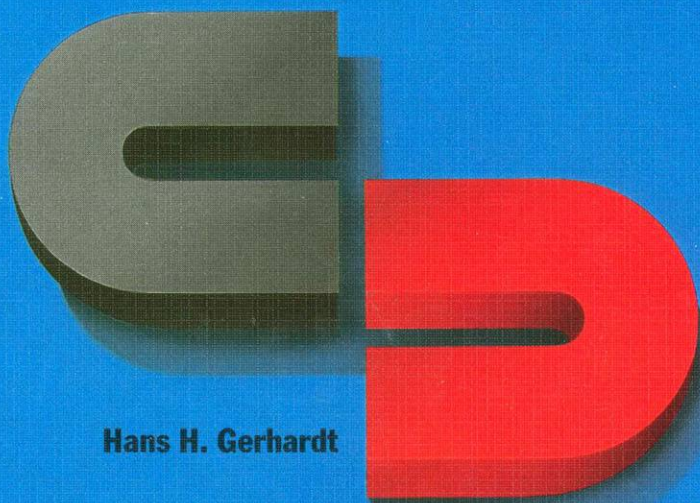




Hans H. Gerhardt

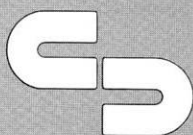
Schneider PC

MS-DOS 3.2

**Eine leichtverständliche Einführung
mit Beispielen, Übungen, alphabetischer
Befehlsübersicht und Vorschlägen
zur Konfiguration.**

Schneider PC MS-DOS 3.2

Hans H. Gerhardt



Schneider PC MS-DOS 3.2

Eine
leichtverständliche
Einführung
mit
Beispielen,
Übungen,
alphabetischer
Befehlsübersicht
und
Vorschlägen
zur
Konfiguration

Markt & Technik
Verlag AG

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Gerhardt, Hans H.:

Schneider PC MS-DOS 3.2 : e. leichtverständl. Einf. mit Beispielen, Übungen,
alph. Befehlsübersicht u. Vorschlägen zur Konfiguration /

Hans H. Gerhardt, – Haar bei München :

Markt-und-Technik-Verlag, 1987.

ISBN 3-89090-427-0

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.

Trotzdem können Fehler nicht vollständig ausgeschlossen werden.

Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische
Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.

Die gewerbliche Nutzung der in diesem Buch gezeigten Modelle und Arbeiten ist nicht zulässig.

MS-DOS ist ein eingetragenes Warenzeichen der Microsoft Corporation, USA

PC-DOS ist ein eingetragenes Warenzeichen der International Business Machines Corporation, USA

15 14 13 12 11 10 9 8 7 6 5 4 3

90 89 88 87

ISBN 3-89090-427-0

© 1987 by Markt & Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, D-8013 Haar bei München/West-Germany

Alle Rechte vorbehalten

Einbandgestaltung: Grafikdesign Heinz Rauner

Druck: Schoder, Gersthofen

Printed in Germany

Inhaltsverzeichnis

Vorwort	9
1 Einführung	11
1.1 Die Hardware	11
1.2 Die Software	12
1.3 Anwendungsmöglichkeiten	13
2 Was ist MS-DOS?	15
3 Die Kompatibilität von PCs und MS-DOS	17
4 Die Tastatur Ihres PCs	19
4.1 Allgemeines	19
4.2 Die Funktionstastenbelegung (Übersicht)	20
4.3 Beispiele für die Anwendung der Funktionstasten	21
5 Eine Übersicht der MS-DOS-Befehle	25
5.1 Allgemeines	25
5.2 Interne Befehle	26
5.3 Externe Befehle	28
6 Diskette und Festplatte	31
6.1 Aktuelles Laufwerk	31
6.2 Einteilung von Disketten und Festplatte	32
6.3 Disketten	33
6.4 Festplatten	35

7	Allgemeine Einführung in das Arbeiten mit MS-DOS	37
7.1	Der neue Schneider PC kommt	37
7.2	Start mit MS-DOS	38
7.3	Installation von MS-DOS	40
7.4	Die wichtigsten Befehle im Überblick	45
7.5	Drucken von Texten und vom Bildschirm	56
7.6	Erzeugen von Sicherungskopien (von Diskette und Festplatte)	57
8	Das MS-DOS-Inhaltsverzeichnis	61
8.1	Befehle zur Bearbeitung von Directories	64
8.2	Anlegen eines Sub-Directory (MKDIR oder MD)	66
8.3	Anzeigen und Ändern des aktuellen Directory (CHDIR oder CD)	67
8.4	Löschen eines Sub-Directory (RMDIR oder RD)	68
8.5	Anzeigen des Inhaltsverzeichnisses eines Directory (DIR)	69
8.6	Löschen von Dateien in einem Directory (DEL oder ERASE)	70
8.7	Kopieren von Dateien zwischen Directories (COPY und XCOPY)	71
8.8	Zusammenfassendes Beispiel	74
9	Datenumleitung und Filter	77
9.1	Datenumleitung	77
9.2	Befehlsverkettung und Filter	79
10	Gerätenamen	83
11	Die MS-DOS-Befehle mit Übungen	85
	APPEND	87
	ASSIGN	89
	ATTRIB	90
	BACKUP	92
	BREAK	95
	CHDIR oder CD	96
	CHKDSK	97
	CLS	99
	COMMAND	100
	COMP	102

COPY	107
CTTY	110
DATE	111
DEBUG	112
DEL oder ERASE	119
DIR	120
DISKCOMP	123
DISKCOPY	125
ECHO	127
EDLIN	129
EXE2BIN	137
EXIT	138
FDISK	139
FIND	142
FOR	145
FORMAT	146
GOTO	149
GRAFTABL	150
GRAPHICS	151
IF	152
JOIN	154
KEYBGR	156
LABEL	157
MKDIR oder MD	158
MODE	159
MORE	162
PATH	163
PAUSE	164
PRINT	165
PROMPT	167
RECOVER	169
REM	170
RENAME oder REN	171
REPLACE	172
RESTORE	174
RMDIR oder RD	176
SET	177
SHARE	179
SHIFT	180
SORT	182
SUBST	183

	SYS	184
	TIME	185
	TREE	186
	TYPE	188
	VER	189
	VERIFY	190
	VOL	191
	XCOPY	192
12	Die BATCH-Befehle	195
13	Die Konfiguration von MS-DOS (CONFIG.SYS)	203
	Anhang	
I	Begriffserklärungen	219
II	Zeichendarstellung/Code-Tabelle	221
III	Umrechnungstabellen	227
IV	Netzwerk (MS-NET)	229
	Stichwortverzeichnis	231
	Übersicht weiterer Markt&Technik-Produkte am Ende des Buches	

Vorwort

Durch das Erscheinen des IBM-PC, Ende 1980 in den USA, gab es den großen Durchbruch auf dem Mikrocomputermarkt (Mikrocomputer gibt es seit Mitte der 70er Jahre – Verbreitung fanden diese ab etwa Ende der 70er Jahre). In den folgenden Jahren brachten viele andere Hersteller kompatible Computer auf den Markt. Es entstand eine ganze Industrie, die zusätzliche Hard- und Software herstellte. Daraus entstand ein Computerstandard, für den heute das wohl größte Angebot an Zusatzhardware und Software zur Verfügung steht.

Durch die immer größere Verbreitung der PCs und der damit verbundenen Fertigung von großen Stückzahlen sowie der Innovation sind die Kosten für die Anschaffung in den letzten Jahren ganz erheblich gesunken. Noch vor wenigen Jahren kostete ein mit dem Schneider PC vergleichbarer Computer mit zwei Diskettenlaufwerken ca. 20000 DM! Der von Ihnen erworbene PC stellt in seiner Preisklasse derzeit eines der modernsten und in der Bedienung fortschrittlichsten Geräte am Markt dar.

MS-DOS (PC-DOS) ist heute das Standard-Betriebssystem für Mikrocomputer der 16-Bit-Personalcomputer. MS-DOS ist nicht nur einfach an verschiedene Hardware anpassbar, sondern auch sehr leicht erlern- und bedienbar.

Das vorliegende Buch ist ein Ergebnis von Seminaren zum Betriebssystem MS-DOS und dadurch entsprechend aufgebaut.

Sie finden in den ersten Kapiteln (1 bis 9) eine Einführung mit den wichtigsten Grundlagen zur Bedienung Ihres Schneider PC unter MS-DOS 3.2 sowie Erläuterungen und Beispiele zu den wichtigen Befehlen.

Die Kapitel 10 bis 12 enthalten alle Befehle alphabetisch geordnet mit einer genauen Beschreibung der entsprechenden Funktion.

In den einzelnen Kapiteln werden viele Beispiele dargestellt, die Sie an Ihrem Schneider PC nachvollziehen sollten. Dadurch wird der Lerneffekt gesteigert. Für den Fortgeschrittenen bietet das Buch ab Kapitel 10 ein hilfreiches Nachschlagewerk (mit MS-DOS-Konfiguration im Kapitel 13).

Im Anhang werden zu den wichtigsten Begriffen Erläuterungen gegeben. Diese wurden bewußt an das Buchende gesetzt, damit Sie später nicht laufend darüber hinwegblättern müssen.

Ferner finden Sie im Anhang eine kurze Darstellung zum Thema Netzwerk (Verbindung mehrerer PCs).

Ziel des Buches ist es, Sie zu befähigen, Ihre Arbeiten mit MS-DOS 3.2 am Schneider PC erledigen zu können. Das Buch soll Ihnen ein ständiger Begleiter sein.

Ich bedanke mich bei allen, die mich so freundlich bei der Beschaffung von Unterlagen oder in anderer Weise unterstützt haben.

Hans H. Gerhardt

Kapitel 1 Einführung

Vorstellung des Schneider PC

Dieses Kapitel will Ihnen kurz die einzelnen Bestandteile Ihres Schneider PC und der mitgelieferten Software vorstellen. Zusätzlich erhalten Sie noch Informationen, welche Anwendungsmöglichkeiten zum Einsatz des PC bestehen.

1.1 Die Hardware

Sie erhalten beim Kauf Ihres Schneider PC eine Zentraleinheit mit einem oder zwei Diskettenlaufwerken oder einem Dis-



Bild 1.1: Die Hardware

- Hardware** kettenlaufwerk und einer Festplatte (einer fest eingebauten »Diskette« mit besonders großer Speicherkapazität und hoher Geschwindigkeit). Zusätzlich einen Monitor, die Tastatur und eine sogenannte »Maus« (mit ihr können Sie nur in Verbindung mit spezieller Software arbeiten). Damit Sie die Ergebnisse Ihrer Arbeit auch auf Papier bringen können, benötigen Sie noch zusätzlich einen Drucker. Sie können natürlich auch einen Plotter anschließen, mit dem Sie beispielsweise Grafiken zeichnen können.
- Datenfernübertragung** Für die Kommunikation mit anderen Computern per Datenfernübertragung gibt es Akustikkoppler oder Modems (von der Post).
- Software** Für die anschließbaren Geräte benötigen Sie in jedem Fall eine Software (Programme), die diese sogenannte Peripherie auch ansteuern kann. Lassen Sie sich also vor dem Kauf beraten, welches Gerät in Verbindung mit welchen Programmen geeignet ist.
- Anschlüsse am PC** Die Anschlüsse für Tastatur und Maus finden Sie an der linken Seite Ihres PC. An der Rückseite wird der Monitor angeschlossen. Daneben sind noch je eine parallele und eine serielle Schnittstelle. Der parallele Anschluß wird in der Regel für den Drucker verwendet. Die serielle Schnittstelle (V.24 RS232C) kann für den Akustikkoppler, ein Modem oder den Plotter verwendet werden. An der Tastatur läßt sich ein Joystick anschließen, der vor allem bei Spielen zum Einsatz kommt. Weitere Informationen zum Aufstellen und Zusammenbau finden Sie in den Kapiteln 1 und 2 des Handbuchs zu Ihrem Schneider PC.

1.2 Die Software

- Zusammen mit der Hardware (sichtbarer Teil des Computers) erhalten Sie vier Disketten mit Software. Ohne diese Software ist ein Computer wertlos (wie ein Auto ohne Motor).
- Betriebssystem** Zum Betrieb eines Computers benötigt man ein Betriebssystem, das die Betriebsmittel (Zentraleinheit, Diskettenlaufwerke, Monitor, Tastatur usw.) verwaltet. Ein solches Betriebssystem ist das in diesem Buch beschriebene MS-DOS 3.2.

Software des Schneider PC Mit Ihrem PC erhalten Sie folgende Software:

- MS-DOS 3.2 (Betriebssystem)
- DOS Plus Version 1.2 (Betriebssystem)
- GEM (Benutzeroberfläche für eine einfache Bedienung – in Verbindung mit der Maus)
- DOODLE (damit können Sie Bilder erstellen)
- Locomotive BASIC2 (Programmiersprache für die Erstellung eigener Programme)

Mit DOS Plus können Sie die meisten Programme verwenden, die für MS-DOS lieferbar sind. Zusätzlich kann mit DOS Plus auch noch Software verwendet werden, die für das Betriebssystem CP/M-86 lieferbar ist.

Über 90% der zur Zeit weltweit verkauften PCs arbeiten mit MS-DOS. Daher gibt es für dieses Betriebssystem auch die meiste Anwendungssoftware.

1.3 Anwendungsmöglichkeiten

Mit der zu Ihrem Schneider PC gelieferten Software können Sie schon eine ganze Menge machen. Sicher werden Sie aber auch sehr schnell feststellen, daß bei einem ernsthaften Einsatz und ohne daß Sie sich selber Programme schreiben wollen, weitere Anwendungssoftware notwendig ist. Dazu haben Sie die Auswahl unter mehreren tausend verschiedenen Programmen.

Anwendungsmöglichkeiten des Schneider PC Hier nur einige Anwendungsmöglichkeiten:

- Textverarbeitungsprogramme (Briefe, Bücher usw. schreiben)
- Finanzbuchhaltung
- Lohn- und Gehaltsabrechnung
- Programmiersprachen (Pascal, Modula, COBOL, Fortran usw.)
- Tabellenkalkulation (Multiplan, Lotus 1-2-3, SuperCalc usw.)
- Datenbanken (für Adressen, Literaturverwaltung, Plattensammlung usw.)
- Fertige Programme für die verschiedensten Branchen

Es gibt für fast jeden Bedarf die richtige Software. Bevor Sie sich ein fertiges Programm kaufen, sollten Sie sich jedoch genau überlegen, was Sie damit machen wollen. Anschließend empfehle ich Ihnen, sich genau beraten zu lassen. Sehen Sie

sich das empfohlene Programm an, und prüfen Sie, ob es Ihren Forderungen entspricht.

Viele Programme kosten mehr als Ihr Schneider PC (es beginnt mit ca. 100 DM und endet bei sehr komplexen Lösungen für mehr als 40000 DM).

Sollten Sie größere Datenmengen mit Ihrem Schneider PC verwalten wollen, so empfiehlt sich unbedingt die Anschaffung einer Festplatte.

Kapitel 2

Was ist MS-DOS?

MS-DOS als Schnittstelle zwischen Hard- und Software

Eigentlich müßte es heißen: Was ist ein Betriebssystem? MS-DOS ist eine Sammlung von Programmen, die zum Betrieb eines Computers notwendig sind.

Ein Computer (Hardware) ist für sich allein genommen ein Gerät, das Sie zwar einschalten, jedoch nicht betreiben können. Erst durch das Betriebssystem »erwacht die Anlage zum Leben«. Ein Betriebssystem ist nichts anderes als eine Programmsammlung, die den Computer und seine Peripherie (Drucker, Plotter, Diskettenlaufwerke, Festplatte, Bildschirm usw.) verwaltet. Ferner unterstützt es die Ausführung von Anwenderprogrammen (Finanzbuchhaltung, Textsystem usw.).

Ein Anwenderprogramm wird speziell für die Lösung einer Aufgabe erstellt, unabhängig vom Computersystem. Die Verbindung zur Hardware wird normalerweise durch das Betriebssystem hergestellt. Zusätzlich können Sie mit MS-DOS viele Aufgaben, wie zum Beispiel das Kopieren von Daten und Programmen, durchführen.

Laden von MS-DOS

Wenn Sie Ihren Schneider PC einschalten, wird ein Teil des MS-DOS in den Speicher (RAM) geladen. Dieser Vorgang erfolgt automatisch (System-Bootup). Ein Programm, das den Bootup durchführt, ist im Computer (ROM) enthalten und wird durch das Einschalten aktiviert. Anschließend erscheint am Bildschirm die Befehlseingabe-Bereitschaftsmeldung (der Prompt).

Aufgaben von MS-DOS

Die Aufgaben des Betriebssystems MS-DOS lassen sich in vier Bereiche aufteilen:

1. Verwaltung und Transport von Daten
2. Unterstützung und Verwaltung der/des Benutzer(s)

3. Zuteilung und Verwaltung der Betriebsmittel (Drucker usw.)

4. Steuerung von Prozessen

MS-DOS als Standard-Betriebssystem

MS-DOS ist die Abkürzung von MicroSoft Disk Operating System. Dieses Betriebssystem konnte sich in kurzer Zeit durchsetzen, vor allem durch die Entscheidung von IBM, ihre Personalcomputer damit auszustatten (die IBM-Bezeichnung ist PC-DOS).

MS-DOS kann heute als (Quasi-16-Bit-)Standard-Betriebssystem bezeichnet werden (bei 8-Bit-Computern CP/M). Das große Angebot an Software ist inzwischen schon allein ein Grund zum Einsatz von MS-DOS.

MS-DOS und XENIX

Welchem Betriebssystem die Zukunft auf den neuen 32-Bit-Computern (z. B. IBM PC/AT) gehört, ist zur Zeit fraglich. Wird das »Rennen« MS-DOS oder XENIX gewinnen? Sicherlich wird entscheidend sein, ob für XENIX künftig mehr Anwendungsprogramme verfügbar sind. Der wesentliche Vorteil von XENIX ist die Mehrplatzfähigkeit. Aber auch MS-DOS entwickelt sich weiter und besitzt in Verbindung mit Netzwerksoft- und -hardware die Mehrplatzfähigkeit.

MS-DOS besteht aus vier Teilen (und diversen Dienstprogrammen):

MS-DOS-Teil	Kurzbeschreibung
BOOT-LOADER	Wird direkt nach dem Start des Computers als erstes Programm ausgeführt. Es startet das MS-DOS, indem es IO.SYS in den Speicher des Computers lädt und das System bootet.
IO.SYS	Enthält alle systemspezifischen Programme zur Abwicklung der Kommunikation mit den angeschlossenen Geräteeinheiten (Drucker usw.).
MSDOS.SYS	Enthält den organisatorischen Teil des Betriebssystems zur Verwaltung der Dateien (Daten und Programme).
COMMAND.COM	Interpretiert alle eingegebenen Befehle und führt sie aus, wobei geprüft wird, ob es sich um einen zulässigen Befehl handelt. Ferner enthält es eigene (die internen) Befehle.

Die zu Ihrem Schneider PC 1512 mitgelieferte Diskette Nr. 1 enthält Ihr MS-DOS.

Kapitel 3

Die Kompatibilität von PCs und MS-DOS

Kompatibilität des Schneider PC

Was ist Kompatibilität? Darunter versteht man, daß Sie Software, die auf dem IBM-PC funktioniert, auch mit Ihrem kompatiblen Schneider PC benutzen können. Damit können Sie aus einer riesigen Palette von mehreren tausend Programmen diejenigen auswählen, die Ihren Bedürfnissen am meisten entgegenkommen.

Zu Ihrem hardwarekompatiblen Schneider PC können Sie Zusatzgeräte oder Einschubkarten (z.B. für den Anschluß Ihres Computers an einen Großcomputer) erwerben, die für den PC-Standard angeboten werden.

Wenn man von Kompatibilität spricht, so versteht man IBM-kompatibel. Grundsätzlich muß bei der Kompatibilität zwischen Hardware und Software unterschieden werden. Viele PCs sind zwar software-, jedoch nicht hardwarekompatibel. Wer ist schon wirklich kompatibel? Nur sehr wenige PCs erreichen einen sehr hohen Stand an Kompatibilität.

Schwierigkeiten bei nicht funktionsfähigen Programmen

Wenn Ihnen oder Ihrem Lieferanten nicht bekannt ist, ob ein Programm oder Hardware-Teil auf Ihrem Schneider PC lauffähig ist, sollten Sie dies vor dem Kauf unbedingt testen. Damit ersparen Sie sich eventuell später auftretende Schwierigkeiten z.B. mit nicht funktionsfähigen Programmen. Bedenken Sie immer, daß es für die Kompatibilität keine Garantie gibt. Soweit vom Hersteller (Schneider) selbst für Ihren PC Programme angeboten werden, können Sie jedoch auf die Lauffähigkeit vertrauen.

Kompatibilität = Vorteil des Standards

Die Kompatibilität hat den Vorteil des Standards. Es gibt eine riesige Palette von Programmen, Peripherie und Hardware-Zusätzen, die Sie verwenden können. Dies bedeutet auch, daß die Hersteller aufgrund der größeren Verkaufszahlen günstiger kalkulieren können.

Was soll Software leisten?

Beim Kauf von Programmen sollten Sie nicht nur auf Kompatibilität achten, sondern vor dem Kauf genau festlegen, was die zu erwerbende Software leisten soll. Bedenken Sie, daß leistungsfähige Programme unter Umständen ein Vielfaches Ihres PC kosten können. Häufig hängt es vor allem von den Programmen ab, ob sich Ihr PC-Kauf gelohnt hat. Programme, für die eine ältere Version als Ihr MS-DOS 3.2 verlangt wird, können Sie normalerweise ohne Probleme einsetzen.

Benutzt Ihre Software bestimmte MS-DOS-Programme, so kann es eventuell zu Schwierigkeiten kommen. Dieser Fall tritt jedoch sehr selten auf.

»Abwärts«-Kompatibilität von MS-DOS

Eine neue Version zu MS-DOS ist immer kompatibel zu den vorherigen Versionen (seit der Version MS-DOS 1).

Manchmal kommt es vor, daß eine Software auf Ihrem Schneider PC nicht läuft. Dies kann verschiedenste Ursachen haben. Eventuell reicht der Speicher Ihres Computers nicht aus, um ein Programm zu benutzen (nicht immer erhalten Sie dabei eine Fehlermeldung). In diesem Fall sollten Sie sich mit Ihrem Lieferanten oder dem Softwarehersteller in Verbindung setzen und diesem genau beschreiben, was passiert.

Kapitel 4

Die Tastatur Ihres PC

4.1 Allgemeines

Die Tastatur Ihres Schneider PC 1512 (Bild 4.1) umfaßt mehrere Gruppen:

Schreibmaschinen-Tastatur

- Kompletter Buchstabensatz entsprechend einer Schreibmaschine (mit der Betätigung der Taste Caps Lock können Sie Ihre Tastatur auf Großschreibung umschalten – dies gilt nur für Buchstaben).

Funktionstasten

- Die Funktionstasten (die Bedeutung der Tasten ist auf den nächsten Seiten näher beschrieben).

Zifferntastenfeld

- Auf der rechten Seite Ihrer Tastatur finden Sie ein Zifferntastenfeld. Um mit diesem arbeiten zu können, muß vorher die Taste Num Lock betätigt werden. Damit wird jedoch gleichzeitig die Möglichkeit der Nutzung als Funktionstasten abgeschaltet. Betätigen Sie Num Lock ein weiteres Mal, so ist der ursprüngliche Zustand wieder hergestellt.

Mit Hilfe der Funktionstasten können Sie die Befehlszeilen leicht ändern. Die zuletzt eingegebene Zeile kann aus dem Tastaturpuffer komplett oder in Teilen zurück in die Befehlszeile (Bildschirmanzeige) kopiert werden. Der Tastaturpuffer



Bild 4.1: Die Tastatur

**Eingabe von
DOS-Befehlen**

ist am Bildschirm nicht sichtbar. Dort befindet sich immer die zuletzt eingegebene Zeile bzw. der zuletzt eingetippte MS-DOS-Befehl. Damit werden folgende Eingaben unterstützt:

Eine Zeile können Sie einfach

- durch Tastendruck wieder eingeben
- korrigieren
- in geänderter Form eingeben.

4.2 Die Funktionstastenbelegung (Übersicht)

**Funktionstasten-
belegung**

Die angegebenen Funktionen gelten nur für MS-DOS! Bei der Benutzung einer Anwendungssoftware haben die Tasten meist eine andere Bedeutung (beachten Sie dazu die Handbücher zu Ihren Programmen).

Taste	Funktion	Beschreibung
F1	kopiert ein Zeichen	kopiert ein Zeichen aus dem Tastaturpuffer in die neue Zeile am Bildschirm
F2x	kopiert bis Zeichen x	kopiert alle Zeichen bis zum eingegebenen Zeichen (x) vom Tastaturpuffer in die neue Zeile am Bildschirm (ohne das Zeichen ab der aktuellen Position in der Zeile)
F3	kopiert alle Zeichen	kopiert alle Zeichen ab der aktuellen Position aus dem Tastaturpuffer in die neue Zeile am Bildschirm
F4x	überspringt Zeichen bis x	überspringt alle Zeichen des Tastaturpuffers, bis das Zeichen (x) erkannt wird (ohne daß am Bildschirm etwas angezeigt wird)
F5	neuer Tastaturpuffer	schreibt die neue Zeile in den Tastaturpuffer
F6	beendet die Eingabe	erzeugt ein Ctrl Z, welches das Zeichen für ein Dateiende darstellt (beachten Sie den COPY-Befehl)
Ctrl+Z	beendet die Eingabe	erzeugt ein Ctrl Z (1Ah), welches das Zeichen für ein Dateiende darstellt (wie Taste F6)
F7	Nullzeichen	gibt ein Ctrl @ aus, welches den Wert 00 (HEX) darstellt
Alt + Break	Eingabe löschen	löscht den Tastaturspeicher (der eingegebene Befehl wird abgebrochen)
Back Space	Eingabe löschen	löscht ein Zeichen aus der Eingabezeile am Bildschirm, und im Tastaturpuffer wird die aktuelle Position um 1 Zeichen zurückgesetzt

Taste	Funktion	Beschreibung
Ins	einfügen	schaltet die Eingabe auf Einfügemodus um (die aktuelle Position im Tastaturpuffer wird dabei nicht verändert)
Del	löschen	löscht ein folgendes Zeichen im Tastaturpuffer, jedoch nicht in der Eingabezeile bzw. am Bildschirm
Esc	Eingabe löschen	löscht die gesamte Eingabezeile, nicht jedoch den Tastaturpuffer
Ctrl+Alt+Del	Systemstart	den Computer neu starten (booten, also das MS-DOS komplett neu laden)
Ctrl+Break	Befehlsabbruch	unterbricht die Ausführung eines MS-DOS-Befehls (siehe Befehl BREAK)
Ctrl+C	Befehlsabbruch	unterbricht die Ausführung eines MS-DOS-Befehls (siehe Befehl BREAK)
Ctrl+H	Zeichen löschen	löscht das zuletzt eingegebene Zeichen am Bildschirm (wie Taste Back Space)
Tab	Tabulator	Cursor am Bildschirm zur nächsten Tabulatorposition bewegen
Ctrl+I	Tabulator	Cursor am Bildschirm zur nächsten Tabulatorposition bewegen (wie Taste Tab)
Ctrl+<RETURN>	Zeilenende	fügt ein Zeilenende bzw. Zeilenvorschub ein
Ctrl+J	Zeilenende	fügt ein Zeilenende bzw. Zeilenvorschub ein
Ctrl+NumLck	Bildschirmstopp	unterbricht die Bildschirmanzeige, bis eine beliebige andere Taste betätigt wird
Ctrl+S	Bildschirmstopp	unterbricht die Bildschirmanzeige, bis eine beliebige andere Taste betätigt wird
Ctrl+P	Druckausgabe	Bildschirmein- und -ausgaben gleichzeitig drucken (erneutes Betätigen schaltet die Funktion ab)
Ctrl+PrtSc	Druckausgabe	Bildschirmein- und -ausgaben gleichzeitig drucken (erneutes Betätigen schaltet die Funktion ab)
Shift-PrtSc	Bildschirm drucken	erstellt eine Kopie der kompletten Bildschirmanzeige am Drucker (Hardcopy-Funktion)

4.3 Beispiele für die Anwendung der Funktionstasten

Inhaltsverzeichnis einer Diskette

Sie wollen sich aus dem Inhaltsverzeichnis einer Diskette die Daten der Datei HANDBUCH.TXT anzeigen lassen. Sie tippen ein:

dir handbuch.txt <RETURN>-Taste

Befehlswiederholung

Der Befehl wird ausgeführt (Ihre Eingabe steht für Sie unsichtbar im Tastaturpuffer). Mit nur zwei Tasten ist es jetzt möglich, diesen Befehl zu wiederholen. Betätigen Sie die Tasten:

F3 <RETURN>

Sie wollen Informationen über sämtliche Dateien, die Sie unter dem Namen HANDBUCH gespeichert haben. Der Namenszusatz unterscheidet die einzelnen Dateien. In der Eingabezeile muß dafür nur .txt gegen den Stern (*) getauscht werden. Geben Sie ein:

F2-Taste und t

Es werden sämtliche Zeichen, bis zum t ausschließlich, in die Eingabezeile (am Bildschirm sichtbar) kopiert. Es erscheint:

dir handbuch.

Geben Sie dann den Stern (*) ein und betätigen Sie die Taste <RETURN>.

Text am Bildschirm anzeigen

Da es sich bei der Datei HANDBUCH.TXT um einen Text handelt, können Sie den Inhalt am Bildschirm anzeigen lassen. Verändern Sie die letzte Eingabe, um folgenden Befehl zu erhalten:

type handbuch.txt

Sie überschreiben die ersten drei Zeichen der alten Zeile (dir), indem Sie die Buchstaben typ tippen. Sie betätigen die Taste Ins, fügen ein e hinzu, betätigen die Tasten F2 * und schreiben als letztes txt. Nach Betätigung von <RETURN> wird der Inhalt auf dem Bildschirm angezeigt.

Ihre Eingaben zum letzten Beispiel noch einmal in der Übersicht:

Tasten	Funktion	Ergebnis am Bildschirm
t	schreibt ein Zeichen	t
y	schreibt ein Zeichen	ty
p	schreibt ein Zeichen	typ
Ins	schaltet um auf einfügen	

Tasten	Funktion	Ergebnis am Bildschirm
e	schreibt ein Zeichen	type
F2*	kopiert Zeichen bis zum *	type handbuch.
t	schreibt ein Zeichen	type handbuch.t
x	schreibt ein Zeichen	type handbuch.tx
t	schreibt ein Zeichen	type handbuch.txt
<RETURN>	gibt die Zeile an das MS-DOS-Programm COMMAND.COM zum Ausführen, gleichzeitig wird die neue Zeile in den Tastatureingabespeicher übernommen	Inhalt wird angezeigt

Tastenkombination

Bei Verwendung einer deutschen Tastatur benötigen Sie bei der Arbeit mit MS-DOS Zeichen, die Sie nur durch Betätigung mehrerer Tasten gleichzeitig erzeugen können:

Tasten-Kombination	Zeichen	Bemerkung
Ctrl + Alt + ü	[	Eckige Klammer auf
Ctrl + Alt + +	]	Eckige Klammer zu
Ctrl + Alt + <	\	Back-Slash
Ctrl + Alt + 2	@	Klammeraffe
Ctrl + Alt + F1		Umschalten auf (US-) ASCII-Tastatur
Alt + 124 (NumTast)	;	Unterbrochener Strich
Ctrl + Alt + F2		Umschalten auf deutsche Tastatur

Kapitel 5

Eine Übersicht der MS-DOS-Befehle

5.1 Allgemeines

Befehle sind Anweisungen an das Betriebssystem, bestimmte Aufgaben durchzuführen (Kopieren, Drucken usw.). Sie werden von Ihnen über die Tastatur eingegeben (und gleichzeitig am Bildschirm angezeigt). Sie können diese auch in einer speziellen Datei speichern und zu einem späteren Zeitpunkt mit einem einzigen Befehl abrufen (siehe Kapitel 12).

Es werden zwei Arten von Befehlen unterschieden:

- | | |
|-------------------------------|--|
| Interne Befehle | Diese sind jederzeit verfügbar. Sie sind in dem Programm COMMAND.COM enthalten, das beim Start des Schneider PC in den Speicher (RAM) geladen wird. |
| Externe Befehle | Dies sind kleine Programme (sogenannte Dienstprogramme), die Sie auf der Diskette oder Festplatte unter einem eigenen Namen finden. Wenn Sie einen externen Befehl eingeben, wird das entsprechende Dienstprogramm in den Speicher Ihres PC geladen und ausgeführt. |
| Ausführung der Befehle | Wenn Sie einen Befehl eingeben, so wird dieser von COMMAND.COM (wird beim Start des PC in den Speicher geladen) ausgeführt. Die höchste Priorität besitzen die internen Befehle. COM- und EXE-Dateien verfügen über zweite und BAT-Dateien über dritte Priorität. Externe Befehle sind an den letzten drei Buchstaben des Dateinamens erkennbar. Dies sind COM, EXE oder BAT. Eine nähere Beschreibung der Dateinamen finden Sie im Kapitel 7.4. |
| Beispiel | COPY ist ein interner Befehl. Sollte auf der Diskette/Festplatte ein externer Befehl mit dem Namen COPY.COM existieren, so wird bei der Eingabe von COPY nicht COPY.COM gestartet, sondern der interne Befehl. |

Suchreihenfolge der Dateien

Beachten Sie also immer folgende Suchreihenfolge bei der Ausführung:

1. Interne Befehle (Bestandteil von COMMAND.COM)
2. COM-Dateien (externe Befehle mit Namenszusatz COM)
3. EXE-Dateien (externe Befehle mit Namenszusatz EXE)
4. BAT-Dateien (externe Befehle mit Namenszusatz BAT)

Achten Sie darauf, daß ein Befehlsname immer nur in eindeutiger Form vorliegt, damit die Ausführung des richtigen Befehls gewährleistet ist.

Wenn ein Befehl vom MS-DOS nicht gefunden wird, erhalten Sie folgende Fehlermeldung:

Bad command or filename

COMMAND.COM

Dieses Programm wird beim Systemstart in den Speicher (RAM – Random Access Memory) geladen. COMMAND.COM besteht aus 3 Teilen:

1. Initialisierungsteil (nur beim Start des Computers notwendig)
2. Beständiger Teil (Programmlader, Fehlerbearbeitung usw.)
3. Überschreibbarer Teil (interne Befehle, Befehlsinterpreter usw.)

Der erste Teil wird von MS-DOS nicht mehr benötigt, wenn er seine Funktion erfüllt hat.

Der beständige Teil bleibt erhalten, bis Sie den Computer neu starten.

Der dritte Teil wird erst überschrieben, wenn der freie Speicher schon belegt ist und zur Ausführung eines Programmes mehr Speicher benötigt wird.

Diese Vorgänge, die MS-DOS automatisch für Sie verwaltet, sind nicht erkennbar.

5.2 Interne Befehle

Direkt ausführbare Befehle

Interne Befehle sind Teil des externen Befehls COMMAND.COM. Sie können diese Befehle jederzeit eingeben, wenn Sie sich in keinem Programm befinden, sondern auf Betriebssystemebene (es gibt zwischenzeitlich Programmiersprachen, wie z.B. GW-BASIC, die einen Aufruf aus einem Programm ermöglichen). Dies wird durch das Befehlsanforderungszeichen, den Prompt (Laufwerk-Buchstabe und > von COMMAND.COM), angezeigt.

Befehle	Funktion
BREAK	aktiviert/deaktiviert Programmunterbrechung
CHDIR oder CD	Directory wechseln
CLS	Bildschirm löschen
COPY	Dateien kopieren
CTTY	Peripheriegerät ändern, von dem Befehle eingegeben werden
DATE	Datum anzeigen/eingeben
DEL oder ERASE	Datei(en) löschen
DIR	Dateien anzeigen
ECHO	unterdrückt die Anzeige von Befehlszeilen in einer Batch-Datei; zeigt einen Kommentar an
EXIT	Verlassen des Betriebssystems und Rückkehr in ein Anwenderprogramm (in Verbindung mit COM-MAND)
FOR	mehrmaliges Wiederholen eines MS-DOS-Befehls in einer Batch-Datei
GOTO	unbedingter Sprungbefehl in einer Batch-Datei
IF	macht die Befehlsausführung von der Erfüllung einer Bedingung abhängig (in Batch-Datei)
MKDIR oder MD	Sub-Directory anlegen
PATH	benennt Wege, die zu einem Verzeichnis führen, in dem ein externer Befehl gesucht werden soll
PAUSE	unterbricht die Abarbeitung einer Batch-Datei und gibt einen Text auf dem Bildschirm aus (wenn fertig, Taste betätigen ...)
PROMPT	ändert Befehlsanforderungszeichen (Prompt)
REM	Kommentare in einer Batch-Datei
RENAME oder REN	Datei umbenennen
RMDIR oder RD	Sub-Directory löschen
SET	setzt den Wert einer Zeichenkette gleich dem Wert einer anderen Zeichenkette
SHIFT	erhöht die Zahl der Variablen in einer Batch-Datei auf mehr als 10
TIME	Uhrzeit anzeigen/ändern
TYPE	Datei-Inhalt anzeigen
VER	zeigt die Versionsnummer von IO.SYS und MS-DOS an
VERIFY	überprüft kopierte Dateien (Schalter)
VOL	zeigt den Namen einer Diskette oder Festplatte an (Label-Name)
d:	Arbeitslaufwerk umschalten

5.3 Externe Befehle

Dienstprogramme von MS-DOS

Externe Befehle sind Dienstprogramme, die vor jedem Start in den Speicher (RAM) von der Diskette/Festplatte geladen werden (führt COMMAND.COM automatisch durch). Ein nachfolgend aufgerufenes Programm (oder Befehl) überschreibt diese wieder.

Befehle	Funktion
APPEND	Einrichten eines Suchpfads für Datendateien
ASSIGN	leitet Laufwerksanfragen um
ATTRIB	Datei auf »Nur lesen« setzen
BACKUP (nur bei Festplattensystem)	Sichern von Dateien einer Festplatte
CHKDSK	überprüft die Platte und/oder eine Datei auf Fehler
COMMAND	führt interne Befehle und externe Befehle aus
COMP	vergleicht den Inhalt zweier Dateien auf Gleichheit
DEBUG	für Programmtest und -modifikation
DISKCOMP	vergleicht zwei Disketten
DISKCOPY	kopiert Disketten komplett
EDLIN	einfaches Textprogramm
EXE2BIN	formt Dateien von .EXE auf .COM um
FDISK	verwaltet die Bereiche einer Festplatte (IBM)
FIND	sucht einen Text
FORMAT	formatiert Diskette oder Festplatte
GRAFTABL	ASCII-Tabelle für Farbgrafik-Mode laden
GRAPHICS	Grafik- und Sonderzeichen auf angeschlossenem Drucker ausgeben
JOIN	temporäres Laufwerk durch Sub-Directory ersetzen
KEYBGR	Zeichensätze für verschiedene Länder laden
LABEL	Zuweisen/Anzeigen von Namen für Disks
MODE	modifiziert MS-DOS für Peripheriegeräte
MORE	zeigt die Bildschirmausgabe seitenweise an
PRINT	Ausgabe von Texten auf den Drucker
RECOVER	erneuert fehlerhafte Dateien
REPLACE	Ersetzen alter Versionen und Einfügen neuer Dateien
RESTORE (nur bei Festplattensystemen)	Zurückkopieren gesicherter Dateien auf Festplatte

Befehle	Funktion
SHARE	File-Sharing-Support laden (für MS-NET)
SORT	sortiert alphabetisch oder numerisch
SUBST	temporär Sub-Directory durch Laufwerk ersetzen
SYS	kopiert Systemdateien IO.SYS und MSDOS.SYS
TREE	zeigt das strukturierte Verzeichnis einer Platte
XCOPY	Kopieren von Dateien und Verzeichnissen

Kapitel 6

Diskette und Festplatte

RAM = interner Speicher

Ein interner Speicher ist in den Computer eingebaut und nennt sich RAM (Random Access Memory). Er wird bei Abschalten des Computers gelöscht.

Damit Sie Daten speichern können, ist ein externer Speicher notwendig. Dieser kann eine Diskette und/oder Festplatte sein.

Festplatte

Eine Festplatte besteht, im Gegensatz zur einzelnen Diskette, aus mehreren nicht biegsamen Platten (darum wird diese auch Hard-Disk genannt). Beide Speichermedien arbeiten im Prinzip auf die gleiche Art und Weise.

Externer Speicher

Damit Sie gezielt auf einen externen Speicher zugreifen können, werden die einzelnen Laufwerke mit Buchstaben bezeichnet. Zusätzlich ist ein Doppelpunkt zur Kennzeichnung notwendig (z. B. B:).

Anzahl der Diskettenlaufwerke	Diskettenlaufwerk- Namen	Festplattenlaufwerk- Namen
1 oder 2	A (und B)	C
3	A, B und C	D
4	A, B, C und D	E

6.1 Aktuelles Laufwerk

Anwählen eines Laufwerks

Geben Sie bei einem Befehl keinen Laufwerknamen an, so benutzt MS-DOS das aktuelle Laufwerk. Am Bildschirm ist es durch den Prompt erkenntlich. Dieser besteht immer aus dem Namen (z. B. A) und dem Zeichen >. Dies ist bei der Arbeit mit Dateien wichtig. Geben Sie einen externen Befehl ein, so sucht MS-DOS diesen auf dem aktuellen Laufwerk

(außer Sie geben vor dem Befehlsnamen zusätzlich den Laufwerknamen und einen Doppelpunkt ein).

Sie können das aktuelle Laufwerk durch die Eingabe eines Namens (z.B. B:) jederzeit ändern.

A> b: <RETURN>-Taste

B>

Sie dürfen kein nicht vorhandenes Laufwerk angeben, da Sie sonst folgende Fehlermeldung erhalten:

Ungültige Laufwerkangabe

Bei dem von Ihnen angewählten Laufwerk ist es nicht erforderlich, daß sich darauf MS-DOS befindet, da dieses bereits in den internen Speicher geladen wurde.

6.2 Einteilung von Disketten und Festplatte

Tracks und Sektoren

Die Informationen werden in konzentrischen Ringen auf der magnetisierten Oberfläche gespeichert. Diese Ringe nennt man Spuren (Tracks), die in einzelne Bereiche (Sektoren) unterteilt sind. Die Größe eines Bereichs ist bei Disketten und Festplatten unterschiedlich.

Informationseinheiten

Bit und Byte

Die Kapazität der einzelnen Bereiche wird in Byte ausgedrückt. Jedes Byte besteht aus 8 Bit (1 Bit ist die kleinste Informationseinheit in Ihrem Computer). Um einen Buchstaben oder ein anderes Zeichen darstellen zu können, benötigt Ihr Schneider PC 8 Bit (also 1 Byte).

Die Größe eines Speichers wird immer in Kilobyte (Kbyte) oder Megabyte (Mbyte) angegeben. Ein Kilobyte besteht aus 1024 Byte, ein Megabyte aus 1024 Kbyte (1.048.576 Byte).

Lesen und Schreiben der Informationen

Ein Elektromagnet im Laufwerk liest und schreibt die Informationen. Man nennt ihn Schreib-Lese-Kopf. Die Steuerung erfolgt durch MS-DOS, und er kann an jeder beliebigen Stelle der Diskette/Festplatte positioniert werden. Die genaue Adresse des gewünschten Bereiches errechnet das Betriebssystem aus den im Directory (Inhaltsverzeichnis) gespeicherten Informationen. In diesem Verzeichnis stehen die Dateinamen, die Angaben, wo die dazugehörigen Daten zu finden sind, usw.

MS-DOS speichert die Daten einer Datei nicht zusammenhängend in einem Stück. Um eine möglichst optimale

Platzausnutzung zu erreichen, wird die sogenannte dynamische Speicherung angewendet. Es wird der nächste freie Sektor gesucht und dort gespeichert. Dieser Vorgang wird so lange wiederholt, bis alle Daten gespeichert sind. Dabei kann es vorkommen, daß die Daten einer Datei über eine ganze Diskette oder Festplatte verstreut sind.

6.3 Disketten

Diskette als Speichermedium

Eine Diskette ist eine dünne, fein magnetisierte Plastikscheibe. Die Scheibe wird in einer Hülle aufbewahrt, die sie fast vollständig bedeckt. Nur ein schmaler Schlitz bleibt für das Lesen und Schreiben offen. Da die Diskette und die darauf gespeicherten Daten sehr leicht zerstört werden können,

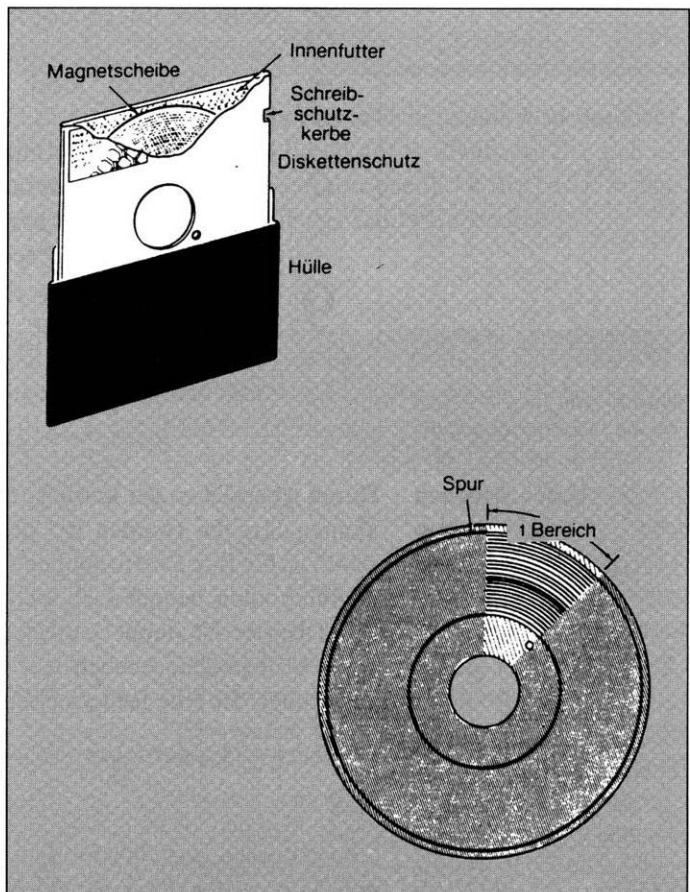


Bild 6.1: Die Diskette

sollten Sie diese immer sehr sorgfältig behandeln. Bitte beachten Sie vor allem folgendes:

- 1 Berühren Sie niemals die Diskettenoberfläche (vor allem nicht den Schreib-Lese-Schlitz)
- 2 Wenn Sie die Diskette dem Laufwerk entnehmen, stecken Sie sie sofort wieder in die Verpackung
- 3 Laufwerke erst öffnen, wenn die Kontroll-Lampe erloschen ist
- 4 Auf die Diskette dürfen kein Staub (z. B. Zigarettenasche) und keine Flüssigkeiten gelangen
- 5 Disketten dürfen nicht in der Umgebung von magnetischen Feldern aufbewahrt werden (auch keine statische Aufladung)
- 6 Nicht mit Kugelschreiber beschriften (möglichst Filzstift)
- 7 Mit Disketten nicht durch Röntgenanlagen gehen
- 8 Keine Disketten im Laufwerk lassen, wenn Sie den Strom ein- oder ausschalten. Durch eventuelle »Stromstöße« kann der Schreib-Lese-Kopf auf die Scheibe »stürzen« (Head-Crash)
- 9 Disketten »vertragen« nur Temperaturen von -10 bis +52 Grad Celsius
- 10 Disketten niemals biegen oder stapeln

Bild 6.2: Zehn goldene Regeln für die Diskettenbehandlung

**Daten schützen
vor Löschen
und Überschreiben**

Durch Überkleben der Kerbe auf der rechten Seite mit einem kleinen Streifen (werden bei neuen Disketten mitgeliefert) schützen Sie Ihre Daten vor dem Löschen und Überschreiben. Dadurch kann jedoch auch nicht mehr gespeichert werden! Die Diskette ist damit schreibgeschützt (Write protected). Wenn Sie trotzdem versuchen, auf der Diskette zu schreiben, so erhalten Sie eine Fehlermeldung:

Dateierstellungsfehler

oder

Schreibgeschützt Fehler schreiben Laufwerk A
Abbrechen, Wiederholen, Ignorieren?

