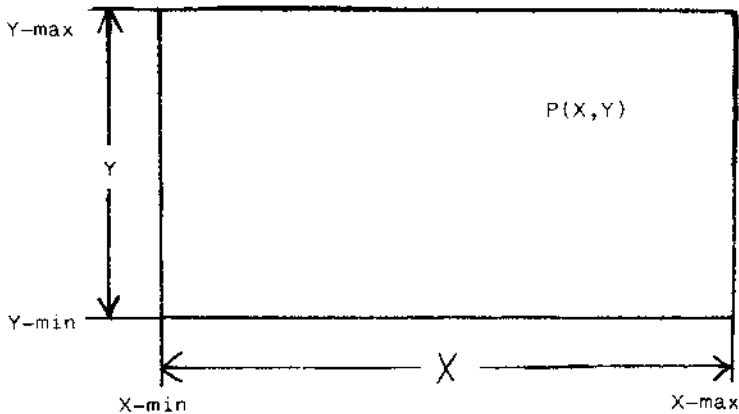
Mit Windows kann nicht innerhalb der PCOS-Ebene gearbeitet werden. Dort steht immer nur der gesamte Bildschirm unter der Window-Nr. 1 zur Verfügung.

Textausgabe in Windows

Im Window kann gleichzeitig sowohl alphanumerisch (Text) als auch graphisch gearbeitet werden. Beim alphanumerischen Arbeiten kann unabhängig von anderen Windows die Anzahl Zeichen pro Zeile (80 oder 64) und die Anzahl Zeilen (16 bis 25) (bezogen auf den Gesamtbildschirm) festgelegt werden. Dies erfolgt mit Hilfe der BASIC-Funktion **WINDOW** oder des PCOS-Befehls **ss**. Der Text-Cursor kann durch die BASIC-Anweisung **CURSOR** unter Angabe von Spalte und Zeile an beliebiger Stelle positioniert und dargestellt werden. Die Position (1,1) (d.h. Spalte 1, Zeile 1) befindet sich stets in der linken oberen Ecke des Windows. Die Form des Cursors kann frei definiert werden (besondere Formulierung der **CURSOR**-Anweisung).

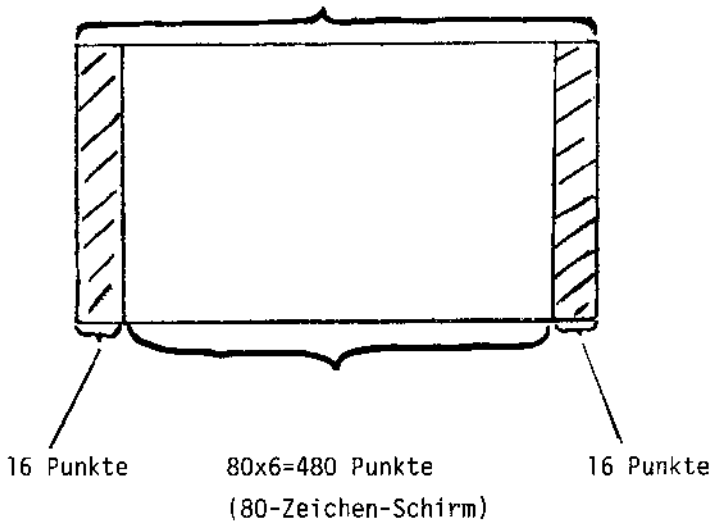
Graphisches Arbeiten in Windows

Für jedes Window kann ein anwenderbezogener Koordinatenbereich definiert werden. Dazu werden über die BASIC-Anweisung **SCALE** die zulässigen minimalen und maximalen X- und Y-Koordinaten festgelegt.



Falls nicht unter Anwenderskalierung gearbeitet wird, sind die Default-Werte für die Koordinaten innerhalb eines Windows:

X-min		Ø
Y-min		Ø
X-max		Anz. horizontaler Elementarpunkte - 1
Y-max		Anzahl vertikaler Elementarpunkte - 1



Skizze:

Matrix der Elementarpunkte für beide Bildschirm-Formate

Der graphische Cursor kann auf einen beliebigen Elementarpunkt innerhalb eines Windows positioniert werden. Die Position $(0,0)$ befindet sich in dem Elementarpunkt in der linken unteren Ecke, falls mit **SCALE** nicht eine andere Koordinateneinteilung definiert wurde.

Es ist möglich, jeden einzelnen Elementarpunkt des Windows über verschiedene Graphik-Anweisungen anzusprechen.

Verwendung von Farbe

Beim Schwarz-Weiß-Bildschirm stehen die Farben schwarz und weiß zur Darstellung der Elementarpunkte zur Verfügung. Dabei unterscheidet man in Hintergrund- und Vordergrund-Farbe. Die Hintergrund-Farbe ist die Farbe, welche der Bildschirm nach Laden des Betriebssystems annimmt. Der Default-Wert für die Hintergrund-farbe beim Schwarz-Weiß-Schirm ist schwarz.

Schwarz-Weiß-Bildschirm

Farbcode	Farbe
Ø	schwarz (Hintergrundfarbe)
1	weiß (Vordergrundfarbe)

Bei den Farbbildschirmen stehen für die Darstellung der Elementarpunkte acht Farben zur Verfügung. Es ist nun zwischen dem Vier-Farben- und dem Acht-Farben-Schirm zu unterscheiden. Farbauswahl und Änderung von Farben kann nicht in der PCOS-Ebene vorgenommen werden.

Aus den vorhandenen acht Farben können für den Vier-Farb-Schirm mit der BASIC-Anweisung **COLOR=** (globale **COLOR**-Anweisung) nur vier Farben ausgewählt werden, mit denen gleichzeitig gearbeitet werden kann. Am Acht-Farb-Schirm können alle acht Farben verwendet werden.

Den Farben ist stets ein Farbcode (Ø - 7) zugeordnet.

Farbbildschirm

Farbcode	Farbe
Ø	schwarz (Default-Hintergrundfarbe)
1	grün (Default-Vordergrundfarbe)
2	blau
3	cyan (türkis)
4	rot (Farbe des Cursors am Acht-Farb-Schirm)
5	gelb
6	magneta (purpur)
7	weiß (Default-Farbe des Cursors am Vier-Farb-Schirm)

Die Bestimmung der Hintergrund- und der Vordergrundfarbe kann für den Schwarz/Weiß- und den Acht-Farb-Schirm ausschließlich durch die BASIC-Anweisung **COLOR** erfolgen.

In der Anweisung **COLOR** ist für den Vier-Farb-Schirm nicht der Farbcode anzugeben, sondern die Position (Ø - 3) der Farben in der Anweisung **COLOR=** (von links abzählend). Diese Position wird in den Format-Beschreibungen mit 'Farbindex' bezeichnet. Für den Schwarz/Weiß- und den Acht-Farb-Schirm decken sich 'Farbcode' und 'Farbindex'.

1.9 besondere Graphik-Befehle

Da es innerhalb des BASIC-Graphik-Pakets keine Möglichkeit gibt, Text in verschiedener Größe und in verschiedener Richtung am Bildschirm abzubilden, gibt es hierfür den PCOS-Befehl `la` (sog. character plotting).

Die Ausgabe von über `la` erzeugten Zeichen(folgen) ist am Drucker nur möglich, wenn dieser graphikfähig ist (z.B. PR 1450, PR 2400).

Ausgaben von Graphik am Drucker sind mit dem PCOS-Befehl `sp` möglich. Dieser Befehl erzeugt eine Hardcopy des gesamten Bildschirminhalts (incl. Text) oder auch des Inhalts eines spezifizierten Windows an einem graphik fähigen Drucker.

Solche Hardcopies können nur aus dem Programm bzw. in der PCOS-Ebene oder im BASIC-Direkt-Mode abgerufen werden. Soll eine Hardcopy während eines BASIC-Programms auf Wunsch möglich sein, muß diese durch Tastaturinput und davon abhängige Verzweigung in `CALL "sp"` erfolgen.

Beispiel:

```
70 K$=INPUT$(1):CALL "sp"  
oder - ("free-running")  
80 IF INKEY$=CHR$(&H87) THEN CALL "sp" ELSE GOSUB 1010  
82 REM Hardcopy bei COMMAND+h-Druck  
90 GOTO 80  
1000 REM Hauptprogramm; Achtung: mögliche Kollision der Inputs  
mit Zeile 80 vermeiden  
  
1010 .  
.  
.  
9899 RETURN
```

Mit Hilfe des PCOS-Befehls **lt** kann der Textinhalt eines Windows am Drucker ausgegeben werden (auch an nicht graphikfähigen Druckern). Mit **la** erzeugte Zeichen(folgen) gelten jedoch als Graphik.

Mit Hilfe der PCOS-Befehle **rf**, **ed** und **wf** können die am Bildschirm auszugebenden Zeichen innerhalb der 8x10-Zeichenmatrix frei definiert werden, so daß bei Drücken einer Taste(nkombination) ein bestimmtes selbstdefiniertes graphisches Zeichen am Bildschirm erzeugt wird.

1.10 besondere Hilfen, Tools

Zum Laden des BASIC-Interpreters, der ab Release 2.0 nicht mehr automatisch resident ist, dient der PCOS-Befehl **ba**. Dieser bewirkt automatisch den Übergang in die BASIC-Ebene.

Mit Hilfe des Aufrufs des PCOS-Befehls **lt** in der BASIC-Ebene kann abgefragt werden, welche Eingabe-Abschlußtaste (CR, S1 oder S2) als letzte betätigt wurde. Diese Abfrage kann im Programm immer wieder nach Betätigen einer solchen Taste verwendet werden.

Um festzustellen, welche PCOS-Befehle in welcher Größe sich aktuell in welchem Teil des Arbeitsspeichers befinden, dient der PCOS-Befehl **dc**. Mit diesem Befehl kann auch die aktuelle Hardware-Konfiguration ausgegeben werden.

Mit dem Full-Screen-Editor (PCOS-Befehl **ed**) ist es möglich, BASIC-Files im ASCII-Format (ASCII-Files mit maximal 80 Zeichen langen Programmzeilen - incl. Zeilennr. -, oder sequentielle BASIC-Datenfiles) mit den Trennzeichen dezimal 13 und dezimal 10 am gesamten Bildschirm mit voller Cursor-Steuerung zu editieren.

Außerdem können mit Hilfe des PCOS-Befehls **ed** Assembler-Programme editiert werden, die anschließend noch zu assemblieren sind (vgl. hierzu Assembler-Handbücher, englisch). Solche Programme sind in Form von PCOS-Textfiles (Trennzeichen dezimal 30) zu erstellen.

PCOS-gesteuerte Prozeduren können ebenfalls komfortabel mit Hilfe des Full-Screen-Editors erstellt werden (vgl. Kapitel 2.5.3, Punkt 3.).

Der PCOS-Befehl **ep** dient dazu, neben den Fehlercodes von PCOS-Fehlermeldungen auch deren kurze englische Bedeutung ausgeben zu lassen (vgl. zur genauen Beschreibung dieser Meldungen Kapitel 4).

Mit dem PCOS-Befehl **kb** kann in BASIC ein kontrollierter Input (anstelle der Anweisungen **INPUT** bzw. **LINE INPUT**) programmiert werden.

Mit dem PCOS-Befehl **is** wird das Paket zur Verwaltung indexsequentieller File-Strukturen geladen.

Um die Schnittstelle zu diesem Paket zu gewährleisten, muß innerhalb aller BASIC-Programme mit indexsequentieller Dateiverwaltung das BASIC-Modul **isam.bas** zugeladen sein (zur genaueren Beschreibung von ISAM vgl. Handbuch ISAM - indexsequentielle Dateiverwaltung).

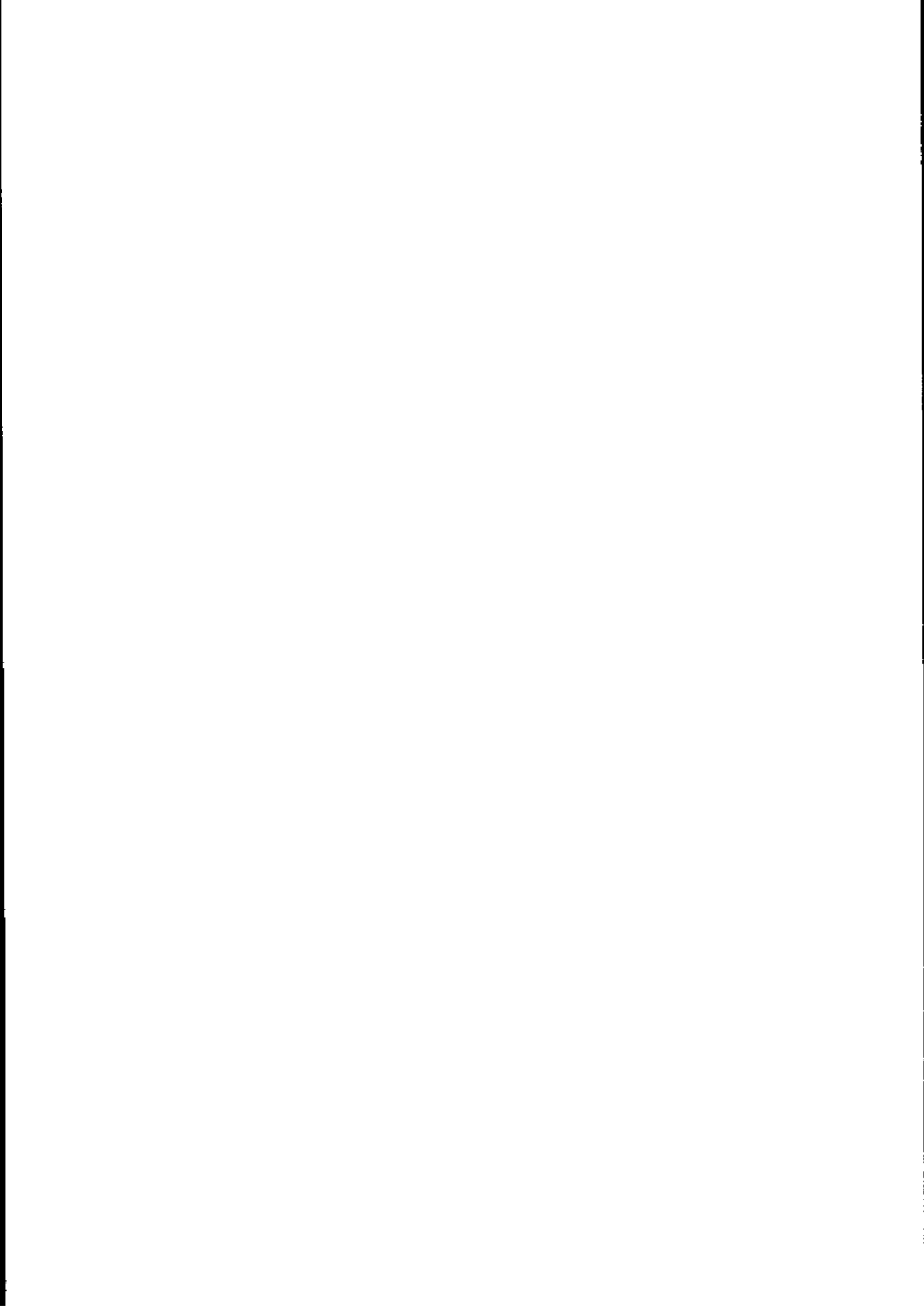
1.11 Verbindungen zu BASIC

Mit Hilfe der BASIC-Anweisungen **CALL** und **EXEC** (vgl. BASIC-Handbuch) können alle PCOS-Befehle in der BASIC-Ebene aufgerufen werden. Das im Arbeitsspeicher befindliche BASIC-Programm und die BASIC-Daten werden davon nicht betroffen (Assembler-Overlay).

Einige PCOS-Befehle (**ba**, **ie**, **rs**, **sc** und insbesondere **sb**) sollten jedoch grundsätzlich in der PCOS-Ebene aufgerufen werden. Notfalls können mit Hilfe einer Prozedur die nötigen Übergänge von Ebene zu Ebene erreicht werden (vgl. Kapitel 2.5.3, Teil 3.).

Es können auch selbsterstellte Routinen in Z8001-Assembler per **CALL** in BASIC aufgerufen werden. Dabei ist Parameterübergabe von BASIC zum Assembler möglich (sowohl **CALL** als auch **EXEC**), aber auch die Übergabe von Werten, die durch den PCOS-Befehl bzw. das Assembler-Overlay entwickelt wurden, an das BASIC-Programm (nur **CALL**, Beispiele: **lt**, **ck**).

Normalerweise bei Aufruf eines PCOS-Befehls erfolgende Ausgaben können mit Hilfe des Parameters **%n** unterdrückt werden. Dabei werden auch erwartete Tastaturinputs automatisch vorgegeben (vgl. Kapitel 2.5.1).



2.1 Formatbeschreibung

Für die Schlüsselwörter von PCOS-Befehlen gilt:

Groß- und Kleinbuchstaben werden vom M20 in der PCOS-Ebene unterschieden. Die Schlüsselwörter der PCOS-Befehle sowie einige wenige Parameter (z.B. **+dprt:**) können jedoch sowohl in Klein- als auch in Großschreibung angegeben werden.

Zur Angabe und Bestimmung eines PCOS-Befehls sind bereits die ersten beiden Buchstaben ausreichend. Groß- und Kleinbuchstaben werden dabei nicht unterschieden. Um PCOS-Befehle optisch sofort von BASIC-Sprachelementen zu unterscheiden, sind in allen Handbüchern nur die ersten beiden Buchstaben in Kleinbuchstaben angegeben.

Im Gegensatz zur BASIC-Ebene sind alle Parameter (z.B. Disketten-Spezifikationen) ohne " einzugeben. Als Texte im echten Sinne zu kennzeichnende Parameter müssen in ' ' eingeschlossen angegeben werden. Enthalten die Texte ', können sie stattdessen in " " eingeschlossen sein.

Um alle möglichen Fälle der Formulierung eines PCOS-Befehls möglichst knapp, aber umfassend darstellen zu können, werden bei der FORMAT-Beschreibung die folgenden Hilfssymbole und Darstellungsmittel verwendet:

{ , } , [,] , ... ,

Großschreibung, Kleinschreibung, Unterstreichung (___), **N** (Negativdarstellung) und Fettdruck.

Diese Hilfssymbole sind bei der Eingabe des PCOS-Befehls und seiner Parameter wegzulassen.

In einem PCOS-Befehl können maximal 256 Zeichen und bis zu 18 Parameter verwendet werden.

Die Parameter sind (wie in den Syntaxbeschreibungen im allgemeinen angegeben) durch Komma zu trennen. Stattdessen kann auch ein Blank zur Trennung eingesetzt werden.

Wird ein Parameter nicht angegeben, ist dafür ein Komma als Platzhalter zu setzen, sofern dahinter noch andere Parameter angegeben werden müssen. Für nichtangegebene Parameter gilt der Wert, der beim letzten davor erfolgten Aufruf des PCOS-Befehls bzw. nach Laden des Betriebssystems Gültigkeit hatte. (Beim PCOS-Befehle **sf** gilt eine hiervon abweichende Regelung).

Bedeutung der Symbole:

{ }

einer der in den Klammern untereinander angeführten Parameter muß verwendet werden (Alternativ-Auswahl)

[]

der/die Parameter in den Klammern kann verwendet werden

...

der/die Parameter in der zuletztgenannte/n [] kann beliebig oft erneut verwendet werden

Default-Wert = Wert wird vom System selbständig angesetzt, sofern der Parameter vom Anwender nicht eingegeben wird. Solche Default-Werte stehen immer in { } oder [].

SPRACHELEMENT
(GROBSCHREIBUNG)

feste Vorgabe als Sprachelement; muß genau so eingegeben werden.

FETTDRUCK

Schlüsselwörter und jeweils erforderliche Sonderzeichen werden **fett** gedruckt.

Begriff
(Kleinschreibung)

Platzhalter, ist durch einen aktuellen Parameter zu ersetzen. Dieser Parameter ist eine vom Anwender zu bestimmende Zeichenfolge, die dem 'Begriff' gerecht wird. Ist der 'Begriff' ein von PCOS auszugebender Text, ist er in ' einzuschließen.

X
(Negativdarst.)

Betätigen der Taste, die das in Negativdarstellung angegebene Zeichen (hier z.B. X) erzeugt, in Verbindung mit der blauen Taste (sog. CONTROL-Taste), aber ohne Eingabeabschlußtaste.

, od. ' od.
(od.)

und andere Sonderzeichen, die in der Syntaxbeschreibung nicht in [] stehen, müssen gesetzt werden, auch wenn ein folgender Ausdruck entfällt, aber noch etwas folgt (Platzhalterfunktion). Wenn gar nichts mehr folgt, muß das Zeichen entfallen. () müssen voll gesetzt werden, wenn nötig.

t

Blank; dieses Symbol wird nur an einigen Stellen ausdrücklich gesetzt, um die benötigte Anzahl Leertasten sofort anzuzeigen. Im allgemeinen ist diese jedoch durch Abzählen der an der entsprechenden Position der Folgezeile stehenden Buchstaben zu ermitteln.

ACHTUNG:

Die Eingabe von Sprachelementen ist nicht "formfrei". Das heißt, daß die Sprachelemente und Parameter nicht einfach hintereinander eingegeben und trennende Blanks dabei weggelassen werden können. Schlüsselwörter müssen durch Blanks von sonstigen Parametern getrennt sein.

BEISPIELE:

1) Die Eingabe

vf1:

führt zur Meldung

ERROR 90 (error in Parameter)

Es muß

vf 1:

eingegeben werden.

2) Die Erklärung

H löscht das zuletzt am Bildschirm abgebildete Zeichen

bedeutet:

Wird die Taste H in Verbindung mit der blauen Taste (CONTROL-Taste), aber ohne Eingabeabschlußtaste, gedrückt, wird das zuletzt am Bildschirm erzeugte Zeichen wieder gelöscht.

2.2 Bildung von Diskettenspezifikationen

Zur vollständigen Identifikation einer Disk(ette) ist anzugeben

- entweder: in welcher Diskettenstation die Diskette sich befindet
- oder: wie die Diskette heißt.

Am Ende dieser Diskettenspezifikation ist stets der : zu setzen (kein Blank davor).

Die gesamte Diskettenspezifikation hat folgendes FORMAT:

$\left. \begin{array}{l} \{ \text{Stationsbez.} \} \\ \{ \text{Diskettenbez.} \} \end{array} \right\} \left[\text{/Disketten-Password} \right] :$

Disketten-Identifizier

Disketten-Spezifikation

Die Diskettenspezifikation

kann also enthalten:

- Die Bezeichnung, die einer Disk(ette) gegeben wurde ("Name der Diskette")
oder
- Die Bezeichnung der Station, in der die Diskette liegt.

Es kann wahlweise die Bezeichnung der Disk(ette) oder die Bezeichnung der Station (0, 10 oder 1) angegeben werden.

Die Diskettenspezifikation ist stets durch : abzuschließen. Der : darf nicht durch Blanks vom Rest getrennt sein. Zusätzlich zur Bezeichnung kann einer Diskette ein Password zugeordnet werden, ohne dessen Kenntnis ein Zugriff auf die Diskette nicht möglich ist. Das Password ist durch / mit dem Disketten-Identifizier zu verbinden. Zwischen 'Disketten-Identifizier' und '/Disketten-Password' darf kein Blank stehen.

Anmerkung:

Aus Vereinfachungsgründen wird im gesamten Handbuch statt des in der EDV gebräuchlichen Begriffs "Volume" stets der Begriff "Diskette" verwendet. Damit kann also auch die Hard-Disk bei HDU-Version gemeint sein. Besonderheiten bei Verwendung der HDU finden Sie in Kapitel 0.2.7.

Bestandteile der Diskettenspezifikation sind im einzelnen:

Stationsbez.: { 0 (rechte Station) bei FDU-Version
1 (linke Station)
oder
10 (HDU) bei HDU-Version
0 (linke Station)

oder

Diskettenbez.: eine Folge von 1 bis zu 14 Zeichen; ausgenommen sind alle in Kapitel 2.4 aufgeführten unzulässigen Zeichen

Disketten-Password: eine Folge von 1 bis zu 14 druckbaren Zeichen; (optional) ausgenommen sind alle in Kapitel 2.4 aufgeführten unzulässigen Zeichen

Beispiele:

Die folgenden Diskettenspezifikationen sind in der PCOS-Ebene zulässig:

- a) 0/DKENNW:
- b) DISKNAME:
- c) 1:
- d) DISK1/KWDISK1:

Bemerkungen:

- Groß- und Kleinbuchstaben sind bei Diskettenbezeichnungen zu unterscheiden!
- Wird keine 'Diskettenspezifikation' angegeben, wird i.a. nur auf der aktuellen Station gesucht. Das ist die zuletzt angesprochene Station. Wird die 'Diskettenbezeichnung' angegeben, wird die Suche nach der Diskette stets in der Station begonnen, auf der zuletzt zugegriffen wurde und ggf. auf der anderen fortgesetzt.
- Sind Disketten-Passwords vergeben worden, sind diese stets anzugeben.
- Disketten-Passwords können mit dem PCOS-Befehl **vp** (Aufhebung durch **vd**) vergeben werden. Ist ein Disketten-Password einmal angegeben worden, kann bis zum nächsten Laden des Systems die Angabe unterbleiben. Dies gilt nicht mehr, wenn ein Diskettenwechsel erfolgt ist.
- Disketten können mit Schreibschutz versehen werden. Dies geschieht durch Überkleben der Ausstanzung am oberen rechten Rand der Diskette mit einem Stück Aluminiumfolie.

2. Filespezifikation:

Sie enthält die Bezeichnung des Files und ggf. eine Angabe über den Typ des Files. Enthält eine Filespezifikation eine Typangabe, muß diese stets als Bestandteil angegeben werden. Zusätzlich zu Filebezeichnung [Typ] kann einem File ein Password zugeordnet werden, ohne dessen Kenntnis ein Zugriff auf das File nicht möglich ist. Das File-Password ist mit / sofort an den File-Identifizier anzuschließen.

Bestandteile der Filespezifikation sind im einzelnen:

Filebez.: eine Folge von 1 bis zu 14 druckbaren Zeichen; ausgenommen sind alle im Kapitel 2.4 aufgeführten unzulässigen Zeichen

Typ:
(optional) eine Folge von mindestens einem druckbaren Zeichen; die maximale Anzahl Zeichen ergibt sich aus 13 minus Länge der Filebezeichnung; ausgenommen sind alle in Kapitel 2.4 aufgeführten unzulässigen Zeichen. Einige Typ-Zeichenfolgen haben für PCOS spezielle Bedeutung und zwar: cmd, sav, SAV, BAS und bas

File-Password:
(optional) eine Folge von 1 bis zu 14 druckbaren Zeichen; ausgenommen sind alle im Kapitel 2.4 aufgeführten unzulässigen Zeichen

Beispiele:

Die folgenden Filenamen sind in der PCOS-Ebene zulässig:

- a) Ø/DKENNW:ANWPROG.BAS/FPW
- b) DISKNAME:DAFILE.DAT
- c) 1:ASCFILE
- d) 1:MODUL1.ASC/FILEPASS
- e) KUNDEN

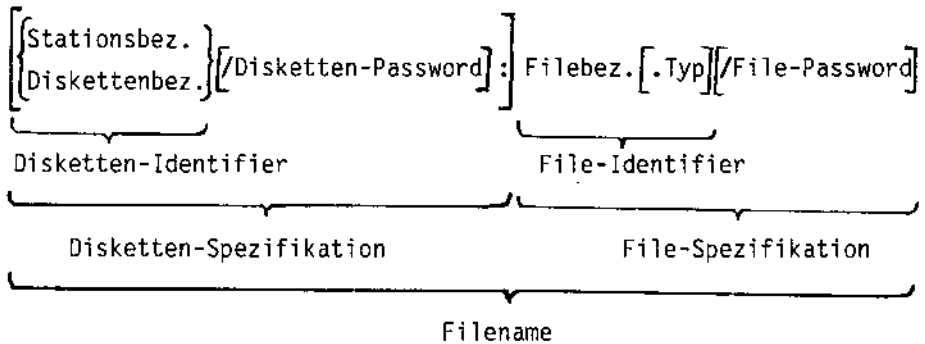
2.3 Bildung von Filenamen

Zur vollständigen Identifikation eines Files ist anzugeben

- auf welcher Diskette oder Diskettenstation das File sich befindet
- welches File dieser Diskette gemeint ist.

Ein Filename besteht somit aus der Diskettenspezifikation und der Filespezifikation.

Der gesamte Filename hat folgendes FORMAT:



1. die Diskettenspezifikation

Sie kann sein:

- Die Bezeichnung, die einer Diskette gegeben wurde ("Name der Diskette")
- Die Bezeichnung der Station, in der die Disk(ette) liegt.

Vgl. zur genaueren Beschreibung Kapitel 2.2

Die Disketten-Spezifikation ist stets durch : abzuschließen. Die File-Spezifikation muß daran sofort angeschlossen werden (Kein Blank!).

2.4 Besonderheiten bei Filenamen

- Groß- und Kleinbuchstaben sind bei Filenamen zu unterscheiden!
- Enthält ein File-Identifizier eine Typangabe (durch Anhängen von .Typ), so ist diese Erweiterung fester Bestandteil und muß beim Aufruf des Filenamens unbedingt angegeben werden.
- Sind Disketten- oder File-Passworts vergeben worden, sind diese stets anzugeben.
- Disketten-Passworts können mit dem PCOS-Befehl **vp** (Aufhebung durch **vd**) vergeben werden; File-Passworts können mit dem PCOS-Befehl **fp** (Aufhebung durch **fd**) vergeben werden.
- Wird beim 'Filenamen' die 'Disketten-Spezifikation' weglassen, wird das File ausschließlich auf der aktuellen Station gesucht. Das ist die zuletzt angesprochene Station. Wird hingegen die 'Diskettenbezeichnung' angegeben, wird die Suche nach der betreffenden Diskette auf der Station begonnen, auf der zuletzt zugegriffen wurde, und gegebenenfalls anschließend auf der anderen fortgesetzt.
Das letztere gilt auch für den Aufruf von PCOS-Befehlen.
- In der PCOS-Ebene ist im Bildschirm vor **>** stets die Station angegeben, auf der zuletzt zugegriffen wurde ("aktuelle Station"). Diese kann mit dem PCOS-Befehl **se** geändert werden.
- Sowohl Disketten als auch Files können mit Schreibschutz versehen werden. Ersteres geschieht durch Überkleben der Ausstanzung am oberen rechten Rand der Diskette mit einem Stück Aluminiumfolie. Der Schreibschutz von Files wird unter Verwendung des PCOS-Befehls **fw** (Aufhebung: **fu**) erreicht.
- Es empfiehlt sich, keine File-Identifizier (Filename.Type) zu verwenden, die länger als 10 Zeichen (incl. Punkt) sind (Maximum: 14), wenn das File mit dem Full-Screen-Editor bearbeitet werden soll.

Der Full-Screen-Editor legt beim Aufruf (**ed**) unter Verwendung des Parameters **%b** eine Sicherungskopie des zu editierenden Files an. Dessen Name ergibt sich durch Anhängen von **.bak** an den Namen des zu editierenden Files.

- Besondere Typangaben:

.cmd

Es handelt sich um einen PCOS-Befehl, der von Diskette aufgerufen und abgearbeitet wird, sofern er nicht mit dem PCOS-Befehl **pl** resident gemacht wurde oder in einer mit **ps** erstellten Betriebssystem-Konfiguration enthalten ist. Nach Abarbeitung verbleibt der PCOS-Befehl nicht im Arbeitsspeicher, es sei denn, er ist resident aufgrund von **pl** bzw. **ps**.

.sav

Es handelt sich um einen PCOS-Befehl, der von Diskette aufgerufen und abgearbeitet wird, sofern er nicht mit **pl** resident gemacht wurde bzw. in einer mit **ps** erstellten Betriebssystem-Konfiguration enthalten ist und er seit Laden des Betriebssystems noch nicht abgerufen wurde. Nach Abarbeitung verbleibt der PCOS-Befehl im Arbeitsspeicher. Ein PCOS-Befehl mit dem Typkennzeichen **.sav** wird also mit dem ersten Aufruf automatisch resident gemacht.

.bas

Es handelt sich um ein BASIC-Programm, das direkt von PCOS her aufgerufen werden kann, ohne daß der PCOS-Befehl **ba** dazu verwendet werden muß. Die Eingabe der ersten beiden Buchstaben des Programms in der PCOS-Ebene reicht bereits aus, um das Programm aufzurufen. Wird das Programm gefunden, wird die BASIC-Ebene angesteuert und das Programm sofort zur Abarbeitung gebracht, sofern der BASIC-Interpreter (**ba**) gefunden wird.

- INIT.BAS (Selbststartprogramm)

Nach dem Laden des Betriebssystems sucht dieses automatisch beide Diskettenstationen daraufhin ab, ob ein File namens INIT.BAS (oder auch init.bas) vorhanden ist (erst auf der zuletzt angesprochenen Station, dann auf der anderen). Wenn ja, wird nicht - wie üblich - die PCOS-Ebene aufgerufen, sondern der BASIC-Interpreter wird geladen, die BASIC-Ebene aufgesucht und INIT.BAS automatisch geladen und gestartet.

Es ist hierzu Voraussetzung, daß der PCOS-Befehl **ba**, das File basic.abs und das File INIT.BAS auf der gleichen Diskette liegen. Ist der PCOS-Befehl **ba** durch ein **ps** zum Bestandteil des Systemfiles gemacht worden (also dauerhaft resident), braucht er nicht auf Diskette zu liegen, da er dann bei der Suche nach INIT.BAS nicht mehr ausgeführt werden muß. INIT.BAS ist ein normales BASIC-Programm, das vom Programmierer erstellt werden kann. Der Aufruf von INIT.BAS kann unterdrückt werden, wenn man während des Ladens des Betriebssystems die **s**- oder die **b**-Taste gedrückt hält (ohne CR-Taste).

s bewirkt den Aufruf der PCOS-Ebene, auch wenn ein INIT.BAS vorhanden ist.

b bewirkt den Aufruf der BASIC-Ebene, ohne ein evtl. vorhandenes INIT.BAS zu aktivieren.

- Unzulässige Zeichen

: Doppelpunkt (außer bei Disketten-Identifizier)	? Fragezeichen (außer bei File-namenauswahl)
. Punkt (außer bei Typangabe)	! Ausrufezeichen
/ Slash (außer bei Passwords)	§ Paragraph-Zeichen
, Komma	= Gleichheitszeichen
+ Plus	ö
* Sternchen (außer bei File-namenwahl)	£ Pfundzeichen
" Anführungszeichen	␣ Blank
- Bindestrich oder Minus	; Strichpunkt
# Nummernzeichen	\$ Dollarzeichen
& Ampersand	@ at-Zeichen
> größer-als-Zeichen	\ Backslash
	' Hochkomma

2.5 Besondere Syntaxelemente

2.5.1 Unterdrückung von zusätzlichen Ausgaben; automatische Eingaben

Beim Aufruf von PCOS-Befehlen (auch in der BASIC-Ebene) werden normalerweise (default-mäßig am Bildschirm) verschiedene zusätzliche Informationen ausgegeben, z.B., wie der PCOS-Befehl heißt, welches Release verwendet wird, welche Segmente nach Adressen des Arbeitsspeichers betroffen sind, wieviele Bytes betroffen sind. Darüber hinaus werden Bedienerführungen in Englisch ausgegeben sowie Informationen über den Ablauf.

Diese Ausgaben können durch Angabe des Parameters `%n` direkt hinter dem Schlüsselwort des PCOS-Befehles und vor der Parameterliste unterdrückt werden. Dabei werden auch alle sonst benötigten Eingaben automatisch von PCOS vorgegeben. Vor und hinter `%n` muß ein Blank stehen. `%n` sollte besser durch ein Komma von dem, was folgt, abgegrenzt werden. Dies ist zwingend, falls hinter `%n` sofort ein Komma als Platzhalter für einen nicht angegebenen Parameter folgt.

Beispiele:

v1 %n Ø: gibt das Inhaltsverzeichnis der Diskette in Station Ø aus, ohne daß nach einer gefüllten Bildschirmseite jeweils eine Taste zur Fortsetzung gedrückt werden muß.

sf %n,,Ø

setzt den dritten Parameter bei sf auf Ø (die ersten beiden werden mit ihrem bisherigen Wert übernommen) und unterdrückt alle Ausgaben.

vc %n Ø: 1:

Das Kopieren der Diskette in Station Ø auf die Diskette in Station 1 geschieht ohne Ausgaben am Bildschirm. Die Eingabe **y** auf die normalerweise erscheinende Frage **"Warning-vcopy deletes all files. Copy disk? (y/n)"** wird von PCOS automatisch vorgenommen.

2.5.2 Filenamenauswahl

Um eine Gruppe von Files mit gemeinsamen Eigenschaften zu definieren, bietet sich an, diese anhand ihrer Namen zu gliedern.

- Beispiele:
1. alle BASIC-Programm-Files enden mit der Zeichenfolge .prg (Typangabe)
 2. alle Files, die ein besonderes Fachgebiet betreffen, haben an 2. und 3. Position die zweistellige Nr. des Fachgebietes

Für die File-orientierten PCOS-Befehle - mit Ausnahme von **fn**, **fm**, **fr** und **rk** - sowie die PCOS-Befehle **vl** und **vq** gibt es die Möglichkeit, jeweils eine Gruppe von Files mit gleichen Eigenschaften, die aus dem File-Identifizier hervorgehen, auszusuchen. Dazu dienen die besonderen PCOS-Sprachelemente ***** und **?**.

***** am Anfang oder Ende gibt an, daß dort jeweils beliebige Zeichen stehen können, und zwar beliebig viele.

? ...(kann wiederholt werden) ab dem Anfang geben an, daß genau so viele Zeichen stehen können, wie durch **?** angegeben.

Beispiele: Die folgenden Files seien auf der Diskette in Station 0 vorhanden:

- A12Konten.rnd
- B12Konten.prg
- B13Namen.prg
- C12Namen.rnd

Mit **fc 0:* .rnd 1:** werden die Files A12Konten.rnd und C12Namen.rnd auf die Diskette in Station 1 übertragen.

Mit **vl 0:B*** werden die Informationen über die Files B12Konten.prg und B13Namen.prg ausgegeben.

Mit **fp 0:?12*,Abt12** werden die Files B12Konten.prg, C12Namen.rnd und A12Konten.rnd alle mit dem File-Password 'Abt12' versehen.

Mit **ff 0:???Konten*** werden die Files B12Konten.prg und A12Konten.rnd auf die logische Länge reduziert.

2.5.3 Wechsel des Ein/Ausgabe-Mediums

Die Eingaben von PCOS-Befehlen und ihren Parametern werden innerhalb der PCOS-Ebene (bzw. bei Anwendung der BASIC-Anweisungen CALL oder EXEC in der BASIC-Ebene) über Tastatur erwartet, wenn nicht ausdrücklich etwas anderes vereinbart wurde.

Die Ausgaben der Informationen zum PCOS-Befehl (vgl. 2.5.1) erfolgen über den Bildschirm, wenn nicht ausdrücklich etwas anderes vereinbart wurde.

Es kann festgelegt werden, daß diese Eingaben bzw. Ausgaben zusätzlich oder nur über andere Einheiten und/oder sequentielle Datenfiles erfolgen sollen.

1. lokaler Wechsel

Bei einem lokalen Wechsel gilt die abweichende Vereinbarung nur für den PCOS-Befehl, in dem sie erfolgt. (Ein lokaler Wechsel ist nur in der PCOS-Ebene möglich.)

Zum Beispiel kann das Inhaltsverzeichnis über den Bildschirm und zusätzlich auf dem Drucker ausgegeben und in ein sequentielles Datenfile aufgezeichnet werden.

Die für einen bestimmten PCOS-Befehl benötigten Eingaben könnten z.B. statt über Tastatur aus einem File oder aus einer über V.24 angeschlossenen anderen Tastatur vorgenommen werden.

Die Einheiten haben defaultmäßig folgende Namen:

prt: Drucker
cons: Bildschirm und Tastatur
ieee: IEEE-488-Driver
com: Standard-RS-232-(V.24-)Driver
com1: erster RS-232-Driver auf Erweiterungsplatine
(TWIN-Board)
com2: zweiter RS-232-Driver auf Erweiterungsplatine
(TWIN-Board)

(Diese Namen können evtl. mit dem PCOS-Befehl **sd** geändert werden.)

Namen von Einheiten müssen stets mit dem : abgeschlossen sein!

Namen von Files enden nie auf :.

Verwendete Ein- bzw. Ausgabefiles müssen sequentiell von dem Typ sein, der in BASIC 8000 verwendet wird. Als Eingabefiles können sie somit über **OPEN"0"**, ... erstellt werden, wobei jeder Befehl (Prozedurzeile: gewünschter PCOS-Befehl, BASIC-Befehl bzw. -Direkt-Anweisung bzw. -Programmzeile, je nach Systemebene und BASIC-Mode) über eine **PRINT#**-Anweisung auf Diskette geschrieben wird.

Solche Datenfiles können komfortabler über den Full-Screen-Editor erstellt werden (PCOS-Befehl **ed**). Dabei muß aber der Parameter **%c** gesetzt werden.

Soll für einen PCOS-Befehl eine andere Eingabeeinheit bzw. ein File zusätzlich zum Zwecke der Eingabe angesprochen werden, so ist direkt vor deren Namen **+s** (ohne Blank) nach dem ersten Schlüsselwort (bzw. dem Parameter **%n**) zu setzen.

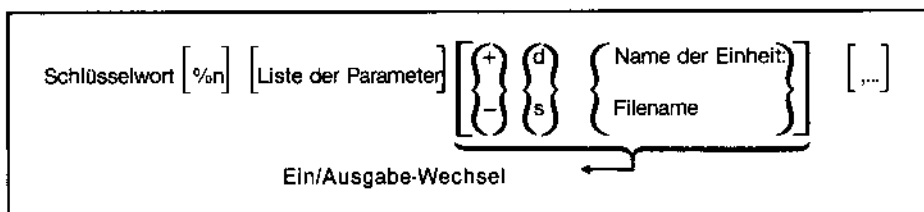
Soll eine Eingabeeinheit bzw. ein Eingabefile abgeschaltet werden, ist **-s** vor dem Namen anzugeben.

Soll eine andere Ausgabeeinheit bzw. ein File zusätzlich zum Zwecke der Ausgabe angesprochen werden, ist vor dem Namen **+d** anzugeben.

Zum Abschalten ist **-d** vor dem Namen anzugeben.

(d) steht für "destination"; s steht für "source")

Es ergibt sich folgendes FORMAT:



Beispiele:

1. Das Inhaltsverzeichnis soll zusätzlich auf dem Drucker und in ein sequentielles Datenfile namens 'INHALT' auf der Diskette in Station 0 ausgegeben werden:

v1 %n +dprt: ,+d0:INHALT

2. Das File 'INHALT' soll auf dem Drucker allein gelistet werden:

f1 0:INHALT +dprt: , -dcons:

Da nur ein Ausgabefile aktiv sein darf, reicht zum Abschalten einer Ausgabe auf ein File auch nur **-d**.

2. Globaler Wechsel

Die Ein/Ausgabeeinheiten bzw. -Files können auch für alles Folgende definiert werden. Dazu wird der Parameter 'Ein/Ausgabe-Wechsel' im obigen Format allein angegeben.

Beispiele: **+scom: ,+dcom:**

Alle folgenden Eingaben und Ausgaben werden über die Standard-RS-232-Schnittstelle abgewickelt. Tastatur und Bildschirm sind zusätzlich aktiv.

+sØ:PROZEDUR.pro,+dprt: , -scons: , -dcons:

Alle Eingaben werden aus dem sequentiellen Datenfile 'PROZEDUR.pro' auf der Diskette in Station Ø entnommen, alle Ausgaben erfolgen auf dem Drucker, die Eingabeeinheit Tastatur und die Ausgabeeinheit Bildschirm sind abgeschaltet.

Im letzten Falle kann die Tastatur nur durch Neuladen des Betriebssystems wieder aktiviert werden, es sei denn, der PCOS-Befehl **+scons:** wird aus dem File 'PROZEDUR.pro' gegeben.

Wird der Bildschirm über **-dcons:** als Ausgabeeinheit gesperrt, gilt dies nicht für Ausgaben, die über BASIC-Anweisungen für Graphik, wie z.B. **LINE**, **PSET**, **CIRCLE**, **DRAW** oder **PUT%**, erfolgen.

3. Prozeduren

Prozeduren sind in sequentiellen Datenfiles festgehaltene Folgen aus PCOS-Befehlen und/oder BASIC-Befehlen, -Direkt-Anweisungen und Programmzeilen, die direkt von PCOS gesteuert werden, indem das betreffende Datenfile mit +sFilename aufgerufen wurde.

Erfolgt dabei **ba**, wird automatisch die Kontrolle (kurzfristig) an den BASIC-Interpreter übergeben; mit **SYSTEM** kann wieder in die PCOS-Ebene verzweigt werden. Innerhalb der BASIC-Ebene hat der BASIC-Interpreter zwar Kontrolle; der Ablauf der Prozedur wird aber weiterhin von PCOS gesteuert. Es ist zu beachten, welche Ebene bzw. welcher BASIC-Mode gerade aktiv ist und welche Ein/Ausgabe-Einheiten (Bildschirm, Tastatur) zu bestimmten Zeitpunkten parallel zum Prozedurfile aktiv sein sollen.

Beispiele für sequentielle Datenfiles als Prozedur:

```
+scons: +dcns:
sb 3,,1000
ba
5 CLEAR:EXEC "+dcns:":Tastatur bleibt deaktiviert
10 ?"ANFANG":Eingaben können vorläufig nur aus dem
30 Prozedurfile gegeben werden. Jede Zeile nach dem
30 Ende dieses Programms würde von (LINE) INPUT-
40 Anweisungen bzw. INPUT*-Funktionen (nicht INKEY*-
50 Funktionen!) im Programm als Eingabezeile mit Ab-
60 schluß durch CR verwendet! (z.B. für Testläufe)
9990 ?"ENDE"-EXEC "+dcns: +scons: -s1-PROZ.org"
9991 Tastatur und Bildschirm frei, Prozedur beendet.
9998 END: System im BASIC-Command-Mode.
9999 Prozedurfile PROZ.org noch offen!
SAVE "1:PROG.prg":SYSTEM
sb 5,,256
ba 1:PROG.prg
```

Dieses File hat den Namen 1:PROZ.pro und kann aus der PCOS-Ebene aufgerufen werden durch +s1:PROZ.pro oder aus der BASIC-Ebene durch EXEC "+s1:PROZ.pro".

Um zu richtigen Zeitpunkten Inputs über Tastatur bzw. Outputs über Bildschirm zuzulassen, könnte eine Prozedur der folgenden Form aufgebaut werden (Name des sequentiellen Datenfiles: 1:PROZ2.pro):

```
-scons: -dcons:
ba 1:EINGABE.prg
REM Ende der Prozedur 1:PROZ2.pro

0 REM Programm EINGABE.prg auf Diskette in 1:
10 exec "+dcons:;+scons:;-s1:PROZ2.pro";'läßt
20 'Schirmausgaben und Tastatureingaben zu, die
30 'nicht mehr durcheinander mit solchen aus dem
40 'Prozedurfile erfolgen. CLOSE für PROZ2.pro
50 'geht nicht, da PCOS die Prozedur kontrolliert
70 'und nicht der BASIC-Interpreter!
90 LINE INPUT "IHRE EINGABE ";T#
990 EXEC "-scons:;-dcons:;+s1:PROZ3.pro"
995 REM weiter aus Prozedurfile 1:PROZ3.pro
```

Über +sFilename oder +dFilename von PCOS direkt geöffnete Files können nur über -sFilename bzw. -dFilename bzw. -d wieder geschlossen werden (sonst: Error 55).

