

Hans H. Gerhardt

DOS 3.3 für PCs und Personal System/2

Eine leichtverständliche Einführung
mit Beispielen, Übungen und
alphabetischer Befehlsübersicht.



DOS 3.3

Hans H. Gerhardt

DOS 3.3

für PCs und Personal System/2

Eine leichtverständliche
Einführung
mit Beispielen,
Übungen
und alphabetischer
Befehlsübersicht

Markt & Technik
Verlag AG

CIP-Titelaufnahme der Deutschen Bibliothek

Gerhardt, Hans H.:

DOS 3.3 für PCs und Personal System 2 :

e. leichtverständl. Einf. mit Beispielen, Übungen u.

alph. Befehlsübersicht / Hans H. Gerhardt. –

Haar bei München : Markt-u.-Technik-Verl., 1988

ISBN 3-89090-547-1

Die Informationen in diesem Produkt 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 Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

MS-DOS und MS-OS/2 sind eingetragene Warenzeichen der Microsoft Corp., USA.

IBM, PC-DOS und Personal System/2 sind eingetragene Warenzeichen der International Business Machines Corp., USA.

15 14 13 12 11 10 9 8
91 90 89

ISBN 3-89090-547-1

© 1988 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	11
1 Einführung	13
1.1 Die PC- und PS/2-Hardware	13
1.2 Die Software	14
1.3 Anwendungsmöglichkeiten	15
2 Was ist DOS?	17
3 Die Kompatibilität von PCs bzw. PS/2 und DOS	19
4 Die Tastatur	21
4.1 Allgemeines	21
4.2 Die Funktionstastenbelegung (Übersicht)	23
4.3 Beispiele für die Anwendung der Funktionstasten	24
4.4 Tastenbezeichnungen bei verschiedenen Tastaturen	27
5 Übersicht über die DOS-Befehle	29
5.1 Allgemeines	29
5.2 Interne Befehle	30
5.3 Externe Befehle	32
6 Allgemeine Einführung in das Arbeiten mit DOS	35
6.1 Der neue PC oder PS/2 kommt	35
6.2 Start mit DOS	36
6.3 Installation von DOS	37
6.4 Die wichtigsten Befehle im Überblick	47
6.5 Drucken von Texten und vom Bildschirm (Hardcopy-Funktion)	58

6.6	Erstellen von Sicherungskopien (von Diskette und Festplatte)	59
7	Diskette und Festplatte	63
7.1	Aktuelles Laufwerk	63
7.2	Einteilung von Diskette und Festplatte	64
7.3	Disketten	65
7.4	Festplatten	68
8	Das DOS-Inhaltsverzeichnis	71
8.1	Befehle zur Bearbeitung von Directories (PATH und APPEND)	74
8.2	Anlegen eines Sub-Directories (MKDIR oder MD)	76
8.3	Anzeigen und Ändern des aktuellen Directories (CHDIR oder CD)	77
8.4	Löschen eines Sub-Directories (RMDIR oder RD)	79
8.5	Anzeigen des Inhaltsverzeichnisses eines Directories (DIR)	79
8.6	Löschen von Dateien in einem Directory (DEL oder ERASE)	81
8.7	Kopieren von Dateien zwischen Directories (COPY und XCOPY)	82
8.8	Zusammenfassendes Beispiel	84
9	Datenumleitung und Filter	87
9.1	Datenumleitung	87
9.2	Befehlsverkettung und Filter	89
10	Gerätenamen	93
11	Die DOS-Befehle mit Übungen	95
	APPEND	97
	ASSIGN	100
	ATTRIB	102
	BACKUP	105
	BREAK	108
	CALL	109
	CHCP	111
	CHDIR oder CD	112
	CHKDSK	114
	CLS	118

COMMAND	119
COMP	122
COPY	125
CTTY	131
DATE	132
DEBUG	134
DEL oder ERASE	147
DIR	148
DISKCOMP	152
DISKCOPY	155
ECHO	157
EDLIN	159
EXE2BIN	167
EXIT	169
FASTOPEN	170
FDISK	171
FIND	174
FOR	177
FORMAT	178
GOTO	183
GRAFTABL	185
GRAPHICS	186
IF	188
JOIN	190
KEYB (nationale Zeichensätze)	192
LABEL	195
MKDIR oder MD	196
MODE	197
MORE	203
NLSFUNC	204
PATH	205
PAUSE	207
PRINT	208
PROMPT	212
RECOVER	214
REM	215
RENAME oder REN	216
REPLACE	217
RESTORE	219
RMDIR oder RD	222
SELECT	223

	SET	225
	SHARE	227
	SHIFT	228
	SORT	230
	SUBST	232
	SYS	234
	TIME	235
	TREE	236
	TYPE	237
	VER	238
	VERIFY	239
	VOL	240
	XCOPY	241
12	Die BATCH-Befehle und ihre Anwendung	245
13	Bearbeitung der Zeichensätze für Drucker und Bildschirm	255
14	Die Konfiguration von DOS (CONFIG.SYS)	257
	BREAK	258
	BUFFERS	258
	COUNTRY	259
	FCBS	260
	FILES	261
	LASTDRIVE	262
	SHELL	263
	STACKS	264
	DEVICE	264
	– ANSI.SYS	265
	– DISPLAY.SYS	270
	– VDISK.SYS	271
	– DRIVER.SYS	275
	– PRINTER.SYS	278

	Anhang	
I	Netzwerk	281
II	Begriffserklärungen	283
III	Umrechnungstabellen	287
IV	Zeichendarstellungen/Code-Tabelle	289
V	Unterschiede DOS 2.11/3.0/3.1/3.2/3.30	295
VI	Bekannte Fehler in der DOS-Version 3.30	301
VII	Übersicht DOS-Interrupts	303
VIII	Übersicht DOS-Funktionen	305
IX	Übersicht DOS-Fehlercodes	327
	Stichwortverzeichnis	331
	Hinweise auf weitere Markt & Technik-Produkte	335

Vorwort

Das Erscheinen des IBM-PC, Ende 1980 in den USA, brachte den großen Durchbruch auf dem Mikrocomputermarkt (Mikrocomputer gibt es seit Mitte der siebziger Jahre – Verbreitung fanden sie ab Ende der siebziger 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 entwickelte sich ein Computerstandard, für den heute das wohl größte Angebot an Zusatzhardware und Software zur Verfügung steht.

Durch immer größere Verbreitung der PCs und der damit verbundenen Fertigung großer Stückzahlen sowie Innovationen sind die Kosten für die Anschaffung in den letzten Jahren ganz erheblich gesunken. Noch vor wenigen Jahren kostete ein mit dem PC vergleichbarer Computer mit zwei Diskettenlaufwerken etwa 20000 DM! Heute werden die vergleichbaren PCs mit wesentlichen Verbesserungen bereits für unter 2000 DM angeboten.

Der bereits rund sieben Jahre alte PC-Standard, welcher sicherlich nicht mehr den technischen Möglichkeiten entspricht, wurde von IBM durchbrochen. Im April 1987 stellte die IBM eine neue Reihe von Systemen unter dem Namen **Personal System /2** vor (es wurden über 100 neue Produkte vorgestellt!).

Vor allem bei der Darstellungsqualität auf dem Bildschirm sind erhebliche Verbesserungen zu beobachten. Durch konsequente Fortentwicklung wurden damit sicherlich die Weichen für die nächsten Jahre gestellt.

MS-DOS (PC-DOS) ist heute das Standard-Betriebssystem für Mikrocomputer der 16-Bit-Personalcomputer-Klasse. DOS ist nicht nur einfach an verschiedene Hardware anpaßbar, sondern auch sehr leicht erlern- und bedienbar. Bei der hier vorgestellten Version 3.30 handelt es sich um den neuesten Stand. Dieses

Betriebssystem wird auch in Zukunft eine sehr große Rolle spielen und damit auch weiterhin als Standard-Betriebssystem anzusehen sein.

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

Sie finden in den ersten Kapiteln (1 bis 9) eine Einführung mit den wichtigsten Grundlagen zur Bedienung Ihres PC bzw. PS/2 mit DOS.

Die Kapitel 10 bis 12 enthalten alle Befehle (alphabetisch geordnet in Kapitel 11) mit einer genauen Beschreibung.

In den einzelnen Kapiteln werden viele Beispiele vorgestellt, die Sie an Ihrem Computer durcharbeiten sollten. Dadurch wird der Lerneffekt gesteigert. Für den Fortgeschrittenen bietet das Buch ab Kapitel 10 einen hilfreichen Nachschlageteil (mit DOS-Konfiguration im Kapitel 14).

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) sowie die Neuerungen und Unterschiede zwischen den DOS-Versionen 2.11 und 3.0 bis 3.30.

Ziel des Buches ist es, Ihnen zu zeigen, wie Sie Ihre Arbeiten mit DOS 3.30 an Ihrem Computersystem erledigen 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

Bestandteile und Anwendungsmöglich- keiten des PC

Dieses Kapitel will Ihnen kurz die einzelnen Bestandteile Ihres PC bzw. PS/2 vorstellen. Zusätzlich erhalten Sie Informationen, die Anwendungsmöglichkeiten zum Einsatz aufzeigen.

1.1 Die PC- und PS/2-Hardware

Hardware

Standardmäßig erhalten Sie beim Kauf Ihres Computers eine Zentraleinheit mit einem Diskettenlaufwerk oder zwei Diskettenlaufwerken oder einem Diskettenlaufwerk und einer Festplatte. Selbstverständlich können Sie auch Computer mit mehr Laufwerken erhalten.

Eine Festplatte ist eine fest eingebaute »Diskette« mit besonders großem Speicher und hoher Geschwindigkeit.

Damit Sie die Ergebnisse Ihrer Arbeit auch auf Papier bringen können, benötigen Sie noch einen Drucker.

Sie können natürlich auch einen Plotter anschließen, mit dem Sie Grafiken usw. zeichnen können.

Datenfernübertragung

Für die Kommunikation mit anderen Computern per Datenfernübertragung (z.B. Telefonleitung) gibt es Akustikkoppler oder Modems (von der Post). Natürlich können Sie auch einen Teletex oder BTX-Anschluß für Ihren Computer bekommen.

Software

Für die anschließbaren Geräte benötigen Sie in jedem Fall eine Software (Programme), die diese sogenannte Peripherie auch benutzen kann. Lassen Sie sich also vor dem Kauf beraten, welche Geräte und Programme für Sie geeignet sind.

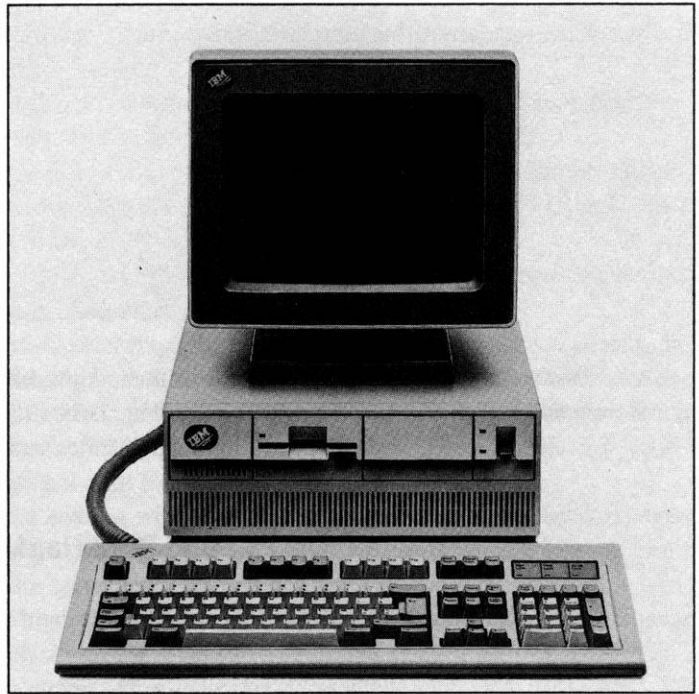


Bild 1.1
*Die Hardware des
Personal System /2*

Anschlüsse am PC

Für den Anschluß der Tastatur, des Druckers, usw. beachten Sie bitte die Beschreibung im Handbuch zu Ihrem Computer. Daneben ist mindestens noch eine parallele und eine serielle Schnittstelle vorhanden. Der parallele Anschluß wird am besten für den Drucker verwendet. Die serielle Schnittstelle (V.24/RS232C) kann für den Akustikkoppler, ein Modem oder den Plotter verwendet werden. Generell ist es mit DOS ohne Änderung möglich, bis zu drei parallele und zwei serielle Schnittstellen zu betreiben.

1.2 Die Software

Zusammen mit der Hardware (sichtbarer Teil des Computers) erhalten Sie eine Start- und Programmdiskette mit DOS (siehe auch Kapitel 6). 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 stellt das in diesem Buch beschriebene DOS dar.

Über 95% der am Weltmarkt zur Zeit verkauften PCs und PS/2 arbeiten mit DOS. Daher gibt es für dieses Betriebssystem auch die meiste Anwendungssoftware (Textverarbeitung, Grafikprogramme, Finanzbuchhaltung usw.).

Darüber hinaus sind für PCs noch viele weitere Betriebssysteme lieferbar (z.B. CP/M-86, Xenix, Unix, MBOS, Concurrent-DOS usw.).

1.3 Anwendungsmöglichkeiten

Mit der in der Standardausstattung enthaltenen Software können Sie schon eine ganze Menge machen. Sicher werden Sie aber 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 PC

Hier nur einige Anwendungsmöglichkeiten:

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

Es gibt für fast jeden Bedarf die richtige Software. Bevor Sie 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 Anforderungen entspricht.

Viele Programme kosten mehr als Ihr Computer (sehr komplexe Lösungen können mehr als 40000 DM kosten). Neben den Kosten für ein komplexes Programm ist es wichtig, daß Ihr Lieferant auch Beratung, Schulung und Service anbietet. Die

damit verbundenen Mehrkosten machen sich sehr häufig schnell bezahlt, wenn Sie diese Leistungen benötigen.

Wollen Sie größere Mengen von Daten mit Ihrem Computer verwalten, so empfiehlt sich unbedingt die Anschaffung einer Festplatte und eines Bandlaufwerkes für die Datensicherung. Eine Festplatte hat neben der hohen Kapazität den Vorteil der hohen Verarbeitungsgeschwindigkeit.

Kapitel 2

Was ist DOS?

DOS als Schnittstelle zwischen Hard- und Software

Eigentlich müßte es heißen: Was ist ein Betriebssystem? DOS ist eine Programmsammlung, die zum Betrieb eines Computers notwendig ist.

Ein Computer (Hardware) ist allein für sich 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 Sammlung von Programmen, die den Computer und seine Peripherie (Drucker, Plotter, Diskettenlaufwerke, Platte, Bildschirm usw.) verwalten. 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 DOS viele Aufgaben, wie zum Beispiel das Kopieren von Daten und Programmen, durchführen.

Laden von DOS

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

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

Aufgaben von DOS

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

DOS ist die Abkürzung von **Disk Operating System**. Dieses Betriebssystem konnte sich seit Anfang der 80er Jahre in kurzer Zeit durchsetzen, vor allem durch die Entscheidung von IBM, ihre PCs damit auszustatten (die IBM-Bezeichnung war bisher PC-DOS).

*DOS als Standard-
Betriebssystem*

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 DOS.

DOS, OS/2 und Unix

Welchem Betriebssystem die Zukunft auf den neuen 32-Bit-Computern (z.B. IBM Personal System /2, Modell 80) gehört, ist zur Zeit fraglich. Wird das »Rennen« DOS, OS/2 oder Unix gewinnen? Sicherlich wird entscheidend sein, ob für Unix künftig mehr Anwendungsprogramme verfügbar sind. Der wesentliche Vorteil von Unix ist die Mehrplatzfähigkeit, die weder von DOS noch von OS/2 künftig geboten wird. Aber auch DOS entwickelt sich weiter und besitzt in Verbindung mit Netzwerksoftware und -hardware die Mehrplatzfähigkeit.

DOS besteht aus vier Teilen (und diversen Dienstprogrammen):

DOS-Teil	Kurzbeschreibung
BOOT-LOADER	Wird direkt nach dem Start des Computers als erstes Programm ausgeführt. Er startet das DOS, indem er IBMBIO.COM in den Speicher des Computers lädt und das System bootet.
IBMBIO.COM	Enthält alle systemspezifischen Programme zur Abwicklung der Kommunikation mit den angeschlossenen Geräteeinheiten (Drucker usw.).
IBMDOS.COM	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.

Kapitel 3 Kompatibilität von PCs bzw. PS/2 und DOS

Kompatibilität von PCs

Was ist Kompatibilität? Darunter versteht man, daß Sie Software oder Hardware, die z.B. auf dem IBM-PC funktioniert, auch in einem anderen Modell benutzen können. Damit können Sie aus einer riesigen Palette von mehreren tausend Programmen oder Hardwareerweiterungen (z.B. Maus) die für Sie am besten geeignete auswählen.

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

Sehr häufig spricht man von »IBM-kompatibel«. Das bedeutet aber nicht immer, daß alles, was auf dem einen PC läuft, auf dem anderen ebenso problemlos funktioniert.

Hardware-Änderungen bei PS/2-Systemen

Einschubkarten, z.B. für BTX/Bildschirmtext, sind zwar für die PC-Reihe nutzbar, können jedoch für die neuen PS/2-Systeme nicht benutzt werden. Die Hardware der Serie PS/2 wurde in einigen wesentlichen Details geändert, was zu einer gewissen Inkompatibilität führt.

Grundsätzlich muß bei der Kompatibilität zwischen Hardware und Software unterschieden werden. Viele PCs sind zwar software-, jedoch nicht hardwarekompatibel zu den bisherigen PCs. Die neue Serie Personal System /2 von IBM stellt dabei ein sehr gutes Beispiel dar. Die Systeme sind zwar weitestgehend software-, jedoch nicht hardwarekompatibel. Fast alle Programme, die problemlos auf PC, XT und AT funktionieren, können auch auf den Modellen der Reihe PS/2 verwendet werden. Dies gilt jedoch nicht für die Hardware. Ausgenommen davon sind jedoch vor allem Drucker.

Schwierigkeiten bei Software-Problemen

Wenn Ihnen oder Ihrem Lieferanten nicht bekannt ist, ob ein Programm oder Hardware-Teil auf Ihrem PC oder PS/2-System lauf-

fähig ist, sollten Sie dies vor dem Kauf unbedingt testen. Damit ersparen Sie sich eventuell später auftretende Schwierigkeiten z.B. mit nicht voll funktionsfähigen Programmen. Bedenken Sie immer, daß es für die Kompatibilität keine Garantie gibt. Soweit vom Hersteller Ihres PC Programme angeboten oder empfohlen werden, können Sie 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 wegen der größeren Verkaufszahlen günstiger kalkulieren können.

Was soll Software heißen?

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 Computers kosten können. Häufig hängt es vor allem von den Programmen ab, ob sich Ihr Computer-Kauf gelohnt hat.

Programme, für die eine ältere Version als Ihr DOS 3.30 verlangt wird, können Sie normalerweise trotzdem ohne Probleme einsetzen.

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

*»Abwärts«-Kompatibilität
von DOS*

Eine neue Version zu DOS ist normalerweise immer kompatibel zu den vorherigen Versionen (seit der DOS-Version 1). Dies gilt auch für künftige Betriebssystem-Versionen zu DOS 3.

Manchmal kommt es vor, daß eine Software auf Ihrem System nicht läuft. Dies kann verschiedenste Ursachen haben (beachten Sie auch das Kapitel über die Konfiguration von DOS). Eventuell reicht der Speicher Ihres Computers nicht aus, um ein Programm zu benutzen (nicht immer erhalten Sie dabei eine korrekte Fehlermeldung). In diesem Fall sollten Sie sich mit Ihrem Lieferanten oder dem Softwarehersteller in Verbindung setzen und diesem genau beschreiben, was nach dem Aufruf des Programmes passiert und welchen Computer Sie benutzen (einschließlich der DOS-Version).

Kapitel 4

Die Tastatur

4.1 Allgemeines

Schreibmaschinen- tastatur

Die Tastatur Ihres PC (Bild 4.1) umfaßt mehrere Gruppen: Kompletter Buchstabensatz entsprechend einer Schreibmaschine (mit der Betätigung der Taste *Caps Lock* bzw. *Gross* können Sie Ihre Tastatur auf Großschreibung umschalten – dies gilt nur für Buchstaben).



Bild 4.1

Die Tastatur des Systems PS/2

Funktionstasten

Zifferntastenfeld

Die Funktionstasten finden Sie in Bild 4.2 dargestellt. Die Bedeutung der Tasten ist auf den nächsten Seiten näher beschrieben. Auf der rechten Seite Ihrer Tastatur finden Sie ein Zifferntastenfeld. Um mit diesem arbeiten zu können, muß vorher die Taste *NumLock* betätigt werden. Damit wird jedoch gleichzeitig die Möglichkeit der Nutzung als Funktionstasten abgeschaltet. Betätigen Sie *NumLock* ein weiteres Mal, so ist der ursprüngliche Zustand wiederhergestellt. Wenn Sie die neue MF-Tastatur verwenden (z.B. bei den Systemen PS/2), können Sie den rechten

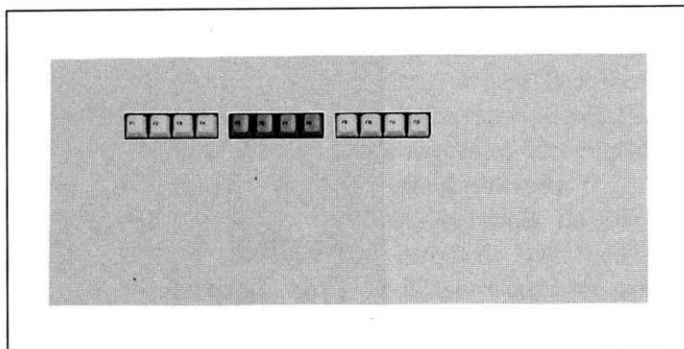


Bild 4.2
Funktionstasten

Teil der Tastatur meist generell umschalten, da die wichtigsten Funktionstasten (bzw. Cursorsteuertasten) links daneben noch einmal separat vorhanden sind.

*Eingabe von DOS-
Befehlen*

Mit Hilfe der Funktionstasten können Sie die Befehlszeilen leicht ändern. Die zuletzt eingegebene Zeile kann aus dem Tastatureingabespeicher (RAM) komplett oder in Teilen zurück in die

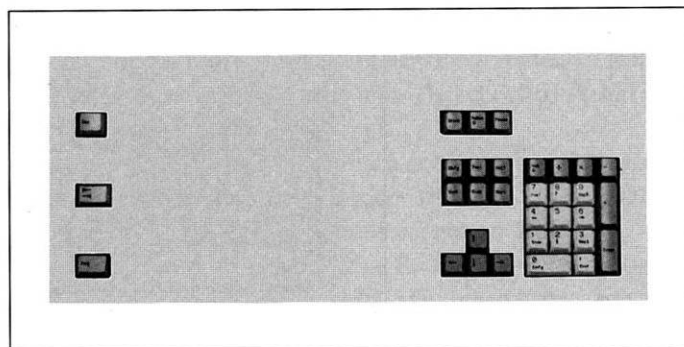


Bild 4.3
Besondere Funktionstasten

Befehlszeile (Bildschirmanzeige) kopiert werden. Der Tastatureingabespeicher ist am Bildschirm nicht sichtbar. Dort befindet sich immer die zuletzt eingegebene Zeile bzw. der zuletzt eingetippte 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)

Funktionstastenbelegung

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

Mit dem Befehl PROMPT (siehe Kapitel 11) können Sie die Funktion bzw. Belegung einzelner Tasten verändern.

Taste	Funktion	Beschreibung
F1	kopiert ein Zeichen	kopiert ein Zeichen aus dem Eingabespeicher in die neue Zeile am Bildschirm
F2x	kopiert bis x	kopiert alle Zeichen bis zum angegebenen Zeichen (x) vom Eingabespeicher (RAM) in die neue Zeile am Bildschirm (ohne das Zeichen ab der aktuellen Position in der Zeile) – dieser Vorgang kann in einer Eingabezeile beliebig oft wiederholt werden
F3	kopiert alle Zeichen	kopiert alle Zeichen ab der aktuellen Position aus dem Eingabespeicher in die neue Zeile am Bildschirm
F4x	überspringt Zeichen bis x	überspringt alle Zeichen des Eingabespeichers bis das Zeichen (x) erkannt wird (ohne daß am Bildschirm etwas angezeigt wird) – dieser Vorgang kann in einer Eingabezeile beliebig oft wiederholt werden
F5	neuer Eingabespeicher	schreibt die neue Zeile in den Eingabespeicher – dies geschieht auch automatisch nach Betätigung der Taste ↵
F6	beendet die Eingabe	erzeugt ein <i>Strg</i> +Z, welches das Zeichen für das Dateiende darstellt (beachten Sie den COPY-Befehl)
Strg+Z	beendet die Eingabe	erzeugt ein <i>Strg</i> +Z (1Ah), welches das Zeichen für ein Dateiende darstellt (wie Taste F6)
F7	Nullzeichen	gibt ein <i>Strg</i> +@ aus, das den Wert 00 (HEX) darstellt
Alt+Untbr	Eingabe löschen	löscht den Tastaturspeicher (der eingegebene Befehl wird abgebrochen)
←	Eingabe löschen	löscht ein Zeichen aus der Eingabezeile am Bildschirm, und im Eingabespeicher wird die aktuelle Position um 1 Zeichen zurückgesetzt
Einfüg	Einfügen	schaltet die Eingabe auf Einfügemodus um (die aktuelle Position im Eingabespeicher wird dabei nicht verändert) – Abschalten durch erneute Betätigung
Entf	Löschen	löscht ein folgendes Zeichen im Eingabespeicher, jedoch nicht in der Eingabezeile bzw. am Bildschirm
Esc	Eingabe löschen	löscht die gesamte Eingabezeile, nicht jedoch den Eingabespeicher

Taste	Funktion	Beschreibung
Strg+Alt+Entf	Systemstart	den Computer neu starten (booten, also das DOS komplett neu laden)
Strg+Untbr	Befehlsabbruch	unterbricht die Ausführung eines DOS-Befehls (siehe Befehl BREAK)
Strg+C	Befehlsabbruch	unterbricht die Ausführung eines DOS-Befehls (siehe Befehl BREAK)
Strg+H	Zeichen löschen	löscht das zuletzt eingegebene Zeichen am Bildschirm (wie Taste ←)
→	Tabulator	Cursor am Bildschirm zur nächsten Tabulatorposition bewegen (wie <i>Strg+I</i>)
Strg+I	Tabulator	Cursor am Bildschirm zur nächsten Tabulatorposition bewegen (wie Taste →)
Strg+↵	Zeilenende	fügt ein Zeilenende bzw. einen Zeilenvorschub ein
Strg+J	Zeilenende	fügt ein Zeilenende bzw. einen Zeilenvorschub ein
Strg+Num ↵	Bildschirmstopp	unterbricht die Bildschirmanzeige, bis eine beliebige andere Taste betätigt wird
Strg+S	Bildschirmstopp	unterbricht die Bildschirmanzeige, bis eine beliebige andere Taste betätigt wird
Pause	Bildschirmstopp	unterbricht die Bildschirmanzeige, bis Strg+Pause betätigt wird
Strg+P	Druckausgabe	Bildschirm- und -ausgaben gleichzeitig drucken (erneutes Betätigen schaltet die Funktion ab)
Strg+Druck	Druckausgabe	Bildschirm- und -ausgaben gleichzeitig drucken (erneutes Betätigen schaltet die Funktion ab)
↵+Druck	Bildschirm drucken	erstellt eine Kopie der kompletten Bildschirmanzeige am Drucker (Hardcopy-Funktion) – beachten Sie hierzu den Befehl GRAPHICS

4.3 Beispiele für die Anwendung der Funktionstasten

Inhaltsverzeichnis einer Diskette

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

dir verzeich.doc ↵-Taste

Befehlswiederholung

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

F3 ↵-Taste

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

F2- und d

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

dir verzeich.

*Text am Bildschirm
anzeigen*

Geben Sie den Stern (*) ein und betätigen Sie die Taste ↵.

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

type verzeich.doc

Sie überschreiben die ersten drei Zeichen der alten Zeile (**dir**), indem Sie die Buchstaben **typ** tippen. Sie betätigen die Taste **Einf**, fügen ein **e** hinzu, betätigen die Tasten **F2*** und schreiben als letztes **doc**. Nach Betätigung von ↵ 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
Einf	schaltet um auf einfügen	
e	schreibt ein Zeichen	type
F2*	kopiert Zeichen bis zum *	type verzeich.
d	schreibt ein Zeichen	type verzeich.d
o	schreibt ein Zeichen	type verzeich.do
c	schreibt ein Zeichen	type verzeich.doc
↵	gibt die Zeile an das DOS-Programm COMMAND.COM zum Ausführen	Inhalt wird angezeigt Gleichzeitig wird die neue Zeile in den Tastatureingabespeicher übernommen

Tastenkombinationen Bei Verwendung einer deutschen Tastatur benötigen Sie zur Arbeit mit DOS Zeichen, die Sie nur durch Betätigung mehrerer Tasten gleichzeitig erzeugen können:

Tasten-Kombination		Zeichen	Bemerkung
MF-Tastatur	andere		
AltGr+8	Strg+Alt+ü	[	Eckige Klammer auf
AltGr+9	Strg+Alt++	]	Eckige Klammer zu
AltGr+B	Strg+Alt+<	\	Back-Slash
AltGr+Q	Strg+Alt+2	@	Klammeraffe
	Strg+Alt+F1		Umschalten auf (US-)ASCII-Tastatur
AltGr+<	>		Verkettungszeichen
	Strg+Alt+F2		Umschalten auf deutsche Tastatur
	oder		
AltGr+<	Alt+124		Verkettungszeichen (Die Zahl 124 darf nur mit dem rechten numerischen Ziffernblock erzeugt werden, ohne daß vorher die Num -Taste betätigt wurde, also als Funktionstasten.)

Auf dem Bildschirm wird das | -Zeichen als in der Mitte unterbrochener Strich dargestellt.

Auf der neuen MF-Tastatur (z. B. bei allen Systemen PS/2) können die auf den Tastenkappen unten rechts angegebenen Zeichen erzeugt werden, indem diese zusammen mit der Taste *AltGr* betätigt werden. Diese Tastatur entspricht der deutschen DIN-Norm.

4.4 Tastenbezeichnungen bei verschiedenen Tastaturen

Unterschiede bei Tastenbezeichnungen

Die Bezeichnungen der Tasten sind bei PCs und PS/2 teilweise unterschiedlich (in diesem Buch werden die Bezeichnungen der dritten Spalte verwandt):

PC, PC/XT, PPC	PC/AT	MF-Tastatur
Ctrl	Strg	Strg
Alt	Alt	Alt/AltGr
Del	Lösch	Entf
Esc	Eing Lösch	Esc
Break	Abbr	Untbr
NumLock	Num ↕	Num ↕
CapsLock	Groß ↕	↕
Ins	Einfg	Einfg
Home	Posl	Posl
End	End	Ende
PgUp	Bild↑	Bild↑
PgDn	Bild↓	Bild↓
PrtSc	Druck	Druck
Scroll Lock	Abbr	Rollen↓

Kapitel 5

Übersicht über die DOS-Befehle

5.1 Allgemeines

Befehle sind Anweisungen an das Betriebssystem, bestimmte Aufgaben durchzuführen (Kopieren, Drucken usw.). Sie werden von Ihnen an der 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 Computers 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 dieser in den Speicher Ihres Computers zur Ausführung geladen.

Ausführung der Befehle

Bei Eingabe eines Befehls wird dieser von COMMAND.COM (wird beim Start in den Speicher geladen) gestartet. Eine nähere Beschreibung der Dateinamen finden Sie im Kapitel 6.4. Die höchste Priorität besitzen die internen Befehle. COM- und EXE-Dateien verfügen über zweite bzw. dritte und BAT-Dateien über vierte Priorität. Externe Befehle sind an den letzten drei Buchstaben des Dateinamens erkennbar. Dies sind COM, EXE oder BAT.

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 einer Form vorliegt, damit die Ausführung des richtigen Befehls gewährleistet ist.

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

Bad command or filename

bzw.

Falscher Befehl oder Dateiname

COMMAND.COM

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

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

Der erste Teil wird von 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 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 Betriebssystem-Ebene arbeiten (es gibt zwischenzeitlich Programmiersprachen, wie z.B. GW-Basic bzw. BasicA, die einen Aufruf aus einem

Programm ermöglichen). Diese wird durch das Befehlsanforderungs-Zeichen, den Prompt (standardmäßig aktueller **Laufwerk-buchstabe** und >), angezeigt.

Befehl	Funktion
BREAK	aktiviert/deaktiviert Programmunterbrechung
CALL	Aufruf einer Batch-Datei aus einer anderen heraus
CHDIR oder CD	Directory wechseln
CLS	Bildschirm löschen
COPY	Dateien kopieren
CTTY	Peripheriegerät ändern, von dem Befehle eingegeben werden
DATE	Datum anzeigen/ändern
DEL oder ERASE	Datei(en) löschen
DIR	Dateien/Verzeichnis anzeigen
ECHO	unterdrückt die Anzeige von Befehlszeilen in einer Batch-Datei; zeigt einen Kommentar an
EXIT	Verlassen der Betriebssystem-Ebene und Rückkehr in ein Anwenderprogramm (in Verbindung mit COMMAND)
FOR	mehrmaliges Wiederholen eines 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 Directory führen, in dem ein externer Befehl gesucht werden soll
PAUSE	unterbricht die Abarbeitung einer Batch-Datei und gibt einen Text auf dem Bildschirm aus (Fortsetzung mit Betätigung einer Taste)
PROMPT	ändert das Befehlsanforderungs-Zeichen (Prompt)
REM	Kommentare in einer Batch-Datei
RENAME oder REN	Dateinamen ändern
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 Version von DOS an
VERIFY	überprüft kopierte Dateien (Schalter)
VOL	zeigt den Namen einer Diskette oder Festplatte an (Label-Name)
d:	aktuelles Laufwerk (Arbeitslaufwerk) umschalten

5.3 Externe Befehle

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

Befehl	Funktion
APPEND	benennt Wege, die zu einem Directory führen, in dem eine (Daten-)Datei gesucht werden soll
ASSIGN	leitet Laufwerksanfragen um
ATTRIB	Datei auf »Nur lesen« oder Dateiänderungs-Attribut setzen
BACKUP	Sichern von Dateien einer Diskette/Festplatte auf eine andere
BASIC oder BASICA	Programmiersprache Basic
CHCP	Zeichensatztablette auswählen, ändern und anzeigen
CHKDSK	überprüft die Platte und/oder eine Datei auf Fehler
COMMAND	führt z.B. interne wie externe Befehle aus sowie Start eines sekundären Befehlsprozessors
COMP	vergleicht den Inhalt zweier Dateien auf Gleichheit
DEBUG	für Programmtest und -modifikation
DISKCOMP	vergleicht zwei Disketten auf Gleichheit
DISKCOPY	kopiert Disketten 1:1
EDLIN	einfaches Textprogramm (Editor)
EXE2BIN	formt Dateien von .EXE auf .COM um
FASTOPEN	schnellerer Dateizugriff durch Speichern der Wege (Pfade) und der zuletzt eröffneten Dateien
FDISK	verwaltet die Bereiche einer Festplatte
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
KEYB	Zeichensätze für verschiedene Länder
LABEL	Zuweisen/Anzeigen von Namen für Diskette/Festplatte
LINK	Programm zum Binden von Programm-Teilmodulen
MODE	modifiziert DOS für Peripheriegeräte
MORE	zeigt eine Bildschirmausgabe seitenweise an
NLSFUNC	Auswählen eines Zeichensatzes bei Verwendung mit DISPLAY.SYS und PRINTER.SYS (CONFIG.SYS), sowie Unterstützung für zusätzliche Landesinformationen und das Wechseln der Zeichensatztablette
PRINT	Ausgabe von Texten auf den Drucker
RECOVER	erneuert fehlerhafte Dateien
REPLACE	Ersetzen alter Versionen und Einfügen neuer Dateien
RESTORE	Zurückkopieren gesicherter Dateien auf Festplatte
SELECT	installiert DOS mit nationalem Zeichensatz, Datum- und Zeitformat

Befehl	Funktion
SHARE	File-Sharing-Support laden (für Netzwerk/MS-NET)
SORT	sortiert Datei-Inhalte
SUBST	temporär Sub-Directory durch Laufwerk ersetzen
SYS	kopiert Systemdateien IBMBIO.COM und IBMDOS.COM
TREE	zeigt das strukturierte Verzeichnis einer Festplatte
XCOPY	Kopieren von Dateien und Verzeichnissen

Kapitel 6

Allgemeine Einführung in das Arbeiten mit DOS

Umgang mit DOS

Dieses Kapitel ist für den Anfänger, der noch keinerlei Kenntnisse im Umgang mit DOS hat. Grundlage für den Inhalt ist die Erfahrung, daß ca. 80 % der Benutzer nur 20 % der Befehle des DOS in der täglichen Arbeit anwenden.

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

6.1 Der neue PC oder PS/2 kommt

Sie erhalten zusammen mit Ihrem Computer einen Bildschirm, einen Drucker und ein Handbuch zu DOS.

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

Ausstattung des PC

Sie haben entweder ein Disketten- oder ein Festplatten-System erworben. Eine Festplatte ist im Grunde genommen nichts anderes, als eine »fest eingebaute schnelle 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 DOS bzw. Ihr Anwendungsprogramm). Dabei wird in der EDV eine Kartei (z.B. Kunden, Lieferanten) als Datei bezeichnet.

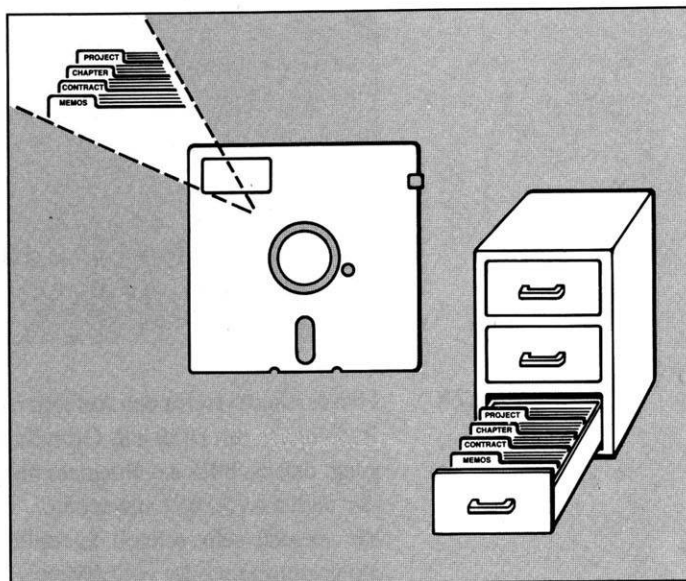


Bild 6.1
Die Diskette als Aktenschrank

Beachten Sie hierzu auch das Kapitel 7 »Diskette und Festplatte«.

6.2 Start mit DOS

Führen Sie Ihre DOS-Diskette in das linke (bzw. erste) Laufwerk ein, und schließen Sie dieses. Beachten Sie hierzu auch das Bedienerhandbuch zu Ihrem Computer.

Starten von DOS

Schalten Sie Ihren Computer ein.

Es läuft am Anfang immer ein Testprogramm in Ihrem System ab, das überprüft, ob alles in Ordnung ist.

Bei Festplatten-Systemen wird von der Diskette gestartet, wenn eine solche eingelegt und das Laufwerk geschlossen ist, ansonsten erfolgt der Start von der Festplatte (diese muß jedoch vor der ersten Benutzung eingerichtet sein).

Am Bildschirm erhalten Sie folgende Anzeige:

Systemdatum: Mi. 2.09.1987

Neues Datum (tt.mm.jj) eingeben:

Tagesdatum

Geben Sie an dieser Stelle das Tagesdatum ein (bitte beachten Sie hierzu die Befehle DATE in Kapitel 11 und COUNTRY in Kapitel 14). Wenn Sie das Datum nicht ändern wollen, betätigen Sie nur die Taste *Return* bzw. *Enter*. Das Datum in der Anzeige

(es kann bei Ihnen ein anderes sein) ist lediglich ein Beispiel (bei dem angegebenen handelt es sich um den 2. September 1987). Als nächstes erscheint am Bildschirm:

Systemzeit: 23.00.34,78

Neue Zeit (hh.mm.ss) eingeben:

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: 23 Uhr, 0 Minuten, 34 Sekunden, 78 Hundertstel). Wenn Sie sich bei dem Datum oder der Uhrzeit vertippen (vorausgesetzt, Sie haben noch nicht *Return* gedrückt), so können Sie mit der Taste *Backspace* (←) wieder rückwärts löschen und die Eingabe korrekt vornehmen. Zum Schluß erscheint am Bildschirm:

IBM Personal Computer DOS

Version 3.30 (C) Copyright International Business Machines Corp 1981, 1987

(C) Copyright Microsoft Corp 1981, 1987

A>

System- Bereitschaftsmeldung

Die Angabe **A>** ist der 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**.

6.3 Installation von DOS

Herstellen von Betriebssystem-Kopien

Sie sollten niemals mit Originaldisketten arbeiten. Für die Erstellung von Kopien ist eine unterschiedliche Handhabung bei Computern mit einem oder zwei Diskettenlaufwerken und solchen mit einer Festplatte erforderlich.

6.3.1 Kopieren von Disketten auf Systemen mit zwei Diskettenlaufwerken

DISKCOPY

Legen Sie Ihre 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 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.

Quellendiskette in Laufwerk A: einlegen

Zioldiskette in Laufwerk B: einlegen

Anschließend eine Taste betätigen ...

Die neue Diskette wird formatiert und eine 1:1-Kopie vom Original erstellt (dabei werden eventuell vorher auf der Zieldiskette vorhandene Daten gelöscht).

Kopiert werden 80 Spuren

18 Sektoren/Spur, 2 Seite(n)

Diskette wird beim Kopieren formatiert

Ende des Kopiervorgangs

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

Eine weitere Kopie erstellen (J/N)?

Drücken Sie bitte *j* (JA) und *Return*, wenn Sie dies wünschen (dazu eine weitere Originaldiskette in Laufwerk A: und/oder eine neue in B: einlegen). Im anderen Falle *n* und *Return*.

Sollte sich nach Beendigung des DISKCOPY die DOS-Diskette nicht im Laufwerk A befinden, erhalten Sie die Meldung

Diskette mit \COMMAND.COM in Laufwerk A:

einlegen und eine Taste betätigen

Bitte verwenden Sie nur Kopien von 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 sofort weitergearbeitet werden kann. Sie können dann später problemlos weitere Kopien vom Original erstellen.

*SELECT für nationale
Tastaturanpassung*

Für die Installation von DOS auf einer neuen Diskette sollte nicht wie bereits beschrieben der Befehl DISKCOPY verwendet werden, sondern SELECT. Mit diesem Dienstprogramm werden

zusätzlich die nationale Tastaturanpassung sowie Datum und Uhrzeitformat installiert.

Für Deutschland würde der Befehl

```
select a: b: 049 gr
```

eingegeben. Dazu muß sich in Laufwerk A die DOS-Startdiskette und im zweiten Laufwerk (B) eine neue befinden. Wenn Sie nur eine Laufwerksbezeichnung angeben (hier b:), verwendet SELECT dieses als Ziel. **049** (internationale Vorwahl für Deutschland) ist der Landescode und **gr** der Tastaturcode (*Germany*).