Wenn Sie trotzdem speichern wollen, müssen Sie den kleinen Streifen von der Diskette entfernen und w betätigen. Im anderen Falle geben Sie den Buchstaben a ein, damit der Vorgang abgebrochen wird.

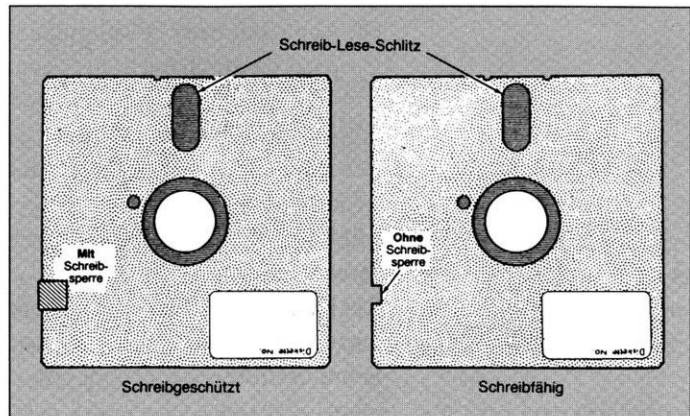


Bild 6.3:
Schreibsperrung der Diskette

Disketten mit dem Schreib-Lese-Schlitz zuerst in das Laufwerk einschieben. Das Bild ist auf der Seite, an der sich die Laufwerk-Klappe befindet (also oben). Ist die Diskette ganz eingeschoben, können Sie das Laufwerk schließen.

6.4 Festplatten

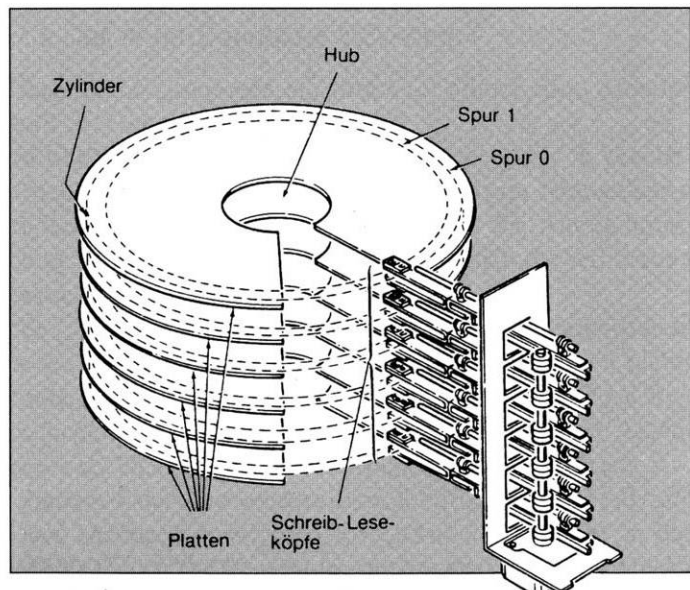


Bild 6.4: *Festplattenlaufwerk*

Festplatte als Speicher-Medium	Ein Festplattenlaufwerk besteht aus mehreren nicht biegsamen (hartes Aluminium), mit Metalloxid-Oberfläche versehenen Scheiben. Der externe Speicher ist in einem staubfreien, luftdicht abgeschlossenen Gehäuse untergebracht. Dadurch ist eine Festplatte viel sicherer als eine Diskette. Die Kapazität einer Festplatte ist wesentlich höher als die einer Diskette, da durch die höhere Sicherheit die Daten dichter auf den Platten gespeichert werden können.
Zugriffszeit	Wegen der wesentlich höheren Umdrehungsgeschwindigkeit ist die Zugriffszeit sehr kurz (mindestens ca. Faktor 10 schneller als bei einer Diskette). Mit MS-DOS können Sie maximal 32 Mbyte pro Laufwerk verwalten. Ein Nachteil der Festplatte ist, daß sie nicht wie eine Diskette leicht transportiert werden kann. Ferner hat sie keinen Schreibschutz.

Kapitel 7

Allgemeine Einführung in das Arbeiten mit MS-DOS

Dieses Kapitel ist für den Anfänger gedacht, der noch keinerlei Kenntnisse im Umgang mit MS-DOS hat. Als Grundlage für den Inhalt dient die Erfahrung, daß ca. 80 % der Anwender nur 20 % der Befehle des MS-DOS für die tägliche Arbeit benötigen.

»Umgang« mit MS-DOS

Sie werden sehr schnell feststellen, daß man kein erfahrener Programmierer oder »EDV-Mann« sein muß, um mit MS-DOS umzugehen. Ich habe versucht, diesen »Umgang« nachfolgend mit möglichst einfachen und verständlichen Worten zu beschreiben. Sollten Sie Vorschläge und Anregungen für dieses Kapitel bzw. das Buch haben, so würde ich mich über Ihre Zuschrift freuen.

7.1 Der neue Schneider PC kommt

Ausstattung des Schneider PC

Sie erhalten zusammen mit Ihrem Computer einen Monitor und ein Handbuch mit MS-DOS.

MS-DOS verbindet die einzelnen Teile Ihres Schneider-Personalcomputers (PC) logisch zu einem einheitlichen System. Ferner ist es erforderlich, damit Sie Anwendungsprogramme benutzen können (MS-DOS stellt die Verbindung zwischen Programm und Hardware dar).

Sie haben entweder ein Disketten- oder Festplattensystem erworben. Eine Festplatte ist im Grunde genommen nichts anderes als eine »fest eingebaute Diskette mit hoher Speicherkapazität«. Vergleichbar sind diese Speicher mit einem Aktenschrank. Wenn Sie bisher Unterlagen, Statistiken, Briefe usw. abgelegt haben, so werden diese künftig auf der Festplatte oder Diskette gespeichert (der Verwalter der Daten ist das MS-DOS bzw. Ihr Anwendungsprogramm). Dabei wird in der

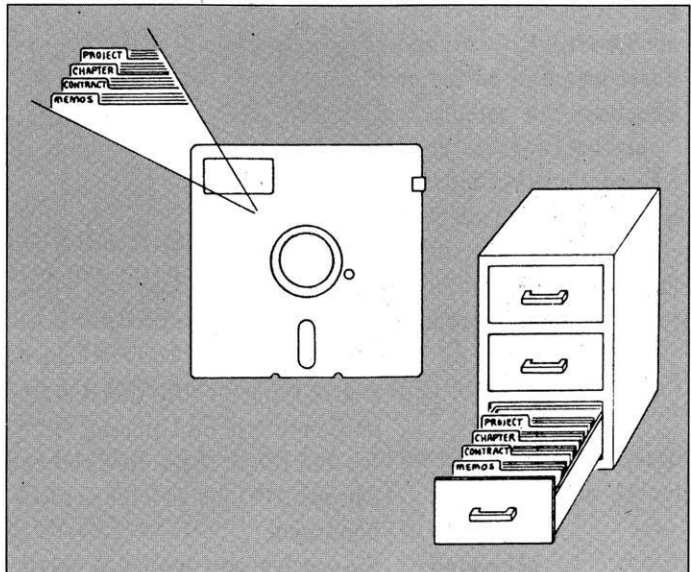


Bild 7.1: Die Diskette als Aktenschrank

EDV eine Kartei (z.B. Kunden, Lieferanten) als Datei bezeichnet.

Beachten Sie hierzu auch das Kapitel 6 »Diskette und Festplatte«.

7.2 Start mit MS-DOS

Schalten Sie bitte Ihren Computer ein. Als erstes kopieren Sie die MS-DOS-Diskette – verwenden Sie niemals Original-Disketten.

Starten von MS-DOS

Schalten Sie Ihren Schneider PC ein. Als erstes erscheint die Meldung

Bitte warten...

Es läuft am Anfang immer ein Testprogramm in Ihrem PC ab, das überprüft, ob alles in Ordnung ist. Nach kurzer Zeit erscheint die Meldung

Legen Sie eine SYSTEM-Diskette in Laufwerk A
Dann beliebige Taste drücken

Legen Sie also Ihre MS-DOS-Diskette (DISC1) in das linke Laufwerk und schließen Sie es. Anschließend drücken Sie eine beliebige Taste, um MS-DOS zu starten.

Start bei Festplattensystemen Bei Festplattensystemen wird das Betriebssystem vom Diskettenlaufwerk gestartet, wenn eine Systemdiskette eingelegt ist. Andernfalls erfolgt der Start von der Festplatte (diese muß jedoch vor der ersten Benutzung eingerichtet sein). Am Bildschirm erhalten Sie folgende Anzeige:

Aktuelles Datum ist Mit 29.10.1986

Neues Datum eingeben (tt-mm-jj):

Tagesdatum Geben Sie an dieser Stelle das Tagesdatum ein (bitte beachten Sie hierzu die Befehle DATE in Kapitel 11 und COUNTRY in Kapitel 13). Wenn Sie ohne weitere Eingabe nur die Taste <RETURN> drücken, so bleibt das angezeigte Datum unverändert. Das Datum in der Anzeige (es kann bei Ihnen ein anderes sein) ist lediglich ein Beispiel (bei dem angegebenen handelt es sich um den 29. Oktober 1986). Als nächstes erscheint am Bildschirm:

Aktuelle Zeit ist 20.45.06,68

Neue Zeit eingeben:

Uhrzeit Geben Sie an dieser Stelle die Uhrzeit ein. Wenn Sie ohne weitere Eingabe nur die Taste <RETURN> drücken, so bleibt die angezeigte Uhrzeit unverändert. Beachten Sie hierzu auch den Befehl TIME im Kapitel 11. Die Uhrzeit in der Anzeige kann bei Ihnen anders lauten (hier: 20 Uhr 45 Minuten 6 Sekunden 68 Hundertstel).

Wenn Sie sich beim Datum oder der Uhrzeit vertippen (vorausgesetzt, Sie haben noch nicht <RETURN> gedrückt), so können Sie mit der Taste Back Space ← wieder rückwärts löschen und die Eingabe korrekt vornehmen.

Zum Schluß erscheint am Bildschirm:

Microsoft(R) MS-DOS(R) Version 3.20

(C)Copyright Microsoft Corp 1981-1986

A>

System-Bereitschaftsmeldung Die Angabe A> ist der MS-DOS-Prompt, die System-Bereitschaftsmeldung. Damit erwartet das Betriebssystem von Ihnen Befehle.

Wenn Sie das Datum und/oder die Uhrzeit nicht richtig eingegeben haben, so korrigieren Sie dies durch Eingabe der Befehle DATE und/oder TIME.

7.3 Installation von MS-DOS

Herstellen von Betriebssystem-Kopien

Wie bereits erwähnt, sollten Sie niemals mit Original-Programmdisketten arbeiten. Für die Erstellung von Kopien ist eine unterschiedliche Handhabung bei PCs mit einem oder zwei Diskettenlaufwerken und solchen mit einer Festplatte erforderlich.

7.3.1 Kopieren von Disketten auf PCs mit 2 Diskettenlaufwerken

DISKCOPY

Legen Sie Ihre MS-DOS-Diskette in Laufwerk A (links) ein. Zum Kopieren benötigen Sie den Befehl DISKCOPY (siehe auch Kapitel 11):

```
A>diskcopy a: b:
```

Nach dem Befehl und a: die Leertaste (Leerschritt) und nach b: die <RETURN>-Taste betätigen. Damit wird der Befehl von der MS-DOS-Diskette gelesen und in den Speicher (RAM) geladen, um diesen dort abzuarbeiten. Sie werden vom Programm aufgefordert, das Original in Laufwerk A: und eine neue Diskette in Laufwerk B: (rechts) einzulegen und <RETURN> zu drücken. Damit wird der Kopiervorgang gestartet.

Legen Sie die QUELL-Diskette in Laufwerk A

Legen Sie die ZIEL-Diskette in Laufwerk B

Dann beliebige Taste drücken

Die neue Diskette wird formatiert und eine 1:1-Kopie vom Original erstellt.

Kopieren: 40 Spuren

9 Sektoren/Spur, 2 Seiten

Formatiert während des Kopierens

Ende des Kopiervorgangs

Wenn der Kopiervorgang beendet ist, werden Sie noch befragt, ob Sie eine weitere Diskette kopieren wollen.

Kopieren einer anderen Diskette (J/N)?

Drücken Sie bitte j (JA) und <RETURN>, wenn Sie dies wünschen (dazu eine weitere Originaldiskette in Laufwerk A:

und eine neue in B: einlegen). Im anderen Falle n (NEIN) und <RETURN>.

Bitte verwenden Sie nur Kopien von MS-DOS und anderen Programmen für die Arbeit am Computer. Empfehlenswert ist es, noch eine zweite Kopie zu erstellen. Dies ist erforderlich, da Disketten nicht »ewig« halten, sondern einer Abnutzung unterliegen und im Falle von Zeitmangel oder einem reibungslosen Ablauf dann sofort weitergearbeitet werden kann. Sie können dann später problemlos weitere Kopien vom Original erstellen.

**Formatieren
von Disketten**

Um leere Disketten für die Aufnahme von Daten zu erstellen, benötigen Sie den Befehl FORMAT (die Beschreibung zu diesem Befehl finden Sie im Kapitel 11).

7.3.2 Kopieren von Disketten auf PCs mit einem Diskettenlaufwerk

**Kopieren
mit einem Laufwerk**

Da Ihnen in der Regel bei einem Festplattensystem nur ein Diskettenlaufwerk zur Verfügung steht, kann der Kopiervorgang nicht von Laufwerk zu Laufwerk erfolgen. Das gleiche gilt, wenn Sie einen Schneider PC mit nur einem Diskettenlaufwerk ohne Festplatte besitzen.

DISKCOPY

Legen Sie Ihre MS-DOS-Diskette in das Laufwerk ein. Zum Kopieren benötigen Sie den Befehl DISKCOPY (siehe auch Kapitel 11):

A>diskcopy a: b:

Nach dem Befehl und a: die Leertaste (Leerschritt) und nach b: die <RETURN>-Taste betätigen. Damit wird der Befehl von der MS-DOS-Diskette gelesen und in den Speicher (RAM) geladen, um diesen dort abzuarbeiten.

Legen Sie die QUELL-Diskette in Laufwerk A
Dann beliebige Taste drücken

Nach Drücken einer beliebigen Taste erscheint kurze Zeit später

Kopieren: 40 Spuren
9 Sektoren/Spur, 2 Seiten

Anschließend wird der Inhalt der Diskette in den Speicher des PC gelesen. Wenn dies abgeschlossen ist, erscheint die Meldung

Legen Sie die ZIEL-Diskette in Laufwerk A
Dann beliebige Taste drücken

Entnehmen Sie die Originaldiskette, und legen Sie eine neue in das Diskettenlaufwerk. Betätigen Sie eine Taste, und der Kopiervorgang beginnt. Bei einer noch nicht benutzten Diskette erscheint die Meldung

Formatiert, während des Kopierens

**Ende des
Kopiervorgangs**

Wenn der Kopiervorgang beendet ist, werden Sie noch befragt, ob eine weitere Diskette kopiert werden soll. Drücken Sie bitte j und <RETURN>, wenn Sie weitere kopieren möchten. Im anderen Falle n und <RETURN>.

Kopieren einer anderen Diskette (J/N)?

Bitte verwenden Sie nur Kopien von MS-DOS und anderen Programmen für die Arbeit am Computer. Empfehlenswert ist es, noch eine zweite Kopie zu erstellen. Dies ist erforderlich, da Disketten nicht »ewig« halten, sondern einer Abnutzung unterliegen und im Falle von Zeitmangel oder einem reibungslosen Ablauf dann sofort weitergearbeitet werden kann. Später lassen sich problemlos weitere Kopien vom Original erstellen.

**Formatieren
von Disketten**

Um leere Disketten für die Aufnahme von Daten zu erstellen, benötigen Sie den Befehl FORMAT (die Beschreibung zu diesem Befehl finden Sie im Kapitel 11).

7.3.3 Einrichten einer neuen Festplatte mit MS-DOS

Bevor Sie die Platte verwenden können, muß diese eingerichtet werden. Beachten Sie auch das Kapitel 6 »Diskette und Festplatte«.

**Einrichten einer
Festplatte**

Legen Sie Ihre MS-DOS-Diskette in das linke Laufwerk, falls sich diese noch nicht dort befindet. Geben Sie

fdisk

ein, um das Programm zu starten. Daraufhin erhalten Sie folgende Anzeige am Bildschirm:

Festplatten Initialisierungsprogramm Version 0.01
(C)Copyright Microsoft, 1985.

FDISK Optionen

Wählen Sie bitte:

1. DOS Partition erstellen
2. Aktive Partition ändern
3. DOS Partition löschen
4. Daten der Partition anzeigen

Geben Sie die Ziffer ein:[1]

Drücke ESC um zu DOS zurückzukehren

Geben Sie die Ziffer 1 ein und betätigen Sie <RETURN>. Wenn die Festplatte noch nicht für MS-DOS vorbereitet war, so werden Sie befragt, ob die gesamte Kapazität für das Betriebssystem eingeteilt werden soll.

DOS-Partition

Sollte bereits eine DOS-Partition vorhanden sein, so muß diese erst gelöscht werden, bevor die Festplatte neu eingeteilt werden kann.

Nach der Vorbereitung der Festplatte startet Ihr Schneider PC neu (dazu muß die MS-DOS-Diskette im Diskettenlaufwerk vorhanden sein).

FDISK

Starten Sie das Programm FDISK erneut und wählen Sie den Punkt 2. Dazu erscheint z.B.:

Aktive Partition wechseln

Partition	Status	Typ	Start	Ende	Größe
1	A	DOS	0	304	305
2	N	xxx	xxx	xxx	xxx
3	N	xxx	xxx	xxx	xxx
4	N	xxx	xxx	xxx	xxx

Die Platte hat 305 Zylinder.

Die momentan aktive Partition ist 1.

Geben Sie die Nummer der Partition ein
die Sie aktivieren möchten:[]

Drücke ESC um ins Menü zurückzukehren

Anzeige der Partitions

Es werden Ihnen alle Partitions angezeigt, und Sie können wählen, welche Sie aktiv setzen wollen (es kann immer nur eine aktiv sein!). Wie Sie aus obiger Anzeige ersehen können, ist es möglich, bis zu 4 Partitions einzurichten. Zu beachten ist dabei, daß immer nur mit der aktiven gearbeitet werden kann.

Starten von MS-DOS auf Festplatte

Wenn Sie Ihren Schneider PC mit dem MS-DOS auf der Festplatte starten wollen, so darf sich keine Diskette im Laufwerk befinden. Nach dem Einschalten startet Ihr PC das MS-DOS von der aktiven Partition.

Wenn Sie im Menü den Punkt 4 anwählen, wird Ihnen der momentane Stand der Festplatteneinteilung angezeigt.

FDISK und FORMAT

Nachdem mit FDISK die Festplatte vorbereitet wurde, muß diese noch formatiert werden (Befehl FORMAT – Kapitel 11). Erst dann ist die Platte bereit, um mit ihr arbeiten zu können. Betätigen Sie die Taste Esc zweimal, um das Programm FDISK zu verlassen. Um die Festplatte zu formatieren, geben Sie ein:

```
format c:/s/v
```

Einrichten der Festplatte löscht alle Daten

Bitte beachten Sie, daß bei der Einrichtung der Festplatte mit FDISK und FORMAT alle eventuell vorher vorhandenen Dateien gelöscht werden.

Der Formatierungs-Vorgang kann je nach Größe der Festplatte mehrere Minuten dauern. Anschließend können Sie noch einen Namen mit maximal 11 Stellen zuweisen, oder Sie betätigen nur die Taste <RETURN>. Am Bildschirm erscheint nach dem Formatieren und der Übertragung des MS-DOS:

```
Formatierung beendet
xxxxxxx Bytes insgesamt auf Diskette
xxxxxxx Bytes verfügbar auf Diskette
```

Nochmal formatieren (J/N)?

Übertragen des DOS-Systems

Nach Beendigung erscheint wieder der Prompt A>.

Beim Formatieren wurde durch die Angabe von /S das MS-DOS-System (IO.SYS, MSDOS.SYS und COMMAND.COM) übertragen. Die Dienstprogramme bzw. alle weiteren auf der Diskette vorhandenen Programme können mit folgendem Befehl auf die Festplatte kopiert werden:

copy a:*. * c:

Festplatten- Formatierung beendet

Wenn wieder A> erscheint, haben Sie eine komplette Kopie von MS-DOS auf der Festplatte, und der Vorgang ist abgeschlossen.

Sie sollten jedoch in jedem Fall noch eine Kopie Ihrer MS-DOS-Diskette fertigen. Gehen Sie dabei wie unter 7.3.2 beschrieben vor.

Kopie prüfen

Um die Richtigkeit der Kopie zu überprüfen, gehen Sie wie folgt vor: Drücken Sie Ctrl+Alt+Del gleichzeitig, um das System neu zu starten. Damit Ihr Computer von der Festplatte starten kann, nehmen Sie bitte die Diskette aus dem Laufwerk (andernfalls versucht das System, MS-DOS von der Diskette zu starten).

Geben Sie Datum und Uhrzeit ein, und tippen Sie den Befehl für die Anzeige des Directory (Inhaltsverzeichnis der Festplatte) ein

dir

und betätigen Sie <RETURN>.

Überprüfen Sie anhand der Anzeige, ob alle Dateien vorhanden sind.

Beachten Sie die Beschreibung zu den Befehlen FDISK, FORMAT und COPY im Kapitel 11.

7.4 Die wichtigsten Befehle im Überblick

7.4.1 Allgemeines (Aufbau der Dateinamen)

Hier stelle ich Ihnen als erstes den Aufbau der Datei/Programm-Namen vor. Ferner die Befehle DIR, MODE, COPY, RENAME, DEL.

Eingabe von Befehlen und Programm-Namen

Wenn Sie den Prompt (z. B. A>) am Bildschirm mit dem Cursor daneben sehen, erwartet MS-DOS die Eingabe eines Befehls. Diesen können Sie in Klein- oder Großbuchstaben eingeben (Kleinbuchstaben behandelt das MS-DOS nach der Eingabe automatisch wie Großbuchstaben – ohne daß dies angezeigt wird).

Wenn Sie sich ansehen wollen, was auf Ihrer Diskette oder Festplatte an Programmen vorhanden ist, so geben Sie den Befehl DIR und <RETURN> ein. Dabei erscheint z. B. am Bildschirm:

Volume in Laufwerk A hat keinen Namen
Verzeichnis von A:\

COMMAND	COM	24044	22.08.86	14.15
ANSI	SYS	1699	22.08.86	14.15
CONFIG	SYS	60	22.08.86	14.15
DRIVER	SYS	1112	22.08.86	14.15
RAMDRIVE	SYS	6614	22.08.86	14.15
ANSI	COM	139	22.08.86	14.16
APPEND	COM	1747	22.08.86	14.16
ASSIGN	COM	1524	22.08.86	14.16
MORE	COM	280	22.08.86	14.16
MOUSE	COM	6990	22.08.86	14.16
SYS	COM	4687	22.08.86	14.16
ATTRIB	EXE	8248	22.08.86	14.16
CHKDSK	EXE	10096	22.08.86	14.16
COMP	EXE	14500	22.08.86	14.16
DEBUG	EXE	15701	22.08.86	14.16
DISKCOMP	EXE	3872	22.08.86	14.17
DISKCOPY	EXE	3984	22.08.86	14.17
EDLIN	EXE	7484	22.08.86	14.17
EXE2BIN	EXE	3082	22.08.86	14.17
FDISK	EXE	16600	22.08.86	14.17
FIND	EXE	6414	22.08.86	14.17
FORMAT	EXE	11277	22.08.86	14.17
GRAFTABL	EXE	8198	22.08.86	14.17
GRAPHICS	EXE	13308	22.08.86	14.17
JOIN	EXE	8950	22.08.86	14.17
KEYBGR	EXE	3029	22.08.86	14.17
LABEL	EXE	2770	22.08.86	14.18
MODE	EXE	13998	22.08.86	14.18
PRINT	EXE	8952	22.08.86	14.18
RECOVER	EXE	4257	22.08.86	14.18
REPLACE	EXE	4952	22.08.86	14.18
SHARE	EXE	8544	22.08.86	14.18
SORT	EXE	1898	22.08.86	14.18
SUBST	EXE	9910	22.08.86	14.18
TREE	EXE	8568	22.08.86	14.18
XCOPY	EXE	5484	22.08.86	14.19

36 Datei(en) 1024 Bytes frei

Bestandteile eines Dateinamens

Sie finden auf der linken Seite der Anzeige eine Reihe von Dateinamen. Diese bestehen aus 2 Teilen, dem File-Namen (maximal 8 Stellen) und File-Typ (maximal 3 Stellen – dieser muß nicht vorhanden sein).

File-Typ

Der Namenszusatz (File-Typ) gibt meist Auskunft über den Inhalt einer Datei. Es haben sich dabei die folgenden wichtigsten Typen-Bezeichnungen ergeben (diese sind nicht alle zwingend, jedoch gebräuchlich):

Typ	Verwendung für
COM	Programm
EXE	Programm
BAT	Batch-Datei, die eine Reihe von Programmaufrufen beinhaltet (siehe Kapitel 12)
ASM	Assembler-Quellen-Programme
COB	COBOL-Quellen-Programme oder
CBL	COBOL-Quellen-Programme
BAS	BASIC-Quellen-Programme
FOR	Fortran-Quellen-Programme
LST	Liste/Protokoll einer Programmübersetzung
OBJ	Objektdatei; erzeugt durch einen Compiler
OVR	Overlay-Datei
TXT	Texte (z.B. Briefe, Berichte) oder
DOC	Texte
BAK	Sicherungsdatei
BIN	Binärdatei
DAT	ASCII-Datei (Daten wie z.B. Adressen, Artikel)
DIF	Datenübergabe-Datei
SYS	Systemdatei
TMP	Temporäre Datei
\$\$\$	Arbeitsdatei (temporäre Datei)

File-Name

Der Dateiname (8 Stellen maximal für File-Namen und 3 für den Typ) kann aus beliebigen Buchstaben und/oder Ziffern bestehen. Sonderzeichen (Leerzeichen, Punkt usw.) sollten grundsätzlich nicht verwendet werden.

Erlaubte Zeichen:

A-Z	0-9	\$	&	#	%	'	(	)
-	@	^	{	}	~	!	_	

Nicht erlaubte Zeichen:

? . , ; : = * / \
+ " < >

Bei Angabe des Dateinamens wird der File-Name vom File-Typ durch einen Punkt getrennt:

BRIEF.DOC
PROGRAMM.COM
PROGRAMM.EXE
BASIC.BAS
01-01-87.TXT

**Dateinamen müssen
sich unterscheiden**

Wichtig ist, daß die Dateinamen in einem Directory (siehe Kapitel 8) nicht gleich sein dürfen, damit man MS-DOS-Texte, Programme usw. vom Namen her unterscheiden kann. Zusätzlich hat jedes Disketten- und Festplattenlaufwerk unter MS-DOS ebenfalls einen Namen:

Anzahl der Diskettenlaufwerke	Diskettenlaufwerk- Namen	Festplattenlaufwerk- Namen
1 oder 2	A (und B)	C
3	A, B und C	D
4	A, B, C und D	E

**Programme von
anderen Laufwerken
starten**

Wenn ein Programm oder eine Datei nicht im aktuellen Laufwerk zu finden ist (steht A > am Bildschirm, so ist das aktuelle Laufwerk A), so muß vor dem Dateinamen der Laufwerksname mit einem zusätzlichen : angegeben werden:

c:program
b:program

Laufwerk wechseln

Wie an den obigen Beispielen erkennbar, werden Programme immer ohne Angabe des File-Typs zum Starten angegeben. Wie kann das aktuelle Laufwerk gewechselt werden?

Tippen Sie den Namen des neuen aktuellen Laufwerks (z.B. B:) ein, und betätigen Sie die Taste <RETURN>. Anschließend erscheint als Bestätigung ein neuer Prompt (z.B. B >). Sollte ein von Ihnen angegebener Dateiname von MS-DOS nicht gefunden werden, so erscheint am Bildschirm die Meldung:

Dateiname falsch oder fehlt

Überprüfen Sie in diesem Fall, ob Sie den Namen richtig eingetippt haben und die Datei auf dem aktuellen Laufwerk oder gegebenenfalls einem anderen vorhanden ist.

Beachten Sie auch das Kapitel 8 »Das MS-DOS-Inhaltsverzeichnis«.

7.4.2 Zeichen zur Zusammenfassung mehrerer Dateien

Bisher haben Sie sich ausschließlich mit genau spezifizierten Dateinamen befaßt. Manchmal ist es jedoch erforderlich, Gruppen von Dateien anzusprechen (z.B. wenn alle BASIC-Programme oder Texte kopiert werden). Dazu benötigen Sie sogenannte Wildcard-Zeichen (* und ?). Im folgenden dazu Beispiele.

Wildcard-Zeichen

Auf einer Diskette in Laufwerk A: sind die Dateien

```
TEST1DAT.TXT  
TEST2DAT.TXT  
TESTADAT.TXT  
TESTNEU1.TXT  
TESTNEU2.TXT  
TESTNEUE.DOC  
PROGRAMM.COM
```

gespeichert. Sie kopieren alle 5 TXT-Dateien von Laufwerk A nach B:

```
copy a:test1dat.txt b:  
copy a:test2dat.txt b:  
copy a:testadat.txt b:  
copy a:testneu1.txt b:  
copy a:testneu2.txt b:
```

oder

```
copy a:*.txt b:
```

oder

```
copy a:test*.txt b:
```

Ein Stern kann für mehrere Zeichen benutzt werden. Er kann für ein oder aber auch für alle Zeichen des File-Namens und

-Typs benutzt werden. Ab der Stelle, an welcher er eingegeben wird, ersetzt er alle Zeichen.

Im vorhergehenden zweiten Beispiel (*.txt) steht der * für beliebige 8 Stellen vor dem Punkt.

Wenn vor dem Stern Namensteile aufgeführt werden (text*.txt), so werden diese mit den entsprechenden Stellen der gespeicherten Dateinamen verglichen und alle weiteren (bis einschließlich der 8. Stelle) können beliebig sein.

Nicht erlaubte Wildcard-Position

Da der Stern ab der Stelle, an der er eingesetzt wird, alle Zeichen ersetzt, ist eine Eingabe wie *dat.txt nicht erlaubt (MS-DOS akzeptiert die Eingabe, behandelt dies jedoch wie *.txt).

Das gleiche gilt für den File-Typ. Wenn Sie

```
copy a:test*.* b:
```

eingeben, so wird jede Datei kopiert, die mit den Buchstaben test beginnt (also zusätzlich zu obigen Dateien auch noch TESTNEUE.DOC). Der File-Typ kann beliebig sein.

Mit ** werden alle Dateien angesprochen.

Kopieren mit Wildcards

Die Eingabe des Kopierbefehls

```
copy a:test*.* b:
```

führt im vorliegenden Beispiel zu der Meldung 0 Datei(en) kopiert, da ein File-Name »TEST« nicht existiert. Richtig müßte der Befehl lauten:

```
copy a:test*.* b:
```

Das Fragezeichen (?) kann für einen einzelnen Buchstaben im Dateinamen stehen:

```
copy a:test?dat.txt b:
```

Es werden alle gespeicherten Dateien kopiert, die bis auf den fünften Buchstaben des Dateinamens mit der Angabe des Namens im Kopierbefehl übereinstimmen:

```
TEST1DAT.TXT
```

```
TEST2DAT.TXT
```

```
TESTADAT.TXT
```

Das Fragezeichen darf an jeder beliebigen Stelle des File-Namens und File-Typs benutzt werden.

Die beiden Kopierbefehle

```
copy a:test????.txt b:
```

```
copy a:test*.txt b:
```

führen somit im vorliegenden Beispiel zum gleichen Ergebnis.

7.4.3 Befehle DIR, COPY, RENAME und DEL (ERASE)

Anzeige eines Disketten- Inhaltsverzeichnisses (Directory) mit DIR

Die Angaben im Kapitel 7.4.2 zur Zusammenfassung von Dateien gelten auch für den Befehl DIR. Die folgenden beiden Beispiele führen zum gleichen Resultat:

```
dir a:
```

```
dir a:*.*
```

Als Ergebnis des Befehls DIR wird am Bildschirm ein Directory (hier von der Diskette) angezeigt:

```
Volume in Laufwerk A ist MS-DOS 3
```

```
Verzeichnis von A:\
```

```
IO      SYS      21357  20.05.85  11.35
MSDOS   SYS      27776  19.05.85  14.42
COMMAND COM      20703   1.01.80   0.04
COMPILER TXT      2048  26.08.85  15.40
      4 Datei(en) 288768 Bytes frei
```

Aufbau des Directory

In der ersten Zeile finden Sie den Namen der Diskette/Festplatte, soweit vorhanden (hier: MS-DOS 3). Dieser wird beim Formatieren eingegeben oder kann nachträglich mit dem Befehl LABEL verändert werden (siehe Kapitel 11, Befehle FORMAT und LABEL). Sollte kein Name vorhanden sein, so wird der Text »Volume in Laufwerk A hat keinen Namen« angezeigt.

In der zweiten Zeile finden Sie den Namen des Laufwerks (in unserem Beispiel A:) und des Directory (Näheres siehe Kapitel 8 »Das MS-DOS-Inhaltsverzeichnis«). Die Anzeige \ bedeutet das erste (ROOT-)Directory.

Am Ende erhalten Sie noch die Angaben, wie viele Dateien angezeigt wurden (nicht unbedingt die Anzahl der im Directory gespeicherten) und wieviel freie Speicherkapazität noch

auf der Diskette/Festplatte ist (in unserem Beispiel sind auf der Diskette noch 288768 Zeichen frei für das Speichern von weiteren Daten/Dateien).

In den Zeilen dazwischen finden Sie die Datei-Einträge.

```
COMMAND  COM      20703  16.04.86  19:34
1234567890123456789012345678901234567890
                10          20          30          40
```

**Zusammensetzung der
Directory-Einträge**

COMMAND	File-Name
COM	File-Typ
20703	Dateigröße (in Bytes bzw. Zeichen)
16.04.86	Datum der Erstellung bzw. letzten Änderung
19:34	Uhrzeit der Erstellung bzw. letzten Änderung

Die Spalten werden z. B. für den folgenden Befehl benötigt:

```
dir | sort /+14
```

Es wird das Inhaltsverzeichnis sortiert ab der Spalte 14 angezeigt (also nach Dateigröße aufsteigend anzeigen).

**Directory »seitenweise«
anzeigen**

Wenn für die Anzeige eines Directory die Zeilen am Bildschirm nicht ausreichend sind (mehr als 25 Zeilen), so können Sie einen der beiden folgenden Befehle (»seitenweise vorwärts blättern«) verwenden:

```
dir *.* | more
```

oder

```
dir *.* /p
```

Kopieren mit COPY

Den Befehl COPY haben Sie bereits im Kapitel 7.4.2 kennengelernt. Weiteres finden Sie im Kapitel 7.5 für das Kopieren von Texten auf den Drucker und eine detaillierte Beschreibung im Kapitel 11.

Als erstes wird bei COPY das »woher« und als zweites das »wohin« angegeben:

copy	autoexec.bat	exec.bat
kopiere	woher	wohin

Im vorliegenden Beispiel sucht das Programm COPY die Datei AUTOEXEC.BAT und erstellt davon eine Kopie, die unter dem Namen EXEC.BAT gespeichert wird. Nach Aus-

**Laufwerk-Angabe
beim Kopieren**

führung des Befehls existiert von AUTOEXEC.BAT eine Kopie gleichen Inhalts mit dem Namen EXEC.BAT.

Normalerweise wird von einem Laufwerk auf ein anderes kopiert. Die Laufwerksbezeichnung müssen Sie nur angeben, wenn diese vom angezeigten Prompt (z.B. A> für aktuelles Laufwerk A) abweicht. Diese Angabe ist immer vor den Dateinamen ohne Zwischenraum erforderlich.

Erhält das »wohin« den gleichen Namen wie das »woher«, so kann auf eine zweite Dateiangabe verzichtet werden.

Beispiele zum Kopierbefehl:

```
copy c:autoexec.bat a:exec.bat
```

AUTOEXEC.BAT von Laufwerk C nach A kopieren – neuer Name EXEC.BAT

```
copy autoexec.bat a:
```

AUTOEXEC.BAT vom aktuellen nach Laufwerk A mit dem gleichen Namen kopieren

```
copy b:keybgr.com
```

KEYBGR.COM vom Laufwerk B in das aktuelle mit dem gleichen Namen kopieren (das »wohin« muß hier nicht angegeben werden)

**Dateien umbenennen
mit RENAME**

Oft soll keine Kopie erstellt werden, sondern eine Datei einen anderen Namen erhalten. Hierzu verwenden Sie statt COPY den Befehl RENAME. Dieser wird wie COPY angewandt.

```
rename autoexec.bat exec.bat
```

**Kurzform von
RENAME = REN**

Im folgenden noch 4 Beispiele, wie in einer möglichst kurzen Form eine oder mehrere Dateien umbenannt werden können (REN ist die Kurzform von RENAME):

```
ren b:ansi.doc *.txt
```

Im Laufwerk B die Datei ANSI.DOC in ANSI.TXT umbenennen (wenn der File-Name oder File-Typ gleichbleiben soll, so kann dieser mit * abgekürzt werden)

```
ren b:*.doc *.txt
```

Der File-Typ DOC wird in TXT geändert (bei allen gefundenen Dateien auf der Diskette in Laufwerk B)


```
ren b:ansi bild
```

Die Datei ANSI in BILD umbenennen
(Diskette in Laufwerk B)

```
ren programm.* neuprog.*
```

Alle Dateien mit dem File-Namen PROGRAMM in NEUPROG ändern (der File-Typ bleibt unverändert)

Wird nur bei der ersten Dateibezeichnung eine Laufwerksbezeichnung angegeben, so gilt diese auch für die zweite Angabe.

Dateien löschen mit DEL oder ERASE

Natürlich benötigen Sie auch einen Befehl zum Löschen von Dateien. Mit DEL oder ERASE (beide Befehle sind in der Bedienung und Auswirkung gleich) können Dateien gelöscht werden:

```
del b:*.*
```

Alle Dateien im Directory auf der Diskette in Laufwerk B werden gelöscht (wenn diese nicht mit dem Programm ATTRIB geschützt sind). Vor dem Löschen werden Sie noch befragt:

Sind Sie sicher (J/N)?

Die Eingabe J und <RETURN> löscht alles.

```
del b:exec.bat
```

Auf der Diskette in Laufwerk B wird EXEC.BAT gelöscht

```
del *.doc
```

Alle Dateien mit dem File-Typ DOC im aktuellen Laufwerk löschen

```
del neuprog.*
```

Alle Dateien mit dem File-Namen NEUPROG im aktuellen Laufwerk löschen

```
del a:test?dat.txt
```

Alle Dateien wie angegeben werden gelöscht, wobei an der 5. Stelle ein beliebiges Zeichen stehen kann (Laufwerk A).

7.4.4 Befehl MODE für die Anpassung des Druckers usw.

Anpassung der Peripherie mit MODE

Mit dem Befehl kann folgendes eingestellt werden:

1. Einstellung des »Weges« zum Drucker
2. Einstellung der Video-(Bildschirm-)Karte (Grafik usw.)
3. Einstellung der Schnittstelle für die Übertragung von Daten zu anderen Computern (über Akustikkoppler, Postmodem usw.)

Drucker einstellen oder umleiten

Wir wollen uns hier nur mit dem Drucker befassen. Weitere Erklärungen zum Befehl MODE finden Sie im Kapitel 11. Als erstes klären Sie bitte ab, ob Ihr Drucker an der parallelen (Normalfall) oder seriellen Schnittstelle angeschlossen ist. Wenn Ihr Drucker an der seriellen Schnittstelle angeschlossen ist, so können Sie die standardmäßige Druckausgabe von der parallelen dorthin umleiten:

```
mode lpt1=com1
```

Es können bis zu 3 parallele Schnittstellen an Ihrem Computer vorhanden sein, welche MS-DOS benutzt. Diese bezeichnet man als LPT1, LPT2 und LPT3. Mit MODE können diese eingestellt werden:

Standardeinstellungen des Druckers

1. Drucker mit 80 Zeichen pro Zeile
Drucker mit 132 Zeichen pro Zeile
2. Druck mit 6 Zeilen pro Inch (1 Inch = 2,54 cm)
Druck mit 8 Zeilen pro Inch
3. Normalerweise erscheint eine Fehlermeldung, wenn etwas gedruckt werden soll und dies nicht erfolgen kann (z. B. weil der Drucker nicht eingeschaltet ist). Dies kann abgeschaltet werden, indem der Buchstabe P beim Befehl MODE mitgegeben wird (dann wartet der Computer so lange, bis gedruckt werden kann).

Die Standardeinstellung wurde oben entsprechend gekennzeichnet (80 Zeichen, 6 Zeilen).

Manche Programmier- sprachen überschreiben MODE

Bitte beachten Sie, daß verschiedene Programmiersprachen (z. B. BASIC) Ihre Einstellungen mit MODE überschreiben (also unwirksam machen). Es muß also mit der Programmiersprache die Einstellung erneut vorgenommen werden.

Beispiele zur Druckereinstellung

Beispiele:

mode lpt1:132,8,P

1. Parallel-Schnittstelle

132 Zeichen pro Zeile

8 Zeilen pro Inch

warten, bis der Drucker »empfangsbereit« ist

mode lpt1:80,8

1. Parallel-Schnittstelle

80 Zeichen pro Zeile

8 Zeilen pro Inch

Fehlermeldung, wenn Drucker nicht »empfangsbereit«

mode lpt1:80,6,P

1. Parallel-Schnittstelle

80 Zeichen pro Zeile

6 Zeilen pro Inch

warten, bis der Drucker »empfangsbereit« ist

mode lpt2:132,6,P

2. Parallel-Schnittstelle

132 Zeichen pro Zeile

6 Zeilen pro Inch

warten, bis der Drucker »empfangsbereit« ist

7.5 Drucken von Texten und vom Bildschirm

Bildschirm-Inhalt auf Drucker ausgeben

Sie können alles, was am Bildschirm angezeigt wird, ausdrucken (dies ist auch möglich, wenn Sie gerade ein Anwendungsprogramm, wie z.B. eine Finanzbuchhaltung, bearbeiten). Betätigen Sie dazu nur die Tasten Shift und gleichzeitig PrtSc. Damit wird alles 1 zu 1 auf das Papier im Drucker »kopiert«. Jedes Zeichen kann jedoch nur dargestellt werden, wenn Sie einen IBM-kompatiblen Drucker mit 8-Bit-Zeichensatz verwenden.

Bildschirm-Ausgabe gleichzeitig auf Drucker umleiten

Durch Ctrl und gleichzeitig PrtSc (oder Ctrl+P) wird der Drucker »zugeschaltet«. Alles, was eingegeben oder angezeigt wird, erscheint gleichzeitig am Bildschirm und Drucker. Abgeschaltet werden kann diese Funktion durch eine wiederholte Betätigung der gleichen Tasten. Diese Funktion ist meist

**Texte drucken mit
COPY und PRINT**

während des Ablaufs von Anwendungsprogrammen gesperrt. Mit den Befehlen COPY und PRINT können Sie Texte drucken (siehe Kapitel 11). Einfaches Beispiel zum Ausdruck der Datei CONFIG.SYS:

```
copy config.sys prn
```

copy Kopierbefehl

config.sys Inhalt dieser Datei soll gedruckt (»ausgegeben«) werden

prn symbolischer Name in MS-DOS für den Drucker

**Texte drucken und
gleichzeitig am PC
arbeiten**

Die günstigste Möglichkeit für den Ausdruck von Texten bietet der Befehl PRINT. Wenn mit diesem Befehl dem Computer mitgeteilt wurde, welche Texte gedruckt werden sollen, können gleichzeitig mit dem PC andere Tätigkeiten durchgeführt werden (natürlich ausgenommen Drucken). Dies ist vor allem bei langen Texten hilfreich. Eine komplette Beschreibung zu PRINT finden Sie in Kapitel 11.

```
print autoexec.bat config.sys
```

Die beiden Dateien AUTOEXEC.BAT und CONFIG.SYS hintereinander drucken

Wenn Sie mit PRINT einen weiteren Text drucken wollen und der Drucker noch beschäftigt ist, so merkt das Programm den neuen Text für den Ausdruck vor.

**7.6 Erzeugen von Sicherungskopien
(von Diskette und Festplatte)****Sicherungskopien von
Disketten/Festplatte
erstellen**

Wenn Sie neue Daten (z. B. Buchungen bei einem Finanzbuchhaltungsprogramm) in den Computer eintippen, sollte täglich vor Arbeitsende eine Kopie der Daten erstellt werden. Dies dient der Sicherheit. Sie benötigen eine Kopie, wenn der Datenbestand auf dem Original verlorengeht oder nicht mehr bearbeitbar ist. Dies kommt zwar sehr selten vor, aber wenn, dann benötigen Sie eine Kopie, damit nicht alles wieder neu einge tippt werden muß (unter Umständen kann dies Wochen dauern).

Auch wenn es jahrelang gutgeht – fertigen Sie Sicherungskopien, die Praxis zeigt, daß dies notwendig ist!

7.6.1 Sicherungskopien bei Computern mit 2 Diskettenlaufwerken

Der Vorgang läuft wie in Kapitel 7.3.1 beschrieben ab. Bevor Sie jedoch eine neue Kopie vom Original fertigen, sollten Sie von Ihrer vorherigen Kopie eine Kopie fertigen, so daß Sie immer eine erste und eine zweite Kopie zusätzlich zum Original haben, falls einmal Ihre aktuelle Kopie fehlerhaft sein sollte.