Es kann nicht mehr als ein Eingabe- und ein Ausgabe-File über Wechsel des Ein/Ausgabemediums aktiv sein.

Der Aufruf von PCOS-Befehlen, die einen Wechsel des Ein/Ausgabe-Mediums bewirken, ist in der BASIC-Ebene nur mit der BASIC-Anweisung EXEC möglich, nicht aber mit CALL.

Ein lokaler Wechsel innerhalb der BASIC-Ebene ist nicht möglich. Jeder Wechsel in einer EXEC-Anweisung gilt bis zur nächsten EXEC-Anweisung, die den alten Wechsel ändert oder bis zum Verlassen der BASIC-Ebene.

FUNKTION:

Laden des BASIC-Interpreters und Übergang von der PCOS-Ebene zur BASIC-Ebene

FORMAT:

```
ba [Filename]
```

WIRKUNG:

Die PCOS-Ebene wird verlassen und der BASIC-Interpreter geladen und aktiviert. Es wird eine Information über den aktuellen Systemzustand gegeben. Ist 'Filename' angegeben, wird danach automatisch das betreffende BASIC-Programm-File auf Diskette gesucht, in den Arbeitsspeicher geladen und zum Laufen gebracht.

BEMERKUNGEN:

- Die logische Größe des Arbeitsspeichers, die dem BASIC-Interpreter zur Verfügung gestellt wird, ist abhängig vom im PCOS-Befehl **sb** gesetzten Parameter 'log. Speichergröße'.
- Der Parameter **%n** kann nicht gesetzt werden. Die Information über den Übergang in die BASIC-Ebene kann jedoch durch **-dcons:** unterdrückt werden. Anschließend kann der Bildschirm in der BASIC-Ebene mit **EXEC "+dcons:*** wieder aktiviert werden.
- Auf der Diskette, auf der der PCOS-Befehl **ba** liegt, muß auch das zugehörige File **basic.abs** vorhanden sein.
- Nach Übergang in die PCOS-Ebene (BASIC-Befehl **SYSTEM**) verbleibt der BASIC-Interpreter nicht resident.

BEISPIELE:

ba +dprt:

Der BASIC-Interpreter wird aktiviert und alles folgende auf dem Drucker dokumentiert.

ba 1:Vorlauf.prg

Das Programm "Vorlauf.prg." auf Station 1 wird aus der PCOS-Ebene heraus aufgerufen.

(brief volume)

FUNKTION:

Abfrage von Filename(n) oder der Diskettenbezeichnung oder der Anzahl freier Sektoren auf einer bestimmten Diskette

FORMAT: (nur BASIC)

```
CALL "bv"(Funktionsname { Stationsnr. } { Diskettenbez. } [Parameterliste] )
```

Funktionsname: Stringausdruck; mögliche Werte: "search" oder "getvolname" oder "diskfree"

Stationsnr.: Integer-Ausdruck; mögliche Werte: -1, 0, 1 oder 10 (-1 für die letzte angesprochene Station); zu setzen bei "search" und "diskfree"

Diskettenbez.: Stringvariable; zu setzen bei "getvolname";

WIRKUNG: bei "search":

Je nach 'Funktionsname' wird ermittelt: welche Files mit bestimmten Eigenschaften am Namen auf der durch 'Stationsnr.' bezeichneten Diskette vorhanden sind.

FORMAT der Parameterliste:

Suchbeginn, Länge, §Ergebnisstring, File-Identifizier

Suchbeginn: Integer-Ausdruck; mögliche Werte:

1: Beginn der Suche beim ersten File des Inhaltsverzeichnisses

Ø: Suche ab der Position, die der beim letzten **by**-Aufruf folgt

Länge: Integer-Ausdruck; mögliche Werte:

Ø: alle File-Identifizier, unabhängig von 'File-Identifizier', werden akzeptiert. Wird dies nicht gewünscht, ist ein Wert anzugeben, der genau gleich der Anzahl Zeichen von 'File-Identifizier' ist. Ist 'Länge' kleiner als die Anzahl Zeichen von 'File-Identifizier', werden von links kommend nur soviel Zeichen aus 'File-Identifizier' entnommen, wie durch 'Länge' bestimmt.

Ergebnisstring: Stringvariable;

enthält nach dem Aufruf den gefundenen File-Identifizier; wird kein passender File-Identifizier gefunden, entsteht der BASIC-Fehler 4Ø, der in einer Fehlerbehandlungsroutine verarbeitet werden kann (BASIC-Anweisung **ON ERROR GOTO**).

Vor Aufruf von **by** muß 'Ergebnisstring' einen Inhalt von genau 14 Zeichen haben; das erste Zeichen sollte ein ? sein.

Beispiel: "?2345678901234"

Funktionen wie **SPACE\$** oder **STRING\$** dürfen nicht zur Zuweisung des Inhalts verwendet werden.

File-Identifizier: String-Ausdruck; enthält den File-Identifizier, nach dem gesucht werden soll. Es kann Filenamenauswahl verwendet werden, allerdings nur *, nicht ?.

BEISPIEL:

```
2 PRINT "Files auf D:, die auf .cmd enden:"
10 FU#="search":SN#="0":L#="5":S#="*.cmd"
20 SA#="?12345678901234"
30 START#="1":PRINT
50 ON ERROR GOTO 2000
60 CALL "bv" (FU#,SN#,START#,L#,SA#,S#)
70 PRINT SA#
80 START#="0":GOTO 60
2000 END
```

bei "getvolname":

welche Diskettenbezeichnung die Diskette hat, die in der zuletzt davor angesprochenen Station einliegt.

Die Parameterliste entfällt.

In die vorher mit Wert zu belegende Stringvariable 'Diskettenbez.' wird linksbündig die Diskettenbezeichnung abgestellt. Hat die Diskette keinen Namen, bleibt der Wert von 'Diskettenbez.' unverändert.

BEISPIEL:

```
1 CLEAR:ON ERROR GOTO 3010
10 FU$="getvolname":V$=SPACE$(14)
15 M$="Station nicht vorhanden/leer oder Lesefehler"
20 LINE INPUT "Station (0,1,10) ";SN$
30 IF INSTR("10",SN$)=0 THEN 20
41 CALL "bv"(FU$,V$):'nur Residentmachung!
50 EXEC "-dcons:":FILES SN$+":":EXEC "+dcons:"
55 'Aktivieren richtige Station!
60 CALL "bv"(FU$,V$)
65 IF V$=SPACE$(14) THEN PRINT "kein Diskettenname":END
70 PRINT "Diskettenname: ";V$:END
3010 EXEC "+dcons:"
3012 IF ERR=57 THEN PRINT M$-RESUME 20
3015 IF ERL=41 AND ERR=40 THEN 3030
3020 ON ERROR GOTO 0:STOP
```

bei "diskfree":

wieviele freie Sektoren auf der durch 'Stationsnr.' bezeichneten
Diskette vorhanden sind.

FORMAT der Parameterliste:

§freie Sektoren

freie Sektoren: Integervariable;
 enthält nach dem Aufruf die Anzahl freier
 Sektoren

BEISPIEL:

```
0 CLEAR:DFX=0:ON ERROR GOTO 3000
10 M$="Station nicht vorhanden/leer oder Lesefehler"
20 LINE INPUT "Station (0,1,10) ";SN$
30 IF INSTR("01",SN$)=0 THEN 20
40 SNZ=VAL(SN$)
50 CALL "bv"("diskfree",SNZ,%DFX)
60 PRINT "Auf der Diskette in Station ";SN$;
70 PRINT ": sind";DFX;"Sektoren frei."
999 END
```

BEMERKUNGEN:

- **bv** kann nur in der BASIC-Ebene mittels der Anweisung **CALL** aufgerufen werden.
- Alle Variablen müssen entweder Typ Integer oder String sein und vor Aufruf einen Inhalt bekommen haben (z.B. durch eine wertzuweisende Anweisung der Art **T%=Ø** oder **T\$=SPACE\$(15)**).
Bei Stringvariablen muß der Inhalt lang genug sein, um alle später zu erwartenden Ergebnisse aufzunehmen. Für 'Ergebnisstring' gilt eine besondere Regelung.
- **bv** bleibt nach dem ersten Aufruf bis zum Neuladen des Betriebssystems resident, d.h. das Laden mit **pl** ist überflüssig; **bv** kann aber mit **pu** jederzeit wieder aus dem Arbeitsspeicher entfernt werden.



FUNKTION:

Befehl zur Kommunikation mit dem RS-232-Driver

FORMAT: (nur BASIC)

```
CALL "ci"(Port-Nr., "RS-232-Befehl",  
          $Fehler-Variable, [Parameterliste])
```

WIRKUNG:

Der 'RS-232-Befehl' wird der spezifizierten RS-232-Einheit gegeben. Die Parameter in der 'Parameterliste' hängen ab vom speziellen RS-232-Befehl. Nach Ausführung des Befehls beinhaltet die 'Fehler-Variable' eine Meldung über das Ergebnis der Operation.

BEMERKUNGEN:

- ci kann nur in der BASIC-Ebene mittels der Anweisung **CALL** aufgerufen werden.
- Anzahl und Reihenfolge der Variablen in der Parameterliste hängen von dem verwendeten 'RS-232-Befehl' ab. Die einzelnen Variablen sind durch Komma voneinander zu trennen.
- Alle Variablen müssen in Form von Integer- bzw. Stringvariablen definiert sein. Alle Variablen müssen vor dem ersten Aufruf von ci eine Wertzuweisung erhalten haben (z.B. durch **M%=0** am Anfang des Programms).

- Der 'RS-232-Befehl' muß in Form einer Konstanten angegeben sein.
- 'Port-Nr.' darf folgende Werte annehmen:
 - Ø: Standard-RS-232-Port
 - 1: erster RS-232-Port auf der Erweiterungsplatine
 - 2: zweiter RS-232-Port auf der Erweiterungsplatine
- Um mit `ci` arbeiten zu können, muß vorher der PCOS-Befehl `rs` gegeben worden sein und über `sc` die Übertragungsparameter gesetzt worden sein.
- Die folgenden RS-232-Befehle sind möglich:
 - `o` Öffnen des Ports
 - `c` Schließen des Ports
 - `w` Senden an den Port
 - `r` Empfangen vom Port
 - `sr` Empfangsstatus vom Port annehmen
 - `sw` Sendestatus vom Port annehmen.
- Die 'Fehler-Variable' kann nach Abarbeitung des 'RS-232-Befehls' folgende Werte enthalten:
 - Ø: Kein Fehler
 - 1: unzulässiger Parameter
 - 2: Fehler beim Öffnen oder Schließen des Ports
 - 3: Zeitfehler
 - 4: Paritätsfehler, Overflow
- Die 'Fehler-Variable' muß mit dem Präfix `§` (ohne Blank) eingeleitet sein. Dies gilt für alle Variablen, die von PCOS an den BASIC-Interpreter zu übergebende Werte beinhalten.
- `ci` bleibt nach dem ersten Aufruf bis zum Neuladen des Betriebssystems ständig resident; die Verwendung von `pl` ist somit überflüssig. `pu` kann jedoch nicht eingesetzt werden.

- Weitere Informationen zur Programmierung von RS-232-Einheiten entnehmen Sie bitte dem Handbuch "V 24-Peripherie, Programmierhandbuch".

BEISPIELE:

```
71 CALL "ci"(COMNR%,"r",%FEHLER%,%RET%)
```




(change key code)

FUNKTION:

Ändern bzw. Abfrage der Wirkung von SHIFT (Feststellfunktion) oder Ändern der Funktion bzw. originären Belegung einer Taste(n-kombination)

FORMAT:

ck	{	%f Umschalter	(Fall I)
		Tastencode [, { Original-ISO-Code }]	(Fall II)

oder

(nur BASIC)

CALL "ck"("%f",Umschalter [,§alter Umschalter]) (Fall I)

Umschalter: vom Typ Integer, wenn Variable

alter Umschalter: Integervariable, in der nach Aufruf von ck der vorherige Wert von 'Umschalter' steht

WIRKUNG:

I

Ist als erster Parameter %f angegeben, wird der Zustand der Feststellfunktion (= Wirkung von SHIFT auf alphanumerischem bzw. numerischem Tastaturblock) ermittelt bzw. geändert.

'Umschalter' darf folgende Werte haben:

- Ø beide Feststellfunktionen inaktiv
- 1 Buchstaben-Feststellfunktion aktiv,
Cursor-Feststellfunktion inaktiv
- 2 Cursor-Feststellfunktion aktiv,
Buchstaben-Feststellfunktion inaktiv
- 3 beide Feststellfunktionen aktiv

Ist die Buchstaben-Feststellfunktion aktiv, werden bei normalem Tastendruck (ohne SHIFT) auf dem alphanumerischen Tastaturblock die Groß- statt der Kleinbuchstaben erzeugt und bei gleichzeitigem Betätigen von SHIFT die Klein- statt der Großbuchstaben.

Dies gilt nicht für Tasten mit Doppelbeschriftung wie z.B. 9/§. (Bei den deutschen Tastaturen gilt es für ä, ö, ü, nicht jedoch bei der Schweizer Tastatur, deutsche Version).

Ist die Cursor-Feststellfunktion aktiv, werden bei normalem Tastendruck auf den numerischen Tastaturblock diejenigen Zeichen erzeugt, die normalerweise durch Umschalten mit SHIFT erzeugt werden, z.B. HOME. Die Ziffern können dann durch Betätigen der jeweiligen Taste i.V. mit SHIFT erzeugt werden.

Bei einem Aufruf in der BASIC-Ebene kann in der Integervariablen 'alter Umschalter' zusätzlich der vorherige Zustand der Feststellfunktion abgefragt werden.

II

Einem Tastencode (vgl. Kap. 3.1) wird ein bestimmter neuer Original-ISO-Code bzw. das entsprechende ISO-Zeichen zugeordnet; es existieren für einige Sonderfunktionen (wie z.B. **C**, logischer Reset oder Aktivieren/Desaktivieren Feststellfunktion) eigene Funktions-Codes.

Bei Weglassen von 'Original-ISO-Code' bzw. 'neues Zeichen' wird angegeben, welcher ISO-Code bzw. Funktionscode welchem Tastencode zugeordnet ist (Angaben dezimal).

Tabellen der Tastencodes sowie der standardmäßig zugeordneten Funktionscodes und Original-ISO-Codes für verschiedene Tastaturen finden sich in den Kapiteln 3.1 und 3.2.

BEMERKUNGEN:

- Die Buchstaben-Feststellfunktion kann jederzeit durch Betätigen von COMMAND-Taste i.V. mit der --Taste (unten rechts im alphanumerischen Tastaturblock) aktiviert werden; die Cursor-Feststellfunktion durch **-** (also CONTROL-Taste i.V. mit der --Taste).
- Ist einer Taste(nkombination) ein Funktionscode zugeordnet, wird bei Ausgabe der Zuordnung Tastencode \leftrightarrow ISO-CODE der Funktionscode ausgegeben und nicht der ISO-Code des Zeichens, der der Taste zugeordnet ist.

- Alle Codes können dezimal oder hexadezimal angegeben werden. Bei Angabe in hexadezimaler Form muß der hexadezimalen Zeichenfolge ein & vorangestellt werden.
- Die Variable 'alter Umschalter' muß vom Typ Integer sein und muß vor dem ersten Aufruf einen Wert erhalten haben (z.B. durch eine Zuweisung wie T%=0).

BEISPIELE:

- ck &C3,8 Die S2-Taste (mit dem Tastencode &C3) erhält die Wirkung von **H** (dezimaler Code 8) zugeordnet.
- ck &C3 (nach Ausführung des obigen ck-Befehls)
Es wird ausgegeben:
KEY = 195 (dezimaler Wert von &C3)
CODE = 8 (zugeordneter Funktions-Code, dezimal)
- ck &64,&AF der Taste **C** wird eine Nulloperation zugeordnet
- ck &64,&A2 (nach Ausführung des vorherigen ck-Befehls)
die ursprüngliche Abbruchfunktion wird **C** wieder zugeordnet
- ck &72,'w' der Taste **Q** wird die Erzeugung von **w** zugeordnet
- 11 S%=1:SA%=-1 in der BASIC-Ebene wird die Buchstaben-Feststellfunktion aktiviert und die Cursor-Feststellfunktion deaktiviert. In SA% steht, je nach vorherigem Zustand, Ø, 1, 2 oder 3.
- 12 CALL "ck"
 ("%f",S%,§SA%)



(describe configuration)

FUNKTION:

Anzeige der aktuellen Hardware-Konfiguration bzw. der Speicherbelegung

FORMAT:

```
dc [% {h
      m} [h]
```

WIRKUNG:

Wird nur `dc` angegeben, wird die Meldung ausgegeben, die nach dem Laden von PCOS und dem Übergang in die PCOS-Ebene am Bildschirm erscheint, z.B.

```
LI.M20 System configuration:
  Total memory size      160 KBytes.
  Free memory size      54076 Bytes.
  Basic memory size     34528 Bytes.
  Display Type:         Black and White.
  Disk drive(s):        2 Ready.
```

Wird hinter `%` der Parameter `h` eingegeben, wird die ausführliche Hardware-Konfiguration ausgegeben, z.B.

```
Hardware Configuration:
  Memory Configuration: 7
  Floppy Disk Drive(s): 320 KByte.
  Drive 0 Diskette:     320 KByte.
  Drive 1 Diskette:     320 KByte.
  Hard Disk Drive:      Present.
  IEEE Board:           Not Present.
  RS-232 Board:         Not Present.
  8086 CPU Board:       Not Present.
  Display Type:         Black and White.
```

Wird hinter % der Parameter m angegeben, wird angezeigt, wie der Speicher aktuell belegt ist, z.B.

Memory Configuration:

Total memory size: 160 KBytes.
 Free memory size: 54076 Bytes.
 Basic memory size: 34528 Bytes.

Address (Hex)		Block Size		Filename (Owner)
Segment	Offset	Hex	Decimal	
00	0004	3FF0	16380	PC05.SAV

Address (Hex)		Block Size		Filename (Owner)
Segment	Offset	Hex	Decimal	
06	0004	7FF0	32764	basic.cmd
06	8004	3FF0	16380	PC05.SAV

Address (Hex)		Block Size		Filename (Owner)
Segment	Offset	Hex	Decimal	
02	0084	0178	376	PC05.SAV
02	0200	1480	5248	basic.cmd
02	1684	1140	4428	PC05.SAV
02	3704	4140	16716	FREE BLOCK
02	6924	0DF2	3570	keyboard.cmd
02	771A	86E0	34528	FREE BLOCK

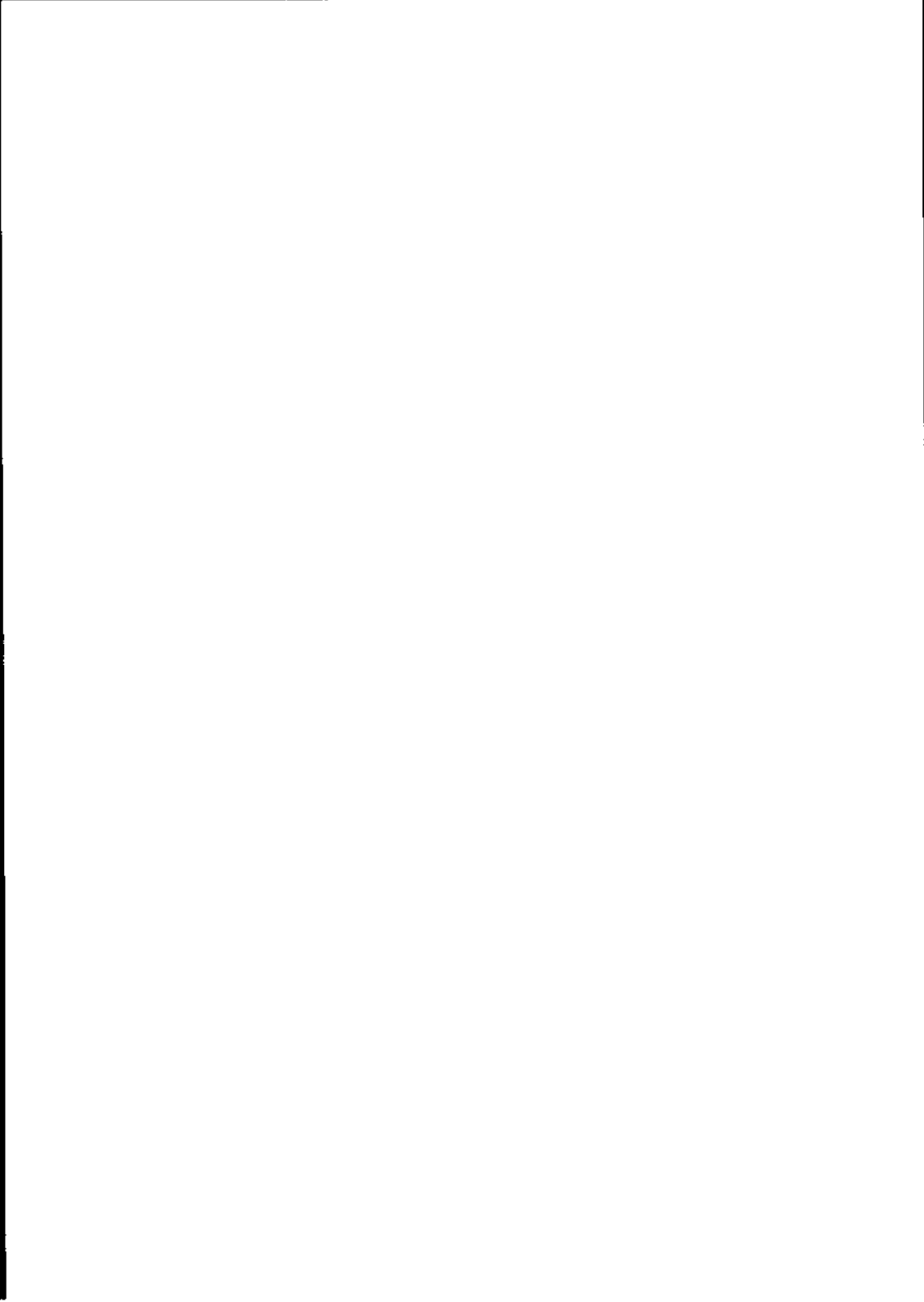
Address (Hex)		Block Size		Filename (Owner)
Segment	Offset	Hex	Decimal	
0A	8004	0120	288	FREE BLOCK
0A	3128	0768	1928	aconfig.cmd
0A	89B4	05A6	1446	label.cmd
0A	8E8E	0120	300	FREE BLOCK
0A	8F8E	004E	78	PC05.SAV
0A	8FE0	03F2	1010	basic.cmd
0A	93D6	0040	76	FREE BLOCK
0A	9426	0800	2048	isam.sav
0A	9C2A	215A	8538	isam.sav
0A	8D88	0020	32	FREE BLOCK
0A	BDAC	0028	40	basic.cmd

Sind nach % sowohl m als auch h angegeben, wird sowohl die ausführliche Hardware-Konfiguration als auch die aktuelle Speicherbelegung angezeigt.

BEISPIELE:

dc +dprt: gibt die Meldung von PCOS nach dem Laden des Betriebssystems sowohl auf dem Bildschirm als auch auf dem Drucker aus

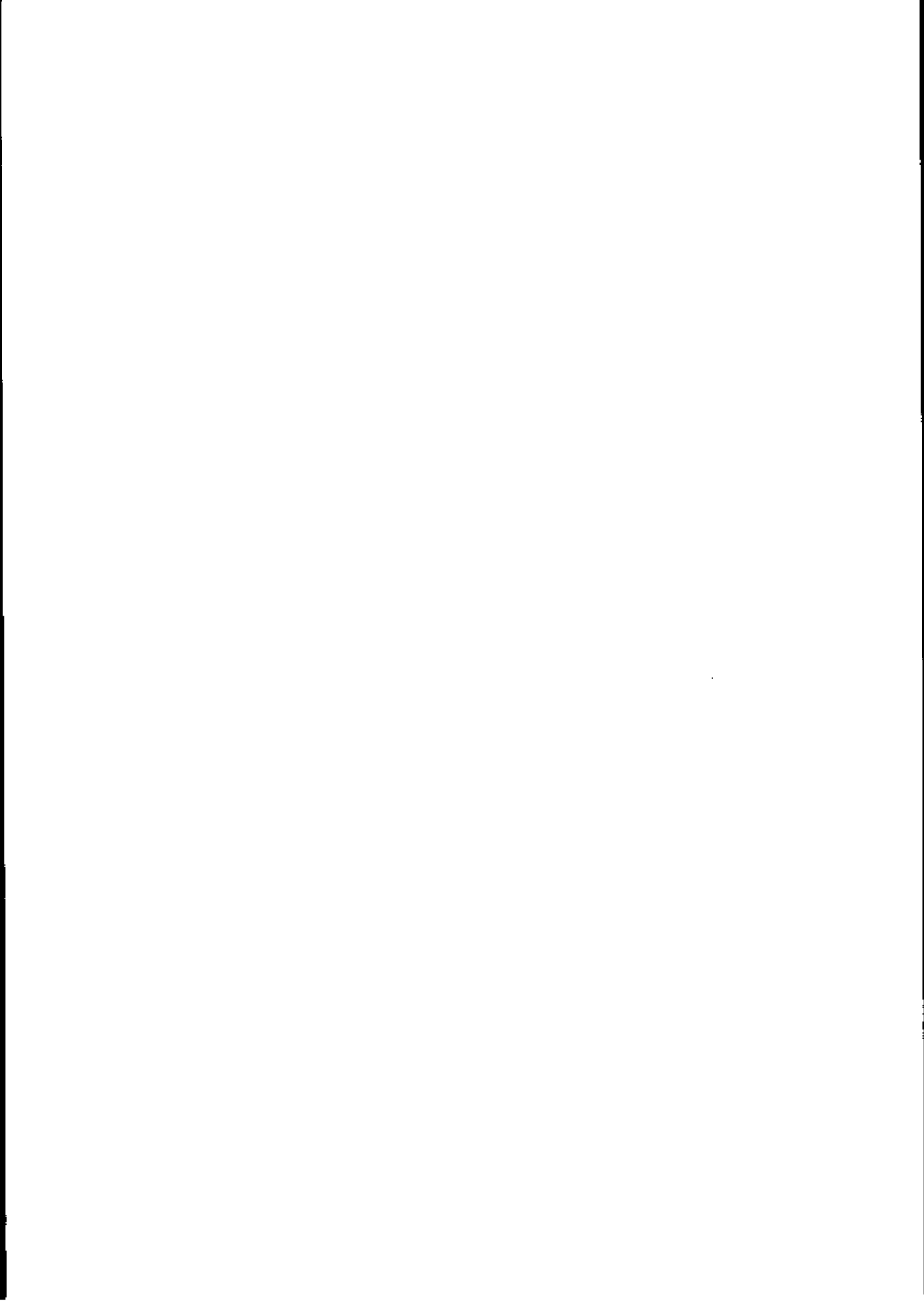
**dc %hm -dcons:,
+dprt:,+dl:KONF.
seq** gibt Hardware-Konfiguration und Speicherbelegung auf dem Drucker, nicht aber auf dem Bildschirm aus und speichert sie in ein sequentielles File namens KONF.seq auf der Diskette in Station 1:



Zu **dc**-Befehl

Folgende Werte sind bei **dc %h** für 'Memory Configuration' möglich:

- 0 8-Farb-System mit einer oder mehr
128-KB-Erweiterung(en)
- 1 4-Farb-System mit einer oder mehr
128-KB-Erweiterung(en)
- 2 8-Farb-System mit einer oder mehr
32-KB-Erweiterung(en)
- 3 4-Farb-System mit einer oder mehr
32-KB-Erweiterung(en)
- 5 Schwarz/Weiß-System ohne Erweiterung
- 6 Schwarz/Weiß-System mit einer oder mehr
128-KB-Erweiterung(en)
- 7 Schwarz/Weiß-System mit einer oder mehr
32-KB-Erweiterung(en)



FUNKTION:

Laden des Full-Screen-Editors und Aufruf eines zu editierenden Files (PCOS-Textfile oder BASIC-File (ASCII- oder sequentielles File))

FORMAT:

ed	[%b]	[%r]	[%c]	Filename
----	------	------	------	----------

WIRKUNG:

Ist kein zusätzlicher Parameter angegeben, wird der Full-Screen-Editor geladen und der Anfang des mit 'Filename' bezeichneten Files zum Editieren freigegeben. Ist das File noch nicht vorhanden, wird in Zeile 1

OK to Create?

gemeldet. Die Frage kann mit **y** (für ja) oder **n** (für nein) beantwortet werden. Bei **y** wird das File angelegt und zum Editieren freigegeben, bei **n** wird der Full-Screen-Editor verlassen.

Ist **%r** angegeben, wird das betreffende File nur gelesen und nicht zur Änderung freigegeben.

Ist das zu editierende File ein ASCII-File (im allg. gewonnen mit Hilfe des BASIC-Befehls **SAVE** Filename ,A), oder ein über BASIC erstelltes sequentielles Datenfile (BASIC-Anweisungen **OPEN "0"**, **PRINT#** oder **WRITE#**), muß der Parameter **%c** angegeben sein. Die

Programm bzw. Textzeilen dürfen nicht länger als 80 Zeichen (incl. Zeilennr.) sein; alle Zeichen hinter dem 80. Zeichen werden vom Full-Screen-Editor ignoriert und beim Abspeichern nicht mehr geschrieben. Wird über `ed %c` ein ASCII-File editiert, müssen die Zeilen nicht in der Reihenfolge der Zeilennummern im File stehen. Die richtige Reihenfolge wird automatisch beim BASIC-Befehl **LOAD** hergestellt. Treten in einem ASCII-File mehrere gleiche Zeilenrn. auf, ist von diesen nach **LOAD** nur noch diejenige vorhanden, die dem File-Ende am nächsten ist.

BASIC-Programmfiles, die nicht mit Parameter **A** bei **SAVE** gespeichert wurden, können mit dem Full-Screen-Editor ebensowenig bearbeitet werden wie Random-Files.

Ist `%b` angegeben, wird vor Aufruf des Files mit dem Full-Screen-Editor zuvor eine Sicherungskopie des Files 'Filename' auf der Diskette erstellt, auf der sich das File befindet. Der Name dieses Sicherungsfiles wird gebildet durch Anhängen von `.bak` an 'Filename'.

Über OLWORD erstellte Textfiles können mit dem Full-Screen-Editor nicht bearbeitet werden.

Filetypen:

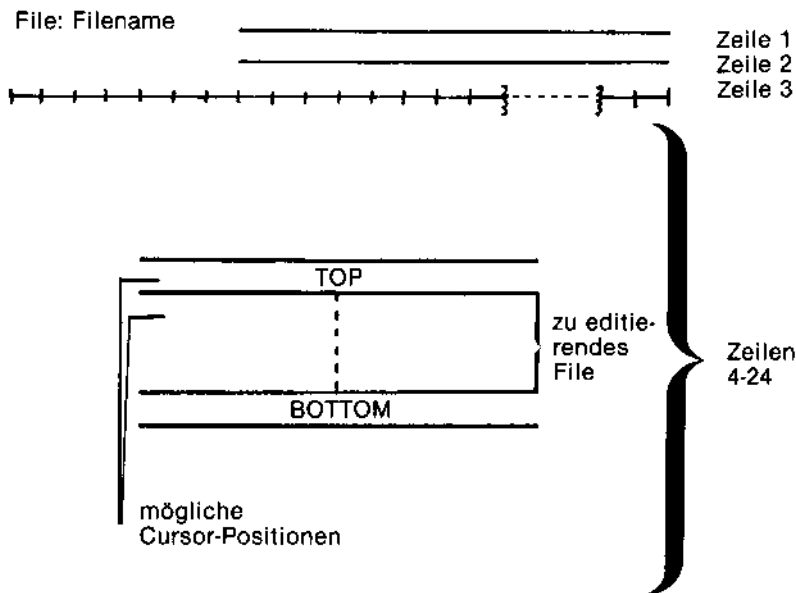
Der Full-Screen-Editor erzeugt Zeilen aus druckbaren ASCII-Zeichen. Die Zeilen werden bei Nichtsetzen von `%c` durch das Zeichen mit dem ISO-Code 30 (dezimal; = RS) getrennt. Solche Files werden als PCOS-Textfiles bezeichnet.

Ist %c gesetzt, werden die einzelnen Zeilen statt durch das Zeichen RS durch eine Kombination der Zeichen mit den ISO-Codes 13 und 10 (dezimal; = CR und LF) getrennt.

Sollen nicht druckbare Zeichen (ISO-Codes 0 bis 31 oder 127, dezimal) in das File aufgenommen werden, steht dafür eine besondere Funktion zur Verfügung.

Bildschirm Aufbau:

Nach Einlesen des Files ergibt sich ein Bildschirm Aufbau im 80 x 25-Format wie folgt:



In Zeile 1 ist der Name des gerade editierten Files angegeben. Meldungen des Programms erscheinen ebenfalls dort.

Zeile 2 ist zur Eingabe von zu suchenden Zeichenfolgen oder Subbefehlen vorgesehen.

Zeile 3 zeigt, wo die Tabulationsstops gesetzt sind (auf jeder vierten Zeilenposition einer).

Die Zeilen 4 bis 24 beinhalten den gerade bearbeiteten Teil des Files. Der Text ist von einer TOP- und einer BOTTOM-Zeile eingerahmt, die nicht Bestandteil des Files werden.

Der Cursor steht vor dem Anfang des Files (TOP-Zeile). Soll die erste Zeile geändert oder erfaßt werden, muß der Cursor erst durch Anwendung der Tastenkombination SHIFT ↓ oder durch Betätigen von ↵ (= CR) um eine Zeile tiefer positioniert werden.

Modes:

Innerhalb des Full-Screen-Editors sind folgende Modes zu unterscheiden:

1. Überschreiben (vorhandene Textpositionen werden überschreiben)
2. Einfügen (es wird an der Cursor-Position eingefügt; der Text der Zeile verschiebt sich entsprechend nach rechts; alle Zeichen hinter dem 80. Zeichen gehen verloren)
3. Command-Mode (zur Eingabe von Subbefehlen oder zu suchenden Zeichenfolgen)

Nach Aufruf des Full-Screen-Editors befindet man sich im Überschreiben. Je nach Mode verändert sich die Form des Cursors.

Funktionen:

Über Taste(nkombinationen) können eine Reihe von Funktionen bewirkt werden; die Funktionen werden auf den folgenden Seiten genauer beschrieben. Es ist zu beachten, daß nach Neubelegungen von Tasten über die PCOS-Befehle **pk** bzw. **ck** diese Funktionen evtl. nicht mehr gewährleistet sind.

Subbefehle und zu suchende Zeichenfolgen:

Innerhalb des Full-Screen-Editors existieren drei Subbefehle, die im Command-Mode aufzurufen sind. Die Subbefehle werden auf den folgenden Seiten genauer beschrieben. Im editierten File zu suchende Zeichenfolgen sind ebenfalls im Command-Mode anzugeben und anschließend durch Anwendung bestimmter Funktionen zu suchen.

Abschluß:

Mit Hilfe verschiedener Funktionen wird der Full-Screen-Editor verlassen, wobei das geänderte File abgespeichert werden kann oder auch nicht.

Beschreibung der Funktionen:

1. Tasten der oberen Tastaturreihe i.V. mit COMMAND (gelbe Taste):

Command-Mode,
Übergang in
Command-Mode
(COMMAND 1)

Cursor in die zweite Zeile zur Eingabe von Subbefehlen (GOTO, DELETE, FILE) oder zu suchender Zeichenfolge; Auslösung der Subbefehle und der Zeichenfolgen durch CR, Suche durch CONTROL bzw. COMMAND 8. Alle Edit-Operationen (z.B. Einfügen, ein Zeichen zurück) beziehen sich auf Zeile 2.
Rückkehr zum Editing des Files durch erneut COMMAND 1.

Löschen Zeile,
Aufnahme Zeile
in Zeilenpuffer
(COMMAND 2)

Löschen der aktuellen Zeile, Folgetext um eine Zeile nach oben; Spaltenposition des Cursors unverändert. Die gelöschte Zeile wird im Zeilenpuffer abgelegt und kann von dort durch COMMAND 5 (Einfügen Zeilen-Puffer) an andere Stelle übertragen werden. COMMAND 2 löscht den alten Inhalt des Zeilenpuffers, es sei denn, mehrere COMMAND 2-Funktionen erfolgen direkt hintereinander; dann werden die betroffenen Zeilen hintereinander im Zeilenpuffer gespeichert.

Aufspalten Zeile
(COMMAND 3)

Übertragung des Textes ab Cursor-Position in die nächste Zeile. Keine Bewegung des Cursors; alle Folgezeilen werden um eine Zeile nach unten verschoben.

Zeilenanfang (COMMAND 4)	Cursor an Zeilenanfang
Einfügen Zeilen- puffer (COMMAND 5)	Einfügen des Inhalts des Zeilenpuffers in die nächste(n) Zeile(n). Alle Folgezeilen entsprechend nach unten verschieben. Inhalt des Zeilenpuffers wird nicht verändert. COMMAND 2/COMMAND 5-Kombinationen können auch verwendet werden, um Zeilen von BASIC-ASCII-Files zu kopieren (z.B. um Zeilennrn. zu verändern). Beachte: %c muß dazu bei ed-Aufruf gesetzt sein.
Ende ohne Speichern (COMMAND 6)	Full-Screen-Editor verlassen, ohne veränderte Fassung des Files auf Diskette zu schreiben. Ist am File geändert worden, wird mit Confirm Abort Bestätigung angefordert; diese erfolgt <u>nur</u> durch erneutes Betätigen von COMMAND 6, andernfalls wird die Funktion nicht ausgeführt.
Setzen Markierungs- zeile (COMMAND 7)	Setzen der Markierungszeile unmittelbar unter der aktuellen Zeile; eine evtl. vorherige Markierungszeile wird gelöscht. Markierungszeile ist kein Bestandteil des Files und kann mit SHIFT 5 (Sprung zu Marke) angesteuert werden.

Suche nach oben (COMMAND 8)	Nach Eingabe einer mit CR abgeschlossenen zu suchenden Zeichenfolge im Command-Mode (erreicht durch COMMAND 1) und anschließendem COMMAND 8 wird das erste Auftreten dieser Zeichenfolge ab der aktuellen Cursor-Position nach oben im File gesucht und der Cursor dorthin positioniert, wenn gefunden. Im File kann direkt weiter editiert werden.
1 Zeile nach oben (COMMAND 9)	Cursor-Position unverändert; die Textzeilen werden um eine Zeile höher gerückt.
halber Schirm nach oben (COMMAND Ø)	Cursor-Position unverändert; die Textzeilen werden um 10 Zeilen höher gerückt.
voller Schirm nach oben (COMMAND ß)	Cursor-Position unverändert; die Textzeilen werden um 20 Zeilen höher gerückt.
TOP-Zeile (COMMAND ^ bzw. COMMAND ^)	File ab Anfang darstellen; Cursor-Position in TOP-Zeile

2. Tasten der oberen Tastaturreihe i.V. mit CONTROL (blaue Taste)

Einfügen (CONTROL I)	Übergang in den Mode "Einfügen"; alle Eingaben werden ab der aktuellen Cursor-Position eingefügt und nicht mehr überschrieben; was über dem rechten Rand der Zeile liegt, geht verloren; erneutes Betätigen von CONTROL I
-------------------------	---

bewirkt den Übergang in den Mode "Überschreiben".

Löschen Zeichen (CONTROL 2)	Löschen des Zeichens über dem Cursor und Rücken des Textes rechts vom Cursor um ein Zeichen nach links.
Verbinden Zeilen (CONTROL 3)	Der Text in der Folgezeile wird unmittelbar an das letzte Zeichen (nicht Blank) der aktuellen Zeile angeschlossen. Was nicht in die Zeile geht, bleibt in der Folgezeile, rückt aber nach links.
Zeilenende (CONTROL 4)	Positionieren des Cursors unmittelbar hinter das letzte Zeichen (nicht Blank) der aktuellen Zeile.
Speichern (CONTROL 5)	Abspeichern eines Zwischenstandes ohne Verlassen des Full-Screen-Editors bzw. des editierten Files bei vorangegangenem FILE-Subbefehl.
Ende mit Speichern (CONTROL 6)	Abspeichern des Standes; Verlassen des Full-Screen-Editors bzw. des editierten Files bei vorangegangenem FILE-Subbefehl.
TAB zurück (CONTROL 7)	um vier Zeichen nach links; falls über den linken Rand, vom rechten Rand her kommend.

Suche nach unten (CONTROL 8) Nach Eingabe einer mit CR abgeschlossenen zu suchenden Zeichenfolge im Command-Mode (erreicht durch COMMAND 1) und anschließendem CONTROL 8 wird das erste Auftreten dieser Zeichenfolge ab der aktuellen Cursor-Position nach unten im File gesucht und der Cursor dorthin positioniert, wenn gefunden. Im File kann direkt editiert werden.

1 Zeile nach unten (CONTROL 9) Cursor-Position unverändert; die Textzeilen werden um eine Zeile nach unten gerückt.

halber Schirm nach unten (CONTROL Ø) Cursor-Position unverändert; die Textzeilen werden um 10 Zeilen nach unten gerückt.

voller Schirm nach unten (CONTROL B) Cursor-Position unverändert; die Textzeilen werden um 20 Zeilen nach unten gerückt.

BOTTOM-Zeile (CONTROL ~ bzw. CONTROL `) Fileende darstellen; Cursor-Position in BOTTOM-Zeile

3. andere Tasten i.V. mit CONTROL (blaue Taste)

Löschen 1 Zeichen + zurück (CONTROL H) Positionieren Cursor eine Position nach links; Überschreiben des bisherigen Zeichens durch Blank; (gleiche Wirkung wie S2).

Löschen Zeilenrest (CONTROL K)	Löschen des Zeilenrests ab Cursor-Position.
Erneuern Anzeige (CONTROL L)	Der aktuell gültige File-Ausschnitt wird am Bildschirm neu abgebildet.
Wiederherstellen Zeile (CONTROL R)	Der Zustand der Zeile wird wieder so hergestellt, wie er war, als der Cursor in diese Zeile gebracht wurde.

4. Abschlußtasten

Zeile einfügen/ Abschluß Command- Mode-Eingabe (↵ = CR)	im File wird eine Leerzeile hinter der aktuellen Zeile eingefügt; ist der Cursor in der 2. Zeile (Befehlszeile), gilt CR als Abschlußtaste.
TAB vor (S1)	um vier Zeichen nach rechts; falls über den rechten Rand, von links kommend.
Löschen 1 Zeichen + zurück (S2)	Positionieren Cursor eine Position nach links; Überschreiben des bisherigen Zeichens durch Blank; (gleiche Wirkung wie CONTROL H).

5. numerische Tasten i.V. mit SHIFT (evtl. (nach CONTROL-) ohne SHIFT)

Cursor oben (SHIFT 8)	Cursor eine Zeile nach oben; gleiche Spalte.
--------------------------	--

Cursor links (SHIFT 4)	Cursor ein Zeichen nach links; gleiche Zeile.
Sprung zu Marke (SHIFT 5)	Cursor wird in Markierungszeile positioniert; (diese muß vorher mit COMMAND 7 definiert worden sein); Markierungszeile selbst ist kein Bestandteil des Files.
Cursor rechts (SHIFT 6)	Cursor ein Zeichen nach rechts; gleiche Zeile.
nächste Zeile (SHIFT 1)	Cursor um eine Zeile nach unten an den Anfang der nächsten Zeile.
Cursor unten (SHIFT 2)	Cursor eine Zeile nach unten; gleiche Spalte.
Einleiten Steuerz. (SHIFT 3)	fügt das ESC-Steuerzeichen (ISO-Code 27, dezimal) ein; die betreffende Position wird in Reverse dargestellt. Das Zeichen ESC dient nur dazu, das Folgezeichen als Steuerzeichen zu kennzeichnen. Der Code des nächsten Zeichens wird nur aus den niedrigsten fünf Bit des ISO-Codes entwickelt.

Bsp.: Die Zeichen mit den ISO-Codes 7 und 23 (dezimal) sollen eingefügt werden:

SHIFT 3, dann SHIFT G, dann SHIFT 3, dann 7.

Um das Zeichen mit dem ISO-Code 127 (dezimal) zu erzeugen, muß SHIFT 3 und anschließend ? eingegeben werden.

Beschreibung der Subbefehle:

Subbefehle und zu suchende Zeichenfolgen können nur im Command-Mode (zu erreichen mit COMMAND 1) in der zweiten Zeile eingegeben werden. Sie sind dort stets mit CR auszulösen. Der Command-Mode wird durch COMMAND 1 wieder verlassen.

GOTO

SYNTAX: GOTO interne Zeilennr.

interne Zeilennr. Integerkonstante;
Wert: > = 0

WIRKUNG:

Nach Eingabe des Subbefehls (Abschluß durch CR) und Verlassen des Command-Modus (COMMAND 1) steht der Cursor in der durch 'interne Zeilennr.' definierten Textzeile.

Die Zeilen des Files werden intern durchnummeriert. Die TOP-Zeile hat die Nummer 0. Die BOTTOM-Zeile zählt nicht. Die 'internen Zeilennrn.' werden nicht angezeigt.

Ist %c gesetzt und wird ein ASCII-File editiert, entspricht die 'interne Zeilennr.' nicht den BASIC-Programmzeilennrn.

Ist 'interne Zeilennr.' größer als die Anzahl Zeilenrn., wird der Cursor an den Anfang der letzten Zeile des Files bewegt.

DELETE

SYNTAX: DELETE

WIRKUNG:

Nach Eingabe des Subbefehls, abgeschlossen mit CR (vorher: COMMAND 1), werden sofort im File alle die Zeilen gelöscht, die zwischen der Markierungszeile und derjenigen Zeile (incl.) stehen, in der der Cursor bei COMMAND 1 positioniert war. Die gelöschten Zeilen werden im Zeilenpuffer gespeichert und können ggf. mit Hilfe von COMMAND 5 (Einfügen Zeilenpuffer) an andere Stelle übertragen werden.

Existiert keine Markierungszeile (zu setzen mit COMMAND 7), wird eine Meldung gegeben.

Nach dem DELETE-Subbefehl kann im File weiter editiert werden, indem mit COMMAND 1 der Command-Mode verlassen wird.

FILE

SYNTAX: FILE Filename
 Filename: PCOS-gerechter Filename (vgl.
 Kapitel 2.2 und 2.3)

WIRKUNG:

Der Subbefehl ermöglicht, ein anderes File zum Ändern aufzurufen,

ohne den Full-Screen-Editor zu verlassen und mit **ed** den Full-Screen-Editor erneut aufzurufen.

Nach dem FILE-Subbefehl (eingegeben im Command-Mode, ausgelöscht durch CR) wird das durch 'Filename' spezifizierte File zum Editieren freigegeben. Ist das File nicht vorhanden, kann es (wie beim direkten Aufruf mit **ed**) durch Beantworten der Frage **OK to Create?** mit **y** angelegt werden.

Das File wird im Bildschirm angezeigt; nach COMMAND 1 zum Verlassen des Command-Modus kann es editiert werden. Es bleibt die Parameterauswahl (**%c**, **%r**, **%b**) gültig, die bei Aufruf des Full-Screen-Editors (PCOS-Befehl **ed**) gesetzt war.

Das Editieren des Files kann durch CONTROL 6 (Ende mit Speichern) oder COMMAND 6 (Ende ohne Speichern) beendet werden. Danach wird das vorher editierte File automatisch wieder an dem Punkt aufgerufen, an dem es verlassen wurde.

Der Inhalt des Zeilenpuffers bleibt über den FILE-Subbefehl hinaus erhalten.