SELECT ruft als erstes das Dienstprogramm FORMAT auf, um die neue Diskette zu formatieren. Eventuell auf dieser vorhandene Daten/Programme werden unwiederbringlich gelöscht:

```
SELECT wird beim ersten Installieren von  
DOS benutzt. Durch SELECT werden alle  
Daten auf der angegebenen Einheit ge-  
löscht und DOS wird installiert.  
Fortfahren (J/N)? J
```

Nach Bestätigung von *Return* erhalten sie folgende Anzeige vor dem Start der Formatierung:

```
Neue Diskette im Laufwerk B: einlegen,  
anschließend die Eingabetaste betätigen
```

Nach Beendigung der Formatierung erhalten Sie folgende Anzeigen:

```
Formatieren beendet  
Systemdateien übertragen  
Weitere Dskt./Platte formatieren (J/N)?
```

Dazu wird die Kapazität der neuen Diskette angezeigt. Nach Eingabe von **n** erhalten Sie die Anzeige

```
DOS-Diskette in Laufwerk A: einlegen  
anschließend eine Taste betätigen
```

Mit XCOPY wird anschließend automatisch der Inhalt der Diskette in Laufwerk A auf diejenige in Laufwerk B kopiert.

Auf der neuen Diskette werden als letztes zwei neue Dateien erstellt:

- CONFIG.SYS mit dem Inhalt:
COUNTRY=049,437,A:\COUNTRY.SYS
- AUTOEXEC.BAT mit dem Inhalt:
PATH \;
KEYB GR 437 A:\KEYBOARD.SYS
ECHO OFF
CLS
DATE
TIME
VER

Damit ist der Befehl SELECT beendet, und Sie besitzen eine neue DOS-Diskette, die eine für Deutschland passende Startroutine beinhaltet.

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).

6.3.2 Kopieren von Disketten auf Systemen mit einem Diskettenlaufwerk

Kopieren mit einem Laufwerk

Da Ihnen in der Regel bei einem Festplatten-PC nur ein Diskettenlaufwerk zur Verfügung steht, kann der Kopiervorgang nicht von Laufwerk zu Laufwerk erfolgen. Das gleiche gilt, wenn Sie einen PC mit nur einem Diskettenlaufwerk ohne Festplatte besitzen. Legen Sie Ihre DOS-Diskette in das Laufwerk ein. Zum Kopieren benötigen Sie den Befehl DISKCOPY (siehe auch Kapitel 11):

DISKCOPY 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 DOS-Diskette gelesen und in den Speicher (RAM) geladen, um diesen dort abzuarbeiten.

Quellendiskette in Laufwerk A: einlegen
Anschließend eine Taste betätigen ...

Legen Sie die zu kopierende Diskette ein. Nach Betätigung einer Taste erscheint nach kurzer Zeit

Kopiert werden 80 Spuren
18 Sektoren/Spur, 2 Seite(n)

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

Zieldiskette in Laufwerk A: einlegen
Anschließend eine Taste betätigen ...

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

Diskette wird beim Kopieren formatiert

Ende des Kopiervorgangs

Wenn der Kopiervorgang beendet ist, werden Sie noch gefragt, ob eine weitere Diskette kopiert werden soll. Drücken Sie bitte *j* und *Return*, wenn Sie eine weitere kopieren möchten. Im anderen Falle *n* und *Return*.

Eine weitere Kopie erstellen (J/N)?

Bitte verwenden Sie nur Kopien von 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 sofort weitergearbeitet werden kann. Sie können dann später problemlos weitere Kopien vom Original erstellen.

*SELECT für nationale
Tastaturanpassung*

Für die Installation von DOS auf einer neuen Diskette sollte nicht wie bereits beschrieben der Befehl DISKCOPY verwendet werden, sondern SELECT. Mit diesem Dienstprogramm werden zusätzlich die nationale Tastaturanpassung sowie Datum und Uhrzeitformat installiert.

Für Deutschland würde der Befehl

select a: b: 049 gr

eingegeben. Dazu muß sich in Laufwerk A die DOS-Diskette befinden. Wenn Sie nur eine Laufwerksbezeichnung angeben,

*Neue Diskette wird
formatiert*

verwendet SELECT dieses als Ziel. **049** ist der Landescode und **gr** der Tastaturcode.

SELECT ruft als erstes das Dienstprogramm FORMAT auf, um die neue Diskette zu formatieren. Eventuell auf dieser vorhandene Daten/Programme werden unwiederbringlich gelöscht.

SELECT wird beim ersten Installieren von DOS benutzt. Durch SELECT werden alle Daten auf der angegebenen Einheit gelöscht und DOS wird installiert.
Fortfahren (J/N)? J

Nach Betätigung von *Return* erhalten Sie folgende Anzeige vor dem Start der Formatierung:

Neue Diskette in Laufwerk B: einlegen,
anschließend die Eingabetaste betätigen

Nach Beendigung der Formatierung erhalten Sie folgende Anzeigen:

Formatieren beendet
Systemdateien übertragen
Weitere Dskt./Platte formatieren (J/N)?

Dazu wird die Kapazität der neuen Diskette angezeigt. Nach Eingabe von **n** erhalten Sie die Anzeige

DOS-Diskette in Laufwerk A: einlegen
anschließend eine Taste betätigen

Mit XCOPY kopieren

Sie werden im folgenden noch mehrmals aufgefordert, die Disketten zu wechseln. Dazwischen wird mit XCOPY automatisch der Inhalt der Diskette in Laufwerk A auf diejenige in »B« kopiert. Während des gesamten Vorganges werden Sie laufend aufgefordert, die DOS-Diskette bzw. die neue Diskette in das Laufwerk einzulegen:

DOS-Diskette in Laufwerk A: einlegen,
anschließend eine Taste betätigen.

bzw.

Diskette in Laufwerk B: einlegen,
anschließend eine Taste betätigen.

Auf der neuen Diskette werden als letztes zwei neue Dateien erstellt:

- CONFIG.SYS mit dem Inhalt:
COUNTRY=049,437,A:\COUNTRY.SYS
- AUTOEXEC.BAT mit dem Inhalt:
PATH A:\
KEYB GR 437 A:\KEYBOARD.SYS
ECHO OFF
CLS
DATE
TIME
VER

Damit ist der Befehl SELECT beendet, und Sie besitzen eine neue DOS-Diskette, die eine für Deutschland passende Startroutine beinhaltet.

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).

6.3.3 Einrichten einer neuen Festplatte mit DOS

Einrichten einer Festplatte

Bevor Sie die Platte verwenden können, muß diese eingerichtet werden. Beachten Sie auch das Kapitel 7 »Diskette und Festplatte«.

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

```
fdisk
```

ein, um das Programm zu starten.

Daraufhin wird Ihnen eine Programmauswahl am Bildschirm angezeigt:

```
IBM Personal Computer
Festplatten-Einrichtungsprogramm 3.30
(C) Copyright IBM Corp. 1983, 1987
FDISK-Menü
Aktuelles Festplattenlaufwerk: 1
```

Eine Auswahl angeben:

1. Erstellen einer DOS-Partition
2. Ändern der aktiven Partition
3. Löschen der DOS-Partition
4. Anzeigen der Partitionsdaten
5. Nächste Festplatte auswählen

Auswahl: [1]

Esc -> Zurück zu DOS

Festplatte in Bereiche aufteilen

Betätigen Sie *Return*. Sie haben die Möglichkeit, Ihre Festplatte in mehrere Bereiche (»Laufwerke«) aufzuteilen. Mindestens muß eine primäre DOS-Partition angelegt werden. Diese Möglichkeit muß vor allem dann genutzt werden, wenn Sie eine Festplatte mit mehr als 32 Mbyte besitzen. Ein logisches Laufwerk kann maximal 32 Mbyte groß sein. Für weitere Informationen zu FDISK beachten Sie bitte Kapitel 11.

Betätigen Sie *Return* für das Erstellen einer primären DOS-Partition. Nach erneutem *Return* wird die gesamte Festplatte für die Verwendung als Laufwerk C: eingerichtet.

DOS-Partition

Sollte bereits eine DOS-Partition vorhanden sein, so muß diese erst gelöscht (Punkt 3) werden, bevor die Festplatte neu eingeteilt werden kann.

Nach der Vorbereitung der Festplatte startet Ihr System neu (dazu muß die DOS-Diskette im Diskettenlaufwerk eingelegt sein).

Der Punkt 5 in der Auswahl erscheint nur, wenn mehr als eine Festplatte in Ihrem Computer angeschlossen sind.

Starten Sie das Programm FDISK erneut, und wählen Sie den Punkt 4. Dazu erscheint z.B. (bei IBM Personal System /2, Modell 50):

Anzeigen der Partitionsdaten

Aktuelles Festplattenlaufwerk: 1

Partition	Status	Art	Start	Ende	Größe
-----------	--------	-----	-------	------	-------

C: 1		A PRI DOS	0	609	610
------	--	-----------	---	-----	-----

Gesamtplattenbereich ist 610 Zylinder.

Anzeige der Partition

Es werden Ihnen alle Partitions angezeigt. Zu beachten ist, daß immer nur eine Partition aktiv sein kann (Status = A).

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

FDISK und FORMAT

Betätigen Sie die Taste *Esc* zweimal, um das Programm FDISK zu verlassen.

Nachdem mit FDISK die Festplatte vorbereitet wurde, muß diese noch formatiert werden (Befehl FORMAT – Kapitel 11). Erst dann ist die Platte fertig, um mit ihr arbeiten zu können.

Um die Installation von DOS zu vereinfachen, befindet sich auf Ihrer DOS-Diskette das Programm SELECT:

- Formatierung der DOS-Partition
- Installation der Tastaturroutine
(z.B. KEYB GR für Deutschland)
- Installation des Landescodes (z.B. für Deutschland)
- die DOS-Dateien auf die Festplatte kopieren

Geben Sie für die Installation des deutschen DOS ein:

```
A>select c:\dos 049 gr
```

Die DOS-Dateien werden in das Sub-Directory (Kapitel 8) **DOS** kopiert, der Landescode für Deutschland (Kapitel 13) und der Tastaturcode GR (Kapitel 11 – KEYB GR) verwendet.

SELECT wird beim ersten Installieren von DOS benutzt. Durch SELECT werden alle Daten auf der angegebenen Einheit gelöscht und DOS wird installiert.

Fortfahren (J/N)? J

Nach Betätigung von *Return* erhalten Sie folgende Anzeige vor dem Start der Formatierung:

```
ACHTUNG ! Alle Daten auf der  
Festplatte C: werden gelöscht!  
Formatieren durchführen (J/N)?
```

Nach Beendigung der Formatierung erhalten Sie folgende Anzeigen:

```
Name (max. 11 Zeichen) oder Eingabetaste  
Formatieren beendet  
Systemdateien übertragen  
Weitere Disk./Platte formatieren (J/N)?
```


Dazu wird die Kapazität der neuen Diskette angezeigt. Nach Eingabe von **n** erhalten Sie die Anzeige

DOS-Diskette in Laufwerk A: einlegen,
anschließend eine Taste betätigen

Mit XCOPY wird anschließend automatisch der Inhalt der Diskette in Laufwerk A auf die Festplatte in das Sub-Directory DOS kopiert.

Auf der neuen Partition werden als letztes zwei Dateien erstellt:

- CONFIG.SYS mit dem Inhalt:
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
- AUTOEXEC.BAT mit dem Inhalt:
PATH C:\;C:\DOS;
KEYB GR 437 C:\DOS\KEYBOARD.SYS
ECHO OFF
CLS
DATE
TIME
VER

SELECT beendet Damit ist der Befehl SELECT beendet, und Sie können Ihr System direkt von der Festplatte starten (dabei darf sich im Laufwerk keine Diskette befinden bzw. muß dieses geöffnet sein), die eine für Deutschland passende Startroutine beinhaltet.

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

Installation auf Festplatte prüfen Zum Abschluß überprüfen Sie Ihre Installation von DOS auf der Festplatte:

Damit Ihr Computer von der Festplatte starten kann, entnehmen Sie bitte die Diskette (im anderen Falle versucht das System DOS von der Diskette zu starten). Betätigen Sie *Strg + Alt + Entf* gleichzeitig, um das System neu zu starten.

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

dir

und betätigen Sie *Return*.

Überprüfen Sie anhand der Anzeige, ob alle Dateien vorhanden sind. Die Dienstprogramme befinden sich im Sub-Directory DOS (dir dos).

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

6.4 Die wichtigsten Befehle im Überblick

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

6.4.1 Allgemeines (Aufbau der Dateinamen)

Eingabe von Befehlen und Programmnamen

Wenn Sie den Prompt (z.B. C >) am Bildschirm mit dem Cursor daneben sehen, erwartet DOS die Eingabe eines Befehls. Diesen können Sie in Klein- oder Großbuchstaben eingeben (Kleinbuchstaben behandelt das 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:

```
Diskette/Platte, Laufwerk C:, hat den
Namen BUCH
Verzeichnis von C:\
COMMAND  COM      25979  18.03.87  12.00
DOS       <DIR>      18.08.87  18.15
CONFIG    SYS        37  18.08.87  18.17
AUTOEXEC  BAT        84  18.08.87  18.17
          4 Datei(en) 20373504 Byte 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 Typenbezeichnungen ergeben (diese sind nicht alle zwingend, jedoch gebräuchlich):

Typ	Verwendung für
COM	Programm
EXE	Programm
BAT	Batch-Programm, das 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
OVN	Overlay-Datei
OVL	Overlay-Datei
TXT	Texte (z.B. Briefe, Berichte) oder
DOC	Dokumente/Texte
BAK	Sicherungsdatei
BIN	Binärdatei
DAT	ASCII-Datei (Daten wie z.B. Adressen, Artikel)
DIF	Standard-Datenübergabe-Datei
SYS	Systemdatei (Treiber)
TMP	Temporäre Datei des Programmes LINK
\$\$\$	Arbeitsdatei (temporäre Datei)

Filenamen 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 DOS Texte, Programme usw. vom Namen her unterscheiden kann. Zusätzlich hat jedes Disketten-/Festplattenlaufwerk unter DOS ebenfalls einen Namen:

Anzahl der Diskettenlaufwerke	Disketten- laufwerk-Name	Festplatten- laufwerk-Name	
1 oder 2	A (und B)	ab	C
3	A, B und C	ab	D
4	A, B, C und D	ab	E

*Programme von anderen
Laufwerken starten*

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

C:program
A:program

Wie an den vorherigen Beispielen erkennbar, werden Programmnamen für die Ausführung immer ohne Angabe des File-Typs verwendet.

Wie kann das aktuelle Laufwerk gewechselt werden?

Laufwerke wechseln

Tippen Sie den Namen des neuen aktuellen Laufwerks (z.B. A:) ein, und betätigen Sie die Taste *Return*. Anschließend erscheint als Bestätigung ein neuer Prompt (z.B. A >). Sollte ein von Ihnen eingegebener Befehl bzw. Programmname von DOS nicht gefunden werden, so erscheint am Bildschirm die Meldung:

Falscher Befehl oder Dateiname

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

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

6.4.2 Zeichen zur Zusammenfassung mehrerer Dateien

Wildcardzeichen 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 Wildcardzeichen (* und ?). Im folgenden dazu Beispiele. Auf einer Diskette in Laufwerk A sind die Dateien (Beispieldiskette)

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

gespeichert.

Das Wildcardzeichen *

Sie kopieren alle 5 TXT-Dateien von Laufwerk A nach C:

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

oder

```
copy a:*.txt c:
```

oder

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

Stern () als Wildcardzeichen* 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 der er angegeben wird, ersetzt er alle Zeichen.

Im vorhergehenden zweiten Beispiel (*.txt) steht der * für beliebige 8 Stellen vor dem Punkt. Es werden der Dateiname und -typ von DOS immer separat behandelt.

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

Da der Stern ab der Stelle, an der er eingesetzt wird, alle Zeichen ersetzt, ist eine Eingabe wie *dat.txt nicht erlaubt. DOS akzeptiert die Eingabe trotzdem kommentarlos und behandelt diese wie *.txt.

Das gleiche gilt für den File-Typ. Wenn Sie

```
copy a:test*.* c:
```

eingeben, so wird jede Datei kopiert, die mit den Buchstaben **test** beginnt. Der File-Typ kann beliebig sein. Zusätzlich zu obigen Dateien wird damit auch **TESTNEUE.DOC** übertragen.

Mit *.* werden alle Dateien angesprochen.

Die Eingabe des Kopierbefehls

```
copy a:test.* c:
```

führt im vorliegenden Beispiel zu der Meldung

```
A:TEST.??? Datei nicht gefunden
          0 Datei(en) kopiert
```

da ein File-Name TEST nicht existiert. Richtig könnte der Befehl lauten:

```
copy a:test*.* c:
```

Das Wildcardzeichen ?

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

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

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
```

*Fragezeichen (?) als
Wildcardzeichen*

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

Die beiden Kopierbefehle

```
copy a:test????.txt c:
copy a:test*.txt c:
```

führen somit im vorliegenden Beispiel zum gleichen Ergebnis.

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

*Anzeige eines Disketten-
Inhaltsverzeichnisses
(Directory) mit DIR*

Die Angaben im Abschnitt 6.4.2 zur Zusammenfassung von Dateien gelten auch für die Befehle in diesem Abschnitt. Die folgenden beiden Beispiele führen zum gleichen Resultat:

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

Als Ergebnis des Befehls DIR wird am Bildschirm ein Inhaltsverzeichnis (hier von der Diskette in Laufwerk A) angezeigt (Beispiel):

```
Diskette/Platte, Laufwerk A, hat keinen
Namen
Verzeichnis von A:\
```

```
COMMAND  COM      25979  18.03.87  12.00
PROG      EXE      49065  11.07.87  16.29
AUTOEXEC  BAT         7  11.07.87  16.58
          3 Datei(en)  1381888 Byte frei
```

Aufbau eines Directorys

In der ersten Zeile finden Sie den Namen der Diskette/Festplatte, soweit vorhanden (hier wurde kein Name vergeben). Dieser wird beim Formatieren angegeben oder kann nachträglich mit dem Befehl LABEL bearbeitet werden (siehe Kapitel 11, Befehle FORMAT und LABEL). Sollte kein Name vorhanden sein, so wird der Text »keinen Namen« angezeigt.

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

Am Ende erhalten Sie noch die Daten, wie viele Dateien angezeigt wurden und wieviel freie Speicherkapazität noch auf der Diskette/Festplatte ist (in unserem Beispiel sind auf der Diskette noch 1381888 Zeichen frei für das Speichern von weiteren Daten/Dateien).

Mit DOS ist es möglich, Dateien zu »verstecken«. Zwei solche Dateien sind IBMDOS.COM und IBMBIO.COM. Daher ist die Anzeige mit DIR nicht unbedingt identisch mit der Anzahl der tatsächlich vorhandenen Dateien. Dies können Sie jedoch normalerweise nicht verändern. Dazu sind detailliertere Kenntnisse notwendig, die den Rahmen dieses Buches überschreiten.

In den Zeilen dazwischen finden Sie die Datei-Einträge.

```
COMMAND  COM      25979  18.03.87  12.00

1234567890123456789012345678901234567890
          10          20          30          40
```

Zusammensetzung der Directory-Einträge

COMMAND	File-Name
COM	File-Typ
25979	Dateigröße (in Bytes bzw. Zeichen)
18.03.87	Datum der Erstellung bzw. letzten Änderung
12.00	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 (nach Dateigröße aufsteigend sortiert anzeigen).

Directory »seitenweise« anzeigen

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

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

oder

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


Wenn Sie ein nicht vorhandenes Laufwerk angeben, erhalten Sie die Fehlermeldung:

Ungültige Laufwerksangabe

Haben Sie vergessen, den deutschen Zeichensatz zu laden, so kommt es sehr leicht vor, daß Sie folgenden Befehl eingeben:

dir a>

Statt des Doppelpunktes wurde versehentlich das Zeichen > eingegeben. Sie erhalten nach *Return* die Fehlermeldung

Fehler beim Erstellen der Datei

Kopieren mit COPY Den Befehl COPY haben Sie bereits im Abschnitt 6.4.2 kennengelernt. Weiteres finden Sie im Abschnitt 6.5 für das Kopieren von Texten auf den Drucker und eine zusammenfassende 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 Ausführung des Befehls existiert von AUTOEXEC.BAT eine Kopie gleichen Inhalts mit dem Namen EXEC.BAT (auf der gleichen Diskette/Festplatte und Directory).

*Laufwerksangabe beim
Kopieren*

Normalerweise wird von einem Laufwerk auf ein anderes kopiert. Die Laufwerksbezeichnung müssen Sie nur angeben, wenn diese vom angezeigten Prompt (z.B. C> für aktuelles Laufwerk C) abweicht. Diese Angabe ist immer vor den Dateinamen ohne Zwischenraum mit einem folgenden Doppelpunkt 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 a:keyb.com
```

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

*Datei 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 ähnlich wie COPY angewandt.

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

Der Dateiname AUTOEXEC.BAT wird geändert in 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 a:ansi.doc *.txt
```

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

```
ren a:*.doc *.txt
```

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

```
ren a:ansi bild
```

Die Datei ANSI in BILD umbenennen (Diskette in Laufwerk A). Die Laufwerksbezeichnung darf nur einmal angegeben werden.

```
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.

Vor dem zweiten Dateinamen darf keine Laufwerksbezeichnung angegeben werden (sonst Fehler Ungültiger Parameter).

Wenn der neue Dateiname bereits existiert oder die umzubenennende Datei nicht gefunden wird, erhalten Sie eine Fehlermeldung am Bildschirm:

Doppelter Dateiname oder
Datei nicht gefunden

*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), sind Sie in der Lage, Dateien zu löschen:

```
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 gefragt:

Sind Sie sicher (J/N)?

Die Eingabe *J* und *Return* löscht alles.

```
del a:exec.bat
```

Auf der Diskette in Laufwerk A 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).

6.4.4 Befehl MODE für die Anpassung des Druckers usw.

Anpassung der Peripherie mit MODE

Mit dem Befehl MODE 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.)
4. Vorbereiten und Verwalten der Zeichensatztabellen für die Funktion »Zeichensatztabelle wechseln«

Wir wollen uns hier nur mit dem Drucker befassen. Weitere Erklärungen zum Befehl MODE finden Sie im Kapitel 11.

Drucker einstellen oder umleiten

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 (diese wird von vielen Programmen standardmäßig benutzt) dorthin umleiten:

```
mode lpt1=com1
```

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

Standardeinstellungen des Druckers

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

Manche Programmier- sprachen über- schreiben MODE

Die Standardeinstellung wurde oben entsprechend gekennzeichnet (80 Zeichen, 6 Zeilen). Bitte beachten Sie, daß verschiedene Programmiersprachen (z.B. Basic) Ihre Einstellungen mit MODE

überschreiben (also unwirksam machen). Es muß mit der Programmiersprache die Einstellung erneut vorgenommen werden.

Beispiele mode lpt1:132,8,P

1. Parallel-Schnittstelle
132 Zeichen pro Zeile
8 Zeilen pro Zoll warten, bis der Drucker »empfangsbereit« ist

mode lpt1:80,8

1. Parallel-Schnittstelle
80 Zeichen pro Zeile
8 Zeilen pro Zoll
Fehlermeldung, wenn Drucker nicht »empfangsbereit«

mode lpt1:80,6,P

1. Parallel-Schnittstelle
80 Zeichen pro Zeile
6 Zeilen pro Zoll
warten, bis der Drucker »empfangsbereit« ist

mode lpt2:132,6,P

2. Parallel-Schnittstelle
132 Zeichen pro Zeile
6 Zeilen pro Zoll
warten, bis der Drucker »empfangsbereit« ist

6.5 Drucken von Texten und vom Bildschirm (Hardcopy-Funktion)

*Bildschirminhalt 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 Taste *Druck*. Damit wird alles 1:1 auf das Papier im Drucker »kopiert«. Jedes Zeichen kann jedoch nur dargestellt werden, wenn Sie einen IBM oder kompatiblen Drucker mit 8-Bit-Zeichensatz verwenden. Teilweise werden die Programme GRAPHICS und GRAFTABL dazu benötigt (siehe Kapitel 11).

*Bildschirmausgabe
gleichzeitig auf
Drucker umleiten*

Durch *Strg* und gleichzeitig *Druck* (oder *Strg+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 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 a:config.sys prn
```

copy	Kopierbefehl
a:config.sys	Inhalt dieser Datei im Laufwerk A soll gedruckt (»ausgegeben«) werden
prn	symbolischer Name in 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 Computer 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.

6.6 Erstellen von Sicherungskopien (von Diskette und Festplatte)

*Sicherungskopien von
Diskette/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 eingetippt werden muß

(unter Umständen kann dies Wochen dauern oder nicht mehr möglich sein).

Auch wenn es Jahre gutgeht – fertigen Sie Sicherungskopien. Die Praxis zeigt, daß dies notwendig ist!

6.6.1 Sicherungskopien bei Computern mit zwei Diskettenlaufwerken

Sicherungskopien bei zwei Laufwerken

Der Vorgang läuft wie in Abschnitt 6.3.1 (DISKCOPY) beschrieben ab. Sie sollten darauf achten, daß Sie immer eine 1. und eine 2. Kopie zusätzlich zum Original haben, falls einmal Ihre aktuelle Kopie fehlerhaft ist.

Legen Sie Ihre 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 die kopiert werden soll, in das rechte Laufwerk B ein, und betätigen Sie die Taste *Return*. Damit wird der Kopiervorgang gestartet.

6.6.2 Sicherungskopien bei Computern mit einem Diskettenlaufwerk

Sicherungskopien bei einem Laufwerk

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

```
diskcopy a: b:
```

ein. Betätigen Sie *Return* für den Start des Programmes. Legen Sie die zu kopierende Diskette in Laufwerk A nach Aufforderung ein. Sie werden abwechselnd aufgefordert, die Quellen-(Original-)Diskette bzw. die Zieldiskette einzulegen, bis der Vorgang abgeschlossen ist.

6.6.3 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

*Dateisicherung bei
Festplatte mit BACKUP*

Mit BACKUP gesicherte Dateien dürfen auf keinen Fall mit dem COPY- oder XCOPY-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/f
```

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

Einzelne Dateien 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-Typ 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 gesichert (beachten Sie dazu auch das Kapitel 8 »Das DOS-Inhaltsverzeichnis«).

Durch die Eingabe **/F** werden neue Disketten vor dem Kopiervorgang erst automatisch formatiert. Wurde eine Diskette bereits vorher verwendet (eventuell ebenfalls zur Datensicherung), so werden alle darauf enthaltenen Daten vor dem Start der Datensicherung gelöscht.

*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

Für die Benutzung eines Bandes bzw. Tape-Streamers beachten Sie bitte unbedingt die dort beiliegende Beschreibung.

Kapitel 7

Diskette und Festplatte

RAM = interner Speicher

In Ihren Computer ist ein interner Speicher eingebaut, das RAM (Random Access Memory). Er wird beim Abschalten des Systems 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 (harten) Platten. Darum wird diese auch Harddisk genannt. Beide Speichermedien arbeiten im Prinzip auf die gleiche Art und Weise.

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. A:).

Anzahl der Diskettenlaufwerke	Disketten- laufwerksnamen	Festplatten- laufwerksnamen
1 oder 2	A (und B)	ab C
3	A, B und C	ab D
4	A, B, C und D	ab E

7.1 Aktuelles Laufwerk

*Anwählen eines
Laufwerks*

Geben Sie bei einem Befehl keinen Laufwerksnamen an, so benutzt 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 DOS diesen auf dem aktuellen Laufwerk (außer Sie geben vor dem Befehlsnamen zusätzlich den Laufwerksnamen und einen Doppelpunkt ein).

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

C> a: ↵ -Taste

A>

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

Ungültige Laufwerksangabe

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

7.2 Einteilung von Diskette 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. Die Numerierung der Spuren beginnt immer außen mit Spur 0.

Diskettenarten

5¼"	Einseitig doppelte Dichte	ssdd	160 Kbyte	40	8	512	48
5¼"	Einseitig doppelte Dichte	ssdd	180 Kbyte	40	9	512	48
5¼"	Doppelseitig doppelte Dichte	dsdd	320 Kbyte	40	8	512	48
5¼"	Doppelseitig doppelte Dichte	dsdd	360 Kbyte	40	9	512	48
5¼"	Doppelseitig doppelte Dichte hohe Kapazität	dsddHD	1,2 Mbyte	80	15	512	96
3½"	Doppelseitig	ds	720 Kbyte	80	9	512	135
3½"	Doppelseitig hohe Kapazität	dsHD	1,44 Mbyte	80	18	512	135

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

1 = Größe (")
 2 = Beschreibung
 3 = Art
 4 = Nettokapazität

5 = Spuren
 6 = Sektoren/Spur
 7 = Sektorgröße
 8 = TPI

Kompatibilität Disketten/Laufwerke

5¼"-Laufwerke	(PC und PC/XT-Klasse mit 360 Kbyte Kapazität) Schreib- und Lesezugriff auf Disketten mit 160/180/320/360 Kbyte
5¼"-Laufwerke	(AT-Klasse mit 1,2 Mbyte Kapazität) Schreib- und Lesezugriff auf alle beschriebenen Disketten. Wird mit einem derartigen Laufwerk eine Diskette mit 160/180/320/360 Kbyte beschrieben/formatiert, so kann diese möglicherweise in einem 360-Kbyte-Laufwerk nicht mehr verarbeitet werden!
3½"-Laufwerke	(720 Kbyte Kapazität) Schreib- und Lesezugriff auf Disketten mit 720 Kbyte
3½"-Laufwerk	(1,44 Mbyte Kapazität) Schreib- und Lesezugriff auf Disketten mit 720 Kbyte und 1,44 Mbyte

Informationseinheiten Bit und Byte

Die Kapazität der einzelnen Bereiche wird in Bytes 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 System 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 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.

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 geschrieben sind. Dabei kann es vorkommen, daß die Daten einer Datei über eine ganze Diskette oder Festplatte verstreut sind.

7.3 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

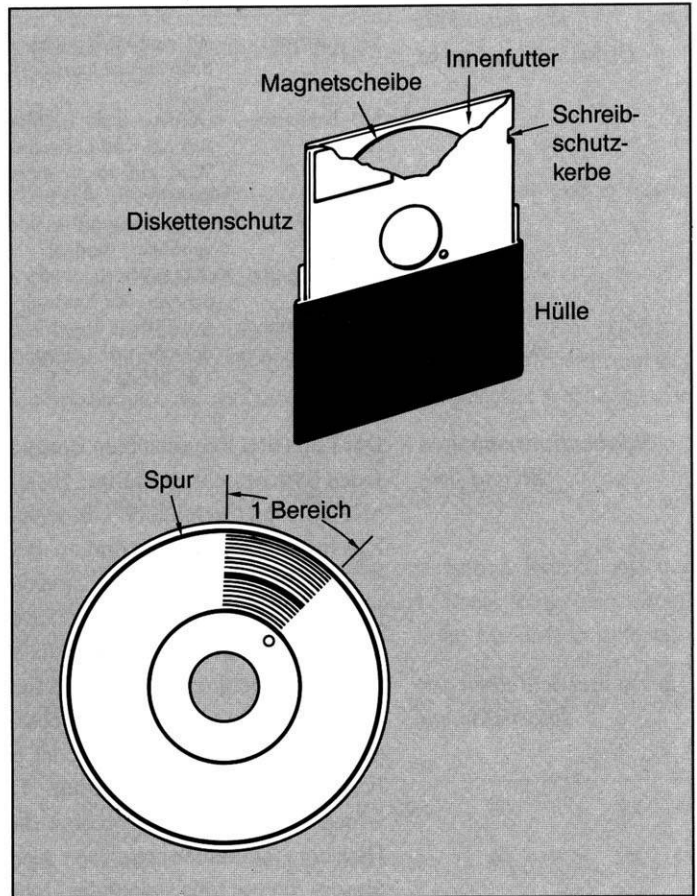


Bild 7.1
Die Diskette

Schreiben offen. $3\frac{1}{2}$ "-Disketten sind vollständig verschlossen. Der Schreib-Lese-Schlitz ist durch einen Metall-Schieberegler geschützt. Dieser wird durch das Einführen der Diskette in ein Laufwerk automatisch geöffnet. Da die Diskette und die darauf gespeicherten Daten sehr leicht zerstört werden können, 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 7.2
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. 3½"-Disketten besitzen einen speziellen Verriegelungsmechanismus in Form eines kleinen Schiebereglers (rechts unten). 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:

Schreibschutz.

Schreibfehler Laufwerk A:

A(bbruch), W(iederholen), U(ebergehen)?

Wenn Sie trotzdem speichern wollen, müssen Sie den kleinen Streifen von der Diskette entfernen bzw. den Schiebereglern nach oben verschieben ($3\frac{1}{2}$ "-Diskette) und **w** betätigen. Im anderen Falle geben Sie den Buchstaben **a** ein, damit der Vorgang abgebrochen wird.

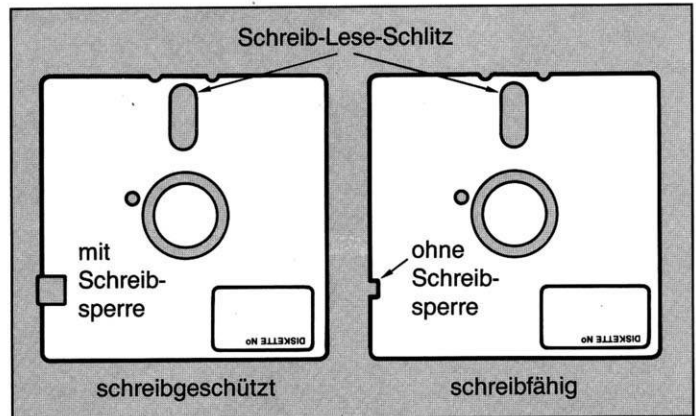


Bild 7.3
Schreibsperre der Disketten

Disketten mit dem Schreib-Lese-Schlitz zuerst in das Laufwerk einschieben. Das Bild ist auf der Seite, an der sich die Laufwerksklappe befindet (also oben). Ist die Diskette ganz eingeschoben, können Sie das Laufwerk schließen. Bei $3\frac{1}{2}$ "-Laufwerken rastet die Diskette durch ein hörbares Klicken ein; ein separates Verschießen ist nicht notwendig.

7.4 Festplatten

Festplatte als Speichermedium

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 wesentlich 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 eine Diskette).

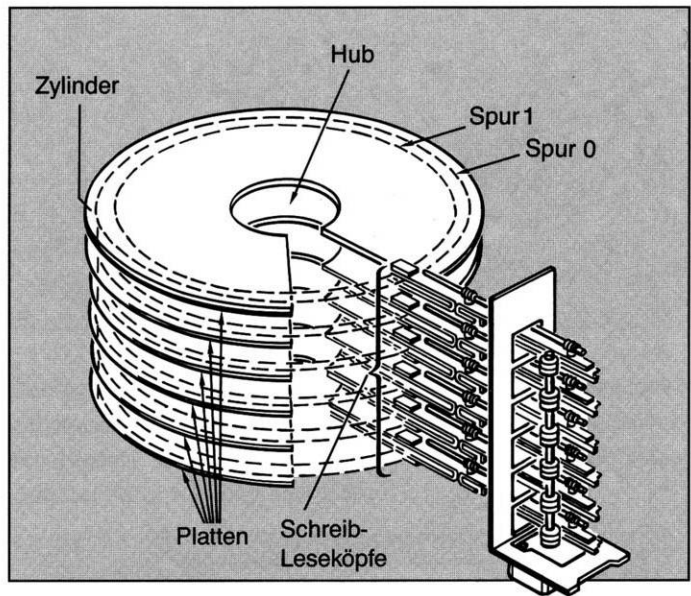


Bild 74
Festplattenlaufwerk

Mit 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 8

Das DOS-Inhaltsverzeichnis

System des Directorys

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

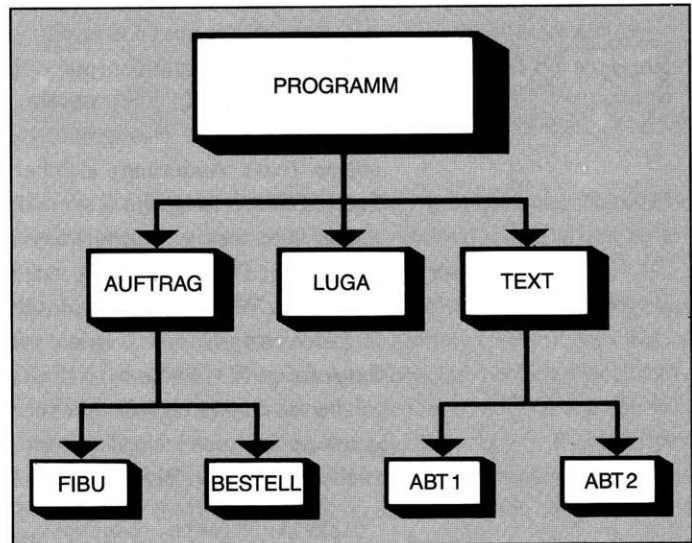


Bild 8.1
Inhaltsverzeichnis
mit Sub-Directories

Hierarchie und Sub-Directories

Die hierarchischen Strukturen von DOS sind mit einem Akten-schrank vergleichbar, in dem die einzelnen Themenbereiche in verschiedenen Schubladen verteilt sind. In einer Schublade können sich aber natürlich auch Unterlagen zu mehreren Gebieten befinden. Jedes Sub-Directory kann man sich als ein solches Gebiet vorstellen.

Die Spitze des Systems ist die »Wurzel« (Root-Directory). Dies ist das erste Verzeichnis, das automatisch durch das Programm

FORMAT (siehe Kapitel 6.3 und 11) angelegt wird. Die weiteren Verzeichnisse, die 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.

*Directory unterteilen in
Sub-Directories*

Mit den Sub-Directories erleichtert Ihnen 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). Hinzu kommt, daß in einem Root-Directory nur eine begrenzte Anzahl von Dateien gespeichert werden kann (die Anzahl ist von der Art des externen Speichers abhängig – 112 bei einer Diskette mit 360 Kbyte). Diese Begrenzung gilt nicht für Sub-Directories.

*Aufbau des Directorys
Anzeigen des Directorys*

Alle Dateien, die mit DOS erstellt werden, erhalten einen Namen. Diese werden zusammen mit weiteren Datei-Informationen in einem Directory (Inhaltsverzeichnis) gespeichert. Von den Informationen sind die Dateigröße, Datum und Uhrzeit der Dateianlage (oder Änderung) sichtbar. Dafür verwenden Sie den Befehl DIR. Er zeigt Ihnen normalerweise alle Dateien an, die in einem Directory gespeichert sind.

Root-Directory

Neben den Dateien 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 (<DIR>). Für das vorhergehende Beispiel kann das Root-Directory wie folgt aussehen (Anzeige mit dem Befehl DIR):

Diskette/Platte, Laufwerk A, hat keinen
Namen

Verzeichnis von A:\

COMMAND	COM	25979	18.03.87	12.00
VERTEIL	BAT	256	15.08.87	01.46
AUFTRAG	<DIR>		10.08.87	14.34
LUGA	<DIR>		10.08.87	14.35
TEXT	<DIR>		10.08.87	14.37
	5 Datei(en)		46126 Byte	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.

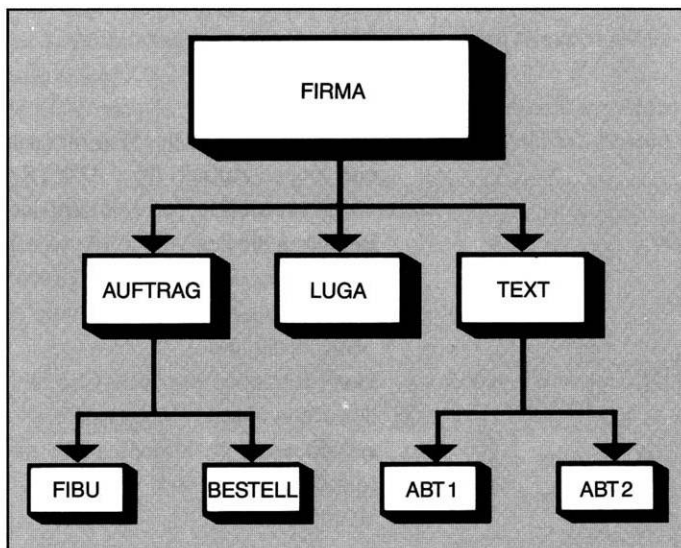


Bild 8.2
Inhaltsverzeichnis mit
Sub-Directory-Namen

Nur die Dateien in **einem Directory** müssen unterschiedliche Namen haben, um sie auch als unterschiedliche Dateien zu erkennen.

Directory wechseln mit CHDIR und 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 Directories werden automatisch zwei Dateien in diesem angelegt (nicht löscher). Diese beiden Einträge bezeichnen das neue Directory (Dateiname .) und das hierarchisch davorliegende (Dateiname ..). Bei Angabe eines Punktes wird das aktuelle, bei zwei Punkten das davorliegende Directory angesprochen. Damit speichert DOS, in welcher Beziehung die einzelnen Directories zueinander stehen.

Directory-Weg angeben mit PATH

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

Beispiel: Das aktuelle Directory ist BESTELL. Das gesuchte Programm EDLIN befindet sich im Sub-Directory TEXT. Der Befehl lautet:

```
...\\text\\edlin
```

oder

```
\\text\\edlin
```

*Einsatz des Back-Slash * Im ersten Fall geht der Weg rückwärts (mit ..\ kommt man immer eine Ebene zurück) über AUFTRAG in das Root-Directory und von dort zu TEXT, in dem sich das Programm EDLIN befindet. Beginnen Sie den Weg mit \, so beginnt DOS im Root-Directory, unabhängig vom aktuellen Directory (zweites Beispiel). Das Zeichen \ erhalten Sie, indem Sie die Tasten *AltGr*+ \ gleichzeitig betätigen.

Maximale »Weg«-Länge Die Länge des Weges darf 63 Stellen nicht überschreiten (ohne den Dateinamen). Programme, die unter älteren 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 (PATH und APPEND)

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

MKDIR oder MD	Anlegen eines Sub-Directorys
CHDIR oder CD	Anzeigen oder Ändern des aktuellen Directorys
RMDIR oder RD	Löschen eines Sub-Directory-Eintrags
PATH	Bezeichnet den Weg durch die Verzeichnisse bis zu einem Programm (externe Befehle/File-Typ COM, EXE oder BAT)
APPEND	Bezeichnet den Weg durch die Verzeichnisse bis zu einer Datendatei (beliebiger File-Typ; auch Programme)

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 Directorys

Die internen Befehle des 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.

Bei den vorstehend angeführten Befehlen handelt es sich mit Ausnahme von XCOPY und APPEND um externe.