Sicherungskopien bei zwei Laufwerken

Legen Sie Ihre MS-DOS-Diskette in das linke Laufwerk ein (Laufwerk A), und geben Sie den Befehl

```
diskcopy a: b:
```

ein. Das Programm fordert Sie auf, die Disketten einzulegen. Legen Sie jetzt die zu kopierende Diskette in Laufwerk A und die Diskette, auf welche kopiert werden soll, in das rechte Laufwerk B ein, und drücken Sie die Taste <RETURN>. Damit wird der Kopiervorgang gestartet.

7.6.2 Sicherungskopien bei Computern mit einem Diskettenlaufwerk

Sicherungskopien bei einem Laufwerk

Legen Sie Ihre MS-DOS-Diskette in das Laufwerk ein (Laufwerk A), und geben Sie den Befehl

```
diskcopy a: b:
```

ein. Legen Sie die zu kopierende Diskette in Laufwerk A ein. Sie werden abwechselnd aufgefordert, die Originaldiskette bzw. die Zieldiskette einzulegen, bis der Vorgang abgeschlossen ist.

7.6.3 Sicherungskopien von Daten auf einer Festplatte

Sicherungskopien bei Festplatte

Der Sicherungsvorgang erfolgt von einer Festplatte auf Disketten (oder wenn vorhanden auf ein Band bzw. Tape-Streamer). Da die zu sichernden Daten meist die Kapazität einer Diskette übersteigen, stehen hierfür zwei spezielle Befehle zur Verfügung:

BACKUP Sicherungskopie erstellen

RESTORE Zurückkopieren auf Festplatte

**Datensicherung bei
Festplatte mit
BACKUP**

Mit BACKUP gesicherte Dateien dürfen auf keinen Fall mit dem COPY-Befehl zurückkopiert werden, sondern nur mit RESTORE.

Hier will ich nur auf den Standardfall eingehen. Weitere Möglichkeiten finden Sie im Kapitel 11 unter BACKUP bzw. RESTORE.

Gesichert werden die Daten von der Festplatte mit dem Befehl:

```
backup c: a:/s
```

Sichern aller Dateien aus Laufwerk C (aus allen Directories) auf Disketten in Laufwerk A (wenn eine Diskette voll ist, werden Sie automatisch aufgefordert, eine neue einzulegen)

Einzelne Files sichern

Wenn nur bestimmte Dateien gesichert werden sollen, so muß hinter C: noch die entsprechende Dateiangabe erfolgen:

```
backup c:*.txt a:
```

Es werden nur die Dateien mit dem File-Type TXT aus dem aktuellen Inhaltsverzeichnis gesichert.

Die Angabe /S bedeutet, daß alle Directories berücksichtigt werden. Ist diese Angabe nicht vorhanden, so wird nur das aktuelle Directory berücksichtigt (beachten Sie dazu auch das Kapitel 8 »Das MS-DOS-Inhaltsverzeichnis«).

**Zurückholen von Daten
auf Festplatte mit
RESTORE**

Für das Zurückholen der Dateien von den Disketten benötigen Sie den Befehl RESTORE. Für die beiden obigen Beispiele lauten die Befehle:

```
restore a: c: /s  
restore a: c:*.txt
```

BACKUP und RESTORE können für folgende Kopierrichtungen verwendet werden:

Festplatte auf Diskette
Diskette auf Diskette
Diskette auf Festplatte
Festplatte auf Festplatte

Kapitel 8

Das MS-DOS-Inhaltsverzeichnis

System des Directory Seit der Version 2.0 von MS-DOS gibt es hierarchische Inhaltsverzeichnisse auf Diskette/Festplatte. Dies ist ein System mit mehreren Ebenen, das wie ein »Baum« aufgeteilt ist. Eine solche Struktur kann wie folgt aussehen:

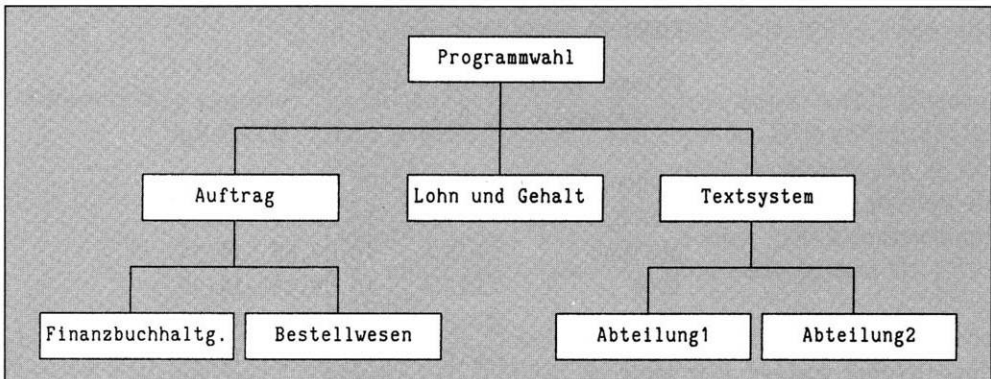


Bild 8.1:
Inhaltsverzeichnis mit
Sub-Directories

Hierarchie und Sub-Directories

Die Spitze des Systems ist die »Wurzel« (Root-Directory). Dies ist das erste Verzeichnis, das automatisch durch das Programm FORMAT (siehe Kapitel 7.3) angelegt wird. Die weiteren Verzeichnisse, welche sich in der Hierarchie darunter befinden, nennt man Sub-Directories. Diese können mit einem speziellen Befehl (MD) angelegt werden.

In der Hierarchie können Sie sich in jeder beliebigen Richtung bewegen.

Aufbau des Directory

Alle Dateien, die mit MS-DOS erstellt werden, erhalten einen Namen. Diese werden zusammen mit weiteren Datei-Informationen in einem Directory (Verzeichnis) gespeichert. Von den Informationen sind die Dateigröße, Datum und Uhrzeit der Dateianlage (oder Änderung) sichtbar. Dafür verwenden

Anzeigen des Directory Sie den Befehl DIR. Er zeigt Ihnen alle Dateien an, die in einem Directory gespeichert sind.

Directory unterteilen in Sub-Directories Mit den Sub-Directories erleichtert Ihnen MS-DOS die Verwaltung von verschiedenen Dateien. Sie haben die Möglichkeit Ihre einzelnen Anwendungen in verschiedenen Sub-Directories anzulegen. Damit erreichen Sie eine erheblich bessere Übersicht. Die Sub-Directories können bei Bedarf so umfangreich wie nötig werden (eine Begrenzung stellt nur die Kapazität des externen Speichers dar).

Root-Directory Neben den Dateinamen erhalten auch die Sub-Directories Namen (nach Ihrer Wahl). Der Name des Root-Directory ist immer \ Die Sub-Directory-Namen werden wie Dateinamen behandelt. Bei der Anzeige des Inhaltsverzeichnisses sind diese am Eintrag in der Spalte mit der Dateigröße erkennbar (Eintrag: <DIR>). Für das vorhergehende Beispiel kann das Root-Directory wie folgt aussehen (Anzeige mit dem Befehl DIR):

Volume in Laufwerk B ist FIRMA

Verzeichnis von B:\

COMMAND	COM	18128	01.01.85	22:50
VERTEIL	BAT	256	15.10.85	01:46
AUFTRAG	<DIR>		10.09.85	14:34
LUGA	<DIR>		10.09.85	14:35
TEXT	<DIR>		10.09.85	14:37
MSDOS	<DIR>		11.09.85	19:05
6 Datei(en)			46126 Bytes frei	

Gesamte Struktur anzeigen mit TREE Wie Sie erkennen können, sind im Inhaltsverzeichnis nur die Sub-Directory-Namen der nächsten Ebene sichtbar. Mit dem Befehl TREE kann die gesamte Struktur angezeigt werden. Nur die Dateien in einem Directory müssen unterschiedliche Namen haben, um sie auch als unterschiedliche Dateien zu erkennen.

Directory wechseln mit CHDIR oder CD So wie es ein aktuelles Laufwerk gibt, gibt es auch ein aktuelles Directory. Zwischen den Directories können Sie mit dem Befehl CHDIR (Kurzform CD – CHange DIRectory) beliebig wechseln.

Mit dem Anlegen eines neuen Directory werden automatisch zwei Dateien in diesem angelegt (nicht löscher). Diese beiden Einträge bezeichnen das neue Directory (Dateiname .) und

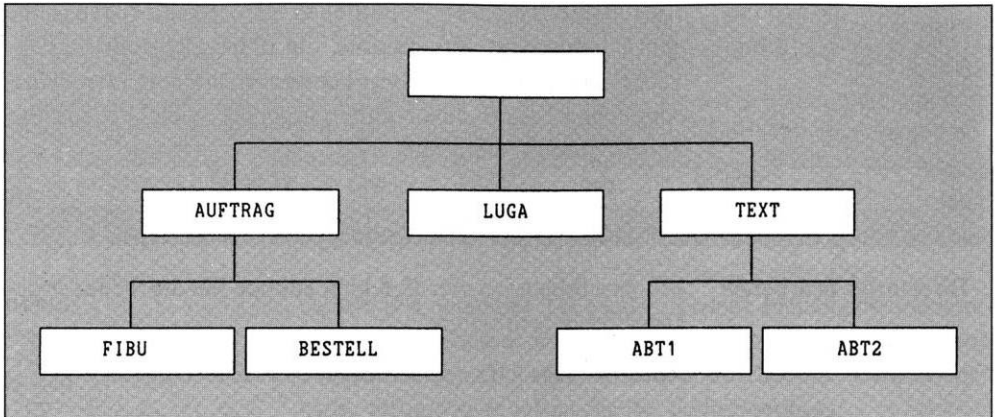


Bild 8.2: Inhaltsverzeichnis mit Sub-Directory-Namen

das hierarchisch davor liegende (Dateiname ..). Bei Angabe eines Punktes wird das aktuelle, bei zwei Punkten das davor liegende Directory angesprochen. Damit speichert MS-DOS, in welcher Beziehung die einzelnen Directories zueinander stehen.

Directory-Weg angeben mit PATH

Benötigen Sie eine Datei, die nicht in dem aktuellen Directory steht, so geben Sie zum Dateinamen den »Weg« (Path) an. Der Weg führt vom aktuellen Directory zu demjenigen, in welchem die gesuchte Datei gespeichert ist. Dazu müssen alle Directory-Namen angegeben werden (als Kennzeichnung steht hinter jedem Directory-Namen das Zeichen \).

Beispiel:

Das aktuelle Directory ist BESTELL. Das gesuchte Programm EDLIN.EXE befindet sich in dem Sub-Directory TEXT. Der Befehl lautet:

Einsatz des back-slash
= \

..\..\text\edlin

oder

\text\edlin

Im ersten Fall geht der Weg rückwärts über AUFTRAG ins Root-Directory und von dort zu TEXT, in dem sich das Programm EDLIN.EXE befindet.

Beginnen Sie den Weg mit \, so beginnt MS-DOS im Root-Directory, unabhängig davon, in welchem Directory Sie sich befinden (zweites Beispiel).

Das Zeichen \ erhalten Sie, indem Sie die Tasten Ctrl + Alt + > gleichzeitig betätigen.

Maximale »Weg«-Länge Die Länge des Weges darf 64 Stellen nicht überschreiten (ohne den Dateinamen). Programme, die unter älteren MS-DOS-Versionen erstellt wurden, können nicht mit den Weg-Angaben arbeiten (davon gibt es noch sehr viele, wie z.B. das Textprogramm WordStar).

8.1 Befehle zur Bearbeitung von Directories

Directories bearbeiten Mit den folgenden vier Befehlen können Sie die Struktur der Directories bearbeiten:

MKDIR oder MD	Anlegen eines Sub-Directory
CHDIR oder CD	Anzeigen oder Ändern des aktuellen Directory
RMDIR oder RD	Löschen eines Sub-Directory-Eintrags
PATH	Bezeichnet den Weg durch die Verzeichnisse bis zu einem Programm (File-Typ COM, EXE oder BAT).
APPEND	Einrichten eines Suchpfads für Datendateien

Außer diesen speziellen Befehlen zur Bearbeitung der Struktur, sind folgende von wesentlicher Bedeutung:

COPY	Kopieren von Dateien von einem Directory zu einem anderen
XCOPY	Kopieren von Dateien und Directories
DEL	Löschen von Dateien (in einem Directory)
DIR	Anzeigen des Inhaltsverzeichnisses eines Directory

Die internen Befehle des MS-DOS sind jederzeit abrufbereit, da sie sich bereits im Speicher des Computers befinden. Externe sind Programme, die beim Aufruf erst in den Speicher von der Diskette oder Festplatte geladen werden müssen. Es bietet sich an, alle Programme des MS-DOS in einem separaten Sub-Directory zu speichern (MSDOS):

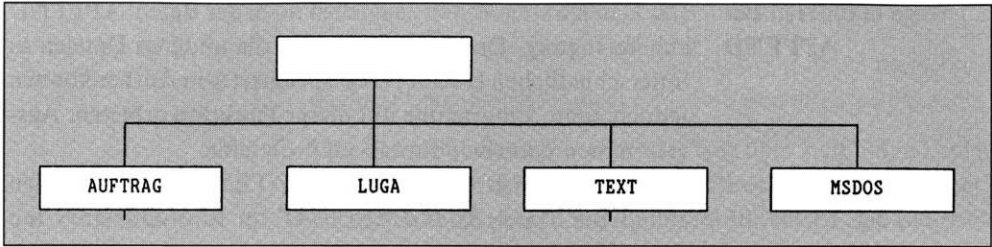


Bild 8.3:
*Directory-Struktur mit
 neuem Sub-Directory für
 MS-DOS*

Geben Sie, während Sie sich in jedem beliebigen anderen Directory befinden, einen externen Befehl ein, so wird das aktuelle Directory nach dem Befehl durchsucht. Wird dieser nicht gefunden, erhalten Sie die Fehlermeldung:

Befehl oder Dateiname falsch

Vor Befehlsnamen Weg angeben

Sie müssen vor dem Befehlsnamen den Weg angeben (\MSDOS\befehl). Sie können es jedoch vermeiden, bei jedem Aufruf eines externen Befehls den Weg anzugeben. Dazu verwenden Sie nach dem Start des Computers den Befehl:

```
path=\msdos
```

Dieser Weg muß nur einmal von Ihnen eingegeben werden. Findet MS-DOS einen eingegebenen Befehl im aktuellen Directory nicht, so sieht es nach, ob mit PATH ein Weg definiert wurde, und verwendet diesen. In dieser Form können auch weitere Wege vorgegeben werden:

Mehrere Wege vorgeben

```
path=\msdos;\text\abt1;\text\abt2;\auftrag\bestell
```

Durch Eingabe von

```
path
```

Gültige Wege anzeigen lassen

können Sie sich den oder die derzeit gültigen Wege anzeigen lassen. Bitte beachten Sie, daß mit jeder Neueingabe von Wegen mit PATH die bisher gültigen gelöscht werden (also bei jeder Änderung immer alles Notwendige angeben).

Die Vereinbarung mit PATH gilt nur für Programme bzw. Befehle (File-Typen COM, EXE und BAT). Alle anderen Dateien müssen sich im aktuellen Verzeichnis befinden (oder Sie geben den Weg an). Dies gilt vor allem für Programme, die in mehrere Teile gesplittet sind.

**Wege definieren mit
APPEND**

Für Dateien steht Ihnen zusätzlich noch der Befehl APPEND zur Verfügung. Damit können auch die anderen Dateien in unterschiedlichen Directories gespeichert sein. Bisher können jedoch kaum Programme mit dieser Funktion arbeiten. Ausgenommen natürlich die MS-DOS-Befehle. Befinden Sie sich im Sub-Directory TEXT und wollen den Inhalt der Datei AUTOEXEC.BAT im Root-Directory am Bildschirm anzeigen:

```
type autoexec.bat
```

Da sich AUTOEXEC.BAT nicht im aktuellen Directory befindet, erhalten Sie die Fehlermeldung:

```
Datei nicht gefunden
```

Hier kann, wie bereits beschrieben, PATH nicht angewendet werden. Sie definieren die Wege für Dateien mit dem Befehl APPEND. Für unser vorheriges Beispiel gilt:

```
append \
```

Durch diesen Befehl wird die Datei AUTOEXEC.BAT erst im aktuellen Directory gesucht und anschließend im Root-Directory (\). Wenn Sie den TYPE-Befehl noch einmal eingeben, erhalten Sie diesmal die Anzeige des Inhalts von AUTOEXEC.BAT.

8.2 Anlegen eines Sub-Directory (MKDIR oder MD)

**Sub-Directory anlegen
mit MKDIR oder MD**

Um ein neues Sub-Directory anzulegen, verwenden Sie den Befehl MKDIR oder die Kurzform MD:

```
md tabelle
```

Das neue Sub-Directory tabelle wird »unterhalb« des aktuellen angelegt. Dabei werden automatisch die beiden Einträge . und .. im neuen Sub-Directory erstellt:

**Inhaltsverzeichnis
eines Sub-Directory**

Volume in Laufwerk A ist MS-DOS 3-2
Verzeichnis von A:\TABELLE

.	<DIR>	12.09.85	23:56
..	<DIR>	12.09.85	23:56
2 Datei(en)		23450 Bytes frei	

Ein neues Sub-Directory kann an jeder beliebigen Stelle unter Angabe des Weges erstellt werden:

```
md \auftrag\bestell\abt1
```

Vom Root-Directory aus führt der Weg über AUFTRAG nach BESTELL. Dort wird das neue Sub-Directory ABT1 angelegt. Die Sub-Directories AUFTRAG und BESTELL müssen vorher schon vorhanden sein.

**Datei- oder Directory-
Namen eindeutig
vergeben!**

Ein Datei- oder Directory-Name darf in einem Directory nur einmal vorkommen, kann jedoch in der gesamten Struktur öfter erscheinen (siehe im vorliegenden Beispiel ABT1, der als Sub-Directory-Name in BESTELL und TEXT vorkommt). Wegen einer besseren Übersicht sollten Sie aber darauf verzichten.

Kann ein neues Sub-Directory nicht angelegt werden, weil der Name schon vorhanden ist, so erhalten Sie die Fehlermeldung

Verzeichnis kann nicht erstellt werden

8.3 Anzeigen und Ändern des aktuellen Directory (CHDIR oder CD)

**Aktuelles Directory
wechseln oder
anzeigen mit CHDIR
oder CD**

Jedes Directory hat einen Namen. Sie können den Namen des aktuellen Directory durch den Befehl

```
cd
```

anzeigen lassen.

Das aktuelle Directory für ein bestimmtes Laufwerk erhalten Sie mit

```
cd b:
```

Sie geben also zusätzlich noch den Laufwerknamen an.

Beispiele für die Anzeige:

```
A:\AUFTRAG\FIBU
```

FIBU ist das aktuelle Directory in Laufwerk A

```
B:\
```

das Root-Directory ist in Laufwerk B aktuell

```
A:\TABELLE
```

TABELLE ist das aktuelle Directory in Laufwerk A

Aktuelles Directory wechseln

Das aktuelle Directory kann sehr leicht geändert werden. Sie geben zusätzlich zum Befehl CD noch den Weg an. Beispiele:

```
cd \
```

Rückkehr zum Root-Directory – von jedem beliebigen Sub-Directory aus

```
cd text
```

neues aktuelles Directory ist TEXT (\ darf nicht angegeben werden). TEXT muß dabei ein Directory in der nächsten Ebene unter dem aktuellen sein

```
cd \auftrag\fibu
```

\ Weg vom Root-Directory aus beginnen
AUFTRAG über AUFTRAG den Weg fortsetzen zu
FIBU (wird neues aktuelles Directory)

```
cd ..
```

vom derzeitigen Directory zum vorherigen wechseln (neues aktuelles ist damit im vorliegenden Beispiel AUFTRAG)

Ungültiges Directory

Wenn Sie den Weg nicht richtig eingeben bzw. der angegebene Directory-Name von MS-DOS nicht gefunden wird, erhalten Sie die Fehlermeldung

Ungültiges Verzeichnis

8.4 Löschen eines Sub-Directory (RMDIR oder RD)

Sub-Directory löschen mit RMDIR oder RD

Das Root-Directory kann nicht gelöscht werden. Für das Löschen von Sub-Directory-Einträgen muß beachtet werden:

1. Im Sub-Directory müssen vorher alle Dateien gelöscht werden (Befehl DEL)
2. Es dürfen sich keine weiteren Directory-Einträge im zu löschenden Sub-Directory befinden
3. Nur die Einträge . und .. bleiben erhalten bzw. werden durch den Befehl RD gelöscht
4. Das aktuelle Directory kann nicht gelöscht werden

Beispiele:

rd luga

Löschen des Sub-Directory LUGA

rd \auftrag\bestell

\ Weg vom Root-Directory aus beginnen

AUFTRAG über AUFTRAG den Weg fortsetzen zu

BESTELL (wird gelöscht)

Fehler beim Löschen von Sub-Directories

Sie erhalten die Fehlermeldung

Ungültiger Pfad, kein Verzeichnis,
oder Verzeichnis nicht leer

wenn einer der folgenden Fehler festgestellt wurde:

1. Sie versuchen eine Datei und nicht ein Sub-Directory mit RD zu löschen
2. Das zu löschende Sub-Directory ist nicht »leer«
3. Sie versuchen, das aktuelle Directory zu löschen
4. Ein angegebener Directory-Name ist falsch

8.5 Anzeigen des Inhaltsverzeichnisses eines Directory (DIR)

Inhaltsverzeichnis in Sub-Directories anzeigen mit DIR

Den Befehl DIR haben Sie bereits im Kapitel 7 kennengelernt. Mit diesem Befehl können Sie sich das Inhaltsverzeichnis jedes Directory anzeigen lassen, unabhängig von Ihrer aktuellen Position.

Beispiele (*. * muß nicht angegeben werden):

dir *. *

oder

dir \

Anzeigen des gesamten Inhaltsverzeichnisses des Root-Directory


```
dir \auftrag\*.*
```

Weg über Root-Directory zu AUFTRAG und dessen gesamtes Inhaltsverzeichnis anzeigen lassen

```
dir \text\abt1\*.txt /p
```

Weg über Root-Directory und TEXT zum Sub-Directory ABT1 und alle Dateien mit dem File-Typ TXT anzeigen lassen (mit /p werden die Dateien seitenweise angezeigt)

```
dir ..
```

Anzeigen des Inhaltsverzeichnisses aus dem hierarchisch vorher liegenden Directory

```
dir b:\sub\*.com
```

im Laufwerk B den Weg über das Root-Directory zu SUB und aus diesem Sub-Directory alle Dateien mit dem File-Typ COM anzeigen

```
dir .
```

oder

```
dir
```

oder

```
dir *.*
```

Anzeigen des Inhaltsverzeichnisses des aktuellen Directory

8.6 Löschen von Dateien in einem Directory (DEL oder ERASE)

Dateien in Sub-Directories löschen mit DEL oder ERASE

Mit den Befehlen DEL oder ERASE können Sie Dateien in jedem vorhandenen Directory löschen (unabhängig vom aktuellen Directory). Bevor Sie ein Sub-Directory mit RD löschen können, müssen Sie mit DEL alle Dateien daraus löschen.

Beispiele:

```
del \auftrag\*.*
```

oder

```
del \auftrag
```

Weg über Root-Directory zu AUFTRAG und alle Dateien daraus löschen (nicht jedoch eventuell darin enthaltene Sub-Directories)

```
del \text\abt1\*.txt
```

Weg über Root-Directory und TEXT zum Sub-Directory ABT1 und alle Dateien mit dem File-Typ TXT löschen

```
del ..
```

Löschen aller Dateien aus dem hierarchisch vorher liegenden Directory

```
del .
```

oder

```
del
```

oder

```
del *.*
```

Löschen aller Dateien aus dem aktuellen Directory

Sicherheitsabfrage bei DEL *.*

Wenn Sie mit einem Befehl (DEL *.*) alle Dateien aus einem Directory löschen, so fragt MS-DOS

Sind Sie sicher (J/N)?

Wenn Sie die Tasten j und <RETURN> betätigen, so wird der Befehl ausgeführt.

8.7 Kopieren von Dateien zwischen Directories (COPY und XCOPY)

Dateien zwischen Sub- Directories kopieren mit COPY und XCOPY

Den Befehl COPY haben Sie bereits im Kapitel 7 kennengelernt. Mit ihm können Sie auch Dateien von einem zu einem beliebigen anderen Directory kopieren. Zu diesem Zweck geben Sie bei »woher« und »wohin« vor dem Dateinamen den Weg an.

Beispiele für COPY Beispiele:

```
copy *.txt \text\abt2
```

alle Dateien aus dem aktuellen Directory mit dem File-Typ TXT in das Sub-Directory ABT2 kopieren

```
copy \msdos\command.com
```

COMMAND.COM aus dem Sub-Directory MSDOS in das aktuelle Directory mit dem gleichen Namen kopieren

```
copy \auftrag\fibu\op.dat \text\abt1\offen
```

die Datei OP.DAT aus dem Sub-Directory FIBU nach ABT1 kopieren und gleichzeitig den Dateinamen auf OFFEN ändern

```
copy a:\msdos\*.com b:
```

aus dem Sub-Directory MSDOS auf der Diskette in Laufwerk A alle Dateien mit dem File-Typ COM ins aktuelle Directory in Laufwerk B kopieren

Möglichkeiten des erweiterten Kopierbefehls XCOPY

Bei dem Befehl XCOPY handelt es sich um einen erweiterten Kopierbefehl. Mit ihm können Sie nicht nur Dateien von einem Directory in ein anderes kopieren. Folgende Möglichkeiten stehen Ihnen zur Verfügung:

1. Kopieren von Dateien innerhalb eines Directory
2. Kopieren von Dateien von einem Laufwerk auf ein anderes
3. Kopieren von Dateien von einem Directory in ein anderes
4. Kopieren von Dateien eines Directory und allen darunter befindlichen auf ein anderes Laufwerk oder in ein anderes Directory (einschließlich der Sub-Directory-Namens)

Beispiele für XCOPY Beispiele:

```
xcopy autoexec.bat autoexec.alt
```

Von Datei AUTOEXEC.BAT im aktuellen Directory/Laufwerk wird eine Kopie mit dem neuen Namen AUTOEXEC.ALT erstellt.

```
xcopy *.bat c:
```

Alle Dateien mit dem File-Typ BAT werden vom aktuellen Directory/Laufwerk in das aktuelle Directory des Laufwerks C kopiert.

```
xcopy \*.bat \text\abt2
```

Alle Dateien mit dem File-Typ BAT werden vom Root-Directory in das Sub-Directory ABT2 (von TEXT) mit den gleichen Namen kopiert.

```
xcopy \text\*.* a:\sicher\s/e/v/d:01-09-86
```

Alle Dateien des Sub-Directory TEXT des aktuellen Laufwerks werden nach Laufwerk A in das Sub-Directory SICHER kopiert. Wenn SICHER auf der Diskette in Laufwerk A noch nicht existiert, wird dieses automatisch angelegt. Wenn eine Datei mit dem Namen SICHER vorhanden ist, wird XCOPY mit einer Fehlermeldung abgebrochen.

Einstellungsmöglichkeiten von XCOPY

- /S Es werden die Dateien des Sub-Directory TEXT sowie alle darunter enthaltenen kopiert (also auch ABT1 und ABT2 mit allen enthaltenen Dateien)
- /E Kopiert ein Sub-Directory auch, wenn darin keine Datei enthalten und es damit leer ist (nur in Verbindung mit /S möglich)
- /V Wenn eine Datei kopiert wurde, wird sofort geprüft, ob die Kopie mit dem Original identisch ist oder während des Kopiervorgangs Daten falsch übertragen wurden
- /D Es werden nur Dateien kopiert, die als Datum den 1.9.86 oder früher besitzen

Einsatz von XCOPY bei einem Laufwerk

Besitzen Sie einen Schneider PC mit nur einem Laufwerk, und Sie kopieren von Laufwerk A nach B, so müssen Sie auch noch /W angeben. Sie werden jeweils aufgefordert, die Disketten zu wechseln:

Beliebige Taste drücken, um Dateien zu kopieren

8.8 Zusammenfassendes Beispiel

`cd \`

das Root-Directory wird zum aktuellen Directory

`md oa`

Anlegen des Sub-Directory OA

`cd oa`

OA wird neues aktuelles Directory

`copy con text.txt`

Erstellen der Textdatei TEXT.TXT (Kopieren vom Bildschirm/Tastatur – CON – in die Datei TEXT.TXT)

Dies ist die Datei TEXT.TXT im Sub-Directory OA

Text zur Datei TEXT.TXT eingeben

Tasten F6 und <RETURN> betätigen

einggegebenen Text speichern (TEXT.TXT)

`dir`

Anzeigen des Inhaltsverzeichnisses des aktuellen Directory OA

`cd ..`

Wechseln zum vorhergehenden Directory (Root-Directory)

`dir /p`

Anzeigen des Inhaltsverzeichnisses des aktuellen Directory (Root-Directory) »seitenweise«

`rd oa`

Löschen des Sub-Directory OA

Sie erhalten eine Fehlermeldung, da sich in OA noch die Datei TEXT.TXT befindet:

Ungültiger Pfad, kein Verzeichnis,
oder Verzeichnis nicht leer

```
del oa\*.*
```

alle Dateien im Sub-Directory OA löschen (Sie werden nach dem Befehl von MS-DOS noch befragt, ob Sie sicher sind, alles löschen zu wollen)

```
rd oa
```

Löschen des Sub-Directory OA

Kapitel 9 Datenumleitung und Filter

Standardein- und -ausgabegerät beeinflussen

MS-DOS-Befehle lesen eingegebene Daten vom Standard-eingabegerät und geben Daten zum Standardausgabegerät weiter. Den sogenannten Treiber dazu nennt man CON (Konsole). Er liest Daten von der Tastatur und gibt Daten zum Bildschirm. Diese Funktion ist im MS-DOS vorgegeben. Die Standardfunktion kann auf zwei verschiedene Arten beeinflusst werden:

1. Datenumleitung
2. Befehlsverkettung (mit Filtern)

Datenumleitung auf anderes Eingabegerät

Bei der Datenumleitung kann ein anderes Eingabegerät als die Tastatur bestimmt werden (z. B. eine Datei, ein anderer Computer – kein Drucker). Ebenso kann ein anderes Ausgabegerät als der Bildschirm bestimmt werden (z. B. der Drucker, eine Datei – nicht die Tastatur).

Bei der Verkettung von Befehlen werden die auszugebenden Daten eines Befehls automatisch als Eingabe zu einem anderen Befehl verwendet.

9.1 Datenumleitung

Zeichen für die Datenumleitung

Für die Datenumleitung gibt es besondere Zeichen als Anweisung an das MS-DOS:

- < (woher) Das Zeichen < bestimmt, daß die Daten von »woher« (Datei oder Gerät) entgegengenommen werden
- > (wohin) Das Zeichen > bestimmt, daß die Daten nach »wohin« (Datei oder Gerät) gesandt werden

> > (wohin) Die Zeichen > > bestimmen, daß die Daten an das Ende von »wohin« angefügt werden (normalerweise Datei)

Beispiele für die Datenumleitung:

```
sort < unsort.txt
```

Der Befehl SORT sortiert Daten zeilenweise in auf- oder absteigender Form. Die Eingabe zu SORT erfolgt hier von der Datei UNSORT.TXT. Die in der Datei enthaltenen Daten werden aufsteigend sortiert am Bildschirm ausgegeben. Ohne < unsort.txt müßte die Eingabe von der Tastatur aus erfolgen; nach Betätigung der Taste F6 würde die Ausgabe sortiert am Bildschirm erfolgen.

```
sort < unsort.txt > sort.txt
```

**Beispiele für die
Datenumleitung**

Die Eingabe zu SORT kommt von der Datei UNSORT.TXT. Die in der Datei enthaltenen Daten werden aufsteigend sortiert in der neuen Datei SORT.TXT auf der Diskette/Festplatte im aktuellen Directory/Laufwerk gespeichert.

```
sort < com1
```

Daten, die über ein Modem kommen, werden sortiert am Bildschirm angezeigt (COM1 ist der Gerätenamen für die erste serielle Schnittstelle – siehe Kapitel 10).

```
sort < unsort.asc > prn
```

Daten, die in der Datei UNSORT.ASC gespeichert sind, werden aufsteigend sortiert ausgedruckt.

```
sort < \text\abt1\buch.txt > \auftrag\fibu\buch.txt
```

Die Eingabe zu SORT kommt von der Datei BUCH.TXT aus dem Sub-Directory ABT1. Die Daten werden sortiert und in das Sub-Directory FIBU mit dem Dateinamen BUCH.TXT gespeichert (der Dateiname darf hier für Ein- und Ausgabe gleich sein, da zwei verschiedene Directories verwendet werden).

```
dir > prn
```

Mit DIR wird das Inhaltsverzeichnis vom aktuellen Laufwerk/Directory angezeigt. Hier erfolgt die Ausgabe nicht am

Bildschirm, sondern wird auf den Drucker (PRN – siehe Kapitel 10) umgeleitet.

```
dir b: > a:inhalt.dir
```

Das Inhaltsverzeichnis des aktuellen Directory auf der Diskette in Laufwerk B wird auf die Diskette in Laufwerk A mit dem Namen INHALT.DIR gespeichert.

```
dir a: >> a:inhalt.dir
```

Das Inhaltsverzeichnis des aktuellen Directory auf der Diskette in Laufwerk A wird auf die Diskette in Laufwerk A an das Ende der Datei INHALT.DIR gespeichert.

9.2 Befehlsverkettung und Filter

Einsatz von Filtern und Befehlsverkettung

Filter sind externe Befehle, die Daten vom Standardeingabegerät lesen, diese verarbeiten und an das Standardausgabegerät weitergeben. Die Filter des MS-DOS können alleine oder in Verbindung mit anderen Befehlen verwendet werden. Folgende Filter gibt es:

- FIND** sucht nach von Ihnen vorgegebenen Zeichen im Standardeingabegerät und gibt die gefundenen Daten am Standardausgabegerät aus
- MORE** gibt Daten, die vom Standardeingabegerät kommen, seitenweise am Standardausgabegerät aus
- SORT** sortiert auf- oder absteigend jene Daten, die vom Standardeingabegerät kommen, und gibt diese am Standardausgabegerät aus

Den Filter SORT haben Sie bereits in Verbindung mit der Datenumleitung kennengelernt (Kapitel 9.1). Beachten Sie, daß Filter nur in Verbindung mit Programmen verwendet werden können, die die Standardein- und -ausgabe des MS-DOS benutzen. Der erste Befehl in einer Befehlsverkettung muß jedoch nur die Standardausgabe verwenden.

Bei einer Befehlsverkettung wird die Standardausgabe eines Befehls zur Standardeingabe des zweiten Befehls. Die Befehle werden mit dem Zeichen `|` verkettet. Die Begrenzung der Eingabezeile liegt lediglich bei der maximalen Länge einer Befehlszeile im MS-DOS (127 Zeichen).

Abarbeitung einer Befehlskette

Während der Abarbeitung einer Befehlskette legt MS-DOS auf der Diskette/Festplatte Dateien an, in der die Ein- und Ausgaben der Befehle zwischengespeichert werden. Für diese Dateien wird die aktuelle Uhrzeit (inklusive der hundertstel Sekunden) verwendet. Die Dateien haben die Bezeichnung:

hhmmsscc (z.B.: 10063455)

Dies ist wegen der Möglichkeit der Nutzung von PCs in einem Netzwerk notwendig, da es vorkommen kann, daß mehrere Benutzer »gleichzeitig« diese sogenannten PIPEs verwenden und damit eine Unterscheidung möglich ist. Diese Dateien werden bei Abschluß der Befehlsverkettung automatisch wieder gelöscht.

Vorzeitiges Abbrechen einer Befehlskette

Sollten Sie die Abarbeitung einer Befehlsverkettung mit Ctrl+Break vor dem Ende abbrechen, so sind die zwischengespeicherten Dateien noch auf der Diskette/Festplatte vorhanden (Sie können diese mit dem Befehl DIR anzeigen lassen).

Sollte der Platz auf einer Diskette oder Festplatte für die Zwischenspeicherung der Daten bei der Abarbeitung einer Befehlsverkettung nicht ausreichen, erhalten Sie eine Fehlermeldung am Bildschirm:

Dateifehler während Datenübergabe (PIPE)

Das Zeichen ! erhalten Sie durch die Eingabe von Alt+1+2+4, wobei die Ziffern 1, 2 und 4 mit dem Tastenfeld rechts an der Tastatur einzugeben sind, ohne die Taste NumLock zu betätigen.

Beispiele für die Befehlsverkettung:

```
dir | sort > prn
```

Das Inhaltsverzeichnis des aktuellen Directory im aktuellen Laufwerk dient als Eingabe für das Sortierprogramm. Nachdem SORT das Inhaltsverzeichnis aufsteigend sortiert hat, erfolgt die Ausgabe auf den Drucker (PRN – siehe Kapitel 10). Beispiel für die sortierte Ausgabe:

```
          7 Datei(en)          3072 Bytes frei
Verzeichnis von A:\IMAGES
Volume in Laufwerk A ist 46001G
```

A		0	4.01.80	0.00
PC1512	GEM	96	1.01.80	2.20
PC1512	IMG	25948	1.01.80	2.20
TIGER	GEM	94	1.01.80	3.28
TIGER	IMG	6671	1.01.80	3.28

Beispiele für Befehlsverkettung

```
dir | sort | more
```

Das Inhaltsverzeichnis des aktuellen Directory im aktuellen Laufwerk dient als Eingabe für das Sortierprogramm SORT. Das sortierte Inhaltsverzeichnis dient als Eingabe für den Befehl MORE, der die Anzeige am Bildschirm »seitenweise« vornimmt.

```
dir | find "DAT" | sort | more
```

Es werden nur Dateien angezeigt, die den File-Typ DAT besitzen. Die gefundenen Dateien werden aufsteigend sortiert und seitenweise am Bildschirm angezeigt.

```
type \text\abt2\text1.txt | more
```

Der Text TEXT1.TXT im Sub-Directory ABT2 wird am Bildschirm seitenweise (MORE) angezeigt (TYPE).

Kapitel 10

Gerätenamen

- Gerätetreiber** Zur ordnungsgemäßen Abarbeitung der Ein- und Ausgaben zu den Peripheriegeräten (Tastatur, Bildschirm, Laufwerke, Drucker, Modems usw.) benötigt MS-DOS Gerätetreiber. Diese beinhalten alle Informationen, die das Betriebssystem Ihres Schneider PC über ein Peripheriegerät wissen muß.
- Blocktreiber** Für den Datentransport zu und von den Laufwerken (Diskette/Festplatte) werden sogenannte Blocktreiber benötigt. Diese verwenden die Befehle/Programme unter MS-DOS.
- Charaktertreiber** Für den Transport der Daten in serieller Form (für Tastatur, Bildschirm, Drucker usw.) werden die Charaktertreiber benötigt. Diese werden durch den Namen des Peripheriegerätes benannt. Diese Gerätenamen sind für MS-DOS reserviert und dürfen nicht zur Bezeichnung von Dateien oder Sub-Directories verwendet werden.

Treibername	Peripheriegerät bzw. Schnittstelle
AUX	asynchrone Schnittstelle (oder COM1)
CLOCK\$	Treiber für Echtzeituhr
COM1	erste serielle Schnittstelle (V.24)
COM2	zweite serielle Schnittstelle (V.24)
CON	Console (Tastatur/Bildschirm)
LPT1	erste parallele Schnittstelle (normalerweise Drucker)
LPT2	zweite parallele Schnittstelle
LPT3	dritte parallele Schnittstelle
NUL	nicht vorhandener Ausgang
PRN	Drucker (oder LPT1)

Tabelle 10.1: Gerätetreiber

Die Namen der Gerätetreiber können Sie in Befehlen verwenden.

Beispiele:

**Einbindung der
Gerätetreiber-Namen
in Befehle**

`copy con datei.dat`

Von dem Gerät CON (also hier die Eingabeeinheit Tastatur) werden die Daten in die Datei DATEI.DAT gespeichert. Alle Daten, die Sie eintippen, werden gleichzeitig am Bildschirm dargestellt und werden in DATEI.DAT gespeichert, wenn Sie die Tasten F6 und <RETURN> betätigen.

`copy datei.dat prn`

Die Daten der Datei DATEI.DAT werden auf das mit PRN bezeichnete Peripheriegerät (Drucker) übertragen.

`copy com1 modem.dat`

Alle Daten, die von der ersten seriellen Schnittstelle (COM1) kommen, werden in der Datei MODEM.DAT gespeichert (z.B. Daten von einem Akustikkoppler, der an COM1 angeschlossen ist). Statt COM1 kann auch der Gerätenamen AUX verwendet werden.

Kapitel 11

Die MS-DOS-Befehle mit Übungen

Beschreibung aller MS-DOS-Befehle

Auf den folgenden Seiten werden sämtliche MS-DOS-Befehle Ihres Schneider PC vorgestellt und beschrieben. Anhand vieler Beispiele werden Ihnen diese näher erklärt. Wiederholen Sie die beschriebenen Beispiele an Ihrem eigenen Schneider PC, um durch die Übung Routine zu erhalten und die Bedienung des MS-DOS zu erlernen. Dieses Kapitel dient auch als umfangreiches Nachschlagewerk. Zu diesem Zweck sind die Befehle alphabetisch geordnet.

Darstellungsform beim Kommandoaufbau

Beachten Sie bitte zur Beschreibung der Befehle die folgende Darstellungsform beim Kommandoaufbau:

d:	Laufwerkbezeichnung
p	Teil eines Festplattenlaufwerkes
s:	Ursprungslaufwerk
u:	Nummer des Laufwerkes
/x	Zusatzeingabe (Parameter)
volumelabel	Name einer Diskette bzw. eines Festplattenlaufwerkes
mm	Monat oder Minute
dd	Tag
jj	Jahr (zwischen 1980 und 1999)
jjjj	Jahr (zwischen 1980 und 2099)
hh	Stunde
ss	Sekunde
cc	Hundertstelsekunde
n oder #	Zahl ohne Kommastellen
dev	Gerätename
cttydev	Gerät, das Ein- und Ausgaben erlaubt
()	(Variable/Zusatzeingabe) muß nicht eingegeben werden

... Wiederholungsdarstellung für Variable/
Zusatzeingabe

Einteilung der Laufwerke Zur Einteilung der Laufwerke beachten Sie bitte das Kapitel 7.
Für die in diesem Kapitel angegebenen Beispiele werden folgende Laufwerkbezeichnungen verwendet:

A: erstes bzw. linkes Laufwerk
B: zweites bzw. rechtes Laufwerk
C: erstes Festplattenlaufwerk
G: RAM-DISK (wenn eingerichtet – diese kann auch eine andere Laufwerkbezeichnung haben, da immer der nächste bei Ihrem System freie Laufwerkname verwendet wird).

APPEND *externer Befehl*

Zweck Wenn Sie einen Dateinamen verwenden, ohne das Directory anzugeben, in dem er sich befindet, so durchsucht MS-DOS nur das Directory des aktuellen Laufwerks. Wird die Datei nicht gefunden, erhalten Sie eine Fehlermeldung.

Mit APPEND werden die Wege zu verschiedenen Directories definiert. Findet MS-DOS eine Datei nicht im aktuellen Directory, so werden anschließend noch alle durchsucht, die mit APPEND definiert wurden.

Die Definitionen von APPEND werden von allen MS-DOS-Befehlen benutzt. Da dieser Befehl in MS-DOS 3.2 neu ist, gilt dies nicht für die derzeit am Markt erhältlichen Anwenderprogramme (Textverarbeitung, Finanzbuchhaltungen usw.).

Kommandoaufbau (d:)(weg)append (d:)\weg;(d:)\weg)(...)
 (d:)(weg)append
 (d:)(weg)append ;

Beispiele append

Da Sie mit APPEND noch keine »Wege« definiert haben, erhalten Sie die Fehlermeldung:

```
Keine weiteren Inhaltsverzeichnisse
append \
```

Wenn Sie eine Datei verwenden, die im Root-Directory gespeichert ist und Sie sich in einem Sub-Directory befinden, so finden nachfolgende Befehle diese Datei.

```
type autoexec.bat
```

Die Datei AUTOEXEC.BAT befindet sich im Root-Directory. Diese wird mit dem Befehl TYPE angezeigt, auch wenn das aktuelle Directory ein Sub-Directory ist, da vorher mit APPEND der Weg zum Root-Directory vereinbart wurde.

```
append
```

Die aktuell mit APPEND vereinbarten Wege werden Ihnen angezeigt:

```
APPEND=\
append \;a:\text\abt1;a:\text\abt2
```

Bei der erneuten Eingabe des APPEND-Befehls werden alle bisherigen Angaben gelöscht und dafür die neuen gespeichert. Bei der Angabe einer Datei wird erst das aktuelle Directory durchsucht und anschließend das Root-Directory, dann das Sub-Directory ABT1 und ABT2, wenn diese nicht gefunden wird.

```
append
```

```
APPEND=\;A:\TEXT\ABT1;A:\TEXT\ABT2
```

Mit APPEND ohne weitere Angaben wird der derzeitige Stand der Vereinbarungen angezeigt.

```
append ;
```

Mit vorstehendem Befehl werden alle Vereinbarungen wieder gelöscht. Sie können dies sehr leicht nachprüfen, indem Sie nur

```
append
```

eingeben. Sie erhalten dann die Anzeige

Keine weiteren Inhaltsverzeichnisse
am Bildschirm.

ASSIGN *externer Befehl*

Zweck Mit ASSIGN kann einem Laufwerk eine andere logische Laufwerkbezeichnung zugeordnet werden (Laufwerkumleitung). Dadurch können Sie andere Laufwerke benutzen als jene, auf denen Ihre Programme eingerichtet sind. Bei Benutzung folgender Befehle niemals verwenden: DISKCOMP, DISKCOPY, FORMAT, PRINT. Ferner darf ASSIGN nicht für ein Netzwerk-Laufwerk verwendet werden.

Kommandoaufbau (d:)(weg)assign
 (d:)(weg)assign (x=y (...))
 x angegebenes Laufwerk
 y gesuchtes Laufwerk

Beispiele assign a=c

Wenn anschließend das Laufwerk a: verwendet wird, benutzt das System das Festplattenlaufwerk c:.

assign a=c b=c

Wenn nach Ausführung des Befehls das Laufwerk a: oder b: verwendet wird, benutzt das System das Laufwerk c:.

assign a=a

Damit wird die Vereinbarung, daß bei Angabe des Laufwerks A das Festplattenlaufwerk C verwendet werden soll, wieder rückgängig gemacht. Bei Angabe der Laufwerkbezeichnung A wird wieder das physikalisch erste Laufwerk (A) verwendet.

assign

Laufwerkumbenennungen wieder löschen bzw. den Ursprungszustand wieder herstellen.

Bemerkungen Leider gibt es Anwendungssoftware, die standardmäßig die Laufwerke A und B benutzt, ohne daß dies von Ihnen geändert werden könnte. Wenn Sie einen Schneider PC mit einer Festplatte besitzen, gibt es kein zweites Laufwerk – und normalerweise wurden auch die Programme auf die Platte kopiert. In diesem Falle kann vor dem Start eines Anwenderprogrammes mit ASSIGN die Laufwerkzuweisung geändert werden (siehe zweites Beispiel).