Die Anzahl gleichzeitig editierbarer Files ist nur begrenzt durch ihre Größe und die Kapazität des Arbeitsspeichers.

Überblick über die Funktionen und Subbefehle nach Zusammenhängen

Ø. Ändern Mode

COMMAND 1

CONTROL 1

1. Bewegen des Cursors im File (ohne Löschen, ohne Veränderung der Zeilenpositionen im Bildschirm)

SHIFT 8

SHIFT 4

SHIFT 6

SHIFT 2

S1

CONTROL 7

COMMAND 4

CONTROL 4

COMMAND 1, GOTO, COMMAND 1

2. Bewegen der Zeilenpositionen im Bildschirm (ohne Löschen)

↵ (= CR)

COMMAND 9

COMMAND Ø

COMMAND B

COMMAND ` (bzw. COMMAND ~)

CONTROL 9

CONTROL Ø

CONTROL B

CONTROL ` (bzw. CONTROL ~)

SHIFT 1

SHIFT 5

COMMAND 1, GOTO, COMMAND 1

3. Einfügen Zeilen oder Zeichen
←j (= CR)
CONTROL 1
COMMAND 5
4. Löschen Zeichen
S2 oder CONTROL H
COMMAND 2
CONTROL K
5. Löschen Zeile(n), Speichern Zeile(n) im Zeilenpuffer
CONTROL 2
COMMAND 1, **DELETE**, COMMAND 1
6. Einfügen gespeicherte Zeile(n)
COMMAND 5
7. Wiederherstellen alten Zeileninhalt
CONTROL R
8. Positionen setzen und selbstgesetzte Positionen suchen
COMMAND 7
SHIFT 5
COMMAND 1, zu suchende Zeichenfolge eingeben, COMMAND 8
COMMAND 1, zu suchende Zeichenfolge eingeben, COMMAND 9
9. Zeilen aufspalten bzw. zusammensetzen
COMMAND 3
CONTROL 3

10. Abspeichern File

CONTROL 5

CONTROL 6

11. Aufruf anderer Files/Verlassen Full-Screen-Editor

COMMAND 1, FILE, COMMAND 1

CONTROL 6

COMMAND 6

12. Einfügen Steuerzeichen

SHIFT 3



FUNKTION:

Ausgabe der kurzen englischen Beschreibung hinter allen PCOS-Fehlermeldungen

FORMAT:

ep [Fehlercode] [,Fehlercode] ...

WIRKUNG:

Ist nur **ep** angegeben, wird nur der PCOS-Befehl bis zum nächsten Laden des Betriebssystems resident gemacht und von da an bei allen auftretenden PCOS-Fehlermeldungen der Fehlercode (ERROR nnn) und dahinter kurz in Englisch die Bedeutung des Fehlers ausgegeben. Ist hinter **ep** ein oder mehrere Fehlercodes (zwischen 0 und 127 incl.) spezifiziert, werden sofort der Fehlercode und die Bedeutungen angezeigt. Bei mehreren Fehlercodes werden die Beschreibungen untereinander ausgegeben.

BEMERKUNGEN:

- **ep** wird bereits durch den ersten Aufruf bis zum Neuladen des Betriebssystems resident. Die Residentmachung durch **pl** kann somit entfallen.
- **ep** kann nicht mit **pu** aus dem Arbeitsspeicher entfernt werden.
- Wird **ep** in der BASIC-Ebene aufgerufen (**CALL** oder **EXEC**), wird dadurch nicht der BASIC-Fehler 40 erzeugt und somit auch nie eine mit **ON ERROR GOTO** aktivierte Fehlerbehandlungsroutine

erreicht.

- Eine detaillierte Beschreibung der PCOS-Fehler findet sich im Kapitel 4 dieses Handbuches.

BEISPIELE:

ep lädt das Paket mit der Kurzbeschreibung der PCOS-Fehler

ep 110 gibt hinter 'ERROR 110 - - -' die Bedeutung des Fehler aus (hier: time-out)

FUNKTION:

Kopieren eines File-Inhaltes in ein anderes oder Anhängen eines File-Inhalts an ein bereits vorhandenes File bzw. Kopieren einer ganzen Gruppe von Files mit gemeinsamen Eigenschaften; die zu kopierenden Files werden in aktueller Länge in direkt aufeinander folgende Sektoren übertragen

FORMAT:

fc	{	{	{ %A }	Filename des zu	{	t	{	Filename des Zielfiles
		}	%F }	kopierenden Files				,
		{	%U }	Filename(nauswahl)	{	t	}	Diskettenspezifikation
		}	des/der zu kopierenden	,				}
				Files				

WIRKUNG:

Im ersten Falle wird der Inhalt des zu kopierenden Files in das Zielfile bzw. auf die durch 'Diskettenspezifikation' bezeichnete Diskette kopiert. Das Zielfile kann auf der gleichen Diskette sein wie das zu kopierende File.

Ist das Zielfile nicht vorhanden, wird es angelegt, und zwar in der Größe: Anzahl belegter Sektoren (Sectors USED) des zu kopierenden Files +1.

Ist das Zielfile bereits vorhanden, wird eine Meldung gegeben:

"File already exists. Do you wish to overwrite (y or n)?".

Bei Eingabe von **n** wird das Programm abgebrochen, bei **y** wird das zu kopierende File überschrieben. Es werden dabei nur die belegten Sektoren des zu kopierenden Files berücksichtigt (Sectors USED).

Wird der Parameter **%F** angegeben, wird das File sofort (ohne Meldung) überschrieben, wenn das Zielfile bereits vorhanden ist ("force copy").

Ist der Parameter **%A** angegeben, muß das Zielfile vorhanden sein. Das zu kopierende File wird an den letzten angelegten (Sectors ALLOCATED) Sektor des Zielfiles angehängt; nur die belegten Sektoren (Sectors USED) werden angehängt ("append").

Es wird angegeben, welches File wohin kopiert wird ("**COPY FILE** 'zu kop. File' TO 'Zielfile' ").

Im zweiten Falle können über Filenamenauswahl 1-n Files kopiert werden. Die Diskette für die Zielfiles muß in der anderen Station liegen. Die Zielfiles brauchen jedoch noch nicht vorhanden zu sein. Es wird angegeben, welches File wohin kopiert wird ("**COPY FILE** 'zu kop. File' TO 'Zielfile' "). Ist der Parameter **%U** angegeben, werden mit Kopierschutz versehene Files auf der Zieldiskette weder ins Inhaltsverzeichnis aufgenommen noch (leer) angelegt.

BEMERKUNGEN:

- Ist **%n** ebenfalls angegeben, wird bei Vorhandensein des Zielfiles dieses automatisch überschrieben (wie bei Verwendung von **%F**).

- Die Diskette mit dem Zielfile darf keinen Schreibschutz haben; ist das Zielfile bereits vorhanden, darf es nicht mit File-Schreibschutz versehen worden sein (vgl. **fw**).
- Ein File-Password sowie ein File-Schreibschutz des zu kopierenden Files wird mit übertragen. Hat das Zielfile bereits ein File-Password, bleibt dieses erhalten.
- Die Geschwindigkeit des Kopiervorganges hängt ab von dem im PCOS-Befehl **ss** gesetzten Parameter 'anzuliegende Sektoren'. Je niedriger dieser Wert ist, desto länger dauert die Durchführung von **fc**.
- Nach **fc** ist das File in direkt aufeinander folgenden Sektoren gespeichert, was das Laden von BASIC-Programmen (BASIC-Befehle **LOAD** bzw. **RUN** oder Anweisung **CHAIN**) sowie Lesen und Beschreiben von Datenfiles beschleunigen kann, wenn diese vor dem Kopiervorgang in sehr viele Teile (**EXTENTS**) zerlegt waren (durch ständige Vergrößerung der anfangs mit PCOS-Befehl **fn** oder BASIC-Anweisung **OPEN** festgelegten File-Länge).
- Reicht die Anzahl aufeinander folgender Sektoren ab dem ersten freien Sektor (!) für das/die Zielfile/s nicht aus, wird der Fehler 61 ("disk filled") gemeldet. Das File ist zwar im Inhaltsverzeichnis angelegt, aber nicht vollständig übertragen.

BEISPIELE:

fc 0:ORFile,1:SichFile

Das File "ORFile" auf der Diskette in Station 0 wird auf die Diskette in Station 1 übertragen, und zwar in ein File namens "SichFile". Existiert dieses, wird der alte Inhalt vollständig überschrieben, andernfalls wird "SichFile" angelegt.

fc %A OD:Datenerg.seq/KENNWORT,OD:Daten.seq

Das File "Datenerg.seq" mit dem File-Password "KENNWORT" auf der Diskette namens OD wird auf derselben Diskette an ein File namens "Daten.seq" angehängt, wobei "Daten.seq" ebenfalls automatisch das File-Password "KENNWORT" erhält.

fc %n Ø:KONTR.rnd,1:

Das File namens "KONTR.rnd" auf der Diskette in Station Ø wird unter dem gleichen Namen auf die Diskette in Station 1 kopiert. Es werden keine Meldungen gegeben.

fc Ø:*.cmd,1:

Alle Files auf der Diskette in Station Ø, die auf '.cmd' enden, werden auf die Diskette in Station 1 übertragen.

FUNKTION:

Aufhebung eines File-Passwords für ein File bzw. mehrere Files mit gemeinsamen Eigenschaften

FORMAT:

fd Filename(enauswahl)

WIRKUNG:

Das File-Password, das bei 'Filename'(enauswahl) mit anzugeben ist, wird von dem File bzw. den Files mit gemeinsamem File-Password, die durch 'Filenamenauswahl' angesprochen sind, entfernt.

BEMERKUNGEN:

- Es muß das für das File bisher gültige Password eingegeben werden.
- Die Diskette darf nicht mit Schreibschutz versehen sein.
- Das File darf nicht mit File-Schreibschutz versehen sein (vgl. **fw** und **fu**).
- Die Vergabe oder Änderung eines File-Passwords erfolgt durch den Befehl **fp**.
- Werden von einer Filenamenauswahl mehrere File-Identifizierer angesprochen, wird für jedes File abgefragt, ob das Password gelöscht werden soll ("**Delete Password of 'File-Identifizierer'?**").
- Ist eine Filenamenauswahl angegeben, müssen alle betroffenen Files das gleiche File-Password oder gar keines haben.

BEISPIELE:

fd KONTEN:DEBIT:X/NURINTERN

entfernt das Passwort "NURINTERN" vom File "DEBIT.X". Das File kann wieder unter dem Namen "DEBIT.X" allein angesprochen werden.

fd 1:F*/P

entfernt das Passwort "P" von allen Files, deren Namen mit F beginnen.

FDISK.COM

Nutzbarmachen der Hard-Disk für mehrere der Betriebssysteme PCOS, MS-DOS, CP/M-86 und UCSD oder auch nur eines der Nicht-PCOS-Systeme.

FORMAT:

fdi

Der Benutzer wird durch einen englischen Dialog geführt, der die Aufteilung der Hard-Disk in entsprechende Bereiche ('Partitions') durchführt oder die vorhandene Aufteilung ändert oder die aktuelle Aufteilung bekanntgibt.

Bemerkungen:

- Vor einer Neuaufteilung muß die gesamte Hard-Disk mit **vv** verifiziert und mit **vf** vollständig formatiert werden. Dazu sind ggf. bereits vorhandene PCOS-Files mit **fc** bzw. **sa** auf Diskette(n) zu sichern. Sollten bereits Files eines anderen Betriebssystems auf der Disk sein, sind diese mit dem jeweiligen Befehl zum Kopieren von Files unter dem jeweiligen Betriebssystem zu sichern.
- Vor Aufteilung der Hard-Disk mit **fdi** sollte über **vv** abgefragt werden, daß keine fehlerhaften Blöcke in der Nähe des Beginns einer neuen Partition liegen.
- Die letzte Partition muß für PCOS reserviert bleiben (Verwaltung der fehlerhaften Blöcke).
- Für ein Betriebssystem darf nur jeweils eine Partition existieren. Nach einer Neuverteilung vorhandener Partitions, die ohne Neuformatieren der gesamten HD erfolgen kann, sind die neu entstandenen Partitions mit den jeweiligen Formatierungsbefehlen für das betreffende Betriebssystem neu zu formatieren. Die Sicherung vorhandener Files und Rückübertragungen von Files

Files müssen mit den jeweiligen Befehlen zum Kopieren von Files durchgeführt werden.

- Nach **fdi** muß der PCOS-Befehl **vn** für die Hard-Disk durchgeführt werden.
- Die gesamte Hard-Disk kann mit dem PCOS-Befehl **vv %d** gelöscht werden. Danach muß erneut **vf** auf die HD angewandt werden.
- Ist 'bootable status' auf **n** gesetzt, kann das jeweilige Betriebssystem nur von FD geladen werden, andernfalls wird automatisch von HD geladen. Dies gilt nicht mehr, wenn das alternative Prozessor-Board gewählt wurde und in der FD-Station eine Diskette einliegt, auf der sich ein vom alternativen Prozessor-Board her ladbares Betriebssystem befindet. (Es kann sich dabei um MS-DOS, CP/M-86 oder UCSD handeln). In diesem Fall wird immer von Diskette geladen. Bei Nichtwahl des alternativen Prozessor-Boards wird PCOS immer von HDU geladen (es sei denn, dort ist 'bootable status' für PCOS auf **n** gesetzt). Das Laden von PCOS von Diskette kann durch Gedrückthalten der Taste **F** nach physischem Reset erreicht werden.
- Haben mehrere der Betriebssysteme MS-DOS, CP/M-86 und UCSD den 'bootable status' **y**, wird nach Auswahl des alternativen Prozessor-Boards eine Auswahl freigegeben, welches der mit 'bootable status' **y** versehenen Betriebssysteme geladen werden soll, oder ob nachträglich von Diskette geladen werden soll. Dies gilt nicht, wenn beim Laden bereits eine betreffende Systemdiskette in der FD-Station einliegt.
- PCOS kann nur nach physischem Reset (Anschalten und wieder Ausschalten bzw. Druck in die Aussparung rechts außen an der Basis des M20) geladen werden.
- Für die alternativen Betriebssysteme ist ein physischer Reset zusätzlich durch Betätigen der gelben (COMMAND-)Taste gleichzeitig mit

Reset möglich.

- Ein physischer Reset führt zur Auswahl des gewünschten Prozessor-Boards. Der logische Reset (blaue Taste gleichzeitig mit RESET) führt zum Neuladen von PCOS bzw. dem Auswahlmenü für die alternativen Betriebssysteme - je nach aktuell aktivem Prozessor-Board.
- Nach dem Aufteilen der Hard-Disk sind die jeweiligen Partitions durch Laden des jeweiligen Betriebssystems von Diskette (linke Station) und Anwendung des jeweiligen File-Kopie-Befehles für die zu übertragenden Files zu belegen. (s. Folgeblätter)

Neubelegung von Partitions:

1. Für PCOS

PCOS-Betriebssystem laden; PCOS wie gewünscht konfigurieren und mit **ps 10:** auf Hard-Disk abspeichern. Anschließend die gewünschten Files mit **fc**-Befehl übertragen (s. dazu PCOS-Handbuch, Kapitel 0.2.7.1, ab Punkt 5., sowie Kapitel 0.2.7.2.).

Die PCOS-Partition alleine kann nur mit **vn** gelöscht werden. **vf** formatiert die gesamte HD.

2. Für MS-DOS

MS-DOS-Partition mit

format c: /s

für MS-DOS formatieren, danach mit Hilfe des MS-DOS-Befehls **copy** die gewünschten Files übertragen (Directories beachten!). Beachten Sie, daß bei bereits vorhandenen Files auf der HDU ein File, das erneut kopiert wird, nochmals angelegt wird (ohne Abfrage).

3. Für CP/M-86

CP/M-86-Partition mit

format /H/S

für CP/M-86 formatieren, danach mit Hilfe des CP/M-Befehles **pip** die gewünschten Files übertragen (User beachten!)

4. Für UCSD

Die UCSD-Partition ist automatisch nach **vf** in PCOS formatiert. Das Formatieren der UCSD-Partition alleine ist nicht möglich. Es muß **vf** für die gesamte HD eingesetzt werden.

F(iler)-Ebene der T(ransfer)-Befehl zum Kopieren von Files aufzurufen und als erstes das File HDUBOOT.CODE zu kopieren: Eingabe von

#4:HDUBOOT.CODE~~6,6~~HDUPRTA:

als Antwort auf die Frage

Transfer what file?

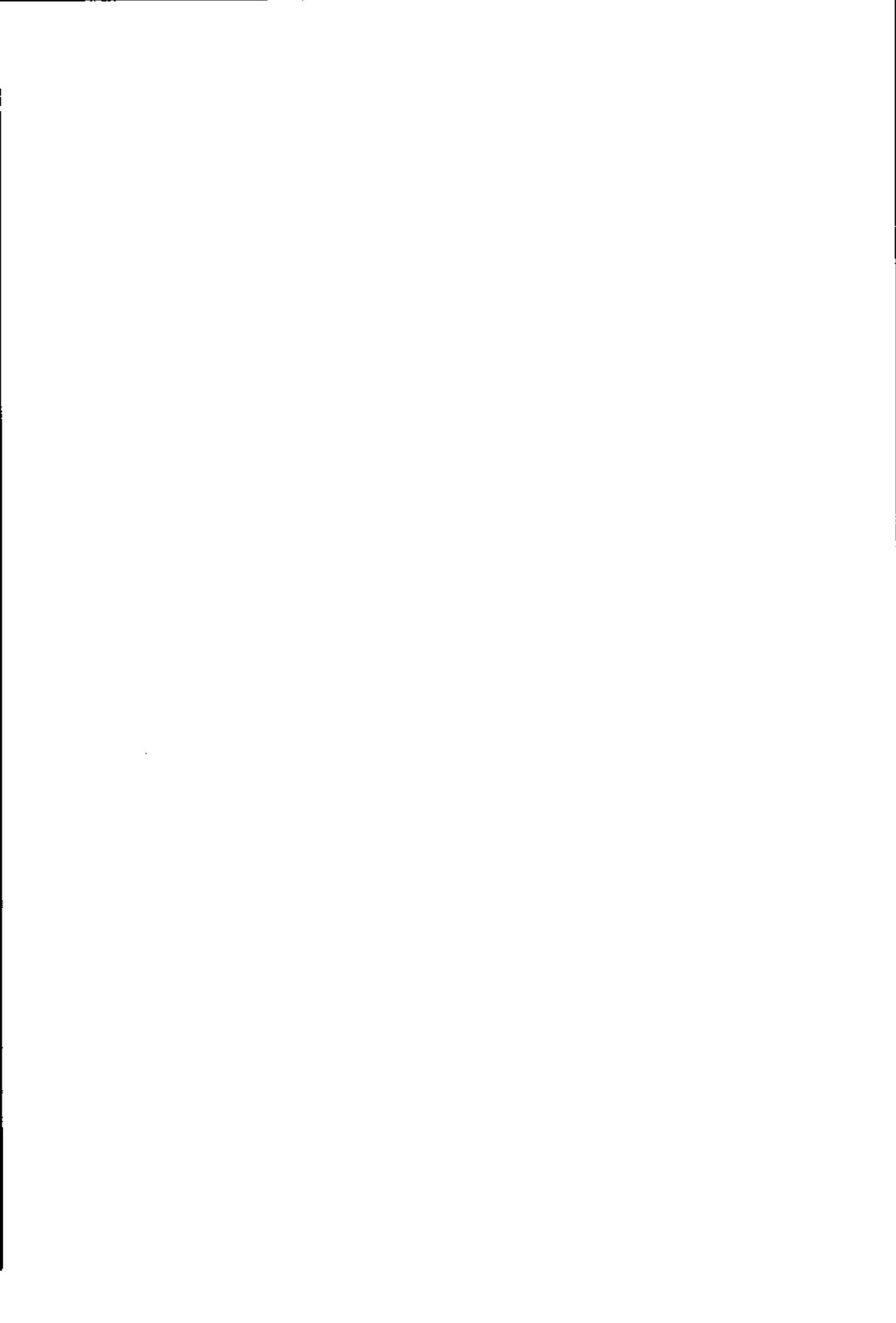
Danach mit

#4:=-~~6,6~~HDUPRTA:\$

die anderen gewünschten Files kopieren. Ist ein File (z.B. HDUBOOT.CODE) bereits vorhanden, sollte die Frage

Remove old xxxx?

mit **n** beantwortet werden. Andernfalls wird das bisherige Zielfile entfernt und das neue File ans Ende des bisherigen Inhaltsverzeichnisses kopiert. HDUBOOT.CODE muß an erster Stelle im Inhaltsverzeichnis stehen.



(file free)

FUNKTION:

Verkürzen aller Files auf einer Diskette oder einer Auswahl davon auf die Anzahl aktuell belegter Sektoren

FORMAT:

```
ff [Disketten-Spezifikation:] [Filename(enauswahl)]
```

WIRKUNG:

Es wird "You may not change disks while FFree in progress. Continue?" gemeldet ("Bitte nicht Disketten wechseln, während das Programm läuft"). Die Eingabe von **n** bewirkt Programmabbruch, **y** die Durchführung.

Bei den durch 'Filename(enauswahl)' ausgewählten Files werden Sektoren im Inhaltsverzeichnis als frei gekennzeichnet, die hinter dem höchsten des Files (Sectors USED) noch für das jeweilige File reserviert sind. Danach wird Sectors ALLOCATED um 1 größer sein als Sectors USED.

Es wird ausgegeben, wieviele Sektoren jedes Files von **ff** betroffen wurden.

BEMERKUNGEN:

- Wird 'Filename(enauswahl)' nicht angegeben, werden alle Files auf Diskette auf aktuelle Länge (in Sektoren) verkürzt.

- **ff** läßt immer mindestens zwei Sektoren als vom File belegt gelten.
- Die angesprochene Diskette darf keinen Schreibschutz haben; die angesprochenen Files dürfen keinen Schreibschutz haben.

BEISPIELE:

ff 1:

Alle Files auf der Diskette in Station 1: werden auf aktuelle Länge (in Sektoren) +1 verkürzt.

ff DISK:DATEN.seq

Auf der Diskette mit dem Namen 'DISK' wird das File 'DATEN.seq' auf die aktuelle Länge (in Sektoren) +1 verkürzt.

Zu **ff**-Befehl

Bei HDU-Version werden Blöcke in der 'Liste der fehlerhaften Blöcke' nicht freigegeben.



FUNKTION:

Löschen eines Files oder einer Auswahl von Files aus dem Inhaltsverzeichnis

FORMAT:

```
fk Filename(enauswahl) [ ,Filename(enauswahl) ] ...
```

WIRKUNG:

Die durch 'Filename(enauswahl)' bezeichneten Files werden aus dem Inhaltsverzeichnis gelöscht und die durch sie belegten Sektoren wieder freigegeben.

BEMERKUNGEN:

- Wird bei 'Filename/enauswahl' nur * angegeben, wird abgefragt, ob wirklich alle Files auf der Diskette gelöscht werden sollen.
- Bei Angabe einer Filenamenauswahl wird einzeln abgefragt, ob das jeweilige File gelöscht werden soll.
- Der BASIC-Befehl KILL hat die gleiche Wirkung wie fk.
- Die angesprochene Diskette darf keinen Schreibschutz haben.
- Keines der durch 'Filename(enauswahl)' angesprochenen Files darf Schreibschutz haben.
- **fk** löscht das File bzw. die Files nur aus dem Inhaltsverzeichnis und gibt die davon belegten Sektoren frei. Der Bereich für die Files selbst wird nicht mit dem Zeichen mit dem ISO-Code Ø vorbelegt. Dies kann bei Programmen, die den CHR\$(Ø) abfragen,

zu Fehlern führen, wenn in Bereichen, die durch **fk** bzw. **KILL** freigegeben wurden, neue Files angelegt oder vorhandene durch Extent-Bildung erweitert wurden. Das Programmpaket ISAM sorgt selbständig für die Vorbelegung der von den Key-Files belegten Sektoren, beachtet aber nicht die Sektoren, die von dem Data-File zu belegen sind!

- **fk** kann durch **rk** wieder aufgehoben werden, wenn an die betreffende Stelle im File noch nichts Neues abgespeichert wurde.
- Die angesprochenen Files dürfen nur das gleiche Password oder gar keines haben.

BEISPIELE:

fk AØ:TEXT.rnd

Das File 'TEXT.rnd' auf der Diskette mit dem Namen 'AØ' wird logisch gelöscht.

fk Ø:*2.prg

Alle Files auf der Diskette in Station Ø, die auf '2.prg' enden, werden von der Diskette gelöscht.

(file list)

FUNKTION:

Ausgabe des Inhalts von einem auf Diskette befindlichen File oder auch der Inhalte einer ganzen Gruppe Files

FORMAT:

```
f1 Filename(enauswahl) [ ,Filename(enauswahl) ] ...
```

WIRKUNG:

Auf der in 'Filename' spezifizierten Diskette (ohne Angabe auf der zuletzt angesprochenen) wird überprüft, ob ein File dieses Namens existiert. Wenn ja, wird sein Inhalt ohne Seitenaufteilung ausgegeben. Ist eine Filenamenauswahl angegeben, werden alle Files hintereinander gelistet.

BEMERKUNGEN:

- Der 'Filename' wird mit ausgegeben.
- Eine Unterbrechung der Ausgabe kann mit **S** erfolgen. Soll die Ausgabe fortgesetzt werden, ist eine beliebige Taste zu drücken.
- Erfolgt die Ausgabe auf Bildschirm oder Drucker, sind die Inhalte von sequentiellen und ASCII-Files "lesbar". Aus anderen Files werden nur die Teile "lesbar" ausgegeben, die darstellbare Zeichen beinhalten. Numerische Werte in Random-Files werden von f1 "nicht lesbar" ausgegeben. Auch aus komprimierten

BASIC-Programmfiles werden nicht alle Informationen "lesbar" ausgegeben.

- Ein mit dem Full-Screen-Editor erfaßtes File kann mit `fl` "lesbar" ausgegeben werden, wenn beim Aufruf des Full-Screen-Editors der Parameter `%c` gesetzt wurde (ASCII-Format).
- Alle aufgeführten Files müssen das gleiche File-Passwort oder gar keines haben.

BEISPIELE:

```
fl 1:A20500 +dl:file
```

gibt den Inhalt des Files "A20500" von der Diskette in Station 1 auf das File "file" aus. Ist "file" auf der Diskette in Station 1 nicht vorhanden, wird es angelegt.

```
fl 1:TEXT1.asc/KW4711 +dprt:,-dcons:
```

gibt den Inhalt des Files "TEXT1.asc" mit dem File-Passwort KW4711 von der Diskette in Station 1 auf dem Drucker aus, jedoch nicht am Bildschirm.

Zu **f1**-Befehl

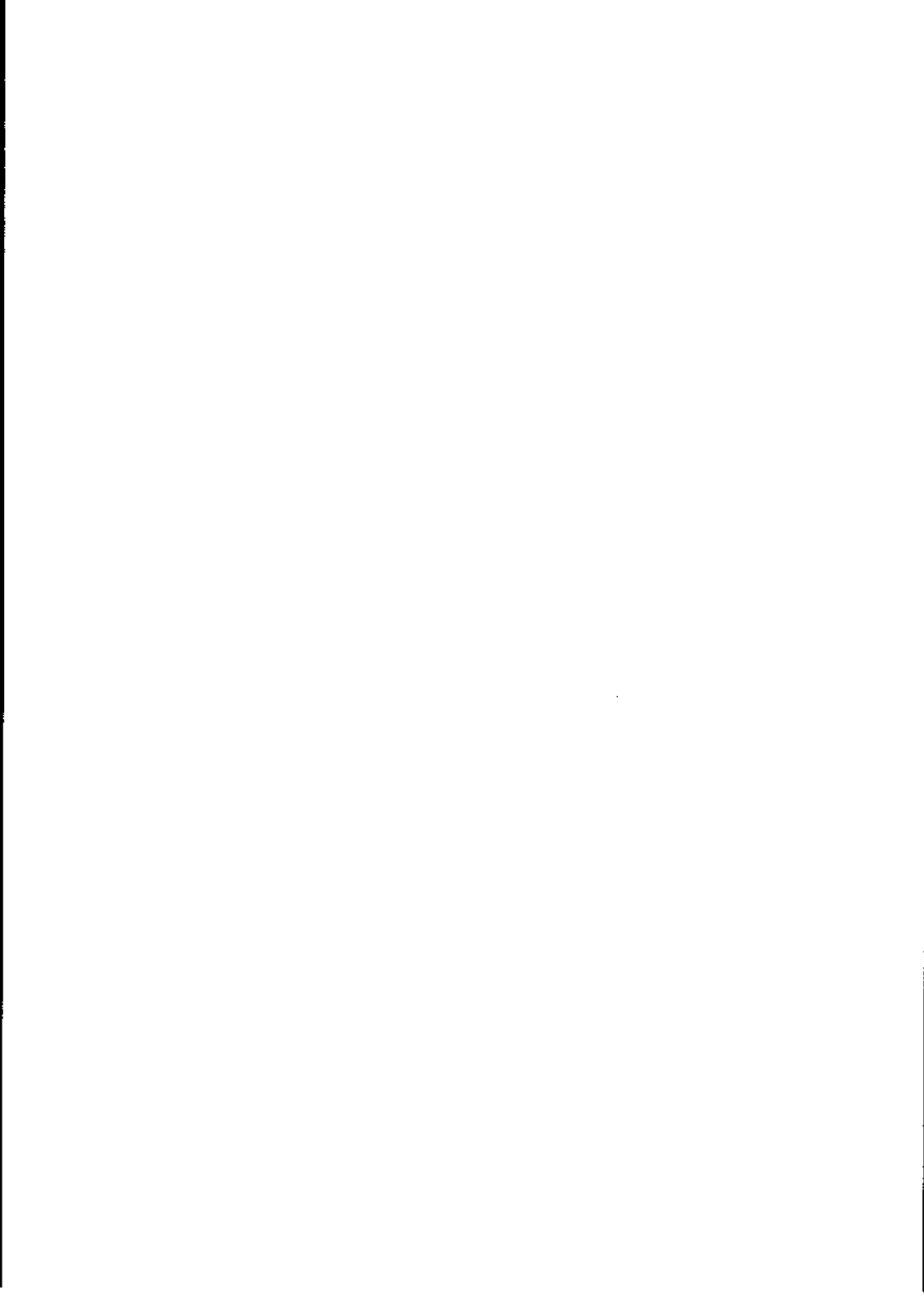
FORMAT:

f1 [**%h**] Filename(nauswahl) [, ...]

%h ergibt Ausgabe in hexadezimaler Form

Beispiel:

f1 %h Ø:ARTIK.rnd Die einzelnen Bytes des Files
ARTIK.rnd auf Station Ø:
werden mit ihrem hexadezimalen
Code ausgegeben.



FUNKTION:

Kopieren eines Files in nur einer Diskettenstation von einer Diskette auf eine andere

FORMAT:

```
fm Originalfilename [ ,Zielfilename ]
```

WIRKUNG:

Es wird gemeldet:

Please put SOURCE disk in drive then press any key (^C to abort):=

"Bitte Diskette mit zu kopierendem File in die Station einlegen, dann beliebige Taste drücken (CONTROL C zum Abbruch)", danach in wievielen Phasen der Vorgang stattfinden wird und

Please put DESTINATION disk in drive, then press any key (^C to abort):=

"Bitte Diskette, auf die das File kopiert werden soll, in die Station einlegen, dann beliebige Taste drücken (CONTROL C zum Abbruch)".

Existiert das File auf der jetzt eingelegten Diskette, erscheint die Meldung

File exists: overwrite, append or quit? (o/a/q)

Bei Eingabe von 'o' wird das Zielfile durch das zu kopierende File ersetzt; bei Eingabe von 'a' wird das zu kopierende File

angehängt; 'q' bewirkt Abbruch. Ist das Zielfile noch nicht vorhanden, wird es angelegt.

BEMERKUNGEN:

- Filenamenauswahl ist nicht möglich.
- Im Gegensatz zu **fc** reduziert **fm** die Länge des Zielfiles nicht auf die Anzahl der 'Sectors USED' +1, sondern behält die angelegte Anzahl Sektoren des Originalfiles bei.
- Ist das File sehr groß, werden die obigen Aufforderungen mehrfach erfolgen.
- Muß das Zielfile angelegt werden, wird es - wenn möglich - in aufeinander folgenden Sektoren angelegt. Die Anzahl der angelegten Sektoren ist ein Vielfaches des Parameters 'Anz.Sektoren' im PCOS-Befehl **ss**. Die somit unnötigerweise belegten Sektoren können mit dem PCOS-Befehl **ff** wieder freigegeben werden.

BEISPIELE:

fm Ø:DATEN.rnd

Das File 'DATEN.rnd' auf der Diskette in Station Ø kann in derselben Station auf eine andere Diskette kopiert werden.

FUNKTION:

Anlegen eines Files (Reservierung von aufeinander folgenden 256-Byte-Sektoren) auf Diskette

FORMAT:

fn Filename , Anzahl Sektoren

Filename: Folge von Zeichen, die den Regeln zur Bildung von Filenamen entspricht

Anzahl Sektoren: Integerzahl;

Minimum: 0

Maximum: Anzahl freier aufeinander folgender Sektoren.

WIRKUNG:

Auf der in 'Filename' spezifizierten Diskette (ohne Spezifikation: der zuletzt angesprochenen) werden soviele 256-Byte-Sektoren für ein File reserviert wie in 'Anzahl Sektoren' angegeben.

Die Sektoren folgen unmittelbar aufeinander.

BEMERKUNGEN:

- Die durch **fn** angesprochene Diskette darf nicht mit Schreibschutz versehen sein.
- Auf der angesprochenen Diskette darf kein File mit dem Namen 'Filename' vorhanden sein.

- Ist die Anzahl der direkt aufeinanderfolgenden freien Sektoren ab dem ersten freien Sektor (!) kleiner als 'Anzahl Sektoren', wird die Meldung **Full extents not allocated:** n gegeben. Das File ist dann nur in der Größe 'Anzahl Sektoren' -n angelegt.
- Sind insgesamt nicht mehr ausreichend Sektoren auf der Diskette frei, wird Error 61 ("disk filled") gemeldet.
- Ist für 'Anzahl Sektoren' 0 angegeben, werden für das File soviel Sektoren reserviert, wie im Parameter 'anzulegende Sektoren' von **ss** festgelegt. Ist für 'Anzahl Sektoren' 1 angegeben, werden dennoch 2 Sektoren reserviert.
- **fn** kann nur zum Neuanlegen von Files verwendet werden. Das Löschen von Files von der Diskette ist durch den PCOS-Befehl **fk** oder den BASIC-Befehl **KILL** vorzunehmen, der alle durch das File aktuell belegten Sektoren wieder freigibt und das File aus dem Inhaltsverzeichnis löscht.
- Durch **fn** wird das File nur in das Inhaltsverzeichnis aufgenommen. Die Sektoren, die dabei für das File reserviert werden, werden nicht mit dem Zeichen mit dem ISO-Code 0 belegt. Dies führt bei Programmen, die den **CHR\$(0)** abfragen, möglicherweise zu Fehlern.
- Files können auch mit der BASIC-Anweisung **OPEN** unter den Zugriffsarten **"R"** oder **"O"** angelegt werden, wenn sie zum Zeitpunkt der **OPEN**-Anweisung noch nicht auf Diskette vorhanden sind.
- Im Gegensatz zur BASIC-Anweisung **OPEN** stellt **fn** sicher, daß
 1. die reservierten Sektoren alle unmittelbar aufeinander folgen

2. genau soviele Sektoren reserviert werden, wie in 'Anzahl Sektoren' festgelegt. (**OPEN** reserviert automatisch soviele Sektoren, wie im PCOS-Befehl **ss** festgelegt bzw. ein Vielfaches davon.)

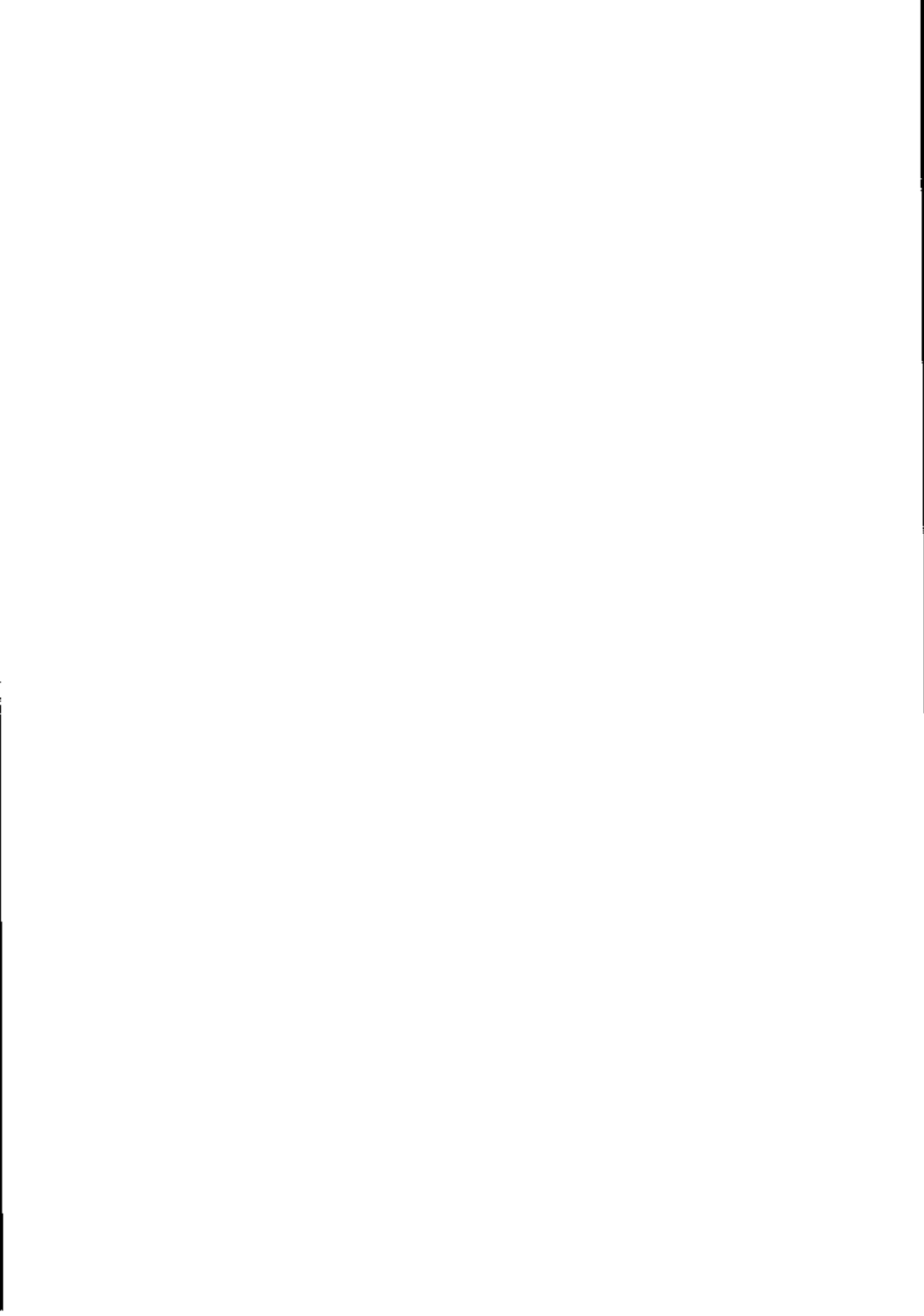
BEISPIELE:

fn 1:FIBU1/FKENNW,15

reserviert auf der Diskette in Station 1 15 aufeinander folgende Sektoren zu je 256 Bytes für das File "FIBU1" mit dem File-Password "FKENNW".

fn meineD/DKW:FILE117.INDSEQ,1000

reserviert auf der Diskette "meineD" mit dem Disketten-Password "DKW" 1000 aufeinander folgende 256-Byte-Sektoren für das File mit dem Namen "FILE117.INDSEQ".



FUNKTION:

Neuvergabe eines File-Password oder Änderung eines bestehenden File-Passwords

FORMAT:

```
fp Filename(enauswahl) [Filename(enauswahl)] ...,neues File-Password
```

WIRKUNG:

Dem mit 'Filename' bezeichneten File bzw. einer ganzen Gruppe von Files mit gemeinsamen Eigenschaften wird das 'neue File-Password' zugeordnet. Hat das File bereits ein Password, ist dieses anzugeben und es wird durch das 'neue File-Password' ersetzt. Werden von einer FilenamenAuswahl mehrere Files angesprochen, wird für jedes File abgefragt, ob das Password vergeben werden soll ("Set Password to 'File-Identifizier'?").

BEMERKUNGEN:

- Die angesprochene Diskette und das angesprochene File dürfen nicht mit Schreibschutz versehen sein.
- Ein bereits vorhandenes File-Password ist beim 'Filenamen' anzugeben. Es braucht zur Änderung nicht vorher mit **fd** entfernt werden.

- Ist das File-Password bekannt, kann es mit **fd** wieder gelöscht werden. PCOS enthält keine Möglichkeit, Passwords sichtbar zu machen. Es ist daher empfehlenswert, organisatorische Maßnahmen zu treffen, die sowohl vor dem Vergessen als auch der unbefugten Verwendung von Passwords schützen.
- Die verdeckte Eingabe von Passwords ist mit **G** möglich. Das Zeichen **G** wird dadurch nicht zum Bestandteil des File-Passwords.
- Bei Ausgabe des ausführlichen Inhaltsverzeichnisses (**vl**) wird bei den Files angegeben, ob sie mit Password versehen sind (**PW**).
- Ein File-Password wird bei Anwendung der PCOS-Befehle **vc**, **vm**, **fc** und **fm** mit übertragen.

BEISPIELE:

fp KONTEN:DEBITOREN,neupass

Auf der Diskette "KONTEN" wird dem File mit dem Namen "DEBITOREN" (bisher ohne Password) das Password "neupass" zugeordnet.

fp Ø:Traegerdaten/altefrage,wiedenn

Auf der in Station Ø einliegenden Diskette wird für das File mit dem Namen "Traegerdaten" das bisherige Password "altefrage" in "wiedenn" geändert.

fp 1:???82*,KENN

Allen Files auf der Diskette in Station 1, deren 4. und 5. Zeichen "82" ist, wird das File-Password "KENN" zugeordnet.

BEFEHL:

FRENAME
(file rename)

rename. cmd



FUNKTION:

Umbenennen eines Files

FORMAT:

fr alter Filename , neue Filespezifikation

WIRKUNG:

Das durch 'alter Filename' bezeichnete File erhält den Namen 'neue Filespezifikation'.

BEMERKUNGEN:

- Bei 'neue Filespezifikation' darf weder File-Password noch Disketten-Spezifikation angegeben werden.
- Ein evtl. vorhandenes File-Password wird nicht verändert.
- **fr** hat die gleiche Wirkung wie der BASIC-Befehl **NAME**.

BEISPIELE:

fr 1:ALT,NEU

Das auf der Diskette in Station 1 befindliche File 'ALT' erhält den Namen 'NEU'

fr Ø:basic.cmd, basic.sav

Der PCOS-Befehl `ba` erhält das Typkennzeichen .sav. Damit wird erreicht, daß nach dem ersten Aufruf des BASIC-Interpreters dieser resident bleibt, d.h., nach Übergang in die PCOS-Ebene nicht erneut von Systemdiskette geladen werden muß.

FUNKTION:

Aufhebung eines File-Schreibschutzes

FORMAT:

```
fu Filename(nauswahl) [,Filename(nauswahl)]...
```

WIRKUNG:

Von dem mit 'Filename' bezeichneten File bzw. allen Files einer Gruppe von Files mit gemeinsamen Eigenschaften wird der Schreibschutz entfernt.

Werden von einer Filenamenauswahl mehrere Files betroffen, wird für jedes File abgefragt, ob der Schreibschutz aufgehoben werden soll ("Clear WP 'File-Identifizier'?").

BEMERKUNGEN:

- Die Vergabe des Schreibschutzes erfolgt mit dem PCOS-Befehl **fw**.
- Ist das File nicht mit Schreibschutz versehen, hat **fu** keine Änderung zur Folge.
- Um File-Schreibschutz aufheben zu können, muß ein evtl. vorhandener Disketten-Schreibschutz durch Entfernen der Aluminiumfolie aufgehoben werden.
- Ein File-Schreibschutz wird bei Anwendung der PCOS-Befehle **vc**, **vm**, **fc** und **fm** mit übertragen.

- Bei Ausgabe des ausführlichen Inhaltsverzeichnisses einer Diskette (PCOS-Befehl **v1**) wird bei den Files angegeben, ob sie mit Schreibschutz versehen sind (WP).

BEISPIELE:

fu FIBUDAT/PW1Ø3:SACHKTN.rnd/FPW29

hebt beim File "SACHKTN.rnd" mit dem File-Password "FPW29" auf der Diskette namens "FIBUDAT" mit dem Disketten-Password "PW1Ø3" einen evtl. vorhandenen Schreibschutz auf.

fu 1:*

von allen Files auf der Diskette in Station 1 wird ein eventueller Schreibschutz entfernt.

FUNKTION:

Vergabe eines File-Schreibschutzes

FORMAT:

```
fw Filename(enauswahl) [ ,Filename(enauswahl) ] ...
```

WIRKUNG:

Das mit 'Filename' bezeichnete File bzw. die durch 'Filenamenauswahl' bestimmten Files erhalten Schreibschutz. Danach sind alle Operationen auf PCOS- und BASIC-Ebene mit diesem File nicht mehr möglich, die eine Veränderung des Files bewirken.

Werden von einer Filenamenauswahl mehrere Files angesprochen, wird für jedes File abgefragt, ob es mit Schreibschutz versehen werden soll ("Set WP 'File-Identifizier'?").

BEMERKUNGEN:

- Ist ein File einer Filenamenauswahl bereits mit Schreibschutz versehen, wird bei der Abfrage sofort "...no change" gemeldet.
- Der Schreibschutz kann mit dem PCOS-Befehl **fu** wieder entfernt werden.
- Ist das File bereits mit Schreibschutz versehen, hat **fw** keine Änderung zur Folge.
- Um die gesamte Diskette mit Schreibschutz zu versehen, ist die Auszustanzung am rechten oberen Rand mit einem Streifen Aluminiumfolie zu überkleben.

- Um File-Schreibschutz vergeben zu können, muß ein evtl. vorhandener Disketten-Schreibschutz durch Entfernen der Aluminiumfolie aufgehoben werden.
- Ein evtl. vorhandenes Disketten- und/oder File-Passwort ist anzugeben.
- Ein File-Schreibschutz wird bei Anwendung der PCOS-Befehle **vc**, **vm**, **fc** und **fm** mit übertragen und bei **f1** mit angezeigt.
- Bei Ausgabe des ausführlichen Inhaltsverzeichnisses einer Diskette (PCOS-Befehl **v1**) wird bei den Files angegeben, ob sie mit Schreibschutz versehen sind (WP).

BEISPIELE:

fw FIBUDAT/PW103:SACHKTEN.rnd/FPW29

versieht das File "SACHKTEN.rnd" mit dem File-Passwort "FPW29" auf der Diskette namens "FIBUDAT" mit dem Disketten-Passwort "PW103" mit Schreibschutz.

fw 1:*.*I*

versieht alle Files auf der Diskette in Station 1, in denen die Zeichenfolge '.I' vorkommt, mit Schreibschutz.