Es bietet sich an, alle Programme des DOS in einem separaten Sub-Directory zu speichern (MS-DOS):

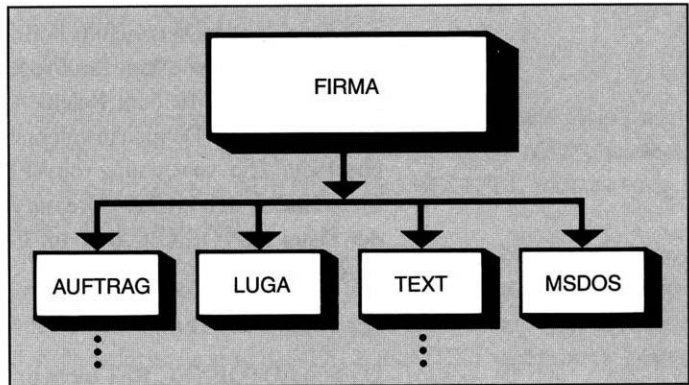


Bild 8.3

Directory-Struktur mit neuem
Sub-Directory für 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:

Falscher Befehl oder Dateiname

*Vor Befehlsnamen
Weg angeben*

Sie müssen vor dem Befehlsnamen den Weg angeben (`\DOS \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=\dos
```

Mehrere Wege vorgeben

Sollten Sie, wie im Kapitel 6 angegeben, die Installation mit SELECT durchgeführt haben, so wird der vorstehende Befehl automatisch beim Start von DOS ausgeführt. Dieser Weg muß nur einmal von Ihnen eingegeben werden. Findet 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:

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

*Gültige Wege
anzeigen lassen*

Durch Eingabe von
path

können Sie sich den oder die derzeit gültigen Wege anzeigen lassen. Bitte beachten Sie, daß bei 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. Mit dem Befehl APPEND (Kapitel 11) können auch die anderen Dateien in unterschiedlichen Directories gespeichert sein. Die Anwendung von APPEND ist mit PATH identisch. Sie befinden 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

Der Befehl muß richtig lauten:

```
type \autoexec.bat
```

oder

```
append \  
type autoexec.bat
```

8.2 Anlegen eines Sub-Directorys (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:

Diskette/Platte, Laufwerk A:, hat keinen
Namen
Verzeichnis von A:\TABELLE

```

.           <DIR>      12.08.87  23.56
..          <DIR>      12.08.87  23.56
          2 Datei(en)    23450 Byte 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 Inhaltsverzeichnis 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 angelegt werden

Diese Fehlermeldung erhalten Sie, wenn ein Sub-Directory oder eine Datei gleichen Namens im Directory schon existiert. Die gleiche Anzeige erfolgt am Bildschirm, wenn die Diskette/Festplatte voll ist und das neue Sub-Directory nicht mehr angelegt werden kann.

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

*Aktuelles Directory
wechseln oder anzeigen
mit CHDIR oder CD*

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

```
cd
```

anzeigen lassen.

Das aktuelle Directory für ein bestimmtes Laufwerk erhalten Sie mit

```
cd b:
```

Sie geben also zusätzlich noch den Laufwerksnamen an.
Beispiele für die Anzeige:

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

FIBU ist das aktuelle Directory in Laufwerk A

```
C:\
```

das Root-Directory ist in Laufwerk C aktuell

```
A:\TABELLE
```

TABELLE ist das aktuelle Directory in Laufwerk A

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 (im aktuellen Laufwerk)

```
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. Das neue aktuelle Directory ist damit im vorliegenden Beispiel AUFTRAG. Wenn Sie den Weg nicht richtig eingeben bzw. der angegebene Directory-Name von DOS nicht gefunden wird, erhalten Sie die Fehlermeldung

Ungültiges Verzeichnis

Ungültiges Verzeichnis

8.4 Löschen eines Sub-Directorys (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 im zu löschenden Sub-Directory keine weiteren Directory-Einträge befinden
3. Nur die Einträge . und .. bleiben bzw. werden durch den Befehl RD gelöscht
4. Das aktuelle Directory kann nicht gelöscht werden

Beispiele

rd luga

Löschen des Sub-Directorys LUGA (dabei muß es sich bei LUGA um ein Sub-Directory des aktuellen Directorys handeln).

rd \auftrag\bestell

\ Weg vom Root-Directory aus beginnen
AUFTRAG über den Weg fortsetzen zu
BESTELL wird gelöscht

Fehler beim Löschen von Sub-Directorys

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 (eventuell falscher Name eingegeben)
2. Das zu löschende Sub-Directory ist nicht »leer«
3. Sie versuchen das aktuelle Directory zu löschen
4. Ein angegebener Directory-Name ist falsch
5. Sie versuchen, das Root-Directory zu löschen

8.5 Anzeigen des Inhaltsverzeichnisses eines Directorys (DIR)

Inhaltsverzeichnis in Sub-Directorys anzeigen mit DIR

Den Befehl DIR haben Sie bereits im Kapitel 6 kennengelernt. Mit diesem Befehl können Sie sich das Inhaltsverzeichnis jedes Directorys anzeigen lassen, unabhängig von Ihrer aktuellen Position.

Beispiele (. * muß nicht
angegeben werden)*

`dir \ *.*`

oder

`dir \`

Anzeigen des gesamten Inhaltsverzeichnisses des Root-Directorys

`dir \auftrag\ *.*`

Weg über Root-Directory zu AUFTRAG und dessen 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 vorherliegenden Directory

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

Im Laufwerk A 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 Directorys

Wird bei DIR statt eines Dateinamens der Name eines Sub-Directorys angegeben, so wird dessen Inhaltsverzeichnis angezeigt. Beispiel (aktuelles Directory muß das Root-Directory sein):

`dir dos`

Anzeige des Inhaltsverzeichnisses des Sub-Directorys DOS

```
dir autoexec.bat
```

Anzeige des Dateieintrags von AUTOEXEC.BAT

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

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

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 vorherliegenden 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 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 6 kennengelernt. Mit ihm können Sie auch Dateien vom einen zu einem beliebigen anderen Directory kopieren. Zu diesem Zweck geben Sie bei »woher« und »wohin« vor den Dateinamen den Weg an.

Beispiele

`copy *.txt \text\abt2`

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

`copy \dos\command.com`

COMMAND.COM aus dem Sub-Directory DOS in das aktuelle Directory 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:\dos\*.com c:`

Aus dem Sub-Directory DOS auf der Diskette in Laufwerk A alle Dateien mit dem File-Typ COM nach dem aktuellen Directory in Laufwerk C 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 Directories
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 Directorys und allen darunter befindlichen auf ein anderes Laufwerk und/oder in ein anderes Directory (einschließlich der Sub-Directory-Namen).

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:1.9.87

Alle Dateien des Sub-Directorys 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 bereits vorhanden ist, bricht XCOPY mit einer Fehlermeldung ab.

Einstellmöglichkeiten von XCOPY

- /S Es werden die Dateien des Sub-Directorys 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 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 1.9.87 oder früher besitzen

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

Zum Kopieren von Datei(en)
eine beliebige Taste betätigen

und

Diskette in Laufwerk B: einlegen,
anschließend eine Taste betätigen

XCOPY liest immer so viele Dateien in den Speicher ein, wie Platz verfügbar ist, und schreibt diese anschließend auf die Zieldiskette/-festplatte. Sie erhalten beim Lesen der Daten am Bildschirm die Meldung:

Quellendatei(en) wird/werden eingelesen

Während des Schreibens auf der Zieldiskette/-festplatte werden Ihnen die kopierten Dateinamen am Bildschirm angezeigt. Sollte der Platz zum Speichern nicht ausreichend sein, so erhalten Sie die Fehlermeldung (der Wert 33 ist ein Beispiel):

Nicht genug Platz auf Diskette/Platte
33 Datei(en) kopiert

XCOPY kopiert häufig schneller als COPY. Bei XCOPY handelt es sich um einen externen, bei COPY um einen internen Befehl. Weitere Informationen zu beiden Befehlen finden Sie im Kapitel 11.

8.8 Zusammenfassendes Beispiel

cd \

Das Root-Directory wird zum aktuellen Directory

md oa

Anlegen des Sub-Directory OA

cd oa

Neues aktuelles Directory wird OA

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

Eingegebenen Text speichern (TEXT.TXT)

dir

Anzeigen des Inhaltsverzeichnisses des aktuellen Directorys OA

```
cd ..
```

Wechseln zum vorhergehenden Directory (Root-Directory)

```
dir /p
```

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

```
rd oa
```

Löschen der 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 DOS noch befragt, ob Sie sicher sind, alles löschen zu wollen)

```
rd oa
```

Löschen des Sub-Directorys OA

Kapitel 9

Datenumleitung und Filter

*Standardein-
und -ausgabegerät
beeinflussen*

DOS-Befehle lesen eingegebene Daten vom **Standardeingabegerät** 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 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 für die Anweisung an das 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
```

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
```

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

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

Die Daten, die in der Datei UNSORT.ASC gespeichert sind, werden aufsteigend sortiert ausgedruckt (PRN).

```
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. Sie wird auf den Drucker (PRN – siehe Kapitel 10) umgeleitet.

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

Das Inhaltsverzeichnis des aktuellen Directorys auf der Festplatte, in Laufwerk C, wird auf die Diskette in Laufwerk A mit dem Namen INHALT.DIR gespeichert.

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

Das Inhaltsverzeichnis des aktuellen Directorys 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 DOS können allein 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 die Daten, die vom Standardeingabegerät kommen, und gibt diese am Standardausgabegerät aus

Den Filter SORT haben Sie bereits in Verbindung mit der Datenumleitung kennengelernt (Abschnitt 9.1). Beachten Sie, daß Filter nur in Verbindung mit Programmen verwendet werden können, die die Standardein- und -ausgabe des 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 DOS (127 Zeichen).

Abarbeitung einer Befehlskette

Während der Abarbeitung einer Befehlskette legt DOS auf der Diskette/Festplatte Dateien an, in der die Ein- und Ausgaben der Befehle zwischengespeichert werden. Für die Namen dieser Dateien wird die aktuelle Uhrzeit aus dem Computer benutzt, die jeweils eine unterschiedliche Bezeichnung ergibt:

21.45.41.98	(Stunden/Minuten/Sekunden/Hundertstel)
15 2D 29 62	Dateiname (die Uhrzeit in hexadezimaler Schreibweise)

Dies ist wegen der Möglichkeit der Nutzung von Systemen 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 *Strg+Untbr* vor dem Ende abbrechen, so sind die zwischengespeicherten Dateien häufig 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. Einfacher ist es jedoch, die Tasten *AltGr+!* (*<*) zu verwenden.

Beispiele für die Befehlsverkettung:

```
dir ! sort > prn
```

Das Inhaltsverzeichnis des aktuellen Directorys 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)      23450 Byte frei
Diskette/Platte, Laufwerk A:, hat keinen
Namen
.                <DIR>      12.08.87  23.56
..               <DIR>      12.08.87  23.56
OB35040F         0    3.09.87  11.53
OB350519         0    3.09.87  11.53
PC              BAS    25948   1.01.87   2.20
TOTAL           DOC     94    1.01.87   3.28
TOTAL           TXT    6671   1.01.87   3.28
Verzeichnis von A:\TABELLE
```

*Beispiele für
Befehlsverkettung*

```
dir ! sort ! more
```

Das Inhaltsverzeichnis des aktuellen Directorys 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. Die Leerstelle vor DAT ist notwendig, da sonst auch Dateien berücksichtigt werden, die die Zeichenkombination DAT im File-Namen besitzen (im Inhaltsverzeichnis befindet sich immer mindestens ein Leerzeichen vor dem File-Typ).

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

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

```
dir | more
```

Das Inhaltsverzeichnis wird »seitenweise« am Bildschirm angezeigt. Alternativ können Sie den DIR-Befehl auch in der folgenden Form anwenden:

```
dir /p  
tree | more
```

Mit dem Befehl TREE wird die hierarchische Struktur eines Disketten-/Festplattenlaufwerks am Bildschirm angezeigt. Diese Anzeige ist meist sehr umfangreich. Daher ist zu diesem Befehl die Verwendung von MORE besonders empfehlenswert. Die Anwendung von TREE finden Sie im Kapitel 11.

Kapitel 10 Gerätenamen

- Gerätetreiber* Zur ordnungsgemäßen Abarbeitung der Ein- und Ausgaben zu den Peripheriegeräten (Tastatur, Bildschirm, Laufwerke, Drucker, Modems usw.) benötigt DOS Gerätetreiber. Diese beinhalten alle Informationen, die das Betriebssystem Ihres Computers 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 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 DOS reserviert und dürfen nicht zur Bezeichnung von Dateien oder Sub-Directories verwendet werden (dies war bis zur DOS-Version 2.11 noch möglich).

Treiber-Name	Peripheriegerät bzw. Schnittstelle
AUX	asynchrone Schnittstelle (oder COM1)
COM1	erste serielle Schnittstelle (V24)
COM2	zweite serielle Schnittstelle (V24)
COM3	dritte serielle Schnittstelle (V24)
COM4	vierte serielle Schnittstelle (V24)
CON	Console (Tastatur/Bildschirm)
LPT1	erste parallele Schnittstelle (normalerweise Drucker/PRN)
LPT2	zweite parallele Schnittstelle
LPT3	dritte parallele Schnittstelle
NUL	nicht vorhandener Ausgang (Dateiende-Erzeugung)
PRN	Drucker (oder LPT1)

Tabelle 10.1
Gerätetreiber

Die Namen der Gerätetreiber können Sie in Befehlen verwenden. Der Gerätetreiber NUL wird normalerweise für das Testen von Anwendungen verwendet. Wird der Treiber als Eingabeeinheit verwendet, so erzeugt er automatisch sofort ein Dateiende-

kennzeichen. Als Ausgabeeinheit simuliert er die Schreib- bzw. Speicheroperationen, nimmt dies aber nicht tatsächlich vor. Daher wird NUL auch als Pseudoeinheit bezeichnet.

Wenn Sie in Ihrem Computer die Schnittstelle COM1 besitzen, kann zusätzlich eine COM2 oder COM3 eingebaut werden. Wenn eine COM4 eingebaut wird, muß mindestens die Schnittstelle COM2 vorhanden sein.

Beispiele `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 (diese werden gleichzeitig auch am Bildschirm dargestellt), werden erst 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. Der Befehl wird **nur** abgeschlossen, wenn von COM1 ein Dateiendekennzeichen kommt – erst dann werden alle Daten gespeichert und die Datei MODEM.DAT gespeichert.

Kapitel 11

Die DOS-Befehle mit Übungen

Beschreibung aller DOS-Befehle

Auf den folgenden Seiten werden die DOS-Befehle Ihres Computers vorgestellt und beschrieben. Die Basis bildet dabei das IBM-PC-DOS 3.30. Anhand vieler Beispiele werden Ihnen diese näher erklärt. Wiederholen Sie die beschriebenen Beispiele an Ihrem eigenen System, um durch die Übung Routine zu erhalten und die Bedienung des 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:	Laufwerksbezeichnung
p	Teil eines Festplattenlaufwerkes
s:	Ursprungslaufwerk
u:	Nummer des Laufwerkes
/x	Zusatzeingabe (Parameter)
volumelabel	Name einer Diskette bzw. Festplattenlaufwerk
mm	Monat oder Minute
dd	Tag
jj	Jahr (zwischen 80 und 99)
jjjj	Jahr (zwischen 1980 und 1999)
hh	Stunden
ss	Sekunden
cc	Hundertstelsekunden
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 6. Für die in diesem Kapitel angegebenen Beispiele werden folgende Laufwerksbezeichnungen verwendet:

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

APPEND *externer Befehl*

Zweck Wenn Sie eine Datei ohne Angabe des Directorys, in dem sie sich befindet, verwenden, durchsucht DOS nur das aktuelle des entsprechenden Laufwerks. Wird die Datei nicht gefunden, erhalten Sie eine Fehlermeldung.

Mit APPEND werden die Wege zu verschiedenen Directorys definiert. Findet DOS eine Datei nicht im aktuellen Directory, so werden anschließend noch alle durchsucht, die mit APPEND definiert wurden. Dadurch können Daten und Texte für Programme, die nur mit dem aktuellen Directory arbeiten können, auch in einem anderen Sub-Directory gefunden werden.

Vergleichbar ist dieser Befehl mit PATH (interner Befehl).

Die Definitionen von APPEND werden von allen DOS-Befehlen benutzt. Aber auch viele andere Programme (z.B. WordStar) funktionieren mit APPEND.

Kommandoaufbau (d:)(weg)append (/X)(/E)
 (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:

Kein APPEND

append \

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

append /X

Muß nach dem Systemstart vor weiteren APPEND-Befehlen eingegeben werden. Damit gelten Weg-Angaben zu APPEND auch für Befehlsdateien (sonst nur mit PATH) mit den File-Typen COM, EXE und BAT.

```
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 wird das bisher Definierte gelöscht und dafür die neuen Angaben 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 die gesuchte Datei nicht gefunden wurde.

```
append
```

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

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

```
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

```
Kein APPEND
```

am Bildschirm.

Bemerkungen APPEND sucht die Datenpfade mit folgenden Systemfunktionen (siehe Anhang VII):

0Fh	Open File (FCB)	Datei eröffnen
23h	Get File Size	Dateigröße holen
3Dh	Open Handle	Dateihandle eröffnen

Bei gleichzeitiger Benutzung von ASSIGN muß immer vorher APPEND angewandt werden.

ASSIGN *externer Befehl*

Zweck Mit ASSIGN kann einem Laufwerk eine andere logische Laufwerksbezeichnung zugeordnet werden (Laufwerksumleitung). Dadurch können Sie andere Laufwerke benutzen als die, auf denen Ihre Programme eingerichtet sind. Bei Benutzung folgender Befehle **niemals** (für das gleiche Laufwerk) verwenden:

BACKUP
RESTORE
DISKCOMP
DISKCOPY
FORMAT
LABEL
JOIN
SUBST
PRINT

Ferner darf ASSIGN nicht für ein Netzwerklaufwerk verwendet werden.

Kommandoaufbau (d:)(weg)assign
(d:)(weg)assign (x=y (...))

x angegebenes Laufwerk
y gesuchtes Laufwerk (dorthin wird umgeleitet)

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 Festplattenlaufwerk 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 Laufwerksbezeichnung A wird wieder das physikalisch erste Laufwerk (A) verwendet.

assign

Alle Laufwerksumbenennungen wieder löschen bzw. den Ursprungszustand wieder herstellen.

Bemerkungen

Leider gibt es immer noch Anwendungssoftware, die standardmäßig die Laufwerke A und B benutzen, ohne daß dies von Ihnen geändert werden könnte. Wenn Sie ein System mit einer Festplatte besitzen, gibt es kein zweites Laufwerk – und normalerweise wurden die Programme auf die Platte kopiert. In diesem Fall kann vor dem Start eines Anwenderprogrammes mit ASSIGN die Laufwerkszuweisung geändert werden (siehe zweites Beispiel).

Sinnvoller ist es, den Befehl SUBST zu verwenden. Eine genaue Beschreibung dazu finden Sie in diesem Kapitel.

Wenn Sie ein nicht vorhandenes Laufwerk verwenden (bei Festplattensystemen mit einem Diskettenlaufwerk und einer Festplattenpartition, alle Spezifikationen ab Laufwerk D), erhalten Sie eine Fehlermeldung:

Ungültiger Parameter

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 (/M). Wenn Sie eine Datei ändern, setzt 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. Mit ATTRIB können die Attribute von Dateien auch angezeigt werden.

Kommandoaufbau (d:)(weg)attrib dateibez(/s)
(d:)(weg)attrib +r dateibez(/s)
(d:)(weg)attrib -r dateibez(/s)
(d:)(weg)attrib +a dateibez(/s)
(d:)(weg)attrib -a dateibez(/s)

+r	auf Read-Only setzen
-r	auf Read-Write setzen
+a	Archivbyte setzen
-a	Archivbyte löschen
/s	Durchsuchen aller Sub-Directories

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 nicht mehr möglich). Wenn Sie anschließend versuchen, die Dateien mit dem Befehl **del *.txt** zu löschen, erhalten Sie eine Fehlermeldung – der Löschbefehl wird nicht ausgeführt:

Zugriff verweigert

attrib *.*

Anzeigen der Attribute (Read-Only und Archivbyte) aller Dateien in einem Verzeichnis. Nur bei Read-Only gesetzten Dateien wird vor dem Dateinamen am Bildschirm ein **R** und bei gesetztem Archivbyte ein **A** angezeigt.

*Beispiel für
Bildschirmdarstellung*

```
A      C:\TEXT\MSDOS.7
      C:\TEXT\CONFIGUR.TXT
A R    C:\TEXT\MSDOS.1
A      C:\TEXT\MSDOS4.TXT
A R    C:\TEXT\MSDOS5.TXT
A R    C:\TEXT\MSDOS6.TXT
A R    C:\TEXT\MSDOS.FUS
      C:\TEXT\APPLY.TXT
A R    C:\TEXT\MSDOS2.TXT
A R    C:\TEXT\MSDOS.2
A      C:\TEXT\MSDOS3.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 (mit Parameter /M) werden alle Dateien kopiert.

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

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

```
attrib -a \DOS\*.*
```

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

```
attrib +a +r *.txt /S
```

Alle Dateien mit dem File-Typ TXT im aktuellen Directory und allen hierarchisch darunterliegenden Sub-Directories auf »Read-Only« setzen. Gleichzeitig wird auch das Attributbyte gesetzt. Anzeige der Attribute zu allen gefundenen Dateien:

```
attrib *.txt/S  
attrib
```

Sie erhalten eine Fehlermeldung, da zumindest eine Dateibezeichnung zusätzlich angegeben werden muß:

Ungültige Anzahl Parameter

```
attrib +r dos
```

Sie erhalten eine Fehlermeldung, da eine Sub-Directory-Bezeichnung nicht mit einem Read-Only versehen werden kann:

Ungültiger Pfad oder Datei nicht gefunden

Bemerkungen

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

Wenn mit BACKUP eine Sicherungskopie erzeugt wird (z.B. auf Diskette), so sind die dort erstellten Sicherungsdateien automatisch auf »Read-Only« gesetzt.

BACKUP *externer Befehl*

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:

*** Die Datei kann nicht gesichert werden ***

Gesichert werden kann:

Festplatte auf Diskette (oder Band) = üblicher Sicherungsvorgang

Festplatte auf Festplatte (oder Band)

Diskette auf Festplatte

Diskette auf Diskette

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

Die erste Angabe (**weg**) bezeichnet 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. Geben Sie zusätzlich auch eine **dateibezeichnung** an, werden nur die angegebenen Dateien gesichert. 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 notwendige Sicherungsdiskette einzulegen).

/f Ist eine neue Diskette für das Sichern von Daten noch nicht formatiert, so erfolgt dies automatisch.

/l(:(d:)(weg)(dateibez)) Erstellen einer Protokolldatei zur Sicherung. Es werden die Sicherungsdisketten-Nummer und die gesicherten Dateien (Namen) gespeichert. Ferner werden das Datum und die Uhrzeit der Sicherung gespeichert.

/d:tt.mm.jj
/t:hh.mm.ss

Wird nur /L angegeben, wird der Name \BACKUP.LOG verwendet.

Nur Dateien mit dem angegebenen oder einem späteren Datum (/d) und/oder Zeit (/t) werden gesichert (das Datum und die Zeit zu einer Datei sehen Sie, wenn Sie den DOS-Befehl DIR eingeben). Das Format der Datumsangabe ist vom Befehl COUNTRY in der Datei CONFIG.SYS abhängig (hier für Deutschland). Beachten Sie hierzu das Kapitel 14 »Die Konfiguration von DOS(CONFIG.SYS/COUNTRY).

/s

Es werden alle Dateien des aktuellen Directorys (Verzeichnis) und die weiter vorhandenen Dateien der Sub-Directories darunter gesichert. Dateien in darüberliegenden Directories werden nicht gesichert (wenn Sie die gesamte Festplatte sichern wollen, sollte das aktuelle Directory das ROOT-Directory sein).

/m

Kopiert nur die Dateien, die seit dem letzten Sicherungsvorgang verändert oder neu erstellt wurden (wird durch das Betriebssystem überwacht) bzw. mit ATTRIB das Archivbyte gesetzt wurde.

Bemerkungen Beachten Sie bitte, daß außer wenn /a verwendet wird die Empfangsdiskette automatisch vor dem Kopieren gelöscht wird!

Beispiele backup c: a: /s

Alle Dateien/Programme der Festplatte Laufwerk C nach Laufwerk A sichern (**nur** das aktuelle und dessen Sub-Directories werden gesichert).

backup c: a: /m/s

Alle Dateien, die im Laufwerk C vorhanden sind und seit dem letzten Backup-Vorgang verändert oder neu erstellt wurden und bei denen mit ATTRIB +a das Archivbyte gesetzt wurde, werden nach Laufwerk A gesichert (nur das aktuelle und dessen Sub-Directory werden gesichert).

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/f/s/m
```

F6-Taste und *Return*, um die Datei GENBACK.BAT zu speichern.
Der Aufruf erfolgt durch:

```
genback c: a:
```

Damit werden vom aktuellen Verzeichnis mit seinen neuen Sub-Directories in Laufwerk C alle Dateien gesichert (/a bedeutet »hinten anhängen«). Die Dateien werden auf Diskette in Laufwerk A gesichert. Vor dem Start werden Sie aufgefordert, die letzte Sicherungsdiskette einzulegen (die Sicherungsdisketten **immer** mit der laufenden Nummer beschriften). Sollte eine neue unformatierte Diskette eingelegt werden, so wird diese automatisch vorher formatiert. Es werden nur Dateien mit gesetztem Archivattribut gesichert.

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 zur Sicherung gefunden |
| 2 | File-Sharing-Konflikte (Netzwerk) – nicht alle Dateien gesichert |
| 3 | Sie haben BACKUP vor dem Bearbeitungsende abgebrochen
(Strg+Untbr) |
| 4 | BACKUP durch 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 *Strg + Untbr* (oder *Strg + C*). Verträgt sich nicht immer mit allen Anwendungsprogrammen (dann am besten auf **OFF** setzen). Mit den Tastenkombinationen *Strg + Untbr* oder *Strg + C* können DOS-Programme immer unterbrochen werden, wenn auf »ON« gestellt wird.

Kommandoaufbau

break on	mit <i>Strg + Untbr</i> können Programme unterbrochen werden
break off	mit <i>Strg + Untbr</i> kann ein Programm nicht unterbrochen werden (ausgenommen Ein-/Ausgaben zu Peripheriegeräten)
break	Anzeigen, ob BREAK auf »off« oder »on« gesetzt ist

Bemerkungen *Strg + Untbr* ist immer gültig bei:
Standardein- und Standardausgaben, Standard-Druckausgabe und Standard-AUX-Operationen (auch COM1/COM2).
BREAK wirkt nicht bei allen Anwendungsprogrammen! Viele Programme verhindern das Abbrechen mit *Strg + Untbr*.
BREAK ist standardmäßig immer auf OFF (nach dem Systemstart). Soll dieser Standardwert auf ON verändert werden, fügen Sie in die Datei CONFIG.SYS den Befehl BREAK=ON ein (siehe Kapitel 14 »Die Konfiguration von DOS«).

CALL *interner Befehl*

Zweck Mit diesem Befehl kann eine andere BATCH-Datei (Kapitel 12) zur Ausführung aufgerufen werden. Nach Beendigung wird die Verarbeitung nach dem CALL-Befehl fortgesetzt. Beachten Sie auch Kapitel 12 für BATCH-Verarbeitung.

Kommandoaufbau call (d:)(weg)dateiname

Beispiele Programm PROG1.BAT:

```
@echo off
echo BATCH-Datei 1
call prog2
call prog4
echo Programmende 1
```

Programm PROG2.BAT:

```
echo BATCH-Datei 2
call prog3
echo Programmende 2
```

Programm PROG3.BAT:

```
echo BATCH-Datei 3
echo Programmende 3
```

Programm PROG4.BAT:

```
echo BATCH-Datei 4
echo Programmende 4
```

Sie erhalten nach dem Start der 1. BATCH-Datei (PROG1) folgende Bildschirmanzeigen:

```
BATCH-Datei 1
BATCH-Datei 2
BATCH-Datei 3
Programmende 3
Programmende 2
BATCH-Datei 4
Programmende 4
Programmende 1
```


Bemerkungen Bitte achten Sie darauf, daß Ihre BATCH-Dateien irgendwann auch ihr Ende erreichen.
Die Grenze für Ihre BATCH-Dateien liegt in der Größe des Speichers Ihres Computers.

CHCP *interner Befehl*

Zweck Mit diesem Befehl werden Zeichensatztabellen für so viele Einheiten (Bildschirm, Drucker) wie möglich ausgewählt. Bitte achten Sie darauf, daß Ihr Drucker eingeschaltet ist. Für weitere Informationen beachten Sie bitte Kapitel 13.

Kommandoaufbau chcp (#)
 # = Zeichensatztabelle

850	Mehrsprachig
860	Portugal
863	Kanada (franz.)
865	Skandinavien
437	Australien, Belgien, Deutschland, Finnland, Frankreich, Großbritannien, Italien, Kanada (engl.), Niederlande, Lateinamerika, Schweden, Schweiz (deutsch und franz.), Spanien, USA

Beispiele chcp

Die aktuelle Zeichensatztabelle wird angezeigt:

Aktive Zeichensatztabelle: 437

chcp 850

Wechseln der aktuellen Zeichensatztabelle. Es wurde die mehrsprachige Tabelle gewählt.

Bemerkungen Der Befehl NLSFUNC muß geladen sein. Wurde ein Einheiten-treiber nicht für die angeforderte Zeichensatztabelle vorbereitet, so kann mit CHCP die Tabelle für die Einheit nicht ausgewählt werden. Beachten Sie dazu auch den Befehl MODE.

CHDIR oder **CD** *interner Befehl*

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

Kommandoaufbau 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 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 wirkungslos, und es erfolgt die Fehlermeldung:

Ungültiges Verzeichnis

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 a:
```

Der Name des aktuellen Directorys wird angezeigt (\, wenn es sich um das Root-Directory handelt):

```
A:\TEXT\ABT1
```

Bemerkung DOS merkt sich generell die aktuellen Directories aller Laufwerke. CD kann mit den Befehlen SUBST und JOIN auch »überlistet« werden. Mit dem Befehl JOIN können Sie einem Laufwerk eine Sub-Directory-Bezeichnung zuweisen. Machen Sie dieses Sub-Directory zum aktuellen, so »befinden« Sie sich im aktuellen Laufwerk B, obwohl als Prompt A> angezeigt wird:

```
A> join b: a:\lauf-b
```

```
A> cd \lauf-b
```

```
A> dir
```

CHKDSK *externer Befehl*

Zweck CHKDSK überprüft eine Festplatte oder Diskette. Es werden alle Directories (Root- und Sub-Directories) mit allen Dateieinträgen überprüft. Vorhandene Fehler können mit dem Programm (Parameter /F) korrigiert werden. CHKDSK zeigt auch eine Statusmeldung am Standardausgabegerät an (Bildschirm). Sie sollten CHKDSK immer anwenden, wenn der Verdacht besteht, daß eine Datei oder Festplatte/Diskette fehlerhaft ist. Das Programm (*CHeck DiSK* = überprüfe Platte) liest die Inhaltsverzeichnisse und überprüft diese 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:)(weg)(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. Zusätzlich werden die Verzeichnisnamen und Datei-/Programmnamen angezeigt (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/Directory abspeichern (diese Datei kann später am Bildschirm angezeigt oder gedruckt werden).

Folgende Angaben enthält der Status-Report am Bildschirm:

- Disketten-/Festplattenname
- Datum und Uhrzeit, wann Diskette/Festplatte erstellt wurde
- Gesamtkapazität der Diskette/Festplatte in Byte
- Anzahl und Gesamtgröße der geschützten Dateien
- Anzahl und Gesamtgröße der Sub-Directories
- Anzahl und Gesamtgröße der Dateien/Programme
- Anzahl der Bytes in fehlerhaften Sektoren
- Angabe des noch freien Speichers auf der Diskette/Festplatte in Byte
- Gesamtgröße des RAM-Speichers (max. 654366)
- Angabe des noch freien Speichers im RAM für Programmausführung

Beispiel einer CHKDSK-Ausgabe (für eine Festplatte mit 20 Mbyte Kapazität und einer RAM-Größe von 1 Mbyte):

```
21170176 Byte Gesamtkapazität
      53248 Byte in 3 geschützten Dateien
      59392 Byte in 25 Verzeichnis(sen)
15058944 Byte in 859 Benutzerdatei(en)
      5998592 Byte auf Diskette/Platte
          verfügbar
      654336 Byte Gesamtspeicher
      575520 Byte frei
```

Bei den geschützten Dateien handelt es sich um die DOS-Systemprogramme IBMBIO.COM und IBMDOS.COM. Werden 3 geschützte Dateien angegeben, so hat Ihre Diskette/Festplatte auch einen Namen. Dieser Name besitzt einen eigenen geschützten Dateieintrag. Einige Anwendungsprogramme erstellen ebenfalls geschützte Dateien. Sub-Directories benötigen Platz für die Speicherung der Inhaltsverzeichnisse.

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 zusammenhängend

Wird eine Gruppe von Dateien überprüft (z.B. *.*), so werden nur die nicht in einem Stück gespeicherten Dateien dargestellt.

C:\PROGRAM\TEXTE\M&T-DOS.3-3\TOTAL.BAK

Enthält 11 nicht zusammenhängende Blöcke

C:\PROGRAM\TEXTE\M&T-DOS.3-3\KAPITEL.1

Enthält 2 nicht zusammenhängende Blöcke

C:\PROGRAM\TEXTE\M&T-DOS.3-3\TOTAL.TXT

Enthält 5 nicht zusammenhängende Blöcke

DOS verwaltet den Speicher in Sektoren. Wird eine neue Datei gespeichert, so beginnt die Suche nach freien Sektoren am Anfang der Diskette/Festplatte. Findet 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 »nicht zusammenhängenden« Blöcken gespeichert.

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

Sie sollten Ihre Festplatte von Zeit zu Zeit überprüfen, da sehr viele nicht zusammenhängende Blöcke zu einer wesentlichen Verlangsamung Ihres Computers führen. Die »geteilten« Dateien können wieder zusammenhängend gespeichert werden, indem Sie mit BACKUP die gesamte Diskette oder Festplatte sichern, diese anschließend formatieren oder löschen und die Sicherungsdiskette(n) mit RESTORE wieder zurücksichern (dabei werden die Dateien fortlaufend ohne Teilung geschrieben).

Sollte Ihre Diskette oder Festplatte einen Namen besitzen, so erhalten Sie z.B. folgende Anzeige:

*Beispiel einer
CHKDSK-Ausgabe*

Diskette/Platte BUCH

erstellt 18 Aug 1987, 18.15

21170176 Byte Gesamtkapazität

53248 Byte in 3 geschützten Dateien

59392 Byte in 25 Verzeichnis(sen)

15058944 Byte in 859 Benutzerdatei(en)

122880 Byte in fehlerhaften Sektoren

5875712 Byte auf Diskette/Platte
verfügbar

654336 Byte Gesamtspeicher

575520 Byte frei

Geben Sie zusätzlich zum Befehl den Parameter /V an, so erhalten Sie in der Anzeige auch alle überprüften Dateinamen. Beispiel (5¼-Zoll-Diskette mit 360 Kbyte; Computer mit 512 Kbyte RAM):

```
Verzeichnis A:\
  A:\IBMBIO.COM
  A:\IBMDOS.COM
  A:\COMMAND.COM
  A:\CHKDSK
Verzeichnis A:\C
  A:\C\AUTOEXEC.BAT
  A:\C\CONFIG.SYS
  ...
  A:\C\SORT.EXE
362496 Byte Gesamtkapazität
 45056 Byte in 2 geschützten Dateien
  2048 Byte in 1 Verzeichnis(sen)
310272 Byte in 38 Benutzerdatei(en)
  5120 Byte auf Diskette/Platte
      verfügbar
524288 Byte Gesamtspeicher
477952 Byte frei
```

Wenn nicht zuordenbare Bereiche von Daten auf der Diskette/Festplatte gefunden werden (nur wenn beim Befehl der Parameter /F angegeben wurde), erhalten diese von CHKDSK einen eigenen Dateinamen (**FILE0000.CHK**, **FILE0001.CHK** usw.). Sie können sich anschließend die Dateien ansehen und, falls sie keine wichtigen Informationen enthalten, löschen.

CLS *interner Befehl*

Zweck CLS löscht den Bildschirm (Standardausgabeeinheit) 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 DOS dar und wird beim Systemstart in den Speicher geladen). Mit diesem externen Befehl wird COMMAND als Sekundär-Befehlsinterpreter geladen. Sie können statt COMMAND.COM auch einen anderen Befehlsinterpreter verwenden.

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 160 und 32768)
	/p	ein neuer COMMAND.COM wird geladen und der alte gelöscht (es kann mit EXIT nicht in die vorherige Ebene zurückgekehrt werden); es wird zusätzlich nach Datum und Uhrzeit abgefragt bzw. AUTOEXEC.BAT ausgeführt
	/c Befehl	definiert eine Zeichenkette bis zur maximalen Eingabelänge von 127 Zeichen (diese wird von COMMAND.COM als DOS-Befehl ausgeführt)
	/d	zusammen mit /p erfolgt damit keine Abfrage nach Datum und Uhrzeit bzw. wird AUTOEXEC.BAT nicht 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 unverän-

dert. Wenn Sie zum Beispiel mit PATH Änderungen vornehmen, so werden diese entsprechend gespeichert. Es erscheint folgende Anzeige am Bildschirm:

```
The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp 1981, 1987
          (C)Copyright Microsoft Corp 1981, 1987
```

Nach Eingabe des Befehls

```
exit
```

kehren Sie wieder in die vorhergehende Befehlsebene zurück und alle Eingaben mit PATH, SET usw. in den Environment-Bereich der höheren Ebene sind wieder gelöscht; es gelten wieder die alten Definitionen. Mit EXIT können Sie nicht zurückkehren, wenn der Parameter /P angegeben wurde.

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

Der bisher geladene COMMAND.COM wird durch einen neuen ersetzt (/p), bzw. vom Sekundär-Befehlsinterpreter kann nicht zum Primär-Befehlsinterpreter zurückgekehrt werden. 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 gestartet (Sekundär-Prozeß), 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.

Geben Sie nach COMMAND einen ungültigen Wert an (nicht vorhandenes Sub-Directory usw.), erhalten Sie die Fehlermeldung

Angeg. COMMAND-Verzeichnis fehlerhaft

Diese Meldung erhalten Sie auch bei Angabe eines Sub-Directorys, in dem der Befehlsinterpreter nicht gefunden wird. Verwenden Sie bei /E einen Wert kleiner als 160 oder größer als 32768, erhalten Sie die Fehlermeldung

Ungültige Umgebungsgröße angegeben

Der Environment-Bereich wird sehr häufig auch »Umgebung« genannt.

Geben Sie zusätzlich zu /C den Parameter /P an, so wird nur /C vom System beachtet.

Bei der Angabe des Parameters /P erhalten Sie die gleichen Anzeigen wie beim Start von DOS:

Systemdatum: Sa. 6.09.1987

Neues Datum (tt.mm.jj) eingeben:

Systemzeit: 14.04.29,37

Neue Zeit (hh.mm.ss) eingeben:

The IBM Personal Computer DOS-Version 3.30

Geben Sie zusätzlich zu /P noch /D an, entfällt die Abfrage nach Datum und Uhrzeit.

Der Parameter /D ist nicht in Ihrem DOS-Handbuch dokumentiert.

Die Angabe eines nicht bekannten Parameters führt zu keiner Fehlermeldung; dieser wird lediglich ignoriert.

Jedes Laden eines neuen Befehlsinterpreters (Sekundär-Prozeß), benötigt zusätzlich 3968 Byte Speicherplatz. Selbstverständlich benötigt auch ein größerer Environment-Bereich zusätzlichen Speicher. Mit CHKDSK können Sie feststellen, wieviel RAM-Speicher Ihnen noch zur Verfügung steht. Dies ist insofern wichtig, da viele Anwendungsprogramme eine nicht unbeträchtliche Menge an RAM-Speicher benötigen (beachten Sie die Angaben in den Handbüchern) und der größte verwendbare Speicherplatz für DOS 640 Kbyte RAM beträgt.

Beachten Sie zu diesem Befehl auch das Kapitel 14.

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, sollten Sie den Befehl DISKCOMP verwenden). Werden mehr als 10 Unterschiede festgestellt, bricht das Programm den Vergleich automatisch ab. Es wird ebenfalls geprüft, ob das Dateiendekennzeichen gespeichert ist. Gefundene Fehler werden am Bildschirm angezeigt.

Die Vergleiche erfolgen zeichenweise, und die Bildschirmausgabe erfolgt in HEX-Darstellung. Die Bedeutung der einzelnen Werte finden Sie im Anhang IV.

Kommandoaufbau (d:)(weg)comp ((d:)(weg)dateiname1 ((d:)(weg)dateiname2))

Beispiele comp a:text1 a:text2 **Return**
 A:TEXT1 und A:TEXT2
 Dateien haben unterschiedliche Größe
 Weitere Dateien vergleichen (J/N)?

Stellt COMP fest, daß die beiden angegebenen Dateien eine unterschiedliche Größe besitzen, wird kein Vergleich vorgenommen, und Sie erhalten die vorher dargestellte Anzeige am Bildschirm (Standardausgabegerät).

Geben Sie bei der Frage, ob Sie weitere Dateien vergleichen wollen **j** ein, so erhalten Sie folgende Anzeigen (natürlich muß jede Frage mit einer Eingabe beantwortet werden, bevor die nächste erscheint):

Ersten Dateinamen eingeben
 Zweiten Dateinamen oder Laufwerk eingeben

Findet COMP eine von Ihnen angegebene Datei nicht, so erhalten Sie die Fehlermeldung:

(d:)(weg)(dateiname) - Datei nicht gefunden

Geben Sie nur den Befehlsnamen ein, so fragt das Programm Sie anschließend nach den beiden zu vergleichenden Dateinamen. Wird von Ihnen nur eine Dateibezeichnung angegeben, werden

Sie nach dem Start von COMP noch nach dem zweiten Namen gefragt.

Bei Angabe von zwei vorhandenen, identischen Dateien erhalten Sie folgende Anzeigen:

```
A:TEXT4 und A:TEXT5
EOF-Markierung nicht gefunden
Dateien sind identisch
Weitere Dateien vergleichen (J/N)?
```

Bei der EOF-Markierung handelt es sich um das Zeichen für Dateiende. Dieses fehlt vor allem sehr häufig bei Programmdateien. Dabei handelt es sich um keinen Fehler, da die Größe einer Datei auch im Inhaltsverzeichnis vermerkt ist und viele Programme diesen Wert benutzen. Diese Meldung hat in erster Linie informativen Charakter.

Bei Angabe von zwei vorhandenen, in der Größe identischen Dateien erhalten Sie zum Beispiel folgende Anzeigen, wenn Unterschiede festgestellt werden:

```
Vergleichsfehler bei OFFSET 0
Datei 1 = 33
Datei 2 = 72
Vergleichsfehler bei OFFSET 8
Datei 1 = 61
Datei 2 = 25
```

Der Vergleich von zwei Dateien erfolgt immer zeichenweise. Dabei hat das erste Zeichen das **OFFSET 0**, das zweite **1** usw. Ihnen werden das Offset und der unterschiedliche Wert der beiden Zeichen angezeigt. Der Wert wird immer hexadezimal dargestellt (eine Tabelle für die Umrechnung bzw. das entsprechende Zeichen finden Sie im Anhang IV).

Bei den Werten in dem vorhergehenden Beispiel handelt es sich um:

```
33 = Ziffer 3
72 = Buchstabe r
61 = Buchstabe a
25 = Zeichen %
```

COMP bricht den Vergleich von zwei Dateien **immer** ab, wenn 10 Vergleichsfehler gefunden wurden – Fehlermeldung:

```
10 Vergleichsfehler - Vergleich beendet
```

Sie können statt Dateinamen auch Laufwerksbezeichnungen oder Sub-Directory-Namen (Root-Directory) angeben. In diesem Fall werden alle in dem angegebenen Directory vorhandenen Dateien verglichen. Beispiele:

```
comp *.txt *.bak
```

Es werden alle Dateien mit dem Typ TXT mit denen des Typs BAK verglichen.

```
comp *.txt b:
```

Alle Dateien mit dem Typ TXT im aktuellen Laufwerk/Directory werden mit den im Namen identischen Dateien im Laufwerk B (aktuelles Directory) verglichen.

```
comp \text\ab1 \text\abt2
```

Alle Dateien im Sub-Directory ABT1 werden mit denen in ABT2 verglichen (soweit Dateinamen aus ABT1 in ABT2 gefunden werden).

```
comp \dos command.com
```

Alle Dateien im Sub-Directory DOS werden mit der Datei COMMAND.COM verglichen. Dies führt natürlich fast ausschließlich zur Fehlermeldung

Dateien haben unterschiedliche Größe

bzw. zu vielen Fehlermeldungen, da normalerweise nur maximal eine Datei identisch sein dürfte.