ATTRIB *externer Befehl*

Zweck Eine Datei vor dem Löschen oder Verändern schützen (auf Read-Only = »Nur-Lesen« setzen). Mit dem Programm kann der Vorgang auch wieder zurückgesetzt werden (auf Read-Write setzen).

Ferner kann mit dem Befehl das sogenannte Archivbyte geändert werden. Dieses wird von den Befehlen XCOPY, BACKUP und RESTORE benutzt. Zu diesen Befehlen gibt es einen Parameter, mit dem Sie entscheiden können, ob eine Datei mit gesetztem Archivbyte kopiert werden soll. Wenn Sie eine Datei ändern, setzt MS-DOS automatisch das Archivbyte. Wenn eine Datei mit gesetztem Archivbyte gesichert bzw. mit XCOPY oder BACKUP kopiert wird, wird es wieder gelöscht bzw. zurückgesetzt. Mit ATTRIB können Sie dieses verändern.

Kommandoaufbau

```
(d:)(weg)attrib dateibez
(d:)(weg)attrib +r dateibez
(d:)(weg)attrib -r dateibez
(d:)(weg)attrib +a dateibez
(d:)(weg)attrib -a dateibez
+r      auf Read-Only setzen
-r      auf Read-Write setzen
+a      Archivbyte setzen
-a      Archivbyte löschen
```

Beispiele attrib +r *.txt

Alle Dateien mit dem File-Typ TXT werden auf Read-Only gesetzt und können damit nur noch gelesen werden (Löschen und Ändern ist nicht mehr möglich). Wenn Sie anschließend versuchen, die Dateien mit dem Befehl del *.txt zu löschen, erhalten Sie eine Fehlermeldung:

Zugriff abgelehnt

Der Löschbefehl wird nicht ausgeführt.

```
attrib *.*
```

Anzeigen der Attribute (Read-Only und Archivbyte) aller Dateien in einem Verzeichnis. Nur bei Read-Only gesetzten Dateien wird am Anfang einer Zeile am Bildschirm ein R und bei gesetztem Archivbyte ein A angezeigt.

Beispiel für Bildschirmdarstellung:

```

A      C:\MSDOS.KRS\MSDOS.7
A      C:\MSDOS.KRS\CONFIGUR.TXT
R A    C:\MSDOS.KRS\MSDOS.1
A      C:\MSDOS.KRS\MSDOS4.TXT
R A    C:\MSDOS.KRS\MSDOS5.TXT
R A    C:\MSDOS.KRS\MSDOS6.TXT
R A    C:\MSDOS.KRS\MSDOS.FUS
        C:\MSDOS.KRS\WSMSG5.OVR
        C:\MSDOS.KRS\WSOVLY1.OVR
        C:\MSDOS.KRS\WS.COM
        C:\MSDOS.KRS\MAILMRGE.OVR
A      C:\MSDOS.KRS\APPLY.TXT
R A    C:\MSDOS.KRS\MSDOS2.TXT

```

```
attrib -r *.txt
```

Löschen des Attributes Read-Only bei allen Dateien mit dem File-Typ TXT. Damit können die Dateien wieder gelöscht oder verändert werden.

```
attrib +a a:*.*
```

Damit wird das Archivbyte bei allen Dateien auf der Diskette in Laufwerk A gesetzt. Beim nächsten XCOPY oder BACKUP werden alle Dateien kopiert.

```
attrib -a a:*.bak
```

Bei allen Dateien mit dem File-Typ BAK wird das Archivbyte zurückgesetzt. Bei dem nächsten XCOPY (mit Parameter /M) oder BACKUP werden diese Dateien nicht kopiert.

```
attrib -a \MSDOS\*.*
```

Bei allen Dateien im Sub-Directory MSDOS wird das Archivbyte zurückgesetzt. Das gesamte Sub-Directory wird beim nächsten Befehl XCOPY (mit Parameter /M) oder BACKUP nicht kopiert.

Bemerkungen

Wenn Sie das Read-Only-Attribut mit `-r` bei Dateien (die dieses nicht besitzen) wieder löschen wollen, so wird dies nur ignoriert. Es erfolgt am Bildschirm keine Anzeige, daß kein entsprechendes Read-Only-Attribut gefunden wurde.

BACKUP *externer Befehl*

(Nur bei Festplattensystemen)

Zweck BACKUP sichert Dateien von Diskette oder Festplatte. Dabei können einzelne Dateien größer sein, als die Empfängerdisk Platz frei hat (dann werden Sie aufgefordert, eine neue Diskette einzulegen, und das Programm setzt das Sichern des Restes der Datei fort).

Wenn das File-Sharing (Netzwerkbetrieb) aktiv ist, können nur Dateien gesichert werden, auf die Sie Zugriff haben (siehe auch Befehl SHARE) – im anderen Falle erhalten Sie eine Fehlermeldung.

Gesichert werden kann:

Festplatte auf Diskette (oder Band) = üblicher Sicherungsvorgang

Festplatte auf Festplatte (oder Band)

Diskette auf Festplatte

Diskette auf Diskette

Kommandoaufbau

backup

backup (d:)(weg) (d:) (/x ...)

Die erste Angabe (d:)(weg) bezeichnet das Laufwerk bzw. das Directory, das gesichert werden soll. Wird der (weg) nicht angegeben, so wird nur das aktuelle Directory gesichert bzw. mit Angabe des Parameters /S das aktuelle Directory und die sich hierarchisch darunter befindlichen Sub-Directories. Die zweite Angabe (d:) bezeichnet das Ziellaufwerk, wohin die Daten gesichert werden.

/x ... /a

Hinzufügen von neuen zu sichernden Dateien zu bereits gesicherten (Sie werden aufgefordert, die letzte Sicherungsdiskette einzulegen).

/d:(tt.mm.jj)

Nur Dateien mit dem angegebenen Datum und nach dem angegebenen werden gesichert (wird das Datum nicht angegeben, so wird dafür das eingebene Tagesdatum verwendet). Ohne Datumsangabe muß /D am Ende der Befehlszeile stehen.

/s

Es werden alle Dateien des aktuellen Directory (Verzeichnis) und die weiter

vorhandenen Dateien des Sub-Directory darunter gesichert. Dateien in darüberliegenden Directories werden nicht gesichert (wenn eine ganze Platte gesichert werden soll, muß das aktuelle Directory das ROOT-Directory sein).

/l(:dateiname) Auf der 1. Sicherungsdiskette wird ein »Logbuch« angelegt, in dem alle gesicherten Dateinamen eingetragen werden. Geben Sie keinen Dateinamen an, so wird automatisch der Name BACKUP.LOG verwendet. Es wird unter anderem die Information gespeichert, auf welcher Sicherungsdiskette sich eine gesicherte Datei befindet. Die Sicherungsdatei können Sie sich mit dem Befehl TYPE am Bildschirm anzeigen lassen:

```
type a:backup.log
```

/m kopiert nur die Dateien, die seit dem letzten Sicherungsvorgang verändert wurden (wird durch das Betriebssystem überwacht), bzw. wenn mit ATTRIB das Archivbyte gesetzt wurde.

/t kopiert nur Dateien, die das Tagesdatum besitzen.

/t:tt.mm.jj kopiert nur Dateien mit einer Uhrzeit im Directory, die ab der angegebenen liegt.

Bemerkung Beachten Sie bitte, daß, außer wenn /a verwendet wurde, die Empfangsdiskette immer vor dem Kopieren gelöscht wird!

Beispiele backup c: a: /s

Alle Dateien/Programme der Festplatte Laufwerk C nach Laufwerk A sichern (die aktuelle und deren Sub-Directories werden gesichert).

```
backup c: a: /m
```

Alle Dateien, die im Laufwerk C vorhanden sind und seit dem letzten Backup-Vorgang verändert wurden bzw. bei denen mit

ATTRIB +a das Archivbyte gesetzt wurde, werden nach Laufwerk A gesichert.

Das Verzeichnis der Sicherungsdisketten kann mit dem Befehl dir angezeigt werden (zusätzlich befindet sich die Datei BACKUPID.@@@ auf der Diskette).

Das »Zurückholen« von mit BACKUP gesicherten Dateien darf nur mit RESTORE erfolgen (nicht mit COPY oder XCOPY).

Verwendung von BACKUP in einer Batch-Datei:

Beispiel Erstellen eines BATCH-Files:

```
copy con genback.bat
backup %1 %2 /a
```

F6-Taste und <RETURN>, um die Datei GENBACK.BAT zu speichern.

Der Aufruf erfolgt durch:

```
genback c: a:
```

Bemerkungen Damit werden vom aktuellen Verzeichnis in Laufwerk C alle Dateien gesichert (/a bedeutet »hinten anhängen«).

Wenn Sie vor der F6-Taste noch den BATCH-Befehl IF ERROR LEVEL verwenden, können folgende Codes abgefragt werden :

- 0 BACKUP ordnungsgemäß beendet
- 1 keine Dateien gefunden
- 2 File-Sharing-Konflikte (Netzwerke) – nicht alle Dateien gesichert
- 3 Sie haben BACKUP vor dem Bearbeitungsende abgebrochen (Ctrl+Break)
- 4 BACKUP durch MS-DOS mit Fehler abgebrochen

Näheres hierzu finden Sie im Kapitel 12 (Die BATCH-Befehle und ihre Anwendung).

BREAK *interner Befehl*

Zweck BREAK aktiviert die Programmunterbrechung durch Ctrl-Break (oder Ctrl-C). Verträgt sich nicht immer mit allen Anwendungsprogrammen (dann bitte auf OFF setzen). Mit den Tastenkombinationen Ctrl+Break oder Ctrl+C können Programme unterbrochen werden, wenn auf »ON« gestellt wird.

Kommandoaufbau

break on	mit Ctrl-Break kann ein Programm unterbrochen werden
break off	mit Ctrl-Break kann ein Programm nicht unterbrochen werden (ausgenommen Ein-/Ausgaben zu Peripheriegeräten)
break	Anzeigen, ob BREAK auf »off« oder »on« gesetzt ist

Bemerkungen Ctrl-Break ist immer gültig bei:
Standardeingaben und -ausgaben, Standard-Druckausgabe und Standard-AUX-Operationen.
BREAK wirkt nicht bei allen Anwendungsprogrammen! Viele Programme setzen beim Start automatisch BREAK auf OFF.
BREAK ist standardmäßig immer auf OFF (nach dem Systemstart). Soll dieser Standardwert auf ON sein, fügen Sie in die Datei CONFIG.SYS den Befehl BREAK=ON ein (siehe Kapitel 13 »Die Konfiguration von MS-DOS«).

CHDIR oder CD *interner Befehl*

Zweck CHDIR ändert das aktuelle Directory (Root- oder Sub-Directory) bzw. zeigt den Directory-Namen an.
Beachten Sie das Kapitel 8 »Das MS-DOS-Inhaltsverzeichnis« mit einer detaillierten Beschreibung.

Kommando Aufbau chdir (d:)(weg)
cd (d:)(weg)

Beispiele cd \

Direkte Rückkehr in das Root-Directory (unabhängig davon, in welchem Directory Sie sich aktuell befinden).

cd text

Zum Sub-Directory TEXT verzweigen. Dieses muß ein Sub-Directory des aktuellen Directory sein.

cd text\abt1

Zum Sub-Directory ABT1 verzweigen (dieses ist ein Sub-Directory von TEXT). Sie müssen immer den gesamten hierarchischen Weg angeben (der Befehl kann also nicht cd abt1 lauten). Da sich vor TEXT kein \ befindet, muß TEXT ein Sub-Directory des aktuellen sein.

Besser ist es, die Wege immer mit dem Root-Directory zu beginnen.

cd ..

Rückkehr zum hierarchisch vorher liegenden Directory. Wenn Sie sich bereits im Root-Directory befinden, bleibt dieser Befehl ohne Fehlermeldung wirkungslos.

cd a:\text\abt1

Im Laufwerk A über das Root-Directory und TEXT zum Sub-Directory ABT1 verzweigen, das das neue aktuelle Directory für Laufwerk A darstellt.

cd

Es wird Ihnen der Name des aktuellen Directory angezeigt (\, wenn es sich um das Root-Directory handelt):

A:\TEXT\ABT1

CHKDSK *externer Befehl*

Zweck CHKDSK überprüft eine Festplatte oder Diskette. Das gesamte Directory mit den Dateieinträgen wird überprüft. Vorhandene Fehler können mit dem Programm korrigiert werden. Es wird bei jedem Programmaufruf auch eine Statusmeldung angezeigt. Das Programm sollte immer angewendet werden, wenn der Verdacht besteht, daß eine Datei oder Festplatte/Diskette fehlerhaft ist.

Das Programm (CHecK DiSK = überprüfe Diskette) liest das Inhaltsverzeichnis und überprüft es auf Vollständigkeit und Fehler.

Fehlerhafte Sektoren kann das Programm nicht korrigieren. CHKDSK darf bei einem Netzwerk-Laufwerk nicht verwendet werden.

Kommandoaufbau

(d:)(weg)chkdsk (d:)(dateiname)(/x)

dateiname Name der zu überprüfenden Datei

/x /f korrigiert gefundene Fehler

/v zeigt laufend die gerade überprüfte Datei an

Beispiele

chkdsk

Aktuelles Laufwerk überprüfen und Statusmeldungen anzeigen.

```
chkdsk c:command.com
```

Programm COMMAND.COM auf dem Festplattenlaufwerk C überprüfen.

```
chkdsk c: /v /f
```

Festplattenlaufwerk C überprüfen und dabei die Verzeichnis-Namen und Datei-/Programm-Namen anzeigen (am Ende werden eventuelle Fehlermeldungen mit ausgegeben). Korrektur von fehlerhaften Dateien oder Daten, die einem Dateinamen nicht zugeordnet werden können.

```
chkdsk c: > druck
```

Festplattenlaufwerk C überprüfen und die Ausgaben in der Datei DRUCK im aktuellen Laufwerk abspeichern (diese Datei kann später am Bildschirm angezeigt oder gedruckt werden).

Folgende Angaben enthält der Status-Report am Bildschirm:

Name der Diskette/Festplatte

- Gesamtkapazität der Diskette/Festplatte in Bytes
- Anzahl und Gesamtgröße der MS-DOS-System-Dateien (ohne COMMAND.COM)
- Anzahl und Gesamtgröße der Sub-Directories
- Anzahl und Gesamtgröße der Dateien/Programme
- Anzahl der Zeichen in fehlerhaften Sektoren
- Angabe des noch freien Speichers auf der Diskette/Festplatte
- Gesamtgröße des RAM-Speichers
- Angabe des noch freien Speichers im RAM für Programmausführung

**Beispiel einer
CHKDSK-Ausgabe:**

```
Volume PLATTE           erzeugt 22 Aug 1986 14.14
362496 Bytes verfügbarer Platz auf Diskette
46080 Bytes in 3 versteckte Dateien
1024 Bytes in 1 Verzeichnisse
312320 Bytes in 44 Benutzerdateien
1024 Bytes in schlechte Sektoren
2048 Bytes frei auf der Diskette

524288 Bytes Speicherplatz insgesamt
466016 Bytes frei
```

Wenn Sie zum Befehl einen Dateinamen angeben, erfolgt zusätzlich die Anzeige, ob die Datei fortlaufend oder in nicht zusammenhängenden Blöcken gespeichert ist:

Alle angegebenen Dateien sind fortlaufend.

A:\ETI-LST.BAS

Enthält 3 nichtfortlaufende Blöcke.

Bemerkungen

MS-DOS verwaltet den Speicher in Sektoren. Wird eine neue Datei gespeichert, so beginnt die Suche nach freien Sektoren am Anfang der Diskette/Festplatte. Findet MS-DOS einen freien Sektor, wird dort gespeichert. Sind noch weitere Daten für das Speichern vorhanden, so wird die Suche fortgesetzt. Dabei kommt es häufig vor, daß der nächste freie Sektor nicht an den vorhergehenden anschließt. Damit wird die Datei in »nichtfortlaufenden« Blöcken gespeichert.

Dies kommt um so häufiger vor, je öfter Sie Dateien löschen und speichern, da die Größen der Dateien teilweise sehr unterschiedlich sind.

Wenn nicht zuordenbare Bereiche von Daten auf der Diskette/Festplatte gefunden werden (nur wenn beim Befehl /f angegeben wurde), erhalten diese von CHKDSK einen eigenen Dateinamen (DATEI000.CHK, DATEI001.CHK usw.). Sie können sich anschließend die Dateien ansehen und gegebenenfalls löschen.

CLS *interner Befehl*

Zweck CLS löscht den Bildschirm und setzt den Cursor in die linke obere Ecke des Monitors.

Kommandoaufbau cls

COMMAND *externer Befehl*

Zweck COMMAND ruft das Programm COMMAND.COM als externen Befehl auf. Das Programm stellt die Verbindung zwischen Ihnen und dem Computer dar. Es wird automatisch beim Systemstart geladen und enthält unter anderem alle internen Befehle (COMMAND.COM stellt den Befehls-Interpreter des MS-DOS dar und wird beim Systemstart in den Speicher geladen).

Arbeiten Sie mit diesem Befehl, müssen Sie über den Aufbau von MS-DOS detaillierte Kenntnisse besitzen (daher wird hier nicht weiter auf diesen Befehl eingegangen).

Kommandoaufbau (d:)(weg)command (d:)(weg)(cttydev)(/x)
cttydev Name des Gerätetreibers (normalerweise CON).
Beachten Sie hierzu auch den Befehl CTTY
/x /e: # damit kann die Größe des Environment-Bereichs angegeben werden (# ist eine Zahl zwischen 128 und 32768)
/p ein neuer COMMAND.COM wird geladen und der alte gelöscht (es kann mit EXIT nicht in die vorherige Ebene zurückgekehrt werden)
/c Befehl definiert eine Zeichenkette bis zur maximalen Eingabelänge von 127 Zeichen (diese wird von COMMAND.COM als MS-DOS-Befehl ausgeführt)

Beispiele command

Erneutes Laden von COMMAND.COM (dies kann auch von BASIC aus erfolgen). Damit befinden Sie sich auf einer höheren Ebene. Der bisherige Inhalt im Environment-Bereich ist unverändert. Wenn Sie mit PATH usw. Änderungen vornehmen, so werden diese entsprechend gespeichert. Nach Eingabe des Befehls

exit

kehren Sie wieder in die vorhergehende Befehlsebene zurück und alle Eingaben mit PATH usw. in den Environment-Bereich sind gelöscht; es gelten wieder die alten Definitionen.

```
command /e:512 /p
```

Der bisher geladene COMMAND.COM wird durch einen neuen ersetzt (/p). Der Environment-Bereich wird auf 512 Zeichen Größe festgelegt (Standardwert ohne Angabe ist 128). Der Environment-Bereich enthält Angaben:

- von welchem Laufwerk/Directory bei Bedarf der COMMAND.COM nachgeladen werden kann
- die mit PATH definierten Wege zu Directories
- alle mit SET gesetzten Variablen.

Den Environment-Bereich können Sie sich jederzeit mit dem Befehl SET anzeigen lassen.

```
command /C dir
```

Dabei wird ein neuer COMMAND unterhalb des bisherigen (oder z.B. BASIC) gestartet, der Befehl DIR ausgeführt und wieder zum vorhergehenden zurückgekehrt. In BASIC erfolgt der Aufruf mit

```
shell "command /C dir"
```

und nach Ausführung sind Sie wieder in BASIC (als vorherigem Programm).

Beachten Sie zu diesem Befehl auch das Kapitel 13.

COMP *externer Befehl*

Zweck COMP vergleicht den Inhalt von zwei Dateien/Programmen oder zwei Gruppen von Dateien auf Gleichheit. Kann nach einem Kopiervorgang (COPY-Befehl) verwendet werden, um festzustellen, ob Original und Kopie einer Datei identisch sind (wenn eine komplette Diskette mit DISKCOPY kopiert wurde, verwenden Sie besser den Befehl DISKCOMP). Wenn mehr als 10 Unterschiede festgestellt wurden, bricht das Programm den Vergleich automatisch ab. Es wird ebenfalls geprüft, ob das Dateienkennzeichen gespeichert ist. Gefundene Fehler werden am Bildschirm angezeigt.

Kommando Aufbau (d:)(weg)comp (/x...) (dateiname1 (dateiname2))

/x... /a Zeigt den Unterschied, die Zeile davor und danach an (wenn die Anzahl der unterschiedlichen Zeilen dazwischen mehr als eine ist, wird diese angezeigt und anschließend drei Punkte)

/b Vergleich erfolgt Zeichen für Zeichen

/c Alle Buchstaben werden als Großbuchstaben verglichen

/l Vergleich erfolgt Zeile für Zeile

/lb # Vergleich erfolgt Zeile für Zeile (b # gibt die Größe des Zwischenspeichers in Zeilen an – ohne Angabe wird als Wert automatisch 100 angenommen)

/n Zusätzlich wird die Zeilennummer angezeigt

/t Tabulatorzeichen werden nicht als Leerräume interpretiert

/w Tabulatorzeichen oder Leerräume, die größer als ein Zeichen sind, werden als ein Leerraum interpretiert (Leerräume am Zeilenanfang und -ende werden ignoriert)

/ # Legt die Zeilen fest, nach denen die Ausgabe von Unterschieden in den beiden Dateien neu erfolgt (ohne Angabe ist der Wert 2)

Beispiele Es liegen zwei Dateien folgenden Inhalts vor:

Datei A.TXT

Zeile eins

Zeile zwei

Zeile Unterschied eins

```
Zeile vier
Zeile fünf
Zeile Unterschied zwei
eine Zeile länger
Datei B.TXT
Zeile eins
Zeile zwei
Zeile Unterschied eins
Zeile vier
Zeile fünf
Zeile zweiter Unterschied
```

```
comp a.txt b.txt
```

Die beiden Dateien werden miteinander verglichen – Bildschirmanzeige:

```
***** a.txt
Zeile zwei
Zeile Unterschied eins
Zeile vier
***** b.txt
Zeile zwei
Zeile Unterschied eins
Zeile vier
*****

***** a.txt
Zeile fünf
Zeile Unterschied zwei
eine Zeile länger
***** b.txt
Zeile fünf
Zeile zweiter Unterschied
*****
```

Für jeden gefundenen Fehler wird jeweils die letzte übereinstimmende Zeile, die unterschiedliche und die nächste gleiche Zeile der beiden Dateien angezeigt. Geben Sie für die Fehlerausgabe einen anderen Wert an:

```
comp /3 a.txt b.txt
```

Es folgt die Anzeige:

```
***** a.txt
Zeile zwei
Zeile Unterschied eins
Zeile vier
Zeile fünf
Zeile Unterschied zwei
eine Zeile länger
***** b.txt
Zeile zwei
Zeile Unterschied eins
Zeile vier
Zeile fünf
Zeile zweiter Unterschied
*****
```

Vergleichen Sie die beiden Dateien in der Form, daß alle Zeichen als Großbuchstaben verglichen werden, so wird der erste Unterschied nicht festgestellt:

```
comp /c a.txt b.txt
***** a.txt
Zeile fünf
Zeile Unterschied zwei
eine Zeile länger
***** b.txt
Zeile fünf
Zeile zweiter Unterschied
*****
```

Wollen Sie wegen einer besseren Übersicht auch die Zeilennummer anzeigen lassen, so benötigen Sie noch /n:

```
comp /n a.txt b.txt
***** a.txt
    2: Zeile zwei
    3: Zeile Unterschied eins
    4: Zeile vier
***** b.txt
    2: Zeile zwei
    3: Zeile Unterschied eins
    4: Zeile vier
*****
```

```

***** a.txt
      5: Zeile fünf
      6: Zeile Unterschied zwei
      7: eine Zeile länger
***** b.txt
      5: Zeile fünf
      6: Zeile zweiter Unterschied
*****

```

Geben Sie zwei sich widersprechende Parameter an, so erhalten Sie eine Fehlermeldung:

```

comp /l/b a.txt b.txt
fc: Schalter nicht kompatibel

```

Bei fehlerhafter Eingabe der Angaben zum Befehl COMP erhalten Sie folgende Fehlermeldung:

```

Format: fc [/a] [/b] [/c] [/l] [/lbNN] [/w] [/t]
[/n] [/NNNN] Datei1 Datei2

```

Ist der Zwischenspeicher für die Bearbeitung der von COMP gefundenen Unterschiede zu klein, so erhalten Sie eine Fehlermeldung und eventuell eine teilweise Ausgabe der Differenzen (soweit diese bearbeitbar waren):

```

comp /lb1 a.txt b.txt
Resync hat nicht geklappt. Dateien zu unterschied-
lich\n² ö***** a.txt
Zeile Unterschied eins
***** b.txt
Zeile Unterschied eins
*****

```

In der Fehlermeldung erscheinen hinter unterschiedlich einige Zeichen, die einen kleinen Fehler im Programm COMP darstellen. Diese brauchen Sie nicht zu berücksichtigen.

Dateien mit dem Typ EXE, COM, SYS, OBJ, LIB oder BIN werden automatisch Zeichen für Zeichen, alle anderen Zeile für Zeile verglichen. Wollen Sie dies ändern, so müssen Sie entweder den Parameter /L oder /B angeben.

```
comp /b a.txt b.txt
0000001E: 55 75
0000002A: 65 45
0000004E: 55 7A
0000004F: 6E 77
00000050: 74 65
00000051: 65 69
00000052: 72 74
00000053: 73 65
00000054: 63 72
00000055: 68 20
00000056: 69 55
00000057: 65 6E
00000058: 64 74
00000059: 20 65
0000005A: 7A 72
0000005B: 77 73
0000005C: 65 63
0000005D: 69 68
0000005E: 0D 69
0000005F: 0A 65
00000060: 65 64
00000061: 69 0D
00000062: 6E 0A
fc: a.txt länger als b.txt
```

Die Vergleiche erfolgen zeichenweise, und die Bildschirmausgabe erfolgt in HEX-Darstellung. Die Bedeutung der einzelnen Werte finden Sie im Anhang IV. In der ersten Spalte ist angegeben, um das wievielte Zeichen es sich handelt, in der zweiten das Zeichen aus A.TXT und in der dritten das Zeichen aus B.TXT.

COPY *interner Befehl*

Zweck Kopiert eine oder mehrere Dateien auf ein anderes oder das gleiche Laufwerk (wenn der Kopiervorgang auf dem gleichen Laufwerk im gleichen Directory erfolgt, müssen die Dateinamen unterschiedlich sein). COPY kann Dateien während des Kopiervorganges verketteten (Concatenation). Überträgt Daten zwischen verschiedenen Ein- und Ausgabekanälen des Computers (z.B. vom Bildschirm zum Drucker).

Kommando Aufbau `copy (d:)dateiname1 (d:)dateiname2 (/x)`

`copy (d:)dateiname1 d: (/x)`

`copy (d:)dateiname1 (d:)weg (/x)`

`copy (d:)weg (d:)(weg) (/x)`

`dateiname1` Name der zu kopierenden Datei

`dateiname2` Name der neuen Datei

`d:` Name des Laufwerkes

`/x` `/v` nach dem Kopieren auf Fehler überprüfen

`/a` Texte kopieren (ASCII-Files) – das Dateiendekennzeichen wird nach dem Kopieren hinzugefügt

`/b` Binäre Dateien kopieren (in der Regel Programme) – alle Zeichen einschließlich dem Dateiendekennzeichen (Ctrl-Z bzw. Hexadezimal 1A) werden kopiert

Beispiele `copy test.txt a:`

Kopiert die Datei TEST.TXT vom aktuellen Laufwerk mit dem gleichen Namen auf die Diskette in Laufwerk A.

`copy test.txt test.alt`

Kopiert die Datei TEST.TXT auf dem aktuellen Laufwerk. Das Duplikat auf dem gleichen Laufwerk/Directory erhält den Namen TEST.ALT.

`copy test.alt a:\sub\test`

Kopieren der Datei TEST.ALT vom aktuellen Laufwerk in das Sub-Directory sub in Laufwerk A mit dem neuen Namen TEST.

```
copy *.sys test.doc
```

Alle Dateien mit dem File-Typ SYS werden in der neuen Datei TEST.DOC zusammengefaßt (oder Befehl `copy config.sys+ansi.sys test.doc`, um die Dateien CONFIG.SYS und ANSI.SYS zu einer neuen Datei TEST.DOC zusammenzufügen).

```
copy autoexec.bat con
```

Dateiinhalte von AUTOEXEC.BAT auf den Bildschirm (con) »kopieren«.

```
copy autoexec.bat prn
```

Dateiinhalte von AUTOEXEC.BAT auf den Drucker (prn) »kopieren«.

```
copy con autoexec.neu
```

Alles, was von der Tastatur nach dem Befehl eingegeben wird, wird in der Datei AUTOEXEC.NEU gespeichert – bzw. erst wenn die Tasten F6 bzw. Ctrl-Z und <RETURN> gedrückt werden.

```
copy autoexec.bat autoexec.bat
```

Es erscheint daraufhin die Fehlermeldung

```
Datei kann nicht in sich selbst kopiert werden
      0 Datei(en) kopiert
```

(eine Datei kann nicht innerhalb des aktuellen Directory mit dem gleichen Namen kopiert werden).

```
copy test.alt a:\sub
```

Kopieren der Datei TEST.ALT aus dem aktuellen Directory in das Sub-Directory sub in Laufwerk A mit dem gleichen Namen. Der Dateiname muß nur einmal angegeben werden (eine zweite Dateinamen-Angabe ist nur erforderlich, wenn der Name der zu kopierenden Datei in das Ziel-Directory ein anderer sein soll).

```
copy autoexec.bat+config.sys start.doc
```

Aus den Dateien AUTOEXEC.BAT und CONFIG.SYS wird eine neue Datei mit dem Namen START.DOC gebildet (»Concatenation«).

```
copy \schneide \schneide2
```

Kopieren aller Dateien aus dem Sub-Directory `schneide` in das bestehende Sub-Directory `schneide2` mit den gleichen Namen.

```
copy test.doc+,, a:test.bsp
```

Mit dem Zusatz `+,,` wird erreicht, daß die Datei `TEST.DOC`, die nach Laufwerk `A` mit dem neuen Namen `TEST.BSP` kopiert wird, das von Ihnen beim Start des Computers eingegebene Datum und die Uhrzeit erhält.

```
copy test.alt+*.txt
```

Es werden alle Dateien mit dem File-Typ `TXT` aus dem aktuellen Directory kopiert. Die Dateien erhalten den Namen `TEST.ALT` (Concatenation).

CTTY *interner Befehl*

Zweck CTTY ändert das Gerät, vom dem aus Befehle eingegeben werden und Anzeigen erfolgen (z.B. können Ein-/Ausgaben anschließend über ein Modem erfolgen). Normalerweise ist dies die Tastatur (und der Bildschirm).

Kommandoaufbau cty device

device AUX: Hilfeingang (normalerweise COM1)

CON Tastatureingabe/Bildschirmausgabe
(Standard)

COM1 erste serielle Schnittstelle

COM2 zweite serielle Schnittstelle (soweit vorhanden)

Beispiele cty aux

Jetzt können Ein-/Ausgaben z.B. über ein Terminal erfolgen, das an AUX angeschlossen ist (in der Regel handelt es sich bei AUX um die erste V.24-Schnittstelle = COM1).

ctty con

Standardzustand herstellen (Tastatureingabe und Bildschirmausgabe).

Bemerkungen Mit vielen Programmen und Programmiersprachen (z.B. BASIC) funktioniert dieser Befehl nicht, da diese die Schnittstellen (z. B. Bildschirm) direkt ansprechen, also das MS-DOS »umgehen«.

DATE *interner Befehl*

Zweck Anzeigen und/oder Ändern des Tagesdatums (Systemdatum). Ein einmal eingegebenes Datum bleibt bis zur nächsten Änderung gespeichert, auch wenn zwischenzeitlich der PC abgeschaltet worden ist. Da Ihr Schneider PC eine Batterie hat, muß das Datum nur einmal eingestellt werden (bis die Batterie leer ist).

Kommando Aufbau `date`
`date tt-mm-jj`

mm = Monat 1 bis 12
tt = Tag 1 bis 31
jj = Jahr 80 bis 99 oder 1980 bis 2099

Beispiele `date 16-05-85`

Das neue Systemdatum ist anschließend der 16. Mai 1985. Es erscheint keine weitere Anzeige auf dem Bildschirm bzw. es kommt sofort wieder der Prompt.

`date`

Anschließend wird das gespeicherte Systemdatum angezeigt

Aktuelles Datum ist Don 16.05.1985

Neues Datum eingeben (tt-mm-jj):

und Sie können ein neues Datum eingeben oder durch Drücken der Taste <RETURN> das Datum unverändert belassen.

Bemerkungen Wenn Sie das Datum in einem falschen Format eingeben, erhalten Sie die Fehlermeldung

Ungültiges Datum

Neues Datum eingeben (tt-mm-jj):

und Sie müssen das Datum neu eingeben.

DEBUG *externer Befehl*

Zweck Es handelt sich primär um ein Testprogramm für Programme. Ferner ist es ein Werkzeug zur Programmierung, das in den Speicher geladene Programme anschließend testet. Für die Arbeit mit DEBUG sind Kenntnisse mit dem Maschinencode der Prozessoren 8088 usw. erforderlich.

Zu diesem Befehl finden Sie auch in Ihren Unterlagen zum Schneider PC keinerlei Angaben. Genauere Beschreibungen würden auch den Rahmen dieses Buches sprengen, da zur Bedienung von DEBUG grundlegende Kenntnisse über die internen Strukturen des Speichers usw. notwendig sind. Daher lernen Sie im folgenden die Befehle von DEBUG nur ganz allgemein kennen. Zur Anwendung beachten Sie bitte, daß es bei fehlerhaften Eingaben sehr leicht zu irreparablen Fehlern auf der Diskette oder Festplatte kommen kann!

Kommandoaufbau debug ((dateiname) (variable)) (< eingabe)

Befehl a (adresse)

Mit diesem Befehl kann anschließend Assembler-Code (8088 usw.) eingegeben werden. Numerische Werte müssen hexadezimal eingegeben werden. Eingabefehler werden sofort angezeigt.

Wird eine adresse angegeben, so wird der Code an der definierten Adresse in den Speicher eingegeben.

Befehl c bereich adresse

Dieser Befehl vergleicht zwei Bereiche des Speichers auf Gleichheit. Wird kein Unterschied festgestellt, so erfolgt keine Anzeige. Im anderen Fall erfolgt die Anzeige mit
adresse1 Byte1 Byte2 adresse2

Beispiel c0:0 15 cs:10

Vergleichen der Speicheradresse 0000:0000 bis 0000:0004 (= 5 Speicherstellen) mit der Adresse CS:000A (CS wird durch die im Code-Segment-Register gespeicherte Adresse ersetzt). Eventuelle Unterschiede werden angezeigt (Beispiel):

0000:0000	88	AB	1158:0010
0000:0001	51	0D	1158:0011
0000:0002	96	BD	1158:0012
0000:0004	60	AB	1158:0014

Befehl **d (adresse)(Lwert)**
d (bereich)

Zeigt den Speicherinhalt ab der angegebenen Adresse an. Angezeigt werden

- Speicheradresse
- Speicherinhalt (hexadezimal)
- Speicherinhalt (ASCII-Zeichen – nicht darstellbare Werte als Punkt)

In jeder Zeile werden 16 Zeichen angezeigt. Ohne weitere Angaben werden immer 128 Byte dargestellt.

Beispiel `dc:0100 10f`

Ab der Adresse 100h wird aus dem aktuellen Segment (Adresse im Register CS) der Speicher angezeigt (16 Byte = 0fh):

```
ODDC:0100 02 EB F2 46 C6 44 FF 00-B8 02 3D CD 21
72 46 8B
```

usw.

Befehl **e adresse (liste)**

Es wird eine Speicheradresse mit ihrem Inhalt angezeigt. Sie können dem Speicher einen neuen Wert zuweisen (die anschließende Betätigung der Leertaste führt zur Anzeige der nächsten Speicherzelle). Wurde ein – eingegeben, so wird die davor liegende Speicherzelle angezeigt.

Beispiele `e cs:0100`

Es wird der Speicher ab der Adresse 100h im aktuellen Segment angezeigt:

```
6603:0100 12._
```

Sie können hier neue Werte eingeben:

```
6603:0100 12.01      EB.      00.42
```

```
e cs:0100 02 EB F2 46 C6
```

Ab der Adresse 100h im aktuellen Segment werden die Werte der nächsten 5 Byte mit 02h, EBh, F2h, 46h und C6h überschrieben.

Befehl f bereich liste

Enthält die Liste weniger Bytes, als der angegebene Bereich groß ist, wird die Liste so lange wiederholt, bis der Bereich gefüllt ist.

Beispiel f0DDC:0100 1100 00

Der Speicher ab der Adresse 0DDC:0100 bis 0DDC:1ff (L 100) wird mit 00h gefüllt (00h wird so lange wiederholt, bis der angegebene Bereich gefüllt ist).

Befehl g (= adresse1) (adresse2...)

Das im Speicher befindliche Programm wird ausgeführt. Wird =adresse1 angegeben, so wird das Programm ab dieser Adresse ausgeführt. Wird adresse2 usw. angegeben, so stoppt DEBUG bei Erreichen dieser Adresse im Speicher die Ausführung und zeigt Ihnen alle Register an.

Befehl h wert wert

Hexadezimal rechnen. Es werden Ihnen zwei Ergebnisse angezeigt. Das erste ist der addierte, das zweite der subtrahierte Wert.

Befehl i wert

Lesen und Anzeigen des Wertes eines Ein- oder Ausgabekanals (wert ist die Kanaladresse).

Befehl l (adresse (laufwerk sektor anzahl))

Laden einer Datei oder von Sektoren in den Speicher. Wenn eine Datei geladen wird, so muß mit dem Befehl n ein Name vorher definiert werden (oder beim Aufruf von DEBUG wird ein Dateiname mit angegeben). Die Anzahl der gelesenen Bytes wird im Register CX abgelegt. Ohne weitere Angaben wird die Datei ab der Adresse CS:0100 in den Speicher geladen (gilt nicht für EXE-Dateien).

Beispiel l 0ddf:0100 1 00 0f

In den Speicher werden ab der Adresse 0DDF:0100 von der Diskette in Laufwerk B (0 = A, 1 = B, 2 = C usw.) ab dem Sektor 0 15 (0F) Sektoren geladen.

Befehl m bereich adresse

Einen Teil des Speichers an eine andere Adresse verschieben.

Beispiel m cs:0100 10a cs:0500

Der Speicherinhalt von cs:0100 bis cs:010A wird an die Adresse cs:0500 verschoben.

Befehl n dateiname (dateiname...)

Mit diesem Befehl werden Dateinamen definiert.

Beispiel n datei.exe
l
n daten1.dat daten2.dat
g

Der Name des Programmes DATEI.EXE wird angegeben und dieses mit dem Befehl l geladen. Anschließend werden zwei Datendatei-Namen angegeben. Mit g wird das Programm gestartet. Diese Eingaben entsprechen dem folgenden Befehl auf MS-DOS-Ebene:

```
datei daten1.dat daten2.dat
```

Geben Sie jetzt den Befehl write ein, wird das zu testende Programm unter dem Namen DATEN1.DAT gespeichert. Deshalb bezeichnen Sie vor jedem Befehl load oder write den Namen noch einmal genau.

Vier bestimmte Bereiche des Arbeitsspeichers beeinflussen Sie mit dem Befehl name:

CS:005C	FCB für die 1.Datei
CS:006C	FCB für die 2.Datei
CS:0080	Anzahl der Zeichen des Namens
CS:0081	alle eingegebenen Zeichen (Dateiname)

Befehl o adresse byte

Sendet ein Byte zu einem Ein- oder Ausgabekanal (die Angabe muß immer hexadezimal erfolgen).

Beispiel o 2f8 4f

Das Byte mit dem Wert 4F wird an die Kanal-Adresse 2F8 gesandt.

Befehl p (=adresse)(wert)

Das geladene Programm wird schrittweise ausgeführt.

Beispiel p = 100 4

Ab der Adresse 0100h werden die nächsten 4 Programmschritte ausgeführt und die Befehle sowie alle Register mit Inhalten angezeigt.

Befehl q (quit)

Programm DEBUG beenden.

Befehl r (register)

Anzeigen und/oder Verändern eines oder mehrerer Register der CPU. Bei Eingabe eines einzelnen Registernamens wird der Inhalt angezeigt, und Sie haben die Möglichkeit, diesen zu verändern. Bei der Betätigung von <RETURN> verändert sich der Wert nicht.

Register	AX	Funktion Akkumulator
	BX	Adreßregister
	CX	Zählregister
	DX	Adreßregister für Ein-/Ausgabe
	CS	Codesegment
	DS	Datensegment
	SS	Stapelsegment
	ES	Extrasegment
	SP	Stapelzeiger
	BP	Basiszeiger
	SI	Indexregister
	DI	Indexregister
	IP	Programmzähler
	F	Statusregister

Kennzeichen für das F-Register:

Überlauf	OV	Überlauf	NV	kein Überlauf
Richtung	DN	Vermindern	UP	Erhöhen
Unterbrechung	EI	Aktiviert	DI	nicht aktiviert
Vorzeichen	NG	Negative	PL	Plus
Null	ZR	Null	NZ	nicht Null
zusätzlicher Übertrag	AC	Hilfsübertrag	NA	kein Hilfsübertrag
Parität	PE	Gerade	PO	Ungerade
Übertrag	CY	Übertrag	NC	kein Übertrag

Bei der Eingabe von F als Registername werden alle Kennzeichen angezeigt, und Sie können die gewünschten Änderungen bzw. Zeichen in beliebiger Reihenfolge eingeben.

Wenn Sie mehr als ein Kennzeichen eingeben, erhalten Sie die Fehlermeldung:

DF

Doppeltes Kennzeichen

Bei einem falschen Kennzeichen:

BF

Falsches Kennzeichen

Wenn DEBUG gestartet wird, sind die Kennzeichen nicht gesetzt und die Register auf Null gesetzt (mit Ausnahme von Register IP, das den Wert 0100h, und SP, das den Wert FFEEh beinhaltet).

Ein Register kann geändert werden, indem R eingegeben wird und dahinter die Registerbezeichnung. Beispiel:

```
rcx
CX 0000
:
```

Anschließend können Sie einen neuen Wert in das Register eingeben.

Befehl s bereich liste

Durchsucht den angegebenen Bereich nach der vorgegebenen Liste von Zeichen und zeigt die erste Adresse an.

Beispiel s cs:0100 L800 04 00 72 10

Im Codesegment wird ab der Adresse 0100h in der Länge von Hexadezimal 800 (= 2048 Zeichen) die Zeichenkette 04h/00h/72h/10h gesucht.

Befehl t (=adresse) (wert)

Der Befehl führt das aktuelle Programm schrittweise aus und zeigt nach jedem Befehl alle Register sowie die nächste auszuführende Funktion. Wird keine Adresse und/oder der Wert eingegeben, so wird eine Funktion ab der aktuellen Adresse (steht in Register IP) ausgeführt. Der Befehl kann nicht nur für das RAM, sondern auch für das ROM benutzt werden.

Beispiel t =cs:0100 10

Ab der Segmentadresse 0100h werden 16 (10h) Funktionen ausgeführt. Nach jeder werden alle Register und die Funktion angezeigt.

Befehl u (adresse) (L wert)**u (bereich)**

Wandelt den Speicherinhalt in Assembler-(Maschinen-) Befehle um und zeigt diese mit der Speicheradresse an.

Beispiel u cs:0100h

Dabei wird z.B. angezeigt:

....:0100	E84CFE	CALL	0A4F
....:0103	7306	JNB	0COB

usw.

Befehl w (adresse (laufwerk sektor anzahl))

Schreibt Daten vom Speicher auf die Diskette oder Festplatte. Wird nichts weiter angegeben, so werden die Daten im Speicher ab der Adresse cs:0100 auf die Diskette mit dem definierten Namen (vorher mit dem Befehl n oder bei Aufruf des DEBUG festgelegt) in der im Register CX festgelegten Länge gespeichert.

Wurde vorher der Befehl g oder t verwendet, so muß erst der Wert in Register CX neu eingegeben werden. Dies ist ebenfalls notwendig, wenn Sie die Länge der Daten verändert haben. Sie können aber auch direkt, unter Umgehung des MS-DOS, auf die Diskette speichern (siehe auch Befehl l).

DEL oder ERASE *interner Befehl*

Zweck Löschen von Dateien/Programmen. Es kann wahlweise der Befehl DEL oder ERASE verwendet werden. Dateien/Programme mit dem Attribut »read-only« können ebenso wie Sub-Directory-Namen (diese nur mit dem Befehl RMDIR oder RD) nicht gelöscht werden (siehe Befehl ATTRIB).

Kommandoaufbau del dateiname
del (d:)weg(dateiname)
erase dateiname
erase (d:)weg(dateiname)

Beispiele del test.alt

Löschen der Datei TEST.ALT im aktuellen Directory (und aktuellen Laufwerk).

```
del a:\sub\test
```

Löschen der Datei TEST aus dem Sub-Directory sub auf der Diskette in Laufwerk A.

```
del a:\sub
```

oder

```
del a:\sub\*.*
```

Mit diesem Befehl werden alle Dateien/Programme im Sub-Directory sub auf der Diskette in Laufwerk A gelöscht.

```
del *.*
```

Löschen aller Dateien/Programme im aktuellen Laufwerk/Directory. Vor dem Löschen werden Sie noch kurz befragt, ob Sie sicher sind, alle Dateien löschen zu wollen.

Sind Sie sicher (J/N)?

DIR *interner Befehl*

Zweck Anzeigen von Inhaltsverzeichnissen mit den gespeicherten Informationen (File-Name, File-Typ, Größe der Datei in Bytes, Datum und Uhrzeit sowie Laufwerkbezeichnung, Name der Diskette/Festplatte, Directory-Name, Anzahl der Dateieinträge einschließlich der Sub-Directory-Namen, freier Speicher). Datum und Uhrzeit werden abhängig vom Landescode (siehe Kapitel 13 »Die Konfiguration von MS-DOS«) angezeigt. Datum und Uhrzeit werden jeweils bei Anlage oder Änderung einer Datei neu gespeichert.

Kommandoaufbau `dir (d:)(dateiname) (/x)`
`dir (d:)(weg) (/x)`
 /x /p Ausgabe mit Stopp nach jeder angezeigten Bildschirmseite
 /w Die Ausgabe erfolgt nur mit den Dateinamen (je 5 Dateinamen in einer Zeile) – man nennt dies die Anzeige in »weiter« Form

Beispiele `dir`
 oder
`dir *.*`

Es wird das Inhaltsverzeichnis des aktuellen Directory/Laufwerks angezeigt.