FUNKTION:

Laden des IEEE-488-(IEC-)Pakets zum Arbeiten mit den IEEE-488-BASIC-Anweisungen

FORMAT:

ie

WIRKUNG:

Das IEEE-488-Paket ermöglicht die Verarbeitung der BASIC-Anweisungen zum Arbeiten mit einer IEEE-488-Schnittstelle. Folgende Möglichkeiten sind vorhanden:

1. Kontrolle von Rücksignalen (IFC) und Operationen an der Frontkonsole (REN);
2. Empfang eines Unterbrechungssignals von einer anderen Einheit des Busses, Identifikation der unterbrechenden Einheit durch Abrufbetrieb (serial polling) und Verarbeitung des Unterbrechungssignals;
3. Schreiben von Kontrollbytes (z.B. "Device clear", "Device Trigger" etc.) an andere Einheiten;
4. Adressierung anderer Einheiten, Schreiben und Lesen von Daten anderer Einheiten;
5. Übertragung von Daten über den IEEE-488-Bus (z.B. Zuordnung von Empfänger- bzw. Sendereigenschaft an bestimmte Einheiten).

BEMERKUNGEN:

- Der Parameter **%n** kann nicht gesetzt werden.
- Der Aufruf von **ie** aus der BASIC-Ebene (über **CALL**) kann zu Schwierigkeiten in der Speicherorganisation führen.
- **ie** bleibt nach dem ersten Aufruf bis zum Neuladen des Betriebssystems ständig resident; die Verwendung von **pl** ist somit überflüssig. **pu** kann jedoch nicht eingesetzt werden.
- Genauere Erklärungen zur Programmierung der IEEE-Schnittstelle finden sich im Handbuch "IEC-Schnittstelle, Programmierhandbuch". Dort werden auch die BASIC-Anweisungen beschrieben, die zur Programmierung der IEC-Schnittstelle zur Verfügung stehen.

BEISPIELE:

ie

lädt das IEEE-488-Paket

FUNKTION:

Das Programm wartet auf eine Eingabe über die Tastatur und weist die Eingabe auf eine Variable zu (ähnlich wie die BASIC-Anweisung **INPUT** bzw. **LINE INPUT**). Maximale Stellenanzahl und Typ der Eingabe können festgelegt werden. Die Cursorsteuerung wird aktiviert. Eine Vorgabe im Eingabefeld kann übernommen bzw. korrigiert werden. Die verwendete Abschlußtaste wird ermittelt.

FORMAT:

```
CALL "kb" [(Spalten/Zeile, Zeilen/Window, Eingabeart, $ {Stringvar. }
           {Integervar. },
           Feldlänge, Eingabespalte, Eingabezeile, $Abschlußtaste)]
```

WIRKUNG:

Es wird festgelegt, welche Art von Tastatureingabe zulässig ist ('Eingabeart', s. folgende Tabelle), wieviele Zeichen maximal eingegeben werden dürfen ('Feldlänge') und ab welcher Position, bezogen auf das aktive Window, die Eingabe erfolgen soll ('Eingabespalte', 'Eingabezeile').

Als Ergebnis des Befehls befindet sich in der 'Stringvar.' bzw. 'Integervar.' (abhängig von 'Eingabeart') der eingegebene Wert. Die Integervariable 'Abschlußtaste' hat den Wert 1, 2 oder 3, je nachdem, ob die Eingabe mit CR, S1 oder S2 abgeschlossen wurde.

In 'Spalten/Zeile' und 'Zeilen/Window' ist stets anzugeben, wie das aktive Window, in dem die Eingabe erfolgt, spezifiziert ist. 'Spalten/Zeile' muß die aktuell gültige Anzahl von Textzeichen pro Textzeile im Window enthalten und 'Zeilen/Window' die aktuell gültige Anzahl Textzeilen des aktiven Windows.

Tabelle der Eingabearten:

Eingabeart	zulässige Eingabe
-1 oder 1	String (beliebige Zeichenfolge)
-2 oder 2	numerisch ohne Prüfungen (nur Ziffern, ein Punkt sowie ein Vorzeichen am Anfang möglich)
-3 oder 3	
-4 oder 4	
-5 oder 5	numerisch mit Prüfung (nur Ziffern, kein Punkt und kein Vorzeichen möglich)
-6 oder 6	Integer-Verarbeitung (Ganzzahlen zwischen -32768 und 32767 incl.)

Auch bei Verwendung der Eingabearten 2 bis 5 wird das Ergebnis in einer Stringvariablen abgestellt (Verwendung des Parameters §Stringvar.). Das Ergebnis muß danach ggf. mit der BASIC-Funktion **VAL** in einen numerischen Wert verwandelt werden. Dabei können die Funktionen **CINT**, **CDBL** oder **CSNG** verwendet werden, um das Ergebnis der Funktion **VAL** in den gewünschten numerischen Typ zu verwandeln und danach auf die gewünschte Variable zuzuweisen. Bei Eingabeart 6 ist der Parameter 'Integervar.' zu verwenden.

Soll im Eingabefeld eine Vorgabe abgebildet werden, die übernommen bzw. geändert werden kann, ist die Eingabeart mit negativem Vorzeichen zu setzen und auf 'Stringvar.' bzw. 'Integervar.' vor Aufruf von kb die gewünschte Vorgabe zuzuweisen.

ACHTUNG:

Soll bei Eingabeart 6 auch ein Vorzeichen zulässig sein, muß für dieses ein Zeichen links vor 'Eingabespalte' vorgesehen werden. Bei der Eingabe kann diese Position nur durch die Cursor-Steuertaste SHIFT + ← erreicht werden.

BEMERKUNGEN:

- Die Cursor-Steuerung geschieht nur noch über
 - SHIFT+ → (ein Zeichen nach rechts)
 - SHIFT+ ← (ein Zeichen nach links)
 - SHIFT+ ↓ (das Zeichen an der aktuellen Position löschen)
 - SHIFT+ ↑ (Einfügen an aktueller Position, Aufhebung durch Betätigen einer anderen Cursor-Steuertaste)
 - SHIFT+HOME (Löschen des gesamten Eingabefeldes)
- **H** und **C** sind nicht mehr aktiv.
- Die Eingabe kann sich über mehrere Window-Zeilen erstrecken.
- Für Eingabeart 1 können max. 225 Zeichen eingegeben werden.
- Bei Eingabeart 6 sind bis zu fünf Zeichen zulässig; das Vorzeichen ist ggf. als 6. Zeichen links vor 'Eingabespalte' einzugeben. Es ist bei Eingabeart 6 bei 'Feldlänge' nicht zu berücksichtigen.
- Die Eingabe muß stets mit einer der Abschlußtasten (CR, S1 oder S2) beendet werden.

- Die Variablen 'Stringvar.', 'Integervar.' und 'Abschlußtaste' müssen vor dem ersten Aufruf von kb einen Wert erhalten haben, z.B. durch Wertzuweisung. Die 'Stringvar.' muß dabei einen Inhalt von mindestens soviel Zeichen haben wie 'Feldlänge' beim kb-Aufruf hat (z.B. durch E\$=SPACE\$(255)).
- Es wird 'Illegal function call' (Fehlercode 5) vom BASIC-Interpreter gemeldet, falls 'Stringvariable', 'Integervariable' oder 'Abschlußtaste' noch keinen Wert haben.
- Es wird Error 13 von PCOS gemeldet, falls für 'Integervariable', 'Abschlußtaste' oder 'Stringvariable' ein falscher Variablentyp verwendet wurde, die betreffende Variable aber bereits einen Wert hat.
- Sind Parameter unplausibel (z.B. eine 'Feldlänge' von 6 bei Eingabeart 6 oder wenn 'Feldlänge' größer ist als 'Stringvariable' aktuell an Zeichen enthält), wird von PCOS Error 90 für den betreffenden Parameter gemeldet.
- Sind zuwenig Parameter angegeben oder die Liste der Parameter weggelassen, wird von kb eine Meldung ausgegeben, die das richtige Format des Befehls beschreibt.

Beispiel:

```

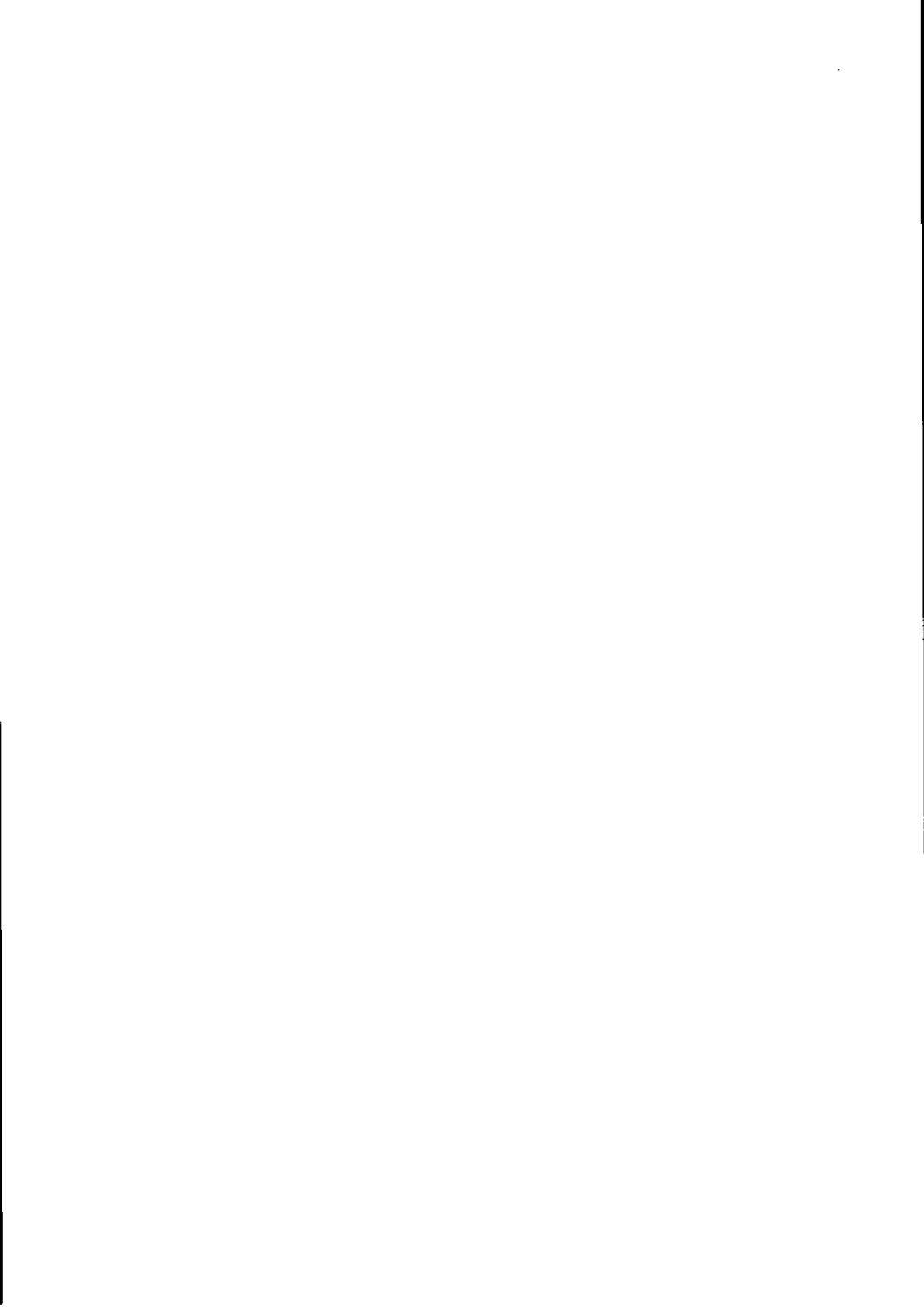
OK
CALL "KB"
V3 dim window <spalten>.<zeilen>,<mode(1,2,3,4,5,6)>,
<variable>,<anz. zeichen>,cursor <spalte>,
cursor <zeile>,<returntaste>
OK

```

BEISPIELE:

```
720 CLEAR:CALL "ss %n,,,,,0"  
730 OH0%=-1:T#=SPACE$(255)  
732 CURSOR(1,5):PRINT "Texteingabe ("&SPACE$(25)+")"  
733 ' ( ) = Feldbegrenzung  
734 G/L: "kb"(64,16,1,67%,25,14 5,50H0%)  
736 EINGABE#=T# ON OH0% GAY0 1001,1101,1201
```

```
10 CLEAR:CALL "ss %n,,,,,1":CALL "bl %n kb"  
20 OH0%=-1:T#=SPACE$(255) 'Anfangswert  
210 FLZ=9:T#=SPACE$(FLZ):INPUT "U# = ";U#  
215 IF LEN(MID$(STR$(U#),2+(U#(0))))>9 THEN 210  
216 'maximal 9 Stellen Vorgabe (+ zählt nicht)  
220 CURSOR(21,5):PRINT "."&SPACE$(FLZ)+":";  
230 LSET T#=MID$(STR$(U#),2*(U#(0))  
240 'linksbündiges Absetzen der Vorgabe U#  
250 CALL "kb"(80,25 -2,6T# FLZ,22,5,60H0%)  
255 Z:=FRE("B"):'Garbage collection  
260 RANDOMFILE_FELD#=#KD$(CDBL(VAL(T#)))  
270 PRINT-PRINT T#:PRINT VAL(T#)
```



FUNKTION:

Darstellung von Zeichen(folgen) in vergrößerter Schrift (Zeichen-Plotten) im aktiven Window

FORMAT:

```
la 'Text' [,X-Position] [,Y-Position] [,Vergrößerung]
    [,Zeichenrichtung] [,Farbcode]
```

WIRKUNG:

Der 'Text' wird im aktiven Window ab dem durch 'X-Position' und 'Y-Position' definierten Elementarpunkt ausgegeben, und zwar in der durch 'Vergrößerung' definierten Größe und in der durch 'Zeichenrichtung' festgelegten Richtung.

Parameter:

Text: aus druckbaren Zeichen bestehend.

Der Text ist in ' ' eingeschlossen anzugeben.
- Enthält der Text ein ', muß 'Text' in " " angegeben werden. Eine Kombination von ' und " in einem Text ist nur über mehrere la-Befehle möglich.

Länge: 1 bis zu soviel Zeichen, daß die Window-Ränder nicht überschritten werden.

X-Position: Nr. des Elementarpunktes im aktiven Window in X-Richtung, in dem der linke Rand des ersten 'Text'-Zeichens beginnt.
Wert: 0 bis Window-Breite in Elementarpunkten
-1
Default-Wert: 0

Y-Position: Nr. des Elementarpunktes im aktiven Window in Y-Richtung, in dem der untere Rand des ersten 'Text'-Zeichen beginnt.
Wert: 0 bis Window-Höhe in Elementarpunkten
-1
Default-Wert: 0

Vergrößerung: Wert: 1 bis 16
1: normale Schriftgröße (allerdings vom normalen Text leicht abweichendes Schriftbild)
Default-Wert: 1

Zeichenrichtung: Wert: 0 bis 2
0: parallel zur X-Achse
1: parallel zur Y-Achse, von unten nach oben
2: parallel zur Y-Achse, von oben nach unten
Default-Wert: 0

Farbcode: Wert: 0 bis 7 für 8-Farben-Schirm,
0 bis 3 für 4-Farben-Schirm,
0 oder 1 für Schwarz/Weiß-Schirm
Default-Wert: aktuelle Vordergrundfarbe

Zeichengröße in Elementarpunkten:

Zeichenrichtung	Vergrößerung: 1	Größe des Elementarpunkt-Blocks/Zeichen			
∅	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>a</td><td>b</td><td>c</td></tr></table>	a	b	c	8 x 10
a	b	c			
1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>c</td></tr><tr><td>b</td></tr><tr><td>a</td></tr></table>	c	b	a	10 x 8
c					
b					
a					
2	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>b</td></tr><tr><td>a</td></tr><tr><td>c</td></tr></table>	b	a	c	10 x 8
b					
a					
c					

BEMERKUNGEN:

- Windows können nur in der BASIC-Ebene angelegt werden (BASIC-Funktion **WINDOW**). Soll **1a** in der BASIC-Ebene angewendet werden (**CALL**-Anweisung) und in einem bestimmten Window ausgegeben werden, ist das betreffende Window vorher mit der **WINDOW%**-Anweisung zu aktivieren. Die 'Positionen' beziehen sich dann auf das aktive Window; d.h., die Anzahl zur Verfügung stehender 'Positionen' verringert sich entsprechend.
- Der unterste linke Rand hat die 'X-Position' ∅ und die 'Y-Position' ∅ (vgl. zur Aufteilung des Bildschirms BASIC-Handbuch, Kapitel 4.3.18.).
- Liegen Punkte der zu zeichnenden Zeichen außerhalb des Windows, wird 'Text' entweder nur teilweise oder gar nicht ausgegeben. Es wird aber kein Fehler gemeldet.

- Wurde **la** mit **pl** resident gemacht und keine Parameter angegeben, werden für 'X-Position' und 'Y-Position' beim ersten Aufruf von **la** die Default-Werte angesetzt.
- Ist vorher ein **la**-Aufruf erfolgt und 'X-Position' bzw. 'Y-Position' nicht angegeben, wird der Punkt als Ausgangspunkt angenommen, der sich als Endpunkt des letzten **la**-Aufrufs ergab.
- Ist im Window eine Anwender-Skalierung vorgenommen worden (BASIC-Anweisung **SCALE**), können die den Anwenderkoordinaten zugeordneten Elementarpunkt-Nummern über die BASIC-Funktionen **SCALEX** bzw. **SCALEY** ermittelt werden.
- Am 8-Farben-Bildschirm gilt für 'Farbcode' folgende Tabelle:

0	schwarz
1	grün
2	blau
3	cyan (türkis)
4	rot
5	gelb
6	magenta (purpur)
7	weiß

Am 4-Farben-Bildschirm gelten die aus der BASIC-Anweisung **COLOR=** abzuleitenden Farbindices (vgl. BASIC-Handbuch, Kapitel 4.3.18.3 und Anweisung **COLOR=**). Für den Schwarz/Weiß-Schirm und den 8-Farben-Schirm gilt der 'Farbcode'.

- Es werden im Text die Zeichen ausgegeben, die im aktuell gültigen Bildschirm-Zeichensatz definiert sind (z.B. auch nach einer Änderung über **rf**, **ed** und **wf**).

BEISPIELE:

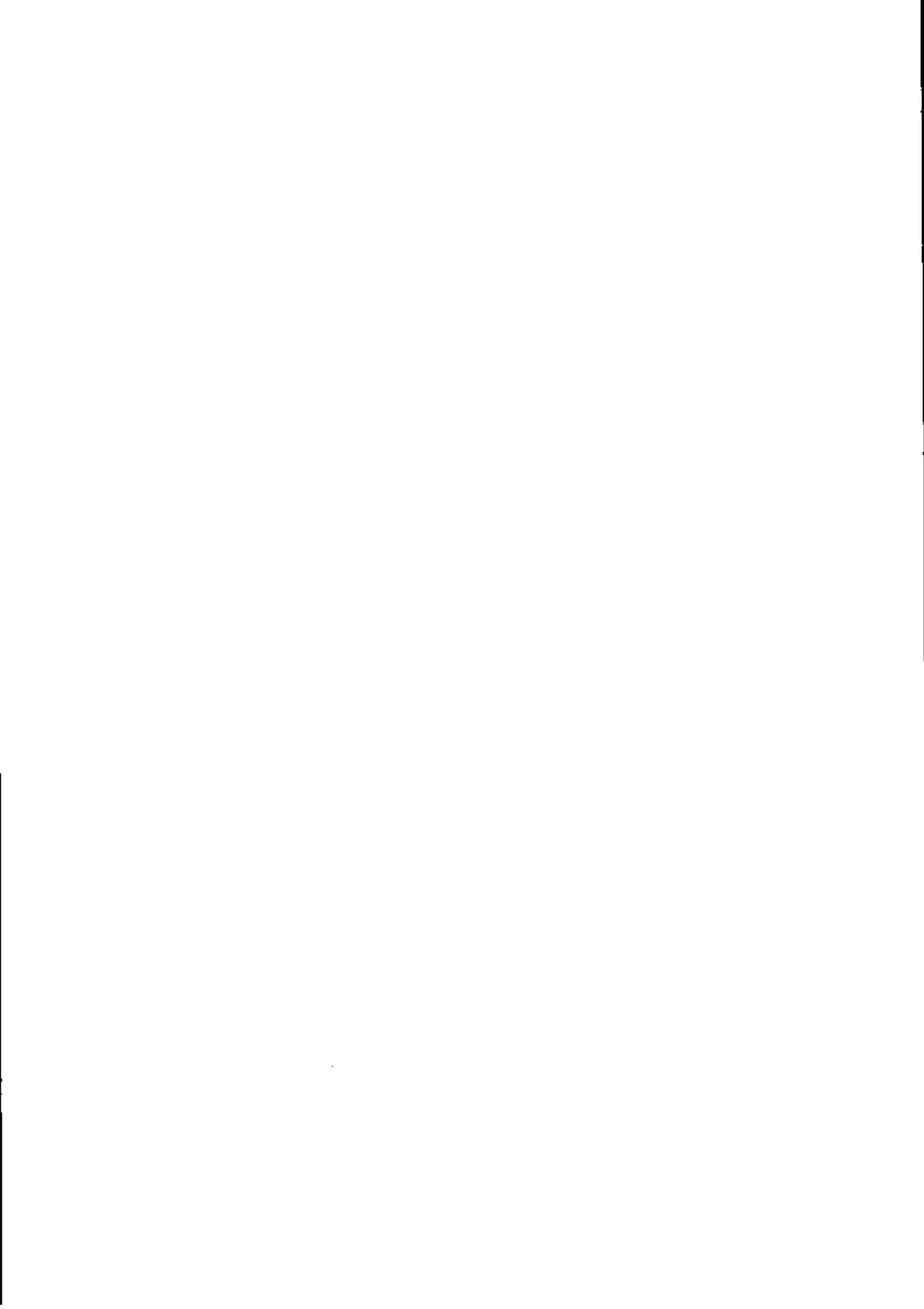
```
1a 'ZEIT',150,100,5,0
```

In der PCOS-Ebene wird der Text ZEIT etwa in der Mitte des Bildschirms (nur 1 Window, 64x60-Format) in 5-facher Vergrößerung und horizontaler Richtung ausgegeben.

```
84 SCALE 0,1,0,1
```

```
85 CALL "1a"("M 20",SCALEX(.25),SCALEY(.75),2,0)
```

In der BASIC-Ebene wird der String M 20 in zweifacher Vergrößerung in horizontaler Richtung ausgegeben. Der Text beginnt im zweiten Viertel der X-Anwender-Skalierung und im letzten Viertel der Y-Anwender-Skalierung.



BEFEHL:**LSCREEN**
(line screen)**lscreen. cmd**FUNKTION:

Ausgabe aller Texte in einem bestimmten Window des Bildschirms auf dem Drucker

FORMAT:

Is [Window-Nr.]

Window-Nr.: 1 bis Anzahl angelegter Windows

WIRKUNG:

Ist der Parameter 'Window-Nr.' nicht angegeben, wird der Text auf dem Drucker ausgegeben, der sich im aktiven Window des Bildschirms befindet. Blanks und Leerzeilen werden berücksichtigt. Ist eine 'Window-Nr.' spezifiziert, wird der Text im betreffenden Window ausgegeben.

BEMERKUNGEN:

- Alle Zeichen, die im Zeichensatz des Druckers vorhanden sind, werden ausgegeben. Graphische Darstellungen (wie über BASIC-Anweisungen **CIRCLE**, **LINE**, **PSET** u.ä. oder über **la** gewonnen) werden nicht berücksichtigt.
- Für die Zeichen, die über Veränderung des Bildschirm-Zeichensatzes erstellt wurden (**rf**, Full-Screen-Editor und **wf**), werden diejenigen Zeichen ausgegeben, die am Drucker dem ISO-Code entsprechen, der der Taste(nkombination) zugeordnet ist.

- Liegt eine graphische Darstellung im Window so, daß mindestens ein Elementarpunkt in der Punktmatrix eines Textzeichens liegt, wird das betroffene Zeichen nicht ausgegeben.
- Ist der Schirm in mehrere Windows unterteilt, muß der gesamte Text im Bildschirm Window für Window ausgegeben werden.
- Die Ausgabe des Textes in einem Window beginnt immer am linken Rand des Druckers, unabhängig von der Lage des Windows.
- Am Farbbildschirm erfolgt die Textausgabe nur dann, wenn der Farbcode der Vordergrundfarbe ungeradzahlig und der der Hintergrundfarbe geradzahlig ist bzw. umgekehrt. Nur die Texte in Vordergrundfarbe werden ausgegeben.
- Am Schwarz-Weiß-Bildschirm werden Texte in normaler und in Reverse-Darstellung ausgegeben, auch gemischt.

BEISPIELE:

!s gibt den gesamten Text im Bildschirm auf dem Drucker aus (PCOS-Ebene, keine Window-Einteilung möglich).

CALL "!s 2" gibt den gesamten Textinhalt in Window 2 auf dem Drucker aus.

(line terminator)

FUNKTION:

Liefert für ein BASIC-Programm in einer Integer-Variablen den Wert 0, 1 oder 2 in Abhängigkeit von der zuletzt betätigten Eingabe-Abschlußtaste

FORMAT: (nur BASIC)

`CALL "lt"($Integer-Variable)`

WIRKUNG:

In der Variablen 'Integer-Variable' steht der Wert

- 0 wenn zuletzt die ↵ -Taste betätigt wurde
- 1 wenn zuletzt die S1 - Taste betätigt wurde
- 2 wenn zuletzt die S2 - Taste betätigt wurde

Bei DES V und DES VI gelten nach s1 16 für die Tasten S3, S5 und S6 noch folgende Werte:

- 3 wenn zuletzt die S3 - Taste betätigt wurde
- 4 wenn zuletzt die S5 - Taste betätigt wurde
- 5 wenn zuletzt die S6 - Taste betätigt wurde.

BEMERKUNGEN:

- lt ist stets resident.
- lt kann nur in der BASIC-Ebene mittels der Anweisung CALL aufgerufen werden.

- Die 'Integer-Variable' muß mit dem Präfix \$ (ohne Blank) eingeleitet sein, um anzuzeigen, daß es sich um einen von PCOS an den BASIC-Interpreter zu übergebenden Wert handelt.
- Die 'Integer-Variable' muß vor dem Aufruf eine Wertzuweisung erhalten haben (z.B. durch T%=-1 am Anfang des Programms).
- Anstelle der S-Tasten kann auch **M** als Abschlußtaste verwendet werden. Auch sie erzeugt das Zeichen mit dem ISO-Code 13. Durch **M** wird jedoch der Inhalt des It-Puffers nicht verändert.
- Werden 2 Eingabeabschlußtasten direkt hintereinander gedrückt (ohne Eingabe eines anderen Zeichens dazwischen), kann It einen falschen Wert liefern.

BEISPIELE:

```
5 TASTE% = -1
```

```
60 LINE INPUT "Text";T$
```

```
70 CALL "It"($TASTE%):ON TASTE% +1 GOTO 1000, 2000, 3000
```

MSCOPY.CMD

Kopieren von Files von PCOS-Diskette nach MS-DOS-Diskette bzw. umgekehrt

FORMAT:

msc Name des zu kop. Files , { Name des Zielfiles
Name der Zielstation: }

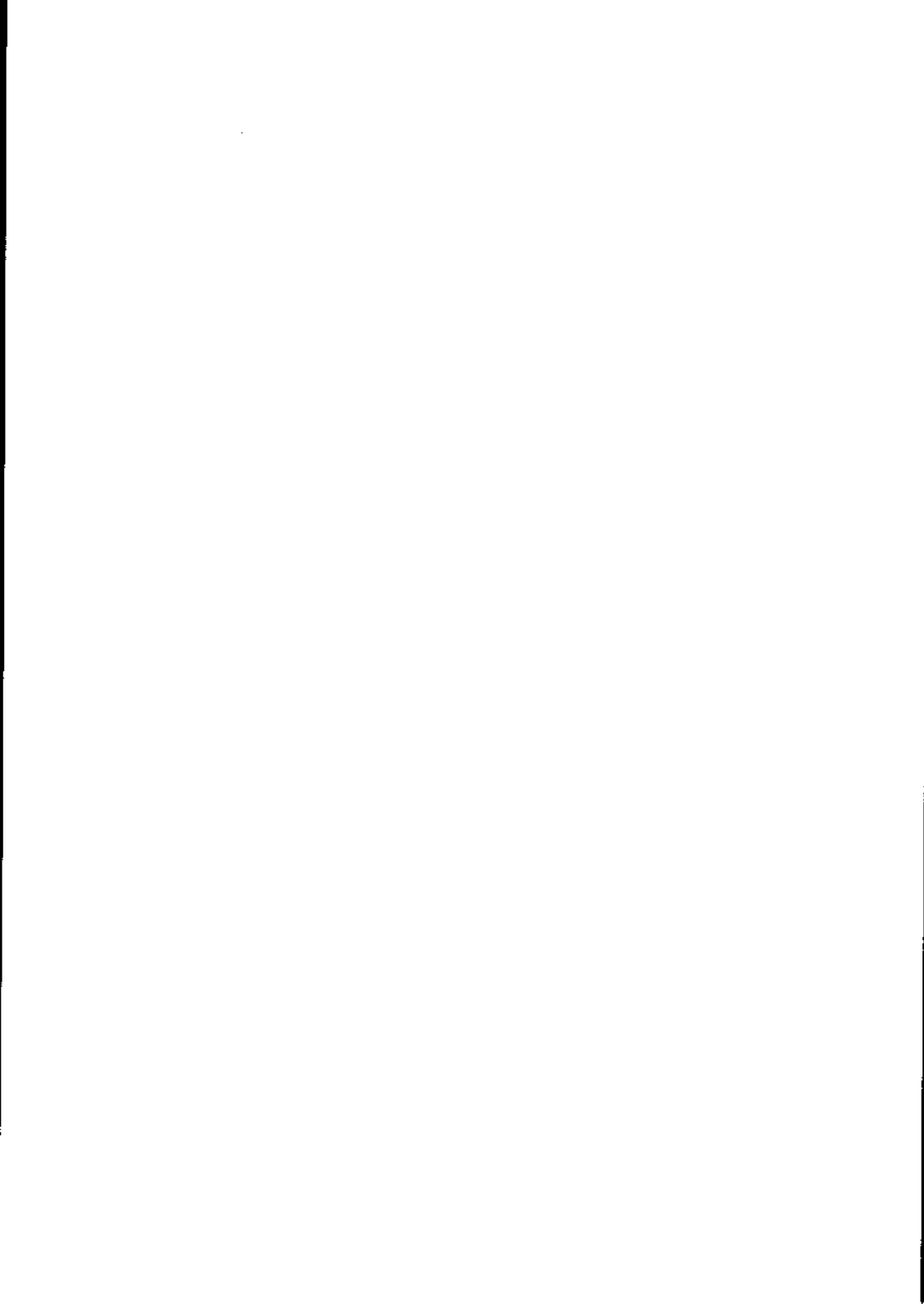
Bemerkungen:

- Die MS-DOS-Diskette muß unter MS-DOS Release 2.0 mit Parameter /8 bei **format** oder unter MS-DOS Release 1.x formatiert worden sein.
- Es existiert keine Kopiermöglichkeit von MS-DOS-Bereich der HD nach PCOS-Diskette.
- Nur Files in der Root-Directory können kopiert werden.
- Pfade können nicht definiert werden.
- Keine Filenamenauswahl möglich.
- Das Zielfile darf noch nicht existieren.

Beispiele:

msc A:DATEN.SEQ, 10: Kopiert das File
DATEN.SEQ von der
MS-DOS-Diskette
in der linken Station
in den PCOS-Bereich
der HDU

msc 1:PROG.asc, A:PROG.ASC Kopiert das File
PROG.asc auf der
PCOS-Diskette in
der linken FD-Station
in das File PROG.ASC
auf der MS-DOS-Diskette
in der rechten
Station



MSDIR.CMD

Ausgabe des Inhaltsverzeichnisses einer MS-DOS-Diskette unter PCOS

FORMAT:

msd Stationsname

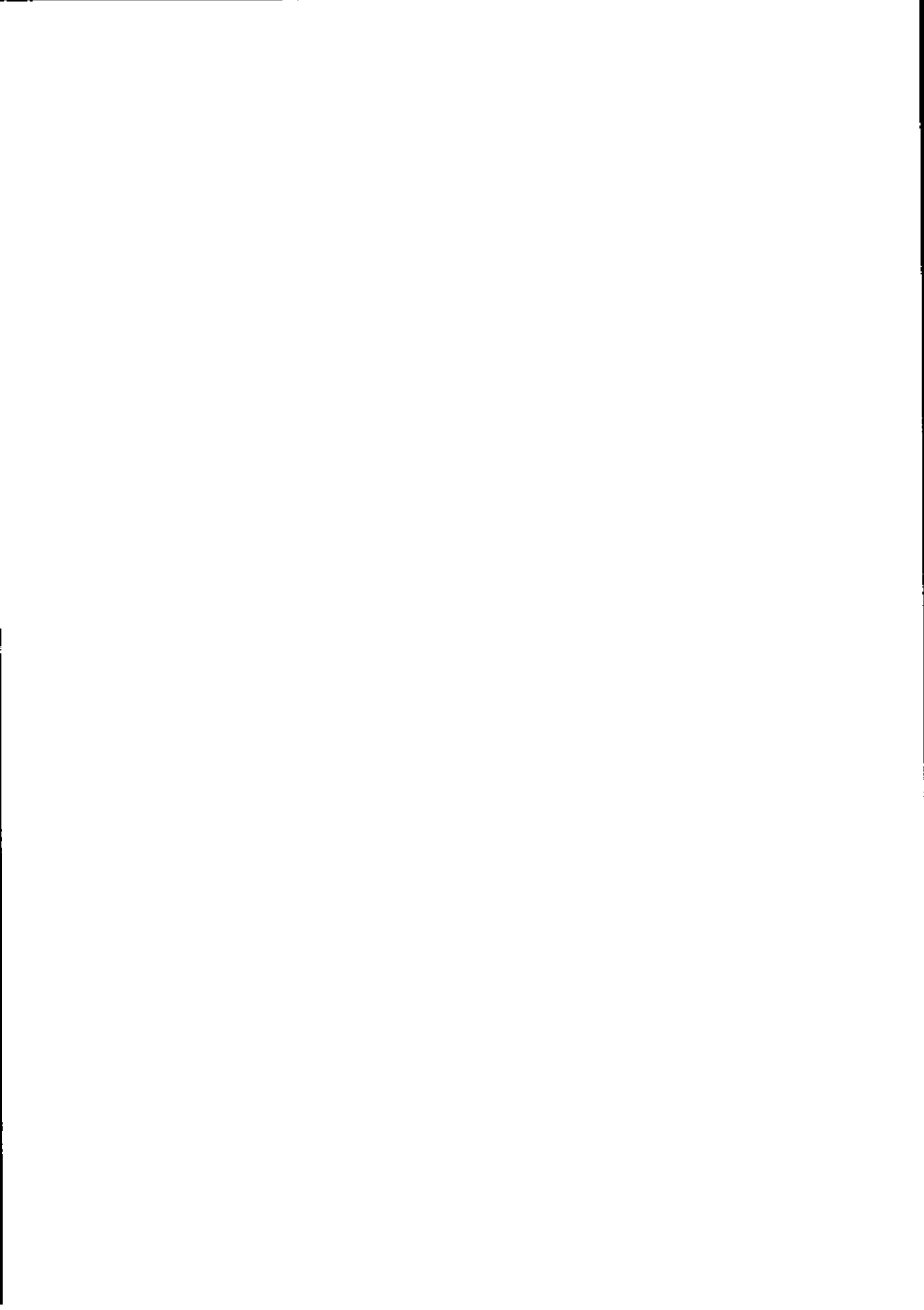
Stationsname: **A:** oder **B:**

Bemerkungen:

- Die MS-DOS-Diskette muß unter MS-DOS Release 2.0 mit Parameter **/8** bei **format** oder unter MS-DOS Release 1.x formatiert worden sein.
- Es existiert keine Ausgabemöglichkeit für den MS-DOS-Bereich der HDU bzw. bei Ein-Disketten-Versionen.
- Nur die Root-Directory kann ausgegeben werden.
- Pfade können nicht definiert werden.

Beispiel:

msd **B:** gibt das Inhaltsverzeichnis der MS-DOS-Diskette in der linken Station aus.





(programmable key)

FUNKTION:

Zuordnung eines Zeichens oder einer Zeichenfolge auf eine Taste(nkombination) oder Herstellung der ursprünglichen Tastenbelegung

FORMAT:

$\text{pk } \left\{ \begin{array}{l} \text{Original-ISO-Code} \\ \text{'Zeichen'} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{neuer ISO-Code} \\ \text{'Zeichen(kette)'} \end{array} \right\} \right] \dots$

WIRKUNG:

Der Taste mit dem Tastencode 'Original-ISO-Code' bzw. der Ursprungsbelegung mit dem Zeichen 'Zeichen' wird diejenige Zeichenfolge zugeordnet, die in der Liste aus 'Zeichen(kette)' und/oder 'neue ISO-Code' beschrieben ist.

Wird nur 'Original-ISO-Code' als Parameter angegeben, wird die ursprüngliche Tastenbelegung wiederhergestellt.

BEMERKUNGEN:

- Die 'Zeichen(kette)' ist in ' ' einzuschließen.
- Enthält die Zeichen(kette) ein ', ist sie in " " einzuschließen. Die Kombination von ' und " ist nur möglich, indem eines der Zeichen mit Hilfe von 'neuer ISO-Code' definiert wird.

- Der 'neue ISO-Code' bzw. der 'Original-ISO-Code' kann dezimal (z.B. 48 für das Zeichen Ø) oder hexadezimal angegeben werden. Im letzten Falle ist vor dem Hexa-Code das Zeichen & zu setzen (z.B. &30 für das Zeichen Ø).
- Die Belegung einer mit **pk** belegten Taste kann nur geändert werden, wenn zuerst ihre ursprüngliche Belegung wiederhergestellt wurde (sonst Error 106).
- Soll die ursprüngliche Belegung einer Taste wiederhergestellt werden, ist nach **pk** nur der Parameter 'Original-ISO-Code' anzugeben. Dieser Code ist derjenige, der ursprünglich für die betreffende Taste galt.
- Die Original-ISO-Codes der Tasten sind dem System stets bekannt.

In der Tabelle im Arbeitsspeicher wird verwaltet

- a) welchen Tastencodes (vgl. **ck** und Kapitel 3.1) welche Original-ISO-Codes zugeordnet sind (abhängig von der Tastatur; vgl. Kapitel 3.2)
- b) welchen Tastencodes (vgl. **ck** und Kapitel 3.2) welche besonderen Funktionen zugeordnet sind.

Bei einer Tastenbelegung über **pk** wird mit einem Verweis festgehalten, welches Zeichen bzw. welche Zeichenfolge anstelle des ISO-Zeichens erzeugt werden soll, das in der obigen Tabelle festgehalten ist.

- Wird versucht, bei einer Taste nur den Parameter 'Original-ISO-CODE' anzugeben (Herstellung der ursprünglichen Tastenbelegung), ohne daß dafür vorher eine Neubelegung erfolgte, wird Error 90 gemeldet.

- Soll erreicht werden, daß die durch eine Tastenbelegung erzeugte, angegebene Zeichenkette sofort abgearbeitet wird, müssen am Ende der Zeichenkette die ISO-Codes 13 (dezimal) und 10 (dezimal) festgelegt sein.
- Sind mehrere Tasten mit dem gleichen 'Original-ISO-Code' vorbelegt, wird sich eine Änderung der Belegung mit **pk** auf alle diese Tasten gleichzeitig auswirken.

BEISPIELE:

pk 132, 'EDIT '

Beim Drücken der Taste **e** gleichzeitig mit der COMMAND-Taste wird der Text EDIT erzeugt.

pk &30, 'NULL'

Beim Drücken der Taste **0** (egal, ob im numerischen oder alphanumerischen Tastaturblock), wird NULL ausgegeben. Beim Drücken der Taste **00** wird NULLNULL ausgegeben.

pk 48

stellt die ursprüngliche Belegung der Tasten **0** und **00** wieder her, falls diese vorher neu belegt worden sind.

pk &82, 'vc 0: 1:', 13, 10, 'y', 13, 10

Bei Betätigen der Taste **c** i.V. mit COMMAND wird der PCOS-Befehl **vc** gesucht und abgearbeitet. Die dabei üblichen Meldungen werden gegeben und auf die Frage

Warning- vcopy deletes files. Copy disk? (y/n)

sofort die Antwort **y** vorgegeben und abgearbeitet. Der Kopier-
vorgang findet somit automatisch statt.

CALL "pk 237, 'FILES ',34,'1:',34,13,10" (BASIC-Ebene)

Wird die Tastenkombination **COMMAND 1** (1 im alphanumerischen
Tastenblock) in der BASIC-Ebene gedrückt, wird sofort das kurze
Inhaltsverzeichnis der Diskette in Station 1 ausgegeben.

Zu **pk**-Befehl

Neues FORMAT:

pk [{ Original-ISO-Code } [{ Neuer ISO-Code }]]
['Zeichen' [, 'Zeichen(kette)'] ...]
%c

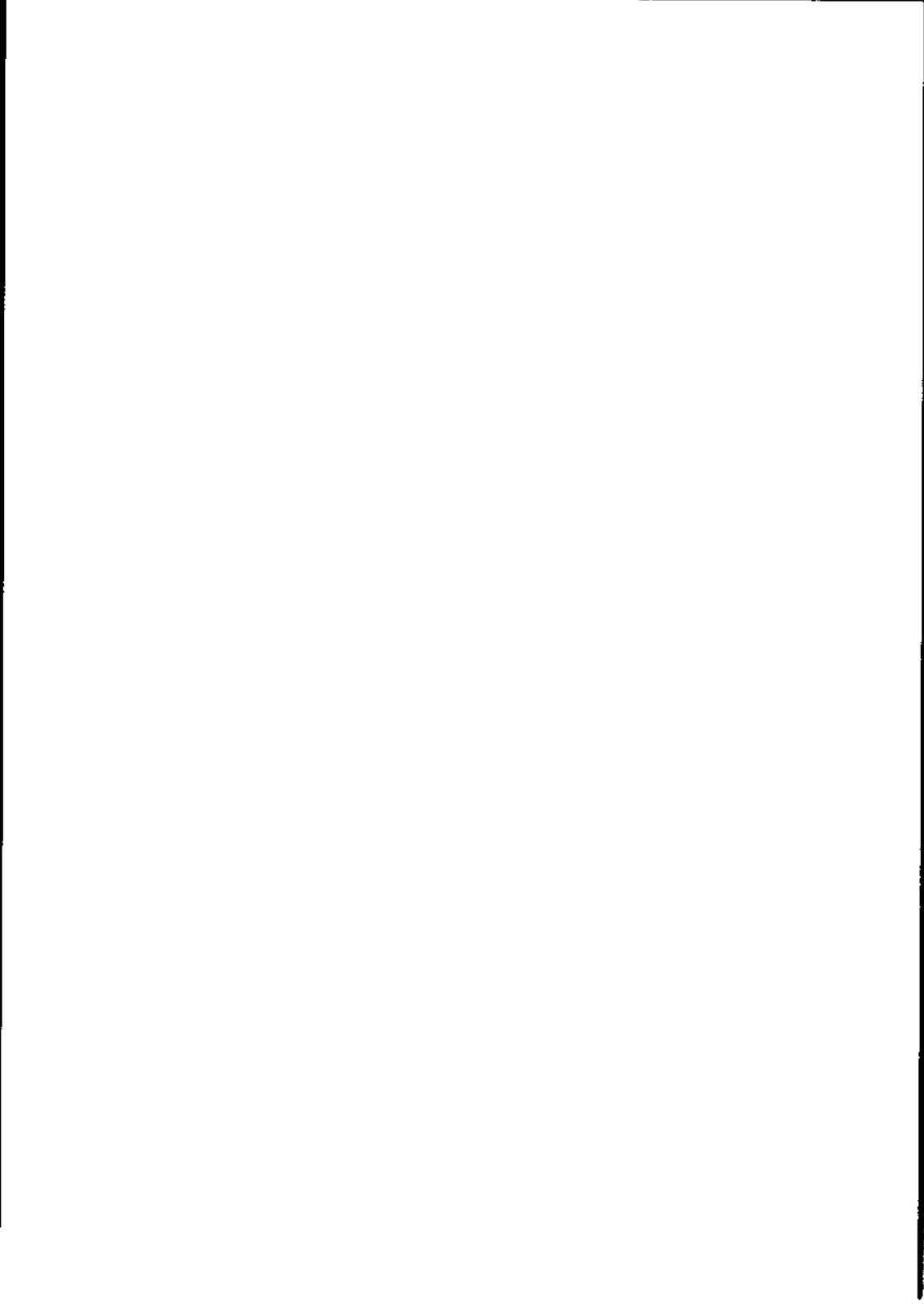
%c Alle mit vorherigen **pk**-Befehlen zugeordneten, vom Standard abweichenden Zeichen(folgen) werden aufgehoben.

Bei Weglassen aller Parameter werden die Belegungen der Tasten angezeigt, die von der Standardbelegung abweichen.

Beispiele:

pk %c Die Standardbelegung der Tastatur ist wieder gültig (**ck**-Definitionen bleiben bestehen).

pk Die von der Standard-Tastenbelegung abweichenden Tastenbelegungen werden ausgegeben.



FUNKTION:

Laden eines nicht-residenten PCOS-Befehls von Systemdiskette in den Arbeitsspeicher (der PCOS-Befehl wird resident gemacht)

FORMAT:

```
p1 PCOS-Befehl [ ,PCOS-Befehl ] ...
```

PCOS-Befehl: Name (Schlüsselwort) eines PCOS-Befehls auf Diskette; ggf. mit vorangestellter Disketten-Spezifikation

WIRKUNG:

Die angegebenen 'PCOS-Befehle' werden auf Diskette gesucht. Die Suche wird auf der Diskette begonnen, auf die zuletzt zugegriffen worden ist. Die Station ist im Bildschirm vor > angezeigt. Wird der PCOS-Befehle gefunden, wird er in den Arbeitsspeicher geladen und bleibt von da an bis zum Neuladen des Systems im Arbeitsspeicher resident.

BEMERKUNGEN:

- p1 ist stets resident.
- p1 verringert den im Arbeitsspeicher zur Verfügung stehenden Platz, allerdings nicht den Platz, der für BASIC bereitsteht.
- Ist der zu ladende PCOS-Befehl bereits resident, wird er nicht nochmals geladen, es erfolgt aber auch keine Fehlermeldung.

- Filenamenauswahl ist nicht möglich.
- Sind PCOS-Befehle durch **pl** resident gemacht, wird ihre Abarbeitung beschleunigt, da das Suchen auf Diskette und das Laden in den Arbeitsspeicher entfällt. Nach **pl** können die betreffenden resident gemachten PCOS-Befehle aufgerufen werden, ohne daß die Systemdiskette beim Aufruf einliegen muß.
- Wird nach **pl** Error 7 gemeldet, (z.B. wenn ein Übergang von der PCOS- in die BASIC-Ebene mit **ba** angefordert wird), kann dieser durch Verminderung der 'log.Arbeitsspeichergröße' (mit PCOS-Befehl **sb**) behoben werden.
- Bei Durchführung von **pl** wird gemeldet, welcher PCOS-Befehl ab welcher Speicheradresse geladen wurde und wieviele Bytes er im Arbeitsspeicher belegt.
- Hat ein PCOS-Befehl das Typkennzeichen .sav, reicht bereits der erste Aufruf, um ihn resident zu machen.
- Ein resident gemachter Befehl kann mit **pu** wieder aus dem Arbeitsspeicher entfernt werden; Ausnahmen: **ci**, **rs**, **ie**, **vm**, **ep**.
- Die grundsätzlich residenten PCOS-Befehle **pl**, **pu** und **lt** können nicht aus dem Arbeitsspeicher entfernt werden.