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 auch verketteten (Concatenation). Ebenso ist die Übertragung von Daten zwischen verschiedenen Ein- und Ausgabekanälen des Computers (z.B. vom Bildschirm zum Drucker) möglich.

Werden mehrere Dateien kopiert, so zeigt Ihnen COPY den jeweils gerade bearbeiteten Dateinamen an.

Ich habe die Besonderheiten von COPY hier genauest möglich dargestellt. Die Beispiele und Angaben können jedoch nur einen Teil der Möglichkeiten beschreiben, da diese sehr vielfältig sind.

Kommandoaufbau `copy (/x)(d:)(weg)dateiname1(/x) (d:)(weg)dateiname2 (/x)`
`copy (/x)(d:)(weg)dateiname1 d: (/x)`
`copy (/x)(d:)(weg)dateiname1 (d:)weg (/x)`
`copy (d:)weg (d:)(weg) (/x)`
`copy (/x)(d:)(weg)dateiname1(/x)(+(d:)(weg)dateiname1)(/x)`
`(d:)(weg)dateiname2`

usw.

`dateiname1` Name der zu kopierenden Datei(en)

`dateiname2` Name der neuen Datei(en)

`d:` Name des Laufwerkes

`/x` `/v` nach dem Kopieren auf richtige Aufzeichnung der Daten überprüfen

`/a` Texte kopieren (ASCII-Files) – das Dateiendekennzeichen wird nach dem Kopieren hinzugefügt bzw. bis dorthin wird kopiert

`/b` binäre Dateien kopieren (in der Regel Programme) – alle Zeichen einschließlich des Dateiendekennzeichens (*Strg+Z* bzw. Hexadezimal **1A**) werden kopiert

Beispiele `copy test.txt a:`

Kopiert die Datei TEST.TXT vom aktuellen Laufwerk/Directory mit dem gleichen Namen auf die Diskette in Laufwerk A in das dort aktuelle Directory.


```
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 autoexec.bat con
```

Dateiinhalt von AUTOEXEC.BAT auf den Bildschirm (Geräte-name **CON**) »kopieren«.

```
copy autoexec.bat prn
```

Dateiinhalt von AUTOEXEC.BAT auf den Drucker (Gerätename **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. *Strg+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 Directorys 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 Dateinamensangabe darf **nur** erfolgen, wenn der Name der Datei in dem Ziel-Directory ein anderer sein soll).

```
copy \schneide \schneide2
```

Kopieren aller Dateien aus dem Sub-Directory **schneide** in das bestehende Sub-Directory **schneide2** mit den gleichen Namen (kopieren kompletter Directories).

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

Aus den Inhalten der Dateien AUTOEXEC.BAT und CONFIG.SYS wird eine neue Datei mit dem Namen START.DOC gebildet (»Concatenation«). Der Vorgang erfolgt im aktuellen Laufwerk/Directory.

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

Alle Inhalte der Dateien mit dem File-Type SYS werden in der neuen Datei TEST.DOC zusammengefaßt (aktuelles Laufwerk/Directory).

```
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 aktuelle Datum und die Uhrzeit erhält. Bei Programmen und Binärdateien vor COPY /B angeben.

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

Es werden alle Dateien mit dem File-Type TXT aus dem aktuellen Directory kopiert. Die Datei, in der alle Daten gesammelt werden, erhält den Namen TEST.ALT (Concatenation).

```
copy autoexec.bat+ \NEU
```

Dieser Kopierbefehl wird nicht ausgeführt; die Dateien des Sub-Directories NEU werden gesammelt, bis ein Dateiendekennzeichen gefunden wird. Die »Sammlung« wird unter dem Dateinamen AUTOEXEC.BAT gespeichert.

```
copy autoexec.bat+con
```

Die nach diesem Befehl am Bildschirm eingegebenen Zeilen werden am Ende der Datei AUTOEXEC.BAT angefügt, wenn *F6* und *Return* betätigt wurde.

```
copy *.txt+*.doc+*.bak+,, \neu
```

Hier werden **nicht** alle Dateien mit den File-Typen TXT, DOC und BAK gesammelt mit dem aktuellen Datum und Uhrzeit in das

Sub-Directory NEU kopiert. Vielmehr sucht COPY die erste Datei mit dem Typ TXT, merkt sich den File-Namen und sucht mit diesem die File-Typen DOC und BAK ab. Findet es dazu passende Dateien, so werden diese gesammelt und mit dem File-Namen und dem File-Typ TXT in das Sub-Directory NEU kopiert. Dort befindet sich anschließend eine Datei.

Beispiele Auf dem aktuellen Laufwerk/Directory befinden sich folgende Dateien:

A.TXT
B.TXT
A.DOC
C.DOC
B.DOC
C.BAK
B.BAK
A.BAK

Im Sub-Directory NEU wird eine neue Datei mit dem Namen A.TXT angelegt, in der alle Texte (ausgenommen C.DOC und C.BAK) gespeichert werden (diese erhält das aktuelle Datum/Uhrzeit). Es werden also die File-Namen zu den ersten angegebenen File-Typen gesucht und die entsprechenden zu den weiter angegebenen Typen. Wäre auch eine Datei mit dem Namen C.TXT vorhanden, so würden alle Texte gesammelt.

Folgende Dateien sind vorgegeben:

A.BAT	mit dem Dateiinhalt	1
B.BAT	mit dem Dateiinhalt	2
C.BAT	mit dem Dateiinhalt	3
NEU	mit dem Dateiinhalt	0

`copy neu+*.bat`

Zu dem in NEU gespeicherten Text werden die Inhalte aller Dateien mit dem File-Typ BAT hinzugefügt. Dies gibt einen neuen Dateiinhalt in NEU:

0
1
2
3

`copy c.bat+*.bat`

Zu dem in C.BAT gespeicherten Text werden die Inhalte aller Dateien mit dem File-Typ BAT hinzugefügt. Während der Ausführung des Befehls erhalten Sie folgende Anzeige am Bildschirm:

```
C.BAT
A.BAT
B.BAT
C.BAT
Inhalt der Zieldatei vor dem Kopieren
verloren gegangen
Inhalt der Zieldatei vor dem Kopieren
verloren gegangen
```

Anschließend sind in der Datei C.BAT folgende Daten gespeichert:

```
3
1
2
```

Es werden die Daten aus den Dateien C.BAT, A.BAT und B.BAT gesammelt und in die Datei C.BAT gespeichert. Dabei geht der bisher unter C.BAT gespeicherte Inhalt gemäß der Anzeige verloren. Dies ist zwar logisch, entspricht jedoch nicht dem Ergebnis, da der bisherige Inhalt bereits als erstes »aufgesammelt« wurde.

```
copy a.bat+*.bat
```

Zu dem in A.BAT gespeicherten Text werden die Inhalte aller mit dem File-Typ BAT hinzugefügt. Während der Ausführung des Befehls erhalten Sie folgende Anzeige am Bildschirm:

```
A.BAT
A.BAT
Inhalt der Zieldatei vor dem Kopieren
verloren gegangen
B.BAT
C.BAT
```

Anschließend sind in der Datei A.BAT folgende Daten gespeichert:

1
2
3
1
2

Es werden die Daten aus den Dateien A.BAT, B.BAT und C.BAT gesammelt und in die Datei A.BAT gespeichert. Dabei geht der bisher unter A.BAT gespeicherte Inhalt gemäß der Anzeige verloren. Dies ist zwar logisch, entspricht jedoch nicht dem Ergebnis, da der bisherige Inhalt bereits als erstes »aufgesammelt« wurde.

CTTY *interner Befehl*

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

Kommandoaufbau `ctty device`

<code>device</code>	AUX	Hilfseingang (normalerweise COM1)
	CON	Tastatureingabe/BildschirmAusgabe (Standard)
	COM1	erste serielle Schnittstelle
	COM2	zweite serielle Schnittstelle (soweit vorhanden)

Beispiele `ctty aux`

Anschließend 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 Bei vielen Programmen und Programmiersprachen (z.B. Basic) funktioniert dieser Befehl nicht, da diese die Schnittstellen (z.B. Bildschirm) direkt ansprechen, also das DOS »umgehen«.

DATE *interner Befehl*

Zweck Anzeigen und/oder Ändern des Tagesdatums (Systemdatum). Ein einmal eingegebenes Datum wird automatisch täglich um Mitternacht weitergezählt, soweit Ihr Computer eine Real-Time-Clock (RTC) besitzt (diese muß normalerweise mit einem speziellen Befehl einmalig eingestellt werden und kann durch den Befehl DATE jederzeit geändert werden).

Besitzt Ihr System keine RTC, so wird das Datum nur so lange weitergeschaltet, bis Sie den Computer abschalten. Bei erneutem Starten steht Ihnen das zuletzt eingegebene bzw. gespeicherte Datum zur Verfügung.

Alle Systeme der AT- und Personal-System/2-Klasse besitzen eine RTC.

Kommandoaufbau

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 16.9.87

Das neue Systemdatum ist anschließend der 16. September 1987. Es erscheint keine weitere Anzeige auf dem Bildschirm bzw. es kommt sofort wieder der Prompt.

date

Anschließend wird das gespeicherte Systemdatum angezeigt

Systemdatum: Mi. 16.9.1987

Neues Datum (tt.mm.jj) eingeben:

und Sie können ein neues Datum eingeben oder durch Drücken der Taste *Return* das Datum unverändert belassen.

Wenn Sie keine gültige Eingabe vornehmen, erhalten Sie die Fehlermeldung:

Ungültige Datumseingabe

Anschließend haben Sie die Möglichkeit, das Datum in der richtigen Form einzugeben.

Bemerkungen Wenn Sie in der Datei CONFIG.SYS (Kapitel 13) einen anderen Landescode als den deutschen vermerken, wird das Datum in einer anderen Schreibweise angefordert:

mm.tt.jj

oder

jj.mm.tt

DEBUG *externer Befehl*

Zweck Es handelt sich primär um ein Testhilfeprogramm. Ferner ist es ein Werkzeug zur Programmierung, mit dem in den Speicher geladene Programme anschließend getestet und/oder modifiziert werden können. Sie haben die Möglichkeit, Dateien zu laden, zu ändern und anzuzeigen. Für die Arbeit mit DEBUG sind Kenntnisse mit dem Maschinencode der Prozessoren 8088, 80286 usw. erforderlich.

Zu diesem Befehl finden Sie in den meisten DOS-Handbüchern wenig oder keinerlei Angaben. Genauere Beschreibungen würden auch den Rahmen dieses Buches sprengen, da zur Bedienung von DEBUG grundlegende Kenntnisse über die internen Strukturen des Speichers, den Maschinencode usw. notwendig sind. Daher lernen Sie im folgenden die Befehle des DEBUG in kurzer Form kennen. Zur Anwendung beachten Sie bitte, daß es bei fehlerhaften Eingaben sehr leicht zu irreparablen Fehlern auf der Diskette oder Festplatte kommen kann! Deshalb empfehle ich Ihnen, bei der Anwendung von DEBUG immer nur mit einer Diskette und einer Kopie von DOS zu arbeiten.

Kommandoaufbau (d:)(weg)debug ((dateiname) (variablen)) (< eingabe)

Eine Datei kann direkt hinter DEBUG angegeben oder nach dem Start des Befehls geladen werden. DEBUG besitzt eigene Befehle. Wenn Sie diese in einer eigenen Datei speichern (mit COPY, EDLIN usw. kann diese erstellt werden), kann diese als **eingabe** beim Aufruf von DEBUG angegeben werden. Während der Ausführung von »eingabe« haben Sie selbst keine Möglichkeit, eine Eingabe zu tätigen. Daher dürfen Sie in diesem Falle nie vergessen, als letzte Zeile den Befehl zum Abschluß von DEBUG (**Q**) in der Eingabedatei anzugeben.

Im folgenden wird Ihnen anhand eines Beispiels die Arbeit mit den DEBUG-Befehlen beschrieben. Am Ende der Beschreibung zu diesem DOS-Befehl finden Sie eine alphabetische Auflistung der DEBUG-Befehle mit einer kurzen Beschreibung der Arbeitsweise.

Anwendungsbeispiele zu DEBUG

Im Kapitel 12 (»Die BATCH-Befehle und ihre Anwendung«) benötigen Sie ein kleines Programm für die Eingabe eines Zeichens, das anschließend mit dem BATCH-Befehl IF ERRORLEVEL weiterverarbeitet wird.

Dieses dazu notwendige Programm EINGABE.COM wollen wir im folgenden erstellen und dabei einige wichtige DEBUG-Befehle kennenlernen.

debug *Return*

Nach Aufruf von DEBUG erscheint der Prompt (hier ein -). Zur Eingabe von Befehlen in Maschinensprache geben Sie ein:

-a *Return*

Anschließend wird Ihnen die Adresse des nächsten freien Speichersegments für die Eingabe angezeigt, und Sie tippen die notwendigen Maschinensprache-Befehle ein (xxxx steht für einen von Aufruf zu Aufruf unterschiedlichen Wert):

```
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
```

-

Mit diesen vier Maschinensprache-Befehlen erreichen Sie, daß bei Aufruf von EINGABE der Cursor am Bildschirm anschließend abwartend unter dem Prompt steht. Tippen Sie ein beliebiges Zeichen ein, und der Prompt erscheint wieder. Das von Ihnen eingegebene Zeichen steht im Register AH. Dieses Register fragt der BATCH-Befehl IF ERRORLEVEL ab (siehe Kapitel 12).

Das Programm muß von Ihnen noch wie folgt weiterbearbeitet werden:

```
-rnx
:0000
9                Return
```

Im Register CX (Befehl r) tragen Sie die Größe (hexadezimal) des von Ihnen eingegebenen Programmes ein. Der alte Registerinhalt wird neben dem Doppelpunkt angezeigt. Der neue Wert läßt sich sehr leicht ermitteln. Die letzte Zahl nach der Eingabe der Maschinensprache-Befehle am linken Rand war **0109**. Sie subtrahieren davon den Wert 100 und erhalten die Programmgröße **9**. Anschließend muß DEBUG noch der Name des Programmes mitgeteilt (Befehl n) und alles gespeichert (Befehl w) werden. Zum Abschluß können Sie DEBUG mit dem Befehl q verlassen.

```
-n eingabe.com      Return
-w                  Return
Schreiben von 0009 Byte
-q                  Return
```

Mit diesem ersten Beispiel haben Sie bereits fünf Befehle kennengelernt. Mit dem damit erstellten Programm können Sie die umfangreiche BATCH-Datei im Kapitel 12 anwenden.

Im folgenden Beispiel wollen wir das Programm erweitern. Zusätzlich soll EINGABE.COM den eingegebenen Wert am Bildschirm anzeigen.

```
debug eingabe.com
```

-

Mit diesem Start von DEBUG wurde EINGABE.COM bereits in den Speicher geladen. Also benötigen Sie den Befehl **n** nicht mehr, da der Programmname schon bekannt ist.

```
-a109
```

Der neue Programmteil wird am Ende des bisherigen angefügt.

```
xxxx:0109  Tab  mov  Tab  ah,02  Return
xxxx:010B  Tab  mov  Tab  dl,a1  Return
xxxx:010D  Tab  int  Tab  21     Return
xxxx:010F                                     Return
```

Der eingegebene Programmteil muß jetzt noch an die richtige Stelle im Programm übertragen werden. Er kommt vor die beiden Befehle für das Programmende (mov ah,4c und int 21).

```
-m cs:105 109 cs:10f
```

Mit diesem Übertragungsbefehl werden die Befehle ab der Speicheradresse **105** bis **109** (Befehle für das Programmende) hinter den neu eingegebenen Teil verschoben, damit wir wieder die richtige Reihenfolge erhalten.

Sehen wir uns das Programm mit dem Befehl **u** an:

```
-u100 113
xxxx:0100      B8070C      MOV      AX,0C07
xxxx:0103      CD21      INT       21
xxxx:0105      B44C      MOV      AH,4C
xxxx:0107      CD21      INT       21
xxxx:0109      B402      MOV      AH,02
```

xxxx:010B	88C2	MOV	DL,AL
xxxx:010D	CD21	INT	21
xxxx:010F	B44C	MOV	AH,4C
xxxx:0111	CD21	INT	21

-

Wie Sie ersehen können, befinden sich die Befehle für das Beenden eines Programmes jetzt an zwei Stellen im Programm. Die Werte hinter den Adressen stellten die »echte« Maschinensprache dar. DEBUG wandelt nach der Eingabe der symbolischen Darstellung diese sofort in Maschinensprache um. Der Befehl

```
INT      21
```

lautet in Maschinensprache

```
CD21
```

Im Rahmen dieses Buches kann Ihnen weder die Maschinensprache noch der symbolische Code vermittelt werden. Sollten Sie sich dafür interessieren, so empfehle ich Ihnen die Anschaffung eines entsprechenden Buches oder den Besuch eines entsprechenden Seminars.

Als nächsten Schritt müssen wir nun alle Befehle ab MOV AH,02 nach vorne an die Adresse 105 verschieben:

```
-m cs:109 113 cs:105
```

```
-u100 113
```

xxxx:0100	B8070C	MOV	AX,0C07
xxxx:0103	CD21	INT	21
xxxx:0105	B402	MOV	AH,02
xxxx:0107	88C2	MOV	DL,AL
xxxx:0109	CD21	INT	21
xxxx:010B	B44C	MOV	AH,4C
xxxx:010D	CD21	INT	21
xxxx:010F	B44C	MOV	AH,4C
xxxx:0111	CD21	INT	21

-

Damit finden Sie jetzt das komplette Programm zwischen den Adressen 0100 und 010F. Zum Abschluß müssen Sie noch die neue Programmgröße in das Register CX eintragen, speichern und DEBUG beenden:

```
-rcx
:0009
f
-w
Schreiben von 000F Byte
-q
```

Sollte DEBUG für das Speichern kein Dateiname bekannt sein, so erhalten Sie bei Eingabe des Befehls **w** die Fehlermeldung:

```
Schreibfehler, keine Zieleinheit
definiert
```

oder

```
(w)rite error, no destination defined
```

Anschließend müssen Sie mit dem Befehl **n** einen Dateinamen definieren und können dann speichern.

Den Dateinamen finden Sie an der Adresse **80** im Speicher Ihres PCs. Ihnen steht für die Anzeige des Speicherinhalts der Befehl **d** zur Verfügung:

```
-d 0
```

Ohne Angabe eines Wertes (hier **0**) beginnt der Befehl an der Adresse 100. Wird er ein weiteres Mal eingegeben, so wird der Teil dargestellt, der auf den bisher angezeigten Inhalt im Speicher folgt. Dies gilt nicht nur für **d**, sondern z.B. auch für **u**.

```
-d
```

Auf den folgenden Seiten finden Sie eine alphabetische Übersicht der einzelnen DEBUG-Befehle mit einer Beschreibung.

DEBUG-Befehle

Befehl **a** (adresse)

Mit diesem Befehl kann anschließend symbolischer Maschinen-code (8088 usw.) bzw. IBM-Personalcomputer-Macroassembler-Code eingegeben werden. Numerische Werte müssen hexadezimal eingegeben werden. Eingabefehler werden sofort angezeigt.

```
^ Error
```

Wird eine **adresse** angegeben, so wird der Code ab dieser in den Speicher geschrieben.

Befehl c 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

adressel Byte1 Byte2 adresse2

Beispiel c0:0 15 cs:10

Vergleichen der Speicheradresse 0000:0000 bis 0000:0004 (= 5 Speicherstellen – **L5**) mit den ab Adresse CS:0010 (CS wird durch die im CS-Register gespeicherte Adresse ersetzt) beginnenden Werten. Eventuelle Unterschiede werden angezeigt.

Beispiel 0000:0001 56 53 56B5:0011**Befehl d** 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 dcs:0100 10f

Ab der Adresse 100 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 .krFFD..8.=M!rF.
d 500 600
```

Der Speicherinhalt zwischen den Adressen 500h und 600h (Bereich im aktuellen Segment) wird angezeigt.

Befehl e 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). Geben Sie einen »-« ein, wird die davorliegende 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 "ABC" 46 C6

Ab der Adresse 100h im aktuellen Segment werden die Werte der nächsten Bytes mit 02h, EBh, F2h, den Buchstaben ABC, 46h und C6h überschrieben (das **h** soll Sie darauf hinweisen, daß es sich bei den Werten um hexadezimale Zahlen handelt).

Befehl f 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. Ist die **liste** zu lang, werden die nicht benötigten Zeichen ignoriert.

Beispiel f 0DDC:0100 1100 00

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

Befehl g g (=adressel) (adresse2...)

Das Programm, das sich im Speicher befindet, wird ausgeführt. Wird die **=adressel** 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.

Keine Stopp-Adressen im Bereich des ROM-BIOS oder ROM-Basic angeben! Falsch angewendet kann dieser Befehl zu Systemabstürzen führen.

Befehl h h wert wert

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

Befehl i i wert

Lesen und Anzeigen des Wertes eines Ein- oder Ausgabekanals (**wert** ist die Kanaladresse). Die Kanaladressen Ihres Systems finden Sie im Technischen Handbuch (soweit der Hersteller ein solches anbietet).

Befehl l 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** vorher ein Name definiert werden. Die Anzahl der gelesenen Bytes wird im Register CX abgelegt (gilt nur beim Einlesen einer Datei). 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 **00** 15 (**0F**) Sektoren geladen.

```
n command.com
l
```

Der Name der zu ladenden Datei wird mit dem Befehl **n** definiert. Mit **l** wird die Datei geladen.

Befehl m 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. Der Inhalt des Speichers an der ursprünglichen Stelle bleibt, soweit keine Überlappung auftritt, unverändert.

Befehl n n dateiname (dateiname...)

Mit diesem Befehl werden Dateinamen definiert.

Beispiel

```
n datei.com
l
n daten1.dat daten2.dat
g
```


Der Name des Programmes DATEI.COM wird angegeben und dieses mit dem Befehl **I** geladen. Anschließend werden zwei Datendatei-Namen angegeben. Mit **g** wird das Programm gestartet. Diese Eingaben entsprechen dem folgenden Befehl auf 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 **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 Kanaladresse 2F8 gesandt. Dieser Befehl ist mit Vorsicht zu behandeln, da sehr leicht die Lauffähigkeit Ihres Computers damit gestört werden kann. Überzeugen Sie sich unbedingt vor der Ausführung dieses Befehls davon, daß die Kanaladresse und das Byte auch richtig sind (Kanaladressen finden Sie im Technischen Handbuch zu Ihrem DOS oder Computer).

Befehl p **p** (=adresse)(wert)

Das geladene Programm wird schrittweise ausgeführt. Dieser Befehl **P**(roceed) ist mit **T**(race) vergleichbar. Mit ihm werden ein Unterprogramm (symbolischer Maschinensprache-Befehl CALL), eine Schleifeninstruktion usw. ausgeführt. **DEBUG** stoppt die Ausführung erst wieder, wenn der Vorgang abgeschlossen ist (nach dem Unterprogramm-Aufruf usw.).

Beispiel xxxx:017A CALL 1000
 xxxx:017D JNZ 2000
 usw.
 xxxx:1000 XOR AX,AX
 usw.
 xxxx:1xxx RET

-p=017a

Der Befehl wird bei CALL gestartet, und das Unterprogramm wird Befehl für Befehl ab der Adresse 1000 ausgeführt. DEBUG stoppt bei der Adresse 017D und zeigt die Registerzustände sowie den nächsten Befehl (JNZ) mit seiner Adresse an.

Befehl q **q**

Das Programm DEBUG beenden.

Befehl r **r** (register)

Anzeigen und/oder Verändern eines oder mehrerer Register des Prozessors (CPU). Bei Eingabe eines einzelnen Registernamens zum Befehl 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
	PC	Program Counter
	F	Statusregister

Kennzeichen für das F-Register:

Überlauf:	OV	Überlauf	NV	kein Überlauf
Richtung:	DN	Vermindern	UP	Erhöhen
Unterbrech./Interrupt:	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 Error

Doppeltes Kennzeichen

Bei einem falschen Kennzeichen:

BF Error

Falsches Kennzeichen

Wenn **DEBUG** gestartet wird, sind die Kennzeichen nicht und die Register auf Null gesetzt (mit Ausnahme von Register **IP**, das den Wert 0100h, und **SP**, das den Wert FFEEh beinhaltet). Die 4 Segmentregister enthalten die Segmentadresse.

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 s bereich liste

Durchsucht den angegebenen Bereich nach der vorgegebenen Liste von Zeichen und zeigt die Adressen, an der diese gefunden wurden.

Beispiel s cs:0100 L800 04 00 72 10 "Buch"

Im Codesegment (CS) werden ab der Adresse 0100h in der Länge von Hexadezimal 800 (= 2048 Zeichen) die aufeinanderfolgen-

den Zeichen **04h/00h/72h/10h** und **Buch** gesucht. Alle Adressen, an welchen diese Zeichenkette gefunden wird (innerhalb der durchsuchten 2048 Zeichen), werden am Bildschirm angezeigt. Findet DEBUG die Zeichen im definierten Bereich nicht, erfolgt keine Anzeige, und es erscheint sofort wieder der Prompt.

Befehl t t (=adresse) (wert)

Der Befehl führt das aktuelle Programm schrittweise aus und zeigt nach jedem Befehl alle Register sowie die nächste auszuführende Funktion (Befehl). 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. Die Zeilen des Bildschirms reichen nicht aus, um alle anzuzeigenden Werte aufzunehmen. Die durchlaufende Anzeige können Sie anhalten und auch wieder fortsetzen, indem Sie die Tasten *Pause* bzw. *Strg+S* betätigen.

```
AX=0000 BX=0000 CX=0009 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0F4E ES=0F4E SS=0F4E CS=0F4E IP=0100 NV UP EI PL NZ NA PO NC
0FAE:0100 B8070C MOV AX,0C07
```

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

Wandelt den Speicherinhalt in assemblerähnliche (symbolische) Befehle um und zeigt diese mit der Speicheradresse und dem Code an.

Beispiel u cs:0100h

Dabei wird z.B. angezeigt:

```
xxxx:0100 E84CFE CALL 0A4F
xxxx:0103 7306 JNB 0C0B
```

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 Disk 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 DOS auf die Diskette/Festplatte speichern (siehe auch Befehl **l**).

Beachten Sie, daß Programme mit dem File-Typ EXE oder HEX nicht mit DEBUG geladen und anschließend verändert werden dürfen. Vorher müssen Sie den Programmnamen ändern!! Der erste Wert eines EXE-Programmes, das Sie mit DEBUG in den Speicher laden, entspricht nicht dem ersten Wert des Programmes. Bei diesem Vorgang wird automatisch ein erster kleiner Teil der Befehle durchgeführt (DEBUG führt dies immer durch, wenn Sie eine Datei mit dem Typ EXE laden).

Versuchen Sie mit den Befehlen **N** und **W** eine Datei mit den Typen EXE oder HEX zu speichern, erhalten Sie die Fehlermeldung

EXE and HEX files cannot be written

DEL oder ERASE *interner Befehl*

Zweck Löschen von Dateien/Programmen. Es können wahlweise die Befehle DEL oder ERASE verwendet werden. Dateien/Programme mit dem Attribut »read-only« (z.B. IBMBIO.COM und IBMDOS.COM) lassen sich mit diesen Befehlen nicht löschen (siehe Befehl ATTRIB). Ebenso Sub-Directory-Namen, die nur mit dem Befehl RMDIR oder RD gelöscht werden können.

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 im 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 *.*

oder

del

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

Sind Sie sicher (J/N)?

Die Frage, ob Sie sicher sind, erscheint immer dann, wenn ein ganzes Directory gelöscht werden soll.

DIR *interner Befehl*

Zweck Anzeigen von Inhaltsverzeichnissen mit den gespeicherten Informationen:

- File-Name
- File-Typ
- Größe der Datei in Byte
- Datum und Uhrzeit
- Laufwerksbezeichnung
- 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 14 »Die Konfiguration von DOS«) angezeigt. Datum und Uhrzeit werden jeweils bei Anlage oder Änderung einer Datei neu gespeichert.

Kommandoaufbau `dir (d:)(weg)(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 Directories/Laufwerks angezeigt.

Diskette/Platte, Laufwerk A:, hat keinen Namen

Verzeichnis von A:\

WSC	COM	25600	1.01.80	12.03
KAPITEL	DOC	84864	10.12.86	2.11
BUCH	<DIR>		26.10.87	2.24
3 Datei(en)			77824 Byte frei	

Die Angabe des freien Speicherplatzes wird auf die nächste durch 1024 (= 1 Kbyte) teilbare Zahl aufgerundet.

```
dir *.sys
```

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

```
dir /w
```

Anzeige des Inhaltsverzeichnis vom aktuellen Directory in »weiter« Form.

Beispiel:

```
Diskette/Platte, Laufwerk A:, hat keinen  
Namen  
Verzeichnis von A:\
```

```
WSC      COM  WSOVLY1  OVR  WSMMSG  OVR  MAILMRGE OVR  
WSTYPE COM 1  
          KAPITEL  BAK  KAPITEL 111  
2  
9 Datei(en)  77312 Byte 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 aufsteigend sortierten Inhaltsverzeichnisses aus Laufwerk C. Ohne > **prn** erfolgt die sortierte Ausgabe auf den Bildschirm bzw. die Standardausgabereinheit.


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

Anzeige der sortierten Dateien aus Laufwerk C. Es wird ab der Spalte 14 aufsteigend sortiert. Wird noch /R hinter /+14 angegeben, erfolgt der Sortiervorgang absteigend. Nach jeder Bildschirmseite stoppt der Ausgabevorgang (**more**):

Weiter -> eine Taste betätigen

Beachten Sie hierzu auch das Kapitel 9.

```
dir datei
```

Es werden alle Dateien mit dem File-Namen DATEI und einem beliebigen File-Typ angezeigt.

```
dir datei.
```

Es werden alle Dateien mit dem File-Namen DATEI und ohne File-Typ angezeigt.

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 <i>oder</i> DIR *.*
DIR ..	Es wird das Inhaltsverzeichnis des hierarchisch vorher liegenden Directorys angezeigt

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

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

Diskette/Platte, Laufwerk C:, hat keinen
Namen

Verzeichnis von C:\GWBASIC

.	<DIR>	31.08.87	12.05
..	<DIR>	31.08.87	12.05
BASICA1	EXE 58720	7.10.87	0.10
FRANCE	BAS 512	9.11.87	0.10
BASICA	EXE 70784	5.11.87	0.14
ADR	BAS 3010	1.01.88	0.14
ADR2	BAS 930	1.01.88	0.15
ADR3	BAS 1094	10.10.87	0.15
WS	47941	10.10.87	0.15
ELITE	BAS 29	8.10.87	8.49
ETI	BAS 643	6.09.87	0.55
WINDOW	BAS 512	1.11.87	0.05
ANWEND	<DIR>	6.10.87	16.03
13 Datei(en)		4931584 Bytes frei	

Diskette/Platte, Laufwerk C:, hat keinen
Namen

Verzeichnis von C:\GWBASIC

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

In der »weiten« Form sind Sub-Directory-Einträge zwar mit ihrem Namen gelistet, aber nicht als solche erkennbar. Doch Sie können dieses Problem mit einem einfachen Trick umgehen, indem Sie alle Sub-Directories zusätzlich mit dem File-Typ benennen.

Beispiele LOTUS.SUB
BASIC.SUB

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 alle Dateien komplett (*.*) übertragen, so sind die beiden Disketten normalerweise nicht identisch. Benutzen Sie in diesem Fall den Vergleichsbefehl COMP. Der passende Kopierbefehl für DISKCOMP ist DISKCOPY.

Der Befehl darf nicht bei einem Netzwerklaufwerk 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

Beispiele diskcomp a: b:

Die Disketten in Laufwerk A und B werden auf Gleichheit überprüft. Die Laufwerksangaben dürfen nicht identisch sein. Bei Ungleichheit der Diskettenformate erhalten Sie eine Fehlermeldung nach der Prüfung.

Erste Diskette in Laufwerk A: einlegen
 Zweite Diskette in Laufwerk B: einlegen
 Anschließend eine Taste betätigen ...

Nach Betätigung einer beliebigen Taste erscheint:

Verglichen werden 40 Spuren
 9 Sektoren/Spur, 2 Seite(n)

Vergleich OK

Weitere Disketten vergleichen (J/N)?

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

Vergleichsfehler auf Seite 0, Spur 0

Diese Meldung wird Ihnen für jede unterschiedliche Seite/Spur angezeigt.

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 Fall während der Programmausführung aufgefordert, die erste bzw. die zweite Diskette abwechselnd einzulegen (Original in Speicher einlesen – Diskettenwechsel – Speicherinhalt mit Kopie vergleichen – Diskettenwechsel ...). Es wird hier immer mit dem aktuellen Laufwerk gearbeitet (am Prompt ersichtlich).

Erste Diskette in Laufwerk A: einlegen
Anschließend eine Taste betätigen . . .
Verglichen werden 80 Spuren
15 Sektoren/Spur, 2 Seite(n)

Wenn der Inhalt der Diskette gelesen wurde, erscheint:

Zweite Diskette in Laufwerk A: einlegen
Anschließend eine Taste betätigen . . .
Vergleich OK
Weitere Disketten vergleichen (J/N)?

Geben Sie nur eine Laufwerksbezeichnung ein, so bezeichnet diese das Quellenlaufwerk (Original) und vergleicht es mit der Diskette im aktuellen Laufwerk (siehe Prompt am Bildschirm). Wird ein unbekannter Parameter oder Bezeichnung hinter dem Befehl angegeben, erhalten Sie die Fehlermeldung:

Ungültiger Parameter
Keine Dateinamen angeben
Befehlsformat: DISKCOMP d: d: [/1]/[8]

Verwenden Sie eine unbekannte Laufwerksbezeichnung oder geben Sie ein Festplattenlaufwerk an, erhalten Sie die Fehlermeldung:

Ungültige Laufwerksangabe
Anggegebenes Laufwerk existiert nicht
oder ist kein Diskettenlaufwerk.

Besitzen die beiden zu vergleichenden Disketten kein identisches Aufzeichnungsformat (z.B. 360-Kbyte-Diskette mit einer 1,2-Mbyte-Diskette vergleichen), erscheint am Bildschirm:

```
Disketten-/Laufwerksart nicht  
kompatibel  
Vergleich beendet
```

Die Meldung erscheint auch, wenn die beiden Diskettenlaufwerke ein unterschiedliches Format besitzen (3½" und 5¼" usw.).

DISKCOPY *externer Befehl*

Zweck Der komplette Inhalt einer Diskette wird auf eine andere übertragen (Erstellung eines Diskettenduplikates). Vor dem Kopiervorgang wird die neue Diskette formatiert, falls dies noch nicht erfolgt ist.

Sollte die Zieldiskette bereits Daten enthalten, so werden diese durch DISKCOPY unwiederbringlich gelöscht.

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

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

Der Befehl darf nicht bei einem Netzwerklaufwerk verwendet werden oder solange der Befehl JOIN aktiv ist.

Kommandoaufbau `diskcopy (s: (d:)) (/x)`
 /x /l 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 besitzen, kann der Kopiervorgang trotzdem erfolgen. Sie werden in diesem Fall während der Programmausführung aufgefordert, die Quellen- bzw. die Zieldiskette einzulegen (Original in Speicher einlesen – Diskettenwechsel – Speicherinhalt auf Zieldiskette schreiben...). Es wird hier immer mit dem aktuellen Laufwerk gearbeitet (am Prompt ersichtlich).

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

Quellendiskette in Laufwerk A: einlegen

Anschließend eine Taste betätigen ...

Kopiert werden 80 Spuren

9 Sektoren/Spur, 2 Seite(n)

Zieldiskette in Laufwerk A: einlegen

Anschließend eine Taste betätigen ...

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

Zieldiskette ist schreibgeschützt!
Anschließend eine Taste betätigen ...

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

Bemerkungen

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

Sie können mit DISKCOPY nur arbeiten, soweit Sie nur mit einem oder zwei gleichen Laufwerkstypen arbeiten (siehe auch DISKCOMP).

Da nach längerem Arbeiten mit einer Diskette sehr häufig die Dateien nicht mehr zusammenhängend, sondern verstreut gespeichert sind, ist es empfehlenswert, den Befehl XCOPY zu verwenden. Damit werden die Dateien zusammenhängend auf der neuen Diskette gespeichert (Geschwindigkeitsvorteil).

ECHO *interner Befehl*

Zweck ECHO kontrolliert die Ausgabe von Befehls- und Kommentarzeilen während der Ausführung einer BATCH-Datei (siehe Kapitel 12) oder 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 Directorys in Laufwerk A wird angezeigt

Bemerkung Soll auch der Befehl ECHO OFF am Bildschirm nicht angezeigt werden, so geben Sie vorher das Zeichen @ ein:

```
@echo off
```

EDLIN *externer Befehl**Allgemeines und Zweck*

EDLIN ist das Dienstprogramm zur Herstellung und Änderung von Textdateien (und damit natürlich auch von Quellenprogrammen). 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 des Speichers Ihres Computers genutzt werden. Davon werden maximal ca. 75 % 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 (d:)(weg)dateiname (/x)

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

Bemerkungen

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

Eingabefehler

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

Dateiname muß angegeben werden

Besitzt der angegebene Dateiname das Attribut »Nur-Lesen«, so erhalten Sie die Fehlermeldung:

Datei hat Status Nur-Lesen

Sie können dieses Attribut mit dem Befehl ATTRIB ändern.

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

Taste	Funktion	Beschreibung
F1 oder ->	kopiert ein Zeichen	kopiert ein Zeichen aus dem Eingabespeicher in die neue Zeile
F2x	kopiert bis Zeichen x	kopiert alle Zeichen bis zum angegebenen Zeichen x aus dem Eingabespeicher in die neue Zeile
F3	alle Zeichen kopieren	kopiert alle Zeichen des Eingabespeichers in die neue Zeile
F4x	überspringen bis Zeichen x	ignoriert alle Zeichen aus dem Eingabespeicher bis zum Zeichen x
F5	neuer Eingabespeicher	schreibt eine neu eingegebene Zeile in den Eingabespeicher
F6	Eingabeende	erzeugt ein <i>Strg-Z</i> -Zeichen (damit wird das Zeichen für Dateiende erzeugt)
F7	Nullzeichen	das Zeichen wird mit dem Wert 00 hexadezimal gespeichert
BackSpace oder <-	ein Zeichen löschen	ein Zeichen im Speicher und in der Eingabezeile löschen
Ins	einfügen	umschalten auf Zeichen einfügen
Entf	löschen	ein Zeichen im Eingabespeicher löschen (nicht jedoch in der sichtbaren Eingabezeile)
Esc	Eingabe löschen	löscht die gesamte Eingabezeile, nicht jedoch den Eingabespeicher

Befehlsübersicht
zu **EDLIN**

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

Befehl (#)a Zeilen anfügen

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 Zeilen kopieren

- 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 soll(en)
 c = Kopierbefehl
- Befehl** (#a)(, #e)d Zeilen löschen
- Zweck** Es werden eine oder mehrere Zeilen gelöscht. Wird nur **d** angegeben, so wird nur die aktuelle Zeile gelöscht.
- Befehl** (# Zeilen ändern/bearbeiten
- 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 Text speichern und Ende
- Zweck** Das Programm beenden und den Text speichern (der alte Text bleibt erhalten und erhält den File-Typ **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/Platte voll -> Änderungen nicht gespeichert
- Befehl** (#)i Zeilen einfügen
- Zweck** Es werden eine oder mehrere neue Zeile(n) vor die angegebene eingefügt. Der »Einfügemodus« bleibt so lange bestehen, bis *Strg+C* oder *Strg+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 Textende bzw. ab der aktuellen Zeile eingefügt.
- Befehl** (#a)(, #e)l Zeilen anzeigen
- Zweck** Anzeigen von Zeilen (nur die im Speicher vorhandenen können angezeigt werden) am Bildschirm. Mit der Tastenkombination *Strg+S* kann gestoppt oder weiter angezeigt werden. Die aktuelle Zeile ist mit einem Stern (*) gekennzeichnet.

- Befehl** (#a),(#e),#m Zeilen verschieben
- 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 Seitenweise Zeilen listen
- Zweck** Zeilen werden seitenweise aufgelistet. Nach Auflistung der Zeilen auf dem Bildschirm wird die letzte angezeigte Zeile zur aktuellen (im Gegensatz zum Befehl I).
- Befehl** q EDLIN verlassen
- 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)(?)R**zeichenkette1**F**zeichenkette2** Text ersetzen
- Zweck** In den bezeichneten Zeilen wird die **zeichenkette1** durch die **zeichenkette2** ersetzt. Wird das ? mit angegeben, so wird vor dem Austauschen des Textes die mit der **zeichenkette1** gefundene Zeile in geänderter Form angezeigt, und Sie werden gefragt, ob Sie austauschen wollen.
- Befehl** (#a),(#e)(?)S**zeichenkette** Text suchen
- Zweck** Suchen einer **zeichenkette** in den bezeichneten Zeilen. Wird ? mit angegeben, so wird die gefundene Zeile angezeigt, und Sie werden gefragt, 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** (#)T**dateiname** Text übertragen/lesen
- Zweck** Kopiert eine Textdatei von Diskette/Festplatte 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 Zeilen speichern

Zweck Schreibt die angegebene Zeilenzahl auf die Diskette/Festplatte zurück und löscht diese anschließend 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 Beginn einer Zeile erkannt werden). Ebenso kann für die Nummer der Zeile nach der letzten im Text vorhandenen das Zeichen »#« verwendet werden.

Mit EDLIN können auch Steuerzeichen eingegeben werden (z.B. für den Druck des Textes). Dabei wird die *Strg*-Taste als *Strg*+*V* eingegeben. Darauf folgt das entsprechende Steuerzeichen als Großbuchstabe. Die meisten Matrixdrucker können wie folgt auf Schmalschrift bzw. auf Normalschrift geschaltet werden:

Strg+*V*+*O* = Schmalschrift

Strg+*V*+*R* = Normalschrift

Strg+*V*+*L* = Seitenwechsel

Mehrere Befehle können auch nacheinander eingegeben werden.

Beispiel 10;-5,+51;e

Die Zeile 10 steht anschließend zum Verändern bereit. Nach *Return* werden die 5 Zeilen vor und nach der aktuellen Zeile (Zeile 10) angezeigt. Anschließend wird der Text gespeichert und EDLIN beendet.

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 bereits existiert, erhalten 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 5¼ 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 zweimal 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.

1,41l

Es werden alle Zeilen bis zur 41. ab der ersten am Bildschirm angezeigt (es kann mit *Strg+S* gestoppt werden – weiter mit *Strg+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 gefragt, 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 anschließend mit dem Befehl **PRINT** drucken.

EXE2BIN *externer Befehl*

Zweck EXE2BIN formt Programme mit dem Format **EXE** (File-Typ) in das Format **COM** oder **BIN** 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 (d:)(weg)dateiname1 (d:)(weg)(dateiname2)
dateiname1 Name des EXE-Programmes
dateiname2 Name des COM-Programmes
Geben Sie keinen File-Typ an, so wird automatisch **BIN** benutzt – wenn Sie **COM** wünschen, muß dies angegeben werden

Beispiele exe2bin programm a:programm.com

Das Programm **PROGRAMM.EXE** wird in **PROGRAMM.COM** umgewandelt und auf der Diskette im Laufwerk **A** gespeichert.

exe2bin programm programm

Das Programm **PROGRAMM.EXE** wird in **PROGRAMM.BIN** umgewandelt (aktuelles Laufwerk/Directory).

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, erstellt mit einem **LINKer**, vorliegen. Der residente und Datenteil des Programmes müssen kleiner als 64 Kbyte sein. Es darf kein sogenanntes **STACK-Segment** enthalten.
Wenn die Fehlermeldung

Datei kann nicht umgewandelt werden

bzw.

File cannot be converted

erscheint, liegt das Ursprungsprogramm nicht im gültigen Format vor.

Für weitere Angaben beachten Sie bitte die Beschreibungen in den DOS-Handbüchern zu Ihrem Computer. Für die Arbeit mit diesem Programm sind fundierte System- und Programmierkenntnisse erforderlich.

EXIT *interner Befehl*

Zweck Dieses Kommando ermöglicht das Verlassen des COMMAND.COM-Prozessors und die Rückkehr zu einem vorherigen Level (z.B. Applikation/Programm), soweit einer existiert. Beachten Sie vor allem auch die Angaben zum Befehl COMMAND in diesem Kapitel.

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 DOS-Prompt (z.B. C>) am Bildschirm, und Sie können beliebige DOS-Befehle, wie z.B.

dir

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

exit

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

FASTOPEN *externer Befehl*

Zweck Bei jedem Zugriff auf eine Datei sucht DOS alle Verzeichnisse ab, die sich im Verzeichnispfad zu der Datei befinden. Bei komplexen Verzeichnisstrukturen kann dies einige Zeit dauern. FASTOPEN speichert die Zugriffspfade von Verzeichnissen und den zuletzt eröffneten Dateien. Bei Zugriff auf eine Datei durchsucht DOS erst die von FASTOPEN im Speicher erstellte Liste. Ist die Datei dort enthalten, wird diese sehr schnell gefunden. Im anderen Fall erfolgt die Suche von DOS wie gewohnt.

Nicht für Netzwerklaufwerke und Laufwerke, die mit ASSIGN, JOIN oder SUBST definiert wurden, verwenden. FASTOPEN kann nur für Festplattenlaufwerke benutzt werden.

Kommandoaufruf (d:)(weg)fastopen d:(= #) (d:(= #)) (...)

d: Laufwerke, für die Zugriffspfade von FASTOPEN gespeichert werden sollen

Maximale Anzahl von Verzeichnissen bzw. Dateien, deren Zugriffspfade für das angegebene Laufwerk gespeichert werden sollen. Ohne Angabe für # gilt als Standardwert 34. Werte zwischen 10 und 999 sind möglich (999 als Summe aller Angaben). Mindestens ein Wert ist anzugeben, welcher der Anzahl der hierarchischen Ebenen auf Ihrem Laufwerk entspricht.

Für jeden gespeicherten Zugriffspfad werden 35 Byte Speicher benötigt.

Beispiele fastopen c: d:

Für die Festplattenlaufwerke C und D wurden je 34 Zugriffswege definiert.

fastopen c:=100

Für das Festplattenlaufwerk C wurde der Wert 100 definiert.

Bemerkungen Überschreitet die Anzahl der Dateizugriffe auf ein Laufwerk den mit FASTOPEN definierten Wert, so wird der am längsten nicht mehr benutzte Zugriffspfad gelöscht und durch den neuesten ersetzt.

FASTOPEN wird am sinnvollsten in die Datei AUTOEXEC.BAT eingebunden (Kapitel 12).

Bei Angabe eines zu großen Wertes kann es vorkommen, daß der Dateizugriff länger dauert als ohne FASTOPEN (z.B. bei 999).

FDISK *externer Befehl*

Zweck Verfügt Ihr System über eine Festplatte, so muß diese für DOS als erstes vorbereitet werden. Erst nach der Bearbeitung mit FDISK und FORMAT kann diese von DOS verwendet werden. Mit DOS kann ein Laufwerk maximal 32 Mbyte besitzen. Sollte Ihre Festplatte eine größere Kapazität aufweisen, so kann diese mit FDISK in mehrere Laufwerke eingeteilt werden.

Kommandoaufbau (d:)(weg)fdisk

Beispiel Geben Sie

fdisk

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