Volume in Laufwerk C hat keinen Namen
 Verzeichnis von C:\

WSC	COM	25600	1.01.80	12.03
WSOVLY1	OVR	46080	1.01.80	3.59
WSMSGs	OVR	27392	1.01.80	3.57
MAILMRGE	OVR	12800	1.01.80	4.49
WSTYPE	COM	1984	24.07.85	9.03
1		0	26.10.85	2.24
KAPITEL	BAK	84864	26.10.85	2.11
KAPITEL	111	85248	26.10.85	2.24
		8 Datei(en)	77824 Bytes frei	

`dir *.sys`

Es werden nur Dateien mit dem File-Typ SYS aus dem aktuellen Directory angezeigt.

```
dir /w
```

Anzeigen des Inhaltsverzeichnisses vom aktuellen Directory in »weiter« Form.

Beispiele Volume in Laufwerk C hat keinen Namen
Verzeichnis von C:\

```
WSC    COM    WSOVLY1 OVR    WSMMSGs  OVR    MAILMRGE OVR
WSTYPE COM    1              KAPITEL BAK    KAPITEL 111
2
          9 Datei(en)    77312 Bytes frei
```

```
dir > prn
```

Drucken des Inhaltsverzeichnisses vom aktuellen Directory/Laufwerk.

```
dir a:\sub
```

Anzeigen des Inhaltsverzeichnisses der Diskette im Laufwerk A – nur aus dem Sub-Directory sub.

```
dir c: | sort > prn
```

Ausdrucken des sortierten Inhaltsverzeichnisses aus Laufwerk C. Ohne > prn erfolgt die sortierte Ausgabe auf den Bildschirm.

```
dir c: | sort /+14 | more
```

Anzeigen der sortierten Dateien aus Laufwerk C (es wird ab der Spalte 14 aufsteigend sortiert – wenn noch /R hinter +14 angegeben wird, erfolgt der Sortiervorgang absteigend). Nach jeder Bildschirmseite stoppt der Ausgabevorgang (more). Beachten Sie hierzu auch das Kapitel 9.

Bemerkungen Wenn statt einer Dateigrößenangabe <DIR> erscheint, so handelt es sich bei dem Namen um einen Sub-Directory-Eintrag.

Befehl	entspricht dem Befehl
DIR	DIR *.*
DIR dateiname	DIR dateiname.*
DIR .typ	DIR *.ext
DIR .	DIR oder DIR *.*
DIR ..	Es wird das Inhaltsverzeichnis der hierarchisch vorher liegenden Directory angezeigt

Hinter dem Eintrag . in einem angezeigten Inhaltsverzeichnis verbirgt sich der Name des aktuellen Directory und hinter .. der Name des hierarchisch vorher liegenden Directory.

Beispiele für einen »normalen« Directory-Ausdruck und einen »weiten« von einem Sub-Directory auf einer Festplatte:

Volume in Laufwerk C ist KURS

Verzeichnis von C:\GWBASIC

BASICA1	EXE	58720	7.02.84	0.10
FRANCE	BAS	512	9.11.84	0.10
BASICA	EXE	70784	5.11.84	0.14
ADR	BAS	3010	1.01.80	0.14
ADR2	BAS	930	1.01.80	0.15
ADR3	BAS	1094	10.04.85	0.15
WS		47941	10.04.85	0.15
ELITE	BAS	29	8.05.85	8.49
ETI	BAS	643	6.02.85	0.55
WINDOW	BAS	512	1.01.80	0.05
ANWEND	<DIR>		6.06.85	16.03
		13 Datei(en)	4931584 Bytes frei	

Volume in Laufwerk C ist KURS

Verzeichnis von C:\GWBASIC

.		..		BASICA1	EXE	FRANCE	BAS	BASICA	EXE
ADR	BAS	ADR2	BAS	ADR3	BAS	WS		ELITE	BAS
ETI-DOS	BAS	ETI-MP1	BAS	ETI-OA	BAS	ETI-PAIN	BAS	ETI-EW1	BAS
ETI-EW2	BAS	ETI-MP2	BAS	ETI-SWL	BAS	ETI	BAS	ETI-GER	BAS
ETI-TKE	BAS	FREIB	BAS	WINDOW	BAS	ANWEND			
		24 Datei(en)	4927488 Bytes frei						

DISKCOMP *externer Befehl*

Zweck Überprüft zwei Disketten auf absolute Gleichheit (der Vergleich erfolgt Track-by-Track). Die beiden Disketten müssen das gleiche Format besitzen und vom Inhalt her absolut identisch sein.

Wenn Sie eine neue Diskette formatiert haben und mit dem Programm COPY eine Diskette komplett (*.*) übertragen, so sind die beiden Disketten anschließend normalerweise nicht identisch.

Der Befehl darf nicht bei einem Netzwerk-Laufwerk oder in Verbindung mit den Befehlen ASSIGN, JOIN und SUBST verwendet werden.

Kommandoaufbau (d:)(weg)diskcomp (s: (d:)) (/x)
 /x /1 vergleicht nur die erste Seite der Disketten
 /8 vergleicht nur die ersten 8 Sektoren (sonst 9)

Beispiel diskcomp a: b:

Die Disketten in Laufwerk A und B werden auf Gleichheit überprüft. Die Laufwerkangaben dürfen nicht gleich sein (also nicht diskcomp a: a:). Bei Ungleichheit erscheint eine Fehlermeldung nach der Prüfung. Beispiel:

Diskette ERSTE in Laufwerk A: einlegen

Diskette ZWEITE in Laufwerk B: einlegen

Wenn fertig, beliebige Taste drücken. . .

Nach Betätigung einer beliebigen Taste erscheint:

Vergleich von 40 Spuren

9 Sektoren pro Spur, 2 Seite(n)

Vergleich in Ordnung

Weitere Disketten vergleichen (J/N)?

Bei einem festgestellten Unterschied erscheint z.B. folgende Fehlermeldung:

Vergleichsfehler auf Seite 0, Spur 0

diskcomp

Es wird dabei nur mit einem Laufwerk gearbeitet (wenn Sie z.B. ein Festplattensystem mit nur einem Diskettenlaufwerk

haben, kann der Vergleichsvorgang trotzdem erfolgen). Sie werden in diesem Falle während der Programmausführung aufgefordert, das Original bzw. die Kopie abwechselnd einzulegen (Original in Speicher einlesen – Diskettenwechsel – Speicherinhalt mit Kopie vergleichen – Diskettenwechsel ...). Es wird hier immer mit dem aktuellen Laufwerk gearbeitet (am Prompt ersichtlich).

Diskette ERSTE in Laufwerk A: einlegen

Wenn fertig, beliebige Taste drücken. . .

Vergleich von 40 Spuren

9 Sektoren pro Spur, 2 Seite(n)

Wenn der Inhalt der Diskette gelesen wurde, erscheint:

Diskette ZWEITE in Laufwerk A: einlegen

Wenn fertig, beliebige Taste drücken. . .

Vergleich in Ordnung

Weitere Disketten vergleichen (J/N)?

Bemerkung Geben Sie nur eine Laufwerkbezeichnung ein, so bezeichnet dieses das Quellenlaufwerk (Original) und vergleicht es mit der Diskette im aktuellen Laufwerk (siehe Prompt am Bildschirm).

DISKCOPY *externer Befehl*

Zweck Der komplette Inhalt einer Diskette wird auf eine andere übertragen (Erstellung eines Disketten-Duplikates). Vor dem Kopiervorgang wird die neue Diskette formatiert, falls diese es noch nicht ist.

Es sind ein oder zwei Laufwerke notwendig. Die Originaldiskette sollte einen Schreibschutz haben, um Fehler zu vermeiden (Kopiervorgang in die verkehrte Richtung).

Mit dem Befehl kann nicht von oder nach einer Festplatte kopiert werden.

Der Befehl darf nicht bei einem Netzwerk-Laufwerk verwendet werden.

Kommandoaufbau `diskcopy (s: (d:)) (/x)`
`/x /1` Es wird nur eine Seite einer Diskette kopiert

Beispiele `diskcopy a: b:`

Die Diskette in Laufwerk A wird auf die Diskette in Laufwerk B übertragen.

`diskcopy`

Es wird dabei nur mit einem Laufwerk gearbeitet. Wenn Sie ein Festplattensystem mit nur einem Diskettenlaufwerk haben, kann der Kopiervorgang trotzdem erfolgen. Sie werden in diesem Falle während der Programmausführung aufgefordert, das Original bzw. die Zieldiskette einzulegen (Original in Speicher einlesen – Diskettenwechsel – Speicherinhalt auf Zieldiskette speichern ...).

Wenn also am Bildschirm der Prompt `A>` steht, so erfolgt der Kopiervorgang von A nach A, wenn `B>`, dann von B nach B.

QUELL Diskette in Laufwerk A: einlegen

Beliebige Taste drücken wenn fertig . . .

Kopiert: 40 Spuren

9 Sektoren/Spur, 2 Seite(n)

ZIEL Diskette in Laufwerk A: einlegen

Beliebige Taste drücken wenn fertig . . .

Sollte die Zieldiskette schreibgeschützt sein, so erhalten Sie eine Fehlermeldung:

Zieldiskette schreibgeschützt

Beliebige Taste drücken wenn fertig . . .

Sie können den Schreibschutz entfernen und die Zieldiskette wieder in das Laufwerk einlegen. Nach Betätigung einer beliebigen Taste wird der Kopiervorgang fortgesetzt.

Bemerkung Sollte sich mit DISKCOPY eine Diskette nicht kopieren lassen, so formatieren Sie mit dem Programm FORMAT eine neue und benutzen Sie den Befehl COPY.

ECHO *interner Befehl*

Zweck ECHO kontrolliert die Ausgabe von Befehls- und Kommentarzeilen während der Ausführung einer Batch-Datei (siehe Kapitel 12) bzw. zeigt einen Kommentar am Bildschirm an. Nach dem Einschalten des Computers ist ECHO standardmäßig eingeschaltet.

Kommandoaufbau

```
echo
echo (on)
echo (off)
echo (text)

on   Einschalten der Anzeigefunktion
off  Ausschalten der Anzeigefunktion
text Anzuzeigender Text
```

Beispiel Erstellen Sie eine Batch-Datei (siehe auch Kapitel 12) wie folgt:

```
copy con echotst.bat
```

Mit dem Befehl COPY können Sie die Batch-Datei ECHOTST.BAT eingeben

```
echo off
echo echo ist ausgeschaltet
echo Inhaltsverzeichnis Laufwerk A
dir a:
echo on
echo echo ist eingeschaltet
dir a:
```

Zum Speichern der Eingaben anschließend die Tasten F6 und <RETURN> betätigen.

Aufgrund der obigen Befehle wird am Bildschirm durch Aufruf von ECHOTST.BAT (nur den Befehl echotst und <RETURN> eingeben) folgendes angezeigt:

```
echo off
echo ist ausgeschaltet
Inhaltsverzeichnis Laufwerk A
```

Inhaltsverzeichnis des aktuellen Directory in Laufwerk A wird angezeigt

```
echo echo ist eingeschaltet  
echo ist eingeschaltet
```

Die erste Textzeile ist der Befehl, die zweite Zeile die vom Befehl am Bildschirm angezeigte Zeile

```
dir a:
```

Inhaltsverzeichnis des aktuellen Directory in Laufwerk A wird angezeigt

EDLIN *externer Befehl***Allgemeines und Zweck**

EDLIN ist das Dienstprogramm zur Herstellung und Änderung von Textdateien (und damit natürlich auch von Quell-Programmen). Es handelt sich dabei um einen zeilenorientierten Editor bzw. ein einfaches Textbearbeitungsprogramm (Maximallänge jeder Zeile ist 253 Zeichen). Die Zeilennummern werden vom Programm automatisch vergeben (1 bis 65529). Die Zeilennummern werden nicht auf der Diskette/Festplatte gespeichert.

Wird bei den Befehlen eine Zeilennummer angegeben, die größer ist als die größte vorhandene, so wird diese automatisch vom Programm in die Zeilennummer der Zeile nach der letzten vorhandenen geändert.

Mit EDLIN können nur 64 Kbyte Ihres Computer-Speichers genutzt werden. Davon werden ca. 75 % maximal mit Text gefüllt (beachten Sie die EDLIN-Befehle a und w).

Folgende Funktionen sind möglich:

- Erstellen und Speichern von neuen Texten
- Überarbeiten von bestehenden Texten (Dateien), alte und neue Fassung speichern
- Anzeigen, Ändern, Einfügen und Löschen von Zeilen
- Suchen, Ersetzen und Löschen von Zeichenketten in einer oder mehreren Zeilen

Kommandoaufbau

(d:)(weg)edlin dateiname (/x)

/x /b Wenn im Text das Zeichen Ctrl-Z vorkommt, muß dieser Parameter angegeben werden, da sonst nur bis zu diesem Zeichen Text bearbeitet werden kann (Ctrl-Z ist bei MS-DOS das Dateiende-Kennzeichen).

Bemerkungen

Wird ein Befehl während der Bearbeitung eines Textes falsch eingegeben, so erhalten Sie eine Fehlermeldung von EDLIN angezeigt:

Eintragsfehler

Geben Sie den Befehl EDLIN ohne den Dateinamen eines zu bearbeitenden Textes ein, so erhalten Sie die Fehlermeldung:

Dateiname muß angegeben werden

Sie können die Funktionstasten Ihrer Tastatur wie folgt bei der Textbearbeitung benutzen:

Taste	Funktion	Beschreibung
F1 oder →	kopiert ein Zeichen	kopiert ein Zeichen aus dem Tastaturpuffer in die neue Zeile
F2x	kopiert bis Zeichen x	kopiert alle Zeichen bis zum angegebenen Zeichen x aus dem Tastaturpuffer in die neue Zeile
F3	alle Zeichen kopieren	kopiert alle Zeichen des Tastaturpuffers in die neue Zeile
F4x	überspringen bis Zeichen x	ignoriert alle Zeichen aus dem Tastaturpuffer bis zum Zeichen x
F5	neuer Tastaturpuffer	schreibt eine neu eingegebene Zeile in den Tastaturpuffer
F6	Eingabeende	erzeugt ein Ctrl-Z-Zeichen (damit wird das Zeichen für Dateiende erzeugt)
F7	Nullzeichen	das Zeichen wird mit dem Wert 00Hexadezimal gespeichert
BackSpace oder ←	ein Zeichen löschen	ein Zeichen im Speicher und in der Eingabezeile löschen
Ins	einfügen	Umschalten auf Zeichen einfügen
Del	löschen	ein Zeichen im Tastaturpuffer löschen (nicht jedoch in der sichtbaren Eingabezeile)
Esc	Eingabe löschen	löscht die gesamte Eingabezeile, nicht jedoch den Tastaturpuffer

Befehlsübersicht zu EDLIN

#	Zeilennummer
#a	Nummer der ersten Zeile
#e	Nummer der letzten Zeile

Befehl (#)a

Zweck Laden von Zeilen der zu bearbeitenden Datei in den Speicher, bis dieser voll bzw. die Datei komplett geladen ist. Wenn schon ein Teil geladen ist, werden die hinzugeladenen Zeilen im Speicher zum bereits vorhandenen Text hinzugefügt. Wenn nur a angegeben wird, versucht das Programm den gesamten Text zu laden.

Befehl (#a),(#e),#n,(#)c

Zweck Kopieren von einer oder mehreren Zeilen an eine andere Stelle des Textes.

#a = erste zu kopierende Zeile
#e = letzte zu kopierende Zeile
#n = Zeilennummer, vor der der Text eingefügt werden soll
= wie oft die Zeile(n) kopiert werden sollen
c = Kopierbefehl

Befehl (#a)(, #e)d

Zweck Es werden eine oder mehrere Zeilen gelöscht. Wird nur d angegeben, so wird nur die aktuelle Zeile gelöscht.

Befehl #

Zweck Eine Zeile neu eingeben bzw. die aktuelle Zeilennummer auf die Zeile # setzen. Wenn Sie, statt die Zeile neu verändert einzugeben, nur <RETURN> drücken, bleibt die Textzeile unverändert.

Befehl e

Zweck Das Programm beenden und den Text speichern (der alte Text bleibt erhalten und erhält den Namenszusatz .BAK). Wenn nicht mehr ausreichend Platz auf der Diskette/Festplatte ist, wird nur soviel wie möglich gespeichert – der Rest ist verloren. Reicht der Speicherplatz auf der Diskette oder Festplatte zum Speichern nicht aus, so gehen Ihre Eingaben verloren und Sie erhalten folgende Fehlermeldung:

Diskette voll. Editierung verloren.

Befehl (#)i

Zweck Es werden eine oder mehrere neue Zeile(n) vor die angegebene Zeile eingefügt. Der »Einfügemodus« bleibt so lange bestehen, bis Ctrl-C oder Ctrl-Z und <RETURN> eingegeben wird. Immer erforderlich, wenn ein neuer Text erstellt wird. Wird der Befehl i ohne Zahl eingegeben, so wird der neu einzugebende Text immer am Dateiende bzw. ab der aktuellen Zeile eingefügt.

Befehl (#a)(, #e)l

Zweck Anzeigen von Zeilen (nur die im Speicher vorhandenen können angezeigt werden) am Bildschirm. Mit der Tastenkombination Ctrl-S kann gestoppt oder weiter angezeigt werden. Die aktuelle Zeile ist mit einem Stern (*) gekennzeichnet.

- Befehl** (#a),(#e),#m
- Zweck** Verschieben von Zeile(n) innerhalb des Textes vor die angegebene Zeile. Wird keine Anfangs-/Ende-Zeilenummer angegeben, so wird nur die aktuelle Zeile verschoben. Anschließend werden die Zeilen automatisch neu nummeriert.
- Befehl** (#a),(#e)p
- Zweck** Zeilen werden seitenweise aufgelistet. Nach Auflistung der Zeilen auf dem Bildschirm wird die letzte angezeigte Zeile zur aktuellen (im Gegensatz zum Befehl l).
- Befehl** q
- Zweck** Programm beenden, ohne den Text zu speichern. Zur Sicherheit fragt Sie das Programm EDLIN nach dem Befehl, ob der Befehl ausgeführt werden soll:
- Editieren abbrechen (J/N)?
- Befehl** (#a),(#e)(?)RzeichenkettelF6zeichenkette2
- Zweck** In den bezeichneten Zeilen wird die zeichenkettel durch die zeichenkette2 ersetzt. Wird das ? mit angegeben, so wird vor dem Austauschen des Textes die mit der zeichenkettel gefundene Zeile angezeigt und Sie werden befragt, ob Sie austauschen wollen.
- Befehl** (#a),(#e)(?)Szeichenkette
- Zweck** Suchen einer Zeichenkette in den bezeichneten Zeilen. Wird ? mit angegeben, so wird die gefundene Zeile angezeigt, und Sie werden befragt, ob es sich um die gesuchte handelt (wenn Sie mit NEIN antworten, sucht das Programm im weiteren Text, ob die Zeichenkette noch einmal vorkommt – der Vorgang kann so lange wiederholt werden, bis das Textende erreicht ist).
- Befehl** (#)Tdateiname
- Zweck** Kopiert eine Textdatei in den Text im Speicher ab der angegebenen Zeile. Nach dem Vorgang wird der Text im Speicher automatisch neu nummeriert. Wird die angegebene Datei nicht gefunden, so erhalten Sie die Fehlermeldung:
- Datei nicht gefunden

Befehl (#)w

Zweck Schreibt die angegebene Zeilenzahl auf die Diskette/Festplatte zurück und löscht diese dann im Speicher. Die noch verbleibenden Zeilen werden neu nummeriert. Dies ist bei Dateien notwendig, die zu groß sind, um komplett in den Speicher zu passen. Normalerweise werden anschließend mit Befehl a weitere Zeilen des Textes von der Diskette/Festplatte nachgeladen.

Wenn Sie statt einer Zeilennummer bei den Befehlen einen ».« verwenden, so bezeichnet dieser die Nummer der aktuellen Zeile (die aktuelle Zeile im Text kann am »*« zu Anfang einer Zeile erkannt werden). Ebenso kann für die Nummer der Zeile nach der letzten im Text vorhandenen das Zeichen »#« verwendet werden.

Beispiel/Übung Rufen Sie bitte das Programm EDLIN wie folgt auf:

```
edlin diskette
```

Daraufhin erscheint am Bildschirm

Neue Datei

EDLIN zeigt Ihnen damit an, daß der Text noch nicht auf der Diskette/Festplatte vorhanden ist. Wenn die Datei DISKETTE schon vorhanden wäre, erhielten Sie die Meldung

Ende der Eingabedatei

Tippen Sie den folgenden Text ein (Befehl i und <RETURN> und Text Zeile für Zeile eintippen):

Eine Diskette besteht aus einer 5.25 Inch großen runden Plastikscheibe, überzogen mit einer feinen magnetisierbaren Oxidschicht. Diese Scheibe wird in einer Hülle aufbewahrt, die sie fast vollständig bedeckt. Nur ein kleiner schmaler Schlitz für den Schreib-Lese-Kopf bleibt frei, der Schreib-Lese-Schlitz.

Die Scheibe selbst und auch die darauf gespeicherten Informationen sind sehr leicht zu zerstören. Sie sollten die Disketten immer sehr sorgfältig behandeln und folgendes beachten:

- Wenn Sie die Diskette aus ihrer Verpackung nehmen, dürfen Sie nie die Oberfläche berühren, vor allem nicht am Schreib-Lese-Schlitz.
- Ist die Diskette nicht im Laufwerk, so stecken Sie sie wieder in die Verpackung.
- Öffnen Sie die Klappe des Laufwerks erst, wenn die rote Kontroll-Lampe wieder erloschen ist.
- Achten Sie darauf, daß kein Staub oder irgendwelche Flüssigkeiten auf die Scheibe gelangen.
- Bewahren Sie die Diskette nicht in der Nähe magnetischer Felder auf. Ein Kugelschreiber kann schon Daten zerstören.
- Gehen Sie mit Disketten nicht durch Röntgenanlagen (z.B. sehr oft bei ausländischen Flughäfen).
- Lassen Sie keine Diskette im Laufwerk, wenn Sie den Strom des Computers ein- oder ausschalten. Durch eventuelle Stromstöße kann der Schreib-Lese-Kopf auf die Diskette »stürzen« (Head-Crash).
- Achten Sie darauf, daß die Disketten nicht zu heiß (über 52 Grad) oder zu kalt (unter 10 Grad) gelagert werden.
- Benutzen Sie zum Beschriften der Disketten niemals einen Kugelschreiber, sondern immer einen weichen Filzschreiber.
- Achten Sie darauf, daß Disketten niemals gebogen oder gestapelt werden.

Wollen Sie sich gegen unbeabsichtigtes Löschen von Dateien schützen, so verdecken Sie die Kerbe an der linken Seite der Diskette mit einem kleinen Streifen (werden bei neuen Disketten immer mitgeliefert). Die damit versehenen Disketten können nur noch gelesen werden.

Übungen zu den Befehlen

30, #, 1, c

Damit wird der Text ab Zeile 30 bis zum Textende (#) vor die Zeile 1 kopiert (die letzten Zeilen an den Textanfang kopieren).

1

und 2 mal <RETURN> betätigen

Die aktuelle Zeile auf die Zeilennummer 1 setzen. Nach dem ersten <RETURN> löscht jede Eingabe den alten Inhalt der Zeile (ausgenommen Sie betätigen wie im vorliegenden Beispiel sofort wieder <RETURN>).

.,6d

oder

,6d

Von der aktuellen Zeile bis zur Zeile 5 den Text löschen (die ersten 5 Zeilen). Wenn die erste die aktuelle Zeile ist, kann statt 1 auch der ».« verwendet werden.

Die aktuelle Zeile erkennen Sie am * zu Beginn einer Textzeile.

1

Es werden 23 Zeilen am Bildschirm angezeigt.

1,411

Es werden alle Zeilen bis zur 41. ab der ersten am Bildschirm angezeigt (es kann mit Ctrl+S gestoppt werden – weiter mit Ctrl+S). Maximal jedoch bis zur letzten im Text vorhandenen.

p

Am Bildschirm werden ab der aktuellen Zeile 23 weitere angezeigt. Wenn Sie erneut p eingeben, werden die folgenden 23 Zeilen am Bildschirm gelistet (der Befehl kann bis Textende wiederholt werden). Die Nummer der aktuellen Zeile wird durch jeden Befehl p verändert.

30,41,1m

Die Zeilen 30 bis 41 einschließlich werden vor die Zeile 1 im Text verschoben.

1, # ?RDisketteF6Disk

Ab der Zeile 1 bis zum Textende (#) wird der Text Diskette durch den Text Disk ersetzt (vor jedem einzelnen Austausch wird die Zeile angezeigt, in der Diskette gefunden wird, und Sie werden befragt, ob der Text in der Zeile ausgetauscht werden soll – es wird dabei die bereits geänderte Zeile angezeigt, damit Sie erkennen können, ob Sie mit der Änderung einverstanden sind).

1,41RDiskF6Diskette

Ab der Zeile 1 bis zur Zeile 41 wird der Text Disk durch den Text Diskette automatisch ersetzt. Die geänderten Zeilen werden am Bildschirm angezeigt.

e

Textbearbeitung beenden und den Text speichern (hier unter dem Namen DISKETTE). Die alte Version des Textes (vor Eingabe des Befehls EDLIN) wird unter dem Namen DISKETTE.BAK gespeichert.

Sie können den Text mit dem Befehl PRINT drucken.

EXE2BIN *externer Befehl*

Zweck EXE2BIN formt Programme mit dem Format .EXE in das Format .COM um. Ein Programm vom Typ .COM hat die Vorteile, daß es weniger Platz auf der Diskette/Festplatte belegt und schneller abläuft.
Bitte beachten Sie, daß nicht jedes Programm mit dem File-Typ EXE in eines mit dem Typ COM umgewandelt werden kann.

Kommandoaufbau (d:)(weg)exe2bin dateiname1 (d:)(dateiname2(.typ))
 dateiname1 Name des .EXE-Programmes
 d: Ziellaufwerk für das umgewandelte Programm
 dateiname2 Name des .COM-Programmes
 typ normalerweise .COM angeben (ohne = .BIN)

Beispiele exe2bin programm c:programm.com

Das Programm PROGRAMM.EXE wird in PROGRAMM.COM auf Laufwerk C umgewandelt.

exe2bin programm c:programm

Das Programm PROGRAMM.EXE wird in PROGRAMM.BIN auf Laufwerk C umgewandelt.

exe2bin programm

Das Programm PROGRAMM.EXE wird in PROGRAMM.BIN auf dem aktuellen Laufwerk im aktuellen Directory umgewandelt.

Bemerkungen Das Ursprungsprogramm muß in einem gültigen EXE-Format vorliegen. Der residente und der Datenteil des Programmes müssen kleiner als 64 Kbyte sein. Es darf kein sogenanntes STACK-Segment enthalten.

Wenn die Fehlermeldung

Datei kann nicht konvertiert werden

erscheint, liegt das Ursprungsprogramm nicht im gültigen Format vor.

Für weitere Angaben beachten Sie bitte die Angaben in den Handbüchern von Microsoft. Für die Arbeit mit diesem Programm sind fundierte Systemkenntnisse erforderlich.

EXIT *interner Befehl*

(Dieser Befehl gilt nur für MS-DOS in Verbindung mit BASICA und GW-BASIC; nicht für BASIC2!)

Zweck Dieses Kommando ermöglicht das Verlassen des COMMAND.COM-Prozessors und die Rückkehr zu einem vorherigen Level (z.B. zu einer Applikation/Programm), soweit einer existiert.

Kommandoaufbau exit

Beispiel Laden Sie BASIC (Programmiersprache/Interpreter):

```
basica
```

Wenn BASIC geladen ist, starten Sie den COMMAND-Prozessor (diesen Befehl gibt es nicht in jedem BASIC):

```
shell "COMMAND.COM"
```

Es erscheint daraufhin der MS-DOS-Prompt (z.B. C>) am Bildschirm, und Sie können beliebige MS-DOS-Befehle, wie z.B.

```
dir
```

eingeben. Damit wird in diesem Falle das Inhaltsverzeichnis am Bildschirm angezeigt. Durch Eingabe von

```
exit
```

kommen Sie wieder in den vorherigen Level, also zu BASIC zurück.

FDISK *externer Befehl*

Zweck Verfügt Ihr Schneider PC über eine Festplatte, so muß diese für MS-DOS als erstes vorbereitet werden. Erst nach der Bearbeitung mit FDISK kann diese für MS-DOS (oder DOS-Plus) verwendet werden.

Kommandoauflbau (d:)(weg)fdisk

Beispiel Geben Sie

fdisk

ein, um das Programm zu starten. Daraufhin erhalten Sie folgende Anzeige am Bildschirm:

Festplatten Initialisierungsprogramm Version 0.01
(C)Copyright Microsoft, 1985.

FDISK Optionen

Wählen Sie bitte:

1. DOS Partition erstellen
2. Aktive Partition ändern
3. DOS Partition löschen
4. Daten der Partition anzeigen

Geben Sie die Ziffer ein:[1]

Drücke ESC um zu DOS zurückzukehren

Geben Sie die Ziffer 1 ein und betätigen Sie <RETURN>.

Bemerkungen Wenn die Festplatte noch nicht für MS-DOS vorbereitet war, so werden Sie befragt, ob die gesamte Kapazität für das Betriebssystem eingeteilt werden soll.

Sollte bereits eine DOS-Partition vorhanden sein, so muß diese erst gelöscht werden, bevor die Festplatte neu eingeteilt werden kann.

Nach der Vorbereitung der Festplatte startet Ihr Schneider PC neu (dazu muß die MS-DOS-Diskette im Diskettenlaufwerk vorhanden sein).

Beispiel Starten Sie das Programm FDISK erneut und wählen Sie den Punkt 2. Dazu erscheint z.B.:

Aktive Partition wechseln

Partition	Status	Typ	Start	Ende	Größe
1	A	DOS	0	304	305
2	N	xxx	xxx	xxx	xxx
3	N	xxx	xxx	xxx	xxx
4	N	xxx	xxx	xxx	xxx

Die Platte hat 305 Zylinder.

Die momentan aktive Partition ist 1.

Geben Sie die Nummer der Partition ein
die Sie aktivieren möchten: []

Drücke ESC um ins Menü zurückzukehren

Bemerkungen Es werden Ihnen alle Partitions angezeigt und Sie können wählen, welche Sie aktiv setzen wollen (es kann immer nur eine aktiv sein!). Wie Sie aus obiger Anzeige ersehen können, ist es möglich, bis zu vier Partitions einzurichten. Zu beachten ist dabei, daß immer nur mit der aktiven gearbeitet werden kann.

Beispiel Sie können eine DOS-Partition auch löschen. Wählen Sie Punkt 3 im Menü:

DOS Partition wird gelöscht

Partition	Status	Typ	Start	Ende	Größe
1	A	DOS	0	304	305
2	N	xxx	xxx	xxx	xxx
3	N	xxx	xxx	xxx	xxx
4	N	xxx	xxx	xxx	xxx

Die Platte hat 305 Zylinder.

Warnung! Daten in der DOS Partition
werden gelöscht. Wollen Sie
weitermachen.....? [N]

Drücke ESC um ins Menü zurückzukehren

Bemerkungen Wenn Sie die aktive DOS-Partition löschen, kann nicht mehr von der Festplatte gestartet werden, und alle gespeicherten Daten sind unwiederbringlich verloren.
Wenn Sie im Menü den Punkt 4 anwählen, wird Ihnen der momentane Stand der Festplatteneinteilung angezeigt.
Nachdem mit FDISK die Festplatte vorbereitet wurde, muß diese noch formatiert werden (Befehl FORMAT). Erst dann ist die Platte fertig, um mit ihr arbeiten zu können.

FIND *externer Befehl*

Zweck FIND sucht in einer oder mehreren Dateien nach einer angegebenen Zeichenfolge und zeigt jede Zeile mit einer entsprechend gefundenen Zeichenkette am Bildschirm an. Bei FIND handelt sich um einen Filter (siehe Kapitel 9).

Kommandoaufbau (d:)(weg)find (/x) "zeichenkette" (dateiname ...)

/x /c zähle Zeilen (es wird nur angezeigt, wie oft die angegebene Zeichenkette in einer Datei gefunden wurde)
/n Anzeigen der Zeilennummer vor jeder gefundenen Zeile
/v Anzeigen aller Zeilen, in denen die Zeichenkette nicht vorkommt

Testdaten Dateiname KAP1

Der Kurs MS-DOS-Einführung wird auf Personal-Computer-Systemen durchgeführt. Dabei gelangen Computer der Baureihe SCHNEIDER PC zum Einsatz. Die Computer der Baureihe SCHNEIDER PC sind vollkompatible Modelle.

Dateiname KAP2

Der Hersteller bietet zum Marktstandard kompatible Computer (SCHNEIDER PC) an.

Dateiname KAP3

Für Personal Computer der kompatiblen Klasse stehen am Weltmarkt eine sehr große Anzahl von Programmen zur Verfügung.

Der Hersteller bietet seinen Kunden hier ein System an (SCHNEIDER PC).

Beispiele find /n "SCHNEIDER PC" kap1 kap2 kap3

Es werden die Zeilen mit der Zeichenkette »SCHNEIDER PC« in den drei Texten gesucht. Dabei soll bei der Ausgabe die Zeilennummer mit angezeigt werden (die Zeichenkette wird im dritten Text nicht gefunden, da diese auf 2 Zeilen verteilt ist). Am Bildschirm erfolgt folgende Anzeige:

----- kap1

[3]BAUREIHE SCHNEIDER PC ZUM EINSATZ.

[4]DIE COMPUTER DER BAUREIHE SCHNEIDER PC SIND VOLL
KOMPATIBLE

----- kap2

[2](SCHNEIDER PC) AN.

----- kap3

dir | find /n "DAT" | sort > prn

Es sollen nur Dateien angezeigt werden, die im Namen die Zeichenkette DAT nicht haben. Dabei wird das Inhaltsverzeichnis nach Dateinamen aufsteigend sortiert ausgegeben. Die Ausgabe soll am Drucker erfolgen. Am Drucker erscheint:

[30]SORT DAT 3228 18.07.85 11.32

[33]DATE TXT 1024 16.07.85 22.43

find /n /v "SCHNEIDER PC" kap1 kap2 kap3

Es werden die Zeilen mit der Zeichenkette »SCHNEIDER PC« in den 3 Texten gesucht. Dabei soll bei der Ausgabe die Zeilennummer mit angezeigt werden (die Zeichenkette wird im dritten Text nicht gefunden, da diese auf 2 Zeilen verteilt ist). Es sollen nur die Zeilen angezeigt werden, in denen die Zeichenkette nicht vorkommt. Am Bildschirm erfolgt folgende Anzeige:

----- kap1

[1]DER KURS MS-DOS-EINFÜHRUNG WIRD AUF PERSONAL-
COMPUTER-

[2]SYSTEMEN DURCHGEFÜHRT. DABEI GELANGEN COMPUTER
DER

[5]MODELLE.

----- kap2

[1]DER HERSTELLER BIETET ZUM MARKTSTANDARD
KOMPATIBLE COMPUTER

----- kap3

[1]FÜR PERSONAL COMPUTER DER KOMPATIBLEN KLASSE
STEHEN AM WELT-

[2]MARKT EINE SEHR GROßE ANZAHL VON PROGRAMMEN ZUR
VERFÜGUNG.

[3]DER HERSTELLER BIETET SEINEN KUNDEN HIER EIN
SYSTEM AN (SCHNEIDER

[4]PC).

FOR *interner Befehl*

Zweck Wiederholt mehrmals einen MS-DOS-Befehl.

Kommando Aufbau Bei Eingabe am Bildschirm:
for %variable in (wert) do befehl
Bei Verwendung in Batch-Dateien:
for %%variable in (wert) do befehl
variable Darf jedes Zeichen (Buchstabe) sein, außer 0 bis 9
wert Dateinamen oder Zeichenketten
befehl MS-DOS-Befehl

Beispiele for %a in (*.bat) do type %a > prn

Im ersten Teil des Befehls wird %a definiert. Dann werden alle Dateien mit dem File-Typ BAT herausgesucht. Anschließend werden der Reihe nach alle BAT-Dateien mit type %a (%a wird durch die Namen der gefundenen Dateien ersetzt) auf den Drucker (> prn) ausgegeben (und zwar mit ihrem Inhalt).

```
A>for %z in (config.sys autoexec.bat) do type %z
```

```
A>type autoexec.bat
```

```
A>type config.sys
```

Dabei werden die Dateien CONFIG.SYS und AUTOEXEC.BAT mit ihrem Inhalt der Reihe nach auf dem Bildschirm angezeigt.

Bemerkungen Wenn der Befehl FOR nicht in einer Batch-Datei verwendet wird, darf nicht %%%, sondern nur % angegeben werden.

FORMAT *externer Befehl*

Zweck FORMAT bereitet eine Diskette oder Festplatte (nach FDISK) zur Aufnahme von MS-DOS (bzw. Dateien). Es wird das Inhaltsverzeichnis initialisiert, die sogenannte »File-Allocation-Table« angelegt und der »Boot-Loader« kopiert. Disketten werden vom Hersteller immer unformatiert zur Verfügung gestellt, d.h. der Computer kann diese Disketten nicht verarbeiten.

FORMAT zerstört alle eventuell vorhandenen Daten unwiederbringlich auf einer Diskette!!

Nach dem Formatieren kann einer Diskette oder Festplatte ein Name zugeordnet werden (ein sogenanntes Label).

Beachten Sie zu diesem Befehl auch die Kapitel 7 und 8 – hier werden die Aufteilung und das Inhaltsverzeichnis näher beschrieben.

Den Befehl nicht bei einem Netzwerk-Laufwerk verwenden.

Kommandoaufbau (d:)(weg)format d: (/x...)

/x /b erstellt Freiraum für das MS-DOS-Betriebssystem und (IO.SYS und MSDOS.SYS) den COMMAND-Processor (mit dem Befehl SYS kann später das System übertragen werden)

/8 nur mit 8 Sektoren pro Spur formatieren (normal 9)

/1 nur einseitig formatieren statt normalerweise doppelseitig

/s nach dem Formatieren wird automatisch MS-DOS kopiert (IO.SYS, MSDOS.SYS und COMMAND.COM) – damit kann der PC von der neuen Disk gestartet werden (bootbar). Diesen Parameter immer beim Formatieren einer Festplatte angeben.

/v nach dem Formatieren einer Diskette/Festplatte werden Sie noch nach dem Namen für die Disk befragt (dieser darf maximal 11 Zeichen lang sein – beachten Sie auch den Befehl LABEL)

Beispiele format

Sie erhalten die Fehlermeldung:

Laufwerksbuchstabe muß angegeben werden

format a:

Damit wird das Programm gestartet, und Sie erhalten folgende Anzeige:

Neue Diskette für Laufwerk A: einlegen
und ENTER drücken wenn fertig

Wenn die entsprechende neue Diskette eingelegt wurde, betätigen Sie <RETURN>. Weiter wird angezeigt:

Kopf: 0 Zylinder: 0

Nach Beendigung der Formatierung erscheint:

Formatierung beendet

362496 Bytes insgesamt auf Diskette
362496 Bytes verfügbar auf Diskette

Nochmal formatieren (J/N)?

format a: /s

Die Diskette in Laufwerk A wird formatiert und MS-DOS übertragen (IO.SYS, MSDOS.SYS und COMMAND.COM).

format a: /s /v

Wie vorheriges Beispiel, nur wird hier nach dem Formatierungsvorgang noch nach dem Namen für die Diskette gefragt.

format c: /s /v

Die Festplatte C formatieren, MS-DOS (IO.SYS, MSDOS.SYS und COMMAND.COM) übertragen und die Eingabe eines Namens ermöglichen.

Bemerkungen

Ohne weitere Angaben wird eine Diskette immer in dem Format eingerichtet, das das Laufwerk besitzt (normalerweise bei Ihrem Schneider PC 40 Spuren je 9 Sektoren mit 360 Kbyte Kapazität). Dies kann durch die Parameter /n und /t beeinflusst werden. Wenn ein Laufwerktyp dies nicht zuläßt, erhalten Sie die Fehlermeldung

Parameter nicht unterstützt

Stellt FORMAT während des Laufs fest, daß Teile der Diskette oder Festplatte nicht in Ordnung sind, so werden diese »ausgeblendet« (damit können diese Teile nicht mehr verwendet

werden). Sie erhalten anschließend die Angabe, wie groß diese Bereiche sind.

Beispiel 1024 Bytes in schlechten Sektoren

FORMAT beendet mit folgenden Ausgangscodes (beachte Befehl IF ERROR LEVEL):

- 0 fehlerlos formatiert
- 3 durch Sie abgebrochen (Ctrl+Break oder Ctrl+C)
- 4 gravierender Fehler
- 5 Formatierung der Festplatte abgebrochen

GOTO *interner Befehl*

Zweck GOTO führt einen unbedingten Sprungbefehl in einer Batch-Datei aus. Kann nur innerhalb einer Batch-Datei verwendet werden (siehe Kapitel 12).

Kommando Aufbau goto marke

Beispiel Eine Batch-Datei wird mit dem Befehl

copy con da.bat

erstellt und enthält folgende Befehle:

```
if exist %1 goto xyz
```

```
goto yz
```

```
:xyz
```

```
echo Die Datei %1 ist vorhanden
```

```
goto ende
```

```
:yz
```

```
echo Die Datei %1 ist nicht vorhanden
```

```
:ende
```

Wenn Sie die Batch-Datei mit

```
da command.com
```

ausführen, erhalten Sie folgende Meldungen am Bildschirm:

```
if exist command.com goto xyz
```

```
:xyz
```

```
echo Die Datei command.com ist vorhanden
```

```
goto ende
```

```
:ende
```

Wenn Sie die Batch-Datei mit

```
da nicht.com
```

ausführen, erhalten Sie folgende Meldungen am Bildschirm :

```
if exist nicht.com goto xyz
```

```
goto yz
```

```
:yz
```

```
echo Die Datei nicht.com ist nicht vorhanden
```

```
:ende
```

Bemerkungen Beachten Sie zu vorstehendem Beispiel auch die Befehle IF und ECHO (siehe auch Kapitel 12 »Die BATCH-Befehle und ihre Anwendung«).

GRAFTABL *externer Befehl*

Zweck Lädt die Tabelle der ASCII-Zeichen (Werte 128 bis 255) in den Speicher für die Benutzung im Grafik-Modus (Farbgrafik-Karte muß im PC installiert sein).

Kommandoaufbau (d:)(weg)graftabl

Bemerkungen Dieser Befehl ist vor allem notwendig, wenn Sie einen anderen als den Original-Zeichensatz (für USA) verwenden (z. B. für den deutschen KEYBGR). Nur wenn dieser Befehl eingegeben wurde, werden z. B. die deutschen Umlaute im Grafik-Modus (MODE-Befehl beachten) richtig dargestellt. Der Befehl muß bei jedem Neustart des Computers erneut eingegeben werden.

GRAPHICS *externer Befehl*

Zweck GRAPHICS dient zur richtigen Ausgabe von Grafikzeichen und Sonderzeichen des Bildschirms auf den Drucker (Hard-copy-Funktion mit Shift-PrtSc).

Kommando Aufbau (d:)(weg)graphics (printer-typ) (/x)

printer-typ	color1	Farbdrucker mit schwarzem Farbband
	color4	Farbdrucker mit vierfarbigem Farbband
	color8	Farbdrucker mit CMY vierfarbigem Farbband
	compact	Compact-Drucker
	graphics	Grafik-Drucker (Standarddrucker bzw. DMP3000 von Schneider)
	colorjet	Tintenstrahlfarbdrucker
/x	/r	Es wird Schwarz als Schwarz und Weiß als Weiß gedruckt (wenn nicht angegeben, wird Schwarz als Weiß und Weiß als Schwarz gedruckt)
	/b	Die Hintergrundfarbe wird gedruckt (nur bei COLOR4 und COLOR8) – wenn nicht angegeben, wird die Hintergrundfarbe nicht gedruckt
	/f	Bildschirm Ausdruck um 90 Grad drehen (bei Grafik)
	/c	Bildschirm Ausdruck auf der Seite zentrieren (bei Grafik)

IF *interner Befehl*

Zweck IF macht eine Befehlsausführung von der Erfüllung einer Bedingung abhängig. Dieser Befehl wird vor allem in Batch-Dateien (siehe Kapitel 12) verwendet.

Kommando Aufbau if (not) bedingung befehl
bedingung ERROR LEVEL nummer
zeichenkette1 = = zeichenkette2
exist dateiname

Beispiele Erstellt wird folgende Batch-Datei (Befehl: copy con such.bat):

```
echo off
chkdsk %1:
if errorlevel 0 echo Der Ausgangscode des
Programmes ist 0.
```

Aufruf der Batch-Datei erfolgt z.B. mit

```
such c
```

Wenn der Fehlercode des Programmes chkdsk 0 ist, wird folgendes am Bildschirm angezeigt:

```
echo off
Der Ausgangscode des Programmes ist 0.
```

Wenn der Fehlercode von chkdsk nicht 0 ist, wird nur der Text echo off am Bildschirm angezeigt.

Erstellt wird folgende Batch-Datei (Befehl: copy con loesche.bat):

```
echo off
if "%1"==" " goto ende
del %1
echo Die Datei %1 wurde gelöscht.
:ende
```

Wenn die Batch-Datei mit

```
loesche da.bat
```

aufgerufen wird, erscheinen folgende Anzeigen am Bildschirm:

echo off

Die Datei da.bat wurde gelöscht.

Bemerkung Die ERRORLEVEL-Nummern (Ausgangscodes) finden Sie im MS-DOS-Handbuch zu Ihrem Schneider PC oder bei einzelnen Befehlen in diesem Buch.

JOIN *externer Befehl*

Zweck Temporär eine Laufwerkbezeichnung mit einem Sub-Directory vertauschen (wenn ein definiertes Sub-Directory angesprochen wird, verwendet das System die Daten im Laufwerk).

Der Befehl darf nicht mit Netzwerk-Laufwerken verwendet werden.

Die verwendete Laufwerkbezeichnung kann nicht verwendet werden, solange JOIN wirksam ist.

Wenn JOIN wirksam ist, dürfen folgende Befehle nicht verwendet werden:

BACKUP

DISKCOMP

DISKCOPY

RESTORE

Bei JOIN dürfen Laufwerke, die mit ASSIGN oder SUBST umgeleitet wurden, nicht benutzt werden.

Kommandoaufbau (d:)(weg)join
(d:)(weg)join d: (d:)weg
(d:)(weg)join d: /d

Beispiele join a: c:\drivea

Der Laufwerkbezeichnung A wurde der Sub-Directory-Name drivea zugewiesen. Wenn anschließend das Sub-Directory angesprochen wird, so erfolgt der Zugriff auf das Laufwerk A. Das Laufwerk A kann nicht mehr mit der Bezeichnung A angesprochen werden. Der Zugriff auf das Laufwerk erfolgt z.B. mit

dir \drivea

Damit wird das Inhaltsverzeichnis von Laufwerk A angezeigt.

join

Anzeigen der umgeleiteten Laufwerksnamen :

A: = > C:\DRIVEA

join a: /d

Der ursprüngliche JOIN-Befehl wird zurückgesetzt und damit der Originalzustand wieder hergestellt. Das Laufwerk A wieder mit a: angesprochen. Der Sub-Directory-Eintrag drivea bleibt bestehen, wird aber wie jeder andere behandelt.