BEISPIELE:

pl vc Der PCOS-Befehl **vc** wird resident gemacht.

pl %n pk Der PCOS-Befehl **pk** wird resident gemacht. Es erfolgt keine Meldung auf dem Bildschirm.

BEFEHL:

PRUN
(program run)

prun. cmd

FUNKTION:

Laden einer bestimmten Betriebssystem-Konfiguration von Diskette

FORMAT:

pr Systemfile-Name

WIRKUNG:

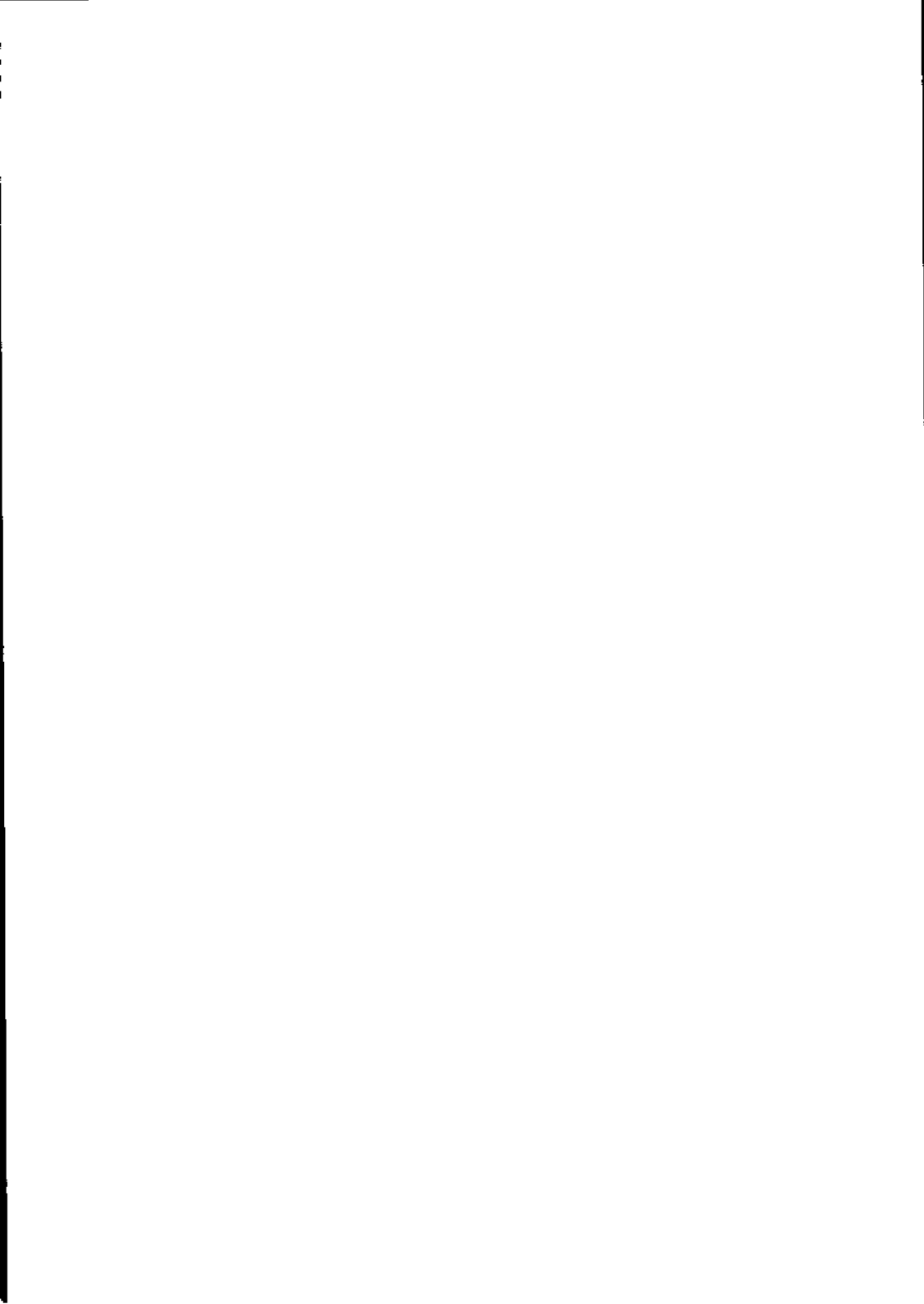
Das mit 'Systemfile-Name' bezeichnete Betriebssystem wird auf Diskette gesucht und geladen.

BEMERKUNGEN:

- Das Betriebssystem muß mit **ps** auf Diskette gespeichert worden sein.
- Wird der Systemfile-Name mit der Diskettenspezifikation **1:** angegeben und der gleicher File-Identifizierer existiert auf der Diskette in Station **0:**, wird das Systemfile von Station **0:** geladen.

BEISPIELE:

pr 1:0S2.sys Die Betriebssystem-Konfiguration namens 0S2.sys wird von Station **1:** als Betriebssystem geladen.



FUNKTION:

Abspeichern der aktuellen, im Arbeitsspeicher befindlichen PCOS-Konfiguration als ladefähiges Betriebssystem auf Diskette

FORMAT:

```
ps [Diskettenspezifikation:] [Filespezifikation]
```

WIRKUNG:

Werden alle Parameter weggelassen, kann die aktuelle Betriebssystem-Konfiguration auf eine in Station Ø: einliegende leere, aber formatierte Diskette unter dem Namen PCOS.SAV als Systemfile abgespeichert werden. Das Systemfile wird dann an erster Stelle im Inhaltsverzeichnis stehen. Wird eine Filespezifikation angegeben, wird die aktuell im Speicher befindliche Betriebssystem-Konfiguration unter diesem Namen ab der nächsten freien Position auf der angesprochenen Diskette abgespeichert. Ist 'Diskettenspezifikation' nicht angegeben, wird auf die Diskette in Station Ø: abgespeichert.

Es wird gefragt: **Save system on file 'Filename'? (y/n):** Nach Eingabe von **y** wird die im Speicher befindliche PCOS-Konfiguration auf der angesprochenen Diskette unter dem angegebenen Namen gespeichert. Es sind darin alle zum Zeitpunkt der Abspeicherung gültigen Default-Werte gesetzt, alle aktuellen Tastenbelegungen, die aktuellen Tabellen der Tastencodes und ihrer zugeordneten ISO

Codes, der aktuell gültige Bildschirm-Zeichensatz und alle zum Zeitpunkt der Abspeicherung residenten PCOS-Befehle enthalten. Dieses File kann zum Laden des Betriebssystems verwendet werden.

Nach Abspeichern des Betriebssystems wird automatisch ein logischer Reset vorgenommen.

BEMERKUNGEN:

- Das File muß nicht vorher angelegt sein.
- Es wird immer das zuletzt auf einer Disk(ette) per **ps** abgespeicherte Systemfile zum Start über den bootstrap herangezogen. Der Name ist beliebig.
- Vor Ausführung des **ps**-Befehles ohne Parameter muß vorher die Station **Ø** angesprochen worden sein (z.B. durch **v1 Ø**:).
- Der Parameter **%n** kann nicht angegeben werden.
- Mit Hilfe des PCOS-Befehls **pr** ist es möglich, wahlweise Betriebssysteme zu laden, die nicht an erster Stelle im Inhaltsverzeichnis stehen.
- Sind auf einer Diskette bzw. Hard-Disk mehrere ladefähige Betriebssysteme gespeichert, wird vom Bootstrap grundsätzlich dasjenige geladen, das als letztes mit **ps** abgespeichert wurde. Andere müssen mit **pr** geladen werden.
- ACHTUNG: Soll ein Betriebssystem z.B. an einer M20 mit 128-KB-RAM ladefähig sein, muß es auch an einer solchen erstellt und mit **ps** abgespeichert worden sein. Entsprechendes gilt für M20 mit größeren RAM-Ausstattungen. Betriebssysteme, die an M20 mit kleinerem RAM konfiguriert wurden, sind jedoch auch an solchen mit größerer RAM-Kapazität ladefähig.

BEISPIELE:

ps speichert das aktuelle Betriebssystem auf der leeren Diskette in Station 0 an erster Position unter dem Namen PCOS.SAV

ps 1:OS02.SYS speichert das aktuelle Betriebssystem auf der Diskette in Station 1 ab der nächsten freien Position unter dem Namen OS02.SYS



FUNKTION:

Entfernen eines mit **pl** resident gemachten oder aufgrund der Typangabe **.sav** nach dem ersten Aufruf automatisch residenten PCOS-Befehls aus dem Arbeitsspeicher

FORMAT:

```
pu PCOS-Befehl [ ,PCOS-Befehl ] ...
```

WIRKUNG:

Der oder die angegebenen PCOS-Befehle werden wieder aus dem Speicher entfernt.

BEMERKUNGEN:

- Die folgenden PCOS-Befehle können mit **pu** nicht mehr aus dem Speicher entfernt werden, wenn sie einmal resident gemacht wurden:
 - ci** (resident nach erstem Aufruf)
 - rs** (resident nach erstem Aufruf)
 - ie** (resident nach erstem Aufruf)
 - ep** (resident nach erstem Aufruf)
 - vm** (resident nach erstem Aufruf)
- Auch PCOS-Befehle, die resident gemacht wurden und anschließend durch **ps** in den direkt ladbaren Teil des Betriebssystems eingebunden wurden, können mit **pu** wieder entfernt werden.

- **pu** kann ohne Feldermeldung für PCOS-Befehle aufgerufen werden, die gar nicht resident sind.
- Die im Arbeitsspeicher gerade residenten PCOS-Befehle können mit dem PCOS-Befehl **dc** abgefragt werden.

BEISPIELE:

- pl vc,vf** macht die PCOS-Befehle **vc** und **vf** bis zum Neuladen des Betriebssystems resident.
- ps 1:COPY.SYS** speichert die im Arbeitsspeicher befindliche Betriebssystem-Konfiguration unter dem Namen **COPY.SYS** auf der Diskette in Station **1** ab.
- pr 1:COPY.SYS** lädt das Betriebssystem namens **COPY.SYS** von der Diskette in Station **1**.
- pu vf** löscht den PCOS-Befehl **vf** aus dem Arbeitsspeicher.

(read font)

FUNKTION:

Übertragen des aktuell gültigen Bildschirm-Zeichensatzes in ein mit dem Full-Screen-Editor bearbeitbares PCOS-Textfile

FORMAT:

rf Tastaturfile

WIRKUNG:

Der aktuell gültige Bildschirm-Zeichensatz wird in das mit 'Tastaturfile' bezeichnete PCOS-Textfile übertragen.

Dort wird automatisch festgehalten:

- a) welche nationale Tastatur als Ausgangs-Zeichensatz zugrundeliegt
- b) wieviele Zeichen über Taste(nkombination) erzeugt werden können (mindestens 95, maximal 190)
- c) aus wievielen Elementarpunktzeilen ein Zeichen besteht (vorläufig immer 10)
- d) welches Bildschirm-Zeichen erzeugt wird, wenn eine bestimmte Taste gedrückt wird. Die Definition des Zeichens geschieht mit Hilfe des ISO-Codes, der dem jeweiligen Tastencode zugeordnet ist.

BEMERKUNGEN:

- Die Beschreibung des Zeichens geschieht durch "Ankreuzen" derjenigen Elementarpunkte einer 8 x 10-Punktmatrix, die bei

Abbildung leuchten sollen.

- Die Punktmatrizen, die für die jeweiligen ISO-Codes gelten, werden im 'Tastaturfile' gespeichert. Die Zeichen sind über den ISO-Code bestimmt und können mit Hilfe des FullScreen-Editors (PCOS-Befehl **ed**) sichtbar gemacht und ggf. geändert werden.
- Ein über Full-Screen-Editor geänderter Bildschirmzeichensatz kann mit **wf** aktiviert werden.
- Die den Tastencodes standardmäßig zugeordneten Original-ISO-Codes sind von der zugrundeliegenden nationalen Tastatur abhängig (s. PCOS-Befehl **sl**) und können in Kapitel 3.2 nachgeschlagen werden. In Kapitel 3.3 ist der Aufbau des 'Tastaturfiles' erklärt.
- Die Zuordnung eines ISO-Codes zu einem Tastencode kann über **ck** geändert werden.

BEISPIELE:

rf 1:ZEICHENS.font

speichert den aktuellen Bildschirmzeichensatz in ein PCOS-Textfile namens ZEICHENS.font auf der Diskette in Station 1.

BEFEHL:

RKILL
(repair kill)

rkill. cmdFUNKTION:

Reaktivieren eines logisch gelöschten Files

FORMAT:

<code>rk [Diskettenspezifikation:]Filespezifikation</code>
--

WIRKUNG:

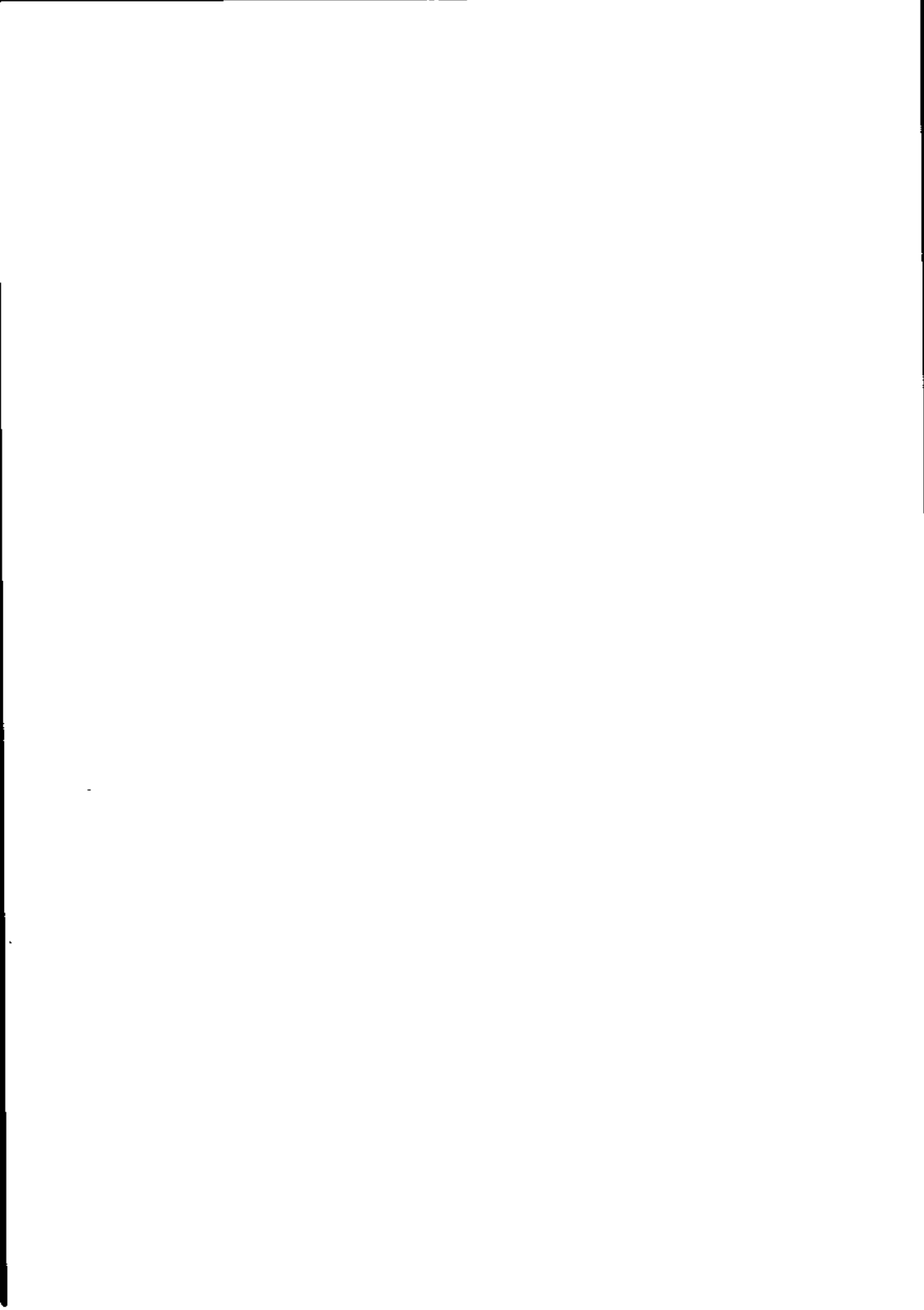
Das durch 'Filename' bezeichnete File wird reaktiviert, nachdem es logisch von Diskette gelöscht worden war. Konnte das File reaktiviert werden, wird "File Successfully Repaired" gemeldet, andernfalls Error 53.

BEMERKUNGEN:

- Wurde das File nach **fk** (oder dem BASIC-Befehl **KILL**) bereits ganz oder teilweise überschrieben, ist **rk** nicht mehr möglich.
- **rk** ist nach **va** nicht mehr erfolgreich.
- Die angesprochene Diskette darf keinen Schreibschutz haben.

BEISPIELE:

rk Ø:alt.asc Das auf der Diskette in Station Ø logisch gelöschte File 'alt.asc' wird reaktiviert.



FUNKTION:

Laden des RS-232C-Pakets zum Arbeiten mit einer RS-232C-(V.24-)Schnittstelle

FORMAT:

rs

WIRKUNG:

Das RS-232-Paket ermöglicht die Verarbeitung von Folgen des PCOS-Befehls **ci** zum Arbeiten mit der Standard-RS-232C-Schnittstelle und/oder mit RS-232C-Schnittstelle(n) der Erweiterungsplatine.

BEMERKUNGEN:

- Die Übertragungsbedingungen (Baud-Rate, Stop-Bits etc.) werden mit dem PCOS-Befehl **sc** gesetzt.
- Die Namen und Eigenschaften der vorhandenen RS-232-Schnittstelle sind durch **sd** abzufragen bzw. zu ändern.
- **rs** sollte in der PCOS-Ebene aufgerufen werden; der Aufruf aus der BASIC-Ebene (mit **CALL**) kann zu Schwierigkeiten in der Speicherorganisation führen.
- **rs** bleibt nach dem ersten Aufruf bis zum Neuladen des Betriebssystems resident. Die Verwendung von **pl** kann somit entfallen. **pu** kann jedoch nicht eingesetzt werden.

- Genauere Hinweise zur Programmierung der RS-232-Schnittstelle finden sich im Handbuch "V.24-Peripherie, Programmierhandbuch".

BEISPIELE:

rs

FUNKTION:

Sichern von auf HDU gespeicherten Files auf Diskette(n) über physische Kopie oder logische gesamte Kopie oder logische selektive Kopie bzw. Rückübertragung von mit **sa** auf Diskette(n) gesicherten Files auf HDU ("Recovery")

FORMAT:

sa Parameterfilename

WIRKUNG:

sa wird verwendet, um Files, die sich auf einer HDU befinden, zu sichern bzw. Files, die von einer HDU gesichert wurden, in einem Recovery-Lauf von Diskette(n) wieder auf eine HDU zu bringen. Es sind folgende Laufarten des Programmes möglich:

- Physisches Sichern bzw. Recovery
- Logisches Sichern bzw. Recovery
- Selektives Sichern bzw. Recovery

Gesteuert wird **sa** durch Parameter, die vorher in einem Parameterfile abgestellt wurden (vgl. dazu Ende des Kapitels). Wird das Programm per **CALL** von BASIC aufgerufen, ist nur noch eine begrenzte Speicherkapazität für BASIC vorhanden. Um diese Kapazität zu erhöhen, kann **sa** in einer Prozedur aufgerufen werden, die die normalerweise nötigen Eingaben über Wechsel des Ein/Ausgabemediums aus einem File statt von der Tastatur entgegennimmt.

Bei einer Prozedur handelt es sich um eine Aneinanderreihung von Befehlen, die in einem sequentiellen File hintereinander gespeichert sind und so abgearbeitet werden, als seien sie über Tastatur eingegeben worden.

Auf den zur Sicherung verwendeten Disketten werden drei Felder initialisiert:

- ein alphanumerischer Name der Diskette(n), Herkunft ist das Parameter-File;
- das Datum des Programmlaufes, Herkunft ist ebenfalls das Parameter-File;
- eine Durchnummerierung der Disketten. Diese wird von sa generiert und verhindert, daß die Reihenfolge der Disketten beim Recoverylauf eine andere ist als beim Sicherungslauf von HD auf MFD.

LAUFARTEN:

Physische gesamte Kopie bzw. physische Recovery

Hierbei wird der gesamte Inhalt der HDU auf 32 320-KB-Disketten gespeichert. Die Disketten erhalten zur Zeitersparnis kein Inhaltsverzeichnis und sind deshalb nicht von PCOS weiter verarbeitbar. Diese Einschränkung wird durch die daraus resultierende Geschwindigkeitserhöhung wettgemacht. Beim Versuch, auf eine Diskette, die mit einer physischen Kopie über sa erstellt wurde, zuzugreifen, wird Error 60 gemeldet.

Es wird eine Numerierung der Disketten generiert, die bei einem physischen Recovery abgeprüft wird. sa meldet, wenn die Reihenfolge der vom Bediener eingelegten Disketten unrichtig ist.

Logische gesamte Kopie bzw. logische gesamte Recovery

Bei der logischen Kopie wird der Inhalt der HDU auf bis zu 32 320-KB-Disketten gespeichert. Der Unterschied zur physischen Kopie besteht darin, daß der Inhalt der HDU File für File auf Diskette übertragen wird. Jede Diskette erhält somit ein Inhaltsverzeichnis und ist von PCOS weiter zu verarbeiten. Durch das Sichern File für File wird jedoch eine erheblich höhere Laufzeit nötig (etwa sechsfache Zeit des physischen Laufes). Wie bei der physischen Kopie wird auch hier eine Numerierung generiert, die bei der Recovery abgeprüft wird. Es werden auch Files bearbeitet, die größer als ein Diskettenfile sind. Eine Bearbeitung von Files auf FD, die größer sind als die betreffende MFD oder sich über mehr als eine MFD erstrecken, ist aber nur mit speziell angepaßten Programmen möglich (PCOS unterstützt keine Multi-Volume-Files).

Logische selektive Kopie bzw. logische selektive Recovery

Bei der selektiven Kopie werden nur die Files, deren Namen im Parameter-File angegeben wurden, übertragen. Wurde im Parameter-File ein Filename angegeben, der nicht existiert, wird die Meldung "ERROR 13 DATEINAME NR. N IST NICHT OK" gegeben. Files, die größer als eine MFD sind, werden ebenfalls bearbeitet.

Es wird ebenfalls eine Disketten-Numerierung generiert.

DAS PARAMETER-FILE

Das Parameter-File dient der Steuerung von **sa** und muß als sequentielles File beim Aufruf von **sa** vorhanden sein. Die Parameter müssen in der richtigen Reihenfolge, durch CR und LF voneinander getrennt, in diesem File stehen. Nachfolgend wird sein Aufbau, die Bedeutung der einzelnen Parameter, ihr Format und ihre Länge beschrieben.

Parameter 1

Wurde **sa** innerhalb einer Prozedur aufgerufen, ist es in der Regel nicht mehr möglich, Eingaben über die Tastatur zu tätigen. Um dieses zu umgehen, dient Parameter 1 und in Abhängigkeit davon Parameter 1.1. Parameter 1 ist alphanumerisch und hat die Länge 1. Er kann folgende Werte annehmen:

- "j": **sa** wird innerhalb einer Prozedur aufgerufen
- "n": **sa** wird nicht innerhalb einer Prozedur aufgerufen

Parameter 1.1

Wurde der Parameter 1 auf "j" gesetzt, muß in Parameter 1.1 der Name des sequentiellen Files angegeben werden, in dem die Prozedureingaben vorabgespeichert sind. Der Name des Prozedurfiles ist alphanumerisch, ist mindestens 3 und höchstens 58 Zeichen lang. Er ist in der Form +sFilename anzugeben.

Beispiel:

Das Prozedurfile befindet sich auf einer Station mit Namen "VOLTEST", hat den Namen "DEV.DAT" und das Passwort "PASS". In diesem Falle sähe Parameter 1.1, natürlich nur wenn Parameter 1 den Wert "j" hat, so aus:

```
+sVOLTEST:DEV.DAT/PASS
```

Hat Parameter 1 den Wert "n", entfällt Parameter 1.1.

Parameter 2

Parameter 2 bestimmt, ob von oder nach HD kopiert werden soll. Er ist alphanumerisch, hat die Länge 1 und kann folgende Werte annehmen:

- "0": es wird von Diskette(n) nach HD kopiert (Recovery)
- "1": es wird von HD nach Diskette(n) kopiert

Parameter 3

Parameter 3 bestimmt die Form des Laufes: physisch, logisch gesamt oder logisch selektiv. Er ist alphanumerisch, hat die Länge 1 und kann folgende Werte annehmen:

- "0": logisch selektiv
- "1": physisch gesamt
- "2": logisch gesamt

Parameter 4

Parameter 4 legt den Namen fest, der zur Identifizierung der Diskette(n) dient. Er ist alphanumerisch, hat eine Länge von 1 bis zu 20 Zeichen und frei wählbar.

Parameter 5

Parameter 5 gibt das Datum des Sicherungslaufes an. Er ist alphanumerisch und hat eine Länge von 8 Zeichen. Folgendes Format ist zulässig (Beispiele):

- "08.04.84"
- "08/04/84"
- "40/50/00".

Es erfolgt also keine Prüfung, ob das Datum zulässig ist.

Parameter 6

Parameter 6 bestimmt die Aktion des Programmes, wenn ein File auf der Zieldisk(ette) bereits existiert. Er ist alphanumerisch, hat die Länge 1 und kann folgende Werte annehmen:

- "j": das File auf der Zieldisk(ette) soll überschrieben werden.
- "n": das File auf der Zieldisk(ette) soll nicht überschrieben werden.
- "?": der Anwender wird gefragt, ob das File auf der Zieldisk(ette) überschrieben werden soll.

Parameter 7

Parameter 7 gibt an, wie Fehler, die beim Lauf von sa auftreten, behandelt werden sollen. Er ist alphanumerisch, hat die Länge 1 und kann folgende Werte annehmen:

- "0": das File, das den Fehler verursachte, wird gelöscht und sa abgebrochen.
- "1": das File, das den Fehler verursachte, wird gelöscht; sa fährt mit der Bearbeitung des nächsten Files fort.
- "2": sa wird sofort abgebrochen.
- "3": sa setzt beim nächsten File fort.

Parameter 8

Parameter 8 bestimmt, was mit Files geschehen soll, die nicht mehr auf die Zieldiskette passen. Er ist alphanumerisch, hat die Länge 1 und kann folgende Werte annehmen:

- "j": Files, die nicht mehr auf eine Diskette passen, werden geteilt.

- "n": Files, die nicht mehr auf eine Diskette passen, werden nicht kopiert.
- "?": sa fragt, ob Files, die nicht mehr auf eine Diskette passen, geteilt werden oder nicht kopiert werden sollen.

Parameter 9

Parameter 9 ist optional. Er kann mehrfach verwendet werden.

Hier sind die Namen aller Files anzugeben, die bei einem selektiven Lauf kopiert werden sollen. Er ist alphanumerisch, hat eine Länge von 1 - 29 und keinen vordefinierten Wert. Gültige Werte sind z.B.:

- "Datei1"
- "Datei2/Passwort"

Ungültige Werte sind z.B.:

- "Ø:Datei1"
- "VOLTEST:DATEI2"

(Es darf also keine Diskettenspezifikation verwendet werden).

Parameter 9 wird nur dann beachtet, wenn Parameter 3 den Wert "Ø" hat. Die Anzahl der Filenamen ist nicht limitiert.

Liste und Werte des Parameter-Files

Parameter-Nr.	ART	Länge	Mögliche Werte
1	alpha-numerisch	1	"j": über Prozedur aufgerufen "n": nicht über Prozedur aufgerufen
1.1	"	3 - 58	+sFilename (Filename: Name des Prozedurfiles)
2	"	1	"0": Kopie Diskette(n) nach HD "1": Kopie HD nach Diskette(n)
3	"	1	"0": selektive Bearbeitung "1": physische Bearbeitung "2": logische Bearbeitung (gesamt)
4	"	1 - 20	Name der Diskette(n), frei wählbar
5	"	8	z.B.: "08.04.84"
6	"	1	"j": existierende Files der Zieldisk(ette) werden überschrieben "n": existierende Files der Zieldisk(ette) werden nicht überschrieben "?: es wird gefragt, ob existierende Files auf der Zieldisk(ette) überschrieben werden sollen

Parameter-Nr.	ART	Länge	Mögliche Werte
7	alpha- nume- risch	1	<p>"0": File, das einen Fehler verursachte, wird auf der Zieldiskette gelöscht und Abbruch</p> <p>"1": Die Bearbeitung wird mit dem nächsten File fortgesetzt. Das File, das den Fehler verursachte, wird gelöscht.</p> <p>"2": Abbruch</p> <p>"3": Fortfahren der Bearbeitung mit dem nächsten File</p>
8	"	1	<p>"j": Files, die nicht mehr auf die Zieldiskette passen, werden aufgeteilt.</p> <p>"n": Files, die nicht mehr auf die Zieldiskette passen, werden nicht kopiert.</p> <p>"?": Es erfolgt eine Abfrage, ob aufgeteilt werden soll oder das Kopieren übergegangen</p>
9	"	1 - 29	z.B. "DATEI/PASSW" (keine Diskettenbezeichnung angeben!)

Aufruf von sa aus einer Prozedur:

BEISPIEL:

Erstellen eines Prozedurfiles und nachfolgender Aufruf aus einem BASIC-Programm:

```
10 OPEN "0".1,"0:DEV PROG"
20 PRINT#1,"SYSTEM"
30 PRINT#1,"sa PARMDAT"
40 PRINT#1,"pe"
50 ANF#=CHR$(34)
60 PRINT#1 "R00"+ANF#+ "FOLGE.BAS"+ANF#+ " 100"
70 CLOSE 1
80 SAVE "0 FOLGE.BAS"
90 EXEC "+sDEV.FROG"
100 PRINT "zurück von Datensicherungslauf"
110 END
```

50%

Dieses Beispielprogramm erstellt ein Prozedurfile (Zeile 10 bis 60), das sa in der PCDS-Ebene aufruft. Das Programm selbst kann in BASIC aufgerufen werden.

Der Aufbau des Parameter-Files für sa wird anhand zweier Beispiele nachfolgend gezeigt:

Programm zum Aufbau eines Parameterfiles mit Namen "PARMDAT", welches einen Aufruf von sa innerhalb einer Prozedur ermöglicht:

```

10 OPEN "0",1,"FARMDAT":'Parameterfilename
20 PRINT#1,"j" : 'Parameter 1
30 PRINT#1,"+sDEV.PROG" : 'Parameter 1.1
40 PRINT#1,"1" : 'Parameter 2
50 PRINT#1,"0" : 'Parameter 3
60 PRINT#1,"FAKTURA" : 'Parameter 4
70 PRINT#1,"08.04.84" : 'Parameter 5
80 PRINT#1,"j" : 'Parameter 6
90 PRINT#1,"2" : 'Parameter 7
100 PRINT#1,"?" : 'Parameter 8
110 PRINT#1,"FAKT1" : 'Parameter 9.1
120 PRINT#1,"FAKT2" : 'Parameter 9.2
130 CLOSE 1:END

```

Würde das Parameterfile über dieses Programm erstellt, würde damit folgendes festgelegt:

1. sa wird in einer Prozedur aufgerufen.
2. Das Prozedurfile heißt "DEV.PROG".
3. Es wird eine Kopie von HD auf Diskette(n) durchgeführt.
4. Es wird selektiv gesichert.
5. Der Name der Diskette(n) lautet **"FAKTURA"**.
6. Das Datum ist **"08.04.84"**.
7. Files, die auf Zieldiskette(n) existieren, werden überschrieben.
8. Bei Auftreten eines Fehlers wird sa abgebrochen.
9. sa stellt eine Abfrage, ob Files geteilt werden sollen, wenn sie nicht mehr auf eine Diskette passen.
10. Die Files, die zu sichern sind, haben die Namen **"FAKT1"** und **"FAKT2"**.

(set BASIC-environment)

FUNKTION:

Verändern von Default-Werten für den BASIC-Interpreter

FORMAT: (wertändernd nur in PCOS)

```
sb [max.Anz.Files], [log.Speichergröße], [max.Anz.Windows]
  [, File-Puffer-Größe]
```

WIRKUNG:

Wird nur **sb** eingegeben, werden die aktuellen Default-Werte ausgegeben. Sind andere Parameter angegeben, werden diese als neue Default-Werte bis zum nächsten Laden des Systems angesetzt.

Parameter:max. Anzahl Files:

Dieser Parameter gibt an, wieviele Files in einem BASIC-Programm gleichzeitig offen sein können.

Wert: 0 bis 15

Default-Wert: 3

log. Speichergröße:

Dieser Parameter gibt an, wieviele Bytes des Arbeitsspeichers dem BASIC-Interpreter zur Verfügung gestellt sind.

Wert: 0 bis 57000; abhängig von Konfiguration (Basis + evtl. RAM-Erweiterungen)

Default-Wert: 36800

max. Anzahl Windows:

Dieser Parameter gibt an, wieviele Windows in einem Programm gleichzeitig angelegt sein können.

Wert: 1 bis 16

Default-Wert: 1

File-Puffer-Größe:

Dieser Parameter gibt an, wieviele Bytes im BASIC-Bereich des Arbeitsspeichers für jedes File automatisch für einen zuzuordnenen Puffer reserviert werden sollen, sobald BASIC aufgerufen wird.

Wert: 1 bis 4096

Default-Wert: 256

BEMERKUNGEN:

- Für eine Änderung der Parameter muß sb in der PCOS-Ebene aufgerufen werden!
- Die Größe des für BASIC-Programme und -Daten zur Verfügung stehenden Speicherplatzes hängt ab von:
 1. 'log. Speichergröße' = gesetzte Speichergröße; maximal können dafür 57 K gesetzt werden;
 2. dem Parameter, 'max.Anz.Files' in Verbindung mit 'File-Puffer-Größe';
 3. dem Parameter 'max.Anz.Windows';

4. der Anzahl der Zeichen, die bei Tastenbelegungen mit pk vergeben wurden.

'log. Speichergröße' wird immer gemäß der Formel:

$$829+(578+R)*F+(W-1)*108$$

verkleinert, wobei:

F: max. Anz. Files

R: File-Puffer-Größe

W: max. Anz. Windows

- Der Error 7 ("Out of memory") kann behoben werden, indem die Parameter 'File-Puffer-Größe' und/oder 'max.Anz.Files' und/oder 'max.Anz.Windows' reduziert werden.
- Beim Laden des Systems werden sofort soviele File-Puffer mit der physischen Größe 'File-Puffer-Größe' eingerichtet, wie mit 'max.Anz.Files' festgelegt. Dies ist unabhängig vom Parameter 'Record-Länge' bei der BASIC-Anweisung **OPEN** für Random-Files: Dieser gibt nur die logische Länge des Puffers an, mit der danach im Programm tatsächlich gearbeitet wird.
- Auch bei 'max.Anz.Windows' = 1 kann mit mehr als einem Window gearbeitet werden. Es kann jedoch später zum Error 7 kommen.
- Wird ein Parameter nicht angegeben (nur das Komma als Platzhalter gesetzt), gilt der Wert des zuletzt erfolgten sb.

BEISPIELE:

sb +dprt: Die aktuellen Default-Werte werden am Drucker ausgegeben.

sb %n 8,,10,500 Es können 8 Files gleichzeitig offen sein; die aktuelle 'log.Speichergröße' wird übernommen; bis zu 10 Windows können ohne Fehler

verwaltet werden; für 8 Files werden sofort Puffer zu je 500 Bytes reserviert. Es wird keine Meldung ausgegeben.

(set communication)

FUNKTION:

Festlegung der Übertragungseigenschaften für einen RS-232-(V.24-) Übertragungsbetrieb

FORMAT:

```
sc Name der Einheit: , [ Bits/ Sek. ] , [ Paritäts- prüfung ] , [ Stop- Bits ] ,  
[ Bitanzahl ] , [ Voll/Halb- Duplex ] , [ XON/ XOFF ] [ Puffer- Größe ]
```

WIRKUNG:

Wird nur **sc** 'Name der Einheit:' angegeben, werden die aktuell gültigen Default-Werte für die RS-232-Übertragung für die betreffende Einheit ausgegeben. Werden weitere Parameter angegeben, werden diese geändert.

Parameter:

Name der Einheit: Name der RS-232-Einheit, mit der der Betrieb stattfindet; entweder
com: bzw. **com1:** bzw. **com2:** oder
mit **sd** vergebener 'Name der Einheit' (immer Doppelpunkt am Ende)

Bits/Sek.: mögliche Werte:
50, 110, 300, 600, 1200, 1800, 2400, 4800
oder 9600
Default-Wert: 9600

Paritäts-Prüfung: entweder
ODD (ungeradzahlig) oder
EVEN (geradzahlig) oder
NONE (keine)
Default-Wert: **NONE**

Stop-Bits: Anzahl der Bits, die das Ende einer Sende-
bzw. Empfangs-Einheit markiert
Ø: 1 stop-bit
1: 1 1/2 stop-bits
2: 2 stop-bits
Default-Wert: 2

Bit-Anzahl: Anzahl der Bits, die eine Sende- bzw.
Empfangs-Einheit bilden
Werte: 5, 6, 7 oder 8
Default-Wert: 7

Voll/Halb-Duplex: gibt an, ob Rückübertragung (Echo) von
Sendeeinheiten stattfindet
Werte:
FULL mit Rückübertragung
HALF ohne Rückübertragung
Default-Wert: **HALF**

XON/XOFF: gibt an, ob Rückmeldungen über den Pufferzustand der Einheit mit Hilfe von DC1- bzw. DC3-Übertragungen stattfinden.

Werte:

ON ja

OFF nein

Default-Wert: **ON**

Puffer-Größe: Größe des Puffers bei der Übertragung (in Bytes)

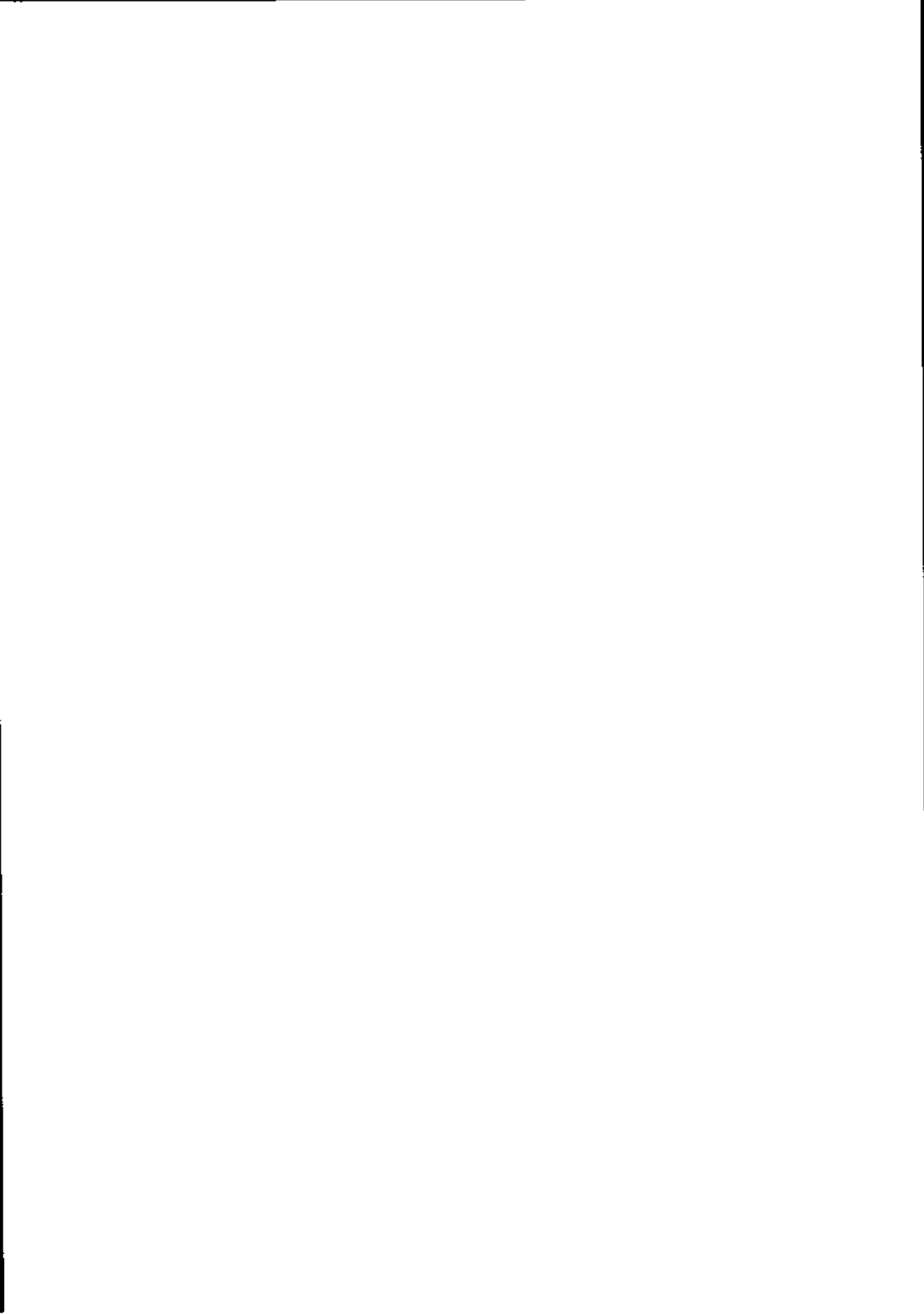
Default-Wert: 128

BEMERKUNGEN:

- Das RS-232-Paket muß zuvor mit **rs** geladen worden sein.
- Die Übertragung der Sende- bzw. Empfangseinheiten sowie der Steuer- und Kontroll-Befehle geschieht in der BASIC-Ebene mit Hilfe von Aufrufen des PCOS-Befehls **ci** (**CALL "ci" (...)**).
- Für die Definition von Voll/Halb-Duplex liegt die amerikanische Bedeutung zugrunde.
- Wird ein Parameter nicht angegeben (nur Setzen des Kommas als Platzhalter), gilt der Wert des für die betreffende Einheit zuletzt abgearbeiteten **sc**.
- **sc** sollte nicht in der BASIC-Ebene aufgerufen werden.

BEISPIELE:

sc %n com: ,,,6 Es werden für die an der Basis-Einheit angeschlossene RS-232-Einheit die zuletzt mit **sc** bzw. nach Laden des Systems gesetzten Default-Werte genommen. Die Sende- bzw. Empfangs-Einheiten sind 6 Bits lang. Eine Ausgabe der Informationen unterbleibt.



FUNKTION:

Feststellen von Namen und Art einer physischen Einheit am M20;
ggf. Festlegen eines Namens für eine physische Einheit

FORMAT:

```
sd [alter Name :] [neuer Name :]
```

WIRKUNG:

Wird nur **sd** eingegeben, wird ein Verzeichnis aller am M20 angeschlossenen Einheiten ausgegeben. (Namen von Nur-Schreibeinheiten sind mit W versehen, solche von Nur-Leseeinheiten mit R und Namen von Schreib-/Lese-Einheiten mit R/W).

Wird nur der Parameter 'alter Name' angegeben, wird Auskunft über die Art dieser Einheit gegeben. Wird zusätzlich der Parameter 'neuer Name' angegeben, wird die betreffende Einheit umbenannt.

BEMERKUNGEN:

- Die Default-Namen der Einheit sind:

prt: Drucker

cons: Bildschirm und Tastatur

ieee: IEEE-488-Driver

com: Standard-RS-232-Driver

com1: erster RS-232-Driver auf der Erweiterungsplatine

com2: zweiter RS-232-Driver auf der Erweiterungsplatine

- Die Namen der Einheiten dürfen bis zu 13 zulässige Zeichen lang sein (vgl. unzulässige Zeichen bei Filenamen; Kapitel 2.4) und müssen mit einem : abgeschlossen sein.

BEISPIELE:

- sd** gibt eine Tabelle über Art und Namen der aktuell vorhandenen Einheiten aus
- sd prt:;Drucker:** die Einheit, die bisher unter dem Namen **prt:** anzusprechen war, erhält den neuen Namen 'Drucker:' zugeordnet
- sd prt:;com:** eine Einheit (z.B. Olivetti ET 121), deren Übertragungseigenschaften zuvor über **rs** und **sc** definiert wurden, wird als serieller Drucker eingesetzt. (Es muß vorher mit **sf** der angeschlossene Drucker als seriell festgelegt worden sein).

Bei identischer Aufgabe für `sa`, wobei jedoch der Aufruf nicht aus einer Prozedur erfolgt, ist das Parameter-File folgendermaßen aufzubauen:

```
10 OPEN "0",1,"PARMDAT":'Parameterfilename
20 PRINT#1,"n"           :'Parameter 1
40 PRINT#1,"1"           :'Parameter 2
50 PRINT#1,"0"           :'Parameter 3
60 PRINT#1,"FAKTURA"    :'Parameter 4
70 PRINT#1,"08.04.84"    :'Parameter 5
80 PRINT#1,"j"           :'Parameter 6
90 PRINT#1,"2"           :'Parameter 7
100 PRINT#1,"?"          :'Parameter 8
110 PRINT#1,"FAKT1"      :'Parameter 9.1
120 PRINT#1,"FAKT2"      :'Parameter 9.2
130 CLOSE 1:END
```

Parameterfiles können leichter unter Zuhilfenahme des Full-Screen-Editors (`ed` mit Parameter `%c`) erstellt werden. Dort könnte z.B. der Inhalt des Parameterfiles `PARMDAT` direkt angegeben werden:

```
n
1
Ø
FAKTURA
08.04.84
j
2
?
FAKT1
FAKT2
```



FUNKTION:

Ansprechen einer gewünschten Diskettenstation

FORMAT:

se Stationsnr.

oder (nur BASIC)

CALL "se"([§] Integervariable)

WIRKUNG:

Es wird versucht, die mit 'Stationsnr.' bzw. 'Integervariable' angegebene Station zu aktivieren.

1. bei Format **se** Stationsnr.:

Die mit 'Stationsnr.' definierte Station wird angesprochen, sofern vorhanden. Ist die Station nicht vorhanden oder es liegt keine Diskette ein oder die einliegende Diskette ist schadhaft, wird keine Fehlermeldung gegeben.

2. bei Formaat **CALL "se"([§] Integervariable):**

Bei Angabe von § wird versucht, die mit 'Integervariable' (mögliche Werte: 0, 1 oder 10) definierte Station anzusprechen

und zu aktivieren. Ist die Station nicht vorhanden, der Diskettenschacht offen oder die einliegende Diskette nicht lesbar, wird in 'Integervariable' der entsprechende BASIC-Fehlercode gemeldet. Ist die Station vorhanden, wird auf die Diskette ein Zugriff durchgeführt und 'Integervariable' erhält den Wert 0.

Ist § nicht angegeben, wird die gleiche Wirkung wie bei 1. erzielt. (Dies kann auch erreicht werden, indem man die Formatvariante 1. via **CALL** einsetzt.)

BEMERKUNGEN:

- 'Integervariable' muß vor dem ersten Aufruf einen Wert erhalten haben.
- 'Integervariable' enthält den betreffenden PCOS-Fehlercode, wenn ein PCOS-Fehler auftritt. Beim Aufruf wird der PCOS-Fehler nicht gemeldet.
- 'Integervariable' enthält den Wert 57, wenn das betreffende Laufwerk nicht vorhanden ist, der Diskettenschacht offen ist oder die einliegende Diskette nicht lesbar ist (z.B. nicht mit **vf** formatiert oder Lesefehler im Inhaltsverzeichnis der Diskette).
- 'Integervariable' enthält den Wert 73, wenn ein anderer Wert als 0, 1 oder 10 eingereicht wurde.