```
IBM Personal Computer
Festplatten-Einrichtungsprogramm 3.30
(C)Copyright IBM Corp. 1983,1987
FDISK-Menü
Aktuelles Festplattenlaufwerk: 1
```

Eine Auswahl angeben:

- 1 Erstellen einer DOS-Partition
2. Ändern der aktiven Partition
3. Löschen der DOS-Partition
4. Anzeigen der Partitionsdaten
5. Nächste Festplatte auswählen

Auswahl: [1]

Esc --> Zurück zu DOS

Geben Sie die Ziffer 1 ein, und betätigen Sie *Return*. Wenn die Festplatte noch nicht für DOS vorbereitet war, so werden Sie gefragt, ob die gesamte Kapazität für das Betriebssystem eingeteilt werden soll. Sie haben die Möglichkeit, die Festplatte in mehrere Bereiche einzuteilen. Dabei können alle Bereich für DOS oder aber die restlichen Bereiche auch für andere Betriebssysteme (CP/M-86, Xenix, usw.) verwendet werden.

Der Punkt 5 in der Auswahl erscheint bei Ihnen nur, wenn mehr als eine Festplatte in Ihrem System eingebaut sind.

Sollte bereits eine DOS-Partition vorhanden sein, so muß diese erst gelöscht werden, bevor die Festplatte neu eingeteilt werden kann. Selbstverständlich ist es möglich, auch mehr als eine DOS- oder andere Partition anzulegen.

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

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

Beispiel

Ändern der aktiven Partition

Aktuelles Festplattenlaufwerk: 1

Partition	Status	Art	Start	Ende	Größe
C: 1	A	PRIDOS	0	609	610

Gesamtplattenbereich ist 610 Zylinder.

Die gegenwärtig aktive Partition ist 1.

Geben Sie die zu aktivierende

Partition ein : []

Esc --> Zum FDISK-Menü

Bemerkungen

Es werden Ihnen alle Partitionen angezeigt, und Sie können diejenige auswählen, die Sie aktiv setzen wollen (es kann immer nur eine aktiv sein!). Zu einem Betriebssystem können mehrere Partitionen auf einer oder mehreren Festplatten gehören. DOS 3.30 kennt zwei Arten von Partitionen, die primäre (max. 32 Mbyte) und die erweiterte. Die erweiterte kann beliebig groß sein und in mehrere Bereiche (logische Laufwerke) aufgeteilt werden. Jedes logische Laufwerk kann max. 32 Mbyte groß sein. Zum Beispiel weist DOS der primären Partition den Laufwerksbuchstaben C und den Bereichen in der erweiterten Partition die Bezeichnungen ab D (bis maximal Z) zu.

Sie können eine DOS-Partition auch löschen. Wählen Sie Punkt 3 im Menü. Daraufhin wird Ihnen eine Auswahl angezeigt. Sie haben die folgenden Löschmöglichkeiten:

1. Primäre DOS-Partition
2. Erweiterte DOS-Partition
3. Logische Laufwerke in der erweiterten DOS-Partition

Wenn Sie die aktive DOS-Partition löschen, kann nicht mehr von der Festplatte gestartet werden, und alle gespeicherten Daten sind unwiederbringlich verloren.

Wenn Sie eine erweiterte DOS-Partition löschen wollen, so müssen erst alle logischen DOS-Laufwerke gelöscht werden.

Wählen Sie im Menü den Punkt **4** an, 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. Die Installation von DOS erfolgt am besten mit SELECT (einschließlich Formatierung – siehe Kapitel 6).

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" ((d:)(weg)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 DOS-Einführung wird auf Personalcomputer-Systemen durchgeführt. Dabei gelangen Computer der Baureihe VERSUCH PC zum Einsatz.

Die Computer der Baureihe VERSUCH PC sind vollkompatible Modelle.

Dateiname KAP2

Der Hersteller bietet zum Marktstandard kompatible Computer (VERSUCH PC) an.

Dateiname KAP3

Für Personalcomputer der kompatiblen Klasse stehen am Weltmarkt eine sehr große Anzahl von Programmen zur Verfügung.

Der Hersteller bietet seinen Kunden hier ein System an (VERSUCH PC).

Beispiele find /n "VERSUCH PC" kap1 kap2 kap3

Es werden die Zeilen mit der Zeichenkette "**VERSUCH 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). Am Bildschirm erfolgt folgende Anzeige:

----- kap1

[3]BAUREIHE VERSUCH PC ZUM EINSATZ.

[4]DIE COMPUTER DER BAUREIHE VERSUCH PC SIND VOLL
KOMPATIBLE

----- kap2

[2](VERSUCH PC) AN.

----- kap3

```
dir | find /n "DAT | sort > prn
```

Es sollen nur Dateien angezeigt werden, die im Namen **nicht** die Zeichenkette **DAT** 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

```
dir | find "<DIR>"
```

Es werden alle Sub-Directory-Einträge aus dem Inhaltsverzeichnis am Bildschirm (Standardausgabeeinheit) angezeigt.

```
find /n /v "VERSUCH PC" kap1 kap2 kap3
```

Es werden die Zeilen mit der Zeichenkette **"VERSUCH 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 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 (VERSUCH

[4]PC).

FOR *interner Befehl*

Zweck Wiederholt mehrmals einen DOS-Befehl.

Kommandoaufbau **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 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 config.sys
```

...

```
A>type autoexec.bat
```

...

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. Hinter DO darf der Befehl FOR nicht verwendet werden.

FORMAT *externer Befehl*

Zweck FORMAT bereitet eine Diskette oder Festplatte zur Aufnahme von DOS (bzw. Dateien) vor. Das Aufzeichnungsformat wird initialisiert und das Medium auf schadhafte Spuren, die für DOS gesperrt werden, untersucht. Es wird das Inhaltsverzeichnis initialisiert, die sogenannte »File-Allocation-Table«, die Dateizuordnungstabelle und der »Boot-Loader« (Systemlader) angelegt. Disketten werden vom Hersteller immer unformatiert zur Verfügung gestellt, d.h., der Computer kann diese Disketten nicht verarbeiten, solange sie nicht formatiert sind.

FORMAT zerstört alle eventuell vorhandenen Daten unwiederbringlich auf einer Diskette oder Festplatte!!

Nach dem Formatieren kann einer Diskette oder Festplatte ein Name zugeordnet werden (ein sogenanntes Label). Dazu ist der Parameter /V erforderlich. Das Label kann auch nachträglich mit dem Befehl LABEL bearbeitet werden.

Beachten Sie zu diesem Befehl auch die Kapitel 6 und 7 – hier werden die Aufteilung und das Inhaltsverzeichnis näher beschrieben. Den Befehl **nicht** bei einem Netzwerklaufwerk verwenden. FORMAT darf nicht für Laufwerke verwendet werden, die durch JOIN oder SUBST zugeordnet worden sind. Eine RAM-Disk (Kapitel 14 – VDISK.SYS) darf ebenfalls nicht formatiert werden (dies ist natürlich auch nicht erforderlich).

Kommandoaufbau

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

/x... /b Erstellt Freiraum für das DOS-Betriebssystem (IBMBIO.COM und IBMDOS.COM) auf einer Diskette (es wird automatisch mit 8 Sektoren pro Spur formatiert), mit dem Befehl SYS kann später das System jeder DOS-Version übertragen werden

/8 Nur mit 8 Sektoren pro Spur formatieren (normal 9 oder 15), wobei zwar 9 oder 15 Sektoren erstellt, jedoch nur 8 benutzt werden (nur bei 5¼"-Disketten anwendbar)

/1 Nur einseitig formatieren statt normalerweise doppelseitig (nur bei 5¼"-Laufwerken)

/s Nach dem Formatieren wird automatisch DOS kopiert (IBMBIO.COM, IBMDOS.COM und COMMAND.COM) – damit kann der Computer von der neuen Disk gestartet werden (bootbar). Die-

sen Parameter immer beim Formatieren einer Festplatte angeben, soweit dies nicht mit SELECT durchgeführt wird.

/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). Der Name dient Ihnen zur Identifizierung.

/4 Zum Formatieren einer Diskette mit 160/180 Kbyte (/8 bzw. /1 notwendig) oder 320/360 Kbyte (/8 bei 320 Kbyte zusätzlich notwendig) in einem 1,2-Mbyte-Laufwerk (AT). In einem anderen Laufwerk mit einer so formatierten Diskette treten häufig Schwierigkeiten auf.

/N: # Mit diesem Parameter kann die Anzahl der Sektoren pro Spur angegeben werden, mit der formatiert werden soll.

/T: # Damit wird angegeben, wie viele Spuren formatiert werden sollen.

Mögliche Parameter für unterstützte Diskettenarten:

160/180 Kbyte	/S,	/V,	/1,	/8,	/B,	/4
320/360 Kbyte	/S,	/V,	/1,	/8,	/B,	/4
1,2 Mbyte	/S,	/V,	/N,	/T		
720 Kbyte/1,44 Mbyte	/S,	/V,	/N,	/T		
Festplatten	/S,	/V				

Wird kein Parameter angegeben, so wird die Diskette entsprechend den Werten des Laufwerks formatiert.

Wird die Größe einer Festplattenpartition verändert, so muß diese anschließend neu formatiert werden.

Die Parameter /S und /V können nicht zusammen mit /B, /N oder /T verwendet werden.

Die Parameter /N und /T müssen immer zusammen verwendet werden.

Beispiele format

Sie erhalten die Fehlermeldung:

Laufwerksbuchstabe muß angegeben werden

format a:

Damit wird das Programm gestartet, und Sie erhalten folgende Anzeige:

Neue Diskette in Laufwerk A: einlegen,
anschließend die Eingabetaste betätigen

Wenn die neue Diskette eingelegt wurde, betätigen Sie *Return*.
Weiter wird angezeigt:

Kopf: 0 Zylinder: 1

Nach Beendigung der Formatierung erscheint:

Formatieren beendet

1457664 Byte Gesamtspeicherbereich

1457664 Byte auf Diskette/Platte
verfügbar

Weitere Dskt./Platte formatieren (J/N)?

format a: /s

Die Diskette in Laufwerk A wird formatiert und DOS übertragen
(IBMBIO.COM, IBMDOS.COM und COMMAND.COM). Werden die DOS-Dateien im aktuellen Laufwerk nicht gefunden, so erhalten Sie die Meldung:

DOS-Diskette in Laufwerk A: einlegen

Zum Abschluß erscheint:

Systemdateien übertragen

format a: /s /v

Wie vorheriges Beispiel, nur wird hier nach dem Formatierungsvorgang noch nach dem Namen für die Diskette gefragt:

Name (max. 11 Zeichen) oder Eingabetaste

Betätigen Sie nur *Return*, ohne etwas einzugeben, so erhält die Diskette keinen Namen.

format c: /s /v

Die Festplatte C formatieren, DOS (IBMBIO.COM, IBMDOS.COM und COMMAND.COM) übertragen und die Eingabe eines Namens ermöglichen.

Formatieren Sie eine bereits benutzte Disk neu, erhalten Sie die Meldung:

Momentanen Namen der Diskette/Platte,
Laufwerk C:, eingeben

Geben Sie zum Formatieren einer Festplatte die Parameter /1, /8, /4 oder /B an, so erhalten Sie die Fehlermeldung:

Parameter mit Festplatte nicht
kompatibel
Fehler beim Formatieren

Dabei wird der Formatiervorgang nicht gestartet.
Stellt das Programm FORMAT fehlerhafte Sektoren fest, so erfolgt die Kapazitätsanzeige (mit /S) z.B. wie folgt:

362496 Byte Gesamtspeicherbereich
69632 Byte vom System verwendet
1024 Byte in fehlerhaften Sektoren
291840 Byte auf Diskette/Platte
verfügbar

Bei Angabe von nicht vorhandenen Laufwerken oder unbekannten Parametern erhalten Sie selbstverständlich eine Fehlermeldung.

format a: /n:9 /t:80

Eine 3½"-Diskette wird mit 720 Kbyte in einem 1,44-Mbyte-Laufwerk formatiert.

Bemerkungen

Ohne weitere Angaben wird eine Diskette immer im Standardformat des Laufwerks eingerichtet (1,44 Mbyte, 1,2 Mbyte, 720 Kbyte oder 360 Kbyte). Diese Einstellung kann durch die bekannten Parameter verändert werden. Wenn ein Laufwerkstyp bzw. Controller dies nicht zuläßt, erhalten Sie eine Fehlermeldung. 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 Byte in fehlerhaften Sektoren

FORMAT ignoriert Laufwerkszuweisungen mit ASSIGN.

FORMAT beendet mit folgenden Ausgangscodes

(beachten Sie den Befehl IF ERRORLEVEL):

- 0 fehlerlos formatiert
- 3 durch Sie abgebrochen (*Strg+Untbr* oder *Strg+C*)
- 4 gravierender Fehler – FORMAT bricht ab
- 5 Formatierung der Festplatte durch Sie abgebrochen
(Antwort »N«)

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).

Kommandoaufbau goto (:)*marke*

Beispiele 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
```

```
Die Datei command.com ist vorhanden
```

```
goto :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
```

```
Die Datei nicht.com ist nicht vorhanden
```

Bemerkungen Beachten Sie zu vorstehendem Beispiel auch die Befehle IF und ECHO (siehe auch Kapitel 12 »Die BATCH-Befehle und ihre Anwendung«).

Wird eine bei GOTO angegebene Marke nicht gefunden, so erhalten Sie die Fehlermeldung:

Name nicht gefunden

GRAFTABL *externer Befehl*

Zweck Lädt eine zusätzliche Tabelle von ASCII-Zeichen (Werte 128 bis 255) in den Speicher für die Benutzung im Grafik-Mode (Farbgrafikkarte muß im System installiert sein). Eine Zeichentabelle finden Sie im Anhang IV.

Kommandoaufbau (d:)(weg)graftabl(/x)

/x	437	USA, Grafikzeichensatztablette (Standardwert)
	850	Mehrsprachige Grafikzeichensatztablette
	860	Portugal, Grafikzeichensatztablette
	863	Kanada, (frz.) Grafikzeichensatztablette
	865	Skandinavien, Grafikzeichensatztablette
	/STATUS	Anzeige der ausgewählten Zeichensatztablette
	?	Anzeige der möglichen Parameter

Bemerkungen Dieser Befehl ist vor allem notwendig, wenn Sie einen anderen als den Originalzeichensatz (für USA) verwenden (z.B. für den deutschen KEYB GR-Treiber). 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. Mit GRAFTABL kann jederzeit der geladene Zeichensatz verändert werden.

GRAFTABL beendet mit folgenden Ausgangscodes (beachten Sie den Befehl IF ERRORLEVEL):

- 0 Fehlerfrei geladen, bisher keine Zeichensatztablette im Speicher (1360 Byte benötigt die Tabelle im Speicher).
- 1 Die neue Tabelle ersetzt an derselben Stelle im Speicher die bisher geladene.
- 2 Es existiert keine Tabelle im Speicher, und es wird keine neue geladen.
- 3 Falscher Parameter; der Befehl wird nicht ausgeführt.
- 4 Falsche DOS-Version; DOS 3.30 erforderlich.

GRAPHICS *externer Befehl*

Zweck GRAPHICS dient zur richtigen Ausgabe von Grafik und Sonderzeichen des Bildschirms auf den Drucker (Hardcopy-Funktion mit *Druck*). Eine Grafikkarte muß in Ihren Computer eingebaut sein.

Kommandoaufbau (d:)(weg)graphics (drucker-typ) (/x)

drucker-typ	color1	Farbdrucker mit schwarzem Farbband
	color4	Farbdrucker mit vierfarbigem Farbband
	color8	Farbdrucker mit CMY-vierfarbigem Farbband
	compact	Compact-Drucker
	graphics	Grafikdrucker (Standarddrucker)
	thermal	IBM-PC-AP-Drucker (Thermodrucker)

/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

/LCD Zum Drucken der Bildschirmanzeige beim IBM-PC-AP-LCD-Bildschirm

Wenn Sie mit einer Monitorauslösung von 640x200 arbeiten, wird der Bildschirmausdruck um 90 Grad gedreht (bei Grafik) – dies geschieht automatisch, wenn Sie im EGA-Mode arbeiten. Wenn Sie ein Assemblerprogramm erstellen, können Sie mit folgenden Befehlen den Ausdruck des Bildschirminhalts erreichen:

```
push bp
int 5
pop bp
```

Diese Befehle können Sie sehr einfach auch mit DEBUG eingeben:

```
debug
-e100
xxxx:0100 xx.55 xx.cd xx.05 xx.5d    Return
-g
```

Zwischen jeder eingegebenen Zahl betätigen Sie die Leertaste. Nachdem Sie **g** eingegeben haben, wird der Bildschirm 1:1 am Drucker abgebildet. Anschließend stoppt Ihr Computer normalerweise, und Sie müssen ihn neu starten, da die Befehle für das Programmende fehlen.

Die obigen Eingaben in der bisher bekannten Eingabeform:

```
debug
-a100
xxxx:0100      Tab      push Tab      bp      Return
xxxx:0101      Tab      int  Tab      5      Return
xxxx:0103      Tab      pop  Tab      bp      Return
xxxx:0104
-g
```

Die obigen Befehle können Sie versuchsweise auch in dem bei DEBUG beschriebenen Beispiel einsetzen und testen.

IF *interner Befehl*

Zweck IF macht eine Befehlsausführung von der Erfüllung einer Bedingung abhängig. Dieser Befehl wird in BATCH-Dateien (siehe Kapitel 12) verwendet.

Kommandoaufbau if (not) bedingung befehl
bedingung ERRORLEVEL nummer
 zeichenkettel == zeichenkette2
 exist (d:)(weg)dateiname

Beispiele Erstellt wird folgende BATCH-Datei (Befehl: **copy con such.bat**):

```
@echo off
chkdsk %1:
if errorlevel 1 echo Fehler Festplatte %1:
```

Aufruf der BATCH-Datei erfolgt z.B. mit

```
such c
```

Wenn der Ausgangscode des Programmes **CHKDSK** 1 oder größer ist, wird folgendes am Bildschirm angezeigt:

Fehler Festplatte C:

Wenn der Ausgangscode von **CHKDSK** kleiner als 1 ist, wird am Bildschirm nichts angezeigt (ausgenommen den Anzeigen des Programmes **CHKDSK**).

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
```

aufgerufen wird, erscheint keine Anzeige am Bildschirm.

Wenn die BATCH-Datei mit

loesche da.bat

aufgerufen wird, erscheint folgende Anzeige am Bildschirm:

Die Datei da.bat wurde gelöscht.

Bemerkungen

Die ERRORLEVEL-Nummern (Beendigungscode) finden Sie im DOS-Handbuch zu Ihrem Computer oder bei einzelnen Befehlen in diesem Buch. Nicht alle Programme erzeugen einen Beendigungscode.

Wichtig ist bei der Benutzung von ERRORLEVEL, daß die IF-Abfrage immer gültig ist, wenn der Ausgangscode gleich **oder** größer als der von Ihnen angegebene Wert ist.

JOIN *externer Befehl*

Zweck Temporär eine Laufwerksbezeichnung mit einem Sub-Directory vertauschen (wenn ein definiertes Sub-Directory angesprochen wird, verwendet das System die Daten im Laufwerk). Der Befehl darf **nicht** mit Netzwerklaufwerken verwendet werden. Die verwendete Laufwerksbezeichnung kann **nicht** verwendet werden, solange JOIN wirksam ist. Wenn JOIN wirksam ist, dürfen folgende Befehle nicht verwendet werden:

BACKUP
DISKCOMP
DISKCOPY
FORMAT
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:\drivec

Der Laufwerksbezeichnung A wurde der Sub-Directory-Name **drivec** 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 \drivec

Damit wird das Inhaltsverzeichnis von Laufwerk A angezeigt.

join

Anzeigen der umgeleiteten Laufwerksnamen:

A: => C:\DRIVEC
join a: /d

Der ursprüngliche JOIN-Befehl wird zurückgesetzt und damit der Originalzustand wiederhergestellt. Das Laufwerk A wird wieder

mit **a:** angesprochen. Der Sub-Directory-Eintrag **drivec** bleibt bestehen, wird aber wie jeder andere behandelt.

Bemerkungen

Wenn Sie die Fehlermeldung für nicht ausreichenden Speicher erhalten, verringern Sie die Anzahl der BUFFER in CONFIG.SYS und starten Ihren Computer erneut (siehe dazu auch das Kapitel 14). 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).

KEYB *externer Befehl*

Zweck Laden einer anderen Tastaturbelegung (für verschiedene Länder/Sprachen) – nach dem Systemstart ist die Tastatur mit Standard-ASCII (USA) belegt.

Die Tastaturanpassungs-Programme vorheriger Versionen dürfen nicht verwendet werden. Beachten Sie dazu auch Kapitel 13.

Kommandoaufbau (d:)(weg)keyb (xx(,(yyy),((d:)(weg)dateiname)))

Land	xx (Tastaturcode)	yyy (Zeichensatztable)
Australien	US	850 oder 437
Belgien	BE	850
Dänemark	DK	850 oder 865
Deutschland	GR	850 oder 437
Finnland	SU	850 oder 437
Frankreich	FR	850 oder 437
Großbritannien	UK	850 oder 437
Italien	IT	850 oder 437
Kanada (engl.)	US	850 oder 437
Kanada (frz.)	CF	850 oder 863
Lateinamerika	LA	850 oder 437
Niederlande	NL	850 oder 437
Norwegen	NO	850 oder 865
Portugal	PO	850 oder 860
Schweden	SV	850 oder 437
Schweiz (dt.)	SG	850
Schweiz (frz.)	SF	850
Spanien	SP	850 oder 437
USA	US	850 oder 437

Wird yyy nicht angegeben, wird die Standard-Zeichensatztable 437 verwendet.

Der anzufügende Dateiname bezieht sich auf die Tastaturdefinitionsdatei. Standardmäßig ist KEYBOARD.SYS anzugeben. Wird der Dateiname nicht angegeben, so wird KEYBOARD.SYS im Root-Directory des aktuellen Laufwerks gesucht. Wird die Datei nicht gefunden, so erhalten Sie die Fehlermeldung:

Ungültige oder fehlende Datei zur Tastaturdefinition

Bemerkungen Wenn ein Tastaturprogramm geladen wird, so benötigt dieses permanent 5472 Byte an Speicherplatz. Durch Laden eines Zeichensatzes für ein anderes Land wird der vorher geladene Zeichensatz im Speicher gelöscht.

Wenn Sie die Tasten *Strg+Alt+F1* betätigen, so wird auf den Zeichensatz für USA umgeschaltet, ohne daß dieser geladen werden muß. Zurückschalten auf den geladenen Zeichensatz (z.B. deutsch) *Strg+Alt+F2*.

Durch die Betätigung folgender Tastenkombinationen lassen sich zusätzliche Zeichen erzeugen:

AltGr+ erzeugt das Zeichen:

2	²
3	³
7	{
8	[
9	]
0	}
ß	\
q	@
+	~
<	!
m	µ

Sollte auf Ihrer Tastatur die Taste *AltGr* nicht vorhanden sein, so beachten Sie für die Erzeugung obiger Zeichen die Angaben in den Handbüchern zu Ihrem Computer.

Wird DOS mit SELECT auf eine Festplatte oder Diskette kopiert, wird KEYB automatisch installiert.

Bei Tasten mit einer Beschriftung auf der Vorderseite wird die angegebene Funktion durch Betätigung von *Strg+Alt+* der entsprechenden Taste erreicht. Es werden jetzt auch Tottastenkombinationen unterstützt. Mit der deutschen Tastaturbelegung betrifft dies nur die Taste mit Akzent. Bei Betätigung wird dabei nichts angezeigt, sondern erst nach Eingabe eines der folgenden Buchstaben:

a e E i o u

Möglich sind damit:

á é É í ó ú à è ì ò ù

Soll ein Akzentzeichen alleine erscheinen, so betätigen Sie nach dieser Taste die Leertaste.

Beachten Sie bitte auch den Befehl **MODE** und das Kapitel 13.

KEYB beendet mit folgenden Ausgangscodes (beachten Sie den Befehl **IF ERRORLEVEL**):

- 0 Fehlerfrei ausgeführt
- 1 Ungültiger Tastaturcode, Zeichensatztablette oder Kommandoeingabe
- 2 Falsche oder fehlende Tastaturdefinitionsdatei
- 3 Es konnte keine Tastaturtablette im Speicher erstellt werden
- 4 Während der Kommunikation mit **CON** (Kapitel 10) trat ein Fehler auf
- 5 Angeforderte Zeichensatztablette wurde nicht vorbereitet (siehe Befehl **MODE**)
- 6 Die Umsetzungtablette für die ausgewählte Zeichensatztablette kann in der residenten Tastaturtablette nicht gefunden werden

LABEL *externer Befehl*

Zweck Erstellen, Ändern oder Löschen eines Namens für eine Diskette oder Festplatte.

Kommandoaufbau (d:)(weg)label
(d:)(weg)label (d:)(name)

Beispiele Zum ersten Mal 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 (bei längerer Eingabe wird der Rest ignoriert).

label a:

Es wird am Bildschirm z.B. angezeigt:

Diskette/Platte, Laufwerk A: hat den
Namen DOS

Name (max. 11 Zeichen) oder Eingabetaste

Wenn Sie nur *Return* drücken, erscheint:

Momentanen Namen löschen (Y/N)?

Wenn Sie den Namen löschen wollen, so drücken Sie nur den Buchstaben *y* 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, die mit den Befehlen **ASSIGN** oder **SUBST** umgeleitet wurden. Ferner darf er nicht für ein Netzwerklaufwerk verwendet werden.

MKDIR oder **MD** *interner Befehl*

Zweck Mit diesem Befehl kann ein neues Sub-Directory angelegt werden (beachten Sie dazu auch das Kapitel 8).

Kommandoaufbau `mkdir (d:)weg`
`md (d:)weg`

Beispiele `mkdir \text`

`oder`

`mkdir text`

Es wird ein neues Sub-Directory mit dem Namen TEXT angelegt (der Eintrag ist im aktuellen Directory sichtbar). Dazu werden das aktuelle Datum und die Uhrzeit gespeichert. Ferner werden im neuen Sub-Directory angelegt:

- . dahinter verbirgt sich der Name des neuen Directorys
- .. dahinter verbirgt sich der Name des hierarchisch vorher liegenden Directorys

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 ROOT-Directory sein).

`md .`

`und`

`md ..`

Mit diesen beiden Befehlen erhalten Sie eine Fehlermeldung:

Verzeichnis kann nicht angelegt werden

Diese Namen können bei dem Befehl MKDIR nicht verwendet werden.

Bemerkung Wenn Sie ASSIGN, JOIN oder SUBST verwenden, sollten Sie die damit benutzten Directory-Namen bzw. Laufwerke nicht mit MD benutzen bzw. sehr vorsichtig sein, da dies unter Umständen zu nicht vorhersehbaren Fehlern führen kann.

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; Vorbereiten und Verwalten von Zeichensatztabellen.

Wenn Sie MODE nach jedem Start Ihres Computers benötigen, sollten Sie diesen Befehl in die Datei AUTOEXEC.BAT einfügen (siehe Kapitel 12). Der Befehl benötigt 496 Byte Speicher für den dauerhaft geladenen Teil.

Kommandoaufbau (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 dieser drucken kann (kann nur mit *Strg+Untbr* beendet werden). Der Parameter sollte nicht für Netzwerkdrucker verwendet 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). Wenn der Drucker nicht eingeschaltet ist, erhalten Sie trotz Parameter **P** eine Fehlermeldung.

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).

Wird der Drucker neu gestartet oder initialisiert, so gelten wieder die Standardwerte. Beachten Sie, daß beim Starten von Basic der Drucker initialisiert wird und die eventuell mit MODE geänderten Werte automatisch auf die Standardwerte geändert werden. Bei Angabe falscher Werte erhalten Sie eine Fehlermeldung. Die

eingestellten Werte werden am Bildschirm angezeigt. Sollte bei der Eingabe des MODE-Befehls der Drucker nicht eingeschaltet sein, so erhalten Sie bei Angabe der Werte für **n** und **m** die Meldung

Druckerfehler

mode (n)(,m)(,t)

Bildschirm konfigurieren

n bezeichnet den Bildschirm-Modus :

40	40	Zeichen pro Zeile (für Farbgrafikadapter)
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

BW und CO schalten gleichzeitig vom aktiven Bildschirmadapter auf den Farbgrafikadapter um.

	MONO	Auf Monochrom-Bildschirmadapter (80 Zeichen pro Zeile) vom aktiven Bildschirmadapter umschalten
m	R	Bildschirm 1 Zeichen nach rechts verschieben (max. 2 Zeichen)
	L	Bildschirm 1 Zeichen nach links verschieben (max. 1 Zeichen)
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.

Der Befehl wirkt sich nur aus, soweit dieser von den in Ihrem System eingebauten Bildschirmadaptern unterstützt wird. Zum Beispiel kann zwischen Farb- bzw. Monochrom-Bildschirmadapter nur geschaltet werden, wenn diese in Ihrem Computer vorhanden sind. Ist dies nicht der Fall, wird Ihre Eingabe ohne Fehlermeldung ignoriert. Gleichzeitig mit dem Befehl wird die vorhandene Bildschirmanzeige gelöscht.

```
mode co40
```

Der Bildschirm wird auf 40 Zeichen pro Zeile umgeschaltet. Damit erscheint die Schrift größer. Gleichzeitig von einem eventuell anderen aktiven auf den Farbgrafik-Bildschirmadapter umschalten.

```
mode COM#(:)baud-rate(,(parity)(,(daten-  
länge)(,(stop-bit)(,(p))))
```

Seriellen Ausgang konfigurieren

#	Nummer des seriellen Ausgangs (1–4, falls vorhanden)
baud-rate	Übertragungsgeschwindigkeit (110, 150, 300, 600, 1200, 2400, 4800, 9600 oder 19200). Es reicht, die ersten beiden Stellen anzugeben
parity	Bestimmt die Art der Parity (odd (O) ungerade, even (E) gerade oder no parity (N) keine); Standardwert=E
daten-länge	(7 oder 8 Bit); Standardwert = 7
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 <i>Strg+Untbr</i> beendet werden)

Nach der Befehlseingabe werden Ihnen die eingestellten Werte zur Kontrolle am Bildschirm angezeigt.

```
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 (z.B. Drucker).

```
mode lpt#(:)=com#
```

Leitet Daten vom parallelen zum seriellen Ausgang um (LPT1 bis LPT3 und COM1 bis COM4).

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.

```
mode einheit codepage prepare  
= ((cplist)(d:)(weg)dateiname)
```

oder

```
mode einheit codepage prepare  
= ((cp)(d:)(weg)dateiname)
```

Vorbereiten von Zeichensatztabellen

einheit	CON, PRN, LPT1, LPT2 oder LPT3 (siehe Kapitel 10)
cp	Zeichensatztabelle (437, 850, 860, 863 oder 865)
cplist	Angabe mehrerer Zeichensatztabellen (Werte in Klammern angeben)
dateiname	Datei mit den Zeichensatztabellen (File-Typ CPI) 4201.CPI = IBM PC Grafikdrucker II 5202.CPI = IBM Quietprinter III, Modell 5202 EGA.CPI = Einheiten, die an HR-Grafikadaptoren angeschlossen sind LCD.CPI = IBM-PC-AP-LCD-Bildschirm
CODEPAGE	kann mit CP abgekürzt werden
PREPARE	kann mit PREP abgekürzt werden

Mit diesem Befehl können Zeichensatztabellen für Bildschirm bzw. Drucker vorbereitet werden. Beachten Sie hierzu auch das Kapitel 14 (DISPLAY.SYS und PRINTER.SYS).

```
mode lpt1: cp prep=((850,437) c:\dos\4201.cpi)
```

Die Zeichensatztabellen 850 und 437 wurden für den Drucker an der 1. parallelen Schnittstelle vorbereitet. Das Schriftbild des Grafikdruckers II IBM 4201 wird verwendet. Nicht jede Zeichensatztabelle kann mit jeder CPI-Datei vorbereitet werden. Bezüglich eventuell auftretender Fehlermeldungen beachten Sie bitte die Erläuterungen in Ihrem DOS-Handbuch. Weitere Informationen finden Sie in Kapitel 13.

```
mode einheit codepage select=cp
```

Auswählen/Aktivieren einer Zeichensatztabelle

einheit CON, PRN, LPT1, LPT2 oder LPT3 (siehe Kapitel 10)
 cp Zeichensatztabelle (437, 850, 860, 863 oder 865)
 Die Zeichensatztabelle muß vorher mit MODE
 vorbereitet sein
 CODEPAGE kann mit CP abgekürzt werden
 SELECT kann mit SEL abgekürzt werden

Die Zeichendaten der Zeichensatztabelle werden in den Drucker
 geladen (nur bei LPT #) und dort aktiviert.
 Bitte beachten Sie, daß dieser Vorgang nur dann korrekt ausge-
 führt wird, wenn auch ein Gerät, entsprechend der CPI-Datei,
 angeschlossen ist.

```
mode lpt1: cp sel=850
```

```
mode einheit codepage (/status)
```

Anzeigen der aktiven Zeichensatztabelle

Es werden die aktive Zeichensatztabelle sowie eine Liste der vor-
 bereiteten Zeichensatztabellen angezeigt. Bei den dabei angezeig-
 ten Hardware-Zeichensatztabellen handelt es sich um die mit
 DEVICE (PRINTER.SYS und DISPLAY.SYS) definierten
 Tabellen. Die angegebenen vorbereiteten Zeichensatztabellen
 wurden mit MODE definiert. Die Angabe /STATUS kann wegge-
 lassen werden.

```
mode lpt:1 cp /status
```

Anschließend erfolgt in unserem Beispiel die Anzeige:

```
Aktive Zeichensatztabelle für
Einheit LPT1 ist 850
Hardware Zeichensatztabellen:
```

```
    Zeichensatztabelle 437
Vorbereitete Zeichensatztabellen:
```

```
    Zeichensatztabelle 850
    Zeichensatztabelle 437
    Zeichensatztabelle nicht vorbereitet
Zeichensatztabellen-Funktion von
MODE STATUS beendet
```

Folgender Eintrag liegt in der Datei CONFIG.SYS vor (Kapitel 14):

```
device=c:\dos\printer.sys lpt1:=(4201,437,3)
```

```
mode einheit cp refresh
```

Reaktivieren einer Zeichensatztafel

Sollte z.B. der Drucker seit dem letzten Aktivieren einer Zeichensatztafel für ihn ausgeschaltet worden sein, so kann die Tabelle reaktiviert werden. Schalten Sie Ihren Drucker ein, und geben Sie folgenden Befehl ein:

```
mode lpt1: cp refresh
```

Die Zeichensatztafel für Ihren Drucker wird in den Drucker-
speicher geladen.

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). Die Daten der Standard-Eingabeeinheit werden seitenweise auf die Standard-Ausgabeeinheit übertragen.

Kommandoaufbau (d:)(weg)more
(d:)(weg)befehl | more
(d:)(weg)more < dateiname

Beispiele type c:test | more

Die Datei TEST auf der Diskette in Laufwerk C 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:

---Fortsetzung---

more < c: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 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.

tree | more

Da die von TREE angezeigten Werte in der Regel sehr umfangreich sind und damit eine Bildschirmseite nicht ausreichend ist, sollten Sie den Befehl immer mit MORE verketteten.

NLSFUNC *externer Befehl*

Zweck Bevor mit dem Befehl CHCP eine Zeichensatztabelle für alle Einheiten geladen werden kann, muß NLSFUNC ausgeführt werden. Damit wird der erweiterte landesspezifische Unterstützungscode geladen.

Kommandoaufbau (d:)(weg)nlsfunc ((d:)(weg)dateiname)

Beispiel nlsfunc c:\dos\country.sys

Bemerkung Wird der Dateiname bzw. das Laufwerk und der Zugriffspfad nicht angegeben, so werden die Angaben aus der Datei CONFIG.SYS zu COUNTRY= verwendet.

PATH *interner Befehl*

Zweck PATH benennt die Wege, die zu einem Verzeichnis führen, in dem ein externer Befehl gesucht werden soll. Ein Befehl, der nicht im aktuellen Directory steht, wird gefunden, wenn dessen Directory 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-Directories SYSTEM in Laufwerk C und TEXTE (Sub-Directory von SUB) in Laufwerk A in jedem anderen Directory verwendet werden.

Ein eingegebener Befehl wird erst im aktuellen Directory gesucht. Findet ihn DOS dort nicht, so wird das Sub-Directory SYSTEM auf Laufwerk C durchsucht. Wird der Befehl dort wiederum nicht gefunden, so durchsucht DOS auch noch das Sub-Directory TEXTE auf Laufwerk A. Erst wenn der Befehl dort wiederum nicht gefunden wird, erhalten Sie eine Fehlermeldung.

```
path a:.
```

Nach dem Durchsuchen des aktuellen Directories auf dem aktuellen Laufwerk wird das aktuelle Directory von Laufwerk A durchsucht.

Bemerkung Wenn Sie die Fehlermeldung erhalten, daß nicht ausreichend Speicher vorhanden ist, so können Sie mit COMMAND den Environment-Speicher erweitern.

Beispiel command /e:1024

Damit wird der Speicher auf 1 Kbyte vergrößert (maximal 32768). Ohne diesen Befehl beträgt die Standardgröße 128 Byte.

Geben Sie bei PATH einen unbekannten oder nicht vorhandenen Weg an, so erhalten Sie **keine** Fehlermeldung. Die fehlerhafte Angabe wird beim Suchvorgang von DOS ignoriert.

Programme (MODE, PRINT, GRAPHICS usw.), die über einen residenten Teil verfügen, sollten vor der Eingabe von PATH geladen werden.

PAUSE *interner Befehl*

Zweck PAUSE unterbricht die Ausführung einer BATCH-Datei (siehe Kapitel 12) und gibt einen Text am Bildschirm aus, durch den Sie aufgefordert werden, zur Fortsetzung der Arbeit eine beliebige Taste zu drücken. Die Abarbeitung einer BATCH-Datei kann durch Betätigung von *Strg+Untbr* abgebrochen werden – in diesem Falle fragt der Computer, ob Sie dies wirklich wollen:

Stapeljob beenden (J/N)?

Kommandoaufbau pause (text)
Der **text** kann maximal 121 Zeichen lang sein.

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 <Strg+
Untbr> können Sie abbrechen
```

erzeugen am Bildschirm die Anzeigen:

```
echo off
Legen Sie in das Laufwerk A eine neue Diskette ein
Weiter -> eine Taste betätigen
```

Ist ECHO eingeschaltet, werden der Befehl und der »text« angezeigt, im anderen Fall (siehe vorhergehendes Beispiel) nur die Aufforderung, eine beliebige Taste für die Fortsetzung der Verarbeitung zu betätigen.

PRINT *externer Befehl*

Zweck Mit PRINT kann eine oder mehrere Textdatei(en) ausgedruckt werden, während ein anderes Programm bearbeitet wird.