Bemerkung Wenn Sie die Fehlermeldung für nicht ausreichenden Speicher erhalten, verringern Sie die Anzahl der BUFFER in CONFIG.SYS und starten Sie Ihren PC erneut (siehe dazu auch das Kapitel 12). Eine weitere Möglichkeit besteht darin, mit COMMAND den Environment-Speicher zu vergrößern. Sollte die Fehlermeldung immer noch erscheinen, so kann der Befehl nicht ausgeführt werden.

Das mit JOIN verwendete Sub-Directory muß leer sein oder noch nicht existieren (in diesem Fall legt JOIN das Sub-Directory neu an).

KEYBGR *externer Befehl*

Zweck Laden einer anderen Tastaturbelegung (für verschiedene Länder/Sprachen) – nach dem Systemstart ist die Tastatur mit Standard-ASCII (USA) belegt.

Kommando Aufbau (d:)(weg)keybgr
Von Schneider sind die Zeichensatzprogramme für folgende Länder lieferbar:

KEYBUK	=	Großbritannien
KEYBGR	=	Deutschland
KEYBFR	=	Frankreich
KEYBIT	=	Italien
KEYBSP	=	Spanien

Bemerkungen Wenn eine Tastaturbelegung geladen wird, so benötigt diese permanent 2 Kbyte zusätzlich an Speicherplatz. Durch Laden eines Zeichensatzes für ein anderes Land wird der vorher geladene Zeichensatz im Speicher gelöscht.

Wenn Sie die Tasten Ctrl-Alt-F1 betätigen, so wird auf den Zeichensatz für USA umgeschaltet, ohne daß dieser geladen werden muß. Zurückschalten auf den geladenen Zeichensatz (Deutsch) Ctrl-Alt-F2.

LABEL *externer Befehl*

Zweck Erstellen, Ändern oder Löschen eines Namens für eine Diskette oder Festplatte.

Kommandoauflbau (d:)(weg)label
(d:)(weg)label (d:)(name)

Beispiele Zum erstenmal werden Sie beim Formatieren nach dem Namen zu einer Disk befragt (wenn Sie zu FORMAT den Parameter /v angegeben haben). Wenn Sie statt einen Namen einzugeben nur <RETURN> drücken, wird kein Name vergeben. Der Name einer Disk darf maximal 11 Zeichen lang sein.

label a:

Es wird am Bildschirm z.B. angezeigt:

Volume in Laufwerk A ist 46001G

Volume Name (11 Zeichen, ENTER = keiner)?

Wenn Sie nur <RETURN> drücken, erscheint:

Aktuellen Volume Namen löschen (J/N)?

Wenn Sie den Namen löschen wollen, so drücken Sie nur den Buchstaben j und <RETURN>.

label a:neu-name

Die Diskette in Laufwerk A erhält den neuen Namen NEU-NAME.

Bemerkungen Der Befehl darf nicht für Laufwerke verwendet werden, zu denen die Befehle ASSIGN oder SUBST angewendet wurden. Ferner darf er nicht für ein Netzwerk verwendet werden.

MKDIR oder **MD** *interner Befehl*

Zweck Mit diesem Befehl kann ein neues Sub-Directory angelegt werden (siehe auch Kapitel 8).

Kommandoaufbau `mkdir (d:)weg`
`md (d:)weg`

Beispiele `mkdir \text`

`oder`

`mkdir text`

`oder`

`md text`

Es wird ein neues Sub-Directory mit dem Namen TEXT angelegt (der Eintrag ist im aktuellen Directory sichtbar). Dazu werden das eingegebene Datum und die Uhrzeit gespeichert. Ferner werden im neuen Sub-Directory die Einträge

- `.` dahinter verbirgt sich der Name des neuen Directory
- `..` dahinter verbirgt sich der Name des hierarchisch vorher liegenden Directory

angelegt.

Mit dem Befehl befinden Sie sich nicht automatisch im neuen Sub-Directory (dazu ist der Befehl CHDIR bzw. CD notwendig).

`md \text\programm`

Es wird ein neues Sub-Directory mit dem Namen PROGRAMM im Sub-Directory TEXT angelegt (TEXT muß ein Sub-Directory-Eintrag im aktuellen Directory sein).

`md .`

`und`

`md ..`

Mit diesen beiden Befehlen erhalten Sie die Fehlermeldung:

`Verzeichnis kann nicht erstellt werden`

Diese Namen können bei dem Befehl MKDIR nicht verwendet werden.

MODE *externer Befehl*

Zweck MODE bestimmt die Betriebsart für den Bildschirm, monochrom oder grafisch; bestimmt das Protokoll für den parallelen oder seriellen Ausgang; Ausgabe vom parallelen zum seriellen Ausgang umleiten.
 Wenn Sie MODE nach jedem Start Ihres Computers benötigen, sollten Sie diesen Befehl in die Datei AUTOEXEC.BAT einfügen (siehe Kapitel 12).

Kommando Aufbau (d:)(weg)mode lpt #: (n) (, (m)) (, (p))

Parallelen Ausgang konfigurieren

- # Nummer des parallelen Ausgangs (1-3, falls vorhanden)
- n Zeichen pro Zeile (80 oder 132)
- m Zeilen pro Inch (6 oder 8)
- p Drucken wiederholt versuchen, d.h. gegebenenfalls warten, bis Drucker eingeschaltet ist (kann nur mit Ctrl-Break beendet werden)

Beispiele mode lpt2:132,,p

Der Drucker am 2. parallelen Ausgang soll bis 132 Zeichen pro Zeile und 6 Zeilen pro Inch drucken. Ferner soll das System gegebenenfalls warten, bis der Drucker bereit ist (keine Fehlermeldung oder Abbruch, wenn der Drucker z. B. nicht auf ON LINE eingeschaltet ist).

mode lpt2:

Der Drucker am 2. parallelen Ausgang wird umgestellt auf 80 Zeichen pro Zeile und 6 Zeilen pro Inch. Ferner erfolgt Abbruch, wenn der Drucker nicht bereit ist (Ausgang auf Standardwerte zurückstellen).

mode (n)(, (m))(, (t))(, (s))

Bildschirm konfigurieren

n bezeichnet den Bildschirm-Modus :

- 40 40 Zeichen pro Zeile
- 80 80 Zeichen pro Zeile (Standardwert)
- BW40 40 Zeichen pro Zeile und Schwarz/Weiß-Modus
- BW80 80 Zeichen pro Zeile und Schwarz/Weiß-Modus
- CO40 40 Zeichen pro Zeile und Farb-Modus
- CO80 80 Zeichen pro Zeile und Farb-Modus

m	R	Bildschirm nach rechts verschieben
	L	Bildschirm nach links verschieben
t		ruft eine Testzeile zur Bildschirmjustage auf – kann nur zusammen mit m verwendet werden. Sie können nach dem Aufruf die Anzeige am Bildschirm je nach Eingabe nach rechts bzw. links verschieben.

mode co40

Der Bildschirm wird auf 40 Zeichen pro Zeile umgeschaltet. Damit erscheint die Schrift größer.

mode COM#:baud-rate(,(parity)(,(daten-länge)
(,(stop-bit)(,(p))))

	Seriellen Ausgang konfigurieren
#	Nummer des seriellen Ausgangs (1 oder 2, falls vorhanden)
baud-rate	Übertragungsgeschwindigkeit (110, 150, 300, 600, 1200, 2400, 4800 oder 9600)
parity	bestimmt die Art der Parity (odd (O), even (E) oder no parity (N))
daten-länge	(7 oder 8 Bit)
stop-bit	(1 oder 2 – Standardwert 2 bei 110 Baud, sonst 1)
p	Ausgabe wiederholt versuchen, auch wenn keine Antwort kommt (kann dann nur mit Ctrl-Break beendet werden)

mode com2:9600,e,8,2

Der serielle Ausgang COM2 wird auf eine Übertragungsgeschwindigkeit von 9600 Baud, even parity, 8 Bit und 2 Stop-Bit eingestellt.

mode com2:300,,,p

Der serielle Ausgang COM2 wird auf eine Übertragungsgeschwindigkeit von 300 Baud, even parity, 7 Bit und 1 Stop-Bit eingestellt. Ferner wird eine Übertragung nicht abgebrochen, wenn der Empfänger der Daten nicht empfangsbereit ist.

mode lpt#:=com#

Leitet Daten vom parallelen zum seriellen Ausgang um (LPT1 bis LPT3 und COM1 bis COM2)

Bevor Sie diesen Befehl verwenden, konfigurieren Sie bitte den seriellen Ausgang (der Parameter *p* ist unbedingt mit anzugeben). Um diese Umleitung wieder aufzuheben, muß nur der parallele Ausgang neu konfiguriert werden.

MORE *externer Befehl*

Zweck Mit MORE können Daten seitenweise auf den Bildschirm ausgegeben werden. Bei dem Befehl handelt es sich um einen Filter (siehe Kapitel 9).

Kommando Aufbau (d:)(weg)more
(d:)(weg)befehl | more

Beispiele type test | more

Die Datei TEST wird am Bildschirm angezeigt – nach jeder vollen Bildschirmseite stoppt die Ausgabe, bis Sie eine beliebige Taste betätigt haben (in der letzten Zeile am Bildschirm erscheint – More –).

```
more < test
```

Mit diesem Befehl erreichen Sie den gleichen Effekt wie vorher. Diese Form der Anwendung ist effektiver, da die Daten nicht wie im ersten Beispiel von MS-DOS zwischengespeichert werden (siehe Kapitel 9). Der Ablauf erfolgt schneller.

```
dir | sort | more
```

Mit DIR wird das Inhaltsverzeichnis des aktuellen Laufwerks ausgegeben. Dieses wird auf der Diskette bzw. Festplatte gespeichert, ohne daß es angezeigt wird. Anschließend wird es aufsteigend sortiert und mit MORE seitenweise am Bildschirm angezeigt.

PATH *interner Befehl*

Zweck PATH benennt die Wege, die zu einem Verzeichnis führen, in dem ein externer Befehl gesucht werden soll. Ein Befehl, der in einem anderen als dem gerade aktuellen Directory steht, wird bei Eingabe gefunden, wenn das Directory, in dem der Befehl steht, mit PATH zugewiesen wurde.
Beachten Sie zu diesem Befehl auch das Kapitel 8.

Kommandoaufruf path (d:)(weg;(d:)weg)...

Beispiele path

Zeigt die aktiven Vereinbarungen mit PATH an. Beispiel:

```
PATH=\MSDOS;\AUFTRAG\FIBU;\TEXT
```

```
path ;
```

Löscht alle mit PATH getroffenen Vereinbarungen.

```
path c:\system;a:\sub\texte
```

Damit können die Befehle des Sub-Directory system in Laufwerk C und texte (Sub-Directory von sub) in Laufwerk A in jedem anderen Directory verwendet werden.

Wenn Sie die Fehlermeldung erhalten, daß nicht ausreichend Speicher vorhanden ist, so können Sie mit COMMAND den Environment-Speicher vergrößern. Beispiel:

```
command /e:1024
```

Damit wird der Speicher auf 1 Kbyte vergrößert (maximal 32768). Ohne diesen Befehl beträgt die Standardgröße 128 Byte.

PAUSE *interner Befehl*

Zweck PAUSE unterbricht die Ausführung einer Batch-Datei (siehe Kapitel 12) und gibt einen Text am Bildschirm aus, mit dem Sie aufgefordert werden, zur Fortsetzung der Arbeit eine beliebige Taste zu drücken. Die Abarbeitung einer Batch-Datei kann durch Betätigung von Ctrl-Break abgebrochen werden – Sie werden noch befragt, ob Sie wirklich abbrechen wollen:

Stapel Job beenden (J/N)?

Kommandoaufbau pause (text)

Beispiel Folgende Befehle in einer Batch-Datei

```
echo off
```

```
echo Legen Sie in das Laufwerk A eine neue  
Diskette ein
```

```
pause Durch Eingabe der Tastenkombination  
<Ctrl-Break> können Sie abbrechen
```

erzeugen am Bildschirm die Anzeigen (echo ist eingeschaltet):

```
Legen Sie in das Laufwerk A eine neue Diskette ein  
Durch Eingabe der Tastenkombination <Ctrl-Break>  
können Sie abbrechen
```

```
Wenn fertig, eine Taste betätigen . . .
```

PRINT *externer Befehl*

Zweck Mit PRINT kann eine oder mehrere Textdateien ausgedruckt werden, während ein anderes Programm bearbeitet wird.

Kommandoaufruf (d:)(weg)print
(d:)(weg)print dateiname1 (/x) (dateiname2 (/x) ...)

- /x /b: # Definition des Puffers von PRINT. Der Standardwert (ohne Angabe dieses Parameters) ist 512 Byte. Die maximale Angabe ist 16384 Byte. Nur beim ersten Aufruf von PRINT angeben. Je größer der Wert ist, um so besser ist die Leistung von PRINT.
- /c Damit werden die angegebenen Dateien aus der Tabelle der noch zu druckenden Dateien entfernt.
- /d:dev Normalerweise wird alles am Drucker (Geräte-name PRN) ausgegeben. Hier können Sie einen anderen Gerätenamen (dev) angeben (z.B. COM1 oder CON). Dieser Parameter muß als erstes angegeben werden.
- /m: # Spezifiziert, wieviel Zeit (Takte) PRINT für den Druck verwenden darf (bis eine weitere von Ihnen gleichzeitig durchgeführte Tätigkeit kommt). Der Standardwert beträgt 2. Angaben von 1 bis 255 sind erlaubt. Nur beim ersten Aufruf von PRINT angeben.
- /s: # Spezifiziert, wieviel Zeit (Takte) während der Arbeit von PRINT für die andere von Ihnen gleichzeitig durchgeführte Tätigkeit zur Verfügung steht. Der Standardwert ist 8. Es kann ein Wert zwischen 1 und 255 angegeben werden. Nur beim ersten Aufruf von PRINT angeben.
- /p Fügt die angegebenen Dateien zu den zu druckenden hinzu.
- /q: # Spezifiziert, wie viele Dateien Sie in die PRINT-Warteschlange schreiben können. Der Standardwert ist 10 (möglich ist ein Wert von 1 bis 32). Nur beim ersten Aufruf von PRINT angeben.
- /t Ende – löscht alle Dateien aus der Warteschlange von PRINT.

/u: # Setzt die PRINT-Wartezeit. Spezifiziert die Anzahl der Takte, bis der Drucker frei ist. Währenddessen wird die von Ihnen gleichzeitig durchgeführte Tätigkeit bearbeitet. Der Standardwert ist 1 (Werte von 1 bis 255 sind erlaubt). Nur beim ersten Aufruf von PRINT angeben.

Beispiele `print config.sys autoexec.bat`

Die Dateien CONFIG.SYS und AUTOEXEC.BAT sollen gedruckt werden. Es erscheint folgende Anzeige:

Name des Ausgabegerätes [PRN]:
Residenter Teil von PRINT installiert

A:\CONFIG.SYS wird momentan gedruckt
A:\AUTOEXEC.BAT ist in der Warteschlange

Hinter dem Doppelpunkt nach [PRN] sollten Sie nur <RETURN> betätigen.

`print`

Es werden alle Dateinamen angezeigt, die sich in der Warteschlange befinden. Wenn keine vorhanden sind, wird angezeigt, daß die Warteschlange leer ist:

PRINT Warteschlange ist leer

Beispiel `B:KAP.1` wird momentan gedruckt
`B:KAP.2` ist in der Warteschlange
`B:KAP.3` ist in der Warteschlange

`print a:test c:test.neu c:test.alt kap1`

Die Dateien TEST auf der Diskette in Laufwerk A, TEST.NEU, TEST.ALT in Laufwerk C und KAP1 im aktuellen Laufwerk sollen ausgedruckt werden.

`print c:test.alt /c`

Die Datei TEST.ALT in Laufwerk C soll nicht gedruckt werden (aus der Drucker-Warteschlange löschen).

`print test*.* /c c:kap2 /p c:kap3`

Die Dateien TEST*.* werden aus der Drucker-Warteschlange gelöscht, soweit diese noch nicht oder gerade im Druck sind. KAP2 und KAP3 von Laufwerk C werden zur Drucker-Warteschlange hinzugefügt.

PROMPT *interner Befehl*

Zweck PROMPT ändert das Systemanforderungszeichen.

Kommando Aufbau prompt (\$) (text)

- Zeilenvorschub (neue Zeile)
- b das Zeichen |
- d Systemdatum (Tagesdatum)
- e Escape-Sequenz
- g das Zeichen >
- h Schritt zurück
- l das Zeichen <
- n das aktuelle Laufwerk (z.B. A oder B)
- p Name des aktuellen Directory
- s Leerschritt
- t Uhrzeit
- v Nummer der MS-DOS-Version
- \$ ein \$-Zeichen
- q ein =-Zeichen

Beispiele prompt \$n:

Der neue Prompt ist A: (A ist das aktuelle Laufwerk).

```
prompt $_Zeit = $t$_Datum = $d$_
```

Der neue Prompt ist die Anzeige der Uhrzeit (vorher neue Zeile am Bildschirm) und in einer neuen Zeile das Datum. Der Cursor steht in der nächsten Zeile. Beispiel:

```
Zeit = 11.54.34,56
```

```
Datum = Mon 10.11.1986
```

```
prompt $n$g
```

Damit wird der »normale« Prompt erzeugt (dies wird auch erreicht, wenn Sie nur den Befehl PROMPT ohne weitere Angaben verwenden).

Mit dem Befehl PROMPT können auch die Funktionstasten neu belegt und andere Tätigkeiten durchgeführt werden. Dazu muß in der Datei CONFIG.SYS die Zeile DEVICE=ANSI.SYS angegeben werden (siehe Kapitel 13).

```
prompt $e[7m$n:$e[m
```

Der Prompt wird invers dargestellt, anschließend wieder auf normale Darstellung umschalten.

```
prompt $e[33;44m$n$g
```

Wenn Sie einen Farbmonitor haben, ist anschließend der Hintergrund blau und die Schrift (Vordergrund) gelb bzw. braun.

```
prompt $e[0;60;"DIR B: ";13p  
prompt
```

Die Funktionstaste F2 (60) erhält den Inhalt »DIR B: <RETURN>«. Wenn Sie anschließend F2 betätigen, wird der Befehl DIR B: ausgeführt. Die Taste kann bis zum nächsten Systemstart nicht mehr umbelegt werden.

Bemerkungen

Die einzelnen Codes für INVERS, Farben usw. finden Sie im Kapitel 13.

Im Rahmen dieses Buches konnte auf den Befehl PROMPT nur wenig eingegangen werden. Beachten Sie hierzu Ihr Handbuch zum Computer. Weitere interessante Möglichkeiten werden laufend in den einschlägigen Computerzeitschriften veröffentlicht.

RECOVER *externer Befehl*

Zweck RECOVER erneuert eine Datei oder Diskette mit fehlerhaften Sektoren.

RECOVER darf nicht bei einem Netzwerk-Laufwerk verwendet werden.

Kommandoaufbau (d:)(weg)recover d:
(d:)(weg)recover (d:)(weg)dateiname

Beispiele recover a:test.neu

Die Datei A:TEST.NEU wird auf fehlerhafte Sektoren überprüft. Fehlerhafte Sektoren werden ausgeschlossen (es wird immer die alte und neue Dateigröße angezeigt). Wenn Fehler festgestellt wurden, fehlen anschließend im Text diese Teile. Sollte es sich um ein Programm handeln, ist dieses in der Regel anschließend nicht mehr brauchbar (daher immer eine Kopie aller Daten und Programme mit möglichst aktuellem Stand fertigen).

recover a:

Die Diskette in Laufwerk A wird auf fehlerhafte Sektoren überprüft.

Bemerkungen Beachten Sie bitte, daß das Programm alle bisherigen Dateinamen löscht und neue Namen vergibt (also vorher eine Kopie fertigen bzw. das Inhaltsverzeichnis ausdrucken, um gegebenenfalls den ursprünglichen Zustand wieder herstellen zu können).

Die Dateien erhalten in der bisherigen Reihenfolge die neuen Namen:

DATEI001.REC

DATEI002.REC

DATEI003.REC

usw.

REM *interner Befehl*

Zweck Befehl für Kommentarzeilen in Batch-Dateien (siehe auch Kapitel 12).

Kommandoaufbau rem (text)
 . (text) Kurzform des Befehls

Beispiel rem
 rem Diese Batch-Datei formatiert und überprüft neue
 Disketten
 rem Der Name dieser Datei ist NEU.BAT
 rem
 pause Legen Sie eine neue Diskette in Laufwerk A
 ein
 rem Die neue Diskette in Laufwerk A wird formatiert
 format a: /s
 rem Jetzt wird das Inhaltsverzeichnis der neuen
 Diskette angezeigt
 dir a:
 rem Jetzt wird die neue Diskette auf Fehler geprüft
 chkdsk a:
 rem Ende der Batch-Datei NEU.BAT

Bemerkungen In einer Kommentarzeile (Befehl REM) dürfen nur Komma, Tabulator oder Leerzeichen als Separator verwendet werden (z.B. kein Punkt).

Wenn Sie vor REM den Befehl

echo off

einfügen, werden die REM-Zeilen nicht angezeigt!

RENAME oder **REN** *interner Befehl*

Zweck Vorhandenen Dateien/Programmen einen neuen Namen zuordnen.

Kommando Aufbau `ren (d:)(weg)dateiname.alt dateiname.neu`
`rename (d:)(weg)dateiname.alt dateiname.neu`

Beispiele `ren test.alt versuch.alt`

Anschließend hat die Datei TEST.ALT den Namen VERSUCH.ALT.

`ren a:\sub\test.neu versuch.neu`

Die Datei TEST.NEU im Sub-Directory sub auf der Diskette in Laufwerk A wird in VERSUCH.NEU geändert. Dabei muß nur beim alten Namen der genaue Weg und die Laufwerkbezeichnung angegeben werden. Zum neuen Namen darf der Weg nicht angegeben werden!

`ren *.txt *.doc`

Dabei erhalten alle Dateien mit dem File-Typ TXT den neuen Typ DOC zugewiesen.

`ren b:text*.* ??s*.*`

Mit diesem Befehl werden nur Teile von Dateinamen verändert. Bearbeitet werden alle Dateinamen in Laufwerk B, die mit TEXT beginnen. Die ersten beiden Zeichen (TE) bleiben unverändert. Der dritte Buchstabe wird in S umgewandelt. Der Rest der Dateinamen bleibt bestehen. Damit wurde also nur das dritte Zeichen in den Namen verändert.

Bemerkung Benennen Sie niemals gekaufte Programme um, es sei denn, Sie werden ausdrücklich dazu aufgefordert, da sonst eine einwandfreie Funktion nicht mehr gewährleistet ist.

REPLACE *externer Befehl*

Zweck Ermöglicht das Ersetzen (Update) von Dateien einer neuen Version einer Software (usw.) mit einer bisherigen auf Ihrer Festplatte oder Diskette. Es werden die alten Dateien durch die neuesten Versionen ersetzt bzw. Dateien, die bisher nicht existierten, hinzugefügt.

Kommando Aufbau (d:)(weg)replace (d:)(weg)dateinamen d:(weg)(/x ...)

- /x /a Hinzufügen neuer Dateien (nicht zusammen mit /D und /S)
- /d Dateien ersetzen, wenn die bisherige älter ist
- /p Vor dem Kopieren einer Datei wird eine Bestätigung verlangt
- /r Dateien mit dem Attribut »Nur-Lesen« werden auch ersetzt
- /s Das Zielverzeichnis und dessen Sub-Directories werden nach einer älteren Version durchsucht
- /w Vor dem Ersetzen oder Hinzufügen von Dateien wird gewartet, bis Sie eine Taste betätigen

Beispiele replace a:*.com c: /s/d

Auf der Diskette in Laufwerk A befinden sich Programme mit dem Typ COM. Die älteren Versionen auf Laufwerk C sollen ersetzt werden, wobei die Root- und Sub-Directories durchsucht werden.

replace c:*.dat a: /p/r/w

Vom aktuellen Directory auf Laufwerk C sollen alle Dateien mit dem Typ DAT kopiert werden. Nach dem Start bzw. vor dem eigentlichen Kopieren werden Sie aufgefordert, eine beliebige Taste zu betätigen (/W), damit der Kopiervorgang gestartet werden kann. Damit können Sie vorher noch eine andere Diskette in Laufwerk A einlegen. Bevor eine Datei kopiert wird, werden Sie gefragt (/P), ob Sie dies auch wirklich wünschen (dazu wird der Dateiname angezeigt). Es werden auch die Dateien ersetzt, die auf der Diskette in Laufwerk A auf »Nur-Lesen« gesetzt sind.

Bemerkungen Nach Beenden von REPLACE gelten folgende Fehlercodes (können mit IF ERRORLEVEL in Batch-Dateien abgefragt werden – Kapitel 12):

- 1 Fehler in der Befehlszeile (z.B. unbekannter Parameter)
- 2 keine passenden Ziel- oder Quelldateien gefunden
- 3 entweder Quellen- oder Zielweg nicht gefunden
- 5 Versuch, schreibgeschützte Datei zu aktualisieren /
Abbruch
- 8 nicht genügend Speicherplatz vorhanden
- 15 Laufwerk existiert nicht
andere Nummern sind MS-DOS-Standardfehler

RESTORE *externer Befehl*

(Nur bei Festplattensystemen)

Zweck RESTORE holt die mit BACKUP gesicherten Dateien auf Diskette oder Festplatte zurück.
Siehe auch Beschreibung zu BACKUP in diesem Kapitel.

Kommandoaufruf (d:)(weg)restore d: ((d:)(weg)(dateinamen) (/x ...)

/x /p Wenn eine Datei das Attribut Read-Only (ATTRIB) hat oder seit dem letzten BACKUP verändert wurde, werden Sie am Bildschirm befragt, ob eine Rückkopie erfolgen soll (gilt auch für IO.SYS und MSDOS.SYS).

/s Es werden auch alle Sub-Directories zurückkopiert.

Beispiele restore a: c: /s

Es werden vom Laufwerk alle Dateien und Sub-Directories nach Laufwerk C zurückkopiert.

restore a: c:\msdos /p

Es werden alle Dateien für das Sub-Directory MSDOS zurückkopiert. Eine Meldung erfolgt, wenn es sich um Dateien handelt, die seit der letzten Sicherung mit BACKUP kopiert wurden, oder es sich um »Nur-Lese«-Dateien handelt.

restore a: c:\auftrag\fibu\prog.com

Das Programm PROG.COM wird von der Sicherungsdiskette in das Sub-Directory FIBU zurückübertragen.

restore a: c:*.dat

Alle Dateien der Sicherungsdiskette, die aus dem aktuellen Directory gesichert wurden, werden wieder zurückübertragen.

Bemerkungen Wenn Sie den Befehl RESTORE in einer Batch-Datei verwenden, kann mit IF ERRORLEVEL anschließend der Fehlercode abgefragt werden:

- 0 RESTORE fehlerfrei beendet
- 1 keine Datei gefunden
- 2 File-Sharing-Konflikt (Netzwerk)
einige oder alle Dateien nicht zurückkopiert

- 3 Sie haben RESTORE abgebrochen (durch Ctrl+Break)
- 4 RESTORE durch Fehler abgebrochen

Weitere Informationen zu Batch-Dateien finden Sie im Kapitel 12.

RMDIR oder RD *interner Befehl*

Zweck Löschen von Sub-Directory-Einträgen (ohne die dort enthaltenen Dateien). Vorher müssen Sie alle im Sub-Directory gespeicherten Dateien und eventuelle weitere Directories löschen (sonst wird der Befehl RD nicht ausgeführt). Beachten Sie hierzu das Kapitel 8.

Kommandoaufbau rmdir (d:)weg
rd (d:)weg

Beispiele md versuch

Sub-Directory VERSUCH anlegen

cd versuch

VERSUCH wird aktuelles Directory

md versuch2

Sub-Directory VERSUCH2 in VERSUCH anlegen

cd ..

aktuelles Directory wird das hierarchisch vor VERSUCH liegende Directory

rd versuch

Löschen des Sub-Directory VERSUCH – nicht möglich, da darin noch VERSUCH2 vorhanden ist. Fehlermeldung:

Ungültiger Pfad, kein Verzeichnis,

oder

Verzeichnis nicht leer

rd versuch\versuch2

Löschen des Sub-Directory VERSUCH2

rd versuch

Sub-Directory VERSUCH kann jetzt gelöscht werden, da dieses leer ist

SET *interner Befehl*

Zweck SET setzt den Wert einer Zeichenkette gleich dem Wert einer anderen Zeichenkette zur Benutzung in Programmen (kommt vor allem im Zusammenhang mit Batch-Dateien zur Anwendung).

Kommandoaufbau set
set (zeichenkette1=(zeichenkette2))

Beispiele set

Es wird der im Environment-Bereich (Zwischenspeicher) gespeicherte Teil angezeigt (nach dem Systemstart befindet sich dort nur der Name des Befehlsinterpreters COMMAND.COM). Beispiel für Anzeige:

```
COMSPEC=C:\COMMAND.COM
```

```
PATH=\MSDOS;\TEXT\ABT1
```

```
set var=text
```

Batch-Datei TEST.BAT:

```
rem Dies ist eine Beispieldatei
```

```
echo off
```

```
echo In der folgenden Zeile wird die Variable VAR  
gezeigt:
```

```
echo %var%
```

Nach Aufruf von TEST.BAT wird angezeigt:

```
REM Dies ist eine Beispieldatei
```

```
ECHO OFF
```

```
In der folgenden Zeile wird die Variable VAR  
gezeigt:
```

```
TEXT
```

```
set TTY=VT100
```

Damit wurde der Name des Gerätetreibers TTY auf VT100 geändert.

```
set TTY=
```

Damit wird die Variable TTY wieder gelöscht

Eine weitere Möglichkeit der Benutzung in Batch-Dateien:

```
if "%display%"=="ON" goto marke
```

Wenn die Variable DISPLAY mit SET vorher auf ON gesetzt wurde, so wird mit GOTO zu :MARKE verzweigt und dort die Batch-Datei weiter bearbeitet.

Bemerkungen Sollten Sie die Fehlermeldung »Kein Umgebungsbereich mehr vorhanden« erhalten, so ist nicht mehr genügend Speicher für Variablen vorhanden! Dieser Speicher hat nur eine auf 128 Zeichen begrenzte Größe (kann mit COMMAND vergrößert werden – COMMAND /E:1024).

SHARE *externer Befehl*

Zweck Installiert den sogenannten File-Sharing-Support, wenn Sie ein Netzwerk einsetzen.

Kommandoaufbau (d:)(weg)share
(d:)(weg)share /x
/x /f: # Definition der Größe des Speichers für Informationen über Dateien, die im Netzwerk verwendet werden. Für jede verwendete (offene) Datei wird Speicher für den vollen Dateinamen und 11 weitere Stellen benötigt. Der Standardwert ist 2.048.
/l: # Der Standardwert ist 20. Es handelt sich hierbei um die Anzahl der sogenannten »Locks« (Dateizugriffsschutz). Wenn der Wert über 20 gesetzt wird, muß auch /F angegeben werden.

Bemerkungen Weitere Informationen finden Sie im Handbuch zur Netzwerk-Software von Microsoft »Networks Manager's Guide« und im Anhang zu diesem Buch.

SHIFT *interner Befehl*

Zweck SHIFT erhöht die Zahl der Variablen in einer Batch-Datei auf mehr als 10. Beachten Sie dazu auch das Kapitel 12.

Kommandoaufbau shift

Beispiele Batch-Datei LOESCHE.BAT:

```
rem Dies ist die Datei LOESCHE.BAT
:anfang
if "%1"==" " goto ende
del %1
shift
goto anfang
:ende
```

Beispiel für den Aufruf der Datei LOESCHE.BAT:

```
loesche datei1 datei2 datei3
```

Beispiel für die Anzeigen bzw. Aktionen:

```
rem Dies ist die Datei LOESCHE.BAT
:anfang
if "datei1"==" " goto ende
del datei1
shift
goto anfang
```

```
:anfang
if "datei2"==" " goto ende
del datei2
shift
goto anfang
```

```
:anfang
if "datei3"==" " goto ende
del datei3
shift
goto anfang
```

```
:anfang
if ""==" " goto ende
:ende
```

Die zum Befehl eingegebenen Dateinamen werden der Reihe nach bearbeitet und im vorliegenden Beispiel gelöscht. Der Befehl SHIFT »holt« also immer den nächsten Namen aus der von Ihnen eingegebenen Befehlszeile.

SORT *externer Befehl*

Zweck Sortieren von Daten. Es handelt sich bei SORT um einen Filter. Die Daten können auf- oder absteigend sortiert werden. Beachten Sie bitte das Kapitel 9.

Kommandoaufbau (d:)(weg)sort (/x) (< dateiname1) (> dateiname2)

/x /r sortiert absteigend

/+ # sortiert die Datenzeilen ab der Spalte #

Beispiele sort < unsort.asc

Der Inhalt der Datei UNSORT.ASC wird aufsteigend sortiert am Bildschirm angezeigt.

```
sort < unsort.asc > sort.asc
```

Der Inhalt der Datei UNSORT.ASC wird aufsteigend sortiert in die Datei SORT.ASC übertragen.

```
dir c: | sort
```

Das Inhaltsverzeichnis des Laufwerks C wird aufsteigend sortiert am Bildschirm angezeigt.

```
dir c: | sort > prn
```

Das Inhaltsverzeichnis des Laufwerks C wird aufsteigend sortiert am Drucker ausgedruckt.

```
dir c: | sort /r /+14 | more
```

Das Inhaltsverzeichnis des Laufwerks C wird absteigend nach Dateigrößen (die Anzahl der Bytes bzw. die Größe jeder Datei steht im Inhaltsverzeichnis ab der 14. Spalte) sortiert auf den Bildschirm ausgegeben. Nach jeder vollgeschriebenen Bildschirmseite stoppt die Ausgabe, bis eine beliebige Taste gedrückt wird.

SUBST *externer Befehl*

Zweck Einem Sub-Directory wird eine Laufwerksbezeichnung zugewiesen. Die Befehle

ASSIGN
BACKUP
DISKCOMP
DISKCOPY
FORMAT
JOIN
LABEL
RESTORE

dürfen nicht verwendet werden.

SUBST darf nicht für ein Netzwerk-Laufwerk verwendet werden. Sollte jedoch SUBST auf Ihrem Schneider PC bei richtiger Anwendung nicht einwandfrei funktionieren, so geben Sie vorher den Befehl SHARE ein.

Kommandoaufbau (d:)(weg)subst
(d:)(weg)subst d: (d:)weg
(d:)(weg)subst d: /x

/x /d Die mit SUBST erstellte Zuweisung löschen

Beispiele subst f: a:\auftrag\fibu

Dem Sub-Directory FIBU wird die Laufwerksbezeichnung F zugewiesen. Wenn Sie z. B. den Befehl dir f: benutzen, erhalten Sie das Inhaltsverzeichnis des Sub-Directory FIBU.

subst

Es werden die von Ihnen definierten und aktiven Zuweisungen angezeigt.

subst f: /d

Die Zuweisung eines Sub-Directory für Laufwerk F wird gelöscht.

Bemerkungen Normalerweise sind nur die Laufwerksbezeichnungen bis E möglich. Wenn Sie in der Datei CONFIG.SYS mit dem Befehl LASTDRIVE arbeiten, können Namen bis maximal Z vergeben werden (siehe Kapitel 13).

SYS *externer Befehl*

Zweck SYS kopiert die Systemdateien IO.SYS und MSDOS.SYS auf eine Diskette/Festplatte. Beachten Sie bitte, daß auf der Empfänger-Diskette/-Festplatte auf den ersten Sektoren ausreichend Platz vorhanden sein muß, sonst wird zwar das System unter Umständen kopiert, es kann aber beim Booten nicht ordnungsgemäß geladen werden. Der erste Bereich (Sektoren) ist meist zu klein, wenn vorher eine ältere Version von MS-DOS dort gespeichert war. Der Befehl darf nicht für ein Netzwerk-Laufwerk verwendet werden.

Kommandoaufbau (d:)(weg)sy s d:

Beispiel sys c:

Das System wird vom aktuellen Laufwerk auf die Festplatte (Laufwerk C) übertragen.

TIME *interner Befehl*

Zweck Anzeigen bzw. Ändern der Uhrzeit.

Kommandoaufruf time
time hh(.mm(.ss(cc)))

hh	Stunde	mögliche Werte von 00 bis 23
mm	Minute	mögliche Werte von 00 bis 59
ss	Sekunde	mögliche Werte von 00 bis 59
cc	hundertstel Sekunde	mögliche Werte von 00 bis 99

Beispiele time

Es wird die aktuelle Uhrzeit angezeigt, und Sie können eine neue Uhrzeit eingeben.

```
time 12.18
```

Die neue Uhrzeit ist 12 Uhr und 18 Minuten (0 Sekunden und 0 hundertstel Sekunden) – das System setzt mit dieser Uhrzeit fort.

```
time 0
```

Setzen der Zeit auf 0. Damit kann die Zeit für eine bestimmte Tätigkeit oder einen Ablauf festgestellt werden.

Bemerkung Mit TIME stellen Sie die Uhrzeit Ihres PC neu. Sie wird in einem Bereich gespeichert und weitergezählt, wo sie von einer in Ihrem Schneider PC vorhandenen Batterie Strom erhält.

TREE *externer Befehl*

Zweck TREE zeigt das strukturierte Verzeichnis einer Diskette oder Festplatte mit an.

Kommandoaufbau (d:)(weg)tree (d:)(/x)

/x /f Zu den Namen der einzelnen Sub-Directories werden auch die darin enthaltenen Dateinamen angezeigt.

Beispiele tree

Es werden alle Sub-Directory-Namen angezeigt.

tree /f | more

Es werden alle Directories und die darin enthaltenen Dateinamen angezeigt. Nach jeder angezeigten Bildschirmseite wird die Anzeige gestoppt, bis Sie eine beliebige Taste betätigen.

tree /f > prn

Es werden alle Verzeichnisse und die darin enthaltenen Dateinamen ausgedruckt.

Beispiel für eine Ausgabe mit dem Parameter /f:

PFADVERZEICHNIS

```
Dateien:      TREE.COM
              SELECT.COM
              MODE.COM
              SORT.EXE
```

Pfad: \MSDOS

Unterverzeichnisse: Keine

```
Dateien:      ASSIGN.COM
              CHKDSK.COM
```

Pfad: \AUFTRAG

Unterverzeichnisse: FIBU

```
Dateien:      Keine
```

Pfad: \AUFTRAG\FIBU

Unterverzeichnisse: Keine

Dateien: FIBU.EXE
JOURNAL.DAT
OP.DAT

TYPE *interner Befehl*

Zweck Den Inhalt einer Datei am Bildschirm anzeigen. Mit Ctrl+S kann die Anzeige jederzeit gestoppt bzw. fortgefahren werden.

Kommandoaufbau type dateiname
type (d:)(weg)dateiname

Beispiele type autoexec.bat

Der Inhalt der Datei AUTOEXEC.BAT wird am Bildschirm angezeigt.

type a:\sub\test.alt

Der Inhalt der Datei TEST.ALT im Sub-Directory sub in Laufwerk A wird angezeigt.

type text | more

Der Inhalt der Datei TEXT wird angezeigt (nach jeder vollen Bildschirmseite wartet die Anzeige, bis eine beliebige Taste gedrückt wird).

type autoexec.bat > prn

Der Inhalt der Datei AUTOEXEC.BAT wird ausgedruckt.

Bemerkungen Statt des Befehls

type autoexec.bat

kann alternativ

copy autoexec.bat con

verwendet werden (siehe Befehl COPY).

VER *interner Befehl*

Zweck Die Versionsnummer von MS-DOS anzeigen.

Kommandoaufbau ver

Beispiel ver

Es erscheint z.B.:

MS-DOS Version 3.20

VERIFY *interner Befehl*

Zweck Wenn mit diesem Befehl »Verify« eingeschaltet wird, erfolgt eine Überprüfung aller Disketten- und Festplatten-Operationen auf korrekte Datenaufzeichnung bzw. -übertragung. Standardmäßig nach dem Systemstart ausgeschaltet.

Kommando Aufbau verify (on)
verify (off)

Beispiele verify

Es wird der momentane Stand angezeigt (ob ein- oder ausgeschaltet):

VERIFY is off

verify on

Prüfen einschalten.

Bemerkungen Wenn VERIFY eingeschaltet ist, benötigt z.B. das Kopieren von Dateien mehr Zeit. Es gibt Programme, die diese Funktion beim Start automatisch ein- oder ausschalten.

VOL *interner Befehl*

Zweck Der Name einer Diskette oder Festplatte wird angezeigt (Label). Der Name kann mit dem Befehl LABEL verändert oder gelöscht werden.

Kommandoaufbau vol (d:)

Beispiel vol a:

Der Name der Diskette in Laufwerk A wird am Bildschirm angezeigt – Beispiel:

Volume in Laufwerk A hat keinen Namen

oder

Volume in Laufwerk B ist SCHNEIDER

XCOPY *externer Befehl*

Zweck XCOPY ist ein erweiterter COPY-Befehl. Mit ihm können Sie nicht nur Dateien, sondern auch ganze Directory-Strukturen kopieren.

Kommandoaufbau (d:)(weg)xcopy (d:)(weg)dateiname (d:)(weg)(dateiname) (/x ...)

/x	/a	Kopiert alle Dateien, die seit dem letzten Kopiervorgang verändert wurden
	/d:tt.mm.jj	Kopiert werden nur die Dateien, die das angegebene oder ein späteres Datum besitzen
	/e	Kopiert leere Sub-Directories und solche, die Dateien beinhalten (dazu muß auch /S verwendet werden)
	/m	Kopiert alle Dateien, die seit dem letzten Kopiervorgang verändert wurden (das entsprechende Kennzeichen wird nach dem Kopieren zurückgesetzt)
	/p	Zu jeder Datei werden Sie vorher befragt, ob diese kopiert werden soll
	/s	Kopiert die aktuelle und alle sich darunter befindlichen Sub-Directories
	/v	Nach dem Kopieren wird die Kopie mit dem Original auf Gleichheit überprüft
	/w	Bevor der eigentliche Kopiervorgang beginnt, muß noch eine beliebige Taste betätigt werden (Sie haben die Möglichkeit, vorher noch die Diskette zu wechseln)

Beispiele xcopy c:\auftrag*.* a: /e/p/s/v/w

Es werden das Sub-Directory AUFTRAG sowie alle sich darunter befindlichen Directories kopiert (/S), unabhängig davon, ob diese Dateien enthalten (/E). Vor dem Start des Kopiervorganges müssen Sie eine beliebige Taste betätigen (/W). Bevor eine Datei kopiert wird, werden Sie befragt, ob dies auch wirklich durchgeführt werden soll (/P). Jede kopierte Datei wird, bevor der Kopiervorgang fortgesetzt wird, mit dem Original auf Gleichheit verglichen (/V).

```
xcopy a:\*.* b:\kop\ /d:01.10.86
```

Alle Dateien des Root-Directory von Laufwerk A werden in das Sub-Directory KOP in Laufwerk B kopiert (ohne eventuell weiter vorhandene Sub-Directories in Laufwerk A). Dabei werden nur Dateien berücksichtigt, die als Datum den 1.10.86 oder danach besitzen.

Bemerkungen Es gibt folgende Fehlercodes für die Verwendung bei IF ERRORLEVEL:

- 0 Keine Fehler
- 1 Keine Datei wurde kopiert
- 2 Befehl durch Ctrl+Break bzw. Ctrl+C abgebrochen
- 4 Der angegebene Weg bzw. das Sub-Directory wurde nicht gefunden
- 5 Nach Diskettenfehler abgebrochen

Kapitel 12 BATCH-Befehle

Befehle in Batch-Dateien ablegen

In einer Batch-Datei werden nacheinander mehrere Befehle auf Betriebssystemebene abgearbeitet. Bei einer direkten Eingabe müßten diese Befehle hintereinander abgerufen werden. Schreiben Sie diese Befehle vorher in eine Batch-Datei, werden diese so abgearbeitet, als wenn sie direkt eingegeben werden. Damit ist es nicht erforderlich, wiederkehrende Routinen bei Bedarf neu zu schreiben.

MS-DOS erkennt eine Batch-Datei am File-Typ BAT. Der Name muß eindeutig sein. Es darf nicht schon ein Programm mit dem gleichen File-Namen und dem File-Typ COM oder EXE vorhanden sein.

Beim Starten sieht das System nach, ob eine Datei mit dem Namen AUTOEXEC.BAT vorhanden ist. Wenn ja, so werden die darin enthaltenen Befehle automatisch abgearbeitet. Ist eine solche nicht im Root-Directory vorhanden, so werden Sie nur nach Datum und Uhrzeit befragt.

Unterbrechen einer Batch-Datei

Der Ablauf einer Batch-Datei (bzw. Programm) kann jederzeit durch Betätigung von Ctrl+Break oder Ctrl+C abgebrochen werden. Sie erhalten die Frage

Stapel Job beenden (J/N)?

am Bildschirm (ob Sie wirklich abbrechen wollen).

Disketten wechseln bei Ablauf einer Batch-Datei

Wenn Sie während des Ablaufs einer Batch-Datei die Diskette wechseln und ein neuer Befehl ausgeführt werden soll, erhalten Sie die Meldung

Disk mit Batch Datei einlegen
und beliebige Taste drücken wenn fertig

Legen Sie dann die Diskette mit der Batch-Datei ein, und betätigen Sie eine beliebige Taste.

Kommandoaufbau dateiname (variable1 (variable2 (...)))

dateiname Batch-Datei mit File-Typ BAT (kann innerhalb der Datei als %0 verwendet werden).

variable# Dies können Dateinamen oder andere Angaben sein. In den Befehlen innerhalb der Datei als %1 bis %9 verwendbar .

Beispiele Erstellen Sie bitte folgende Batch-Datei (mit dem Befehl COPY oder EDLIN):

```
copy con autoexec.bat
```

```
oder
```

```
edlin autoexec.bat
```

```
echo off
```

```
date
```

```
time
```

```
keybgr
```

```
cls
```

(F6 und <RETURN>)

Batch-Datei testen Starten Sie Ihren Computer neu, indem Sie die Tastenkombination (gleichzeitig drücken) Ctrl+Alt+Del betätigen. Anschließend erscheint am Bildschirm:

```
echo off
```

```
Aktuelles Datum ist Son 26.10.1986
```

```
Neues Datum eingeben (tt-mm-jj):
```

```
Aktuelle Zeit ist 21.48.38,68
```

```
Neue Zeit eingeben:
```

Anschließend wird der deutsche Zeichensatz geladen und der Bildschirm gelöscht.

Erstellen Sie die folgende Batch-Datei mit dem Namen UEB.BAT:

```
copy %1 a:%2.neu
```

```
type %1
```

```
pause
```

```
type %2.neu
```

```
rem Dies ist die Batch-Datei
```

```
type %0.bat
```

Beispiele Drei Beispiele für den Aufruf von UEB.BAT:
Geben Sie jetzt ein:

```
ueb diskette.asc diskette
```

Den Namen UEB können Sie in der Batch-Datei mit %0, DISKETTE.ASC mit %1 und DISKETTE mit %2 verwenden.