BEISPIELE:

se 10

253 STATION%=10:CALL "se" (§STATION%)

(set format)

FUNKTION:

Verändern von Default-Werten für den angeschlossenen Drucker

FORMAT:

```
sf [ Status ], [ Druckerbez. ], [ Anz.Zeilen/Seite ],  
  [ Zeilenschaltungen ], [ Zeichenformat ],  
  [ Schnittstellentyp ], [ 'Kopftext' ]
```

WIRKUNG:Wird nur **sf** eingegeben, werden die aktuellen Default-Werte ausgegeben.

Alle zusätzlich angegebenen Parameter werden als neue Default-Werte übernommen.

Parameter:Status (von nicht angegebenen Parametern):Wert: **off** oder **on**'off' legt fest, daß für nicht angegebene Parameter die Default-Werte gültig sein sollen.'on' legt fest, daß für nicht angegebene Parameter diejenigen Parameter Gültigkeit haben sollen, die im letzten **sf** gesetzt

wurden. Ist noch kein **sf** erfolgt, haben die Werte Gültigkeit, die beim letzten **ps** festgehalten wurden.

Default-Wert: **off**

Druckerbez.:

Typbezeichnung des angeschlossenen Druckers;

Wert: **PR1450**, **PR1471**, **PR2400**, **PR1481**, **PR2300**, **ET-121**, **ET-231**,
PR-430 oder **TRANSP**.

TRANSP legt fest, daß im freien Format gedruckt werden soll
(keine Berücksichtigung von CR und LF).

Default-Wert: **PR1450**

Anz.Zeilen/Seite:

Anzahl der zu druckenden Zeilen pro Blatt;

Wert: 0 bis 65535 (0 gibt an, daß nie Seitenvorschub von PCOS
her erfolgen soll)

Default-Wert: 60

Zeilenschaltungen:

Anzahl der Zeilenschaltungen, zwischen den gedruckten Zeilen (1 =
normale Zeilenschaltung)

Wert: 1 bis 'Anz.Zeilen/Seite'

Default-Wert: 1

Zeichenformat:

FORMAT: XY

X: n oder w

n: normaler Zeichenabstand

w: gesperrter Druck (ein Zeichen Abstand)

Y: c oder e oder p

c (compressed): 16.6 Zeichen/Zoll

e (elite): 12.5 Zeichen/Zoll

p (price): 10.0 Zeichen/Zoll

Default-Wert: ne

Schnittstellentyp:

se: seriell (RS-232C)

pa: parallel (Centronics)

Default-Wert: pa

Kopftext:

Dieser Text wird auf jedem neuen Blatt in der ersten Zeile am oberen linken Rand ausgegeben (maximal 24 Zeichen).

Default-Wert: Kein Kopftext

Der 'Kopftext muß in ' ' eingeschlossen sein.

Soll der Kopftext ein ' enthalten, muß er in " eingeschlossen sein. Eine Kombination von " und ' ist nur mit Hilfe von CHR\$ möglich.

Um den Kopftext bereits auf dem ersten Blatt auszugeben, ist über BASIC ein FF an den Drucker zu senden (LPRINT CHR\$(12)).

Wird als Kopftext 'ü' gesetzt, ist kein Kopftext definiert.

BEMERKUNGEN:

- Der Drucker muß eingeschaltet sein (und darf nicht auf LOCAL stehen), bevor PCOS geladen wird und wenn ein **sf**-Befehl gegeben wird.
- Der Drucker hat eigene Intelligenz und eigenes Memory und speichert die ihm gesendeten Steuerzeichen und -befehle bis zu 30 Sekunden. Es sollte deshalb nach Ausschalten ca. 30 Sekunden bis zum Neueinschalten gewartet werden.
- Die Steuerung des Druckers geschieht (von obigen Parametern abgesehen) durch Senden von Steuerzeichen in der BASIC-Ebene mit der BASIC-Anweisung **LPRINT** (vgl. BASIC-Handbuch, Kapitel 5.1). Dabei ist besonders zu beachten, daß der Parameter 'Anz.Zeilen/Seite' auf \emptyset gesetzt sein muß, soll ein Programm-(Zähler-)gesteuerter Blattvorschub erreicht werden.
- Ist für 'Status' default-mäßig **off** gesetzt, verhält sich **sf** bei nicht angegebenen Parametern anders als die anderen PCOS-Befehle, denn es wird stets auf die Default-Werte zurückgegriffen und nicht auf vorher gesetzte **sf**-Werte bzw. auf die **sf**-Werte, die durch eine mit **ps** erstellte neue Systemkonfiguration festgelegt wurden!
- Soll 'Status' verändert werden, darf nur dieser Parameter angegeben werden.

BEISPIELE:

sf off,pr1450,0,1,nc,pa,'DEUTSCHE OLIVETTI'

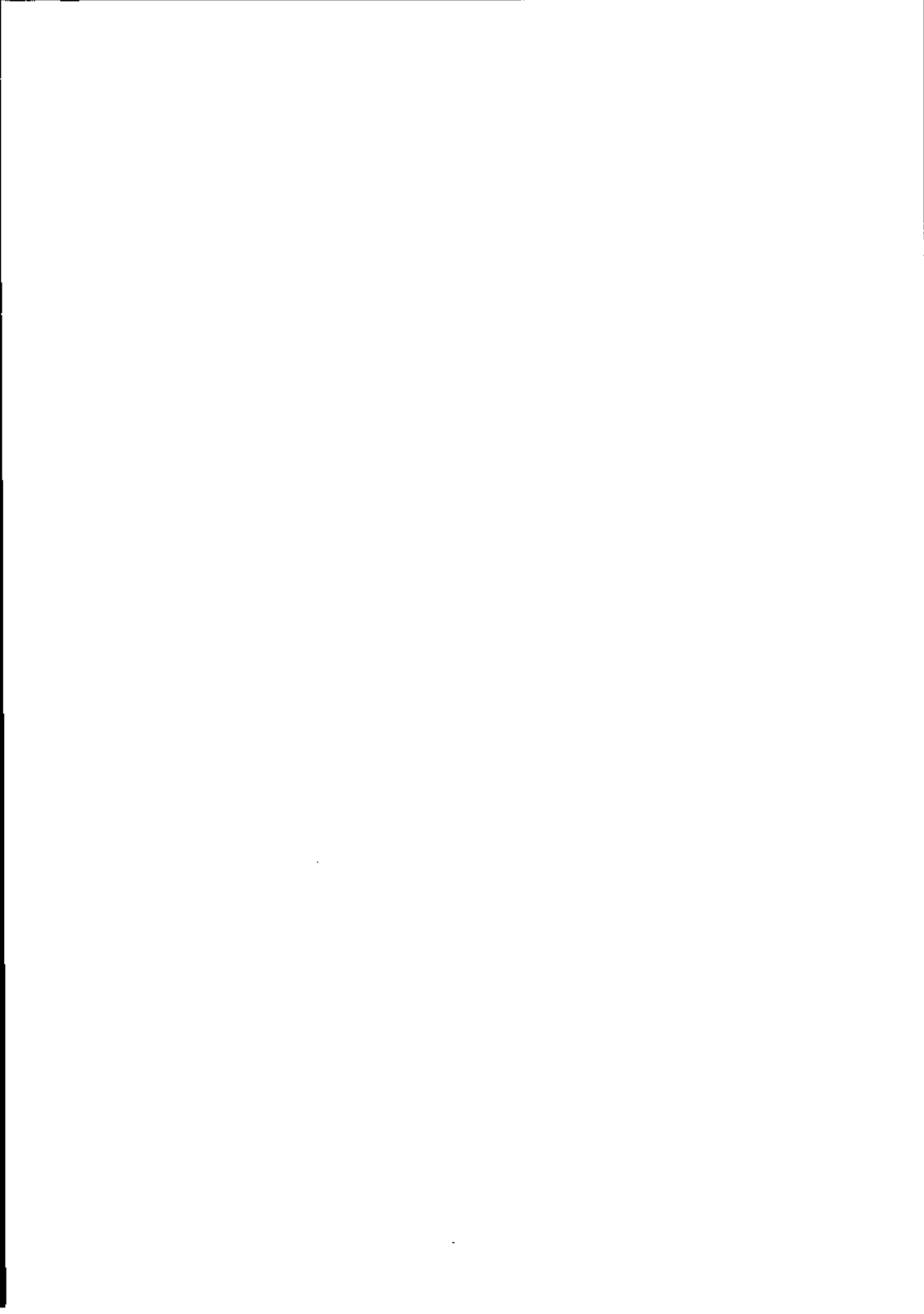
Aktiviert einen Drucker PR1450.

Setzt einen Seitenvorschub außer Kraft, der von PCOS gesteuert wird (d.h., es wird ein BASIC-Programmgesteuerter Vorschub über Zähler oder über ein Druckertabulationsprogramm vorgenommen); die Zeilenschaltung ist "normal"; es wird mit normalem Zeichenabstand mit 16.6 Zeichen/Zoll ausgegeben.

Der Drucker ist über eine Parallel-Schnittstelle angeschlossen. Auf jeder neuen Seite (außer der allerersten) wird automatisch in der ersten Zeile links der Text DEUTSCHE OLIVETTI ausgegeben.

CALL "sf %n,,0"

Von PCOS wird kein Seitenvorschub mehr gesteuert. Es wird keine Meldung ausgegeben.



Zu **sf**-Befehl

Für 'Druckerbezeichnung' können auch die folgenden Werte gesetzt werden:
PR 2835 und **PR-320**

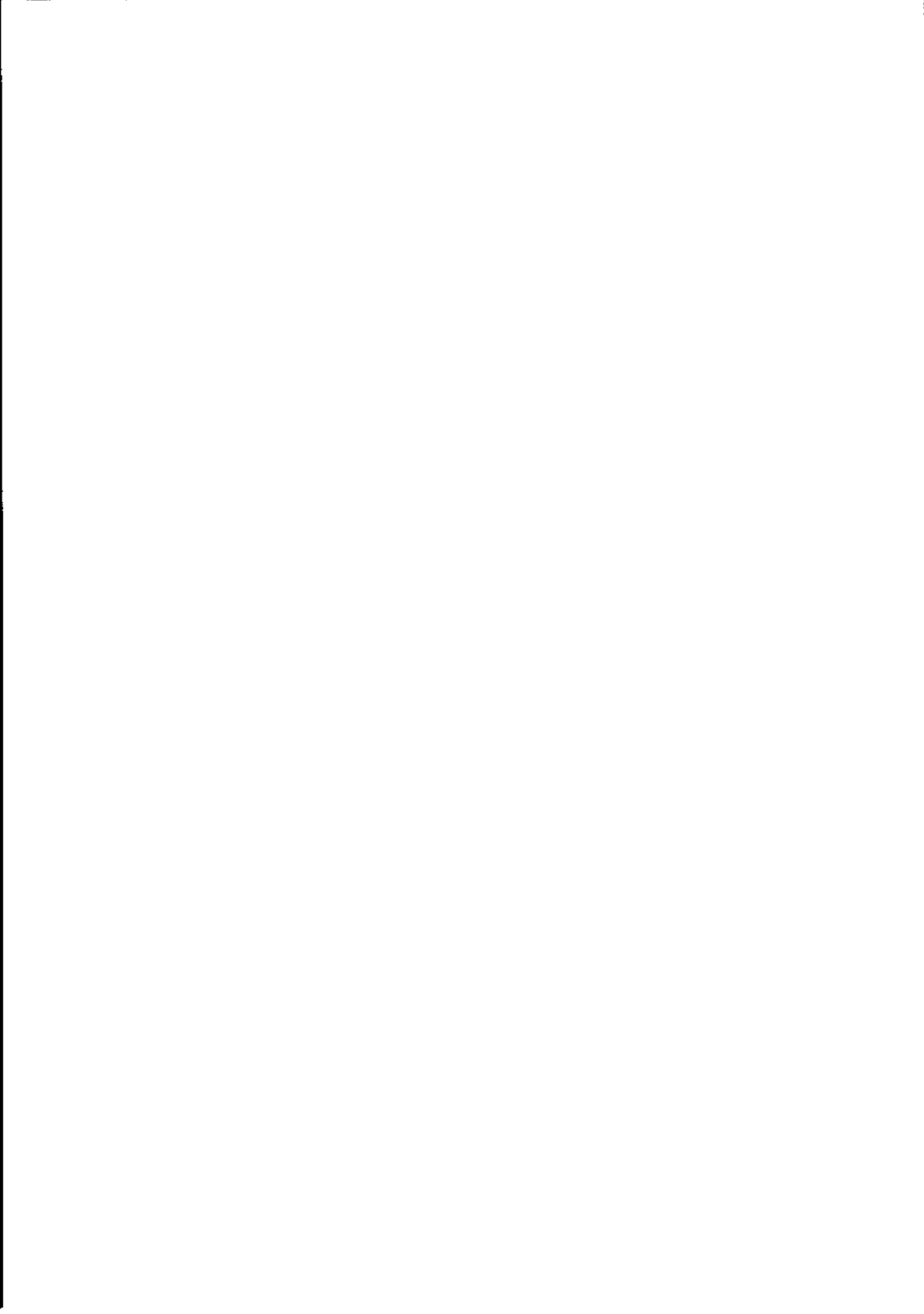
Für 'Schnittstellentyp' können auch die folgenden Werte gesetzt werden:

s1: seriell (an der ersten Schnittstelle der Erweiterungsplatine; **com1**:)

s2: seriell (an der zweiten Schnittstelle der Erweiterungsplatine; **com2**:)

se: bezieht sich nur auf die Standard-RS232C-Schnittstelle

Vor Definition eines seriellen Druckers muß der **rs**-Befehl gegeben worden sein.



(set language)

FUNKTION:

Auswahl der nationalen Tastatur

FORMAT:

```
s1 [Tastaturvariante]
```

WIRKUNG:

Ist nur s1 angegeben, wird die Auswahl der nationalen Tastatur ermöglicht. Bildschirminhalt:

```
Set Language Rev. 2.0h
Available Country Configurations :

Italy ..... 0          Yugoslavia ..... 10
West Germany ..... 1   Norway ..... 11
France ..... 2         Greece ..... 12
Great Britain ..... 3   Switzerland/France ... 13
United States ..... 4   Switzerland/Germany .. 14
Spain ..... 5          Germany (Original) ... 15
Portugal ..... 6       Datev ..... 16
Sweden/Finland ..... 7
Denmark ..... 8
```

Enter Your Selection by Number (or q to quit) -->

Nach Eingabe des zur gewünschten nationalen Tastatur gehörigen Wertes zwischen 0 und 16 incl. (Abschluß mit CR) wird die entsprechende Tastatur ausgewählt. Bei Eingabe von q bleibt die aktuelle Tastatur gültig.

Ist nach s1 eine Nummer zwischen 0 und 16 angegeben, wird sofort die entsprechende Tastatur aktiviert.

BEMERKUNGEN:

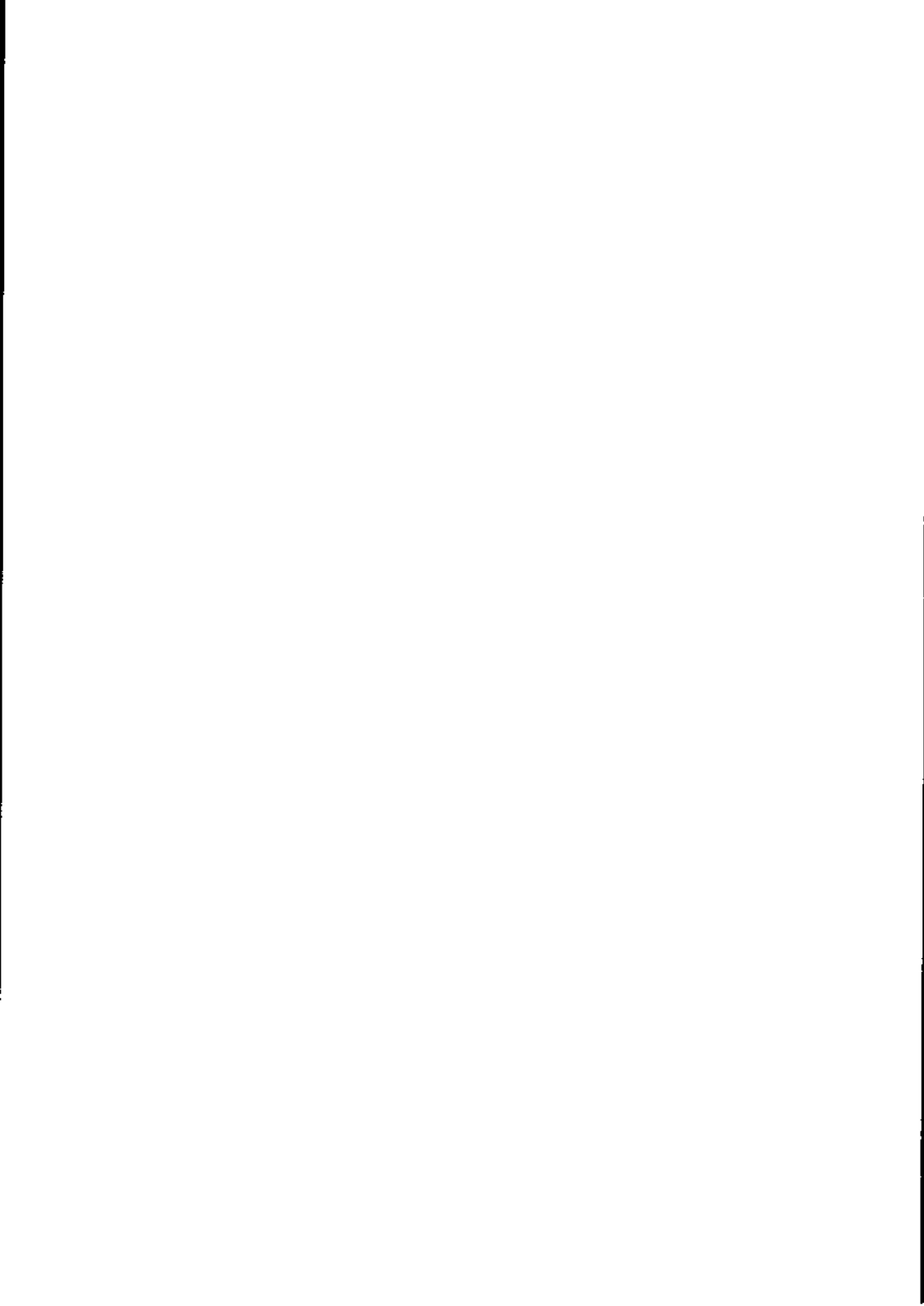
- Folgende Tastaturvarianten sind für den deutschen Sprachraum vorgesehen:
 - 1 neue deutsche Tastatur (DIN 2137)
 - 15 alte deutsche Tastatur
 - 16 DATEV-Tastatur für DES V bzw. DES VI
 - 14 Schweizer Tastatur, deutschsprachige Version

- Bei Anwendung von **ps** wird die gerade gültige Tastaturversion als Default-Wert beim Neuladen des Betriebssystems unterstellt.
- Die aktuell zugrundeliegende Tastatur kann ermittelt werden mit Hilfe von **rf** und anschließender Abfrage der ersten zwei Zeilen des bei **rf** angegebenen 'Tastaturfiles' (z.B. mit Hilfe des Full-Screen-Editors).
- Bei Auswahl einer neuen nationalen Tastatur müssen die Files **font.all** und **kb.all** auf Diskette vorhanden sein.
- Liegt die US-ASCII-Tastatur anstelle einer europäischen Tastatur zugrunde, wird das Tagesdatum in der Form **MM/TT/[JJ]JJ** verwaltet.
- Die wichtigsten möglichen Tastaturen, die zugeordneten Standard-Tastencodes und die den Tastencodes standardmäßig zugeordneten Sonderfunktionen bzw. Original-ISO-Codes können in den Kapiteln 3.1 und 3.2 nachgeschlagen werden.
- Die Default-Tastatur ist US-ASCII.
- Für DES V und DES VI muß die Tastatur 16 gewählt werden, wenn die Tasten S3, S5 und S6 Funktion als Abschlußtaste erhalten sollen.

BEISPIELE:

sl gibt die Auswahl der gewünschten Tastatur frei.

sl 1 wählt die Tastatur DIN 2137 aus.



FUNKTION:

Abbildung des Inhalts eines Windows oder des gesamten Bildschirms auf dem Drucker (Hardcopy)

FORMAT:

<code>sp [Window-Nr.] , [Darstellung] , ['Kopftext'] [,Datum/Zeit]</code>

WIRKUNG:

Auf dem Drucker wird der Inhalt eines Windows bzw. des gesamten Bildschirms abgebildet.

Window-Nr.: Ø: gesamter Schirm
Wert: Ø bis 16
Default-Wert: Ø

Darstellung: p oder n
n: Bildschirminhalte in Hintergrundfarbe werden weiß abgebildet, Vordergrund- und sonstige Inhalte schwarz (normale Abbildung)
p: Bildschirminhalte in Hintergrundfarbe werden schwarz abgebildet, Vordergrund- und sonstige Inhalte weiß (Reverse-Darstellung).
Beim Farbschirm werden alle Farbe, die nicht Hintergrundfarbe sind, als Vordergrundfarbe interpretiert.
Default-Wert: n

Kopftext: Zeichenfolge von 1 bis zu 24 druckbaren Zeichen; das erste Zeichen darf nicht numerisch sein. Der Kopftext wird links über der Hardcopy ausgegeben. Er ist in ' ' einzuschließen.

Enthält der Text ein ', muß er in " " eingeschlossen sein. Eine Kombination von ' und " ist nur mit Hilfe von CHR\$ möglich.

Default-Wert: keine Kopftext-Ausgabe

Datum/Zeit: **dt** oder **no**

dt: Der Inhalt der speziellen Variablen DATE\$ und TIME\$ wird über der Hardcopy ausgegeben.

no: Keine Ausgabe

Default-Wert: **no**

BEMERKUNGEN:

- Der Befehl ist nur für hardcopy-fähige Drucker möglich (PR 2400, PR 1450); sonst wird Error 110 gemeldet.
- Der hardcopy-fähige Drucker muß vorher über sf konfiguriert worden sein (Parameter 'Druckerbez.').
- Wurde sp mit pl resident gemacht, werden bei Aufruf unter Weglassen von Parametern dafür diejenigen Werte angesetzt, die beim letzten Aufruf gültig waren.
- 'Kopftext' und 'Datum/Zeit' werden hintereinander vom linken Rand des Papiers ausgehend, ausgegeben. Es folgt eine Leerzeile.
- Wird ein spezielles Window abgebildet, erfolgt diese Abbildung stets ab dem linken oberen Papierrand.

BEISPIELE:

sp bewirkt eine Hardcopy des gesamten Bildschirms

sp 3,p,'SEGMENT 1',dt

Ab dem oberen linken Papierrand wird der Text SEGMENT 1, gefolgt vom Tagesdatum, der Uhrzeit und einer Leerzeile, ausgegeben. Die in Window-Nr. 3 des Bildschirms enthaltenen Informationen werden in Reverse-Darstellung auf dem Drucker abgebildet.



Zu **sp**-Befehl

Für 'Darstellung' kann auch der Wert **c** gesetzt werden. Dies ermöglicht Farbausgaben am PR1481.



BEFEHL:

SSYS

ssys. cmd



(set system environment)

FUNKTION:

Ändern von Default-Werten für das Betriebssystem PCOS

FORMAT:

<pre> ss [Datum ,] [Uhrzeit] , [Verifikation] , [anzulegende Sektoren] [,Bildschirmformat] </pre>
--

WIRKUNG:

Wird nur ss eingegeben, werden die aktuell gültigen Parameter angezeigt. Alle angegebenen zusätzlichen Parameter gelten als Default-Wert bis zum nächsten Laden von PCOS.

Parameter:

Datum: Anfangswert des Datums für die Echtzeituhr

FORMAT: TT/MM/JJ oder TT/MM/JJJJ

wobei

TT	= Tag	(01-31)	} numerisch
MM	= Monat	(01-12)	
[JJ]JJ	= Jahr	(1900 - 9999)	

mit beliebigem alphanumerischem Trennzeichen anstelle von / . (vgl. DATE\$ im BASIC-Handbuch)

Default-Wert: 01/01/1982

Uhrzeit: Anfangswert der Uhrzeit für die Echtzeituhr
FORMAT: HH:MM:SS
 wobei
 HH = Stunde (00-23)
 MM = Minute (00-59) } numerisch
 SS = Sekunde (00-59)
 mit beliebigem alphanumerischem Trennzeichen
 anstelle von : . (vgl. **TIME\$** im BASIC-Hand-
 buch)
 Default-Wert: 00:00:01

Verifikation:

Dieser Parameter legt fest, ob von PCOS nach jedem Schreibvorgang auf Diskette eine Rückmeldung des Geschriebenen erfolgen soll (Lesen nach Schreiben).
 Wert: 0 (nein) oder 1 (ja)
 Default-Wert: 0

Anzulegende Sektoren:

Dieser Parameter legt fest, wieviele Sektoren zu je 256 Bytes auf Diskette von PCOS automatisch für ein File reserviert werden sollen, falls das File noch nicht vorhanden ist (beim BASIC-Befehl **SAVE** bzw. der BASIC-Anweisung **OPEN**), bzw. um wieviele Sektoren das File automatisch vergrößert wird, falls es zu klein ist (beim BASIC-Befehl **SAVE** und den BASIC-Anweisungen **PRINT#** oder **WRITE#** oder **PUT**).
 Wert: 1 bis Disk(etten)-Kapazität in Sektoren nach **vf**
 Default-Wert: 8

Bildschirmformat:

64 Zeichen pro Zeile bei 16 Zeilen im Gesamtschirm = 0

80 Zeichen pro Zeile bei 25 Zeilen im Gesamtschirm = 1

Default-Wert: 0

BEMERKUNGEN:

- Für nicht angegebene Werte gilt der zuletzt definierte Wert.
- Für das 'Datum' wird keine Plausibilitätsprüfung (z.B. auf den 29.02.81) durchgeführt.
- Das Default-Trennzeichen für 'Datum' ist / und das für 'Uhrzeit' ist :. Wird das Jahrhundert nicht angegeben, wird als Default-Wert 19 eingesetzt.
- 'Datum' und 'Uhrzeit' werden bei physischem Reset wieder auf die obengenannten Default-Werte gesetzt. Der physische Reset erfolgt beim Neuladen des Systems nach Ausschalten (ON/OFF-Schalter) oder nach Stecken eines Stiftes o.ä. in die Ausnehmung rechts außen an der Basis. Drücken von **RESET** gilt als logischer Reset und setzt Datum und Uhrzeit nicht zurück auf Default-Werte.
- Wird bei 'Verifikation' = 1 eine Rückmeldung als falsch erkannt, wird Error 57 (als PCOS-Fehler) gemeldet. Die zu schreibenden Werte sind noch vorhanden; der Schreibversuch kann wiederholt werden. Alle Schreibvorgänge dauern erheblich länger.
- Wird bei 'anzulegende Sektoren' 1 gesetzt, werden bei der Neuanlage von Files dennoch 2 Sektoren verwendet.
- Das Bildschirmformat kann auch innerhalb der BASIC-Ebene für jedes Window geändert werden (vgl. BASIC, Funktion **WINDOW** und besonders die Anmerkung zur Anweisung **SCALE**).
- Liegt die US-ASCII-Tastatur anstelle einer europäischen Tastatur zugrunde (vgl. s1), wird 'Datum' nicht mehr in der Form

TT/MM/JJJJ verwaltet, sondern in der Form MM/TT/JJJJ.

- Liegt der Wert von 'Datum' oder 'Uhrzeit' beim Versuch der Änderung außerhalb der Grenzen, die dafür zulässig sind, wird keine Fehlermeldung gegeben, der aktuelle Inhalt von 'Datum' bzw. 'Uhrzeit' jedoch nicht verändert (zugrundeliegende Tastatur beachten!).

BEISPIELE:

ss +dprt: Die aktuellen Default-Werte werden am Drucker ausgegeben.

ss %n ,,,,3 Alle aktuellen Default-Werte werden übernommen; bei Neuanlage eines Files werden 3 Sektoren bzw. ein Vielfaches davon angelegt; bei automatischer Vergrößerung wird analog vorgegangen. Es wird keine Meldung ausgegeben.

ss 09.02.84, 08:02:00
Anfangs-Datum und -Uhrzeit werden gesetzt.

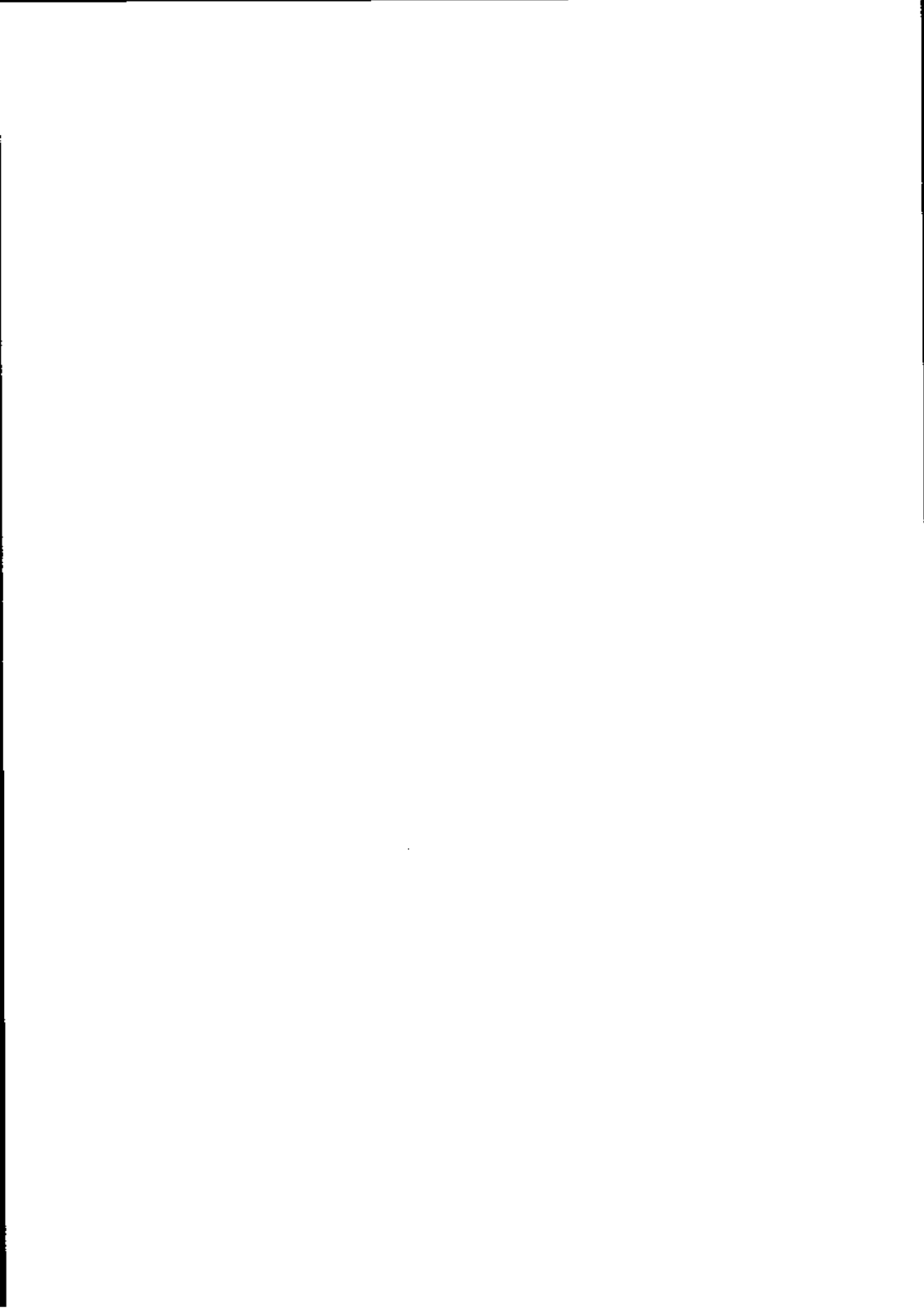
Zu **ss**-Befehl

Neuer optionaler Parameter (als letzter Parameter)

, [Anzahl Sekunden]

Der Wert darf zwischen 1 und 30 liegen und legt die Anzahl Sekunden fest, die sich der Antrieb der zuletzt angesprochenen Diskettenstation nach einem Zugriff weiterdreht. (Default-Wert: 2)

Der aktuelle Wert wird bei **ps** nicht abgespeichert.



FUNKTION:

Physisches Sortieren des Inhaltsverzeichnisses einer Diskette (alphabetische Reihenfolge der Filenamen)

FORMAT:

va [Diskettenspezifikation:]

WIRKUNG:

Wird der Parameter 'Diskettenspezifikation' angegeben, wird das Inhaltsverzeichnis der betreffenden Diskette in alphabetische Reihenfolge sortiert, andernfalls dasjenige der zuletzt angesprochenen.

Ein schnelleres Durchsuchen des Inhaltsverzeichnisses wird möglich.

BEMERKUNGEN:

- Die angesprochene Diskette darf nicht mit Schreibschutz versehen sein.
- Das Systemfile PCOS.SAV steht auch nach **va** am Anfang des Inhaltsverzeichnisses.

BEISPIELE:

va 1: Das Inhaltsverzeichnis der Diskette in Station 1 wird physisch neu sortiert.



BEFEHL:

VCOPY
(volume copy)

vcopy. cmdFUNKTION:

Kopieren einer Diskette in einer Disketten-Station auf eine Diskette in der anderen Station

FORMAT:

<p>vc Disketten-Spezifikation des Originals:</p>	$\left\{ \begin{array}{l} / \\ , \end{array} \right\}$	<p>Disketten-Spezifikation der Empfängerdiskette:</p>
--	--	---

WIRKUNG:

Die 'Disketten-Spezifikation' des Originals gibt entweder die Station an, in der sich die Originaldiskette befindet oder die Diskettenbezeichnung der Originaldiskette. Ein eventuell vorhandenes Disketten-Passwort ist dabei anzugeben. Die Disketten-Spezifikation der Empfängerdiskette gibt entweder die Station an, in der sich die Empfängerdiskette befindet oder die Diskettenbezeichnung, den die Empfängerdiskette vor dem Kopieren hat. Der Befehl wird - falls erforderlich - geladen und am Bildschirm die Meldung:

Warning- vcopy deletes all files. Copy disk? (y/n)

angezeigt.

Ein Diskettenwechsel in beiden Stationen ist hier noch möglich. Wird **n** CR eingegeben, so wird die Ausführung des Befehls abgebrochen.

Wird **y** CR eingegeben, so wird die mit 'Disketten-Spezifikation des Originals' gekennzeichnete Diskette abschnittsweise gelesen

und auf die mit 'Disketten-Spezifikation der Empfängerdiskette' bezeichnete Diskette übertragen.

Es wird ausgegeben, welche Sektoren jeweils gerade gelesen bzw. geschrieben werden.

Nach dem Kopieren sind beide Disketten identisch und haben den Namen und das Password der Originaldiskette. Es wird dann **vcopy complete** gemeldet.

BEMERKUNGEN:

- Die Empfängerdiskette muß mindestens einmal mit **vf** formatiert worden sein, sonst ist es nicht möglich, sie später wieder zu kopieren.
- Vor der Ausführung kann **vc** mit **pl** in den Arbeitsspeicher geladen werden. Dadurch werden ständige Diskettenwechsel (Neu-Einlegen der Systemdiskette vor dem Aufruf von **vc**) überflüssig.
- Die Richtung des Kopiervorganges, also welche Diskette als Originaldiskette und welche als Empfängerdiskette aufgefaßt wird, ist nur durch die Reihenfolge der Parameter im Befehl bestimmt. Durch entsprechende Angabe in der Reihenfolge kann auch eine leere Diskette auf eine volle Diskette kopiert werden.
- Die Empfängerdiskette darf nicht mit Schreibschutz versehen sein; das Original darf Schreibschutz haben.
- Es empfiehlt sich, das Original vor dem Kopieren mit Schreibschutz zu versehen, um ein versehentliches Überschreiben zu vermeiden.
- Am Ende des Kopiervorganges dürfen die Disketten erst entnommen werden, wenn keine Kontrollampe mehr leuchtet.
- **vc** bewirkt ein physisches Duplikat der Originaldiskette. Alle Passwords, Protect-Kennzeichen, die Aufteilung der Files und

Leerplätze sind auf der Kopie in gleicher Weise wie im Original gespeichert.

- **vc** kann mit **c** nicht abgebrochen werden. Wird der Diskettenschacht der Empfängerdiskette geöffnet, wird mit Fehlermeldung (Error 57) abgebrochen. Die Empfängerdiskette darf so nicht mehr verwendet werden (neues **vf** oder **vc** nötig).
- Jede Diskette erhält bei der Initialisierung vom System identifizierende Parameter zugewiesen. Diese Parameter werden von **vc** nicht übertragen.
- Hat die Originaldiskette ein Disketten-Passwort, kann dieses bei der Eingabe weggelassen werden.
- Das Passwort der Empfängerdiskette muß angegeben werden.
- Wird

Disk contains proprietary files - use FCopy to back it up

gemeldet, sind entweder Files auf der Diskette mit Kopierschutz versehen, oder die zu kopierende Diskette ist niemals mit **vf** formatiert worden. Im ersten Fall kann **fc** zum Kopieren der nicht geschützten Files verwendet werden. Im zweiten Fall ist die Diskette nicht mehr kopierbar.

- Es ist nur möglich, eine 320-KB-Diskette auf eine 320-KB-Diskette zu kopieren bzw. eine 160-KB-Diskette auf eine 160-KB-Diskette; d.h. verschiedene Typen sind beim Kopiervorgang nicht möglich.

Zur Bestimmung des Diskettentyps vgl. **vf**.

BEISPIELE:

I.

Eingelegt sei in Station 0 eine Systemdiskette und in Station 1 die zu kopierende Diskette.

Es ist einzugeben:

pl vc (ist nur nötig, wenn dies nicht bereits vorher erfolgte).

Die Systemdiskette wird aus Station 0 genommen und durch die Empfängerdiskette ersetzt.

Eingabe von : **vc 1: 0:**

Es wird angezeigt:

Warning- vcopy deletes all files. Copy disk? (y/n)

Eingabe von: **y**

Die Diskette in Station 1 wird als Original genommen und auf die Diskette in Station 0 kopiert.

Am Ende wird

vcopy complete gemeldet.

II.

Eingabe von: **vc %n 0: 1:**

Die Diskette in Station 0 wird auf die Diskette in Station 1 kopiert. Es wird am Bildschirm nichts angezeigt. **vc** muß hierbei resident sein, wenn nicht in Station 0 die Systemdiskette einliegt. Ein Diskettenwechsel während des Kopiervorgangs ist nicht möglich, da

Warning- vcopy deletes all files. Copy disk? (y/n)

nicht gemeldet wird und die Antwort **y** vom System automatisch vorgenommen wird.

BEFEHL:

VDEPASS
(volume depass)

vdepass. cmd

FUNKTION:

Entfernen eines bekannten Passwords von einer Diskette

FORMAT:

vd { Diskettenbezeichnung Stationsbezeichnung } /Disketten-Password:
--

WIRKUNG:

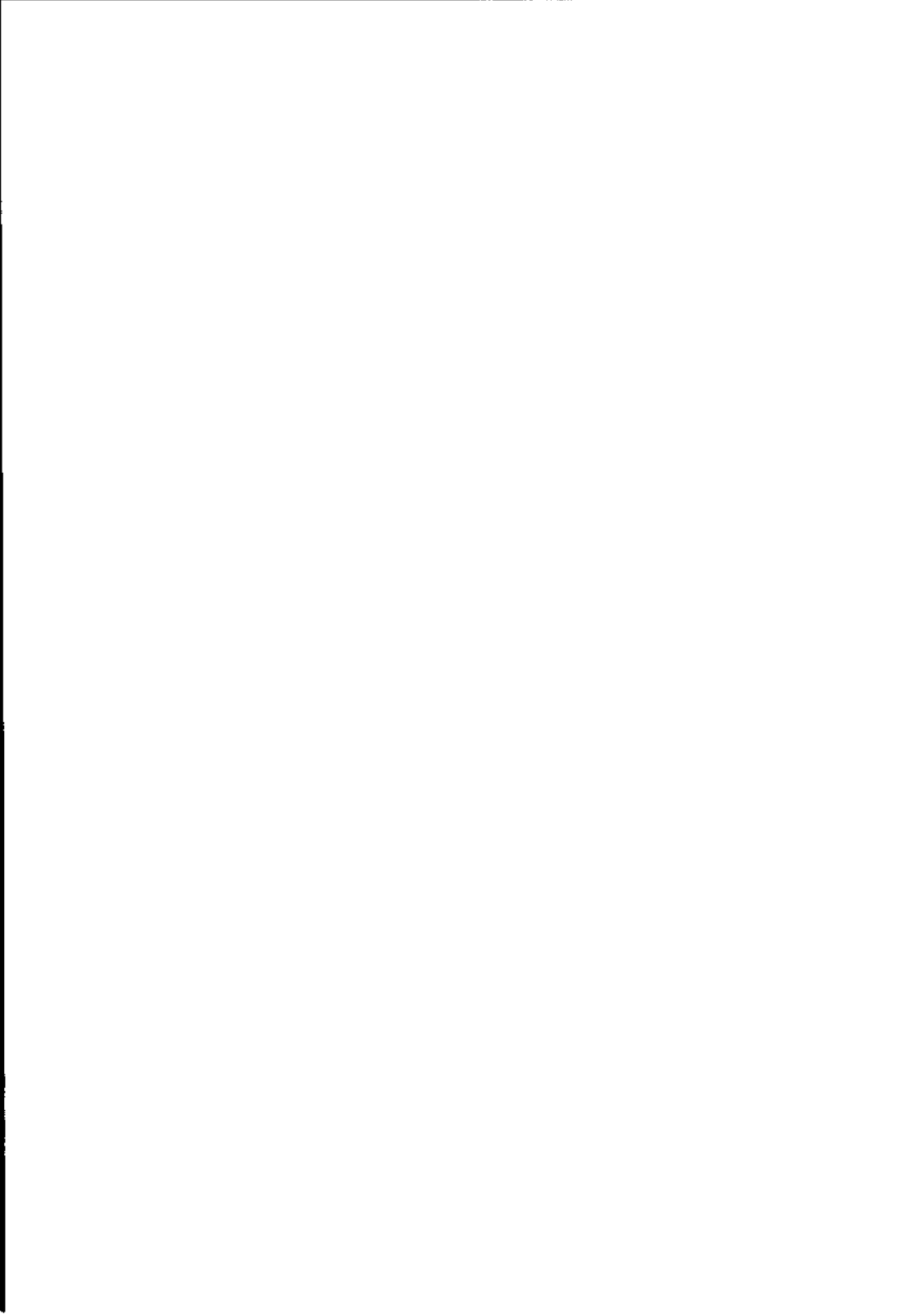
Das Password der angegebenen Diskette wird entfernt.

BEMERKUNGEN:

- Es muß das gültige Password für die Diskette eingegeben werden.
- Die Diskette darf nicht mit Schreibschutz versehen sein.
- Die Vergabe eines Disketten-Passwords erfolgt durch den Befehl **vp**.

BEISPIELE:

vd KONTEN/CHEF: entfernt das Password "CHEF" von der Diskette "KONTEN". Die Diskette kann wieder unter Angabe des Namens "KONTEN" ohne Password angesprochen werden.



FUNKTION:

physisches Formatieren einer Diskette bzw. der Hard-Disk; bei Hard-Disk Sperren neuer bei **vv** erkannter fehlerhafter Blöcke

FORMAT:

vf [%s] alte Disketten-Spezifikation: [, neue Disketten-bezeichnung]

Disketten-Spezifikation:	Disketten- oder Stationsbezeichnung für die angesprochene Diskette unter Angabe des evtl. vorhandenen Disketten-Passworts (mit : am Ende)
--------------------------	---

neue Disketten-bezeichnung:	neue Diskettenbezeichnung, ohne Disketten-Passwort (ohne : am Ende)
-----------------------------	---

%s:	Die Diskette soll als 160-KB-Diskette (einseitig, doppelte Dichte) formatiert werden, obwohl sie in einer 320-KB-Station einliegt.
-----	--

WIRKUNG:

Fehlt der Parameter 'neue Diskettenbezeichnung', wird der Diskette kein Namen vergeben. Ist der Parameter angegeben, erhält die Diskette den unter 'neue Diskettenbezeichnung' angegebenen Namen.

Es wird am Bildschirm der Text:

Warning: VFormat deletes all files. Format disk? (y/n)

angezeigt. Diskettenwechsel ist jetzt noch möglich. Wird **n** eingegeben, wird der Befehl nicht ausgeführt. Wird **y** eingegeben, wird das Programm fortgesetzt. Ist die Diskette noch nicht am M20 formatiert worden, sind meistens die zwei folgenden Fragen (**Diskette appears password protected. Format disk? (y/n)** und **Diskette is OPE formatted. Format disk anyway? (y/n)**) ebenfalls mit **y** zu beantworten. Danach werden alle Spuren der Diskette physisch formatiert. Es wird angezeigt, welche Spur formatiert wird. Am Ende wird **Formatting Complete** gemeldet.

BEMERKUNGEN:

- Jede Diskette und auch die Hard-Disk müssen vor ihrer erstmaligen Verwendung am M20 formatiert werden.
- Sind nach **vv** bei einer Hard-Disk neue Blöcke in die Liste der fehlerhaften Blöcke aufgenommen worden, muß - um diese im weiteren zu sperren - **vf** vorgenommen werden. Hat die Hard-Disk bereits Inhalt, ist dieser zuerst mit **fc** (bzw. der von OLIVETTI Deutschland verfügbaren Prozedur **save.cmd**) auf Diskette(n) zu sichern und nach **vf** mit **fc** bzw. **sa** wieder auf Hard-Disk zurückzuübertragen.
- Durch das Formatieren werden alle Files, die auf einer Diskette enthalten sind, gelöscht.
- Diejenigen Bereiche der Diskette, die für Files frei sind, werden vollständig mit dem Zeichen mit dem ISO-Code **Ø** belegt.
- Hat die zu formatierende Diskette ein Disketten-Passwort oder ist sie noch nie formatiert worden, erscheint nach der Eingabe von **y** auf die Frage **...Format disk? (y/n)** die weitere Frage **Diskette appears password protected. Format disk? (y/n)**,

die ebenfalls mit **y** zu beantworten ist. Ist die Diskette vorformatiert (neugekaufte Disketten sind dies meistens), kann auch die weitere Frage

Diskette is OPE formatted. Format disk anyway? (y/n)

erscheinen. Dies ist dann ebenfalls mit **y** zu beantworten.

- Wird das Formatierungsprogramm vorzeitig abgebrochen (z.B. durch **Ctrl-C**), wird mit

Formatting Aborted. WARNING: Diskette may be unusable.

davor gewarnt, diese Diskette zu verwenden.

- Der Diskettenschacht darf nicht geöffnet werden, solange die rote Kontrollampe leuchtet.
- Die Diskette darf nicht mit Schreibschutz versehen sein.
- **vf** enthält alle Funktionen von **vn**.
- **%s** ist nicht anzugeben, wenn die Diskette in einer 160-KB-Station formatiert wird.
- Es wird empfohlen, mehrere Disketten (z.B. eine Schachtel) in einem Zuge zu formatieren. Dazu sollte der Befehl **vf** zunächst mit dem PCOS-Befehl **pl** in den Arbeitsspeicher geladen und dann ausgeführt werden.