Kommandoaufbau (d:)(weg)print
 (d:)(weg)print (/x) (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 nach dem Start Ihres Systems angeben. Je größer der Wert ist, um so besser ist die Leistung von PRINT (Beschleunigung).
 /c Damit werden die angegebenen Dateien aus der Tabelle der noch zu druckenden Dateien entfernt (die Datei vor und die Dateinamen nach dem Parameter bzw. bis /P kommt). Auf den Drucker wird folgende Nachricht ausgegeben:
 Datei xxxxxxxx.xxx aus Druckerwarteschlange gelöscht
 /d:dev Normalerweise wird alles am Drucker (Gerätename PRN) ausgegeben. Hier können Sie einen anderen Gerätenamen (dev) angeben (Kapitel 10). Dieser Parameter muß als erstes angegeben werden (nur möglich beim ersten Aufruf von PRINT). Wird ein nicht vorhandenes Gerät angegeben, kann es zu nicht vorhersehbaren Reaktionen kommen.
 /m: # Spezifiziert, wieviel Zeit (Takte) PRINT für den Druck verwenden darf (bis eine weitere von Ihnen gleichzeitig durchgeführte Tätigkeit bearbeitet wird). Der Standardwert beträgt 2. Angaben von 1 bis 255 sind erlaubt. Kann nur beim ersten Aufruf von PRINT angegeben werden.

/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. Kann nur beim ersten Aufruf von PRINT angegeben werden.
/p	Fügt die angegebenen Dateien zu den zu druckenden hinzu (der Dateiname davor und alle folgenden bzw. bis /C kommt).
/q: #	Spezifiziert, wie viele Dateien Sie in die PRINT-Warteschlange schreiben können. Der Standardwert ist 10 (möglich ist ein Wert von 1 bis 32). Kann nur beim ersten Aufruf von PRINT angegeben werden.
/t	Ende – löscht alle Dateien aus der Warteschlange von PRINT, und der laufende Druck wird abgebrochen. Eine Nachricht zum Löschvorgang wird ausgedruckt (ein Signalton ertönt am Drucker): Alle Dateien aus Druckerwarteschlange gelöscht
/u: #	Setzt die PRINT-Wartezeit. Spezifiziert die Anzahl der Takte, bis der Drucker frei ist. Nach Ablauf der Zeit wird die von Ihnen gleichzeitig durchgeführte Tätigkeit bearbeitet. Der Standardwert ist 1 (Werte von 1 bis 255 sind erlaubt). Kann nur beim ersten Aufruf von PRINT angegeben werden. Ohne Angabe von Parametern gilt /P für die angegebenen Dateien.

Beispiele `print config.sys autoexec.bat`

Die Dateien CONFIG.SYS und AUTOEXEC.BAT sollen gedruckt werden. Es erscheint folgende Anzeige:

Name der Ausgabeeinheit [PRN]:

Residenter Teil von PRINT geladen

A:\CONFIG.SYS wird gerade gedruckt

A:\AUTOEXEC.BAT ist in der Warteschlange

Hinter dem Doppelpunkt nach [PRN] sollten Sie nur *Return* betätigen, ausgenommen, Sie verwenden eine andere als die Standarddruck-Ausgabeeinheit.

```
print
```

Es werden alle Dateinamen angezeigt, die sich in der Warteschlange befinden. Wenn keine vorhanden, wird angezeigt, daß die Warteschlange leer ist:

Die Druckerwarteschlange ist leer

B:\KAP.1 wird gerade gedruckt

B:\KAP.2 ist in der Warteschlange

B:\KAP.3 ist in der Warteschlange

Sollte der Drucker nicht bereit sein, erhalten Sie zusätzlich die Meldung:

Ausgabeeinheit ist evtl. ausgeschaltet
oder "OFFLINE". Bitte überprüfen.

```
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 a:test.alt /c
```

Die Datei TEST.ALT in Laufwerk A soll nicht gedruckt werden (aus der Drucker-Warteschlange löschen).

```
print test*. * /c a:kap2 /p a:kap3
```

Die Dateien TEST*. * werden aus der Druckerwarteschlange gelöscht, soweit diese noch nicht oder gerade im Druck sind. KAP2 und KAP3 von Laufwerk A werden zur Druckerwarteschlange hinzugefügt.

```
print a:test.neu a:test.alt c:test/c
```

Die Dateien TEST.NEU und TEST.ALT auf der Diskette in Laufwerk A werden zur Druckerwarteschlange hinzugefügt. Die Datei TEST wird aus der Warteschlange gelöscht.

Bemerkungen Findet PRINT einen Fehler auf der Diskette/Festplatte, geschieht folgendes:

- Der Druck der Datei wird abgebrochen
- Ausdruck einer Fehlernachricht
- Seitenvorschub und Warnsignal
- Die nächste Datei aus der Warteschlange wird gedruckt

Während PRINT aktiv ist bzw. der Druckvorgang läuft, darf mit keinem anderen Programm gedruckt werden, da es sonst normalerweise zu einer Fehlermeldung kommt (»Kein Papier«) oder PRINT abgebrochen wird (gleiche Reaktion wie die Verwendung des Parameters /T).

Die Angabe einer nicht vorhandenen oder benutzbaren Ausgabeinheit führt zu nicht vorhersehbarem Verhalten Ihres Computers.

PRINT dürfen Sie nicht am Server (Zentrale) eines Netzwerkes benutzen.

Sie sollten während eines Druckvorganges mit PRINT die Befehle FORMAT, DISKCOPY und DISKCOMP nicht verwenden, da im Einzelfall hierbei irreparable Fehler auf der Diskette/Festplatte entstehen können.

PROMPT *interner Befehl*

Zweck PROMPT ändert das Systemanforderungszeichen.

Kommandoaufbau prompt \$(text)

—	Zeilenvorschub (neue Zeile)
b	das Zeichen !
d	Datum
e	ESC-Zeichen
g	das Zeichen >
h	Schritt zurück bzw. Löschen des vorhergehenden Zeichens
l	das Zeichen <
n	das aktuelle Laufwerk (z.B. A oder C)
p	Name des aktuellen Laufwerks/Directorys
s	Leerschritt
t	Uhrzeit
v	Nummer der DOS-Version
\$	das Zeichen \$
q	das Zeichen =

Beispiele prompt \$n:

Der neue Prompt ist **A:** (A ist in diesem Fall 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.

Zeit = 11:54:34:56

Datum = 21.09.87

prompt \$n\$g

Damit wird der »normale« Prompt erzeugt (dies wird auch erreicht, wenn Sie nur den Befehl PROMPT ohne weitere Angaben verwenden).

prompt \$p\$g

Damit werden das aktuelle Laufwerk und Directory sowie das Zeichen > zu Systemanforderungszeichen. Diesen Befehl sollten Sie in die Datei AUTOEXEC.BAT einfügen (Kapitel 12).

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 14).

```
prompt $e[7m$p:$e[m
```

Der Prompt wird invers dargestellt und anschließend wieder auf normale Darstellung umgeschaltet, damit von Ihnen vorzunehmende Eingaben »normal« angezeigt werden.

```
prompt $e[33;44m$n$g
```

Wenn Sie einen Farbmonitor haben, ist anschließend der Hintergrund blau und die Schrift (Vordergrund) gelb bzw. rötlich.

```
prompt $e[s$e[1;1H$p$d $t$e[K$e[u-
```

Dieser Befehl bewirkt, daß in der obersten Bildschirmzeile das aktuelle Laufwerk/Directory, Datum und Uhrzeit angezeigt werden. In der Eingabezeile erscheint als Systemanforderungszeichen ein Bindestrich. Beachten Sie hierzu ANSI.SYS im Kapitel 14.

```
prompt $e[0;60;"DIR A:";13p
prompt
```

Die Funktionstaste *F2* (60) erhält den Inhalt **DIR A: Return**. Wenn Sie anschließend *F2* betätigen, wird der Befehl DIR A: ausgeführt. In älteren DOS-Versionen konnte die Taste bis zum nächsten Systemstart nicht mehr umbelegt werden (dies konnte zu Schwierigkeiten in Anwenderprogrammen führen, die selbst auch die Tasten mit einem anderen Code versahen, was dann nicht mehr möglich war). Der zweite PROMPT-Befehl ist notwendig, damit das Systemanforderungszeichen wieder ordnungsgemäß am Bildschirm angezeigt wird (nach dem ersten PROMPT fehlt das Anforderungszeichen am Bildschirm).

Bemerkungen

Die einzelnen Codes für INVERS, Farben usw. finden Sie im Kapitel 14.

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 vorgestellt.

RECOVER *externer Befehl*

Zweck RECOVER erneuert eine Datei oder Diskette/Festplatte mit fehlerhaften Sektoren.
RECOVER darf nicht bei einem Netzwerklaufwerk 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 oft nach dem »recovery« nicht mehr brauchbar (daher immer eine Kopie aller Daten und Programme mit möglichst aktuellem Stand fertigen – Befehl BACKUP).

recover a:

Die Diskette in Laufwerk A wird auf fehlerhafte Sektoren überprüft (dieses Beispiel sollten Sie nicht üben!).

Bemerkungen Beachten Sie bitte, daß das Programm alle bisherigen Dateinamen löscht und neue Namen vergibt (also vorher eine Kopie fertigen, um gegebenenfalls den ursprünglichen Zustand wiederherstellen zu können). Die Dateien erhalten in der bisherigen Reihenfolge die neuen Namen:

FILE0001.REC
FILE0002.REC
usw.

Wird ein Dateiname angegeben, so wird dieser soweit möglich nicht verändert. Die Dateigrößen sind nach dem Programmablauf meist nicht mehr mit der ursprünglich bei DIR angezeigten Größe identisch!

REM *interner Befehl*

Zweck Befehl für Kommentarzeilen in BATCH-Dateien (siehe auch Kapitel 12).

Kommandoaufbau rem (text)

Beispiel Folgende BATCH-Datei als 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 Verzeichnis 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). Ein Text darf maximal 123 Zeichen lang sein. Wenn Sie vor REM den Befehl

```
echo off
```

einfügen, werden die REM-Zeilen nicht angezeigt! Sonderzeichen wie z.B. ! oder > dürfen nur mit Hochkomma angegeben werden ("!" oder ">").

RENAME oder **REN** *interner Befehl*

Zweck Vorhandenen Dateien/Programmen einen neuen Namen zuordnen.

Kommandoaufbau `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`

Der Name der Datei TEST.NEU im Sub-Directory **sub** auf der Diskette in Laufwerk A wird in VERSUCH.NEU geändert. Dabei dürfen nur beim alten Namen der genaue Weg und die Laufwerksbezeichnung angegeben werden.

`ren *.txt *.doc`

Dabei erhalten alle Dateien mit dem File-Typ TXT den neuen Typ DOC zugewiesen, sofern die File-Namen übereinstimmen.

`ren a:text*.* ??s*.*`

Mit diesem Befehl werden nur Teile von Dateinamen verändert. Bearbeitet werden alle Dateinamen in Laufwerk A, die mit TEXT beginnen. Die ersten beiden Zeichen (**TE**) bleiben unverändert. Der dritte Buchstabe wird in **S** umgewandelt. Der Rest der Dateinamen bleibt unverändert. Damit wurde also nur das dritte Zeichen in den Namen verändert.

`ren a:test *.tst`

Die Datei TEST auf der Diskette in Laufwerk A erhält den File-Typ TST:

Alter Dateiname:	TEST
Neuer Dateiname:	TEST.TST

Bemerkung Benennen Sie **niemals** gekaufte Programme um, es sei denn, Sie werden extra dazu aufgefordert, da sonst eine einwandfreie Funktion nicht mehr sichergestellt ist.

REPLACE *externer Befehl*

Zweck Mit diesem Befehl können Dateien einer auf Festplatte oder Diskette vorhandenen Software durch neue, aktuellere Versionen ersetzt werden. Alte Dateien werden durch die neuen Versionen gleichen Namens ersetzt bzw. Dateien, die in der alten Fassung noch nicht existiert haben, hinzugefügt.

Kommandoaufbau (d:)(weg)replace (d:)(weg)dateinamen (d:)(weg)(/x ...)

- /x ... /a Hinzufügen neuer Dateien, die auf der Zieldiskette/Festplatte nicht vorhanden sind (nicht zusammen mit /S anwendbar).
- /p Vor dem Kopieren einer Datei wird eine Bestätigung verlangt, ob kopiert werden soll oder nicht.
- /r Dateien mit dem Attribut »Nur-Lesen« werden auch ersetzt.
- /s Das Zielverzeichnis **und** dessen Sub-Directories werden nach einer älteren Version bzw. gleichen Dateinamen durchsucht (kann nicht zusammen mit Parameter /A angewandt werden).
- /w Es wird vor dem Ersetzen oder Hinzufügen von Dateien gewartet, bis Sie eine Taste betätigen.
Bei Angabe von /W und /A: Zum Hinzufügen von Datei(en) eine beliebige Taste betätigen.
Bei Angabe von /W ohne /A: Zum Ersetzen von Datei(en) eine beliebige Taste betätigen.

Beispiele replace a:*.com c: /s

Auf der Diskette im Laufwerk A befinden sich Programme mit dem Dateityp COM. Diese sollen die Programme mit den gleichen Namen auf der Festplatte (Laufwerk C) ersetzen, wobei das Root- und die Sub-Directories durchsucht werden. Sind einzelne Dateien in Laufwerk C, also in verschiedenen Directories gespeichert, werden alle mit der neuen Version ausgetauscht.

replace c:*.dat a: /p/r/w

Vom aktuellen Directory auf Laufwerk C alle Dateien mit dem Typ DAT kopieren. 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 das »Nur-Lesen«-Attribut besitzen.

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 Datei hat »Nur-Lesen«-Attribut (mit /R noch mal versuchen)
 - 8 nicht genügend Speicherplatz vorhanden
 - 11 ungültige (bzw. ungültige Anzahl) von Parametern
 - 15 Laufwerk existiert nicht
 - 22 ungültige DOS-Version
- andere Nummern sind DOS-Standardfehler

RESTORE *externer Befehl*

Zweck RESTORE holt die mit BACKUP gesicherten Dateien auf Festplatte (oder Diskette) zurück. Mit BACKUP gesicherte Dateien dürfen nur mit diesem Befehl zurückgesichert werden! Wenn ASSIGN, JOIN oder SUBST aktiv sind, muß RESTORE mit großer Vorsicht angewandt werden. Siehe auch Beschreibung zu BACKUP in diesem Kapitel.

Kommandoaufbau (d:)(weg)restore d: ((d:)(weg)(dateinamen) (/x...)
 /x... /p Wenn eine Datei das Attribut »Nur-Lesen« (Befehl ATTRIB) hat oder seit dem letzten BACKUP verändert wurde, werden Sie am Bildschirm gefragt, ob eine Rückkopie erfolgen soll.

/s Es werden auch alle Sub-Directories zurückkopiert (ab dem aktuellen oder angegebenen Directory).

/b:tt.mm.jj Zurückkopieren aller Dateien, die an oder vor dem angegebenen Datum geändert wurden.

/e:hh.mm.ss Zurückkopieren aller Dateien, die an oder vor der angegebenen Uhrzeit geändert wurden.

/a:tt.mm.jj Zurückkopieren aller Dateien, die an oder nach dem angegebenen Datum geändert wurden.

/l:hh.mm.ss Zurückkopieren aller Dateien, die an oder vor der angegebenen Uhrzeit geändert wurden.

/m Zurückkopieren aller Dateien, die seit dem letzten Sichern (BACKUP) verändert oder gelöscht wurden.

/n Zurückkopieren aller Dateien, die auf der Zielfestplatte (oder -diskette) nicht mehr vorhanden sind.

Beispiele restore a: c:*. * /s

Es werden vom Laufwerk A 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 (PROG.COM muß mit BACKUP aus dem gleichen Directory gesichert worden sein).

```
restore a: c:*.dat
```

Alle DAT-Dateien der Diskette, die aus dem aktuellen Directory mit BACKUP 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 für das Zurückkopieren gefunden
- 2 File-Sharing-Konflikt (Netzwerk), einige oder alle Dateien nicht zurückkopiert
- 3 Sie haben RESTORE abgebrochen (durch *Strg+Untbr* oder *ESC*)
- 4 RESTORE durch Fehler abgebrochen

Es können im Netzwerk nur Dateien zurückkopiert werden, für die Sie die Zugriffsberechtigung besitzen. Im anderen Fall erhalten Sie die Fehlermeldung (Fehlercode 2):

```
(d:)(weg)dateiname
```

```
*** Datei kann nicht zurückgespeichert werden ***
```

Wird nur ein Teil von mit BACKUP gesicherten Dateien zurückkopiert, so legen Sie die Sicherungsdiskette ein, die die gesuchten Dateien enthält (siehe LOG-Datei, die mit BACKUP erstellt werden kann). Wissen Sie nicht, auf welcher Diskette die Daten gesichert sind, so legen Sie die erste Sicherungsdiskette ein. RESTORE sucht die Dateien selbsttätig und fordert Sie bei Bedarf auf, die nächste Diskette einzulegen.

Die Dateien IBMBIO.COM, IBMDOS.COM und COMMAND.COM können nicht zurückkopiert werden. Diese Dateien sollten Sie mit SYS und COPY von Ihrer Original-DOS-Diskette kopieren. Das Zurückkopieren kann auf das gleiche oder ein anderes Laufwerk erfolgen, unabhängig davon, auf welchem Laufwerk die Sicherung mit BACKUP stattgefunden hat.

Für das Zurückkopieren in die einzelnen Directories ist es erforderlich, daß die Dateien in die gleichen Directories kommen, in denen sie mit BACKUP gesichert wurden. Gesicherte Dateien eines Directories können nicht in ein beliebiges Directory anderen Namens kopiert werden.

Beim Zurückkopieren mit dem Parameter /s werden eventuell nicht mehr vorhandene Sub-Directories neu angelegt.

Wenn APPEND/X aktiv ist, muß vor BACKUP und RESTORE erst der Befehl APPEND ohne Zusatzangaben eingegeben werden.

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 funktioniert der Befehl RD nicht). 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

rd versuch

Löschen des Sub-Directorys VERSUCH – nicht möglich, da darin noch VERSUCH2 vorhanden ist. Fehlermeldung:

Ungültiger Pfad, kein Verzeichnis,
oder Verzeichnis nicht leer

rd versuch\versuch2

Löschen des Sub-Directorys VERSUCH2

rd versuch

Sub-Directory VERSUCH kann jetzt gelöscht werden, da es leer ist

Bemerkungen Wenn Sie die Befehle ASSIGN und JOIN verwenden, ist bei der Benutzung von RD Vorsicht geboten. Ein Directory, das mit SUBST umgeleitet wurde, kann nicht gelöscht werden. Ein Directory kann nur gelöscht werden, wenn sich darin nur noch die Dateien ».« und »..« befinden.

SELECT *externer Befehl*

Zweck Installiert auf einer neuen Diskette DOS mit der nationalen Tastaturanpassungs- sowie passendem Datums- und Uhrzeitformat.

Beachten Sie hierzu das Kapitel 7.

Kommandoaufbau (d:)(weg)select (s:) (d:)(weg) xxx yy

s: Nur Laufwerk A oder B können angegeben werden (ohne Angabe standardmäßig A)

(d:)(weg) Angabe des Directorys, in das die externen DOS-Befehle kopiert werden sollen

xxx Landescode (Datums- und Uhrzeitformat, Währungssymbol, Dezimalzeichen, Sortierreihenfolge und Großschreibung)

yy Tastaturcode (z.B. GR für Deutschland)

Land	Landescode	Tastaturcode
Arabischer Sprachraum	785	nicht vorhanden
Australien	061	US
Belgien	032	BE
Dänemark	045	DK
Deutschland	049	GR
Finnland	358	SU
Frankreich	033	FR
Großbritannien	044	UK
Hebräischer Sprachraum	972	nicht vorhanden
Italien	039	IT
Kanada (engl.)	001	US
Kanada (frz.)	002	CF
Lateinamerika	003	LA
Niederlande	031	NL
Norwegen	047	NO
Portugal	351	PO
Schweden	046	SV
Schweiz (deutsch)	041	SG
Schweiz (frz.)	041	SF
Spanien	034	SP
USA	001	US

SELECT verwendet die Befehle FORMAT und XCOPY. Ferner werden die Dateien AUTOEXEC.BAT (Kapitel 12) und CONFIG.SYS (Kapitel 14) erstellt.

Beispiele `select a: c:\dos 049 gr`

Hierbei kann auf die Laufwerksangabe A: verzichtet werden, da diese den Standardwert darstellt. Das Festplattenlaufwerk C wird installiert.

Sie erhalten folgende Meldung am Bildschirm:

```
SELECT wird beim ersten Installieren von
DOS benutzt. Durch SELECT werden alle
Daten auf der angegebenen Einheit ge-
löscht und DOS wird installiert.
Fortfahren (J/N)? J
```

Für die Fortsetzung des Programmes geben Sie **j** bzw. *Return* ein. Anschließend wird mit **FORMAT** die Diskette formatiert und dann werden automatisch mit **XCOPY** alle Dateien (*.*) übertragen. Nach dem Formatieren werden Sie noch nach dem Namen für die Festplatte gefragt.

Bitte achten Sie darauf, daß sich während des gesamten Vorganges die DOS-Diskette im Laufwerk A befindet. Befindet sich die DOS-Diskette in einem anderen Laufwerk, so muß vor der Ziel Laufwerksangabe auch das DOS-Laufwerk angegeben werden (nur A oder B möglich).

Befindet sich das Programm für die Tastaturanpassung nicht auf der Diskette in Laufwerk A, so erhalten Sie eine entsprechende Meldung, mit der Sie aufgefordert werden, die Diskette mit dem Programm einzulegen. Wird die Datei nicht gefunden, so erhalten Sie die Fehlermeldung:

Tastaturroutine nicht gefunden

Folgende Dateien werden auf der Zieldiskette (oder Festplatte) erstellt:

CONFIG.SYS	COUNTRY=049,437,C:\DOS\COUNTRY.SYS
AUTOEXEC.BAT	PATH C:\;C:\DOS;
	KEYB GR,437,C:\DOS\KEYBOARD.SYS
	ECHO OFF
	DATE
	TIME
	VER

SET *interner Befehl*

Zweck SET setzt den Wert einer Zeichenkette gleich dem Wert einer anderen Zeichenkette zur Benutzung in Programmen (kommt vor allem in Zusammenhang mit BATCH-Dateien zur Anwendung). Die mit SET getroffenen Vereinbarungen werden in den Environment-Bereich (DOS-Umgebung) geschrieben.

Die Eingabe von SET ohne weitere Angaben führt zur Anzeige der gesamten Werte aus dem Environment-Bereich. Geben Sie nur die **zeichenkettel** ein, so wird diese zusammen mit **zeichenkette2** gelöscht.

In den Environment-Bereich schreiben auch die Programme PROMPT und PATH. Ferner wird die COMSPEC-Angabe dort gefunden (Angabe, wo der Kommandointerpreter – COMMAND.COM – gefunden werden kann). Diese Angabe wird beim Start von DOS geschrieben.

Der Environment-Bereich ist 127 Zeichen groß und wird bei Bedarf automatisch dynamisch erweitert (siehe auch COMMAND.COM in diesem Kapitel). Laden Sie ein Programm mit einem residenten Teil (MODE, PRINT, GRAPHICS usw.), so können Sie nur 127 Zeichen benutzen, außer Sie erweitern diesen mit dem Befehl **COMMAND /E**.

Programme mit einem residenten Teil sollten stets vor Eingaben mit PATH usw. erfolgen.

Kommandoaufbau

set

set (zeichenkettel=(zeichenkette2))

Die Angaben bei **zeichenkettel** werden automatisch in Großbuchstaben umgewandelt.

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=\DOS;\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
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 Platz mehr im Umgebungs-bereich** erhalten, so ist nicht mehr genügend Environment-Speicher vorhanden! Dieser Speicher hat nur eine auf 127 Zeichen begrenzte Größe (kann mit **COMMAND** vergrößert werden – z.B. **COMMAND /E:1024**).

Beachten Sie auch den Befehl **SHELL** im Kapitel 14.

SHARE *externer Befehl*

Zweck Installiert den sogenannten File-Sharing-Support, wenn Sie ein Netzwerk einsetzen.

Unterstützung für gemeinsamen Dateizugriff und Diskettenaustauschschutz in Netzwerken.

Vereinzelt notwendig, wenn die Befehle SUBST und JOIN benutzt werden (sollten diese beiden Befehle trotz fehlerfreier Anwendung nicht akzeptiert werden, benutzen Sie vorher den Befehl SHARE, auch wenn Sie kein Netzwerk besitzen).

Kommandoaufbau (d:)(weg)share

(d:)(weg)share /x

/x /f: # Definition der Größe des Speichers für Informationen zu Dateien, die im Netzwerk verwendet werden. Für jede verwendete (offene) Datei wird Speicher für den vollen Dateinamen und 11 weitere Stellen benötigt. Der Standardwert für # ist 2048.

/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 Findet SHARE in CONFIG.SYS die Angabe von FCBS z.B. mit dem Standardwert 4,0, so wird diese Tabellenangabe automatisch auf 16,8 geändert.

Versuchen Sie, SHARE ein zweites Mal zu laden, erhalten Sie die Fehlermeldung:

SHARE bereits installiert

Manche Computer können feststellen, ob ein Diskettenlaufwerk geöffnet wurde. SHARE kann diese Information verarbeiten, wenn dies während einer kritischen Verarbeitung geschieht.

Wird ein Laufwerk geöffnet, bevor DOS eine Lese- oder Schreiboperation abgeschlossen hat, so überprüft es den Namen der Diskette, bevor der Vorgang fortgesetzt wird. Ist der Name der Diskette ein anderer, so erhalten Sie eine Meldung, die richtige Diskette wieder einzulegen – anschließend setzt DOS die Verarbeitung fort.

Weitere Informationen finden Sie im Handbuch zur Netzwerksoftware von Microsoft (»Networks Manager's Guide«) oder IBM 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

Beispiel BATCH-Datei LOESCHE1.BAT:

```
rem Dies ist die Datei LOESCHE1.BAT
:anfang
if "%1"==" " goto ende
del %1
shift
goto anfang
:ende
```

Beispiel für den Aufruf der Datei LOESCHE1.BAT:

```
loesche1 datei1 datei2 datei3
```

Beispiel für die Anzeigen bzw. Aktionen:

```
rem Dies ist die Datei LOESCHE1.BAT
if "datei1"==" " goto ende
del datei1
shift
goto anfang
if "datei2"==" " goto ende
del datei2
shift
goto anfang
if "datei3"==" " goto ende
del datei3
shift
goto anfang
if ""==" " goto ende
```

Die zum Befehl eingegebenen Dateinamen werden der Reihe nach bearbeitet und im vorliegenden Beispiel gelöscht. Der Befehl **SHIFT** »holt« also immer den folgenden Namen aus der von Ihnen eingegebenen Befehlszeile.

	loeschel	dateil	datei2	datei3
	%0	%1	%2	%3
SHIFT	%0		%1	%2
SHIFT	%0			%1
SHIFT	%0			

SORT *externer Befehl*

Zweck Sortieren von Daten. SORT ist ein Filter. Die Daten können auf- oder absteigend sortiert werden. Beachten Sie dazu auch das Kapitel 9. Die Daten werden von der Standard-Eingabeeinheit gelesen und auf die Standard-Ausgabeeinheit ausgegeben.

Kommandoaufruf (d:)(weg)sort (/x) (< dateiname1) (> dateiname2)

/x /r sortiert absteigend (sonst aufsteigend)

/+ # sortiert die Datenzeilen ab der Spalte # (Standardwert ist 1)

dateiname1 und **dateiname2** dürfen nicht identisch sein.

Zeichen, die in der Codetabelle (Anhang IV) über 127 stehen, werden teilweise mit einem anderen Stellenwert sortiert (nur wenn kein Landescode geladen wurde):

ASCII-Wert (dezimal)	Stellenwert	ASCII-Wert (dezimal)	Stellenwert
128 Ç	67	153 Ö	79
129 ü	85	154 Ü	85
130 é	69	155 ø	36
131 â	65	156 £	36
132 ä	65	157 Ÿ	36
133 à	65	158 Þ	36
134 â	65	159 ð	36
135 ç	67	160 á	65
136 ê	69	161 í	73
137 ë	69	162 ó	79
138 è	69	163 ú	85
139 ï	73	164 ñ	78
140 î	73	165 Ñ	78
141 ì	73	166 ã	166
142 Ä	65	167 °	167
143 Å	65	168 ¿	63
144 É	69	169 ƒ	169
145 æ	65	170 ƒ	170
146 Æ	65	171 ½	171
147 ó	79	172 ¼	172
148 ö	79	173 ï	33
149 ò	79	174 «	34
150 û	85	175 »	34
151 ù	85	225 ß	83
152 ÿ	89		

Für die Sortierung ist folgendes zu beachten, wenn ein Landescode geladen wurde:

Zeichen über 127 (siehe vorherige Tabelle) werden gemäß den Regeln für das ausgewählte Land sortiert.

Die zu sortierenden Dateien dürfen maximal 63 Kbyte groß sein. Kleinbuchstaben werden wie Großbuchstaben behandelt, und es wird immer zeilenweise sortiert.

Beispiele `sort < a: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 a: | sort`

Das Inhaltsverzeichnis des Laufwerks A wird aufsteigend sortiert am Bildschirm angezeigt.

`dir a: | sort > prn`

Das Inhaltsverzeichnis des Laufwerks A wird aufsteigend sortiert am Drucker ausgedruckt.

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

Das Inhaltsverzeichnis des Laufwerks C wird absteigend nach Dateigrößen (die Anzahl der Byte 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 wurde.

Inhaltsverzeichnis nach Monaten und Jahren sortieren:

`dir c: > dir.dat`

`sort /+27 < dir.dat > dir.srt`

Das nach Monaten sortierte Verzeichnis (DIR.SRT) muß jetzt mit einem Editor (z.B. EDLIN) oder einem Textprogramm bearbeitet werden. Damit anschließend nicht nach der Uhrzeit sortiert wird, muß diese bzw. alles hinter der Jahreszahl Stehende aus den einzelnen Zeilen gelöscht werden.

`sort +30 < dir.srt > dir.dat`

SUBST *externer Befehl*

Zweck Einem Sub-Directory wird eine Laufwerksbezeichnung zugewiesen. Die Befehle

ASSIGN

BACKUP

DISKCOMP

DISKCOPY

FDISK

FORMAT

JOIN (nicht in Verbindung mit den gleichen Directories bzw. Laufwerken)

LABEL

RESTORE

dürfen nicht verwendet werden, wenn SUBST aktiv ist. Folgende Befehle dürfen nicht in Verbindung mit den bei SUBST verwendeten Wegen stehen:

CHDIR oder CD

MKDIR oder MD

RMDIR oder RD

PATH

SUBST darf nicht für ein Netzwerklaufwerk verwendet werden, sonst erhalten Sie die Fehlermeldung:

SUBST für Netzwerklaufwerk nicht möglich

Sollte der Befehl auf Ihrem Computer 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: b:\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-Directorys FIBU. Die Angabe des **Weges** sollte grundsätzlich wie im vorliegenden Beispiel im Root-Directory beginnen.

subst

Es werden Ihnen die definierten und aktiven Zuweisungen angezeigt:

F: => B:\AUFTRAG\FIBU

subst f: /d

Die Zuweisung eines Sub-Directorys für Laufwerk F wird gelöscht.

Bemerkungen

Normalerweise sind nur die Laufwerksbezeichnungen bis E möglich. Wenn Sie in der Datei CONFIG.SYS mit dem Befehl LAST-DRIVE arbeiten, können Namen bis maximal Z vergeben werden (siehe Kapitel 14).

Der Befehl SUBST ist sinnvoll, wenn Sie mit Programmen (z.B. WordStar 3.4) arbeiten, die mit **Wegen** nicht arbeiten können. In diesem Falle legen Sie ein Sub-Directory für die Texte an und weisen diesem mit SUBST eine Laufwerksbezeichnung zu, die Sie in Ihrem Programm verwenden können. Ebenso ist der Befehl sinnvoll, wenn Sie häufig mit sehr langen »Weg«-Angaben arbeiten müssen.

SYS *externer Befehl*

Zweck SYS kopiert die Systemdateien IBMBIO.COM und IBMDOS.COM auf eine Diskette/Festplatte. Beachten Sie bitte, daß auf der Empfängerdiskette/-festplatte auf den ersten Sektoren ausreichend Platz vorhanden sein muß, sonst wird zwar das System unter Umständen kopiert, es kann aber beim Starten nicht ordnungsgemäß geladen werden.

Der erste Bereich (Sektoren) ist meist zu klein, wenn vorher eine ältere Version von DOS dort gespeichert war. Daher muß in diesem Fall vorher ein BACKUP durchgeführt und anschließend mit FORMAT neu formatiert werden.

Der Befehl darf nicht für ein Netzwerklaufwerk verwendet werden.

Kommandoaufbau (d:)(weg)sys d:

Beispiel sys a:

Das System wird vom aktuellen Laufwerk auf die Diskette in Laufwerk A übertragen.

Bemerkungen Soll mit SYS das DOS auf eine bereits formatierte Diskette oder Festplatte übertragen werden, die eventuell bereits Daten beinhaltet, so muß bei FORMAT der Parameter /S oder /B angegeben worden sein.

Die Dateien IBMBIO.COM und IBMDOS.COM müssen immer die beiden ersten Verzeichniseinträge und in aufeinanderfolgenden Sektoren gespeichert sein (dies gilt nur für IBMBIO.COM).

TIME *interner Befehl*

Zweck Anzeigen bzw. Ändern der Uhrzeit. Bei Systemen mit einer batteriegepufferten Uhrzeit (Real Time Clock – RTC) wird diese ebenfalls verändert.

Kommandoaufbau `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 eingeben:

Systemzeit: 01.23.17,98

Neue Zeit (hh.mm.ss) 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 Computers neu. Sie wird in einem Bereich gespeichert und weitergezählt.

TREE *externer Befehl*

Zweck TREE zeigt das strukturierte Verzeichnis einer Diskette oder Festplatte an (wahlweise mit oder ohne die gespeicherten Dateinamen).

Kommandoaufruf (d:)(weg)tree (d:)(/x)

/x /f Zu den Namen der einzelnen 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 (durch den Filter MORE).

```
tree /f > prn
```

Es werden alle Verzeichnisse und die darin enthaltenen Dateinamen ausgedruckt.

Beispiel für eine Ausgabe mit dem Parameter /F:

VERZEICHNISPFAD FÜR DISKETTE/PLATTE BUCH

```
Datei(en):      TREE.COM
                SELECT.COM
                MODE.COM
                SORT.EXE
```

Pfad: \DOS

Unterverzeichnis(se): Keine(s)

```
Datei(en):      ASSIGN.COM
                CHKDSK.COM
```

Pfad: \AUFTRAG

Unterverzeichnis(se): FIBU

```
Datei(en):      Keine
```

Pfad: \AUFTRAG\FIBU

Unterverzeichnis(se): Keine(s)

```
Datei(en):      FIBU.EXE
                JOURNAL.DAT
                OP.DAT
```

TYPE *interner Befehl*

Zweck Den Inhalt einer Datei am Bildschirm anzeigen. Mit *Strg+S* kann mit der Anzeige jederzeit gestoppt bzw. fortgefahren werden. Durch die Betätigung von *Strg+Druck* wird die gleichzeitige Ausgabe an den Drucker eingeschaltet (Abschaltung durch erneute Betätigung).

Kommandoauflbau `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 a:text | more`

Der Inhalt der Datei A:TEXT wird angezeigt (nach jeder vollen Bildschirmseite wartet die Anzeige, bis Sie eine beliebige Taste betätigt haben).

`type c:\autoexec.bat > prn`

Der Inhalt der Datei AUTOEXEC.BAT im Root-Directory auf Laufwerk C wird ausgedruckt.

Bemerkungen Statt des Befehls

`type autoexec.bat`

kann alternativ

`copy autoexec.bat con`

verwendet werden (siehe Befehl COPY).

Statt des Befehls

`type autoexec.bat > prn`

oder der Betätigung von *Strg+Druck* statt `> PRN` für die Druckausgabe können Sie den Befehl

`copy autoexec.bat prn`

verwenden (siehe Befehl COPY).

VER *interner Befehl*

Zweck Die Versionsnummer von DOS anzeigen.

Kommandoaufbau ver

Beispiel ver

Es erscheint z.B.:

IBM Personal Computer DOS-Version 3.30

VERIFY *interner Befehl*

Zweck Wenn mit diesem Befehl »Verify« eingeschaltet wird, erfolgt eine Überprüfung aller Disketten und Festplatten-Schreiboperationen auf korrekte Datenaufzeichnung bzw. -übertragung. Wenn die Funktion eingeschaltet ist, wird nach jedem Schreiben überprüft, ob die Daten problemlos von DOS wieder gelesen werden können. Dies führt zu einer Verlangsamung Ihres Computers bei jeder Schreiboperation. Standardmäßig nach dem Systemstart ausgeschaltet.

Kommandoaufbau verify (on)
verify (off)

Beispiele verify

Es wird der momentane Stand angezeigt (ob ein- oder ausgeschaltet):

VERIFY ist OFF

verify on

Prüfen einschalten.

set verify

Die Überprüfung kann auch mit diesem Befehl abgeschaltet werden, da VERIFY den Environment-Bereich benutzt. Ebenso kann für das Einschalten der Befehl SET verwendet werden:

set verify=on

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. VERIFY wird **nicht** durchgeführt, wenn Daten auf die Festplatte eines Netzwerks geschrieben werden.

VOL *interner Befehl*

Zweck Der Name einer Diskette oder Festplatte wird angezeigt (Label).
Der Name kann mit dem Befehl LABEL verändert oder gelöscht werden.
Beachten Sie beim Befehl FORMAT den Parameter /V für die Eingabe eines Namens für Diskette/Festplatte.

Kommandoaufbau vol (d:)

Beispiel vol a:

Der Name der Diskette in Laufwerk A wird am Bildschirm angezeigt. Beispiel:

Diskette/Platte, Laufwerk A:, hat keinen
Namen

oder

Diskette/Platte, Laufwerk A:, hat den
Namen DOS

Bemerkung Ohne Angabe einer Laufwerksbezeichnung wird der Name des aktuellen Laufwerks angezeigt.

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 oder neu hinzugekommen sind (Dateiänderungsattribut = 1).
/d:tt.mm.jj	Kopiert werden nur die Dateien, die das angegebene oder ein späteres Datum besitzen.
/e	Kopiert leere Sub-Directories wie auch solche, die Dateien beinhalten (zusätzlich muß dazu auch /S angegeben werden).
/m	Kopiert alle Dateien, die seit dem letzten Kopiervorgang verändert wurden (das entsprechende Kennzeichen wird bei der Quelldatei nach dem Kopieren zurückgesetzt) – siehe auch BACKUP.
/p	Zu jeder Datei werden Sie vorher gefragt, ob diese kopiert werden soll (dazu werden der Dateiname und die Directory-Bezeichnung angezeigt)
/s	Kopiert die aktuelle bzw. angegebene und alle sich darunter befindlichen Sub-Directories (ohne diesen Parameter wird nur das angegebene bzw. aktuelle kopiert); angenommen sind leere Directories, wenn kein /E zusätzlich angegeben wurde.
/v	Nach dem Kopieren wird die Kopie auf ordnungsgemäße Speicherung hin überprüft (benötigt zusätzliche Zeit).
/w	Bevor der eigentliche Kopiervorgang beginnt, muß noch eine beliebige Taste betätigt werden (Sie haben die Möglichkeit, vorher noch die Diskette zu wechseln):

Zum Kopieren von Datei(en) eine beliebige Taste betätigen

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 gefragt, ob dies auch wirklich durchgeführt werden soll (/P). Jede kopierte Datei wird, bevor der Kopiervorgang fortgesetzt wird, auf ordnungsgemäße Speicherung hin überprüft (/V).

xcopy a:*.* c:\kop /d:01.10.87

Alle Dateien des Root-Directorys von Laufwerk A werden in das Sub-Directory **KOP** in Laufwerk C kopiert (ohne eventuell weiter vorhandene Sub-Directories in Laufwerk A). Dabei werden nur Dateien berücksichtigt, die als Datum den 1.10.87 oder danach besitzen.

xcopy c:\sub a:\ /s

Die sich hierarchisch unter dem Sub-Directory SUB befindlichen Directories werden mit allen darin enthaltenen Dateien kopiert (/S). Eventuell vorhandene leere Sub-Directories werden dabei nicht berücksichtigt (dazu ist zusätzlich der Parameter /E notwendig). Das Sub-Directory SUB wird auf Laufwerk A nicht angelegt, sondern deren Dateien in das Root-Directory kopiert.

xcopy a:\sub c:\aend /s

Der Kopiervorgang findet hier in umgekehrter Richtung wie im vorhergehenden Beispiel statt, nur daß jetzt auch das Sub-Directory SUB in das neue AEND kopiert wird (Umbenennen eines Sub-Directorys während des Kopierens).

xcopy b:\ b:\sub /s

Der Befehl kann nicht ausgeführt werden, und Sie erhalten die Fehlermeldung:

Rekursive Kopie nicht möglich

Bemerkungen Es gibt folgende Fehlercodes für die Verwendung bei IF ERROR-LEVEL:

- 0 Keine Fehler
- 1 Keine Datei wurde kopiert
- 2 Befehl durch *Strg+Untbr* bzw. *Strg+C* abgebrochen
- 4 Der angegebene Weg bzw. das Sub-Directory wurde nicht gefunden
- 5 Nach Diskettenfehler abgebrochen

Bei XCOPY können Sie nicht mit Gerätenamen (Kapitel 10) arbeiten.

Für die Angabe von Dateinamen gilt das gleiche wie bei COPY. Eine Datei, die größer ist als die Empfangsdiskette/-festplatte, kann nicht mit XCOPY kopiert werden (dies ist generell nur als Datensicherung mit BACKUP möglich).

Auf der Zieldiskette/Festplatte vorher eventuell nicht vorhandene Sub-Directories werden automatisch angelegt.

XCOPY sollte nicht verwendet werden, wenn APPEND/X aktiv ist.

Dateien, auf die kein Lesezugriff besteht, können nicht kopiert werden, »Nur-Lese«-Dateien gleichen Namens auf der Zieldiskette/Festplatte können mit einer zu kopierenden Datei nicht überschrieben werden (Befehl ATTRIB beachten).

Kapitel 12

Die BATCH-Befehle und ihre Anwendung

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 würden. Damit ist es nicht erforderlich, wiederkehrende Routinen bei Bedarf neu zu schreiben.

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. Ferner darf auch kein interner Befehl gleichen Namens existieren.

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 gefragt, und es erscheint anschließend die »Copyright-Meldung« mit der DOS-Versionsnummer.

Unterbrechen einer BATCH-Datei

Der Ablauf einer BATCH-Datei (bzw. Programm) kann jederzeit durch Betätigung von *Strg+Untbr* oder *Strg+C* abgebrochen werden. Sie erhalten die Frage:

Stapeljob 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:

Diskette mit der Stapeldatei einlegen,
anschließend eine Taste betätigen

am Bildschirm. Legen Sie dann die Diskette mit der BATCH-Datei wieder ein, und betätigen Sie eine beliebige Taste.

Kommandoaufbau (d:)(weg)dateiname (variable1 (variable2 (...)))
dateiname BATCH-Datei mit File-Typ BAT (der Name 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.

Wenn Sie das Zeichen % in einem Dateinamen verwenden, so muß, wenn dieser Name innerhalb einer BATCH-Datei angegeben wird, statt %, %% geschrieben werden.