Im vorliegenden Beispiel erhalten Sie folgende Anzeigen:

```
A>copy diskette.asc a:diskette.neu
      1 Datei(en) kopiert
```

```
A>type diskette.asc
```

Es wird der Inhalt der Datei DISKETTE.ASC am Bildschirm angezeigt.

```
A>pause
```

Wenn fertig, eine Taste betätigen . . .

```
A>type DISKETTE.neu
```

Es wird der Inhalt der Datei DISKETTE.NEU am Bildschirm angezeigt.

```
A>rem dies ist die Batchdatei
```

```
A>type UEB.bat
```

Es wird der Inhalt der Datei UEB.BAT am Bildschirm angezeigt.

```
ueb text
```

```
A>if "TEXT"==" " goto ende
```

```
A>:weiter
```

```
A>if "TEXT"==" " goto ausgang
```

```
A>if not exist TEXT goto nicht
```

```
A>del TEXT
```

```
A>echo Datei TEXT ist gelöscht
```

```
Datei TEXT ist gelöscht
```

```
A>goto aus
```

```
A>:aus
```

```
A>shift
```

```
A>goto weiter
```

```
A>if ""==" " goto ausgang
```


ueb

A>if ""==" goto ende

A>:ende

A>echo Datei nicht angegeben

Datei nicht angegeben

A>goto ausgang

**Batch-Datei
zum Diskettenprüfen**

Die Batch-Datei für die Überprüfung einer Datei, ob diese vorhanden ist, könnte wie folgt aussehen:

echo off

if exist %1 goto ja

echo Die Datei %1 ist nicht vorhanden

goto ende

:ja

echo Die Datei %1 ist vorhanden

:ende

**Beispiel
für »DOS-Menü«**

Die Batch-Datei AUTOEXEC.BAT könnte wie folgt aussehen:

echo off

keybgr

date

time

path=\msdos

cls

echo Programmauswahl:

echo -----

echo DIR Anzeigen des Inhaltsverzeichnisses

echo FIBU Finanzbuchhaltung

echo LUGA Lohn und Gehalt

echo TEXT Textprogramm

echo -----

echo Programmaufruf durch Eingabe des Namens

rem

**Beispiel für
»Lösch-Datei«**

Beispiel für die Batch-Datei zum Löschen von Dateien (LOE.BAT):

```
if "%1"==" " goto ende
:weiter
if "%1"==" " goto ausgang
if not exist %1 goto nicht
del %1
echo Datei %1 ist gelöscht
goto aus
:nicht
echo Datei %1 nicht vorhanden
goto aus
:ende
echo Datei nicht angegeben
goto ausgang
:aus
shift
goto weiter
:ausgang
```

**Aufruf der
»Lösch-Datei«**

Beispiel für den Aufruf von LOE.BAT:

```
loe niemals

A> if "NIEMALS"==" " goto ende
      :weiter
A> if "NIEMALS"==" " goto ausgang
A> if not exist NIEMALS goto nicht
      :nicht
A> echo Datei NIEMALS nicht vorhanden
Datei NIEMALS nicht vorhanden
A> goto aus
A> shift
A> goto weiter
      :weiter
A> if ""==" " goto ausgang
```

Bemerkungen

Normalerweise wird als erstes in der Batch-Datei der Befehl echo off angegeben, damit nicht jeder Befehl angezeigt wird. Die Anzeige der Ansprungpunkte (z.B. :weiter) erfolgt nicht. Für die Programmauswahl beim Start des Computers kann ein einfaches Programm gewählt werden (AUTOEXEC.BAT):

**Beispiel eines
Menü-Programms**

Hans H. Gerhardt

```
:Author
:start
echo off
cls
echo -----
echo
echo 1 = Finanzbuchhaltung
echo 2 = Textprogramm
echo 3 = MS-DOS
echo
echo Wählen Sie 1, 2 oder 3
echo -----
:schleife
eingabe
if errorlevel 51 goto eins
if errorlevel 50 goto zwei
if errorlevel 49 goto drei
goto schleife
:eins
cls
echo Programm beendet
goto ende
:zwei
if not exist TEXT.COM goto nicht
TEXT
goto start
:drei
if not exist FIBU.COM goto nicht
FIBU
goto start
:nicht
echo Das gewählte Programm ist nicht vorhanden
pause
goto start
:ende
```

Mit dem Programmende von EINGABE kann ein Wert, den Sie an der Tastatur eintippen, in einen Bereich geschrieben werden, den Sie anschließend mit ERRORLEVEL lesen und verarbeiten können. Das Programm EINGABE kann sehr einfach mit dem Befehl DEBUG erstellt werden:

Programm	A> debug	
»EINGABE«	-n eingabe.com	(<RETURN>)
	-a	(<RETURN>)
	xxxx:0100 (Tab) mov (Tab) ax,0c07	(<RETURN>)
	xxxx:0103 (Tab) int (Tab) 21	(<RETURN>)
	xxxx:0105 (Tab) mov (Tab) ah,4c	(<RETURN>)
	xxxx:0107 (Tab) int (Tab) 21	(<RETURN>)
	xxxx:0109	(<RETURN>)
	-rcx	(<RETURN>)
	CX 0000	
	:9	(<RETURN>)
	-w	(<RETURN>)
	Schreibe 0009 Bytes	
	-q	(<RETURN>)
	A>	

Bemerkungen Weitere Beispiele finden Sie im Kapitel 11 bei den einzelnen Befehlen – z.B. bei

ECHO

FOR

GOTO

IF

PAUSE

REM

SHIFT

Kapitel 13

Die Konfiguration von MS-DOS (CONFIG.SYS)

Schnittstelle zu Anwendungs- programmen

Die Datei MSDOS.SYS ist der »Direktor« Ihres PC. Sie erzeugt die Instruktionen für die zu erledigenden Arbeiten. Die Vorgänge laufen hardwareunabhängig ab (der hardwareabhängige Teil zur Bedienung der Peripherie ist IO.SYS). MSDOS.SYS ist eine intelligente Schnittstelle für Anwendungsprogramme. Es verfügt über Programmteile zur Dateiverwaltung, Steuerung von Informationsblöcken bei Disketten-/Festplattenzugriff usw.

MSDOS.SYS anpassen mit CONFIG.SYS

Es können Aufrufe der Anwenderprogramme erkannt und bei Bedarf an IO.SYS zur Ausführung weitergeleitet werden.

MSDOS.SYS können Sie an Ihre Anforderungen anpassen (erweitern). Dazu wird die Datei CONFIG.SYS benötigt. Es handelt sich dabei um eine Textdatei (vergleichbar mit AUTOEXEC.BAT), die Sie mit EDLIN oder COPY erstellen können.

Startvorgang von MS-DOS

Während des Ladevorganges von MS-DOS wird nach der Datei CONFIG.SYS gesucht. Wenn diese gefunden wird, werden die darin enthaltenen Befehle ausgeführt.

Der Startvorgang in Einzelschritten (vereinfachte Darstellung):

- 1 Lesen des ersten Sektors von der Diskette/Festplatte
- 2 Laden des Programmes BOOT-LOADER und starten
- 3 Der BOOT-LOADER lädt IO.SYS und übergibt die Kontrolle
- 4 IO.SYS initialisiert die Peripherie und lädt MSDOS.SYS
- 5 IO.SYS übergibt die Kontrolle an das Modul SYSINIT (stellt die Verbindung zwischen IO.SYS und MSDOS.SYS her), das mit IO.SYS geladen wurde

- 6 SYSINIT sucht CONFIG.SYS und führt die darin enthaltenen Befehle aus
- 7 SYSINIT lädt COMMAND.COM (oder den in CONFIG.SYS angegebenen Befehls-Interpreter)
- 8 COMMAND.COM sucht AUTOEXEC.BAT und führt die darin enthaltenen Befehle aus (erst hier kann der Startvorgang durch Ctrl+Break abgebrochen werden)

Die CONFIG.SYS-Befehle

BREAK = on

BREAK = off

Programm- unterbrechung

Dieser Befehl modifiziert die Programmunterbrechung durch die Tastenkombination Ctrl+Break. Ist die Funktion eingeschaltet, wird das laufende Programm sofort nach Betätigen der Tasten unterbrochen. Ist die Funktion ausgeschaltet, wird das Programm nur noch bei einer Ein- oder Ausgabe von einem Peripheriegerät (Bildschirm, Tastatur, Drucker usw.) mit Ctrl+Break unterbrochen. Ohne Modifizierung ist die Funktion ausgeschaltet.

BUFFERS = Nummer

Zwischenspeicher

Bei der Nummer handelt es sich um eine dezimale Zahl zwischen 1 und 99. Sie legt die Anzahl der Zwischenspeicher fest, die im RAM zusätzlich (zwei sind generell vorhanden) für Daten reserviert werden. Dadurch wird vielfach der Programmablauf schneller. Jeder Zwischenspeicher (Puffer) belegt 528 Zeichen (Byte) des Arbeitsspeichers. Selten benötigen Anwenderprogramme mit BUFFERS eine Mindestspeichergröße, die Sie in den Unterlagen zum Programm finden. Die Puffer werden von MS-DOS bzw. Ihren Programmen für die Zwischenspeicherung von Daten beim Lesen und Schreiben auf einer Festplatte bzw. Diskette benötigt.

Daten werden immer sektorweise gelesen bzw. geschrieben. Wenn Sie einen Datensatz (z.B. eine Kundenadresse) lesen, werden erst die Puffer durchsucht, ob dieser dort bereits vorhanden ist. Wenn nicht, wird er von der Festplatte/Diskette in den Puffer geholt (ein Sektor) und von dort zu Ihrem Programm übertragen. Daten, die beim Lesen im Puffer gefunden

werden, beschleunigen damit die Programmgeschwindigkeit erheblich.

Nicht immer bedeutet eine größere Anzahl von Puffern eine Beschleunigung der Verarbeitung. Vor allem ist zu beachten, daß eine größere Anzahl von Puffern auch erheblich mehr Speicher benötigt. Dies kann dazu führen, daß der Platz im Speicher für ein Programm zu klein wird.

Für Festplatten-PCs wird ein Mindestwert von 3 empfohlen. Wenn Sie mit einer Datenbank arbeiten, führt in der Regel ein Wert zwischen 10 und 25 zu den besten Ergebnissen.

```
buffers = 15
```

COUNTRY = Nummer

Landeskennziffer

Bei der Nummer handelt es sich um eine dezimale Zahl zwischen 1 und 999. Sie bestimmt den landesspezifischen Datensatz, der nachgeladen werden kann. Dieser enthält Angaben über das Format für die Datumsanzeige und Uhrzeit. Ebenso ist das Währungssymbol gespeichert und weitere Angaben für das Formatieren von Ausgaben auf dem Bildschirm oder Drucker. Es ist somit möglich, das Format für Uhrzeit und Datum auf das gewohnte Aussehen zu verändern. Wird der Befehl nicht eingegeben, ist automatisch der amerikanische Datensatz geladen.

Bei der dreistelligen Nummer handelt es sich um die internationale Telefonvorwahl des entsprechenden Landes. Ist das gewünschte Land nicht vorhanden, müssen Sie die Kennziffer des ähnlichsten Landes verwenden.

Diese Angabe bewirkt nicht, daß die Meldungen anschließend in der entsprechenden Sprache angezeigt werden. Ferner muß für die richtige Tastaturbelegung auch noch ein spezielles Programm geladen werden (KEYBGR für Deutschland).

Land	Kennziffer	Land	Kennziffer
Australien	061	Italien	039
Belgien	032	Kanada	001
Dänemark	045	Niederlande	031
Deutschland	049	Norwegen	047
Finnland	358	Schweiz	041
Frankreich	033	Spanien	034
Großbritannien	044	USA	001
Israel	972		

FCBS = Nummer1,Nummer2

Dateisteuerblöcke Dieser Befehl ist nur sinnvoll, wenn im Netzwerk gearbeitet und der Befehl SHARE geladen wird (gemeinsamer Dateizugriff). Dies gilt jedoch auch nur, wenn ein Programm zur Dateiverwaltung die Dateisteuerblöcke (FCB – File Control Block) benutzt.

Vorsicht bei der Eingabe Daher sollten Sie diesen Befehl nur verwenden, wenn das ausdrücklich in der Beschreibung zu einem Ihrer Programme angegeben ist!

Sie bestimmen mit dem Wert für Nummer1 die maximale Zahl der gleichzeitig durch einen FCB geöffneten Dateien. Der Wert für Nummer2 bestimmt die Zahl der geschützten FCBs. Versucht ein Anwendungsprogramm mehr FCBs zu öffnen, als mit Nummer1 bestimmt wurde, schließt MS-DOS automatisch den zuletzt geöffneten FCB. Dies ist erforderlich, da der benutzte Speicherplatz benötigt wird. Mit dem Wert für Nummer2 bestimmen Sie die Zahl der FCBs, die unbedingt offengehalten werden müssen.

Ohne Eingabe des Befehles wird Platz für vier FCBs reserviert. Davon ist keiner geschützt. Für Nummer1 können beliebige Werte im Bereich 1 bis 255 eingegeben werden. Für Nummer2 können beliebige Werte im Bereich 0 bis 255 eingegeben werden. Dabei darf die Zahl für Nummer2 nicht größer sein als für Nummer1. Beziehen sich zwei verschiedene FCBs auf dieselbe Datei, zählt MS-DOS dies als einen FCB. Sind die Werte für Nummer1 und Nummer2 gleich, kann MS-DOS keine weitere Datei öffnen und gibt eine Fehlermeldung aus, wenn dies trotzdem versucht wird. Sie sollten dann den Wert für Nummer1 erhöhen.

FCBs und Netzwerke Haben Sie Programme für ein Netzwerk installiert, und ein Programm versucht eine größere Zahl von FCBs zu öffnen als mit Nummer1 bestimmt, schließt MS-DOS den zuletzt geöffneten FCB wieder. Dies können Sie verhindern, indem Sie mit Nummer2 eine bestimmte Anzahl von FCBs schützen. MS-DOS erkennt diese FCBs dann nicht mehr als zuletzt geöffnet.

Versucht Ihr Programm eine Datei zu bearbeiten, deren FCB zwangsweise geschlossen wurde, erscheint die Fehlermeldung:

FCB fehlt

Benutzen Ihre Programme keine FCBs, sondern Dateinummern, benötigen Sie diesen Befehl nicht.

FILES = Nummer

Maximale Anzahl gleichzeitig geöffneter Dateien

Bei der Nummer handelt es sich um eine dezimale Zahl zwischen 5 und 99. Sie legt die Anzahl der gleichzeitig zu benutzenden bzw. geöffneten Dateien fest.

Geben Sie den Befehl nicht ein, legt MS-DOS den Wert mit 8 fest. Eine Zahl kleiner als 5 wird nicht beachtet.

Das Betriebssystem benötigt schon zur internen Verwaltung fünf Dateien. Somit können ohne Angabe von FILES drei Dateien von Anwenderprogrammen benutzt werden. Sämtliche Ein- und Ausgabeoperationen des Betriebssystems werden mit Hilfe dieser fünf Dateien ausgeführt.

Ein- und Ausgabe- operationen

Eine geöffnete Datei stellt auch ein gestartetes Programm dar. Ferner benötigt ein installiertes Netzwerk ebenfalls eine bestimmte Anzahl von Dateien. Beachten Sie hierzu unbedingt die Angaben im Handbuch zu Ihren Programmen und einem eventuell installierten Netzwerk.

Wenn Sie eine derartige Angabe im Handbuch zu einem Programm nicht beachten, kann dies Programm nicht korrekt ablaufen.

Für jede über die Standardanzahl (8) hinzugefügte Datei werden weitere 48 Zeichen (Byte) im Speicher benötigt.

Genutzte Dateien von MS-DOS

Von MS-DOS standardmäßig genutzte Dateien

Nummer	Name	Funktion	Bemerkungen
0000	STDIN	Standardeingabe	kann umgeleitet werden
0001	STDOUT	Standardausgabe	kann umgeleitet werden
0002	STDERR	Fehlerausgabe	kann nicht umgeleitet werden
0003	STDAUX	Ausgabe auf Modem	kann nicht umgeleitet werden
0004	STDPRN	oder Drucker	kann nicht umgeleitet werden

Standardausgabe ist normalerweise der Bildschirm. Diese Ausgabe können Sie beispielsweise auf den Drucker umleiten:

```
dir > prn
```

Die Standardausgabe wird in diesem Beispiel vom Bildschirm auf den Drucker umgeleitet.

LASTDRIVE = Buchstabe**Maximale Anzahl der logischen Laufwerke**

Sie bestimmen mit diesem Befehl die Zahl der maximal benutzbaren logischen Laufwerke. Da unter MS-DOS Laufwerkbezeichnungen mit einem Buchstaben vorgenommen werden, bestimmt die Eingabe des Buchstabens das letzte gültige Laufwerk.

Ohne Eingabe des Befehles ist E das letzte logisch gültige Laufwerk. Bestimmen Sie eine kleinere Laufwerkangabe als tatsächlich Laufwerke vorhanden sind, wird der Befehl ignoriert.

Diese Angabe wird häufig benötigt, wenn Sie im Netzwerk und/oder mit dem externen Befehl SUBST arbeiten:

```
lastdrive = z
```

```
subst w: c:\daten
```

Mit dieser Angabe kann das Sub-Directory DATEN auf dem physikalischen Laufwerk C mit der Laufwerkbezeichnung W angesprochen werden.

SHELL = Dateiname**Name und Directory eines Kommandointerpreters**

Dieser Befehl veranlaßt MS-DOS, einen Kommandointerpreter zu laden, wie hinter SHELL spezifiziert (statt normalerweise COMMAND.COM).

Dies ermöglicht einem Systemprogrammierer, einen eigenen Kommandointerpreter zu erstellen und zu verwenden.

Da die Angabe bei SHELL den Namen nur für das Laden beim Systemstart verändert, ist noch folgende Angabe nach dem Start Ihres PC notwendig:

```
set comspec=Dateiname
```

Damit wird gewährleistet, daß auch in der weiteren Verarbeitung beim Nachladen des Kommandointerpreters (wird häufig beim Verlassen eines Anwendungsprogrammes automatisch durchgeführt) der eigene verwendet wird.

```
shell = B:\SYSTEM\COMMAND.COM
```

Beim Start Ihres Schneider PC wird COMMAND.COM vom Laufwerk B aus dem Sub-Directory SYSTEM geladen.

Zusätzlich ist nach dem Start noch folgender Befehl erforderlich:

```
set comspec=B:\SYSTEM\COMMAND.COM
```

DRIVPARM = ...

Mit diesem Eintrag in CONFIG.SYS können Sie die vorhandenen Werte für die angeschlossenen Laufwerke ändern.

DRIVPARM = /d:ddd (/c) (/f:ff) (/h:hh) (/n) (/s:ss) (/t:ttt)

- /d** Nummer des externen Laufwerks (zwischen 0 und 255)
 - 0 = Laufwerk A
 - 1 = Laufwerk B
 - 2 = Laufwerk C
 - usw.
- /c** Wenn das Laufwerk während der Arbeit »abgeschlossen« sein soll
- /f** Spezifiziert die Kapazität und Art des Laufwerks:
 - 0 = 320/360 Kbyte
 - 1 = 1,2 Mbyte
 - 2 = 720 Kbyte (ohne Angabe ist dies der Standardwert)
 - 3 = 8 Zoll einfache Dichte
 - 4 = 8 Zoll doppelte Dichte
 - 5 = Festplatte
 - 6 = Bandlaufwerk
 - 7 = anderes
- /h** Anzahl der Köpfe des Laufwerks (zwischen 1 und 99)
- /n** Wenn es sich um ein nicht wechselbares Medium handelt
- /s** Anzahl der Sektoren pro Spur (zwischen 1 und 99)
- /t** Spuren pro Seite (zwischen 1 und 999)

Bitte beachten Sie dazu, daß die Hardware (hier der Controller) die von Ihnen angegebenen Änderungen auch verarbeiten können muß!

DEVICE = Dateiname

Einheitentreiber

Dieser Befehl ruft eine Datei (Device Driver) auf, die als Erweiterung von MSDOS.SYS geladen wird. Diese Datei enthält das Steuerprogramm (Device Driver) für ein Peripheriegerät (Device). Damit wird der Datenfluß zwischen der Zen-

traleinheit und dem Gerät überwacht und falls erforderlich entsprechend verändert.

Drei Dateien erhalten Sie mit Ihrer MS-DOS-Systemdiskette. Dies sind ANSI.SYS für die erweiterten Bildschirm- und Tastaturfunktionen, RAMDRIVE.SYS für die Simulation einer Diskette im RAM und DRIVER.SYS für die Installation von zusätzlichen Laufwerken mit verschiedenen Formaten.

ANSI.SYS

Mit der Datei ANSI.SYS wird die standardmäßig vorhandene Eingabe- und Ausgabeunterstützung des MS-DOS ersetzt.

Damit erhalten Sie eine wesentlich erweiterte Bildschirm- und Tastatursteuerungsmöglichkeit. Die Angabe

```
device = ansi.sys
```

wird von vielen Anwendungsprogrammen benötigt (beachten Sie hierzu bitte die Angaben im Handbuch zu Ihren Programmen). Normalerweise ist es ratsam, diesen Eintrag in der Datei CONFIG.SYS zu machen.

Der Gerätetreiber ANSI.SYS verhält sich wie der normale Treiber für die Eingabe von der Tastatur und der Ausgabe auf den Bildschirm mit dem Namen CON. Seine Besonderheit besteht darin, daß er die standardisierten ANSI-Steuersequenzen (internationale Norm) in maschinenspezifische Befehle umsetzen kann. Dadurch ist es möglich, Programme maschinenunabhängig zu schreiben. Differenzen der Hardware werden durch den Treiber ANSI.SYS ausgeglichen.

Sämtliche Funktionen werden mit dem Wert 27 dezimal oder 1B hexadezimal eingeleitet. Dies wird durch die Zeichenfolge ESC dargestellt. Das Zeichen # ist eine Variable, die mit dem entsprechenden dezimalen Wert einzugeben ist.

Die angegebenen Funktionen von ANSI.SYS können nicht mit BASIC verwendet werden, da diese Programmiersprache den Device-Treiber nicht beachtet.

Im folgenden finden Sie die einzelnen Funktionen der Norm, die in ANSI.SYS vorhanden sind.

Cursorposition (CUP)

```
ESC [ # 1; # 2H
```

Der Befehl CUP bewegt den Cursor zu der durch die Variablen # bestimmten Stelle auf dem Bildschirm. #1 ist die Nummer der Zeile, #2 die der Spalte. Der vorgegebene Wert ist eins, wenn die Variablen nicht angegeben werden. Der Cursor wird

damit in die linke obere Ecke des Bildschirms gesetzt (Home Position).

**Cursor nach oben
(CUU)**

ESC [# A

Der Befehl CUU bewegt den Cursor eine Zeile nach oben, ohne die Spalte zu verändern. Geben Sie die Anzahl der Zeilen (#) nicht an, so wird der Cursor um eine Zeile nach oben bewegt. Der Befehl wird ignoriert, wenn der Cursor bereits in der obersten Zeile steht.

**Cursor nach unten
(CUD)**

ESC [# B

Der Befehl CUD bewegt den Cursor eine Zeile nach unten, ohne die Spalte zu verändern. Geben Sie den Wert (#) nicht an, so wird der Cursor um eine Zeile nach unten bewegt. Der Befehl wird ignoriert, wenn sich der Cursor bereits in der untersten Zeile befindet.

**Cursor nach rechts
(CUF)**

ESC [# C

Der Befehl CUF bewegt den Cursor eine Spalte nach rechts, ohne die Zeilenposition zu verändern. Geben Sie den Wert (#) nicht an, so wird der Cursor um eine Spalte nach rechts bewegt. Der Befehl wird ignoriert, wenn sich der Cursor bereits in der äußersten rechten Spalte befindet.

**Cursor nach links
(CUB)**

ESC [# D

Der Befehl CUB bewegt den Cursor nach links, ohne die Zeilenposition zu verändern. Geben Sie den Wert (#) nicht an, so wird der Cursor um eine Spalte nach links verschoben. Der Befehl wird ignoriert, wenn sich der Cursor bereits in der äußersten linken Spalte befindet.

**Cursor positionieren
(HVP)**

ESC [# 1; # 2 f

Der Befehl HVP bewegt den Cursor zu der durch die Variablen # bestimmten Stelle auf dem Bildschirm. #1 ist die Nummer der Zeile, #2 die der Spalte. Der vorgegebene Wert ist eins, wenn die Variablen nicht angegeben werden; der Cursor wird damit in die linke obere Ecke des Bildschirms gesetzt (Home Position). Dieser Befehl entspricht CUP.

Cursorposition abfragen (DSR)	ESC [6n Der Befehl DSR fragt die momentane Cursorposition am Bildschirm ab. Die Ausgabe erfolgt dadurch automatisch mit dem Befehl CPR (Cursorposition anzeigen).
Cursorposition anzeigen (CPR)	ESC [# ; #R Dieser Befehl zeigt die momentane Cursorposition an. #1 bezeichnet die Zeile, #2 die Spalte.
Cursorposition speichern (SCP)	ESC [s Der Befehl SCP speichert die aktuelle Cursorposition. Die Werte können mit dem Befehl RCP wieder zurückgeholt und der Cursor damit an die ursprüngliche Position verschoben werden.
Cursor an die gespeicherte Position setzen (RCP)	ESC [u Der Befehl RCP setzt den Cursor an die Stelle zurück, an der er war, als der Befehl SCP eingegeben wurde.
Bildschirm löschen (ED)	ESC [2J Der Befehl ED löscht den gesamten Bildschirm und setzt den Cursor in die linke obere Ecke (entspricht dem internen MS-DOS-Befehl CLS).
Zeile löschen (EL)	ESC [K Der Befehl EL löscht ab der momentanen Cursorposition die gesamte restliche Zeile.
Farbwahl (SGR)	ESC [# ; ... ; #m Mit dem Befehl SGR können Sie die Farbe und/oder Darstellungsform der Zeichen am Bildschirm verändern. Die Variablen (#) bestimmen die neuen Darstellungsarten. Nach Eingabe des Befehls SGR werden alle folgenden Zeichen in der neuen Form dargestellt. Erst ein erneuter SGR-Befehl verändert dies wieder.

**Wertetabelle
der Farben bzw.
Darstellungsformen
von Zeichen**

Wert (#)	Farbe
0	normal (schwarz/weiß beim Farbmonitor)
1	intensiv (hellere Darstellungsform)
4	unterstreichen (nur für Monochrome)
5	blinkend
7	invers (weiß/schwarz)
8	Zeichen sind am Bildschirm unsichtbar
30	schwarzer Vordergrund
31	roter Vordergrund
32	grüner Vordergrund
33	brauner (gelber) Vordergrund
34	blauer Vordergrund
35	Vordergrund magenta (lila)
36	Vordergrund cyan (blau)
37	weißer Vordergrund
40	schwarzer Hintergrund
41	roter Hintergrund
42	grüner Hintergrund
43	brauner (gelber) Hintergrund
44	blauer Hintergrund
45	Hintergrund magenta (lila)
46	Hintergrund cyan (blau)
47	weißer Hintergrund

Modus auswählen (SM)

ESC [= #h
ESC [?7h

**Der Befehl SM
verändert den
Bildschirm-Modus**

Modus	Wert (#)	Funktion
Text	0	40 Zeichen/Zeile, 25 Zeilen monochrome
Text	1	40 Zeichen/Zeile, 25 Zeilen Farbbildschirm
Text	2	80 Zeichen/Zeile, 25 Zeilen monochrome
Text	3	80 Zeichen/Zeile, 25 Zeilen Farbbildschirm
Grafik	4	Mittlere Auflösung in Farbe 320 x 200 mit 40 Zeichen/Zeile Textdarstellung
Grafik	5	Mittlere Auflösung monochrome 320 x 200 mit 40 Zeichen/Zeile Textdarstellung
Grafik	6	Hochauflösende Grafik (monochrome) Textdarstellung mit 80 Zeichen/Zeile
	7	Zeichen nach Zeilenende werden in die nächste Zeile geschrieben

Modus zurücksetzen ESC [= # 1
(RM) ESC [?71

Der Befehl RM entspricht dem Befehl SM. Bei Eingabe einer 7 werden Zeichen nach dem Zeilenende unterdrückt.

**Beispiele für die
Anwendung anhand
des Befehls PROMPT
(Kapitel 11)**

```
prompt $e[0;68;"CLS";13p$n$g
```

Die Funktionstaste F10 (68) erhält den Inhalt »CLS <RETURN>«. Wenn Sie anschließend F10 betätigen, wird der Befehl CLS ausgeführt. Mit \$n\$g wird der normale Prompt wieder am Bildschirm dargestellt.

Wenn mit PROMPT die Belegung einer Taste geändert wird, kann diese nur durch einen erneuten Start Ihres PC wieder verändert werden! Beachten Sie, daß dies zu Problemen mit Anwendungsprogrammen führen kann, wenn diese versuchen, beim Start die Tasten ebenfalls neu zu belegen, was nicht möglich ist.

Als Prompt soll die Anzeige der Uhrzeit und des Datums dienen. Der Cursor soll in der Zeile darunter stehen – Beispiel:

```
Zeit = 11.54.34,56
```

```
Datum = 21.09.85
```

—

```
prompt $_Zeit = $t$_Datum = $d$_
```

Der »normale« Prompt wird mit folgenden Befehlen wahlweise erzeugt:

```
prompt $n$g
```

oder

```
prompt
```

Das aktuelle Laufwerk mit der aktuellen Directory-Bezeichnung soll als Prompt angezeigt werden. Der Cursor für die Eingabe steht darunter.

```
prompt $p$_
```

Der Prompt wird negativ (invers) dargestellt. Anschließende Eingaben sollen »normal« am Bildschirm dargestellt werden (\$e[m]):

```
prompt $e[7m$n:$e[m
```

```
prompt $e[33;44m$n$g
```

Wenn Sie einen Farbmonitor haben, ist anschließend der Hintergrund blau und die Schrift (Vordergrund) braun bzw. gelb. Das Zurückschalten auf die »normalen« Farben:

```
prompt $e[37;40m$n$g
```

In der ersten Zeile am Bildschirm sollen das aktuelle Laufwerk und das Directory angezeigt werden. Dies soll immer dann aktualisiert werden, wenn das System ein neues Systemanforderungszeichen darstellt:

```
prompt $e[s$e[1;1f$P$e[u$g
```

RAMDRIVE.SYS

Simulation eines Laufwerks im RAM (Arbeitsspeicher)

Die Datei RAMDRIVE.SYS ist ein funktionaler Blocktreiber, der es ermöglicht, im Arbeitsspeicher ein Diskettenlaufwerk zu simulieren. Dieses Laufwerk kann von Ihnen wie jedes andere normale Disketten- oder Festplattenlaufwerk benutzt werden. Es hat den Vorteil, erheblich schneller zu arbeiten. Sämtliche Operationen werden nur im Arbeitsspeicher ausgeführt. Der Zugriff auf die vergleichsweise langsamen externen Peripheriegeräte entfällt. Die gespeicherten Daten sind allerdings mit jedem Neustart oder nach dem Ausschalten des PC verloren.

```
DEVICE = (d:)(weg)ramdrive.sys (B (S (D (/E))))
```

Vor RAMDRIVE.SYS kann angegeben werden, woher der Treiber für die RAM-Disk geladen wird (Laufwerk und Directory).

B ist die Größe des Laufwerks in Kilobyte (ohne = 64/Minimum = 16).

S ist die Größe eines Sektors in Byte (ohne = 128/möglich sind 128, 256, 512 oder 1024).

D ist die maximale Anzahl der Einträge in das Inhaltsverzeichnis (ohne = 64/Minimum = 2/Maximum = 1024).

Falls Sie Werte eingeben, achten Sie darauf, daß Sie die Angaben immer durch ein Leerzeichen trennen. Bei jeder anderen Schreibweise werden die Zusatzeingaben nicht erkannt und die Standardwerte angenommen.

Rufen Sie den Treiber mehrmals auf, z.B.:

```
DEVICE=C:\SYSTEM\VDISK.SYS 100
```

```
DEVICE=C:\SYSTEM\VDISK.SYS 256
```

wird jedesmal ein weiteres Laufwerk eingerichtet.

Beim Start Ihres Schneider PC erscheint zum Beispiel folgende Meldung:

```
Microsoft RAM Drive Version 1.16
Virtuelle Diskette C:
    Diskettengröße: 64 K
    Sektorgröße: 128 bytes
    Zuordnungseinheit: 1 sectors
    Verzeichnis Einträge: 64
```

Benutzen Sie die RAM-Disk, um ein Laufwerk im Arbeitsspeicher zu simulieren, beachten Sie, daß es sich nur um ein vorübergehendes Laufwerk handelt. Sämtliche dort gespeicherten Dateien werden bei einem Neustart oder Ausschalten des Gerätes gelöscht. Arbeiten Sie mit diesem Laufwerk, sollten Sie also in regelmäßigen Abständen die dort befindlichen Dateien auf eine Diskette bzw. Festplatte kopieren/sichern.

DRIVER.SYS

Anschluß weiterer externer Laufwerke

Sollten Sie zusätzliche Laufwerke an Ihren Schneider PC anschließen und diese nicht standardmäßig mit Ihrem MS-DOS lauffähig sein, so können Sie einen speziellen Treiber in die Datei CONFIG.SYS installieren.

DEVICE = driver.sys /d:ddd (/c) (/f:ff) (/h:hh) (/n) (/s:ss) (/t:ttt)

- /d Nummer des externen Laufwerks (zwischen 0 und 255)
Wenn Sie ein drittes Laufwerk anschließen, so ist dies Nr. 2
- /c Wenn Laufwerk während der Arbeit »abgeschlossen« sein soll
- /f Spezifiziert die Kapazität und Art des Laufwerks:
 - 0 = 320/360 Kbyte
 - 1 = 1,2 Mbyte
 - 2 = 720 Kbyte (ohne Angabe ist dies der Standardwert)
 - 3 = 8 Zoll einfache Dichte
 - 4 = 8 Zoll doppelte Dichte
 - 5 = Festplatte
 - 6 = Bandlaufwerk
 - 7 = anderes
- /h Anzahl der Köpfe des Laufwerks (zwischen 1 und 99)

/n Wenn es sich um ein nicht wechselbares Medium handelt
 /s Anzahl der Sektoren pro Spur (zwischen 1 und 99)
 /t Spuren pro Seite (zwischen 1 und 999)

**Beispiel für den
 Inhalt der Datei
 CONFIG.SYS**

```
DEVICE      = \SYSTEM\ANSI.SYS
DEVICE      = \SYSTEM\RAMDRIVE.SYS 360 512 112
BREAK       = ON
BUFFERS     = 6
FILES       = 12
COUNTRY     = 049
LASTDRIVE  = L
```

Zuerst werden die Dateien ANSI.SYS (für erweiterte Bildschirm- und Tastaturfunktionen) und RAMDRIVE.SYS (RAM-Disk) als Teil von MSDOS.SYS nachgeladen. Das Speicherlaufwerk hat eine Kapazität von 360 Kbyte und entspricht damit einem normalen Diskettenlaufwerk. Es wird im niedrigen Speicherbereich installiert.

Der Befehl BREAK = ON modifiziert das Betriebssystem so, daß die Tastenkombination Ctrl + Break das Programm jederzeit unterbricht.

Zusätzlich sind 6 Zwischenspeicher (insgesamt also 8) für Plattenzugriffe reserviert.

Es können maximal 12 Dateien gleichzeitig geöffnet werden (fünf für das Betriebssystem und sieben für eigene Programme).

Dann wird der Datensatz für Deutschland geladen und ermöglicht damit die Datumseingabe in der Form: Tag-Monat-Jahr. Die Zeit wird im 24-Stunden-Format angezeigt. Die Zahl der logischen Laufwerke wird auf 12 erhöht.

Auf Ihrer GEM-Desktop-Diskette befindet sich das Programm NVR. Mit diesem Programm können Sie verschiedene Systemparameter einstellen und speichern. Diese gehen nach dem Ausschalten Ihres Schneider PC nicht verloren (wegen der Batterie)! Damit kann die Größe der RAM-Disk eingestellt werden usw.

Damit die mit NVR eingestellten Werte nach dem Start Ihres Schneider PC auch zur Verfügung stehen, müssen Sie den Befehl

ANSI

eingeben. Am besten fügen Sie diesen Befehl in die Datei AUTOEXEC.BAT ein.

Für die Einstellungen der RAM-Disk muß das Programm NVR auf die MS-DOS-Diskette kopiert und in CONFIG.SYS folgender Befehl eingefügt werden:

```
device = ramdrive.sys nvr
```

Anhang I – Begriffserklärungen

Im folgenden finden Sie einige wichtige Ausdrücke mit ihrer Erläuterung.

Alphanumerisch	Buchstaben, Ziffern und Sonderzeichen, mit denen Informationen bzw. Daten dargestellt werden.
ANSI	Abkürzung für American National Standard Institute. Dieses Institut setzt Industriestandards. Am häufigsten tritt dieser Name in Verbindung mit Programmiersprachen auf. Beachten Sie auch den Bildschirmtreiber ANSI.SYS.
Anwenderprogramme	Diese stellen Programme für eine bestimmte Aufgabenstellung, wie Finanzbuchhaltung, Textverarbeitung, Auftragsverwaltung usw., dar.
Ausgabe	Das Sichtbarmachen oder Speichern von Ergebnissen einer Computertätigkeit. Eine Ausgabe kann auf den Drucker, Bildschirm, die Diskette usw. erfolgen.
Batch-Befehle	Sammlung von Befehlen in einer Datei mit dem File-Typ BAT. Wenn der File-Name als Befehl eingegeben wird, verarbeitet Ihr Computer Befehl für Befehl.
Booten	Das Laden und Starten des Betriebssystems.
Byte	Damit wird eine Speicherzelle eines Computers bezeichnet. Jedes Zeichen wird mit einem Byte dargestellt.
Centronics-Schnittstelle	Parallele Schnittstelle (meist Druckeranschluß), die sich als weitverbreiteter Standard eingeführt hat.
Cursor	Ein Zeichen (meist ein blinkendes Unterstreichungszeichen), das die aktuelle Position am Bildschirm anzeigt und die nächste Eingabe erwartet.
Datei	Eine Sammlung von Daten (z.B. Text, Buchungen, Aufträge). Diese können erweitert oder geändert werden. Eine Datei ist mit einer Kartei vergleichbar. Auch ein Programm kann als Datei bezeichnet werden.

Directory	Verzeichnis der Dateien. Es können mehrere auf einer Diskette/Festplatte vorhanden sein (Root-Directory/Sub-Directory). Das Directory befindet sich in einem festgelegten Bereich.
Diskette	Eine dünne, biegsame Plastikscheibe mit einer magnetisierten Oberfläche. Sie wird auch als Floppy oder Floppydisk bezeichnet.
Editor	Einfaches Textbearbeitungsprogramm für das Erstellen oder Verändern von Daten (z.B. Texten).
Hardcopy	Erstellen einer 1:1-Kopie vom Bildschirm auf den Drucker.
Hardware	Der »sichtbare« Teil Ihres Computers. Alle physischen Teile.
Peripherie	Alle an einen Computer angeschlossenen Geräte (auch Diskette, Festplatte und Bildschirm).
RAM	Abkürzung für Random Access Memory. Ein flüchtiger Speicher. Nach dem Abschalten des Computers wird alles gelöscht. Beim Start wird das MS-DOS dorthin geladen (externe Befehle und Programme werden von der Diskette/Festplatte in den RAM zur Ausführung geladen).
ROM	Abkürzung für Read Only Memory. Ein Speicher, in dem die darin enthaltenen Daten/Programme nur gelesen werden können. Die Informationen werden beim Ausschalten des Computers nicht gelöscht.
Software	Alle Programme, die in einem Computer bearbeitet werden können. Erst durch die Software können Sie mit Ihrem PC arbeiten.
Speicher oder Arbeitsspeicher	Es handelt sich um den im Computer eingebauten Speicher, in dem Informationen aufbewahrt werden (Programme und Daten). Siehe auch RAM und ROM.
Winchester oder Festplatte	Die Bezeichnung bezieht sich auf eine bestimmte Technologie einer Festplatte. Es handelt sich um mehrere »harte Disketten«, welche als Stapel in einem staubdichten Gehäuse untergebracht sind. Dieses System zeichnet sich durch eine höhere Schnelligkeit und Sicherheit aus.