BEISPIELE:

- | | |
|-------------------------|--|
| vf 1: | Die Diskette in Station 1 wird formatiert |
| vf DATEN/geheim: | Die Diskette mit der Diskettenbezeichnung "DATEN" und dem Disketten-Password "geheim" wird formatiert |
| vf DATENALT:;NEU | Die Diskette mit der Disketten-Spezifikation "DATENALT:" wird formatiert und erhält den Disketten-Identifizier "NEU" |

vf %n 1:

Die Diskette in Station 1 wird formatiert. Es werden keine Informationen am Bildschirm ausgegeben. Die Eingabe von y wird vom System automatisch vorgenommen.

FUNKTION:

Ausgabe des ausführlichen Inhaltsverzeichnisses einer Diskette oder eines Auszuges über bestimmte Files mit gemeinsamen Eigenschaften

FORMAT:

```
v| [Disketten-Spezifikation:] [Filename(orauswahl)]
```

WIRKUNG:

Fehlt der Parameter 'Disketten-Spezifikation', wird das Inhaltsverzeichnis der zuletzt angesprochenen Diskette angezeigt. Ist dieser Parameter angegeben, wird das Inhaltsverzeichnis der durch 'Disketten-Spezifikation' bestimmten Diskette ausgegeben. Durch Filenamenauswahl kann auch nur eine Gruppe von Filenamen mit gemeinsamen Eigenschaften herausgesucht werden.

Folgende Informationen werden angezeigt:

- Nummer der Diskettenstation, auf die sich v| bezieht
- Diskettenbezeichnung der Diskette, falls vergeben (ohne evtl. Disketten-Passwort)

Für jedes File wird angezeigt:

- File-Identifizier der Files (ohne evtl. File-Passwort)
- Anzahl der vom File tatsächlich belegten Sektoren (SECTORS USED) (= höchster beschriebener Sektor)

- Anzahl der für das File reservierten Sektoren (SECTORS ALLOCATED)
- Anzahl der physisch getrennten Teile, in die das File zerlegt ist (EXTENTS)
- Kennzeichen, ob das File mit Schreibschutz versehen wurde (WP)
- Kennzeichen, ob das File durch ein File-Password geschützt ist (/PW)

Nach jeweils einer Bildschirmseite wird eine Zwischensumme, bzw. nach Anzeige aller Files die Endsumme durch folgende Informationen angezeigt:

- Anzahl der bisher angezeigten Files
- Anzahl der bisher insgesamt belegten Sektoren
- Anzahl der bisher insgesamt reservierten Sektoren
- Gesamtzahl der Teile, in die die bisher angezeigten Files aufgeteilt sind
- Anzahl der noch freien Sektoren auf der Diskette

Wurde das Ende einer Bildschirmseite erreicht, erfolgt die Meldung:

Hit any key to continue

Wird eine beliebige Taste gedrückt, wird die Anzeige des Inhaltsverzeichnisses fortgesetzt.

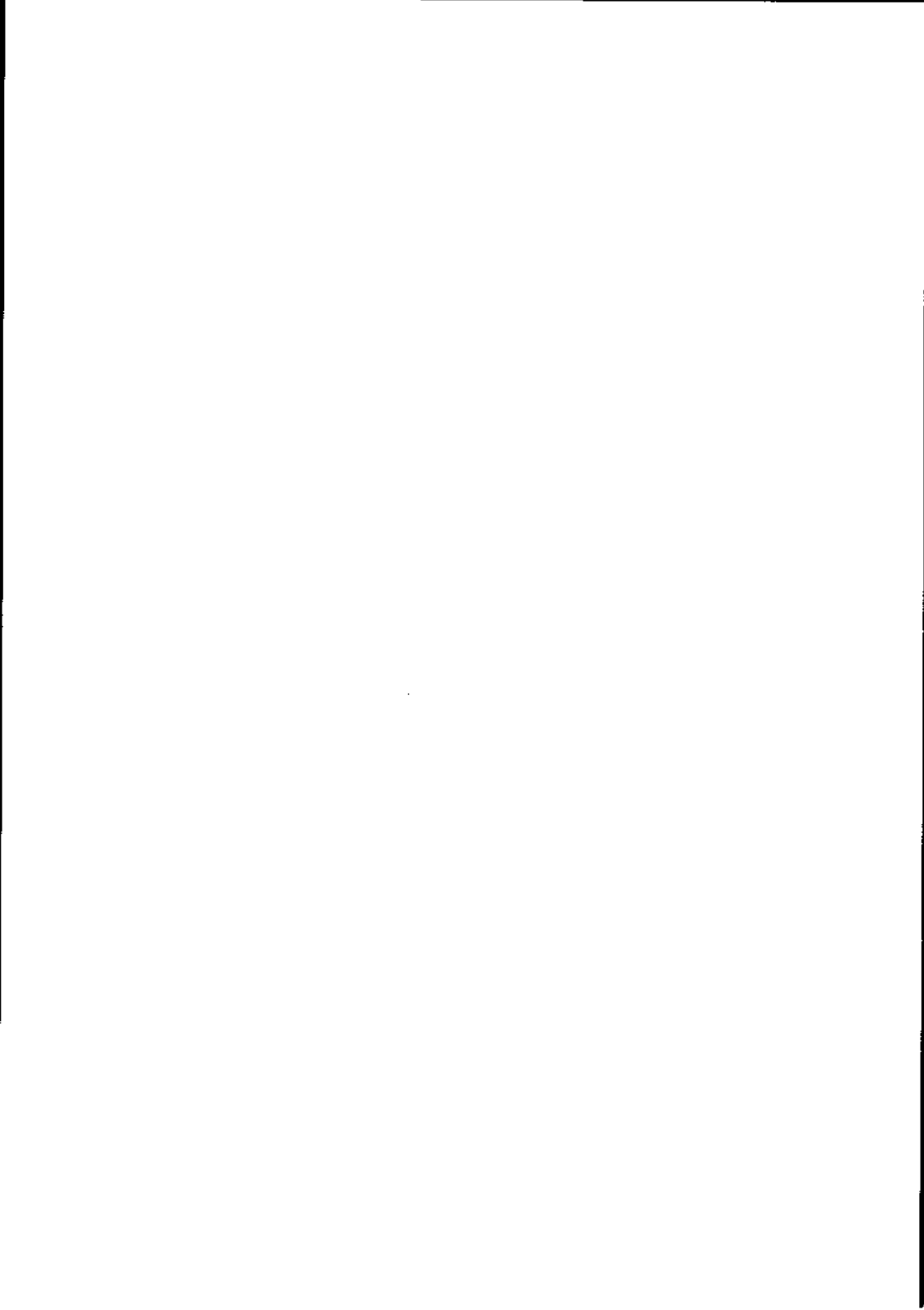
BEMERKUNGEN:

- Die Diskette wird durch **v1** nicht verändert und kann daher mit Schreibschutz versehen sein.
- Hat die Diskette ein Password, kann dieses beim Aufruf von **v1** weggelassen werden.

- Enthalten Filenamen File-Passwords, wird nur gekennzeichnet, daß ein Password vergeben wurde. Das Password selbst wird nicht ausgegeben.
- Eine Unterbrechung der Ausgabe kann mit **S** erfolgen. Betätigen einer beliebigen Taste bewirkt Fortsetzung.

BEISPIELE:

- v1 Ø:** gibt das Inhaltsverzeichnis der Diskette in Station Ø am Bildschirm aus.
- v1 %n DISK/PW4711:** gibt das Inhaltsverzeichnis der Diskette mit der Diskettenbezeichnung "DISK" und dem Disketten-Password "PW4711" am Bildschirm aus, wobei keine Unterbrechung aufgrund voller Bildschirmseiten erfolgt.
- v1 1: +dprt:** gibt das Inhaltsverzeichnis der Diskette in Station 1 auf dem Bildschirm und dem Drucker aus.
- v1 Ø:* .cmd** gibt alle Files in Station Ø mit .cmd am Ende des Namens aus
- v1 Ø:?c*** gibt alle Files in Station Ø mit dem Buchstaben c an zweiter Stelle des Namens aus



FUNKTION:

Kopieren einer Diskette in nur einer Diskettenstation (Station Ø:)

FORMAT:

<code>vm</code> [Disketten-Spezifikation des Originals:] [Disketten-Spezifikation 'der Zieldiskette:]

WIRKUNG:

Nachdem `vm` gesucht und geladen wurde, wird gemeldet:

Warning- vmove deletes all files and PCOS. Vmove disk? (y/n)

Die Eingabe von `n` führt zum Abbruch. Die Eingabe `y` (ohne CR) bewirkt den sofortigen Programmbeginn. Die Bedienung im Bildschirm gibt Auskunft, was zu tun ist.

"Please insert source diskette in drive Ø"=

"Bitte zu kopierende Diskette in Station Ø einlegen"

"Press any key when ready"=

"Beliebige Taste drücken, wenn fertig"

"Please insert destination diskette in drive Ø"=

"Bitte Zieldiskette in Station Ø einlegen".

Während Kontrollampen leuchten, dürfen die Schächte noch nicht geöffnet werden.

Zwischen den 3 Teilphasen wird gemeldet, was geschieht.

Am Ende ist der Inhalt der zu kopierenden Diskette vollständig auf die Zieldiskette übertragen. Da das Betriebssystem aus dem Arbeitsspeicher gelöscht wurde, ist am Ende die Systemdiskette einzulegen und neu das System zu laden, wenn die Frage **Do you want to make more copies? (y/n)** mit **n** beantwortet wurde. Wird die Frage mit **y** beantwortet, kann nochmals kopiert werden.

BEMERKUNGEN:

- Der Kopiervorgang kann nur in Station Ø durchgeführt werden; die Angabe einer Stationsbezeichnung bei **vm** ist deshalb unmöglich.
- Hat die Zieldiskette ein Password, müssen sowohl 'Disketten-Spez. des Originals' als auch 'Disketten-Spez. der Zieldiskette' (mit Password) angegeben werden. Hat nur die Originaldiskette ein Password, reicht 'Disketten-Spez. des Originals' (mit Password).
- Original- und Zieldiskette müssen vom gleichen Typ sein; d.h., daß z.B. eine 160-KB-Diskette nicht auf eine 320-KB-Diskette kopiert werden kann.
- Der Disketteninhalt wird rein physisch kopiert; das Inhaltsverzeichnis, der Diskettenname, die Passwords und die Verteilung der Files auf der Zieldiskette sind identisch mit der zu kopierenden Diskette.
- **pu** kann nicht eingesetzt werden, um **vm** wieder aus dem Arbeitsspeicher zu entfernen.
- Wird das Ende des Kopiervorganges gemeldet, kann erneut kopiert werden; wird die entsprechende Frage mit **n** beantwortet, muß das Betriebssystem neu geladen werden. Dies geschieht durch Einlegen der Systemdiskette und Betätigen einer beliebigen Taste (**press any key when ready**).

- Wird während des Kopiervorgangs versehentlich die falsche Diskette eingelegt, wird dies vom Programm gemeldet ("Source diskette still in drive 0" bzw. "Destination diskette still in drive 0").

BEISPIELE:

vm Der Kopiervorgang wird in Station 0: durchgeführt.



FUNKTION:

logisches Löschen des gesamten Inhaltes einer Diskette (Löschen des Inhaltsverzeichnisses)

FORMAT:

vn	[%s]	alte Disketten-	[neue Disketten-
		Spezifikation:	,	bezeichnung

Disketten-
Spezifikation: Disketten- und Stationsbezeichnung für die angesprochene Diskette unter Angabe des evtl. vorhandenen Disketten-Passworts (mit : am Ende)

neue Disketten-
bezeichnung: neue Diskettenbezeichnung, ohne Disketten-Passwort (ohne : am Ende)

%s: Die Diskette wird als 160-KB-Diskette behandelt (einseitig, doppelte Dichte), obwohl sie in einer 320-KB-Station einliegt.

WIRKUNG:

Es wird am Bildschirm der Text:

Warning- vnew deletes all files. Initialize disk? (y/n)
angezeigt.

Wird **n** eingegeben, wird die Ausführung des Befehles abgebrochen.
Wird **y** eingegeben, werden alle Sektoren als freie Sektoren
vermerkt und ein leeres Inhaltsverzeichnis eingerichtet.
Fehlt der Parameter 'neue Diskettenbezeichnung', wird der
Diskette kein Name vergeben. Ist dieser Parameter angegeben,
erhält die Diskette den unter 'neue Diskettenbezeichnung'
angegebenen Namen.

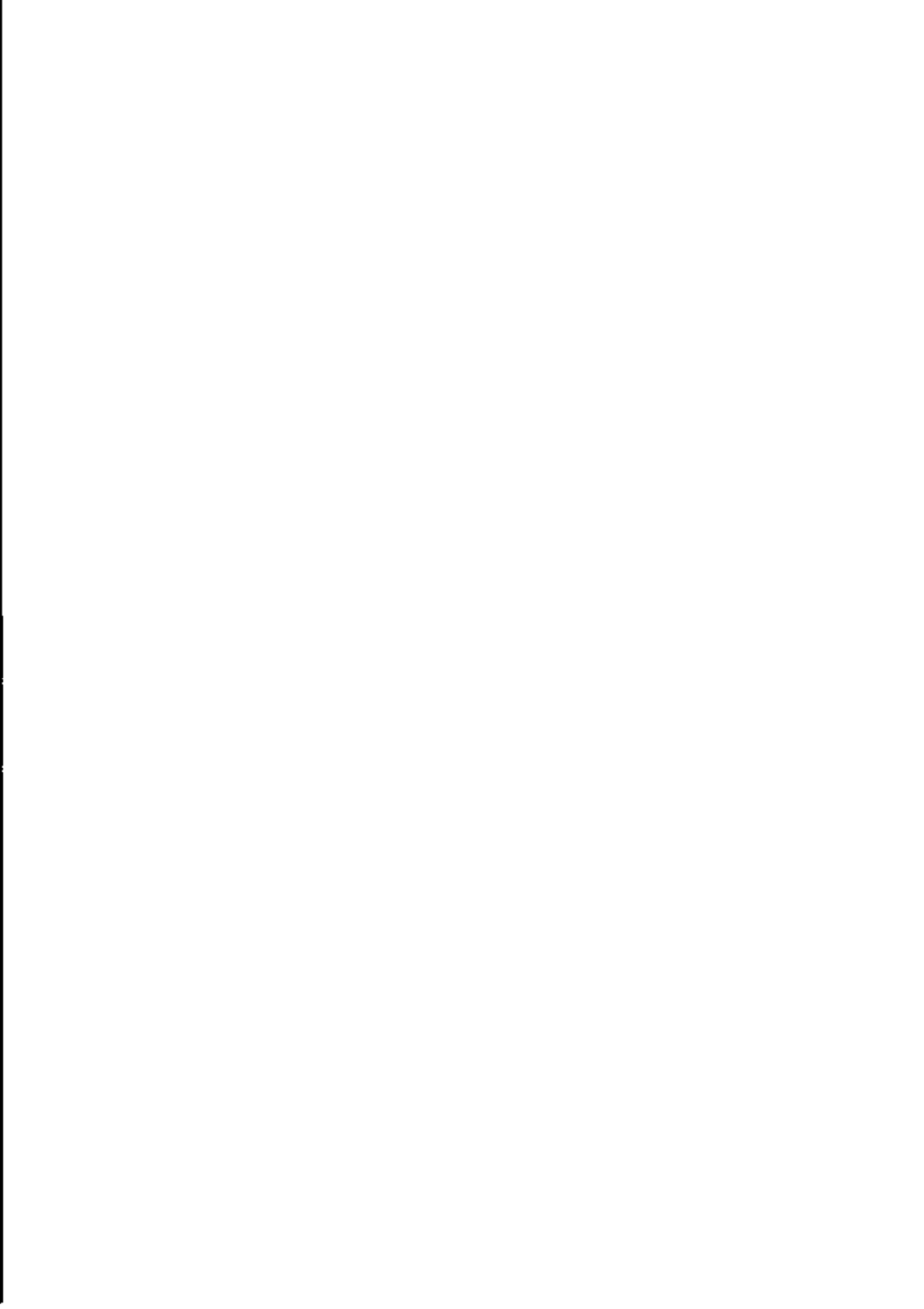
BEMERKUNGEN:

- Ein evtl. vorhandenes Disketten-Passwort ist anzugeben.
- Es können nur solche Disketten mit **vn** initialisiert werden, die
bereits früher mit **vf** formatiert wurden. **vn** kann also nur zur
Wiederverwendung von bereits gebrauchten Disketten angewendet
werden.
- Im Gegensatz zu **vf** belegt **vn** die für Files zur Verfügung
stehenden Bereiche der Diskette nicht mit dem Zeichen mit dem
ISO-Code Ø, sondern löscht nur das Inhaltsverzeichnis und gibt
alle Sektoren frei. Dies führt bei Programmen, die den **CHR\$(Ø)**
abfragen, möglicherweise zu Fehlern. Deshalb wird auch bei
Wiederverwendung von Disketten dringend **vf** zum Löschen empfoh-
len.
- Die Diskette darf nicht mit Schreibschutz versehen sein.
- **vf** führt alle Funktionen von **vn** ebenfalls durch. Es ist daher
nicht erforderlich, für neue Disketten beide Befehle auszu-
führen.
- **%s** ist nicht anzugeben, wenn die Diskette in einer 160-KB-Station einliegt.
- **vn** wird besonders bei der Neuinstallation einer Hard-Disk
benötigt (vgl. Kapitel 0.2.7).

BEISPIELE:

vn 1: Das Inhaltsverzeichnis der in Station 1 eingelegten Diskette wird gelöscht. Die Diskette ist neu initialisiert und hat keinen Namen.

vn 1:,NEUDISK Die Diskette in Station 1 wird neu initialisiert und erhält den Namen "NEUDISK".



FUNKTION:

erstmalige Vergabe eines Passwords für eine Diskette oder Änderung eines Disketten-Passwords

FORMAT:

vp Disketten-Spezifikation: ,neues Disketten-Password

WIRKUNG:

Der durch 'Disketten-Spezifikation' bestimmten Diskette wird das durch 'Disketten-Password' angegebene Password zugeordnet.

BEMERKUNGEN:

- 'Disketten-Spezifikation' kann entweder die Stationsbezeichnung oder die Diskettenbezeichnung enthalten; ein bereits vorhandenes Disketten-Password ist anzugeben.
- Die Diskette darf nicht mit Schreibschutz versehen sein.
- Wird einer Diskette ein Disketten-Password zugeordnet, kann auf diese Diskette nur noch unter Angabe dieses Passwords zugegriffen werden. Das Password ist unbedingt beim ersten Zugriff auf die Diskette nach dem Einlegen anzugeben. Es darf bei weiteren Zugriffen entfallen, solange die Diskette eingelegt bleibt und/oder für diese Diskettenstation nicht eine andere Diskette angesprochen wird.

- Soll das Disketten-Passwort wieder entfernt werden, ist dies mit dem Befehl **vd** unter Angabe des bisherigen Disketten-Passworts möglich. PCDS enthält keine Möglichkeit, Passworts sichtbar zu machen. Es ist daher empfehlenswert, organisatorische Maßnahmen zu treffen, die sowohl vor dem Vergessen als auch der unbefugten Verwendung von Passworts schützen.
- Die verdeckte Eingabe von Passworts ist mit **G** möglich. Das Zeichen **G** wird dadurch nicht zum Bestandteil des Disketten-Passworts.

BEISPIELE:

vp KONTEN:,CHEF Der Diskette mit dem Disketten-Identifizier "KONTEN" (bisher ohne Disketten-Passwort) wird das Passwort "CHEF" zugeordnet.

vp TEWI237/frage:,antwort Der Diskette mit der Diskettenbezeichnung TEWI237 und dem Disketten-Passwort "frage" wird das neue Disketten-Passwort "antwort" zugeordnet.

vp Ø:,ersterPass Die Diskette in Station Ø, die bisher kein Passwort hatte, erhält das Disketten-Passwort "ersterPass" zugeordnet.

BEFEHL:**VQUICK**
(volume quick)**vquick. cmd**FUNKTION:

Ausgabe aller Filenamen auf einer Diskette oder einer Auswahl davon (schnelles Inhaltsverzeichnis) sowie der Anzahl freier Sektoren

FORMAT:

```
vq [Disketten-Identifizier:] [Filename(enauswahl)]
```

WIRKUNG:

Von der durch 'Disketten-Identifizier' spezifizierten Diskette (ohne Angabe von der zuletzt angesprochenen) wird eine Zusammenstellung der Filenamen ausgegeben. Durch Filenamenauswahl kann auch nur eine Gruppe von Filenamen mit gemeinsamen Eigenschaften herausgesucht werden.

BEMERKUNGEN:

- Die Angabe eines evtl. vergebenen Disketten-Passworts kann unterbleiben.
- Vor den Filenamen (bei 64 x 16-Bildschirm in vier Spalten, bei 80 x 25-Bildschirm in fünf) werden Stationsbez., Diskettenbez. (falls vergeben) und die Anzahl freier Sektoren auf der Diskette ausgegeben.
- Der BASIC-Befehl **FILES** hat die gleiche Wirkung wie **vq**.

BEISPIELE:

vq 0:* .cmd

Alle Filenamen auf der Diskette in Station 0 mit der Typangabe .cmd werden ausgegeben.

vq 1:?s*

Alle Filenamen auf der Diskette in Station 1, deren zweiter Buchstabe ein s ist, werden ausgegeben.

BEFEHL:**VRENAME**
(volume rename)**rename. cmd**FUNKTION:

Vergabe eines Namens (Diskettenbezeichnung) oder Ändern des bestehenden Namens einer Diskette

FORMAT:

vr Disketten-Spezifikation: [,neue Diskettenbez.]

Disketten-Spezifikation: Disketten- oder Stationsbezeichnung für die angesprochene Diskette unter Angabe des evtl. vorhandenen Disketten-Passworts (mit : am Ende)

neue Diskettenbezeichnung: neue Diskettenbezeichnung, ohne Disketten-Passwort (ohne : am Ende)

WIRKUNG:

Enthält 'Disketten-Spezifikation' die Bezeichnung einer Station, erhält die dort einliegende Diskette den durch 'neue Diskettenbezeichnung' bestimmten Namen. Enthält 'Disketten-Spezifikation' eine Diskettenbezeichnung (Name einer Diskette), wird dieser in den durch 'neue Diskettenbezeichnung' bestimmten geändert. Fehlt der Parameter 'neue Diskettenbezeichnung', wird die alte Diskettenbezeichnung entfernt. Die Diskette ist dann nur noch durch die Station, in der sie einliegt (+ ein evtl. noch vorhandenes Disketten-Passwort), identifiziert.

BEMERKUNGEN:

- **vr** bezieht sich nur auf die Diskettenbezeichnung. Ein eventuell vergebenes Disketten-Passwort bleibt unverändert.
- Eine Diskette, die bisher noch keinen Namen hatte, kann mit einem Namen versehen werden, indem als erster Parameter die Stationsbezeichnung angegeben wird.
- Die Diskette darf nicht mit Schreibschutz versehen sein.

BEISPIELE:

vr Ø:,KONTEN Die in Station Ø eingelegte Diskette erhält den Namen "KONTEN"

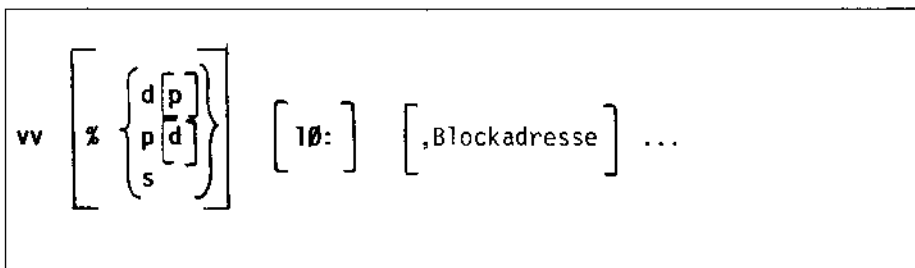
vr KONT:,ALTK Die Diskette mit dem Namen "KONT" wird umbenannt in "ALTK"

**vr KONTEN/CHEF:,
ALTKONTEN** Die Diskette mit dem Namen "KONTEN" und dem Passwort "CHEF" wird umbenannt in die Diskette "ALTKONTEN" mit dem Passwort "CHEF". Die neue Disketten-Spezifikation lautet daher "ALTKONTEN/CHEF:"

vc %n Orig: 1: Die in Station Ø liegende Diskette mit dem Namen "Orig" wird auf die Diskette in Station 1 kopiert; die Diskette in Station 1 erhält den Namen "Sich". Alle Bildschirmausgaben werden unterdrückt. Der Kopierlauf ist automatisch.
vr %n 1:, Sich

FUNKTION:

Überprüfung der Hard-Disk auf fehlerhafte Sektoren (Blöcke) und Aufnahme derselben in die Liste der fehlerhaften Blöcke auf der Hard-Disk; ggf. Vereinbarung selbstgewählter Blöcke als fehlerhaft

FORMAT:WIRKUNG:

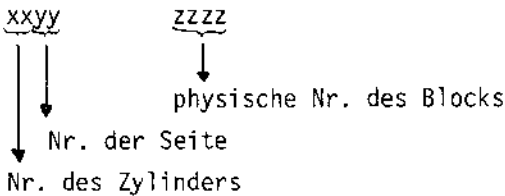
Die Hard-Disk wird auf fehlerhafte Blöcke untersucht. Auf der Hard-Disk wird von PCOS auf Zylinder 0, Seite 0, Block 1 eine Liste der fehlerhaften Blöcke geführt. **vv** nimmt neu als fehlerhaft erkannte (bzw. als solche "definierte") Blöcke in diese auf.

In allen Fällen werden diese neuen als fehlerhaft vermerkten Blöcke jedoch nicht automatisch gesperrt, sondern nur durch ein späteres **vf**.

Ist **%d** angegeben, wird ein destruktiver Text durchgeführt. Der bisherige Hard-Disk-Inhalt wird zerstört und am Ende die Anzahl aller (auch der früher evtl. schon erkannten) fehlerhaften Blöcke ausgegeben, nicht aber deren Adressen.

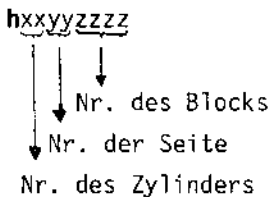
Ist **%p** angegeben, wird ein nicht-destruktiver Test durchgeführt. Die neuen fehlerhaften Blöcke werden am Ende mit ihren Adressen aufgelistet.

Aufbau dieser Adresse (hexadezimal):



Ist **%s** angegeben, muß am Ende in einer Liste von Blockadressen angegeben werden, welche Blöcke zusätzlich in die Liste der fehlerhaften Blöcke aufgenommen werden sollen (z.B., um von **vv** nicht als fehlerhaft erkannte, aber als fehlerhaft vermutete Blöcke explizit zu sperren). Auch dermaßen zusätzlich als fehlerhaft definierte Blöcke sind erst nach späterem **vf** ausgeschaltet.

Format einer 'Blockadresse' (alle Werte hexadezimal):



BEMERKUNGEN:

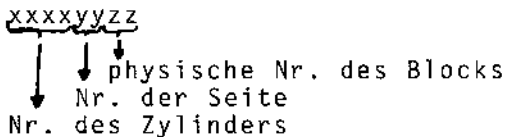
- Wird kein %-Parameter angegeben, wird %p (nicht-destruktiver Test) angenommen.
- Sind keine neuen fehlerhaften Blöcke durch vv erkannt worden, kann vf danach unterbleiben.
- Um neu in die Liste der fehlerhaften Blöcke aufgenommene Blöcke zu sperren, muß die Hard-Disk mit vf vollständig neu formatiert werden. Sind auf der Hard-Disk bereits Files gespeichert, müssen diese zuvor per fc (bzw. sa) auf Diskette(n) gesichert und nach vf wieder mit fc (bzw. sa) rückübertragen werden.
- Der destruktive Test ist schneller als der nicht-destruktive.
- Um den Ablauf des nicht-destruktiven Tests etwas zu beschleunigen, kann die Bildschirm-Ausgabe darüber, welcher Block gerade getestet wird, durch %n unterdrückt werden.
- Folgende %...-Kombinationen sind zulässig:
 - %dp:** erst destruktiver Test, dann Ausgabe der neuen fehlerhaften Blöcke
 - %pd:** erst Ausgabe der alten Liste fehlerhafter Blöcke, dann destruktiver Test (mit Ausgabe der Anzahl jetzt fehlerhafter Blöcke)
 - %sp:** erst Aufnehmen zusätzlicher Blöcke in die Liste fehlerhafter Blöcke, dann nicht-destruktiver Test mit Ausgabe der neuen fehlerhaften Blöcke
 - %ps:** erst Ausgabe der Liste fehlerhafter Blöcke (nicht-destruktiver Test), dann Aufnehmen zusätzlicher Blöcke in diese Liste
- vv kann nur auf einer vorher einmal mit vf formatierten Hard-Disk durchgeführt werden. Andernfalls wird Error 111 gemeldet.

BEISPIELE:

- vv l0:** nicht-destruktiver Test, Update der Liste fehlerhafter Blöcke
- vv %nd l0:** destruktiver Test, Update der Liste fehlerhafter Blöcke, nur Fehlermeldungen werden ausgegeben
- vv %s h0l0l000l:** Zylinder 1, Seite 1, Block 1 wird explizit in die Liste der fehlerhaften Blöcke aufgenommen

Zu vv-Befehl

Die Adresse der Blöcke ist in der bisherigen Beschreibung fehlerhaft beschrieben. Sie ist stattdessen wie folgt aufgebaut (hexadezimale Werte für die einzelnen Stellen):



Damit ergibt sich das

Format einer 'Blockadresse' (alle Stellen hexadezimal):



Zwischen die einzelnen Zeichen darf kein Blank gesetzt werden. Ist xxxx nicht angegeben, wird 0000 angenommen.

Bei allen vv-Befehlen, die den Parameter %s verwenden, muß mindestens eine solche Blockadresse angegeben werden.

Beispiel:

vv %s 10: h00020ba4

Zylinder 2, Seite 11, Block 164 wird explizit in die Liste der fehlerhaften Blöcke aufgenommen.

Das letzte Beispiel im PCOS-Handbuch ist falsch.

Mit %s nach vf ausgeschaltete Sektoren können nicht wieder freigegeben werden.

Ist %p nach vv angeführt, wird nur die Liste der fehlerhaften Sektoren in der Liste der fehlerhaften Blöcke ausgegeben.

Wird bei vv keiner der Parameter %p bzw. %d angegeben, wird ohne Vorwarnung ein destruktiver Test ausgeführt (nicht, wie im PCOS-Handbuch angegeben, ein nicht-destruktiver).

FUNKTION:

Aktivieren des Original-Zeichensatzes oder eines Bildschirmzeichensatzes, der mit **rf** in ein PCOS-Textfile gespeichert und ggf. mit Hilfe des Full-Screen-Editors (**ed**) geändert wurde.

FORMAT:

wf [Tastaturfile]

WIRKUNG:

Wird nur **wf** eingegeben, wird derjenige Zeichensatz aktiviert, der beim letzten Laden des Betriebssystems Gültigkeit hatte (Original-Zeichensatz) und alle sonstigen Tastaturtabellen aus dem Arbeitsspeicher entfernt. Ist ein 'Tastaturfile' angegeben, wird derjenige Zeichensatz aktiviert, der in dem genannten 'Tastaturfile' beschrieben ist.

BEMERKUNGEN:

- Im Tastaturfile ist gespeichert:
 - a) welche nationale Tastatur die Grundlage zur Erstellung des in 'Tastaturfile' beschriebenen Bildschirmzeichensatzes war
 - b) wieviele Zeichen in 'Tastaturfile' definiert sind (mindestens 95, maximal 190)
 - c) aus wievielen Elementarpunktzeilen ein Zeichen besteht (vorläufig immer 10)

d) welche 8 x 10-Punktmatrix erzeugt werden soll, wenn eine Taste gedrückt wird, deren Tastencode ein bestimmter ISO-Code zugeordnet ist.

- Das 'Tastaturfile' wird erstellt durch Abspeichern eines aktuell gültigen Zeichensatzes (**rf**) und ggf. Ändern der Punktmatrizen, die bestimmten ISO-Codes zugeordnet sind, und zwar mit Hilfe des Full-Screen-Editors (PCOS-Befehl **ed**).
- Die standardmäßige Zuordnung zwischen Tastencode und ISO-Code ist abhängig von der mit **sl** gewählten nationalen Tastatur (vgl. Kapitel 3.1 und 3.2).
- Die Zuordnung eines ISO-Codes zu einem Tastencode kann mit **ck** abgefragt bzw. geändert werden.
- Am Drucker bzw. über V.24- oder IEEE-488-Schnittstelle werden weiterhin Zeichen mit denjenigen Punktmatrizen ausgegeben, die für die betreffenden Einheiten und den jeweiligen ISO-Code gültig sind.
- Das 'Tastaturfile' belegt im Arbeitsspeicher Platz, bis entweder
der Befehl **wf** ohne Parameter gegeben wird oder
das Betriebssystem neu geladen wird.
Vor Aktivieren eines neuen Zeichensatzes durch **wf** empfiehlt sich deshalb, **wf** ohne Parameter 'Tastaturfile' abarbeiten zu lassen.
- Die Beschreibung des Aufbaus des 'Tastaturfiles' findet sich in Kapitel 3.3.

BEISPIELE:

wf Ø:ZEICHENS1.font

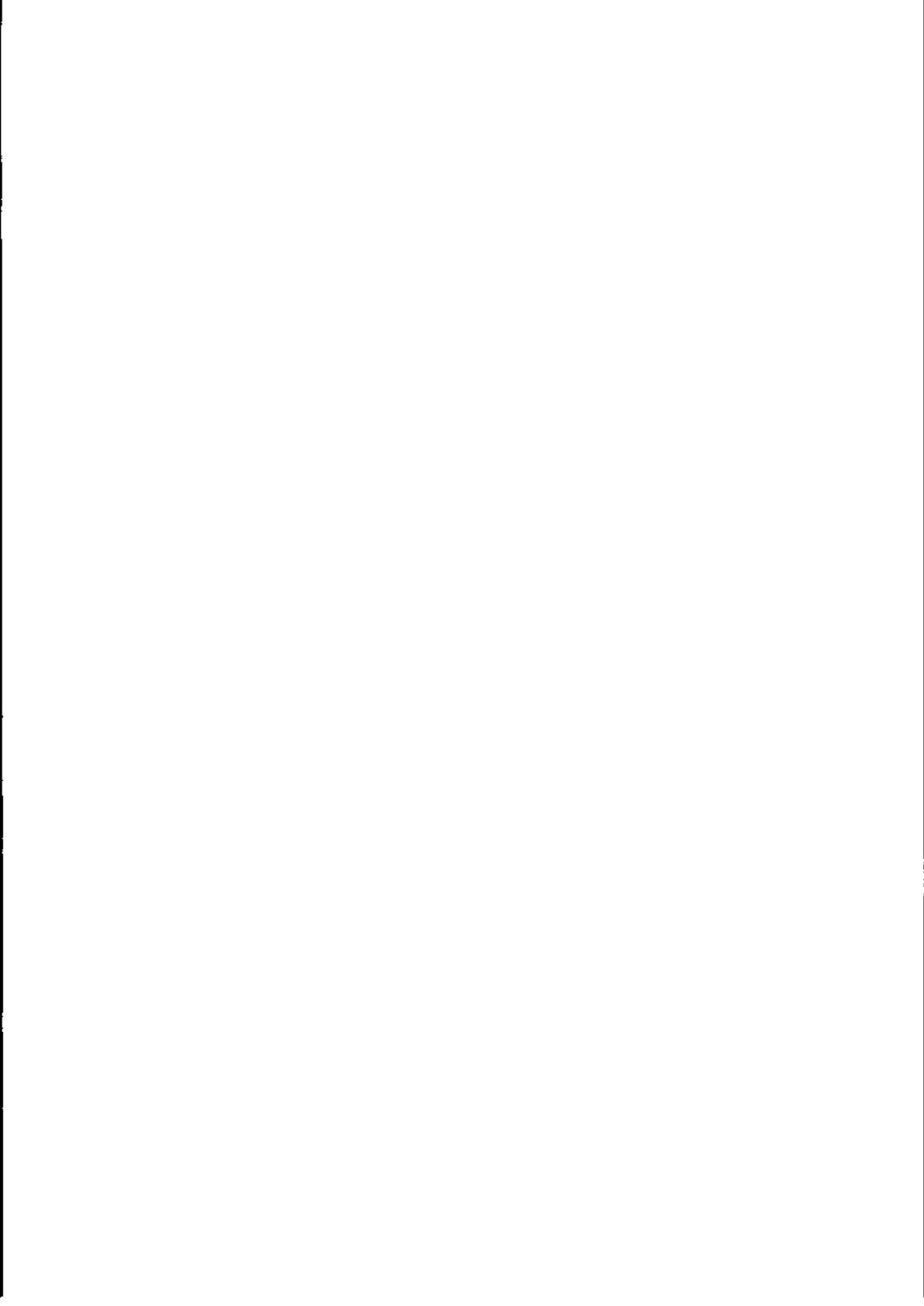
aktiviert den im PCOS-Textfile ZEICHENS1.font auf der Diskette in Station Ø festgelegten Bildschirmzeichensatz.

wf

desaktiviert den zuletzt aktivierten Zeichensatz und entfernt ihn aus dem Arbeitsspeicher; der beim Laden des Betriebssystems gültige Zeichensatz wird aktiviert.

wf Ø:ZEICHENS2.font

aktiviert den im PCOS-Textfile ZEICHENS2.font auf der Diskette in Station Ø festgelegten Bildschirmzeichensatz.



2.7 Gegliederter Überblick

1. DISKETTEN-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck	File- namen- auswahl
Kurz- form	mnemonischer Name	Typ		
bv	BVOLUME	3)	Abfrage File- oder Diskettenname	6)
va	VALPHA	2)	Inhaltsverzeichnis alphabetisch ordnen	
vc	VCOPY	2)	Kopieren Diskette (2 Laufwerke)	
vd	VDEPASS	2)	Löschen Disketten-Passwort	
vf	VFORMAT	2)	Formatieren Diskette	
vl	VLIST	2)	volles Inhaltsverzeichnis	x
vm	VMOVE	3)	Kopieren Diskette (1 Laufwerk)	9), 13)
vn	VNEW	2)	log. Löschen Diskette	
vp	VPASS	2)	Vergabe Disketten-Passwort	
vq	VQUICK	2)	schnelles Inhaltsverzeichnis	x
vr	VRENAME	2)	Umbenennen Diskette	
vv	VVERIFY	2)	Überprüfung Hard-Disk	
se	SETV	2)	Abfragen/Aktivieren Station	

2. FILE-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck	File- namen- auswahl
Kurz- form	mnemonischer Name	Typ		
fc	FCOPY	2)	Kopieren File/s	x
fd	FDEPASS	2)	Löschen File-Passwort	x
ff	FFREE	2)	File verkürzen	x
fk	FKILL	2)	log. Löschen File	x
fl	FLIST	2)	Ausgabe File-Inhalt	x
fm	FMOVE	2)	Kopieren File (1 Laufwerk)	
fn	FNEW	2)	Anlegen File	
fp	FPASS	2)	Vergabe File-Passwort	x
fr	FRENAME	2)	Umbenennen File	
fu	FUNPROT	2)	Aufheben File-Schreibschutz	x
fw	FWPROT	2)	Vergabe File-Schreibschutz	x
rk	RKILL	2)	Reaktivieren gelöschtes File	

3. PCOS-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck
Kurz- form	mnemonischer Name	Typ	
pl	PLOAD	1)	Residentmachung eines PCOS-Befehls
pr	PRUN	2)	Laden bestimmtes Betriebssystem
ps	PSAVE	2)	Ab speichern aktuelle PCOS-Konfiguration
pu	PUNLOAD	1)	Entfernen eines resident-gemachten PCOS-Befehls aus Arbeitsspeicher

4. DEFAULTWERT-ÄNDERNDE P C O S - BEFEHLE

Schlüsselwort			Zweck
Kurz- Form	mnemonischer Name	Typ	
sb	SBASIC	2)	für den BASIC-Interpreter 5)
sc	SCOMM	2)	für RS-232-Übertragung 10)
sd	SDEVICE	2)	für Name einer peripheren Einheit
sf	SFORMAT	2)	für Drucker
sl	SLANG	2)	für nationalen Zeichensatz 8)
ss	SSYS	2)	für PCOS
{+} {-}	{s} {d}	1)	für Ein/Ausgabe-Medium

5. TASTATUR-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck
Kurz- Form	mnemonischer Name	Typ	
ck	CKEY	2)	Ändern Tastenfunktion 7)
lt	LTERM	1)	Abfrage Abschlußtaste 6)
pk	PKEY	2)	Zuordnung Zeichen(folge) auf Taste
rf	RFONT	2)	Ab speichern aktuellen Zeichensatz 8)
wf	WFONT	2)	Aktivieren abgespeicherten Zeichen- satz 8)

6. GRAPHIK-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck
Kurz-Form	mnemonischer Name	Typ	
la	LABEL	2)	Zeichen-Plotten
ls	LSCREEN	2)	Bildschirmtext ausdrucken 7)
sp	SPRINT	2)	Hardcopy 7)

7. SCHNITTSTELLEN-ORIENTIERTE P C O S - BEFEHLE

Schlüsselwort			Zweck
Kurz-Form	mnemonischer Name	Typ	
ci	CI	3)	Kommunikation mit RS232-Driver 6), 12), 13)
ie	IEEE	3)	Laden IEEE-488-Paket 10), 13)
rs	RS232	3)	Laden RS232(V24)-Paket 10), 13)

8. PROGRAMMIERHILFEN

Schlüsselwort			Zweck
Kurz-Form	mnemonischer Name	Typ	
ba	BASIC	2)	Aufruf BASIC-Interpreter 8), 10)
dc	DCONFIG	2)	Anzeige Hardware-/Speicher-Konfiguration
ed	EDIT	2)	Aufruf Full-Screen-Editor 8), 9), 13)
ep	EPRINT	3)	Anzeige Bedeutung PCOS-Fehler 13)
kb	KBOARD	2)	kontrollierter Input aus BASIC 6)
is	ISAM	3)	Laden ISAM-Paket 8)
sa	SAVE	2)	Sichern Diskette-HD + umgekehrt 8), 11)


Bemerkungen:

- 1) immer resident
- 2) Typ .cmd
- 3) Typ .sav

- 5) wertändernd nur in PCOS-Ebene
- 6) nur in BASIC-Ebene einsetzbar
- 7) bei bestimmten Parametern nur in BASIC-Ebene einsetzbar
- 8) Hilfsdatei(en) notwendig
- 9) anschließendes Neuladen notwendig
- 10) möglichst nur in PCOS-Ebene einsetzen
- 11) Aufruf aus Prozedur möglich
- 12) nur nach **rs** und **sc**
- 13) entladen mit **pu** nicht möglich

Tabelle der Sonderfunktionen

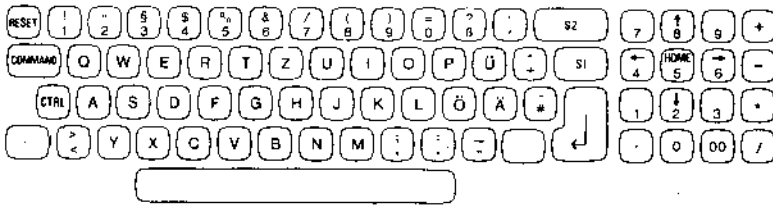
Alle Angaben sind hexadezimal.

CODE (hexadezimal)	standardmäßige Taste(nkombination)	Beschreibung
A0	RESET	Logischer Reset
A1	-	(reserviert)
A2	C	Programmabbruch
A3	S	Pausenfunktion
A4	-	Feststellfunktion num. Tastaturblock
A5	COMMAND -	Feststellfunktion alphanum. Tastaturblock
A6	00	zweimal Zeichen, das dem ISO-Code 30 (hexadezimal) zugeordnet ist
A7		Abschlußtaste (CR im Tastaturpuffer, 0 im LTERM-Puffer)
A8	S1	Abschlußtaste (CR im Tastaturpuffer, 1 im LTERM-Puffer)
A9	S2	Abschlußtaste (CR im Tastaturpuffer, 2 im LTERM-Puffer)
AD	-	(reserviert)
AE	-	(reserviert)
AF	-	Null-Operation

G kann nicht un belegt werden; **i** = ISO-Code 09,

H = ISO-Code 08.