Beispiele Erstellen Sie bitte folgende BATCH-Datei (mit dem Befehl COPY oder EDLIN):

```
edlin autoexec.bat
```

oder

```
copy con autoexec.bat
@echo off
date
time
keyb gr 437,c:\dos\keyboard.sys
cls
```

(F6 und Return)

BATCH-Datei testen Starten Sie Ihren Computer neu, indem Sie die Tastenkombination (gleichzeitig drücken) *Strg+Alt+Entf* betätigen. Anschließend erscheint am Bildschirm:

```
Systemdatum: So. 26.10.1987
Neues Datum (tt.mm.jj) eingeben:
Systemzeit: 21.48.38,68
Neue Zeit (hh.mm.ss) eingeben:
```

Anschließend wird der deutsche Zeichensatz geladen und der Bildschirm gelöscht.

Durch die Angabe von @ vor einem Befehl (hier ECHO OFF) wird dieser bei der Ausführung einer BATCH-Datei nicht am Bildschirm angezeigt. Die Angabe von @ wirkt für einen einzelnen Befehl genauso wie ECHO OFF für alle folgenden Befehle.

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 Batchdatei UEB.BAT
type %0.bat
```

*Beispiele für den Aufruf
von UEB.BAT*

```
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 am Bildschirm:

```
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
Weiter -> eine Taste betätigen...
```

```
A>type DISKETTE.neu
```

Es wird der Inhalt der Datei DISKETTE.NEU am Bildschirm angezeigt

```
A>rem dies ist die Batchdatei UEB.BAT
```

```
A>type UEB.bat
```

Es wird der Inhalt der Datei UEB.BAT am Bildschirm angezeigt

```
ueb text
```

Diese Eingaben führen zu folgenden Befehlsausgaben der BATCH-Datei:

```
A>copy text a:.neu
      1 Datei(en) kopiert
```

```
A>type text
```

Anzeige der Datei TEXT

```
A>pause
```

Weiter -> eine Taste betätigen...

```
A>type .neu
```

Anzeige der Datei .NEU

```
A>rem Dies ist die Batchdatei UEB.BAT
```

```
A>type ueb.bat
```

Anzeige der Batchdatei UEB.BAT

*Anzeige der BATCH-
Datei UEB.BAT*

Geben Sie nur den File-Namen der BATCH-Datei (UEB) ein, so führt dies zu einer fehlerhaften Ausführung der Befehle. Die BATCH-Datei für die Überprüfung einer Datei auf Vorhandensein, 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
keyb gr 437,c:\dos\keyboard.sys
date
time
path=\dos
cls
prompt $e[s$e[1;1HAktuelles Laufwerk/Directory:
$P$e[K$e[u-echo .
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 eine 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
```

*Beispiele für den Aufruf
von LOE.BAT*

```
loe niemals
A> if "NIEMALS"==" " goto ende
A> if "NIEMALS"==" " goto ausgang
A> if not exist NIEMALS goto nicht
A> echo Datei NIEMALS nicht vorhanden
Datei NIEMALS nicht vorhanden
A> goto aus
A> shift
A> goto weiter
A> if ""==" " goto ausgang
```

*Zweites Beispiel für den
Aufruf von LOE.BAT*

```
loe
A> if ""==" " goto ende
A> echo Datei nicht angegeben
Datei nicht angegeben
A> goto ausgang
```

*Drittes Beispiel zum
Aufruf von LOE.BAT
(angenommen ist, daß
TEXT existiert)*

```
loe text
A> if "TEXT"==" " goto ende
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> shift
A> goto 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. Wenn auch ECHO OFF nicht angezeigt werden soll, so geben Sie davor das Zeichen @ an. Es erfolgt keine Anzeige der Anspringpunkte (z.B. :weiter) bei der Ausführung. Für die Programmauswahl (Menüprogramm) beim Start des Computers kann ein einfaches Programm gewählt werden (AUTOEXEC.BAT):

*Beispiel eines Menü-
Programms*

```
:Author Hans H. Gerhardt
:start
@echo off
rem Programm START.BAT
prompt $e[s$e[1;60H$p$e[K$e[u
cls
echo -----
echo
echo 1 = DOS
echo 2 = Textprogramm
echo 3 = Finanzbuchhaltung
echo
echo Wählen Sie 1, 2 oder 3
echo -----
:schleife
eingabe
if errorlevel 51 goto drei
if errorlevel 50 goto zwei
if errorlevel 49 goto eins
```

```
goto schleife
:eins
cls
echo Programm beendet
goto ende
:zwei
call text
goto start
:drei
call fibu
goto start
:ende
```

Programm für den Start des Textprogrammes:

```
rem Programm TEXT.BAT
if not exist \text\ws.com goto nicht
cd text
ws
cd ..
goto ende
:nicht
echo Das Textprogramm ist nicht vorhanden!
pause
:ende
```

Programm für den Start der Finanzbuchhaltung:

```
rem Programm FIBU.BAT
if not exist \fibu\buchh.exe goto nicht
cd fibu
buchh
cd ..
goto ende
:nicht
echo Das Finanzbuchhaltungsprogramm ist nicht
vorhanden!
pause
:ende
```

Mit dem Programm EINGABE kann ein von Ihnen eingegebener Wert in den Speicherbereich geschrieben werden, den Sie anschließend mit **ERRORLEVEL** lesen und verarbeiten können.

Das Programm EINGABE kann mit dem Befehl DEBUG erstellt werden (siehe Befehl DEBUG im Kapitel 11).

Wichtig ist dabei, daß die Werte bei ERRORLEVEL in absteigender Reihenfolge benutzt werden. Die Dezimalwerte für die verschiedenen Ziffern, Buchstaben usw. finden Sie im Anhang IV. WS.COM und BUCHH.EXE sind nur Beispiele.

Erstellung des Programmes EINGABE.COM:

```
debug
-neingabe.com
-a
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
-
```

Mit diesen 4 Maschinensprache-Befehlen erreichen Sie, daß bei Aufruf von EINGABE der Cursor am Bildschirm anschließend abwartend steht. Tippen Sie ein Zeichen ein, so steht dieses anschließend im Register AH. Dieses Register fragt der BATCH-Befehl IF ERRORLEVEL ab.

Das Programm muß von Ihnen noch wie folgt weiterbearbeitet werden:

```
-rcx
:0000
9                                Return
-w                                Return
Schreibe 0009 Bytes
-q
```

Beispiel einer BATCH-Datei zur Umbelegung der Funktionstasten F7 bis F10 (beachten Sie hierzu auch den Befehl PROMPT im Kapitel 11):

<i>BATCH-Datei zur</i>	@echo off
<i>Funktionstasten-</i>	cls
<i>Belegung</i>	rem TASTEN.BAT

```

echo Neubelegung der Funktionstasten F7 bis F10
if %1==ein goto an
if %1==EIN goto an
if %1==aus goto aus
if %1==AUS goto aus
goto ende
:an
@echo on
prompt=$e[0;65;"dir a: ";13p
prompt=$e[0;66;"dir b: ";13p
prompt=$e[0;67;"type "p
prompt=$e[0;68;"copy "p
prompt
goto end
:aus
prompt=$e[0;65;0;65p
prompt=$e[0;66;0;66p
prompt=$e[0;67;0;67p
prompt=$e[0;68;0;68p
:ende
prompt

```

Die BATCH-Datei wird mit den Parametern EIN oder AUS aufgerufen. In Abhängigkeit davon erhalten die Funktionstasten F7 bis F10 neue Werte bzw. wird die Standardfunktion wieder eingeschaltet.

Das vorliegende Beispiel funktioniert nur, wenn sich beim Start Ihres Computers die Datei CONFIG.SYS (siehe Kapitel 14) im Root-Directory mit dem Eintrag

```
device=ansi.sys
```

befindet.

Weitere Beispiele finden Sie im Kapitel 11 bei den einzelnen Befehlen – z.B. bei

CALL	IF
ECHO	PAUSE
FOR	REM
GOTO	SHIFT

Kapitel 13

Bearbeitung der Zeichensätze für Drucker und Bildschirm

Zeichensatztabellen wechseln

Mit DOS 3.30 besteht die Möglichkeit, Zeichensatztabellen zu wechseln bzw. zu aktivieren. Damit werden Zeichen für die Angabe an Drucker und Bildschirm definiert. Im vorliegenden Buch wird in diesem Zusammenhang nur PC-DOS 3.30 von IBM berücksichtigt. Bei anderen Herstellern stehen teilweise andere Zeichensatztabellen zur Verfügung (beachten Sie hierzu das DOS-Handbuch zu Ihrem System).

Die einzelnen Zeichensatztabellen finden Sie im Anhang IV.

Mehrsprachige Zeichensatztabelle

Ab DOS 3.30 gibt es eine mehrsprachige Zeichensatztabelle (850), die für alle Länder verwendet werden kann. Die in den bisherigen DOS-Versionen verwendete Zeichensatztabelle besitzt die Nummer 437. Diese kann selbstverständlich für die USA und auch für Deutschland verwendet werden. Nur für Portugal, Norwegen/Dänemark und Kanada (frz.) gibt es separate Tabellen (860, 865 und 863). Frühere DOS-Versionen haben nur 5 unterstützt, die neue Version jedoch 11.

Sollten Sie bei Benutzung einer neuen Zeichensatztabelle mit Ihren Programmen Darstellungsprobleme haben, so müssen Sie die Zeichensatztabelle 437 aktivieren bzw. in CONFIG.SYS und AUTOEXEC.BAT einsetzen.

Land	Tastaturcode	Zeichensatztabelle
Australien	US	850 oder 437
Belgien	BE	850
Dänemark	DK	850 oder 865
Deutschland	GR	850 oder 437
Finnland	SU	850 oder 437
Frankreich	FR	850 oder 437
Großbritannien	UK	850 oder 437
Italien	IT	850 oder 437

Land	Tastaturcode	Zeichensatztabelle
Kanada (engl.)	US	850 oder 437
Kanada (frz.)	CF	850 oder 863
Lateinamerika	LA	850 oder 437
Niederlande	NL	850 oder 437
Norwegen	NO	850 oder 865
Portugal	PO	850 oder 860
Schweden	SV	850 oder 437
Schweiz (deutsch)	SG	850
Schweiz (frz.)	SF	850
Spanien	SP	850 oder 437
USA	US	850 oder 437

In der CONFIG.SYS müssen mit DEVICE (Kapitel 14) Bildschirm und Drucker beschrieben werden.

```
DEVICE=C:\DOS\DISPLAY.SYS CON:=(EGA,437,1)
```

In AUTOEXEC.BAT (z.B.) können weitere Befehle für die Zeichensatzbearbeitung aufgenommen werden.

Die Beschreibung zu den einzelnen Befehlen für die Zeichensatztabellenbearbeitung finden Sie im Kapitel 11.

Befehle zur Zeichensatz- tabellenbearbeitung

Befehl	Funktion
CHCP	Zeichensatztabelle für alle Geräte/Einheiten auswählen.
GRAFTABL	Tabelle mit zusätzlichen Zeichen für den Grafikmodus laden (ASCII-Zeichen 128 bis 255).
KEYB	Tastaturzeichentabelle laden (Tastaturanpassung).
MODE	Vorbereiten von Zeichensatztabellen Auswählen/Aktivieren einer Zeichensatztabelle Anzeige der aktiven Zeichensatztabelle Reaktivieren einer Zeichensatztabelle
NLSFUNC	Laden des erweiterten landesspezifischen Unterstützungs-codes. Ferner wird damit die Verwendung des Befehls CHCP ermöglicht.
SELECT	Mit diesem Befehl wird bei der Installation einer Diskette/Festplatte eine Tastaturanpassung vorgenommen und die angegebene Zeichensatztabelle installiert.

Kapitel 14 Die Konfiguration von DOS (CONFIG.SYS)

*Schnittstelle zu
Anwendungsprogrammen*

Die Datei IBMDOS.COM ist der »Direktor« Ihres Computers. Sie erzeugt die Instruktionen für die zu erledigenden Arbeiten. Die Vorgänge laufen hardwareunabhängig ab (der hardwareabhängige Teil zur Bedienung der Peripherie ist IBMBIO.COM).

IBMDOS.COM ist ein intelligentes Interface für Anwendungsprogramme. Es verfügt über Programmteile zur Dateiverwaltung, Steuerung von Informationen bei Disketten-/Festplattenzugriff usw. Es können Aufrufe der Anwenderprogramme erkannt und bei Bedarf an IBMBIO.COM zur Ausführung weitergeleitet werden.

*IBMDOS.COM anpassen
mit CONFIG.SYS*

IBMDOS.COM können Sie an Ihre Anforderungen anpassen (erweitern). Dazu wird die Datei CONFIG.SYS benötigt. Es handelt sich um eine Textdatei (vergleichbar mit AUTOEXEC.BAT), die Sie mit EDLIN, COPY oder einem Textverarbeitungsprogramm erstellen. Haben Sie Ihr System mit dem Befehl SELECT installiert, so befindet sich bereits eine CONFIG.SYS-Datei auf Ihrer Festplatte/Diskette.

Während des Ladevorganges von DOS wird nach der Datei CONFIG.SYS gesucht. Wenn diese gefunden wird, werden die darin enthaltenen Befehle ausgeführt.

Startvorgang von DOS

Der Startvorgang Ihres Computers in Einzelschritten (vereinfachte Darstellung):

- 1 Lesen des ersten Sektors von der Diskette/Festplatte
- 2 Laden des BOOT-LOADER und starten
- 3 Der BOOT-LOADER lädt IBMBIO.COM und übergibt die Kontrolle
- 4 IBMBIO.COM initialisiert die Peripherie und lädt IBMDOS.COM

- 5 IBMBIO.COM übergibt die Kontrolle an das Modul SYSINIT (stellt die Verbindung zwischen IBMBIO.COM und IBMDOS.COM her), das mit IBMBIO.COM 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 Befehlsinterpreter)
- 8 COMMAND.COM sucht AUTOEXEC.BAT und führt die darin enthaltenen Befehle aus (erst ab hier kann der Startvorgang durch *Strg* + *Untbr* abgebrochen werden)

Im folgenden finden Sie die CONFIG.SYS-Befehle.

BREAK = on

BREAK = off

Programmunterbrechung

Dieser Befehl modifiziert die Programmunterbrechung durch die Tastenkombination *Strg* + *Untbr*. 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 *Strg* + *Untbr* unterbrochen. Ohne Modifizierung ist die Funktion ausgeschaltet.

Diese Funktion wird von vielen Anwendungsprogrammen »blockiert«. Das heißt, unabhängig von dem eingestellten Wert kann ein derartiges Programm nicht abgebrochen werden.

BUFFERS = Nummer

Zwischenspeicher

Bei der **Nummer** handelt es sich um eine dezimale Zahl zwischen 1 und 99, die Anzahl der Zwischenspeicher festlegt, die im RAM zusätzlich für Daten reserviert werden. Dadurch wird vielfach der Programmablauf schneller.

Standardwerte, wenn BUFFERS nicht angegeben wird:

- 3 wenn ein Diskettenlaufwerk größer als 360 Kbyte ist
- 5 wenn der Speicher (RAM) größer als 128 Kbyte ist
- 10 wenn der Speicher (RAM) größer als 256 Kbyte ist
- 15 wenn der Speicher (RAM) größer als 512 Kbyte ist
- 2 in allen anderen Fällen

Jeder Zwischenspeicher (Puffer) belegt 528 Zeichen (Byte) des Arbeitsspeichers. Vereinzelt benötigen Anwenderprogramme mit BUFFERS eine Mindestspeichergröße. In diesem Fall finden Sie Angaben dazu in den Unterlagen zu Ihren Programmen.

Die Puffer werden vom 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.

Wenn Sie mit einer Datenbank arbeiten, führt in der Regel ein Wert zwischen 10 und 25 zu den besten Ergebnissen.

Beispiel buffers = 15

COUNTRY = xxx;(yyy)((d:)dateiname)

Landeskennziffer

Bei **xxx** handelt es sich um eine dezimale Zahl (internationale Telefonvorwahl) zwischen 1 und 999, die landesspezifischen Datensatz bestimmt, der nachgeladen werden kann. Dieser enthält Angaben über das Format für die Datumsanzeige, Uhrzeit, Sortierfolgen, Großschreibung und Zeichenumsetzung. Ebenso sind das Dezimalzeichen, das Währungssymbol und weitere Angaben für das Formatieren von Ausgaben auf dem Bildschirm oder Drucker gespeichert. Damit kann das Format für Uhrzeit und Datum auf das gewohnte Aussehen verändert werden. **yyy** ist die Nummer der gewünschten Zeichensatztabelle. Ist das gewünschte Land nicht vorhanden, müssen Sie die Kennziffer des ähnlichsten Landes verwenden.

Der Dateiname gibt an, in welcher Datei die Informationen zum Befehl COUNTRY gespeichert sind. Ohne den Befehl COUNTRY werden der Landescode 001 und die Zeichensatztabelle 437 verwendet.

Die Angabe COUNTRY 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 (KEYB).

Land	Landescode	Tastaturcode
Arabischer Sprachraum	785	nicht vorhanden
Australien	061	US
Belgien	032	BE
Dänemark	045	DK
Deutschland	049	GR
Finnland	358	SU
Frankreich	033	FR
Großbritannien	044	UK
Hebräischer Sprachraum	972	nicht vorhanden
Italien	039	IT
Kanada (engl.)	001	US
Kanada (frz.)	002	CF
Lateinamerika	003	LA
Niederlande	031	NL
Norwegen	047	NO
Portugal	351	PO
Schweden	046	SV
Schweiz (deutsch)	041	SG
Schweiz (frz.)	041	SF
Spanien	034	SP
USA	001	US

Beispiel country=049,437,c:\dos\country.sys
oder

country=049,,c:\dos\country.sys

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. Daher sollten Sie diesen Befehl nur verwenden, wenn dies 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.

*Vorsicht bei der
Eingabe*

Versucht ein Anwendungsprogramm, mehr FCBs zu öffnen, als mit Nummer1 bestimmt wurde, schließt DOS automatisch den am längsten nicht mehr benutzten geöffneten FCB. Dies ist erforderlich, da der benutzte Speicherplatz benötigt wird. Mit dem Wert Nummer2 bestimmen Sie die Zahl der FCBs, die unbedingt offengehalten werden müssen.

Ohne Eingabe des Befehls wird Platz für vier FCBs reserviert. Davon ist kein FCB 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 mehr als ein FCB auf dieselbe Datei, zählt DOS dies als einen FCB. Sind die Werte für Nummer1 und Nummer2 gleich, kann 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 DOS einen alten FCB wieder. Dies können Sie verhindern, indem Sie mit Nummer2 eine bestimmte Anzahl von FCBs schützen. DOS erkennt diese FCBs dann nicht mehr als zuletzt geöffnet (nur die darüber hinausgehenden merkt sich DOS zu diesem Zweck).

Versucht Ihr Programm, eine Datei zu bearbeiten, dessen FCB zwangsweise geschlossen wurde, erscheint die Fehlermeldung:

Dateisteuerblock (FCB) nicht verfügbar

A(bbruch), W(iederholen), I(gnорieren), U(ebergangen)?

Benutzen Ihre Programme keine FCBs, sondern Dateinummern, benötigen Sie diesen Befehl nicht. Beachten Sie dazu auch den CONFIG-Befehl FILES.

FILES = Nummer

*Maximale Anzahl gleich-
zeitig geöffneter Dateien*

Bei der Nummer handelt es sich um eine dezimale Zahl zwischen 8 und 255. Sie legt die Anzahl der gleichzeitig benutzbaren bzw. geöffneten Dateien fest.

Geben Sie den Befehl nicht ein, legt DOS den Wert mit 8 fest. Eine Zahl kleiner als 8 wird nicht beachtet. Ein Programm kann maximal mit **20** gleichzeitig geöffneten Dateien arbeiten. Die Höchstzahl an Dateinummern für jeden Verarbeitungsvorgang ist 65534.

Ein- und Ausgabeoperationen

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.

Eine geöffnete Datei stellt auch ein gestartetes Programm dar. Auch ein installiertes Netzwerk oder ein Hintergrundprogramm wie PRINT benötigt 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 ein Programm nicht korrekt ablaufen.

Jede über die Standardanzahl (8) hinzugefügte Datei benötigt weitere 48 Zeichen (Byte) im Speicher.

Von DOS standardmäßig genutzte Dateien*Genutzte Dateien von DOS*

Nummer	Name	Funktion	Bemerkungen
0000	STDIN	Standardeingabe	kann umgeleitet werden
0001	STDOUT	Standardausgabe	kann umgeleitet werden
0002	STDERR	Fehlerrausgabe	kann nicht umgeleitet werden
0003	STDAUX	Ausgabe auf Modem	kann nicht umgeleitet werden
0004	STDPRN	Ausgabe auf Drucker	kann nicht umgeleitet werden

Standardausgabe ist normalerweise der Bildschirm. Diese Ausgabe kann von Ihnen z.B. auf den Drucker umgeleitet werden:

```
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 DOS Laufwerksbezeichnungen mit einem Buchstaben vorgenommen werden, bestimmt die Eingabe des Buchstabens das letzte gültige Laufwerk.

Ohne Eingabe des Befehls ist E das letzte logisch gültige Laufwerk. Bestimmen Sie eine kleinere Laufwerksangabe als physikalisch 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\abt1
```

Mit dieser Angabe kann das Sub-Directory ABT1 auf dem physikalischen Laufwerk C mit der Laufwerksbezeichnung W angesprochen werden.

SHELL = Dateiname

*Name und Directory
eines Kommando-
interpreters*

Dieser Befehl veranlaßt DOS, einen Kommandointerpreter zu laden, wie hinter SHELL spezifiziert (statt normalerweise COMMAND.COM im Root-Directory).

Dies ermöglicht einem Systemprogrammierer, einen eigenen Kommandointerpreter zu erstellen und zu verwenden.

Da die Angabe bei SHELL den Namen nur für das Laden des Kommandointerpreters ändert, ist noch folgende Angabe nach dem Start Ihres Systems notwendig (am besten wird dies in AUTOEXEC.BAT als erster Befehl eingefügt):

```
set comspec=d:\dateiname
```

Diese Angaben gelten nur für einen Kommandointerpreter der höchsten Ebene, beim Systemstart wird immer der COMMAND.COM geladen. Der neue Kommandointerpreter kann nur als Sekundärprozeß nachgeladen werden.

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 = C:\SYSTEM\COMMAND.COM
```

Beim Start Ihres Systems wird COMMAND.COM vom Laufwerk C aus dem Sub-Directory SYSTEM geladen. Zusätzlich ist nach dem Start noch folgender Befehl erforderlich:

```
set COMSPEC=C:\SYSTEM\COMMAND.COM
```

Hinter COMMAND.COM können zusätzlich Parameter angegeben werden (/P usw.).

Beachten Sie hierzu auch die Angaben zum Befehl COMMAND im Kapitel 11.

STACKS = nummer1,nummer2

*Überschreibt die
Standardwerte
für Stacks*

nummer1 definiert die Anzahl der Stacks; Werte zwischen 8 und 64 sind möglich.

nummer2 definiert die Größe eines Stacks (in Byte); Werte zwischen 32 und 512 sind möglich.

Ein Sonderfall liegt vor, wenn beide Werte als 0 angegeben werden. In diesem Fall soll STACKS nicht unterstützt werden (keine dynamischen Stapel).

Dieser Eintrag in CONFIG.SYS ist nur notwendig, wenn dieser in einem Handbuch zu einem Programm gefordert wird. Ferner, wenn Sie eine Fehlermeldung erhalten, daß der Stack-Bereich zu klein ist.

Standardwerte für PC, XT und Portable PC:

nummer1 = 0

nummer2 = 0

Standardwerte für alle anderen Systeme:

nummer1 = 9

nummer2 = 128

DEVICE = (d:)(weg)dateiname

Einheitentreiber

Dieser Befehl ruft eine Datei (Device Driver) auf, die als Erweiterung von IBMDOS.COM geladen wird. Diese Datei enthält das Steuerprogramm (Device Driver) für ein Peripheriegerät (Device). Damit wird der Datenfluß zwischen der Zentraleinheit und dem Gerät überwacht und falls erforderlich entsprechend verändert.

Die von DOS standardmäßig geladenen Einheitentreiber unterstützen:

- Standardeingabeeinheit
- Standardausgabeeinheit
- Standarddrucker
- Diskettenlaufwerke
- Festplattenlaufwerke
- Systemuhr

Folgende Einheitentreiber erhalten Sie mit Ihrer DOS-Systemdiskette:

ANSI.SYS	Für erweiterte Standardein- und -ausgabe.
DISPLAY.SYS	Für Bildschirm (damit kann für CON die Zeichensatztafel gewechselt werden).
DRIVER.SYS	Blockeinheitentreiber.
PRINTER.SYS	Für Drucker (damit können für PRN, LPT1, LPT2 und LPT3 die Zeichensatztabellen gewechselt werden).
VDISK.SYS	Für virtuelle Laufwerke (RAM-Diskette)

Zusätzlich wird zu verschiedenen Produkten ebenfalls ein Treiber auf Diskette mitgeliefert (z.B. zur MOUSE).

ANSI.SYS

Mit der Datei ANSI.SYS wird die standardmäßig vorhandene Eingabe- und Ausgaberoutine des DOS ersetzt

Damit erhalten Sie eine wesentlich erweiterte Bildschirm- und Tastatursteuerungsmöglichkeit. Die Angabe

```
device = (d:)(weg)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 generell 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.

Alle 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.

Beispiele zur Verwendung finden Sie z.B. im Kapitel 11 zum Befehl PROMPT.

Im folgenden finden Sie die einzelnen Funktionen der Norm, die in ANSI.SYS vorhanden sind.

<i>Cursorposition</i> (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 1, wenn die Variablen nicht angegeben werden; der Cursor wird damit in die linke obere Ecke des Bildschirms gesetzt (Home Position).
<i>Cursor positionieren</i> (HVP)	ESC [# 1; # 2f 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 1, wenn die Variablen nicht angegeben werden; der Cursor wird damit in die linke obere Ecke des Bildschirms gesetzt (Home Position). Dieser Befehl entspricht CUP.
<i>Cursor nach oben</i> (CUU)	ESC [# A Der Befehl CUU bewegt den Cursor eine Zeile nach oben, ohne die Spalte zu verlassen. 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.
<i>Cursor nach unten</i> (CUD)	ESC [# B Der Befehl CUD bewegt den Cursor eine Zeile nach unten, ohne die Spalte zu verlassen. 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.
<i>Cursor nach rechts</i> (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.

<i>Cursor nach links</i> (CUB)	ESC [#D Der Befehl CUB bewegt den Cursor nach links, ohne die Zeilenposition zu verlassen. 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.
<i>Cursorposition abfragen</i> (DSR)	ESC [6n Der Befehl DSR fragt die momentane Cursorposition am Bildschirm ab. Die Ausgabe erfolgt dadurch automatisch mit dem Befehl CPR (Cursorposition anzeigen). Diese Funktion sollte nicht mit dem Befehl PROMPT benutzt werden.
<i>Cursorposition anzeigen</i> (CPR)	ESC [# ; #R Dieser Befehl zeigt die momentane Cursorposition an. #1 bezeichnet die Zeile, #2 die Spalte.
<i>Cursorposition speichern</i> (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.
<i>Cursor an die gespeicherte Position setzen</i> (RCP)	ESC [u Der Befehl RCP setzt den Cursor an die Stelle zurück, an der er war, als der Befehl SCP eingegeben wurde.
<i>Bildschirm löschen</i> (ED)	ESC [2J Der Befehl ED löscht den gesamten Bildschirm und setzt den Cursor in die linke obere Ecke (entspricht dem internen DOS-Befehl CLS).
<i>Zeile löschen</i> (EL)	ESC [K Der Befehl EL löscht ab der momentanen Cursorposition die gesamte restliche Zeile einschließlich der Stelle, an der der Cursor steht.
<i>Farbwahl</i> (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. Ein erneuter SGR-Befehl verändert dies wieder.

Wert(#)	Farbe/Funktion
0	normal (schwarz-weiß beim Farbmonitor)
1	intensiv (hellere Darstellungsform)
2	Darstellungsintensität vermindern
3	Kursivschrift an
4	unterstreichen (nur für Monochrome-Monitore)
5	blinkend
6	schnelles Blinken
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
48	Subscript
49	Superscript

Die Werte 30 bis 47 entsprechen der ISO-Norm 6429. Nicht alle obigen Werte werden von allen Monitoren bzw. Videokarten unterstützt.

Modus auswählen ESC [= # h
(SM) ESC [? 7 h

*Der Befehl SM verändert
den Bildschirmmodus*

Modus	Wert(#)	Funktion
Text	0	40 Zeichen/Zeile, 25 Zeilen monochrom
Text	1	40 Zeichen/Zeile, 25 Zeilen Farbbildschirm
Text	2	80 Zeichen/Zeile, 25 Zeilen monochrom
Text	3	80 Zeichen/Zeile, 25 Zeilen Farbbildschirm
Grafik	4	Mittlere Auflösung in Farbe 320 x 200 mit 40 Zeichen/Zeile Textdarstellung

Modus Wert(#) Funktion

Grafik	5	Mittlere Auflösung monochrom 320x200 mit 40 Zeichen/Zeile Textdarstellung
Grafik	6	Hochauflösende Grafik (monochrom) 640 x 200 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 II)

```
prompt $e[0;68;"CLS";13p$n$g
```

Die Funktionstaste *F10* (68) erhält den Inhalt »CLS Return«. Wenn Sie *F10* betätigen, wird der Befehl CLS ausgeführt. Mit *\$n\$g* wird der normale Prompt wieder am Bildschirm dargestellt. Der Wert 13 definiert die *Return*-Taste, der Buchstabe ist der »Befehl«, damit das System erkennt, daß es sich um eine Tastaturänderung handelt.

Als Prompt soll die Anzeige der Uhrzeit und des Datums dienen. Der Cursor soll in der Zeile darunter stehen.

Beispiele Zeit = 11.54.34,56
Datum = 21.09.87

```
—
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. Anschließend 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
```

Dabei wird als erstes die aktuelle Cursorposition gesichert. Anschließend den Cursor links oben am Bildschirm positionieren und das aktuelle Laufwerk/Directory anzeigen. Als letztes wird die anfängliche Cursorposition wieder zurückgesichert. Als Prompt erscheint an der linken Seite am Bildschirm das Zeichen >.

DISPLAY.SYS

*Wechseln der Zeichensatz-
tabelle für Bild-
schirmadapter*

Zeichensatztabellen für den IBM-PC-AP-LCD- und HR-Grafikadapter-Bildschirm können mit MODE vorbereitet werden, wenn der Einheitentreiber DISPLAY.SYS in CONFIG.SYS eingetragen wird.

Sollten Sie zusätzlich ANSI.SYS verwenden, so muß dieser Treiber für die erweiterte Bildschirm- und Tastatursteuerung vorher in CONFIG.SYS eingetragen werden.

Kommandoaufbau

```
device = [d:][weg]display.sys con[:]=(typ[, [hwcp][, n]])
oder
device = [d:][weg]display.sys con[:]=(typ[, [hwcp][, (n.m)]])
typ    Bildschirmapadertyp (MONO, CGA, EGA, LCD)
hwcp   Zeichensatz (437, 850, 860, 863, 865)
```

- n** Anzahl zusätzlicher Tabellen, die unterstützt werden können (Wert zwischen 0 und 12)
- m** Anzahl der verschiedenen Schriftarten, die für jede Zeichensatztafel unterstützt werden können (8x8- oder 8x14-Schriftgröße)

Typ	Einheit	Standardwert für n	n	Standardwert für m	m
MONO	Monochrom-Bildschirmadapter	0	0	0	0
CGA	Farbgrafikadapter	0	0	0	0
EGA	HR-Farbgrafikadapter	1	1-12	2	1-2
LCD	IBM PC AP	1	1-12	1	1

Durch die Angabe von **n** definieren Sie, daß weitere mit dem Befehl **MODE** vorbereitete Zeichensatztabellen benutzt werden können.

VDISK.SYS

Simulation eines Laufwerks im RAM (Arbeitsspeicher)

Die Datei **VDISK.SYS** ist ein funktionaler Blocktreiber, der es ermöglicht, im Arbeitsspeicher ein oder mehrere Laufwerke zu simulieren. Diese können von Ihnen wie jedes andere normale Disketten- oder Festplattenlaufwerk benutzt werden. Es hat den Vorteil, erheblich schneller zu arbeiten. Alle 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 Computers verloren. Der Treiber benötigt 800 Byte Speicher.

Kommandoaufbau

`device = (d:)(weg)vdisk.sys (bem)(b) (bem)(s) (bem)(d) (/e):(m)`

Vor **VDISK.SYS** kann angegeben werden, woher der Treiber für die RAM-Disk geladen wird (Laufwerk und Directory).

bem ist eine Bemerkung (das Zeichen / ist, wie auch Umlaute, nicht erlaubt).

b ist die Größe des Laufwerks in Kilobyte (ohne = 64/Minimum = 1). Sind weniger als 64 Kbyte bei der Installation verfügbar, so wird keine RAM-Disk angelegt. Der restliche verfügbare Speicher muß mindestens 64 Kbyte betragen – im anderen Falle verkleinert der Treiber automatisch Ihren angegebenen Wert.

s ist die Größe eines Sektors in Byte (ohne = 128/möglich sind 128, 256 oder 512). Geben Sie einen falschen Wert an, so wird automatisch der Standardwert 128 benutzt. Kleine Werte sollten

Sie verwenden, wenn Sie mit relativ kleinen Dateien arbeiten. Ein größerer Wert erhöht den Datendurchsatz und damit die Verarbeitungsgeschwindigkeit (dies führt bei kleinen Dateien aber zu einer Speicherplatzverschwendung).

d ist die maximale Anzahl der Einträge in das Root-Directory der RAM-Disk (ohne = 64/Minimum = 2/Maximum = 512). Der angegebene Wert wird an die nächsthöhere Sektorgröße (**s**) angeglichen. Geben Sie beispielsweise 18 an und die Sektorengöße ist 512, ändert der Treiber den Wert auf 32:

18 x 32 Byte (Verzeichniseintragsgröße) =
576 Byte -> 1024 (2 x 512)
1024 : 32 Byte = 32 Verzeichniseinträge

Wurde die Größe der RAM-Disk zu klein angegeben, um die Dateizuordnungstabelle (Directory und zwei zusätzliche Sektoren) aufzunehmen, so wird die Verzeichnisgröße so lange um jeweils einen Sektor herabgesetzt, bis die notwendigen Bedingungen erfüllt sind. Ist dies bei einem Sektor immer noch nicht möglich, so erhalten Sie eine Fehlermeldung beim Start des Computers, und die RAM-Disk wird nicht angelegt.

Ein Verzeichniseintrag wird für den Namen der RAM-Disk benötigt.

/e ist der Parameter, damit die RAM-Disk im Speichererweiterungsbereich über 1 Mbyte verwenden kann (nur bei Systemen mit 80286- oder 80386-Prozessor). Ist in Ihrem System keine Speichererweiterung eingebaut, erhalten Sie eine Fehlermeldung.

m ist die größte Sektorenanzahl (**s**), die gleichzeitig übertragen werden. Der Standardwert ist 8 (erlaubt sind 1 bis 8).

Treten Datenverluste auf, so ist der Wert für **m** bis auf 1 und **s** bis auf 128 herabzusetzen. Ist der Fehler damit immer noch nicht behoben, so kann die RAM-Disk nicht im erweiterten Speicherbereich benutzt werden. Tritt der Fehler auch im normalen Speicher auf, so ist Ihr Computer vermutlich defekt.

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:\dos\vdisk.sys 160 512 64
```

```
device=c:\dos\vdisk.sys 256
```

Es werden 2 RAM-Disks mit 160 Kbyte und 256 Kbyte eingerichtet. Diese Möglichkeit ist vor allem für die Benutzung des erweiterten Speicherbereichs interessant.

Beim Start Ihres Computers erscheinen zum Beispiel folgende Meldungen:

```
VDISK Version 3.30 Virtuelle Platte D:
```

```
Sektorgröße angepaßt
```

```
Verzeichniseinträge angepaßt
```

```
Puffergröße:          64 KB
```

```
Sektorgröße:          128
```

```
Verzeichniseinträge:  64
```

```
VDISK Version 3.30 Virtuelle Platte E:
```

```
Sektorgröße angepaßt
```

```
Verzeichniseinträge angepaßt
```

```
Größe des Transferpuffers angepaßt
```

```
VDISK nicht installiert --
```

```
nicht genug Speicherplatz
```

```
VDISK Version 3.30 Virtuelle Platte E:
```

```
Sektorgröße angepaßt
```

```
Verzeichniseinträge angepaßt
```

```
Ungültiger Parameter
```

```
Puffergröße:          64 KB
```

```
Sektorgröße:          128
```

```
Verzeichniseinträge:  64
```

Die Einträge in CONFIG.SYS zu VDISK für die drei Beispiele sehen wie folgt aus:

```
device=vdisk.sys 64
```

```
device=vdisk.sys 128 /e
```

```
device=vdisk.sys 64 /a
```

Da der Parameter /a nicht bekannt ist, wurde im letzten Beispiel eine Fehlermeldung ausgegeben und die RAM-Disk innerhalb der ersten 640 Kbyte des Speichers eingerichtet.

Bei der Puffergröße handelt es sich um die Größe der RAM-Disk.

Im zweiten Beispiel wurde die RAM-Disk nicht installiert, da der verwendete Computer keine zusätzliche Speichererweiterung besitzt.

Die Laufwerksbezeichnungen beginnen immer ab der nächsten freien Bezeichnung (2 Diskettenlaufwerke = C; mit Festplattenlaufwerk = D; usw.).

Wenn Sie die RAM-Disk benutzen, um ein Laufwerk im Arbeitsspeicher zu simulieren, sollten Sie beachten, daß es sich nur um ein vorübergehendes Laufwerk handelt. Alle dort gespeicherten Dateien werden bei einem Neustart oder dem 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.

Eine sinnvolle Anwendung wäre z.B., ein Programm mit vielen Modulen (einzelnen Programmdateien) dorthin zu kopieren, damit dieses schneller abläuft (den Kopierbefehl können Sie in AUTOEXEC.BAT eintragen).

Daten-Dateien sollten Sie nur in einer RAM-Disk bearbeiten, wenn dies unbedingt notwendig ist oder diese nur unwesentlich verändert werden, da bei Stromausfall oder dem Ausschalten des Systems (ohne vorher die Daten auf Diskette oder Festplatte zu kopieren) alle Änderungen verloren sind!

Beispiel für Speicherplatzberechnung

Mit CHKDSK kann die Speicherplatzbelegung angezeigt werden – hier ohne CONFIG.SYS:

```
654.336 Byte Gesamtspeicher (= 640 Kbyte RAM)
599.040 Byte frei verfügbar
-> 55.296 Byte belegt DOS bereits
```

Wenn Sie CONFIG.SYS anlegen und als erstes ANSI.SYS laden
device=ansi.sys

ergibt sich ein neuer frei verfügbarer Speicherplatz:

```
597.456 Byte -> 1.584 Byte belegt ANSI.SYS
```

Zusätzlich wird folgender Befehl in CONFIG.SYS eingetragen:

```
device=vdisk.sys 128
```

Damit ergibt sich folgende Berechnung:

128 Kbyte	x	1.024	=	131.072 Byte (einschl. Verzeichniseinträge)
64	x	32	=	2.048 Byte (64 Einträge je 32 Zeichen)
				127.360 Byte tatsächlich verfügbarer Speicher
6 Sek.	x	128	=	768 Byte für die RAM-Disk-Verwaltung
				465.616 Byte verbleibender RAM

Wie Sie aus vorstehendem Beispiel leicht erkennen können, ist nicht der volle Speicherplatz einer RAM-Disk verfügbar. Der zusätzlich verlorene Speicherplatz ist abhängig von der Größe der virtuellen Disk.

Zusätzlich wird eine zweite RAM-Disk angelegt:

```
device=vdisk.sys 200
```

Daraus ergibt sich folgende Berechnung:

200 Kbyte	x	1.024	=	205.568 Byte (einschl. Verzeichniseinträge)
64	x	32	=	2.048 Byte (64 Einträge je 32 Zeichen)
				200.192 Byte tatsächlich verfügbarer Speicher
6 Sek.	x	128	=	768 Byte für die RAM-Disk-Verwaltung
				260.048 Byte verbleibender RAM

Mit CHKDSK kann auch die restliche verfügbare Kapazität in einer RAM-Disk festgestellt werden.

Der Eintrag in CONFIG.SYS kann auch wie folgt aussehen:

```
device=vdisk.sys Diskette=256 Sektor
=512 Verzeichniseintraege=32
```

DRIVER.SYS

Zuordnen einer logischen Laufwerksbezeichnung

Sollten Sie zusätzliche Laufwerke an Ihren Computer anschließen und diese nicht standardmäßig mit Ihrem DOS lauffähig sein, so können Sie einen speziellen Treiber in die Datei CONFIG.SYS installieren. Dafür kann DRIVER.SYS *nicht* verwendet werden. Mit DRIVER.SYS können mehrere logische Laufwerksbezeichnungen zugeordnet werden. Nicht für Festplattenlaufwerke; beachten Sie hierzu den Befehl SUBST (Kapitel 11).

<i>Parameter zu</i>	device=driver.sys /d: # (/c) (/f: #) (/h: #) (/n) (/s: #) (/t: #)
<i>DRIVER.SYS</i>	/d: # Nummer des physischen Laufwerks (zwischen 0 und 255) Wenn Sie ein drittes Laufwerk anschließen, so ist dies Nr. 2 Die erste physische Festplatte hat die Nummer 128 /c Wenn Laufwerk während der Arbeit »abgeschlossen« sein soll Bei PC-DOS bezeichnet dieser Parameter das Diskettenänderungs-Signal (z.B. AT; wenn für ein 1,2-Mbyte-Laufwerk z.B. als zweites logisches Laufwerk ein 320/360-Kbyte-Laufwerk zugeordnet wird) /f: # Spezifiziert die Kapazität und Art des Laufwerks: 0 = 320/360 Kbyte bzw. 160/180 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 = Festplattenlaufwerk 6 = Bandlaufwerk 7 = 1,44 Mbyte /h: # Anzahl der Köpfe des Laufwerks (zwischen 1 und 99) Der Standardwert ist zwei (zweiseitiges Diskettenlaufwerk) /n Wenn es sich um ein nicht wechselbares Medium handelt (z.B. Festplatte) /s: # Anzahl der Sektoren pro Spur (zwischen 1 und 99) Der Standardwert ist 9. /t: # Spuren pro Seite (zwischen 1 und 999) Der Standardwert ist 80.