Anhang II – Zeichendarstellung/Code-Tabelle

N U L		000 _b ^@ 000 _b 00 _H 00000000 _b	D L E		016 _b ^P 020 _b 10 _H 00010000 _b		032 _b ' ' 040 _b 20 _H 00100000 _b		048 _b '0' 060 _b 30 _H 00110000 _b
S O H		001 _b ^A 001 _b 01 _H 00000001 _b	D C 1		017 _b ^Q 021 _b 11 _H 00010001 _b		033 _b '!' 041 _b 21 _H 00100001 _b		049 _b '1' 061 _b 31 _H 00110001 _b
S T X		002 _b ^B 002 _b 02 _H 00000010 _b	D C 2		018 _b ^R 022 _b 12 _H 00010010 _b		034 _b ' "' 042 _b 22 _H 00100010 _b		050 _b '2' 062 _b 32 _H 00110010 _b
E T X		003 _b ^C 003 _b 03 _H 00000011 _b	D C 3		019 _b ^S 023 _b 13 _H 00010011 _b		035 _b '#' 043 _b 23 _H 00100011 _b		051 _b '3' 063 _b 33 _H 00110011 _b
E O T		004 _b ^D 004 _b 04 _H 00000100 _b	D C 4		020 _b ^T 024 _b 14 _H 00010100 _b		036 _b '\$' 044 _b 24 _H 00100100 _b		052 _b '4' 064 _b 34 _H 00110100 _b
E N Q		005 _b ^E 005 _b 05 _H 00000101 _b	N A K		021 _b ^U 025 _b 15 _H 00010101 _b		037 _b '%' 045 _b 25 _H 00100101 _b		053 _b '5' 065 _b 35 _H 00110101 _b
A C K		006 _b ^F 006 _b 06 _H 00000110 _b	S Y N		022 _b ^V 026 _b 16 _H 00010110 _b		038 _b '&' 046 _b 26 _H 00100110 _b		054 _b '6' 066 _b 36 _H 00110110 _b
B E L		007 _b ^G 007 _b 07 _H 00000111 _b	E T B		023 _b ^W 027 _b 17 _H 00010111 _b		039 _b ' '' 047 _b 27 _H 00100111 _b		055 _b '7' 067 _b 37 _H 00110111 _b
B S		008 _b ^H 010 _b 08 _H 00001000 _b	C A N		024 _b ^X 030 _b 18 _H 00011000 _b		040 _b '(' 050 _b 28 _H 00101000 _b		056 _b '8' 070 _b 38 _H 00111000 _b
H T		009 _b ^I 011 _b 09 _H 00001001 _b	E M		025 _b ^Y 031 _b 19 _H 00011001 _b		041 _b ')' 051 _b 29 _H 00101001 _b		057 _b '9' 071 _b 39 _H 00111001 _b
L F		010 _b ^J 012 _b 0A _H 00001010 _b	S U B		026 _b ^Z 032 _b 1A _H 00011010 _b		042 _b '*' 052 _b 2A _H 00101010 _b		058 _b ':' 072 _b 3A _H 00111010 _b
V T		011 _b ^K 013 _b 0B _H 00001011 _b	E S C		027 _b ^[_ 033 _b 1B _H 00011011 _b		043 _b '+' 053 _b 2B _H 00101011 _b		059 _b ';' 073 _b 3B _H 00111011 _b
F F		012 _b ^L 014 _b 0C _H 00001100 _b	F S		028 _b ^\ 034 _b 1C _H 00011100 _b		044 _b ',' 054 _b 2C _H 00101100 _b		060 _b '<' 074 _b 3C _H 00111100 _b
C R		013 _b ^M 015 _b 0D _H 00001101 _b	G S		029 _b ^] 035 _b 1D _H 00011101 _b		045 _b '-' 055 _b 2D _H 00101101 _b		061 _b '=' 075 _b 3D _H 00111101 _b
S O		014 _b ^N 016 _b 0E _H 00001110 _b	R S		030 _b ^_ 036 _b 1E _H 00011110 _b		046 _b '.' 056 _b 2E _H 00101110 _b		062 _b '>' 076 _b 3E _H 00111110 _b
S I		015 _b ^O 017 _b 0F _H 00001111 _b	U S		031 _b ^` 037 _b 1F _H 00011111 _b		047 _b '/' 057 _b 2F _H 00101111 _b		063 _b '?!' 077 _b 3F _H 00111111 _b

Bild II.1: PC-Code-Tabelle
mit Zeichendarstellung

**Bild II.1: PC-Code-Tabelle
mit Zeichendarstellung
(Fortsetzung)**

Q	064 ₀ 'š' 100 ₀ 40 _H 01000000 _B	P	080 ₀ 'P' 120 ₀ 50 _H 01010000 _B	‘	096 ₀ '‘' 140 ₀ 60 _H 01100000 _B	p	112 ₀ 'p' 160 ₀ 70 _H 01110000 _B
A	065 ₀ 'A' 101 ₀ 41 _H 01000001 _B	Q	081 ₀ 'Q' 121 ₀ 51 _H 01010001 _B	a	097 ₀ 'a' 141 ₀ 61 _H 01100001 _B	q	113 ₀ 'q' 161 ₀ 71 _H 01110001 _B
B	066 ₀ 'B' 102 ₀ 42 _H 01000010 _B	R	082 ₀ 'R' 122 ₀ 52 _H 01010010 _B	b	098 ₀ 'b' 142 ₀ 62 _H 01100010 _B	r	114 ₀ 'r' 162 ₀ 72 _H 01110010 _B
C	067 ₀ 'C' 103 ₀ 43 _H 01000011 _B	S	083 ₀ 'S' 123 ₀ 53 _H 01010011 _B	c	099 ₀ 'c' 143 ₀ 63 _H 01100011 _B	s	115 ₀ 's' 163 ₀ 73 _H 01110011 _B
D	068 ₀ 'D' 104 ₀ 44 _H 01000100 _B	T	084 ₀ 'T' 124 ₀ 54 _H 01010100 _B	d	100 ₀ 'd' 144 ₀ 64 _H 01100100 _B	t	116 ₀ 't' 164 ₀ 74 _H 01110100 _B
E	069 ₀ 'E' 105 ₀ 45 _H 01000101 _B	U	085 ₀ 'U' 125 ₀ 55 _H 01010101 _B	e	101 ₀ 'e' 145 ₀ 65 _H 01100101 _B	u	117 ₀ 'u' 165 ₀ 75 _H 01110101 _B
F	070 ₀ 'F' 106 ₀ 46 _H 01000110 _B	V	086 ₀ 'V' 126 ₀ 56 _H 01010110 _B	f	102 ₀ 'f' 146 ₀ 66 _H 01100110 _B	v	118 ₀ 'v' 166 ₀ 76 _H 01110110 _B
G	071 ₀ 'G' 107 ₀ 47 _H 01000111 _B	W	087 ₀ 'W' 127 ₀ 57 _H 01010111 _B	g	103 ₀ 'g' 147 ₀ 67 _H 01100111 _B	w	119 ₀ 'w' 167 ₀ 77 _H 01110111 _B
H	072 ₀ 'H' 110 ₀ 48 _H 01001000 _B	X	088 ₀ 'X' 130 ₀ 58 _H 01011000 _B	h	104 ₀ 'h' 150 ₀ 68 _H 01101000 _B	x	120 ₀ 'x' 170 ₀ 78 _H 01111000 _B
I	073 ₀ 'I' 111 ₀ 49 _H 01001001 _B	Y	089 ₀ 'Y' 131 ₀ 59 _H 01011001 _B	i	105 ₀ 'i' 151 ₀ 69 _H 01101001 _B	y	121 ₀ 'y' 171 ₀ 79 _H 01111001 _B
J	074 ₀ 'J' 112 ₀ 4A _H 01001010 _B	Z	090 ₀ 'Z' 132 ₀ 5A _H 01011010 _B	j	106 ₀ 'j' 152 ₀ 6A _H 01101010 _B	z	122 ₀ 'z' 172 ₀ 7A _H 01111010 _B
K	075 ₀ 'K' 113 ₀ 4B _H 01001011 _B	[	091 ₀ 'Ä' 133 ₀ 5B _H 01011011 _B	k	107 ₀ 'k' 153 ₀ 6B _H 01101011 _B	{	123 ₀ 'ä' 173 ₀ 7B _H 01111011 _B
L	076 ₀ 'L' 114 ₀ 4C _H 01001100 _B	\	092 ₀ 'ö' 134 ₀ 5C _H 01011100 _B	l	108 ₀ 'l' 154 ₀ 6C _H 01101100 _B	!	124 ₀ 'ö' 174 ₀ 7C _H 01111100 _B
M	077 ₀ 'M' 115 ₀ 4D _H 01001101 _B	]	093 ₀ 'ü' 135 ₀ 5D _H 01011101 _B	m	109 ₀ 'm' 155 ₀ 6D _H 01101101 _B	}	125 ₀ 'ü' 175 ₀ 7D _H 01111101 _B
N	078 ₀ 'N' 116 ₀ 4E _H 01001110 _B	^	094 ₀ 'ˆ' 136 ₀ 5E _H 01011110 _B	n	110 ₀ 'n' 156 ₀ 6E _H 01101110 _B	~	126 ₀ 'ˆ' 176 ₀ 7E _H 01111110 _B
O	079 ₀ 'O' 117 ₀ 4F _H 01001111 _B	_	095 ₀ '˘' 137 ₀ 5F _H 01011111 _B	o	111 ₀ 'o' 157 ₀ 6F _H 01101111 _B	△	127 ₀ '˘' 177 ₀ 7F _H 01111111 _B

**Bild II.1: PC-Code-Tabelle
mit Zeichendarstellung
(Fortsetzung)**

	128 _D ^@ 200 _D 80 _H 10000000 _B		144 _D ^P 220 _D 90 _H 10010000 _B		160 _D ' ' 240 _D A0 _H 10100000 _B		176 _D '0' 260 _D B0 _H 10110000 _B
	129 _D ^A 201 _D 81 _H 10000001 _B		145 _D ^Q 221 _D 91 _H 10010001 _B		161 _D '!' 241 _D A1 _H 10100001 _B		177 _D '1' 261 _D B1 _H 10110001 _B
	130 _D ^B 202 _D 82 _H 10000010 _B		146 _D ^R 222 _D 92 _H 10010010 _B		162 _D ' ' 242 _D A2 _H 10100010 _B		178 _D '2' 262 _D B2 _H 10110010 _B
	131 _D ^C 203 _D 83 _H 10000011 _B		147 _D ^S 223 _D 93 _H 10010011 _B		163 _D ' #' 243 _D A3 _H 10100011 _B		179 _D '3' 263 _D B3 _H 10110011 _B
	132 _D ^D 204 _D 84 _H 10000100 _B		148 _D ^T 224 _D 94 _H 10010100 _B		164 _D '\$ ' 244 _D A4 _H 10100100 _B		180 _D '4' 264 _D B4 _H 10110100 _B
	133 _D ^E 205 _D 85 _H 10000101 _B		149 _D ^U 225 _D 95 _H 10010101 _B		165 _D '% ' 245 _D A5 _H 10100101 _B		181 _D '5' 265 _D B5 _H 10110101 _B
	134 _D ^F 206 _D 86 _H 10000110 _B		150 _D ^V 226 _D 96 _H 10010110 _B		166 _D '& ' 246 _D A6 _H 10100110 _B		182 _D '6' 266 _D B6 _H 10110110 _B
	135 _D ^G 207 _D 87 _H 10000111 _B		151 _D ^W 227 _D 97 _H 10010111 _B		167 _D ' ' ' 247 _D A7 _H 10100111 _B		183 _D '7' 267 _D B7 _H 10110111 _B
	136 _D ^H 210 _D 88 _H 10001000 _B		152 _D ^X 230 _D 98 _H 10011000 _B		168 _D ' (' 250 _D A8 _H 10101000 _B		184 _D '8' 270 _D B8 _H 10111000 _B
	137 _D ^I 211 _D 89 _H 10001001 _B		153 _D ^Y 231 _D 99 _H 10011001 _B		169 _D ') ' 251 _D A9 _H 10101001 _B		185 _D '9' 271 _D B9 _H 10111001 _B
	138 _D ^J 212 _D 8A _H 10001010 _B		154 _D ^Z 232 _D 9A _H 10011010 _B		170 _D ' * ' 252 _D AA _H 10101010 _B		186 _D ' ' ' 272 _D BA _H 10111010 _B
	139 _D ^K 213 _D 8B _H 10001011 _B		155 _D ^[233 _D 9B _H 10011011 _B		171 _D ' + ' 253 _D AB _H 10101011 _B		187 _D ' ; ' 273 _D BB _H 10111011 _B
	140 _D ^L 214 _D 8C _H 10001100 _B		156 _D ^\ 234 _D 9C _H 10011100 _B		172 _D ' , ' 254 _D AC _H 10101100 _B		188 _D ' ' < ' 274 _D BC _H 10111100 _B
	141 _D ^M 215 _D 8D _H 10001101 _B		157 _D ^] 235 _D 9D _H 10011101 _B		173 _D ' - ' 255 _D AD _H 10101101 _B		189 _D ' ' = ' 275 _D BD _H 10111101 _B
	142 _D ^N 216 _D 8E _H 10001110 _B		158 _D ^~ 236 _D 9E _H 10011110 _B		174 _D ' ' ' 256 _D AE _H 10101110 _B		190 _D ' ' > ' 276 _D BE _H 10111110 _B
	143 _D ^O 217 _D 8F _H 10001111 _B		159 _D ^_ 237 _D 9F _H 10011111 _B		175 _D ' / ' 257 _D AF _H 10101111 _B		191 _D ' ' ? ' 277 _D BF _H 10111111 _B

**Bild II.1: PC-Code-Tabelle
mit Zeichendarstellung
(Schluß)**

	192 _D 'Š' 300 _O C0 _H 11000000 _B		208 _D 'P' 320 _O D0 _H 11010000 _B		224 _D 'ı' 340 _O E0 _H 11100000 _B		240 _D 'p' 360 _O F0 _H 11110000 _B
	193 _D 'A' 301 _O C1 _H 11000001 _B		209 _D 'Q' 321 _O D1 _H 11010001 _B		225 _D 'a' 341 _O E1 _H 11100001 _B		241 _D 'q' 361 _O F1 _H 11110001 _B
	194 _D 'B' 302 _O C2 _H 11000010 _B		210 _D 'R' 322 _O D2 _H 11010010 _B		226 _D 'b' 342 _O E2 _H 11100010 _B		242 _D 'r' 362 _O F2 _H 11110010 _B
	195 _D 'C' 303 _O C3 _H 11000011 _B		211 _D 'S' 323 _O D3 _H 11010011 _B		227 _D 'c' 343 _O E3 _H 11100011 _B		243 _D 's' 363 _O F3 _H 11110011 _B
	196 _D 'D' 304 _O C4 _H 11000100 _B		212 _D 'T' 324 _O D4 _H 11010100 _B		228 _D 'd' 344 _O E4 _H 11100100 _B		244 _D 't' 364 _O F4 _H 11110100 _B
	197 _D 'E' 305 _O C5 _H 11000101 _B		213 _D 'U' 325 _O D5 _H 11010101 _B		229 _D 'e' 345 _O E5 _H 11100101 _B		245 _D 'u' 365 _O F5 _H 11110101 _B
	198 _D 'F' 306 _O C6 _H 11000110 _B		214 _D 'V' 326 _O D6 _H 11010110 _B		230 _D 'f' 346 _O E6 _H 11100110 _B		246 _D 'v' 366 _O F6 _H 11110110 _B
	199 _D 'G' 307 _O C7 _H 11000111 _B		215 _D 'W' 327 _O D7 _H 11010111 _B		231 _D 'g' 347 _O E7 _H 11100111 _B		247 _D 'w' 367 _O F7 _H 11110111 _B
	200 _D 'H' 310 _O C8 _H 11001000 _B		216 _D 'X' 330 _O D8 _H 11011000 _B		232 _D 'h' 350 _O E8 _H 11101000 _B		248 _D 'x' 370 _O F8 _H 11111000 _B
	201 _D 'I' 311 _O C9 _H 11001001 _B		217 _D 'Y' 331 _O D9 _H 11011001 _B		233 _D 'i' 351 _O E9 _H 11101001 _B		249 _D 'y' 371 _O F9 _H 11111001 _B
	202 _D 'J' 312 _O CA _H 11001010 _B		218 _D 'Z' 332 _O DA _H 11011010 _B		234 _D 'j' 352 _O EA _H 11101010 _B		250 _D 'z' 372 _O FA _H 11111010 _B
	203 _D 'K' 313 _O CB _H 11001011 _B		219 _D 'Ä' 333 _O DB _H 11011011 _B		235 _D 'k' 353 _O EB _H 11101011 _B		251 _D 'ä' 373 _O FB _H 11111011 _B
	204 _D 'L' 314 _O CC _H 11001100 _B		220 _D 'Ö' 334 _O DC _H 11011100 _B		236 _D 'l' 354 _O EC _H 11101100 _B		252 _D 'ö' 374 _O FC _H 11111100 _B
	205 _D 'M' 315 _O CD _H 11001101 _B		221 _D 'Ü' 335 _O DD _H 11011101 _B		237 _D 'm' 355 _O ED _H 11101101 _B		253 _D 'ü' 375 _O FD _H 11111101 _B
	206 _D 'N' 316 _O CE _H 11001110 _B		222 _D 'İ' 336 _O DE _H 11011110 _B		238 _D 'n' 356 _O EE _H 11101110 _B		254 _D 'ß' 376 _O FE _H 11111110 _B
	207 _D 'O' 317 _O CF _H 11001111 _B		223 _D 'ı' 337 _O DF _H 11011111 _B		239 _D 'o' 357 _O EF _H 11101111 _B		255 _D 'ö' 377 _O FF _H 11111111 _B

Erläuterung zur Tabelle

In der Tabelle finden Sie zu jedem Zeichen verschiedene Werte:

Links im Kästchen das ASCII-Zeichen (US-Tastatur)

Ferner in den 3 Zeilen daneben:

Dezimalwert des Zeichens (000D)

Zeichendarstellung, wenn KEYBGR geladen ist (^@)

Oktale Darstellung (000O)

Hexadezimale Darstellung (00H)

Binäre Darstellung (00000000B)

In der Tabelle finden Sie alle mit PCs darstellbaren Zeichen. Sie können diese in Verbindung mit der Umrechnungstabelle in Anhang III benutzen.

Jedes Zeichen können Sie am Bildschirm in Verbindung mit der Taste Alt darstellen. Dazu muß erst der dezimale Wert eines Zeichens errechnet werden.

Beispiele

Zeichen	HEX-Wert	Umrechnung	Dezimalwert	Zeichenerstellung
	9C	9 = 144 C = 12	156	Alt+156
ä	FB	F = 240 B = 11	251	Alt+251
	9E	9 = 144 E = 14	158	Alt+158
:	BA	B = 176 A = 10	186	Alt+186
§	15	1 = 16 5 = 5	021	Alt+21

Um die Zeichen in der ersten Spalte zu erhalten, drücken Sie die Taste Alt und halten diese fest. Dazu tippen Sie die bis zu drei Ziffern auf der rechten Seite Ihrer Tastatur (numerischer Tastenblock). Nach Loslassen der Alt-Taste wird das Zeichen am Bildschirm angezeigt.

Bitte beachten Sie, daß nicht jeder Drucker alle Zeichen darstellen kann.

Nicht in jedem Anwendungsprogramm können mit der Alt-Taste alle möglichen Zeichen dargestellt werden.

Anhang III – Umrechnungstabellen

HEX	DEZ	HEX	DEZ	HEX	DEZ	HEX	DEZ
0	0	0	0	0	0	0	0
1	4.096	1	256	1	16	1	1
2	8.192	2	512	2	32	2	2
3	12.288	3	768	3	48	3	3
4	16.384	4	1.024	4	64	4	4
5	20.480	5	1.280	5	80	5	5
6	24.576	6	1.536	6	96	6	6
7	28.672	7	1.792	7	112	7	7
8	32.768	8	2.048	8	128	8	8
9	36.864	9	2.304	9	144	9	9
A	40.960	A	2.560	A	160	A	10
B	45.056	B	2.816	B	176	B	11
C	49.152	C	3.072	C	192	C	12
D	53.248	D	3.328	D	208	D	13
E	57.344	E	3.584	E	224	E	14
F	61.440	F	3.840	F	240	F	15

Bild III.1:
*Umrechnungstabelle
Hexadezimal/Dezimal*

Beispiele für die Umrechnung von Zahlen

Hexadezimale Zahl	HEX/DEZ	DEZ/HEX	Dezimale Zahl
EB	E = 224 B = 11 = 235	235–224(E) = 11 11– 11(B) = 0	235
F3	F = 240 3 = 3 = 243	243–240(F) = 3 3– 3(3) = 0	243
7E	7 = 112 E = 14 = 126	126–112(7) = 14 14– 14(E) = 0	126

Hexadezimale Zahlen werden aufgeteilt. Wenn eine solche Zahl zwei Stellen hat, suchen Sie den ersten Wert in der vorletzten und den zweiten in der letzten Spalte und addieren diese, um den dezimalen Wert zu erhalten. Bei vierstelligen Zahlen beginnen Sie in der viertletzten Spalte, usw.

Sie rechnen eine dezimale Zahl um, indem Sie die nächstniedrigere Zahl in der Tabelle suchen (und merken das erste Zeichen vor), diese subtrahieren und für das Ergebnis die nächstniedrigere Zahl (bzw. bei Gleichheit die gleiche) aus der Tabelle suchen.

Beispiel $\text{Zahl } 2764 = 2764 - 2560(\text{A}) = 204 - 192(\text{C}) = 12 - 12(\text{C}) = 0 = > \text{ACC}$
 $\text{Zahl } 2050 = 2050 - 2048(8) = 2 - 0(0) = 2 - 2(2) = 0 = > 802$

Anhang IV – Netzwerk (MS-NET)

PCs im Verbund Ein Netzwerk ist einfach ausgedrückt eine Punkt-zu-Punkt-Verbindung. Es werden mehrere Schneider PCs durch Kabel verbunden. Normalerweise benutzen die zusammengeschlossenen PCs eine oder mehrere gemeinsame Festplatten. Ein Computer im Verbund dient dabei als sogenannter Server (Zentrale).

MS-DOS 3.2 bietet die Betriebssystem-Grundlage für den Einsatz eines Netzwerkes. Zum Betrieb ist zusätzlich eine Netzwerk-Betriebssoftware (z.B. MS-NET) notwendig.

Hardware zum Netzwerk Außerdem benötigt man noch entsprechende Hardware, die die physikalische Verbindung bzw. die Hardware-Ebene der Kommunikation darstellt (z.B. von 3com).

Damit MS-DOS 3.2 mit seinen Netzwerk-Funktionen auch genutzt werden kann, muß die Netzwerk-Betriebssoftware diese unterstützen. Ferner können Programme, welche für Netzwerke erstellt wurden, die von MS-DOS zur Verfügung gestellten Funktionen nutzen.

Record-Locking Mit MS-DOS 3.2 ist es möglich, das sogenannte Record-Locking zu benutzen. Darunter versteht man, daß ein von einem Benutzer von einer Festplatte gelesener Datensatz (z.B. eine Kundenadresse) nicht von einem anderen benutzt werden kann. Hierbei unterscheidet man:

Nur Lesen möglich

Nur Schreiben möglich

Lesen und Schreiben möglich

usw.

Netzwerk-Server Zu beachten ist, daß die »Zentrale« (der Server) während des Betriebs von MS-NET nicht als Arbeitsplatz genutzt werden soll. Er verwaltet das gesamte Netzwerk. Verschiedene Netzwerke bieten die Möglichkeit, nicht nur die Laufwerke der

Zentrale zu nutzen, sondern auch auf solche einzelnen Arbeitsplätze zuzugreifen.

**Kauf von Netzwerk-
Software**

Beim Kauf von Programmen müssen Sie darauf achten, daß diese auch Ihr Netzwerk voll unterstützen. Software, welche dies nicht gewährleistet, kann beim Betrieb zu erheblichen Problemen führen.

Für das Verständnis von Netzwerken und deren Handhabung sind detaillierte Kenntnisse in MS-DOS und der Netzwerk-Architektur notwendig. Bitte beachten Sie hierzu die einschlägige Literatur.

Stichwortverzeichnis

8088-Prozessor 112

A

Adreßregister 116
 Akkumulator 116
 Akustikkoppler 12, 55, 84
 Alphanumerisch 219
 Anschlüsse
 - am PC 12
 ANSI 219
 ANSI-Steuersequenzen 210 ff.
 ANSI.SYS 210 ff.
 Anwenderprogramme 219
 Anwendungsmöglichkeiten
 - des Schneider PC 13
 Anwendungsprogramme 37
 Anwendungssoftware 13
 APPEND 64, 66, 87 f.
 Archivbyte 90 f., 93 f.
 ASCII-File 107
 ASCII-Zeichen 150
 Assembler-Code 112
 ASSIGN 89, 123, 154, 183
 ATTRIB 90 f., 93 f., 119
 AUX 84, 110

B

back-slash 63
 BACKUP 58 f., 90 ff.,
 154, 174, 183
 Basiszeiger 116
 BAT-Dateien 25 f.
 Batch-Befehle 195 ff.
 Batch-Datei 127, 145, 149, 152,
 164, 170, 177, 180
 Batch-File 94
 Baudrate 160
 Befehle
 - Interne 25
 - Externe 25, 28
 - Ausführung der 25
 Befehlsanforderungszeichen 26
 Befehlsinterpreter 100, 177
 Befehlskette 80
 Befehlsnamen 32, 65
 Befehlsverkettung 77, 79, 81
 Befehlswiederholung 22
 Befehlszeile 19, 79
 Bereiche (Sektoren) 32

Betriebssystem 12, 25, 146
 Betriebssystem-Kopien 40
 Bildschirm 15, 56, 77 f., 83,
 110
 - konfigurieren 159
 Bildschirm-Modus 213
 Bildschirmanzeige 22
 Bildschirmausdruck 151
 Bildschirmausgabe
 - auf Drucker umleiten 55
 Bildschirminhalt
 - auf Drucker ausgeben 56
 Bit 32
 Blocktreiber 83
 Boot-Loader 16, 146, 203
 Booten 184
 BREAK 95, 204
 BUFFERS 155, 204
 Byte 32

C

Charaktertreiber 83
 CHDIR (CD - CHange
 DIRectory) 62, 64, 67, 96,
 158
 CHKDSK 97 ff.
 CLS 99
 COBOL 13
 Code-Segment-Register 112
 Codesegment 116 f.
 COM-Dateien 25 f.
 COM1 78, 84, 110, 160
 COM2 110, 160
 COMMAND 16, 25 f., 28, 44,
 72, 100 f.
 COMP 102 ff.
 Compact-Drucker 151
 CON 84, 100, 108, 110
 Concatenation 107 ff.
 CONFIG.SYS 57, 95, 203 ff.
 COPY 25, 45, 50 ff., 59, 64,
 71 f., 94, 102, 107 ff., 123, 192
 COUNTRY 205
 CP/M-86 13
 CTTY 100, 110
 Cursor 45, 99
 Cursorfunktionen 210 ff.

D

DATE 39, 111
 Datei-Einträge 52
 Dateianlage 61
 Dateideckungszeichen 102, 107,
 129

Dateinamen 47 f., 87
 Dateisteuerblöcke 206
 Daten
 - löschen 34
 - überschreiben 34
 Datenfernübertragung 12
 Datensegment 116
 Datensicherung 59
 Datentransport 83
 Datenumleitung 77 ff.
 Datum 39, 111
 DEBUG 112 ff.
 DEL (ERASE) 45, 51, 54, 64, 68,
 70 f., 119
 Device 110
 DEVICE 167, 209 f.
 Dienstprogramme
 - von MS-DOS 25, 28, 44
 DIR 45, 51, 64, 69 f., 78, 81, 101,
 120 ff.
 Directory (Inhaltsverzeichnis)
 32, 45, 48, 51, 54, 66, 80, 87,
 92, 97, 108, 119, 186
 - Anzeige des 62, 78
 - Aufbau des 51, 61
 - Namen 96
 - Wechseln des 62
 - Weg 63
 Directory-Struktur 65
 DISKCOMP 89, 102, 123 f.,
 154, 183
 DISKCOPY 40 f., 89, 102,
 125 f., 154, 183
 Diskette 15, 28, 33, 59, 92
 Diskettenlaufwerk 11, 15
 DOODLE 13
 DOS Plus 13
 DOS-Partition 43, 139
 DOS-System
 - Übertragen des 44
 DRIVER.SYS 210, 216
 DRIVPARAM 209
 Drucken 25
 Drucker 15, 55 f., 77, 79, 83, 159

E

ECHO 127 f.
 Editor 129
 EDLIN 129 ff.
 - Befehle 130 ff.
 Einfügemodus 131
 Einheitentreiber 209
 Environment-Bereich 100 f., 177
 Environment-Speicher 155, 163

ERROR LEVEL 152

Escape-Sequenz 167

EXE-Dateien 25 f.

EXE2BIN 137

EXIT 100, 138

Extrasegment 116

F

Farbdrucker 151

FCB (File Control Block) 206

FDISK 42 ff., 139 ff.

Festplatte 12, 14 f., 25, 31, 33,

36, 44 f., 59, 92

– als Speichermedium 36

– Kapazität einer 36

– Einrichten einer 42, 139 ff.

– Einteilung der 44

File-Allocation-Table 146

File-Namen 47 f., 52

File-Sharing 92, 179

– Konflikte 94

File-Typ 47 f., 52, 59, 90, 108

FILES 207

Filter 77, 79, 142, 162

FIND 79, 142 ff.

FOR 145

FORMAT 41 f., 44 f., 61, 89,

126, 141, 146 ff., 183

Formatieren

– von Disketten 41 f., 146 ff.

Formatierungs-Vorgang 44

Fortran 13

Funktionstasten 19, 21, 130

– Belegung 20

G

GEM 13

Gerätename 85

Gerätetreiber 83 f., 177

GOTO 149

Grafik-Drucker 151

Grafik-Karte 55

Grafik-Modus 150

Grafikzeichen 151

GRAFTABL 150

GRAPHICS 151

Großbuchstaben 45, 102

GW-BASIC 26

H

Hard-Disk 31

Hardcopy 151

Hardware 12, 15

HEX-Darstellung 106

Hexadezimal 114

Hierarchie 61

Hintergrundfarbe 151

I

IF 152 f.

Indexregister 116

Informationen

– Lesen und Schreiben

von 32

Inhaltsverzeichnis

(Directory) 21, 59

Initialisierungsprogramm

– für Festplatten 139

IO.SYS 16, 44

J

JOIN 123, 154 f., 183

K

Kanaladresse 114

KEYBFR 156

KEYBGR 53, 150, 156, 205

KEYBIT 156

KEYBSP 156

KEYBUK 156

Kilobyte 32

Kleinbuchstaben 45

Kommandoaufbau

– von MS-DOS-Befehlen 85

Kommentar

– am Bildschirm 127, 170

Kompatibilität

– des Schneider PC 17

– von MS-DOS 18

Konfiguration

– von MS-DOS 203 ff.

Kopierbefehl 50

Kopieren 25, 107 ff., 192 f.

Kopiervorgang 40, 102, 125 f.,

192 f.

L

LABEL 51, 146, 157, 183, 191

Landescode 120

Landeskennziffer 205

LASTDRIVE 208

Laufwerk

– Aktuelles 31

– Anwählen eines 31

– Wechseln des 48

Laufwerkbezeichnung 85 f., 89,

183

Laufwerksname 32, 48, 208

Laufwerkumleitung 89

Laufwerkzuweisung 89

Lesen-Schreiben (Read-Write)

90

Locomotive BASIC2 13

Lotus 1-2-3 13

Löschen

– von Dateien/Programmen

119

LPT1 55, 160

LPT2 55, 160

LPT3 55, 160

M

Maschinencode 112

Maus 12

Megabyte 32

MKDIR (MD) 64, 66, 158

MODE 45, 55 f., 150, 159

Modem 12, 55, 83, 110

Modula 13

Monitor 12

MORE 79, 81, 162

MS-DOS 13, 16, 18

– Aufgaben von 15

– als Standard-Betriebssystem

16

– Installation von 40

– Laden von 15

– Starten 38

– Starten von Festplatte 39

– Umgang mit 37

MS-DOS-Befehle 20

– Beschreibung aller 85

MSDOS.SYS 16, 44

Multiplan 13

N

Namenszusatz 22, 47

Netzwerk 80, 157, 179, 206 f.

Netzwerk-Laufwerk 89, 97, 123,

125, 154, 169, 183 f.

Netzwerkbetrieb 92

Nur-Lesen (Read-Only) 90

NVR 217

P

Parameter 85

Parity 160

Partitions

– Anzeige der 44

Pascal 13

PATH (Weg) 64 f., 100, 163

PAUSE 164

PC-DOS 16
 Peripherie 15, 55, 83
 PIPEs 80
 Plotter 12, 15
 PRINT 57, 89, 165
 - Warteschlange 165 f.
 - Wartezeit 166
 PRN 79, 84, 108
 Programmunterbrechung 95
 Programmzähler 116
 Prompt 15, 26, 31, 39, 111,
 214 ff.
 PROMPT 167 f.

Q
 QUELL-Diskette 40

R
 RAM - Random Access
 Memory 26, 31, 117
 RAM-DISK 86, 215 f.
 RAM-Speicher 98
 RAMDRIVE.SYS 210, 215
 Read-Only (Nur-Lesen) 90 f.,
 172, 174
 Read-Write (Lesen-Schreiben)
 90
 RECOVER 169
 Register 114 ff.
 Registername 116
 REM 170
 RENAME (REN) 45, 51, 53,
 171
 REPLACE 172 f.
 RESTORE 58 f., 90, 94, 154,
 174 f., 183
 RMDIR (RD) 64, 68, 119, 176
 ROM (Read-Only-Memory) 117
 Root-Directory 51, 61 f., 66, 68,
 87, 93, 96

S
 Schneider PC
 - Ausstattung des 37
 Schnittstelle 55, 78, 83, 110
 Schreib-Lese-Kopf 32
 Schreib-Lese-Schlitz 35
 Schreibmaschinen-Tastatur 19
 Schreibschutz 36, 126
 Segment 113
 Sektoren 32 f., 98, 114, 169
 SET 101, 177 f.
 SHARE 92, 179, 183, 206
 SHELL 208

SHIFT 180 f.
 Sicherungsdiskette 92 f.
 Sicherungskopien 57 f.
 Software 12
 Sonderzeichen 47, 151
 SORT 78 f., 81, 182
 Sortieren 182
 Speicher
 - Externer 31
 Speicher (RAM) 15, 25, 28
 Speicheradresse 113
 Speicherinhalt 113, 115
 Speicherkapazität 51
 Speichermedien 31
 Speicherzelle 113
 Sprungbefehl 149
 Spuren (Tracks) 32
 STACK-Segment 137
 Standard
 - Eingabegerät 77, 79
 - Ausgabegerät 77, 79
 Standard-Druckausgabe 95
 Standarddrucker 151
 Standardeinstellungen
 - des Druckers 55
 Stapelsegment 116
 Stapelzeiger 116
 Status-Report 98
 Statusmeldung 97
 Statusregister 116
 Stop-Bit 160
 Sub-Directory 61 f., 66, 68, 78,
 87, 91, 93, 96, 107 f., 119, 154,
 158, 176, 183
 SUBST 123, 154, 183, 208
 Suchreihenfolge
 - der Dateien 26
 Supercalc 13
 SYS 184
 System-Bereitschaftsmeldung 39
 System-Bootup 15
 Systemanforderungszeichen
 167 f.
 Systemdateien 184
 Systemdatum 111
 Systemstart 26, 100, 190, 208

T
 Tabulatorzeichen 102
 Tape-Streamer 58
 Tastatur 12, 19, 25, 77, 78, 80,
 83, 108, 110
 Tastaturbelegung 156, 205, 210
 Tastaturpuffer 19, 22

Tastenfeld 80
 Tastenkombination 23
 Terminal 110
 Testprogramm 38
 Text drucken 57
 Textbearbeitung 130
 Textdatei 131
 TIME 39, 185
 Tintenstrahlfarbdrucker 151
 Tracks 32
 TREE 62, 186 f.
 Treiber 77
 TYPE 66, 81, 87, 93, 188

U
 Uhrzeit 39, 185
 Update
 - von Dateien 172
 Ursprungslaufwerk 85

V
 V.24-Schnittstelle 110
 VER 189
 Vergleich
 - von Disketten 123
 VERIFY 190
 Versionsnummer 189
 VOL 191

W
 Währungssymbol 205
 Weg (PATH) 63, 87
 Wildcard
 - Kopieren mit 50
 - Position 50
 Write protected 34

X
 XCOPY 64, 71 ff., 90 f., 94,
 192 f.
 XENIX 16

Z
 Zählregister 116
 Zeichenkette 131, 142
 Zeichensatzprogramm 156
 Zentraleinheit 11
 ZIEL-Diskette 40
 Zifferntastenfeld 19
 Zugriffszeit 36
 Zwischenspeicherung
 - von Daten 80, 204

SCHULUNG

Praxiserprobte Arbeitsbücher zum Selbststudium und Gruppenunterricht für dBASE III PLUS, WORD 3.0 und LOTUS 1-2-3

- Alle theoretischen Grundlagen werden Ihnen mit praxisorientierten Beispielen und Übungen ausführlich erläutert.

- Dem Lehrmaterial liegt eine Diskette bei, die alle Übungsdateien in verschiedenen Bearbeitungsphasen enthält. So haben Sie jederzeit den sofortigen Anschluß an das Lehrprogramm.

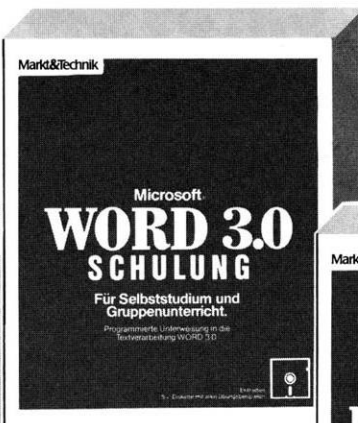
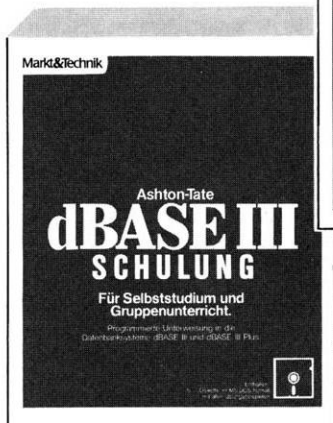
- Für Ihre handschriftliche Beantwortung der Fragen ist ausreichend Platz vorgesehen.

Später können Sie den »Frage-Teil« durch die Lösungen im Anhang ersetzen.

- Wenn Sie das Schulungspaket durchgearbeitet haben, bleibt Ihnen ein komplettes Nachschlagewerk einschließlich umfangreichem Befehlsverzeichnis für die tägliche Praxis.

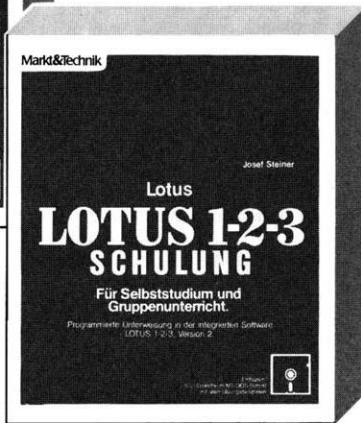
- Software-Anforderung: dBASE III, WORD 3.0, bzw. LOTUS 1-2-3 Version 2

Dr. P. Albrecht
dBASE-III-PLUS-Schulung
1987, 547 Seiten,
inkl. Beispieldiskette
Best.-Nr. 90449
ISBN 3-89090-449-1
DM 98,-/sFr 90,20/öS 764,40



H. Niemeier
WORD-3.0-Schulung
1987, 456 Seiten,
inkl. Beispieldiskette
Best.-Nr. 90450
ISBN 3-89090-450-5
DM 98,-
sFr 90,20/öS 764,40

J. Steiner
LOTUS-1-2-3-Schulung
Version 2
1987, 562 Seiten
inkl. Beispieldiskette
Best.-Nr. 90451
ISBN 3-89090-451-3
DM 98,-/sFr 90,20/öS 764,40




Mark&Technik
Zeitschriften · Bücher
Software · Schulung

Mark&Technik-Produkte erhalten
Sie bei Ihrem Buchhändler, in
Computer-Fachgeschäften oder in den
Fachabteilungen der Warenhäuser

Junior- MicroPro. WordStar. Das Kraftpaket zum Junior-Preis:

Ausgereift und zuverlässig bietet Junior-WordStar mit MailMerge in der Version 3.1 auch Computer-Neulingen alle Leistungsmerkmale einer professionellen Textverarbeitung für PCs und Kompatible:

- Ausführliche Arbeitshilfen am Bildschirm
- Fettdruck, Unterstreichen, Hoch- und Tiefstellen
- Blocksatz, Flatterrand, zentrierte Texte, Einrückungen
- Textbausteine, Tabellen und Zahlenkolonnen
- Korrektur durchgehender Fehler mit Suchen und Ersetzen

DM 399,-*

Mit MailMerge können Sie Serienbriefe mit persönlicher Anrede an eine beliebige Anzahl von Adressen schreiben und auch die Adreßaufkleber drucken. Ebenso können Standardaufsätze zur wiederholten Verwendung abgerufen und Postanschriften und Rechnungssätze aus Datenbanken, wie zum Beispiel Junior-dBASE II, ergänzt werden. MailMerge ermöglicht die Erstellung maßgeschneiderter Serienbriefe mit persönlicher Note per Tastendruck.

Hardware-Anforderung:

IBM-PC oder kompatibler Computer unter PC-DOS/MS-DOS ab Version 2.0, mindestens ein Diskettenlaufwerk, mindestens 256 Kbyte Speicher.

Best.-Nr. 55104

DM 399,-/sFr 345,-/öS 2950,-*

Ergänzende Literatur:

G. Jürgensmeier
WordStar für PCs und Kompatible

Best.-Nr. 90224

DM 59,-/sFr 54,30/öS 460,20

Übrigens:

In der Junior-Serie von Markt&Technik gibt es auch noch dBASE II, Framework, Multiplan und Word - bewährte Standardsoftware für PCs und Kompatible zum Junior-Preis!

Fragen Sie Ihren Computerefachhändler. Oder fordern Sie ausführliche Informationen direkt beim Verlag an.

*inkl. MwSt. Unverbindliche Preisempfehlung



706261

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0
Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollenstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56
ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 67 75 26
Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 48 15 38-0.

Junior- dBASE II. Das Kraftpaket zum Junior-Preis:

dBASE II, das meistverkaufte Programm unter den Datenbanksystemen, eröffnet Ihnen in der Junior-Version (2.43*) optimale Möglichkeiten der Datenverwaltung: Auf einfache Weise können Sie vollständige Datenbanken erzeugen, Daten hinzufügen, löschen, verändern, darstellen und ausdrucken. Die leicht erlernbare, integrierte Programmiersprache erlaubt auch dem unerfahrenen EDV-Anwender, komplette, leistungsfähige Anwendungen wie Finanzbuchhaltung, Lagerhaltung, Kundenverwaltung, Fakturierung usw. selbst zu erstellen.

DM 399,-*

Hardware-Anforderung:

IBM-PC oder kompatibler Computer unter PC-DOS/MS-DOS ab Version 2.0, mindestens ein Diskettenlaufwerk, mindestens 128 Kbyte Speicher.
Best.-Nr. 55103
DM 399,-*/sFr 345,-/öS 2950,-*

Die ideale Ergänzung: dSoft II

- dAdreß: Adreßverwaltung
- dArtikel: Verwaltung von Artikelstammdaten

- dRechnung: Direkt- und Stapelfakturierung mit den Daten aus dAdreß und dArtikel
Sofort einsetzbar, leicht anpaßbar (Quellcode wird mitgeliefert)
Best.-Nr. M&T 500
DM 399,-*/sFr 345,-/öS 2950,-*

Weiterführende Literatur:

Dr. P. Albrecht
Das Datenbanksystem
Junior-dBASE II
Das unentbehrliche Zusatzhandbuch für die tägliche Arbeit mit Junior-dBASE II.
Best.-Nr. 90483
DM 68,-/sFr 62,60/öS 530,40

Übrigens:

In der Junior-Serie von Markt & Technik gibt es auch WordStar, Framework, Multiplan und Word - leistungsstarke Standardsoftware für PCs und Kompatible zum Junior-Preis!
Fragen Sie Ihren Computerfachhändler. Oder fordern Sie ausführliche Informationen direkt beim Verlag an.

*inkl. MwSt. Unverbindliche Preisempfehlung



7002602

Markt & Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0
Bestellungen im Ausland bitte an: SCHWEIZ: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56
ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526
Überreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

Junior- Multiplan. Das Kraftpaket zum Junior-Preis:

Multiplan-Junior ist ein äußerst vielseitiges und benutzerfreundliches Tabellenkalkulationsprogramm für IBM-PCs oder kompatible Computer:

- hervorragend geeignet für Analyse- und Planungsberechnungen.
- Unschlagbar, wenn es um Budgetplanung, Produktkalkulation, Zins- und Zinseszinsberechnung, Kostenstellenrechnung und Kfz-Kostenabrechnung geht.
- Multiplan-Junior ist durch die bedienerfreundliche Menütechnik sehr leicht erlernbar. Also genau das richtige Kalkulationsprogramm für Einsteiger!

DM 299,-*

- Das elektronische Arbeitsblatt umfaßt eine Tabellengröße von 255 Zeilen mit 63 Spalten. Wenn das noch nicht genügt, der kann mehrere Tabellen miteinander verknüpfen.
- Mit speziellen Formatierungs-, Aufbereitungs- und Druckanweisungen können Sie Ihre Daten optimal aufbereiten und perfekte Präsentationsunterlagen erstellen.

Hardware-Anforderung:

IBM-PC oder kompatibler Computer, mindestens 128 Kbyte RAM-Speicher, ein Diskettenlaufwerk, MS-DOS 1.1 oder höhere Version.
Best.-Nr. 55102
DM 299,-*/sFr 269,-/öS 2150,-*

Ergänzende Literatur:

Dr. P. Albrecht
Multiplan deutsch
2. überarb. Auflage 1987
Best.-Nr. 90476
DM 58,-/sFr 53,40/öS 452,40

Übrigens:

In der Junior-Serie von Markt&Technik gibt es noch WordStar, dBASE II, Framework und Word - ausgereifte Standardsoftware für PCs und Kompatible zum Junior-Preis.

Fragen Sie Ihren Computerfachhändler. Oder fordern Sie ausführliche Unterlagen direkt beim Verlag an.

*inkl. MwSt. Unverbindliche Preisempfehlung



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56

ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526

Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

Junior-
Microsoft Word.

Das Kraftpaket zum Junior-Preis:

Eine Idee macht Furore:

Word-Junior – das leistungsfähige, vielseitige und komfortable Textverarbeitungsprogramm für IBM-PCs und Kompatible. Zu einem unglaublichen Preis.

- Mit Word-Junior erledigen Sie mühelos Ihre gesamte Schreibarbeit im Büro oder im privaten Bereich.

- Word-Junior gibt Ihren Texten Format. Sie haben die Wahl zwischen verschiedenen Schrifttypen und -größen (je nach Drucker) sowie Absatz- und Druckformaten (Einrücken, Blocksatz, Flattersatz usw.).

- Gestalten Sie die Seiten ganz nach Ihrem persönlichen

DM 399,-*

Geschmack (Seitenhöhe, Randbegrenzungen, Spaltenzahl, Zwischenräume usw.).

- Mit Word-Junior können Sie bis zu acht Dokumente parallel bearbeiten und gleichzeitig auf dem Bildschirm anzeigen. Wenn Sie Textteile zwischen diesen Dokumenten kopieren oder verschieben wollen, ist das für Word-Junior ein leichtes Spiel.

- Und mit der Maus werden Ihre Befehle sogar noch schneller und effizienter in einem Arbeitsgang erledigt.

Überzeugen Sie sich selbst: Mit Word-Junior besitzen Sie eine Textverarbeitung der Luxusklasse!

Hardware-Anforderung:

IBM-PC oder kompatibler Computer, mindestens 192 Kbyte Speicher, zwei Diskettenlaufwerke oder Festplatte, MS-/PC-DOS ab Version 2.0. Best.-Nr. 55111
DM 399,-*/sFr 345,-/öS 2950,-*

Ergänzende Literatur:

R. Wendel
Textverarbeitung mit
Microsoft Word-Junior
(mit zusätzlichen Druckertreibern auf Diskette)
Best.-Nr. 90235
DM 49,-/sFr 45,10/öS 382,20*

Übrigens:

In der Junior-Serie von Markt&Technik gibt es noch dBASE II, Framework, WordStar, Multiplan - ideale Standardsoftware für Einsteiger zum Junior-Preis.

Fragen Sie Ihren Computerfachhändler. Oder fordern Sie ausführliche Unterlagen direkt beim Verlag an.

*inkl. MwSt. Unverbindliche Preisempfehlung



706264

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56

ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 67 75 26

Überreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 48 15 38-0.

Junior- Framework. Das Kraftpaket zum Junior-Preis:

Dieses integrierte Programmpaket für PCs und kompatible Computer erfüllt alle Ansprüche, die an einen modernen »elektronischen Schreib-« gestellt werden:

- Textverarbeitung.
- Ob Sie einen einseitigen Brief oder einen 50seitigen Bericht erstellen müssen: Mit wenigen Tastenanschlägen können Sie kopieren, löschen, umstellen.
- Tabellenkalkulation.
- Wenn Sie wie die meisten Anwender nicht nur mit Worten, sondern auch mit Zahlen arbeiten, wissen Sie die starke Tabellenkalkulation in Junior-Framework zu schätzen.
- Datenbanksystem.
- Beim Marktführer Ashton-Tate kann

DM 399,-*

man von dem Datenbankteil nur das Beste erwarten. Sie sortieren, aktualisieren, listen nach Kriterien Ihrer Wahl.

- Grafik.
- Torten-, Balken- und Liniengrafik gehören zum Standardrepertoire von Junior-Framework.
- Konzepte.

Ein Ordnungsverfahren, mit dem Sie komplexe Aufgaben schnell und übersichtlich lösen.

- Programmiersprache FRED.
- Mit dieser eingebauten Sprache erstellen Sie selbst komplexe

Anwendungen für Ihren Bedarf. Dabei können Sie fast alle Funktionen der Teilbereiche von Junior-Framework verwenden.

Hardware-Anforderung:

IBM-PC oder kompatibler Computer, 384 Kbyte Speicher, zwei Diskettenlaufwerke oder ein doppel-seitiges Diskettenlaufwerk und eine Festplatte, PC-/MS-DOS ab Version 2.0.

Best.-Nr. 55114

DM 399,-/sFr 345,-/öS 2950,-*

Ergänzende Literatur:

R. Kost

Junior-Framework

Best.-Nr. 90500

DM 59,-/sFr 54,30/öS 460,20

Übrigens:

In der Junior-Serie von Markt&Technik gibt es außerdem WordStar, dBASE II, Multiplan und Word - alle Produkte zum Junior-Preis.

Fragen Sie Ihren Computerfachhändler. Oder fordern Sie ausführliche Unterlagen direkt beim Verlag an.

*inkl. MwSt. Unverbindliche Preisempfehlung



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56

ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526

Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 48 15 38-0.

Schneider PC: MS-DOS 3.2

Der Autor:

HANS H. GERHARDT wurde 1956 in München geboren. Nach Absolvierung des Studiums der Betriebswirtschaftslehre war er fünf Jahre als Anwendungs- und Systemprogrammierer in einem namhaften deutschen Großunternehmen tätig.

Seit rund drei Jahren arbeitet Hans H. Gerhardt als freier Dozent für verschiedene Institute und beschäftigt sich daneben mit der Programmierung von Software.

Mit dem Schneider PC 1512 erhalten Sie das Betriebssystem MS-DOS 3.2. Derzeit werden über 90% aller PCs weltweit mit MS-DOS betrieben. Damit steht Ihnen die größte Software-Bibliothek der Welt zur Verfügung, die unter diesem leicht erlernbaren Betriebssystem lauffähig ist.

Dieses Buch bietet Ihnen eine leichtverständliche Einführung zum Betriebssystem MS-DOS 3.2 Ihres Schneider PC. In kompakter Form werden Ihnen die wichtigsten Funktionen übersichtlich mit vielen Übungsbeispielen dargestellt.

Das Buch gliedert sich in drei Teilbereiche:

1. Eine Einführung, die dem unerfahrenen Anwender grundlegende Betriebssystem-Kenntnisse vermittelt. Hier finden Sie auch einen kurzen Überblick zur Hard- und Software Ihres Schneider PC 1512. Zusätzlich werden Ihnen einige Anwendungsbeispiele wie Kopieren von Disketten, Druckeranpassung und Einrichten von Directory-Unterverzeichnissen vorgestellt.
2. Eine alphabetisch geordnete Übersicht aller dokumentierten MS-DOS-Befehle mit Übungen.

3. Eine Erklärung der Anpassung von MS-DOS 3.2 an die verschiedensten Anforderungen (vor allem CONFIG.SYS).

Der Autor wendet sich mit seinem Buch zum Schneider PC MS-DOS 3.2 nicht nur an den unerfahrenen Anfänger, sondern im zweiten und dritten Teil insbesondere auch an den Fortgeschrittenen. Damit wird es zum ständigen Begleiter eines jeden Schneider-PC-Benutzers.

Hardware-Anforderungen:
Schneider PC 1512

Software-Anforderungen:
Betriebssystem MS-DOS 3.2

ISB N 3-89090-427-0



4 001057 904279

DM 49,-

sFr 45,10

öS 382,20