Deutsche Tastatur (neu, DIN 2137)



alphanumerischer Tastaturblock

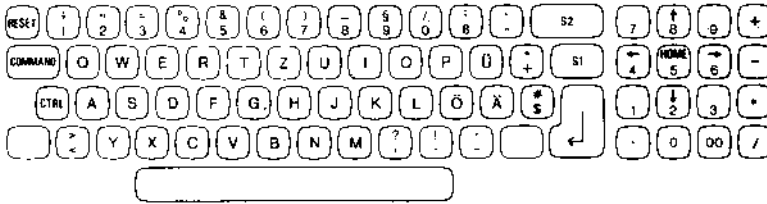
Tastensymbol	Beschreibung	allein	mit SHIFT	mit CONTROL	mit COMMAND
00	RESET	DD	DE	A0	DF
01	<	3C	3E	7F	F8
02	a	61	41	01	80
03	b	62	42	A1	81
04	c	63	43	A2	82
05	d	64	44	04	83
06	e	65	45	05	84
07	f	66	46	06	85
08	g	67	47	07	86
09	h	68	48	08	87
0A	i	69	49	09	88
0B	j	6A	4A	0A	89
0C	k	6B	4B	0B	8A
0D	l	6C	4C	0C	8B
0E	m	6D	4D	0D	8C
0F	n	6E	4E	0E	8D
10	o	6F	4F	0F	8E
11	p	70	50	10	8F
12	q	71	51	11	90
13	r	72	52	12	91
14	s	73	53	A3	92
15	t	74	54	14	93
16	u	75	55	15	94
17	v	76	56	16	95
18	w	77	57	17	96

19	x	78	58	18	97
1A	z	7A	5A	1A	99
1B	y	79	59	19	98
1C	0	30	3D	E0	EC
1D	1	31	21	E1	ED
1E	2	32	22	E2	EE
1F	3	33	23	E3	EF
20	4	34	24	E4	F0
21	5	35	25	E5	F1
22	6	36	26	E6	F2
23	7	37	2F	E7	F3
24	8	38	28	E8	F4
25	9	39	29	E9	F5
26	ß	7E	3F	EA	F6
27	.	27	60	EB	F7
28	ü	7D	5D	00	13
29	+	2B	2A	FB	1C
2A	ö	7C	5C	1E	FC
2B	ä	7B	5B	1F	FD
2C	é	23	5E	1D	9F
2D	.	2C	3B	FE	F9
2E	.	2E	3A	FF	FA
2F	-	2D	5F	A4	A5
C0	SPACE	20	20	20	20
C1	↵	A7	A7	A7	A7
C2	S1	A8	A8	A8	A8
C3	S2	A9	A9	A9	A9

numerischer Tastaturblock

Tastaten-code	Beschreibung	allein	mit S-IF1	mit CONTROL	mit COMMAND
C4	.	2E	2E	B0	2E
C5	0	30	30	B1	30
C6	00	A6	A6	B2	A6
C7	1	31	1C	B3	31
C8	2	32	9A	B4	32
C9	3	33	1D	B5	33
CA	4	34	9B	B6	34
CB	5	35	9C	1B	35
CC	6	36	9D	B8	36
CD	7	37	1E	B9	37
CE	8	38	9E	8A	38
CF	9	39	1F	BB	39
D0	+	2B	2B	BC	2B
D1	-	2D	2D	BD	2D
D2	*	2A	2A	BE	2A
D3	/	2F	2F	BF	2F

Deutsche Tastatur (alt)



alphanumerischer Tastaturblock

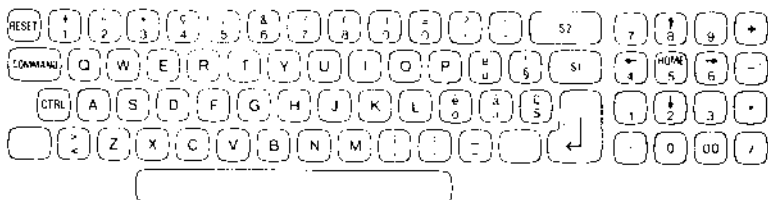
Tastencode	Beschreibung	allein	mit SHIFT	mit CONTROL	mit COMMAND
00	RESET	DD	DE	A0	DF
01	<	3C	3E	7F	F8
02	a	61	41	01	80
03	b	62	42	A1	81
04	c	63	43	A2	82
05	d	64	44	04	83
06	e	65	45	05	84
07	f	66	46	06	85
08	g	67	47	07	86
09	h	68	48	08	87
0A	i	69	49	09	88
0B	j	6A	4A	0A	89
0C	k	6B	4B	0B	8A
0D	l	6C	4C	0C	8B
0E	m	6D	4D	0D	8C
0F	n	6E	4E	0E	8D
10	o	6F	4F	0F	8E
11	p	70	50	10	8F
12	q	71	51	11	90
13	r	72	52	12	91
14	s	73	53	A3	92
15	t	74	54	14	93
16	u	75	55	15	94
17	v	76	56	16	95
18	w	77	57	17	96

19	x	78	58	18	97
1A	z	7A	5A	1A	99
1B	y	79	59	19	98
1C	0	30	3D	E0	EC
1D	1	31	21	E1	ED
1E	2	32	22	E2	EE
1F	3	33	23	E3	EF
20	4	34	24	E4	F0
21	5	35	25	E5	F1
22	6	36	26	E6	F2
23	7	37	2F	E7	F3
24	8	38	28	E8	F4
25	9	39	29	E9	F5
26	0	7E	3A	EA	F6
27	-	5E	60	EB	F7
28	ü	7D	50	00	13
29	+	28	2A	FB	1C
2A	ö	7C	5C	1E	FC
2B	ä	7B	5B	1F	FD
2C	\$	24	23	1D	9F
2D	,	2C	3F	FE	F9
2E	.	2E	21	FF	FA
2F	-	2D	27	A4	A5
C0	SPACE	20	20	20	20
C1	↵	A7	A7	A7	A7
C2	S1	A8	A8	A8	A8
C3	S2	A9	A9	A9	A9

numerischer Tastaturblock

Tastencode	Beschreibung	allein	mit SHIFT	mit CONTROL	mit COMMAND
C4	.	2E	2E	B0	2E
C5	0	30	30	B1	30
C6	00	A6	A6	B2	A6
C7	1	31	1C	B3	31
C8	2	32	9A	B4	32
C9	3	33	10	B5	33
CA	4	34	9B	B6	34
CB	5	35	9C	1B	35
CC	6	36	9D	B8	36
CD	7	37	1E	B9	37
CE	8	38	9E	BA	38
CF	9	39	1F	BB	39
D0	+	2B	2B	BC	2B
D1	-	2D	2D	BD	2D
D2	*	2A	2A	BE	2A
D3	/	2F	2F	BF	2F

Schweizer Tastatur, deutsche Version



alphanumerischer Tastaturblock

Tastencode	Beschreibung	allein	mit SHIFT	mit CONTROL	mit COMMAND
00	RESET	00	0E	A0	DF
01	<	3C	3E	7F	F8
02	a	61	41	01	80
03	b	62	42	A1	81
04	c	63	43	A2	82
05	d	64	44	04	83
06	e	65	45	05	84
07	f	66	46	06	85
08	g	67	47	07	86
09	h	68	48	08	87
0A	i	69	49	09	88
0B	j	6A	4A	0A	89
0C	k	6B	4B	0B	8A
0D	l	6C	4C	0C	8B
0E	m	6D	4D	0D	8C
0F	n	6E	4E	0E	8D
10	o	6F	4F	0F	8E
11	p	70	50	10	8F
12	q	71	51	11	90
13	r	72	52	12	91
14	s	73	53	A3	92
15	t	74	54	14	93
16	u	75	55	15	94
17	v	76	56	16	95
18	w	77	57	17	96

19	x	78	58	18	97
1A	z	7A	5A	1A	99
1B	y	79	59	19	98
1C	0	30	30	E0	EC
1D	1	31	21	E1	ED
1E	2	32	22	E2	EE
1F	3	33	23	E3	EF
20	4	34	24	E4	F0
21	5	35	25	E5	F1
22	6	36	26	E6	F2
23	7	37	2F	E7	F3
24	8	38	28	E8	F4
25	9	39	29	E9	F5
26	,	27	3F	EA	F6
27	;	5E	60	EB	F7
28	ü	7D	50	00	13
29	§	40	21	FB	1C
2A	ö	7C	7E	1E	FC
2B	ä	7B	5B	1F	FD
2C	§	24	23	1D	9F
2D	,	2C	38	FE	F9
2E	.	2E	3A	FF	FA
2F	-	2D	5F	A4	A5
C0	SPACE	20	20	20	20
C1	↵	A7	A7	A7	A7
C2	S1	A8	A8	A8	A8
C3	S2	A9	A9	A9	A9

numerischer Tastaturblock

Tastencode	Beschreibung	allein	mit SHIFT	mit CONTROL	mit COMMAND
C4	.	2E	2E	B0	2E
C5	0	30	30	B1	30
C6	00	A6	A6	B2	A6
C7	1	31	1C	B3	31
C8	2	32	9A	B4	32
C9	3	33	10	B5	33
CA	4	34	9B	B6	34
CB	5	35	9C	1B	35
CC	6	36	9D	B8	36
CD	7	37	1E	B9	37
CE	8	38	9E	BA	38
CF	9	39	1F	BB	39
D0	+	2B	2B	BC	2B
D1	-	2D	2D	BD	2D
D2	*	2A	2A	BE	2A
D3	/	2F	2F	BF	2F

USA-ASCII-Tastatur



alphanumerischer Tastaturblock

Tastaten-code	Beschriftung	allein	mit SHIFT	mit CONTROL	mit COMMAND
00	RESET	00	DE	A0	0F
01	\	5C	7C	7F	F8
02	a	61	41	01	80
03	b	62	42	A1	81
04	c	63	43	A2	82
05	d	64	44	04	83
06	e	65	45	05	84
07	f	66	46	06	85
08	g	67	47	07	86
09	h	68	48	08	87
0A	i	69	49	09	88
0B	j	6A	4A	0A	89
0C	k	6B	4B	0B	8A
0D	l	6C	4C	0C	8B
0E	m	6D	4D	0D	8C
0F	n	6E	4E	0E	8D
10	o	6F	4F	0F	8E
11	p	70	50	10	8F
12	q	71	51	11	90
13	r	72	52	12	91
14	s	73	53	A3	92
15	t	74	54	14	93
16	u	75	55	15	94
17	v	76	56	16	95
18	w	77	57	17	96

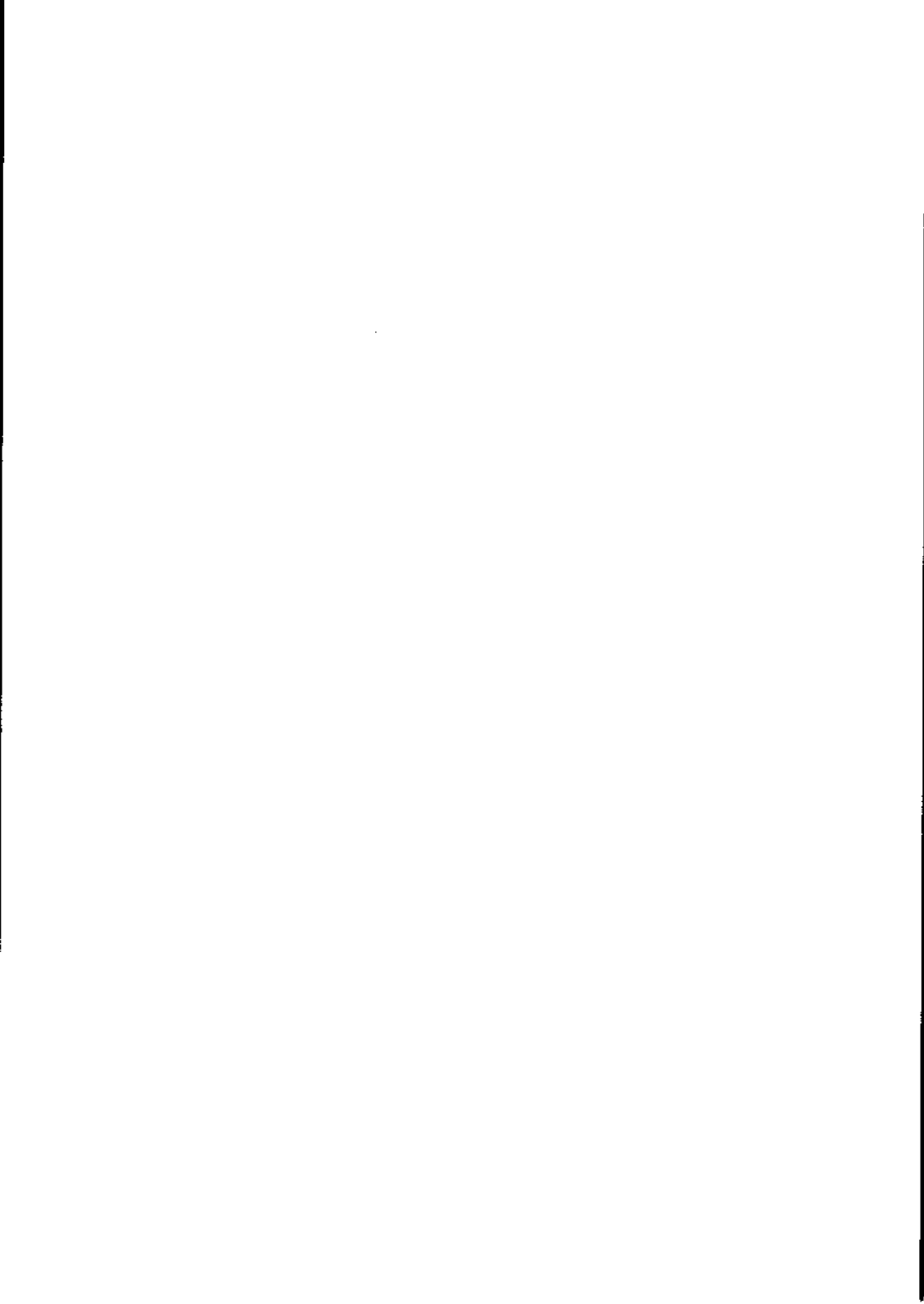
19	x	78	58	18	97
1A	y	79	59	19	98
1B	z	7A	5A	1A	99
1C	0	30	3D	E0	EC
1D	1	31	21	E1	ED
1E	2	32	22	E2	EE
1F	3	33	23	E3	EF
20	4	34	24	E4	F0
21	5	35	25	E5	F1
22	6	36	26	E6	F2
23	7	37	27	E7	F3
24	8	38	28	E8	F4
25	9	39	29	E9	F5
26	-	2D	3D	EA	F6
27	-	5E	7E	EB	F7
28	@	4C	6C	EC	13
29	[5B	7B	EB	1C
2A	;	3B	2B	1E	FC
2B	:	3A	2A	1F	FD
2C]	5D	7D	1D	9F
2D	,	2C	3C	FE	F9
2E	.	2E	3E	FF	FA
2F	/	2F	3F	A4	A5
CC	SPACE	2D	2D	2D	2D
C1	↵	A7	A7	A7	A7
C2	S1	A8	A8	AB	A8
C3	S2	A9	A9	A9	A9

numerischer Tastaturblock

Tastensymbol	Beschriftung	allein	mit SHIFT	mit CONTROL	mit COMMAND
C4	.	2E	2E	B0	2E
C5	0	30	30	B1	30
C6	00	A6	A6	B2	A6
C7	1	31	1C	B3	31
C8	2	32	9A	B4	32
C9	3	33	1D	B5	33
CA	4	34	9B	B6	34
CB	5	35	9C	1B	35
CC	6	36	9D	B8	36
CD	7	37	1E	B9	37
CE	8	38	9E	8A	38
CF	9	39	1F	8B	39
DO	+	2B	2B	BC	2B
D1	-	2D	2D	BD	2D
D2	*	2A	2A	BE	2A
D3	/	2F	2F	BF	2F

USA-ASCII-Tastatur mit BASIC-Ganzworten





3.3 Tastaturfile

Alle Zeichen, die nach Tastendruck am Bildschirm angezeigt werden können, sind in Punktmatrizen im Arbeitsspeicher festgehalten. Standardmäßig existieren solche Punktmatrizen für die ISO-Codes 32 bis 127. Jeder Taste ist ein Original-ISO-Code zugeordnet (vgl. **ck**, Kapitel 1.6 und 3.2). Je nach ISO-Code wird das Zeichen abgebildet, dessen Punktmatrix aktuell zu diesem ISO-Code gehört. Die aktuell gültigen Punktmatrizen können mit Hilfe des Full-Screen-Editors (PCOS-Befehl **ed** ohne Parameter **%c**) unter Aufruf des Tastaturfiles eingesehen und ggf. geändert werden. Nach Aufruf des Full-Screen-Editors unter Angabe des gewünschten Tastaturfiles werden im Tastaturfile folgende Informationen bereitgestellt:

- Zeile 1: beliebiger Text zur Beschreibung des Tastaturfiles

- Zeile 2: Nummer der zugrundeliegenden nationalen Tastatur (vgl. **sl**); kann beliebiger Text sein; wird bei **wf** nicht berücksichtigt

- Zeile 3: Anzahl der Elementarpunktzeilen für jedes Zeichen (standardmäßig 10); wird bei **wf** nicht berücksichtigt

- Zeile 4: Anzahl der im Tastaturfile beschriebenen Punktmatrizen (dezimal), gefolgt von mindestens einem beliebigen Zeichen. Das Minimum ist 95 (für die ISO-Codes 32 bis 127); das Maximum ist 190. D.h., daß den ISO-Codes 128

bis maximal 222 zusätzliche Punktmatrizen zugeordnet werden können. Dieser Wert wird bei **wf** berücksichtigt.

Beispiele:

WEST GERMANY	ZSATZ 3
Country 1	aus: deutsch II
matrix height = 10	matrix height = 10
95 characters	115 Zeichen

Es folgen die einzelnen Punktmatrizen (ab ISO-Code 32), denen (nur zur Information) eine laufende Nr. (ab 32) vorangestellt ist. Die Punktmatrizen werden, unabhängig von der vorangestellten Nr., in sequentieller Folge den ISO-Codes ab 32 zugeordnet.

Beispiel einer Punktmatrix:

```
50
-----
---XXX-
---X---X
-----X
-----X-
-----X--
---X---
---XXXXX
-----
-----
```

Es handelt sich also um jeweils 10 Zeilen zu 8 Spalten. Punkte, die nicht erleuchtet dargestellt werden sollen, sind mit '-' zu markieren, die anderen mit 'X' (Großbuchstabe).

Je nach Zeichen (Ober-, Unterlängen) können alle 10 Punktzeilen verwendet werden.

Für das 64 x 16-Bildschirmformat sind die ersten drei Punktspalten links normalerweise frei; sie dienen zur Abgrenzung der einzelnen Zeichen und werden als "Zwischenraum" verwendet.

Beim 80 x 25-Bildschirmformat werden die beiden ersten Punktspalten von links überhaupt nicht abgebildet. Die dritte wird normalerweise zur Abgrenzung der Zeichen verwendet.

Es empfiehlt sich, die Spalten 1 und 2 nicht zu belegen, wenn auch im 80 x 25-Format mit dem betreffenden Tastaturfile gearbeitet werden soll.

Über den Full-Screen-Editor können vorhandene Punktmatrizen eingesehen, geändert oder neue aufgenommen wrden.

Soll die Anzahl der abbildbaren Zeichen erhöht werden, ist folgendermaßen vorzugehen:

1. Anzahl beschriebener Punktmatrizen in Zeile 4 anpassen;
2. ans aktuelle Ende die weiteren Punktmatrizen anhängen (die davor angegebene Nr. beschreibt nicht automatisch den ISO-Code des Zeichens, sondern dient nur der Information. Der zugeordnete ISO-Code wird aus der laufenden Nr. der Punktmatrix im Tastaturfile +31 automatisch ermittelt.)

3. Nicht allen Original-ISO-Codes, die größer als 127 sind, sind Tasten(kombinationen) zugeordnet. Es können aber mit Hilfe des PCOS-Befehls **ck** solche Zuordnungen neu erfolgen, z.B. auch vorhandene geändert werden.

Soll ein bestimmtes, dermaßen erstelltes/geändertes Tastaturfile aktiviert werden, ist der PCOS-Befehl **wf** einzusetzen.

ACHTUNG: Vor **wf** 'Tastaturfile' sollte zuerst durch **wf** (ohne Parameter) das aktuell gültige aus dem Arbeitsspeicher entfernt werden.

Bemerkungen:

- Die über **wf** definierten Zeichen gelten nur am Bildschirm, nicht jedoch am Drucker oder z.B. bei V24-Übertragung.
- Sind Tastenbelegungen mit **pk** geändert worden, gilt die dort definierte Zeichenfolge, nicht nur am Bildschirm, sondern auch am Drucker bzw. bei Übertragungen. Über **pk** geänderte Belegungen können mit **pk** 'ISO-Code' (ohne weitere Parameter) aufgehoben werden.
- Die den Tasten zugeordneten ISO-Codes können über **ck** geändert werden.
- Bei Anwendung von **ps** wird neben der zugrundeliegenden nationalen Tastatur auch der aktuelle Bildschirmzeichensatz abgespeichert.
- Ist die in Zeile 4 angegebene Anzahl Punktmatrizen geringer als die im 'Tastaturfile' tatsächlich vorhandene, werden bei **wf** nur die durch 'Anzahl Punktmatrizen' definierten Zeichen berücksichtigt.

Fehlermeldungen und -Codes

ohne *: nur BASIC
* : auch PCOS
** : nur PCOS

Code | Meldung

1	NEXT without FOR
2	Syntax error
3	RETURN without GOSUB
4	Out of DATA
5	Illegal function call
6	Overflow
7*	Out of memory
8	Undefined line number
9	Subscript out of range
10	Duplicate Definition
11	Division by zero
12	Illegal direct
13*	Type mismatch
14	Out of string space
15	String too long
16	String formula too complex
17	Can't continue
18	Undefined user function
19	No RESUME
20	RESUME without error
22	Missing operand
23	Line buffer overflow
26	FOR without NEXT
29	WHILE without WEND
30	WEND without WHILE
31	IEEE: Invalid talker/listener address
32	IEEE: Talker = listener address
33	IEEE: Unprintable error
34	IEEE: Board not present
35*	Window not open
36*	Unable to create window
37	Invalid action-verb
38	Parameter out of range
39	Too many dimensions
40	x)

Code | Meldung

50		FIELD overflow
51		Internal error
52		Bad file number
53*		File not found
54*		Bad file mode
55*		File already open
57*		Disk I/O error
58*		File already exists
59**		Disk type mismatch
60**		Disk not initialized
61*		Disk full
62*		Input past end
63*		Bad record number
64*		Bad file name
66		Direct statement in file
67		Too many files
68*		Internal error
69*		Volume name not found
70*		Rename error
71*		Volume number error
72*		Volume not enabled
73*		Invalid Password
74*		Illegal disk change
75*		Write Protected
76*		Error in Parameter
77*		Invalid number of parameters
78*		File not OPEN
79*		Printer error
80**		Copy protected file
81*		Paper Empty error
82*		Printer Fault error
92**		Command not found
93**		Control C (from console)
99**		Bad load file
101**		Error in time or date
108**		Call user error
110**		Time out
111**		Invalid device

- x) Es wird nur "in 'Zeilenr.'" gemeldet. Ein PCOS-Fehler ist aufgetreten. Der Fehler 40 kann nicht mit der **ERROR**-Anweisung auf **ERR** zugewiesen werden. Nicht bei allen PCOS-Fehlern erhält **ERR** den Wert 40

Der Code des Fehlers kann in BASIC mit **PRINT ERR** abgefragt werden.

Die Bedeutung eines Fehlers kann in PCOS mit **ep** abgefragt werden.

Nicht aufgeführte Fehler-Codes bewirken in BASIC die Meldung "Unprintable error". Sie können in Anwenderprogrammen über eine Fehlerbehandlungsroutine beliebig besetzt werden. Der höchste zulässige Fehlercode ist 255.

Beinhaltet **ERR** den Wert \emptyset , ist entweder seit dem Laden des BASIC-Interpreters noch kein Fehler aufgetreten oder zuletzt eine Fehlerbehandlungsroutine mit **RESUME** verlassen worden. \emptyset kann auf **ERR** nicht zugewiesen werden.

Beschreibung der BASIC- und PCOS-Fehlermeldungen

(Nicht aufgeführte Fehler-Codes bewirken in BASIC die Meldung "Unprintable error".)

ERROR BEDEUTUNG

- 1 Es wurde ein **NEXT** erreicht, dem kein **FOR** zugeordnet ist (fehlendes Stack).
- 2 Es wurde ein Syntaxfehler erkannt. Das System verzweigt in den Edit-Mode und zeigt sofort die entsprechende Zeile (Zeilennummer) an. Der Syntaxfehler kann mit den Möglichkeiten des Edit-Modes behoben werden. Sollen vor der Behebung noch Variablenwerte geprüft werden, so muß der Edit-Mode mit Q verlassen werden, da andernfalls die Variableninhalte gelöscht werden. Wurde das Programm mit **SAVE** und Parameter **P** gespeichert, erfolgt zusätzlich die Meldung "Illegal function call".
- 3 Es wurde ein **RETURN** erreicht, ohne daß der Rücksprung aus dem Unterprogramm möglich ist (fehlendes Stack).
- 4 Es wird versucht, mehr Daten aus **DATA**-Anweisungen zu lesen als verfügbar sind.
- 5 Unzulässiger Aufruf einer Funktion. Dies kann durch falsche Angabe der Parameter oder durch Ausführung von Anweisungen oder Befehlen, die für ein geschütztes Programm unzulässig sind, verursacht sein, z.B.
 - a) negativer oder zu großer Array-Index
 - b) **LOG**-Argument ≤ 0
 - c) **SQR**-Argument < 0
 - d) Nicht-Integer-Exponent bei negativer Basis
 - e) Aufruf von **USR** ohne Anfangsadresse
 - f) falsches Argument in:
MID\$, LEFT\$, RIGHT\$, TAB, SPC,
STRING\$, SPACE\$, INSTR, ASC, CHR\$,
ON...GOTO, oder ON...GOSUB

- 6 Erzeugung einer zu großen Zahl. Es wird die größtmögliche Zahl gesetzt und mit dieser Zahl unmittelbar fortgesetzt.
- 7 Überlauf im Speicher. Das kann durch zu große Datenmengen, zu viele offene Rücksprungadressen oder ein zu großes Programm verursacht werden. Es kann auch versucht worden sein, ein für die aktuelle Speichergröße zu großes Assembler-Programm bzw. einen zu umfangreichen PCOS-Befehl zu laden.
- 8 Es wird versucht, eine nicht vorhandene Programmzeile anzusprechen.
- 9 Es wird versucht, ein Arrayelement mit einem unzulässigen Index anzusprechen. Entweder liegt der Index über 10 und ist nicht über eine **DIM**-Anweisung als zulässiger Index festgelegt oder der Array ist mit **ERASE** aufgehoben worden, oder es liegt die Anweisung **OPTION BASE 1** zugrunde, obwohl ein Index 0 angesprochen wird. Der Fehler kann auch auftreten, ohne daß einer dieser Fälle vorliegt. Nach Einfügen einer Anweisungszeile (z.B. **REM xxx**) direkt vor der den Fehler verursachenden Zeile wird der Fehler nicht mehr auftreten.
- 10 Ein bereits dimensionierter Array soll zum zweiten Mal dimensioniert werden. Entweder soll eine zweite **DIM**-Anweisung für einen Array ausgeführt werden oder die **DIM**-Anweisung steht erst nach der ersten Verwendung des Arrays. Der Fehler kann auch auftreten, ohne daß einer dieser Fälle vorliegt. Nach Einfügen einer Anweisungszeile (z.B. **REM xxx**) direkt vor der den Fehler verursachenden Zeile wird der Fehler nicht mehr auftreten.
- 11 Es soll durch 0 dividiert werden. Es wird die größte darstellbare Zahl angenommen (abhängig vom Typ der Operanden) und unmittelbar damit weitergerechnet.
- 12 Es wird versucht, eine Anweisung im Direkt-Mode auszuführen, die nur in

- Programmen verwendet werden kann (z.B. **DEF**).
- 13 Es wird versucht, einem String einen numerischen Wert oder einer numerischen Variablen einen String zuzuweisen oder unterschiedliche Typen zu vergleichen oder falsche Typen als Parameter einer Anweisung bzw. eines PCOS-Befehls oder einer Assembler-Routine zu verwenden.
- 14 Die Summe der Längen der verwendeten Strings übersteigt den dafür benötigten Speicherbereich. Mit der Funktion **FRE** (Stringausdruck) kann der Inhalt des Arbeitsspeichers neu geordnet werden.
- 15 Durch Ausführen der Operation würde der String mehr als 255 Zeichen lang werden.
- 16 Bei Ausführung der Stringoperation wird eine zu tiefe Schachtelung des Stacks erreicht. Die Operation sollte in einzelne Teiloperationen zerlegt werden.
- 17 Es wurde **CONT** eingegeben, es ist jedoch die Information über das Programm verloren (z.B., wenn nach einem Syntaxfehler durch den Edit-Mode das Stack gelöscht wurde oder eine neue Programmzeile aufgenommen wurde) oder keine Programmzeile mehr zur Ausführung vorhanden.
- 18 Aufruf einer selbstdefinierten Funktion, die nicht zuvor mit **DEF FN** vereinbart wurde.
- 19 Es wird aufgrund eines Fehlers in eine Fehlerbehandlungsroutine gesprungen, die kein **RESUME** enthält.
- 20 Es soll eine **RESUME**-Anweisung ausgeführt werden, ohne daß zuvor aufgrund eines Fehlers in die Fehlerbehandlungsroutine gesprungen wurde.
- 22 Es fehlt ein Operand nach einem Operator.
- 23 Es wurde versucht, eine Anweisungszeile mit mehr als 255 Zeichen einzugeben oder ein Random-File mit **LOAD** in den Arbeitsspeicher zu laden.
- 26 Es soll ein **FOR** ausgeführt werden, ohne daß ein zuzuordnendes **NEXT** gefunden

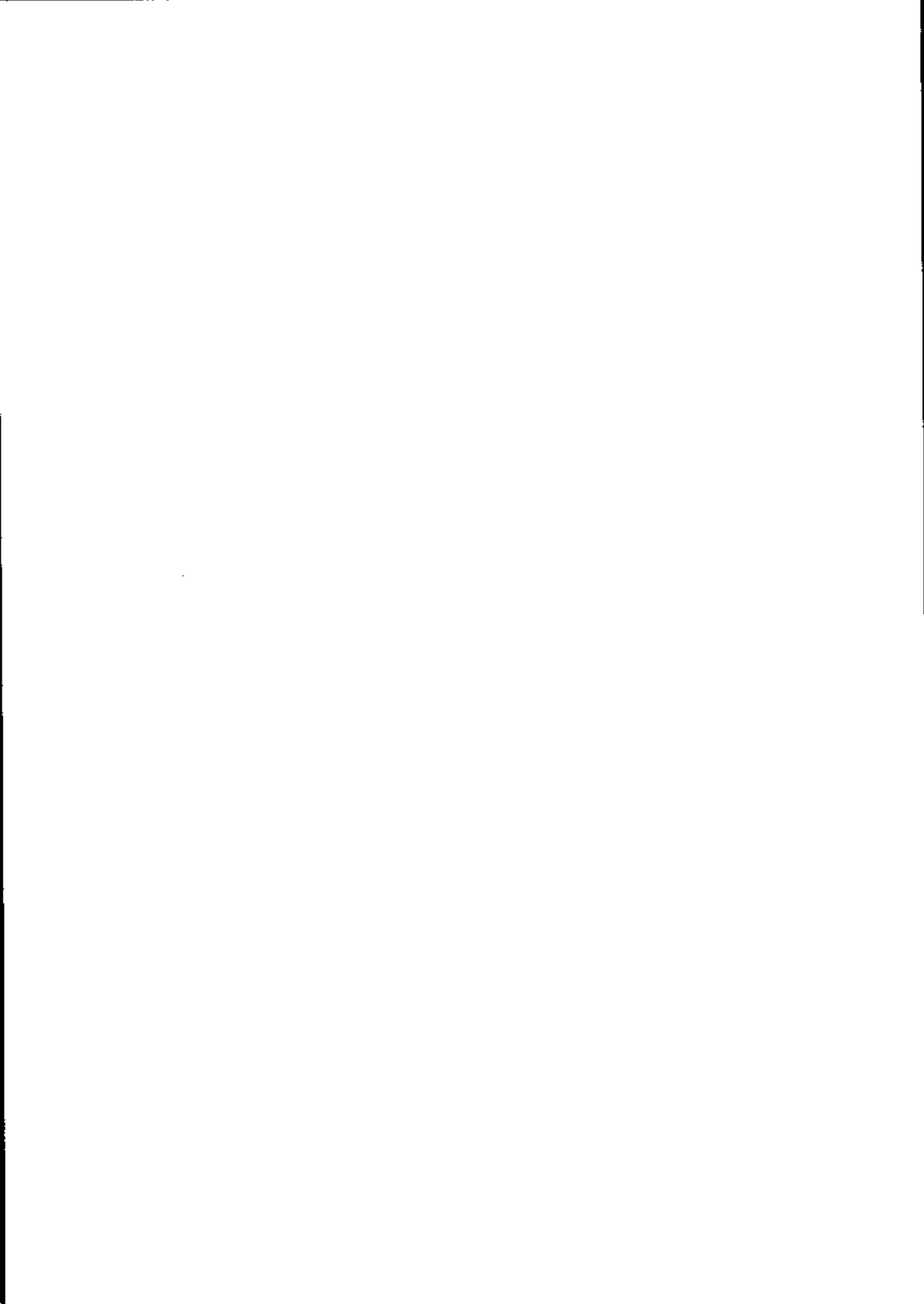
- wird. Die **NEXT**-Anweisung wird unmittelbar bei der Ausführung von **FOR** gesucht.
- 29 Es soll ein **WHILE** ausgeführt werden, ohne daß ein zuzuordnendes **WEND** vorhanden ist. Die **WEND**-Anweisung wird unmittelbar bei der Ausführung von **WHILE** gesucht.
- 30 Es soll ein **WEND** ausgeführt werden, ohne daß zuvor ein **WHILE** ausgeführt wurde.
- 31 Verwendung einer unzulässigen Sender/Empfänger-Adresse.
- 32 Es wurde versucht, eine Sender-Adresse mit einem Sende-Befehl bzw. eine Empfänger-Adresse mit einem Empfänger-Befehl anzusprechen.
- 33 Es wurde ein Fehler erzeugt, dem keine Fehlermeldung entspricht und ohne daß eine Fehlerbehandlungsroutine auszuführen ist (IEEE 488).
- 34 Es wird versucht, eine IEEE-Schnittstelle anzusprechen, obwohl keine aktiviert bzw. vorhanden ist.
- 35 Es wird ein Window angesprochen, das nicht eröffnet wurde.
- 36 Das Eröffnen eines Windows ist nicht möglich weil:
- entweder in **sb** eine geringere Anzahl vereinbart wurde oder
 - dieses Window bereits existiert oder
 - die Parameter fehlerhaft sind (z.B. die Teilung außerhalb des aktiven Windows erfolgen müßte)
- 37 In einer **LINE**-, **CIRCLE**- oder **PUT%**-Anweisung ist für 'Operand' ein unzulässiger Operand angegeben.
- 38 Ein Parameter überschreitet die zulässigen Grenzen. Diese Meldung tritt auch auf, wenn der in **GET%** angegebene Array nicht genügend Platz für die Speicherung des Bildes bietet.
- 39 Es wird ein Array mit mehr Dimensionen angegeben, als für die Anweisung zulässig ist (z.B. bei **GET%**).
- 40 Während der Ausführung von **EXEC** oder **CALL** ist ein **PCOS**-Fehler entstanden.

- 50 Für ein Random-File wird versucht, den Random-File-(FIELD)-Puffer mit mehr Zeichen zu definieren, als beim Öffnen des Files mit **OPEN** unter 'Recordlänge' vereinbart wurde.
- 51 Nicht-spezifizierte Meldung (interner Fehler). Bitte berichten Sie die Fehlerkondition an die Olivetti-Zentrale (Zentrale Software) in Frankfurt.
- 52 Es wird versucht, ein File anzusprechen, das nicht geöffnet ist, oder der Wert für 'Filenummer' liegt außerhalb des gültigen Bereiches (größer als 15 oder größer als in **sb** gesetzt).
- 53 Es wird kein File mit dem verlangten Namen gefunden (falsche Schreibweise des Filenamens oder falsche Diskette angesprochen, z.B. weil die Diskettenspezifikation nicht angegeben ist und zuletzt eine andere Diskette angesprochen wurde).
- 54 Es wird versucht, eine Anweisung für ein File eines anderen Types auszuführen (z.B. **GET** für sequentielles File) oder es wurde vorher in **OPEN** eine falsche 'Zugriffsart' angeführt.
- 55 Es wird versucht, eine bereits vergebene 'Filenummer' erneut zu vergeben oder ein offenes File mit **KILL** zu löschen. Der Fehler kann auch auftreten, weil ein anderes File offen ist. Er ist dann durch Eingabe von **CLOSE** im Direkt-Mode zu beheben.
- 57 Beim Zugriff auf Diskette tritt ein Übertragungsfehler auf. Dieser Fehler weist auf beschädigte Disketten oder eine schlecht justierte Diskettenstation hin. Es ist Datenverlust anzunehmen; (Recoverylauf nötig)! Eventuell liegt nur keine Diskette in der angesprochenen Station ein, oder die Diskette ist mit Schreibschutz versehen.
- 58 Es wird versucht, einem File einen Namen zu geben, der auf der Diskette bereits existiert.

- 59 Es wurde für die auszuführende Operation ein Diskettentyp unterstellt, der für das vorhandene Laufwerk nicht möglich ist.
- 60 Die angesprochene Diskette ist noch nicht mit **vf** formatiert.
- 61 Die Diskette ist voll. Es sind weniger Sektoren verfügbar als aufgrund des gültigen Parameters 'Anzahl Sektoren' in **ss** angelegt werden sollen. Es ist anzunehmen, daß nicht alle Daten abgespeichert wurden; (Recoverylauf nötig)!
- 62 Es wird versucht, aus einem sequentiellen File mehr Daten zu lesen als im File verfügbar sind. Die **EOF**-Funktion kann zur Prüfung eingesetzt werden.
- 63 Die Angabe der 'Record-Nummer' für ein Random-File liegt außerhalb des gültigen Bereiches (von 1 bis 32767).
- 64 Fehlerhafter Aufbau des Filenamens, z.B. zu lang oder mit unzulässigen Zeichen (z.B. Blank).
- 66 Die Meldung tritt auf, wenn beim Laden Programmzeilen ohne Zeilennummer auftraten (z.B. nach Laden eines PCOS-Befehls, einer Assembler-Routine oder eines sequentiellen oder Random-Files in den Arbeitsspeicher mit **LOAD**). Das Laden wird an der fehlerhaften Stelle abgebrochen.
- 67 Es wird versucht, ein weiteres File anzulegen, obwohl im Inhaltsverzeichnis der Diskette kein Platz mehr ist (max. 196 File-Namen bei 320-KB-Diskette; bei HD mehr möglich).
- 68 Nicht-spezifizierte Meldung (interner Fehler). Bitte berichten Sie die Fehlerkondition an die Olivetti-Zentrale (Zentrale Software) in Frankfurt.
- 69 Bei einer Diskettenspezifikation wurde eine Diskettenbezeichnung angegeben, unter der keine Diskette einliegt.
- 70 Unzulässige neue Diskettenbezeichnung beim Umbenennen einer Diskette.
- 71 Bei einer Diskettenspezifikation wurde eine nicht zulässige Stationsbezeichnung

- oder die Bezeichnung einer Station,
die in der aktuellen Konfiguration nicht
vorhanden ist, angegeben.
- 72 Bei einer Diskettenspezifikation wurde
das Disketten-Passwort nicht oder unrichtig
angegeben.
- 73 Bei der Angabe einer Filespezifikation
wurde das File-Passwort nicht oder un-
richtig angegeben.
- 74 Während auf einer Diskette ein File
offen ist, wurde die betreffende Diskette
ausgetauscht. Nachdem die richtige Diskette
wieder eingelegt ist, kann weitergearbeitet
werden.
- 75 Das File, das zum Schreiben angesprochen
wurde, ist durch PCOS-Befehl fw mit
Schreibschutz versehen worden. Der Fehler
kann auch auftreten, wenn einer System-
diskette ein Disketten-Passwort zugeordnet
wurde. Sind Disketten niemals mit vf
formatiert worden, wird beim Kopierversuch
ebenfalls Error 75 gemeldet. Diese Disketten
können nicht mehr kopiert werden.
- 76 Ein Parameter wurde nicht korrekt angegeben.
- 77 Es wurden zu viele Parameter angegeben.
- 78 Für die File-Nummer, unter der eine
FIELD- bzw. eine Lese- oder Schreibopera-
tion durchgeführt werden soll, ist keine
OPEN-Anweisung mehr gültig bzw. das
unter PCOS angesprochene Prozedurfile
ist nicht mehr offen (+sFilename bzw.
+dFilename muß erfolgen).
- 79 Der Drucker steht auf LOCAL oder ist
nicht aktiv.
- 80 Es wurde versucht, ein File zu kopieren,
das von Olivetti mit Kopierschutz versehen
worden ist.
- 81 Der Drucker hat kein Papier mehr.
- 82 Am Drucker ist eine Hardware-Fehlerkondition
aufgetreten.
- 92 Der angegebene PCOS-Befehl wurde nicht
gefunden (z.B. weil er auf der einliegenden
Diskette nicht vorhanden ist bzw. ein

- falsches Schlüsselwort angegeben wurde).
- 99 Der aufgerufene PCOS-Befehl bzw. die aufgerufene Assembler-Routine ist nicht mit dem Release des geladenen Betriebssystems kompatibel oder das aufgerufene File vom falschen Typ.
- 101 Bei `ss` wurde ein falscher Wert für 'Datum' oder 'Uhrzeit' angegeben.
- 108 Beim Abarbeiten eines PCOS-Befehls oder einer Assembler-Routine wurde eine Fehlerkondition erzeugt.
- 110 Die angesprochene Einheit ist nicht aktiv (z.B. der Drucker nicht angeschlossen).
- 111 Die angegebene Einheit ist nicht vorhanden (z.B. falscher Name, unzulässige Zeichen oder : vergessen oder die einliegende Disk(ette) ist noch nicht mit `vf` formatiert).
- vor Parametern eines PCOS-Befehls kann dies ebenfalls verursachen, da damit ein angewähltes neues Ein/Ausgabe-Medium abgestellt wird.



5. Fehlermeldungen während der Autodiagnose

- Dreieck: Fehler an der CPU
- Rechteck: ROM-Fehler; an den parallel angeschlossenen Drucker werden folgende Informationen gesendet:
- E (XN),
wobei
X: H oder L
höchstes (H) bzw. niedrigstes (L) Byte
N: ROM-Nummer (Ø bis 4)
- Raute: RAM-Fehler; an den parallel angeschlossenen Drucker werden folgende Informationen gesendet:
- EM (C) (BB) (SSSS) (WWW),
wobei
C: RAM-Konfigurations-Nr. (oktal)
BB: Nummer der fehlerhaften Bank (hexadezimal)
SSSS: erwartete Werte
WWW: aktuelle Werte

vier senkrechte

Striche:

unzulässige Unterbrechung

E C0	Fehler im parallelen 8255-Port-IC
E C1	Fehler im 6845-Video-Controller-IC
E C2	Fehler im 1797-Disk-Drive-Controller-IC
E C3	Fehler im 8253-Echtzeit-IC
E C4	Fehler im 8251-USART-IC
E C5	Fehler im 8251-RS232C-USART-IC
E C6	Fehler im 8259-Interrupt-Controller-IC
E K0	Keine Antwort von der Tastatur
E K1	Selbstüberprüfungsfehler bei der Tastatur
E I0	Non-vectorred Interrupt
E I1	Vectorred Interrupt
E D0	Fehler in Disketten-Drive 0
E D1	Fehler in Disketten-Drive 1
E D10	Fehler in Hard-Disk-Drive 10

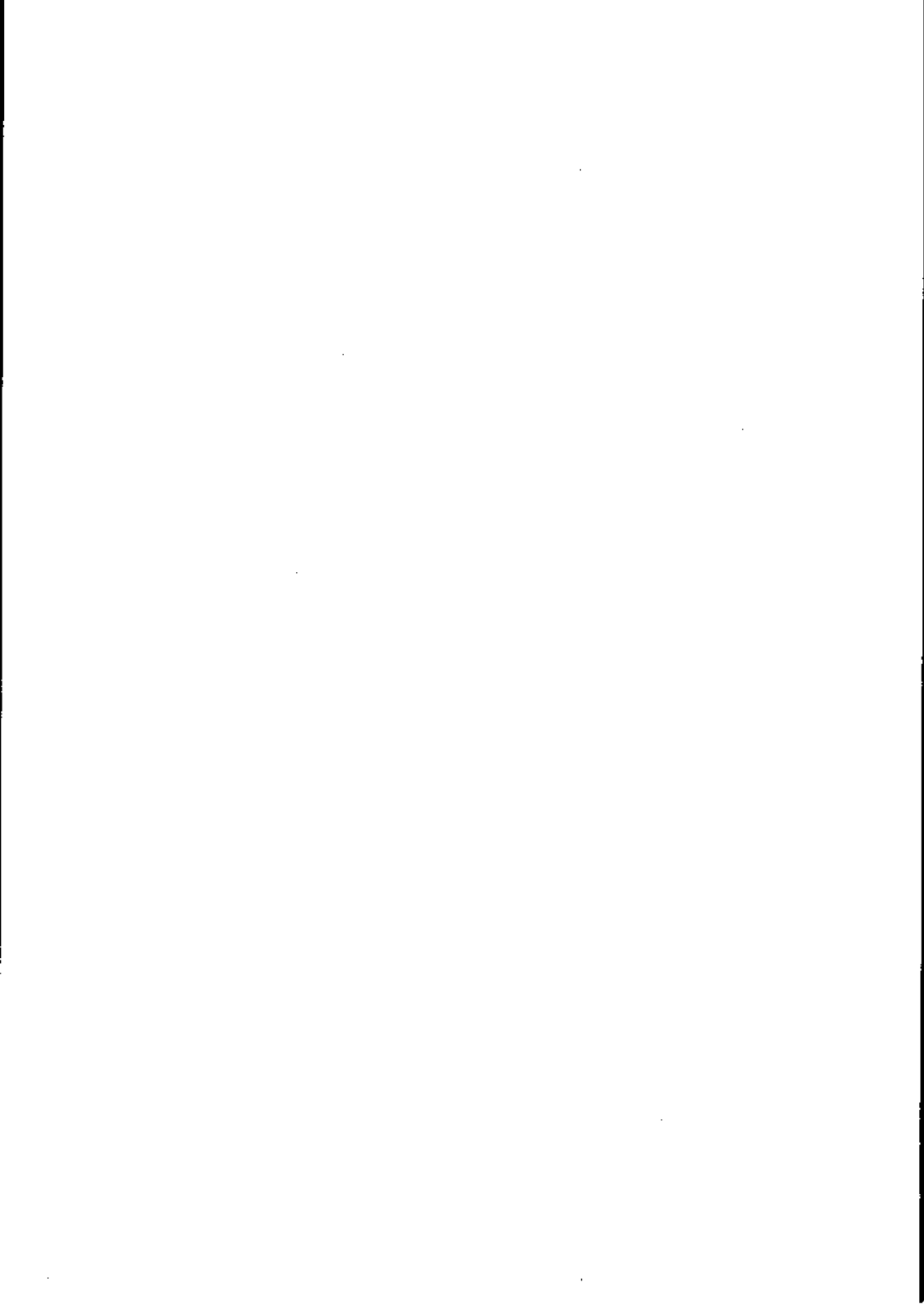
Fehlermeldungen vom Bootstrap:

Insert system disk
and type any key Keiner der Disk(etten)-Driver ist be-
reit; Systemdiskette einlegen und belie-
bige Taste drücken

Invalid File Error
(xx) on drive (n) xx kann folgende Werte annehmen:
ØØ: unzulässiger Extent-Zähler beim FD
Ø1: unzulässiger File-Typ
Ø2: unzulässiger Sektor-Zähler
Ø3: EOF-Fehler
Ø4: Parameter außerhalb zulässiger
Grenze

Disk Error: (xx) xx ist zweistellig hexadezimal; die
einzelnen Bits zeigen folgende Fehler
an:

Bit-Wertigkeit	Fehler
Ø	unzulässiger Parameter
1	nach restore keine Spur Ø
2	Suchfehler
3	Datenübertragungsfehler
4	Record nicht gefunden
5	Schreibfehler
6	Schlechter Block (bei Hard-Disk); Schreibschutz (bei Diskette)
7	Drive nicht bereit



6. Tabelle zur Umrechnung dezimal/hexadezimal/dual

$$0_{(10)} - 127_{(10)}$$

erst Spalte, dann Zeile

hexadezimal		0	1	2	3	4	5	6	7
	dual	0000	0001	0010	0011	0100	0101	0110	0111
0	0000	0	16	32	48	64	80	96	112
1	0001	1	17	33	49	65	81	97	113
2	0010	2	18	34	50	66	82	98	114
3	0011	3	19	35	51	67	83	99	115
4	0100	4	20	36	52	68	84	100	116
5	0101	5	21	37	53	69	85	101	117
6	0110	6	22	38	54	70	86	102	118
7	0111	7	23	39	55	71	87	103	119
8	1000	8	24	40	56	72	88	104	120
9	1001	9	25	41	57	73	89	105	121
A	1010	10	26	42	58	74	90	106	122
B	1011	11	27	43	59	75	91	107	123
C	1100	12	28	44	60	76	92	108	124
D	1101	13	29	45	61	77	93	109	125
E	1110	14	30	46	62	78	94	110	126
F	1111	15	31	47	63	79	95	111	127

Beispiel:

hexadezimal: 3D = dual: 00111101 = dezimal: 61

$$128_{(10)} - 255_{(10)}$$

erst Spalte, dann Zeile

hexadezimal		8	9	A	B	C	D	E	F
	dual	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	128	144	160	176	192	208	224	240
1	0001	129	145	161	177	193	209	225	241
2	0010	130	146	162	178	194	210	226	242
3	0011	131	147	163	179	195	211	227	243
4	0100	132	148	164	180	196	212	228	244
5	0101	133	149	165	181	197	213	229	245
6	0110	134	150	166	182	198	214	230	246
7	0111	135	151	167	183	199	215	231	247
8	1000	136	152	168	184	200	216	232	248
9	1001	137	153	169	185	201	217	233	249
A	1010	138	154	170	186	202	218	234	250
B	1011	139	155	171	187	203	219	235	251
C	1100	140	156	172	188	204	220	236	252
D	1101	141	157	173	189	205	221	237	253
E	1110	142	158	174	190	206	222	238	254
F	1111	143	159	175	191	207	223	239	255

Beispiel:

hexadezimal: BE = dual: 10111110 = dezimal: 190