```
device=driver.sys /d:0 /t:80 /s:15 /h:2 /c /f:1
```

Einem PC/AT mit 1,2 Mbyte und einem 320/360-Kbyte-Laufwerk, sowie einer Festplatte wird das logische Laufwerk D für das erste Laufwerk (1,2 Mbyte) zugeordnet. Damit kann das erste Diskettenlaufwerk mit den Bezeichnungen A: und D: verwendet werden.

In der folgenden Tabelle finden Sie die Angaben, welche logischen Laufwerksbezeichnungen bei bestimmten Systemkonfigurationen für /D: angegeben werden können:

*Logische Laufwerks-
bezeichnungen*

	Interne Disketten- laufwerke	Interne Festplatten- laufwerke	Externe Laufwerke angeschl.	Physische Laufwerks- nummer (D: #)	Logischer Laufwerks- buchstabe
	1	0	Nein	0	C:
	1	0	Nein	1	Fehler
	1	1	Nein	0	D:
	1	1	Nein	1	Fehler
	1	1	Nein	128	D:
	1	2	Nein	0	E:
	1	2	Nein	1	Fehler
	1	2	Nein	128	E:
	1	0	Ja	0	C:*
	1	0	Ja	1	Fehler
	1	0	Ja	2	C:
	1	1	Ja	0	D:
	1	1	Ja	1	Fehler
	1	1	Ja	2	D:
	1	1	Ja	128	D:
	1	2	Ja	0	E:
	1	2	Ja	1	Fehler
	1	2	Ja	2	E:
	1	2	Ja	128	E:*
	1	2	Ja	129	E:
	2	0	Nein	0	C:
	2	0	Nein	1	C:
	2	0	Nein	2	Fehler
	2	1	Nein	0	D:
	2	1	Nein	1	D:
	2	1	Nein	2	Fehler
	2	1	Nein	128	D:
	2	2	Nein	0	E:
	2	2	Nein	1	E:
	2	2	Nein	2	Fehler
	2	2	Nein	128	E:
	2	2	Nein	129	E:
	2	0	Nein	0	C:*
	2	0	Ja	1	C:*
	2	0	Ja	2	C:
	2	1	Ja	0	D:*
	2	1	Ja	1	D:*
	2	1	Ja	2	D:
	2	1	Ja	128	D:*
	2	2	Ja	0	E:*
	2	2	Ja	1	E:*
	2	2	Ja	2	E:
	2	2	Ja	128	E:*
	2	2	Ja	129	E:

* Das externe Laufwerk wird nicht erkannt

PRINTER.SYS

Wechseln der Zeichensatztabelle für Drucker

Zeichensatztabellen für Drucker können mit MODE vorbereitet werden, wenn der Einheitsreiber PRINTER.SYS in CONFIG.SYS eingetragen wird.

Bei IBM kann dies nur für die Druckermodele 4201 und 5202 erfolgen.

Kommandoaufruf

device=[d:][weg]printer.sys lpt # [:]=(typ[, [hwcp][,n]])
oder

device=[d:][weg]printer.sys

lpt # [:]=(typ[, [(hwcp1,hwcp2,...)][,n]])

lpt # PRN, LPT1, LPT2 oder LPT3

typ Druckertyp 4201 oder 5202

hwcp Zeichensatz (437, 850, 860, 863 oder 865)

n Anzahl zusätzlicher Tabellen, die unterstützt werden können (Wert zwischen 0 und 12)

Wird für **n** ein Wert größer als null angegeben, so können mit dem Befehl MODE weitere vorbereitete Zeichensatztabellen aktiviert werden.

Beispiel für den Inhalt der Datei CONFIG.SYS

```
device = c:\dos\ansi.sys
device = c:\dos\display.sys con:(ega,437,3)
device = c:\dos\vdisk.sys 360 512 112
break = on
buffers = 6
files = 12
country = 049,437,c:\dos\country.sys
lastdrive = 1
```

Zuerst werden die Dateien ANSI.SYS (für erweiterte Bildschirm- und Tastaturfunktionen) und VDISK.SYS (RAM-Disk) als Teil von IBMDOS.COM 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 *Strg + Untbr* ein Programm jederzeit unterbricht.

Weitere 6 Zwischenspeicher (insgesamt also 8) sind für Plattenzugriffe reserviert.

Es können maximal 12 Dateien gleichzeitig geöffnet werden (5 für das Betriebssystem und 7 für eigene Programme).

Dann wird der Datensatz für Deutschland geladen (COUNTRY), der damit die Datumseingabe in der Form ermöglicht: Tag.Monat.Jahr. Die Zeit wird im 24-Stunden-Format angezeigt. Die Zahl der logischen Laufwerke wird auf 12 erhöht (LASTDRIVE=L).

Anhang I Netzwerk

PCs im Verbund

Ein Netzwerk ist einfach ausgedrückt eine Punkt-zu-Punkt-Verbindung. Mehrere Computer werden durch Kabel verbunden. Normalerweise benutzen die zusammengeschlossenen Computer eine oder mehrere gemeinsame Festplatten. Ein Computer im Verbund dient dabei als sogenannter Server (Zentralrechner).

DOS bietet die Betriebssystemgrundlage für den Einsatz eines Netzwerkes. Zum Betrieb ist zusätzlich eine Netzwerk-Betriebssoftware (z.B. MS-NET oder IBM-Token-Ring-Netzwerk) notwendig. Außerdem ist eine entsprechende Hardware erforderlich, die die physikalische Verbindung bzw. die Hardware-Ebene der Kommunikation darstellt (z.B. von 3com oder IBM).

Hardware zum Netzwerk

Damit DOS 3.30 mit seinen Netzwerkfunktionen auch genutzt werden kann, muß die Netzwerk-Betriebssoftware diese unterstützen. Ferner können Netzwerkprogramme die von DOS zur Verfügung gestellten Funktionen nutzen.

Record-Locking

Mit DOS 3.30 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 grundsätzlich 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 einzelner Arbeitsplätze zuzugreifen.

Beim Kauf von Programmen müssen Sie darauf achten, daß diese auch **Ihr Netzwerk voll unterstützen**. Software, die dies nicht gewährleistet, kann beim Betrieb zu erheblichen Problemen führen.

Für das Verständnis von Netzwerken und deren Handhabung sind detaillierte Kenntnisse in DOS und der Netzwerkarchitektur notwendig. Bitte beachten Sie hierzu die einschlägige Literatur.

Anhang II Begriffserklärungen

Nachfolgend finden Sie einige wichtige Ausdrücke mit Erläuterung.

<i>Akustikkoppler</i>	Damit können Informationen zwischen einzelnen Computern über das Telefonnetz ausgetauscht werden. Zusätzlich zum Akustikkoppler ist noch ein normales Telefon und ein Programm zur Übertragung der Daten notwendig.
<i>Alphanumerisch</i>	Buchstaben, Ziffern und Sonderzeichen, mit denen Informationen bzw. Daten dargestellt werden.
<i>ANSI</i>	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.
<i>Anwenderprogramm</i>	Damit sind Programme für eine bestimmte Aufgabenstellung, wie Finanzbuchhaltung, Textverarbeitung, Auftragsverwaltung usw., gemeint.
<i>ASCII</i>	Abkürzung für American Standard Code for Information Interchange. Es handelt sich dabei um einen standardisierten Zeichencode. Er stellt gerade im PC-Bereich weltweit einen Industriestandard dar (beachten Sie die Tabelle in Anhang IV).
<i>Assembler</i>	Ein Programm für die Umwandlung von symbolischem Code in »Maschinensprache«.
<i>Asynchron</i>	Eine Betriebsart, bei dem das folgende Kommando von bestimmten Signalen gestartet oder gestoppt wird. Bei der Datenübertragung spricht man von Start- und Stopp-Bit. Eine weitere Betriebsart ist Synchron.
<i>Ausgabe</i>	Das Sichtbarmachen oder Speichern von Ergebnissen einer Computertätigkeit. Eine Ausgabe kann auf Drucker, Bildschirm, Diskette usw. erfolgen.

<i>BATCH-Befehle</i>	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.
<i>Booten</i>	Das Laden und Starten des Betriebssystems.
<i>Byte</i>	Damit wird eine Speicherzelle eines Computers bezeichnet. Jedes Zeichen wird mit einem Byte dargestellt. Ein Byte besteht aus 8 Bit (einzelne binäre Ziffern). Ein Bit ist die kleinste Speichereinheit, die ein Computer verarbeiten kann.
<i>Centronics-Schnittstelle</i>	Parallele Schnittstelle (meist Druckeranschluß), die sich als weitverbreiteter Standard eingeführt hat.
<i>Compiler</i>	Ein Programm, das den Code einer Hochsprache (Cobol, Modula, Pascal usw.) in Maschinensprache umwandelt.
<i>Composite Video</i>	Ein Signal, das alle Elemente enthält, die ein Monitor zur Darstellung eines Bildes in »Schwarz-Weiß« oder »Farbe« benötigt.
<i>Cursor</i>	Ein Zeichen (meist ein blinkendes Unterstreichungszeichen), das die aktuelle Position am Bildschirm anzeigt und die nächste Eingabe »freigibt«.
<i>Datei</i>	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.
<i>Datenverarbeitung</i>	Allgemeiner Oberbegriff für die Veränderung von Informationen durch bzw. mit einem Computer.
<i>Debugging</i>	Suche und Beseitigung von Fehlern (die wörtliche Übersetzung ist Entwanzung). Dieser Begriff wird vor allem im Zusammenhang mit Programmen verwendet, die noch fehlerhaft sind. Beachten Sie hierzu den Befehl DEBUG in Kapitel 11.
<i>Dienstprogramm</i>	Ein Programm, das für eine Reihe von Aufgaben eingesetzt werden kann (Befehle des Betriebssystems), wie z. B. die Bearbeitung der Festplatte.
<i>Directory</i>	Verzeichnis der Dateien. Es können mehrere auf einer Diskette/Festplatte vorhanden sein (Root-Directory/Sub-Directory). Das Directory befindet sich in einem festgelegten Bereich.
<i>Diskette</i>	Eine dünne, biegsame Plasticscheibe mit einer magnetisierten Oberfläche (Oxydschicht). Sie werden auch als Floppies oder Floppydisks bezeichnet.
<i>EBCDIC</i>	Abkürzung für Expanded Binary Coded Decimal Interchange Code. Dieser Zeichencode bezieht sich auf die Informationsübertragung zwischen Peripheriegeräten und anderen Computern. Er wird vor allem bei Großcomputern verwendet (IBM usw.). Im Gegensatz dazu verwenden PCs ASCII.

<i>Editor</i>	Einfaches zeilenorientiertes Textbearbeitungsprogramm für das Erstellen oder Verändern von Daten (z.B. Texten).
<i>EPROM</i>	Abkürzung für Erasable Programmable Read Only Memory. Dieser elektronische Baustein kann mittels ultravioletter Bestrahlung gelöscht und wieder neu programmiert werden. Es handelt sich dabei um einen nichtflüchtigen Speicher.
<i>Firmware</i>	Ein Programm, das fester Bestandteil des physischen Teils eines Computers ist.
<i>Hardcopy</i>	Erstellen einer 1:1-Kopie vom Bildschirm auf den Drucker.
<i>Harddisk</i>	Siehe Winchester.
<i>Hardware</i>	Der »sichtbare« Teil Ihres Computers. Alle physischen Teile.
<i>Hexadezimal</i>	Ein Zahlensystem mit der Basis 16 (unser gebräuchliches Zahlensystem hat die Basis 10). Zu den Dezimalziffern 0 bis 9 kommen noch die Buchstaben A(=10) bis F(=15) hinzu. Auf diese Weise läßt sich ein Byte immer als zweistellige Hexadezimalzahl darstellen.
<i>Kommunikation</i>	Daten- bzw. Informationsübertragung von einem Punkt zu einem anderen.
<i>künstliche Intelligenz (KI)</i>	Unter diesem Begriff werden Computeranwendungen zusammengefaßt, die eine gewisse Eigenintelligenz aufweisen, die wir normalerweise als menschlich bezeichnen würden. Dazu gehören logisches Schlußfolgern, kreatives Weiterdenken und Lernfähigkeit.
<i>Peripherie</i>	Alle an einen Computer angeschlossenen Geräte (auch Diskette, Festplatte und Bildschirm).
<i>RAM</i>	Abkürzung für Random Access Memory. Ein flüchtiger Speicher. Nach dem Abschalten des Computers wird alles gelöscht. Beim Start wird das DOS dorthin geladen (externe Befehle und Programme werden von der Diskette/Festplatte in den RAM zur Ausführung geladen).
<i>ROM</i>	Abkürzung für Read Only Memory. Ein Speicher, in dem Daten/Programme nur gelesen werden können. Die Informationen werden beim Ausschalten des Computers nicht gelöscht.
<i>Software</i>	Alle Programme, die in einem Computer bearbeitet werden können. Erst durch die Software können Sie mit Ihrem Computer arbeiten.
<i>Speicher oder Arbeitsspeicher</i>	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«, die als Stapel in einem staubdichten Gehäuse untergebracht sind. Dieses System zeichnet sich durch eine höhere Schnelligkeit und Sicherheit aus.
- Zylinder* Eine Festplatte besteht normalerweise aus mehreren festen Scheiben. Jede Oberfläche hat einen eigenen Schreib-Lese-Kopf. Mit Zylinder werden die Spuren bezeichnet, auf die gleichzeitig zugegriffen werden kann (die also übereinander angeordnet sind).

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 4stelligen Zahlen beginnen Sie in der viertletzten Spalte usw.

Sie rechnen eine dezimale Zahl um, indem Sie die nächstniedrigere Zahl in der Tabelle suchen (und das erste Zeichen vormerken), diese subtrahieren und für das Ergebnis die nächstniedrigere Zahl (bzw. bei Gleichheit die gleiche) aus der Tabelle suchen.

Beispiel Zahl 2764 = 2764-2560(A) = 204-192(C) = 12-12(C) = 0 => ACC
Zahl 2050 = 2050-2048(8) = 2- 0(0) = 2- 2(2) = 0 => 802

Anhang IV

Zeichendarstellungen/Code-Tabelle

Hex Digits 1st 2nd	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		▶		0	@	P	'	p	Ç	É	á	⋮	┐	⌌	α	≡
-1	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	┐	⌌	β	±
-2	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	┐	⌌	Γ	≥
-3	♥	!!	#	3	C	S	c	s	â	ô	ú		┐	⌌	π	≤
-4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	┐	—	⌌	Σ	f
-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	┐	+	⌌	σ	J
-6	♠	—	&	6	F	V	f	v	â	û	ª	┐	⌌	⌌	μ	÷
-7	•	↕	'	7	G	W	g	w	ç	ù	º	┐	⌌	⌌	τ	≈
-8	■	↑	(	8	H	X	h	x	ê	ÿ	¿	┐	⌌	⌌	Φ	°
-9	○	↓	)	9	I	Y	i	y	ë	Ö	┐	┐	⌌	┐	Θ	•
-A	■	→	*	:	J	Z	j	z	è	Ü	┐	⌌	⌌	┐	Ω	•
-B	♂	←	+	;	K	[	k	{	ï	¢	½	┐	⌌	■	δ	√
-C	♀	┐	,	'<	L	\	l		î	£	¼	┐	⌌	■	∞	"
-D	♪	↔	-	=	M	]	m	}	ì	¥	ì	┐	⌌	■	ø	²
-E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	┐	⌌	■	ε	■
-F	☼	▼	/	?	O	_	o	△	Å	f	»	┐	⌌	■	∩	

Bild IV.1 PC-Code-Tabelle mit Zeichendarstellung für USA (Nr. 437)

Hex Digits 1st 2nd → ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		▶		0	@	P	'	p	Ç	É	á	⋮	⌒	ø	Ó	-
-1	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	⌒	Ð	β	±
-2	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	⌒	Ê	Ô	=
-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⌒	È	Ò	¼
-4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	⌒	—	Ê	ö	¶
-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	+	ı	Ö	§
-6	♠	—	&	6	F	V	f	v	â	û	ª	Ã	ã	İ	μ	÷
-7	•	↕	'	7	G	W	g	w	ç	ù	º	Ä	Å	İ	þ	~
-8	■	↑	(	8	H	X	h	x	ê	ÿ	ı	©	⌒	İ	þ	°
-9	○	↓	)	9	I	Y	i	y	ë	ÿ	®	⌒	⌒	ı	Ú	..
-A	■	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	ı	Û	.
-B	♂	←	+	;	K	[	k	{	ï	ø	½	⌒	⌒	■	Ü	¹
-C	♀	⌒	,	<	L	\	l		î	£	¼	⌒	⌒	■	ý	³
-D	♪	↔	-	=	M	]	m	}	ı	Ø	ı	¢	=	ı	Ý	²
-E	♫	▲	.	>	N	^	n	~	Ä	×	«	¥	⌒	ı	-	■
-F	☼	▼	/	?	O	_	o	△	Å	f	»	⌒	□	■	'	

Bild IV.2 PC-Code-Tabelle mit Zeichendarstellung (mehrsprachig/Nr. 850)

Hex Digits 1st 2nd → ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		▶		0	@	P	'	p	Ç	É	á	⋮	⌒	⌒	α	≡
-1	☺	◀	!	1	A	Q	a	q	ü	À	í	⋮	⌒	⌒	β	±
-2	☹	↕	"	2	B	R	b	r	é	È	ó	⋮	⌒	⌒	Γ	≥
-3	♥	!!	#	3	C	S	c	s	â	ô	ú	⋮	⌒	⌒	π	≤
-4	♦	¶	\$	4	D	T	d	t	ã	õ	ñ	⋮	⌒	⌒	Σ	f
-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	⋮	⌒	⌒	σ	J
-6	♠	—	&	6	F	V	f	v	Á	Ú	ª	⋮	⌒	⌒	μ	÷
-7	•	↕	'	7	G	W	g	w	ç	ù	º	⋮	⌒	⌒	τ	≈
-8	■	↑	(	8	H	X	h	x	ê	Î	¿	⋮	⌒	⌒	Φ	°
-9	○	↓	)	9	I	Y	i	'y	Ê	Ï	Ò	⋮	⌒	⌒	Θ	•
-A	☉	→	*	:	J	Z	j	z	è	Ü	⌒	⋮	⌒	⌒	Ω	•
-B	♂	←	+	;	K	[	k	{	ï	¢	½	⋮	⌒	⌒	δ	✓
-C	♀	⌒	,	<	L	\	l		ô	£	¼	⋮	⌒	⌒	∞	ⁿ
-D	♪	↔	-	=	M	]	m	}	ì	Û	ì	⋮	⌒	⌒	∅	²
-E	🎵	▲	.	>	N	^	n	~	Ã	Pt	«	⋮	⌒	⌒	ε	■
-F	☼	▼	/	?	O	_	o	△	Â	Ó	»	⋮	⌒	⌒	∩	

Bild IV.3 PC-Code-Tabelle mit Zeichendarstellung für Portugal (Nr. 860)

Hex Digits 1st 2nd	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		▶		0	@	P	`	p	Ç	É			L	⊥	α	≡
-1	☺	◀	!	1	A	Q	a	q	ü	È	'		⊥	≡	β	±
-2	☹	↑	"	2	B	R	b	r	é	Ê	ó		⊥	⊥	Γ	≥
-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⊥	⊥	π	≤
-4	♦	¶	\$	4	D	T	d	t	Â	Ë	..	⊥	—	⊥	Σ	f
-5	♣	§	%	5	E	U	e	u	à	Ï	~	⊥	+	⊥	σ	J
-6	♠	—	&	6	F	V	f	v	¶	û	³	⊥	⊥	⊥	μ	÷
-7	•	↑	'	7	G	W	g	w	ç	ù	·	⊥	⊥	⊥	τ	≈
-8	■	↑	(	8	H	X	h	x	ê	Ï	ı	⊥	⊥	⊥	Φ	°
-9	○	↓	)	9	I	Y	i	y	ë	Ô	ı	⊥	⊥	⊥	Θ	•
-A	⊙	→	*	:	J	Z	j	z	è	Û	ı	⊥	⊥	⊥	Ω	•
-B	♂	←	+	;	K	[	k	{	ĩ	¢	½	⊥	⊥	■	δ	✓
-C	♀	⊥	,	<	L	\	l		î	£	¼	⊥	⊥	■	∞	n
-D	♪	↔	-	=	M	]	m	}	=	Û	¾	⊥	⊥	■	∅	²
-E	♫	▲	.	>	N	^	n	~	À	Û	«	⊥	⊥	■	ε	■
-F	☼	▼	/	?	O	_	o	△	§	f	»	⊥	⊥	■	∩	

Bild IV.4 PC-Code-Tabelle mit Zeichendarstellung für Kanada/frz. (Nr. 863)

Hex Digits 1st 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		▶		0	@	P	`	p	Ç	É	á	⋮	⋮	⋮	α	≡
-1	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	⋮	⋮	β	±
-2	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	⋮	⋮	Γ	≥
-3	♥	!!	#	3	C	S	c	s	â	ô	ú	⋮	⋮	⋮	π	≤
-4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	⋮	⋮	⋮	Σ	f
-5	♣	§	%	5	E	U	e	u	å	ò	Ñ	⋮	⋮	⋮	σ	J
-6	♠	—	&	6	F	V	f	v	å	û	ª	⋮	⋮	⋮	μ	÷
-7	•	↕	'	7	G	W	g	w	ç	ú	º	⋮	⋮	⋮	τ	≈
-8	■	↑	(	8	H	X	h	x	ê	ÿ	¿	⋮	⋮	⋮	Φ	°
-9	○	↓	)	9	I	Y	i	y	ë	Ö	⌈	⋮	⋮	⋮	Θ	•
-A	■	→	*	:	J	Z	j	z	è	Ü	⌈	⋮	⋮	⋮	Ω	•
-B	♂	←	+	;	K	[	k	{	ĩ	ø	½	⋮	⋮	⋮	δ	✓
-C	♀	⌈	,	<	L	\	l		î	£	¼	⋮	⋮	⋮	∞	ⁿ
-D	♪	↔	-	=	M	]	m	}	ï	Ø	ı	⋮	⋮	⋮	∅	²
-E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	⋮	⋮	⋮	ε	■
-F	☼	▼	/	?	O	_	o	△	Å	f	□	⋮	⋮	⋮	∩	

Bild IV.5 PC-Code-Tabelle mit Zeichendarstellung für Skandinavien (Nr. 865)

Erläuterung zur Tabelle

In den Tabellen finden Sie alle mit Computern 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	Dezimal-Wert	Zeichenerstellung
£	9C	9 = 144 C = 12	156	ALT+156
√	FB	F = 240 B = 11	251	ALT+251
Pt	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 V Unterschiede DOS 2.11/3.0/3.1/3.2/3.30

Wenn Sie bereits mit älteren 3.xx-Versionen von DOS gearbeitet haben, werden Sie sehr schnell feststellen, daß sich sehr viel geändert hat. Die wesentlichen Neuerungen seit den ersten Versionen sind vor allem:

- Unterstützung der nationalen Zeichensätze mit richtiger Datum- und Uhrzeitdarstellung sowie Dezimalkomma statt Punkt usw.
- Umfangreiche Möglichkeiten für die Benutzung von Zeichensatztabellen
- Einführung einer mehrsprachigen Zeichensatztabelle
- Schnellere Dateiverarbeitung (siehe auch FASTOPEN)
- Zugriff auf Dateien über Suchpfade (APPEND)
- Einrichten von mehreren logischen Laufwerken auf einer Festplatte (FDISK)
- Verbesserte Programme für die Datensicherung (BACKUP und RESTORE)
usw.

Allein diese Erweiterungen machen es für Sie sinnvoll, die alte DOS-Version gegen die neue auszutauschen. Sie sollten aber beachten, daß DOS 3.30 mehr Speicher (RAM) benötigt.

Hervorzuheben ist vor allem, daß die neue Version auf allen PC-Modellen lauffähig ist (nicht nur auf den sogenannten ATs). Im Rahmen dieses Buches kann nicht auf alle Neuerungen eingegangen werden (z.B. neue Funktionsaufrufe – diese werden nur von Programmierern benötigt).

Neue Befehle

Befehl	Kurzbeschreibung	
APPEND	Veranlaßt das System zum Durchsuchen bestimmter Directories nach Dateien, die im aktuellen bzw. angegebenen nicht gefunden wurden.	ab DOS 3.30
ATTRIB	Setzen eines Read-Only-Attributes («Nur lesen») – Software-Schreibschutz.	ab DOS 3.0
CALL	Aufruf einer BATCH-Datei als Unterprogramm.	ab DOS 3.30
CHCP	Die aktuelle Zeichensatztafel auswählen/anzeigen.	ab DOS 3.30
FASTOPEN	Schneller Zugriff auf Dateien.	ab DOS 3.30
GRAFTABL	Unterstützt erweiterten Zeichensatz im Grafik-Mode (z.B.deutsche Umlaute).	ab DOS 3.0
JOIN	Ersetzt ein Laufwerk durch ein Sub-Directory (Unterverzeichnis).	ab DOS 3.1
LABEL	Disketten-/Plattenname ändern.	ab DOS 3.0
NLSFUNC	Auswahl einer Zeichensatztafel.	ab DOS 3.30
REPLACE	DOS, Programme und Dateien werden geändert oder bestehende ersetzt.	ab DOS 3.2
SELECT	Nationale Tastaturbelegung und Datum/Uhrzeit-Darstellung wählen.	ab DOS 3.0
SHARE	Installieren des File-Sharing-Support (für Netzwerkeinsatz).	ab DOS 3.0
SUBST	Ersetzt eine Sub-Directory (Unterverzeichnis) durch ein Laufwerk.	ab DOS 3.1
XCOPY	Dateien und Verzeichnisse kopieren.	ab DOS 3.2

*Neue DOS-
Konfigurations-Befehle
(CONFIG.SYS)*

Beachten Sie zu diesen Befehlen das Kapitel 14 mit näheren Erläuterungen.

Befehl	Kurzbeschreibung	
COUNTRY	Setzen von landesabhängigen Informationen (Datum/Uhrzeit-Darstellung, Dezimalpunkt).	ab DOS 3.0
DEVICE DRIVER.SYS	Zuordnen von logischen Laufwerksbezeichnungen.	ab DOS 3.2

Befehl	Kurzbeschreibung	
DISPLAY.SYS	Ermöglicht es, eine Zeichensatztafel an einen Bildschirmadapter anzupassen.	ab DOS 3.30
PRINTER.SYS	Ermöglicht es, eine Zeichensatztafel an einen Drucker anzupassen.	ab DOS 3.30
FCBS	Definiert die Anzahl der maximal gleichzeitig geöffneten FCBs, wenn ein Netzwerk eingesetzt wird.	ab DOS 3.0
LASTDRIVE	Maximale Anzahl der logischen (nicht physikalischen) Laufwerke.	ab DOS 3.0
STACKS	Nummer und Größe von Stacks setzen.	ab DOS 3.2

Geänderte Befehle

Änderungen ab Version 2.11 sind angegeben (Neuerungen ab 3.2 sind besonders gekennzeichnet).

Befehl	Kurzbeschreibung	
ATTRIB	Es kann jetzt auch das Dateiänderungsattribut gesetzt werden.	ab DOS 3.2
BACKUP	Sichern von Dateien (normalerweise von Platte auf Diskette). Unterstützt wird jetzt auch das Sichern zwischen verschiedenen Medien (Platte auf Diskette, Platte auf Platte usw.). Datumsdarstellung jetzt auch deutsch (Tag.Monat.Jahr) möglich. Neue Parameter. Formatieren während des Sicherns. Erstellen einer Protokolldatei.	ab DOS 3.30
COMMAND	Es kann jetzt auch die Größe des Environment-Bereiches geändert werden (Umgebungsbereich).	ab DOS 3.2
DATE	Datum anzeigen/eingeben. Es ist jetzt landesabhängig möglich, das Datum in der Form Monat.Tag.Jahr oder Tag.Monat.Jahr oder Jahr.Monat.Tag einzugeben. Das Datum wird jetzt auch in der Systemuhr gespeichert (RTC).	ab DOS 3.30
DISKCOMP	Vergleichen von 2 Disketten. Es werden jetzt auch Laufwerke mit 1,2 Mbyte unterstützt. Ab DOS 3.2 auch 3 1/2-Zoll-Disketten mit 720 Kbyte.	

Befehl	Kurzbeschreibung	
DISKCOPY	Kopieren einer Diskette auf eine zweite. Es werden jetzt auch Laufwerke mit 1,2 Mbyte unterstützt. Ab DOS 3.2 auch 3 1/2-Zoll-Disketten mit 720 Kbyte.	
FDISK	Einrichten von mehreren logischen Laufwerken auf einer Festplatte.	ab DOS 3.30
FORMAT	Formatieren einer Diskette oder Festplatte. Es werden jetzt auch Laufwerke mit 1,2 Mbyte unterstützt. Ab DOS 3.2 auch 3 1/2-Zoll-Disketten mit 720 Kbyte. Zusätzlich wurden ab DOS 3.2 weitere Sicherheitsmechanismen eingebaut.	
GRAFTABL	Ermöglicht das Auswählen und Laden mehrerer verschiedener Zeichensatztabellen.	ab DOS 3.30
GRAPHICS	Kopieren von einer Grafik-Bildschirmdarstellung auf einen Drucker. Es werden jetzt mehrere Drucker unterstützt.	
KEYB	Ersetzt den bisherigen Befehl KEYBxx. Ermöglicht das Arbeiten mit zusätzlichen Tastaturbelegungen.	ab DOS 3.30
LIB LINK	Diese beiden Programme wurden erweitert (nur für Programmierer).	
MODE	Ohne Eingabe von Parametern wird der momentane Status angezeigt. Verwaltung von Zeichensatztabellen. Auswahl zusätzlicher Schnittstellen/Kommunikationseinheiten.	ab DOS 3.2 ab DOS 3.30
PRINT	Drucken (läuft gleichzeitig zu anderer Arbeit ab) Neue Parameter für den Druck.	
RESTORE	Zurückkopieren von mit BACKUP gesicherten Dateien. Unterstützt wird jetzt auch das Sichern zwischen verschiedenen Medien (Platte auf Diskette, Platte auf Platte, usw.) Datumsdarstellung jetzt auch deutsch (Tag.Monat.Jahr) möglich. Verbessert und erweitert.	ab DOS 3.30
SELECT	Mit diesem Dienstprogramm kann jetzt auch die Festplatte eingerichtet werden.	ab DOS 3.2
TIME	Die Uhrzeit wird jetzt auch in der Systemuhr gespeichert (RTC).	

Weitere Änderungen

Bei den externen Befehlen können Sie jetzt vor dem Befehl den Namen des Directorys (Verzeichnis) angeben, in welchem sich dieser befindet (ab Version 3.0).

In der DOS-Version 3.30 wurden im Vergleich zu 3.2 fast alle Befehle, wenn auch teilweise nur geringfügig, geändert. Vor allem wurde die Anzeige von Fehlermeldungen usw. erheblich verbessert. Auf den vorherigen Seiten konnte auf die Unterschiede der beiden Versionen nur im Bezug auf wesentliche Änderungen eingegangen werden.

Bisher war es notwendig, ein Programm, welches in einem Sub-Directory benötigt wurde, in dieses zu kopieren oder aber mit PATH (siehe Kapitel 13) eine entsprechende Zuweisung durchzuführen.

Das Befehlsformat bis Version 2.xx war:

Laufwerksangabe:Befehlsname Parameter

Das Befehlsformat ab Version 3.xx ist:

**Laufwerksangabe:Verzeichnisname\ Befehlsname
Parameter**

Es werden ab DOS 3.30 auch 1,44-Mbyte-3 $\frac{1}{2}$ -Zoll-Diskettenlaufwerke unterstützt.

Für Programmierer ist noch zu erwähnen, daß sich einige Funktionsaufrufe geändert haben und einige neu hinzugekommen sind. Ferner gibt es neue bzw. geänderte Interrupts.

Anhang VI Bekannte Fehler in der DOS-Version 3.30

- | | |
|--|---|
| <i>Name der Diskette/Festplatte</i> | Wenn eine Diskette/Festplatte keinen Namen besitzt (FORMAT ohne /V) oder der Namen mit dem Befehl LABEL gelöscht wurde, so wird dieser falsch angezeigt.
Besitzt die Festplatte einen Namen und die Diskette nicht, so wird als Diskettenname der Name der Festplatte angezeigt. Besitzt auch die Festplatte keinen Namen und Sie haben eine RAM-Disk eingerichtet, so wird der Name der RAM-Disk angezeigt.
Der Name einer Festplatte kann nicht gelöscht werden (mit LABEL), wenn keine RAM-Disk angelegt ist. Im anderen Fall wird der Name der RAM-Disk für die Festplatte angezeigt. |
| <i>Zuweisung mit JOIN</i> | Bei der Anzeige des Inhaltsverzeichnisses des Sub-Directorys wird zwar der Inhalt der Diskette richtig angezeigt, aber als Name wird der der Festplatte angezeigt, soweit die Diskette keinen eigenen besitzt. |
| <i>Leere Diskette</i> | Ist eine eingelegte Diskette leer und Sie lassen sich mit DIR das Inhaltsverzeichnis anzeigen, so wird fälschlich das Root-Directory der Festplatte angezeigt. Sind auf einer Diskette Daten gespeichert, erfolgt die Anzeige korrekt. Das gleiche gilt, wenn Sie den Befehl CHKDSK anwenden. |
| <i>Fehler in der Festplattenverwaltung</i> | Sollten Sie zu den ersten gehören, die DOS 3.30 von IBM bezogen haben, so benötigen Sie unbedingt eine geänderte Version mit dem Treiber DASDDRVR.SYS (bitte Installationsanleitung des Lieferanten beachten). Ohne diesen funktioniert Ihre Festplatte innerhalb kurzer Zeit nicht mehr! |

Zu dieser Änderung wird auch eine neue Version von BACKUP ausgeliefert.

Bitte fragen Sie Ihren Lieferanten, ob Ihre Version bereits dem neuesten Stand entspricht.

Vorstehende Fehler konnten nur mit den neuen IBM-Personal-Systemen /2 festgestellt werden.

Anhang VII Übersicht DOS-Interrupts

Weitere ausführliche Informationen entnehmen Sie bitte dem technischen Handbuch zu DOS 3.30.

Der Aufruf eines DOS-Interrupts erfolgt durch den Assembler-Befehl INT.

Beispiel (Tab) INT (Tab) 21

20h Programm beenden

Sie sollten den Funktionsaufruf 4Ch diesem Interrupt vorziehen.

Aufruf: CS = Segmentadresse des P.S.P.

Ergebnis: -

21h Funktionsaufrufe

Aufruf: AH = Nummer der Funktion (00h – 63h)
andere Register müssen funktionsspezifisch geladen werden

Ergebnis: je nach aufgerufener Funktion

22h Adresse bei Programmbeendigung

23h Adresse bei Strg+C Abbruch (BREAK)

24h Abbruchadresse bei Fehlern (Critical Error Handler Vektor)

25h Lesen von Diskette/Festplatte mit absoluter Adresse

Aufruf: AL = Laufwerksnummer
(0=A; 1=B; usw.)
CX = Zahl der zu lesenden Sektoren
DX = erster Sektor
DS:BX = Puffer-Übertragungsadresse für die Sektoren

Ergebnis: CF = 0 kein Fehler
1 Fehler

AH = 02h Adreßmarkierung nicht gefunden
04h Sektor nicht gefunden
08h DMA-Überlauf
10h CRC fehlerhaft (Prüfsummenfehler)
20h Controller fehlerhaft
40h Positionierung fehlerhaft (SEEK)
80h Gerät antwortete fehlerhaft

26h Schreiben auf Diskette/Festplatte mit absoluter Adresse

Aufruf:	AL	=	Laufwerksnummer (0=A; 1=B; usw.)
	CX	=	Zahl der zu schreibenden Sektoren
Ergebnis:	DX	=	erster Disksektor
	DS:BX	=	Pufferadresse für die Sektoren
	CF	=	0 kein Fehler 1 Fehler
	AH	=	02h Adreßmarkierung nicht gefunden 03h Disk ist schreibgeschützt 04h Sektor nicht gefunden 08h DMA-Überlauf 10h CRC fehlerhaft (Prüfsummenfehler) 20h Controller fehlerhaft 40h Positionierung fehlerhaft (SEEK) 80h Gerät antwortete fehlerhaft

27h Programm beenden und resident halten

Sie sollten den Funktionsaufruf 31h diesem Interrupt vorziehen.

Aufruf:	DX	=	erstes Byte nach dem nicht zu löschenden Programm
	CS	=	Segmentadresse des P.S.P.
Ergebnis:	–		

28h Druck-Handler

Dieser Interrupt wird von DOS intern aufgerufen und darf von anderen Programmen nicht benutzt werden.

29h Schnelle Console-Ausgabe

Dieser Interrupt wird von DOS intern aufgerufen und darf von anderen Programmen nicht benutzt werden.

2Ah bis 2Eh werden von DOS benutzt und sind reserviert!

2Fh Multiplex Interrupt

Funktion 01h	–	Druckerspöoler
Funktion 02h	–	ASSIGN
Funktion 10h	–	SHARE
Funktion B7h	–	APPEND

Interrupts 30h bis 3Fh sind für DOS reserviert!

Anhang VIII Übersicht DOS-Funktionen

Weitere ausführliche Informationen entnehmen Sie bitte dem technischen Handbuch zu DOS 3.30.

00h Programm beenden

Aufruf: AH = 00h
CS = Segmentadresse des P.S.P.
Ergebnis: -

01h Lesen eines an der Tastatur eingegebenen Zeichens und Anzeige

Aufruf: AH = 01h
Ergebnis: AL = ASCII-Wert der betätigten Taste (8 Bit)

02h Anzeige eines Zeichens am Bildschirm

Aufruf: AH = 02h
DL = ASCII-Wert des auszugebenden Zeichens
Ergebnis: -

03h Empfangen eines Zeichens von AUX

Aufruf: AH = 03h
Ergebnis: AL = ASCII-Wert des gelesenen Zeichens

04h Ausgabe eines Zeichens an AUX

Aufruf: AH = 04h
DL = ASCII-Wert des auszugebenden Zeichens
Ergebnis: -

05h Ein Zeichen an den »Drucker« ausgeben (PRN)

Aufruf: AH = 05h
DL = ASCII-Wert des zu druckenden Zeichens
Ergebnis: -

06h Direkte Eingabe von Tastatur/direkte Ausgabe an den Bildschirm

Aufruf: AH = 06h
DL = FFh hole Zeichen von der Tastatur
= 00h-FEH Zeichen zum Monitor senden
Ergebnis: - bei DL = 00h bis FEh
ZF = 0 AL enthält ein Zeichen (8 Bit)
ZF = 1 kein Zeichen vorhanden

07h Direkte Eingabe von der Tastatur, ohne Ausgabe

Aufruf: AH = 07h
Ergebnis: AL = Zeichen von der Tastatur

- 08h Lesen eines Zeichens von der Tastatur, ohne Anzeige**
Aufruf: AH = 08h
Ergebnis: AL = Zeichen von der Tastatur
- 09 Anzeige einer Zeichenkette am Bildschirm**
Aufruf: AH = 09h
DS:DX = Adresse der auszugebenden Zeichenkette
Ergebnis: –
- 0Ah Gepufferte Tastatureingabe**
Aufruf: AH = 0Ah
DS:DX = Segment:Offset des Eingabespeichers
DX zeigt auf Eingabespeicher mit dem Format:
Byte 1 = maximale Zeichenzahl (n)
Byte 2 = tatsächliche Zeichenzahl ohne 0Dh (wird von DOS eingetragen)
Byte 3 – n = Eingabespeicher (von STDIN)
Ergebnis: –
- 0Bh Überprüfen, ob ein Zeichen an der Tastatur eingegeben wurde**
Aufruf: AH = 0Bh
Ergebnis: AL = FFh Zeichen im Eingabespeicher
AL = 00h keine Zeichen im Eingabespeicher
- 0Ch Eingabepuffer löschen und Eingabefunktion aufrufen**
Aufruf: AH = 0Ch
AL = 01h, 06h, 07h, 08h oder 0Ah
(Die angegebene Funktion wird aufgerufen); mit jedem anderen Wert erfolgt keine weitere Verarbeitung
DS:DX = Segment:Offset des Eingabespeichers
(Nur bei AL=0Ah)
Ergebnis: AL = 00h Eingabespeicher war leer; keine weitere Verarbeitung
AL = 8-Bit-Eingabezeichen (gilt nicht für Funktion 0Ah)
- 0Dh Diskettenwechsel (Disk-Reset)**
Aufruf: AH = 0Dh
Ergebnis: –
- 0Eh Laufwerk auswählen bzw. wechseln**
Aufruf: AH = 0Eh
DL = Laufwerksnummer
(0=A, 1=B usw.)
Ergebnis: AL = Anzahl der logischen Laufwerke
- 0Fh Eröffnen einer Datei**
Aufruf: AH = 0Fh
DS:DX = zeigt auf einen ungeöffneten FCB
Ergebnis: AL = 00h Datei gefunden
AL = FFh Datei nicht gefunden
- 10h Schließen einer Datei**
Aufruf: AH = 10h
DS:DX = Adresse des geöffneten FCB

- Ergebnis: AL = 00h Eintrag im Inhaltsverzeichnis gefunden
 AL = FFh Eintrag im Inhaltsverzeichnis nicht gefunden
- 11h Ersten Directory-Eintrag suchen**
 Aufruf: AH = 11h
 DS:DX = Adresse eines ungeöffneten FCB
 Ergebnis: AL = 00h Datei gefunden
 AL = FFh Datei nicht gefunden
- 12h Nächsten Directory-Eintrag suchen**
 Aufruf: AH = 12h
 DS:DX = zeigt auf einen ungeöffneten FCB
 Ergebnis: AL = 00h Datei gefunden
 AL = FFh Datei nicht gefunden
 bzw. keine weiteren Dateien
- 13h Datei(en) löschen**
 Aufruf: AH = 13h
 DS:DX = Adresse eines ungeöffneten FCB
 Ergebnis: AL = 00h Datei gefunden
 AL = FFh Datei nicht gefunden
 oder Datei(en) mit Read-Only-Attribut
- 14h Datei sequentiell lesen**
 Aufruf: AH = 14h
 DS:DX = Adresse eines geöffneten FCB
 Ergebnis: AL = 00h Datensatz gelesen
 01h Dateiende erreicht, keine weiteren Zeichen
 gelesen
 02h Segmentüberlauf (DTA reicht über
 Programmsegmentende hinaus)
 03h Dateiende erreicht, Teil-Satz wurde gelesen
 und mit 00h zur vollständigen Länge auf-
 gefüllt
- 15h Datei sequentiell schreiben**
 Aufruf: AH = 15h
 DS:DX = Adresse eines geöffneten FCB
 Ergebnis: AL = 00h Datensatz geschrieben
 01h Disk voll
 02h Segmentüberlauf (DTA reicht über
 Programmsegmentende hinaus)
- 16h Erstellen einer neuen Datei**
 Aufruf: AH = 16h
 DS:DX = Adresse eines ungeöffneten FCB
 Ergebnis: AL = 00h Datei erstellt oder geleert
 AL = FFh Datei nicht erstellt (Directory voll)
- 17h Dateiname ändern**
 Aufruf: AH = 17h
 DS:DX = Adresse eines »Spezial«-FCB
 Ergebnis: AL = 00h Datei(en) gefunden (Name(n) geändert)
 AL = FFh Datei nicht gefunden, oder neuer Name
 bereits vorhanden
- 19h Aktuelles Laufwerk feststellen**
 Aufruf: AH = 19h
 Ergebnis: AL = aktuelles Laufwerk (0=A, 1=B usw.)

1Ah Adresse des Datenübertragungspuffers DTA neu bestimmen

Aufruf: AH = 1Ah

DS:DX = Adresse des neuen DTA

Ergebnis: –

1Bh Holen der Daten des aktuellen Laufwerks

Aufruf: AH = 1Bh

Ergebnis: AL = FFh Funktion fehlerhaft abgeschlossen
oder
Sektoren pro Cluster

CX = Byte pro Sektor

DX = Anzahl der Cluster des Laufwerks

DS:BX = Adresse des FAT-ID-Byte

1Ch Holen der Daten eines Laufwerks

Aufruf: AH = 1Ch

DL = Laufwerksnummer (0 = aktuelles Laufwerk,
1 = A usw.)Ergebnis: AL = FFh falsche Laufwerksnummer
oder
Sektoren pro Cluster

CX = Byte pro Sektor

DX = Anzahl der Cluster des Laufwerks

DS:BX = Adresse des FAT-ID-Byte

1Fh Holen des D.P.B. (Disk Parameter Block) für aktuelles Laufwerk

Aufruf: AH = 1Fh

Ergebnis: DS:BX = zeigt auf DPB (Disk Parameter Block) des
aktuellen Laufwerkes**21h Wahlfreies Lesen mit Satznummer (Random Read)**

Aufruf: AH = 21h

DS:DX = Adresse eines geöffneten FCB

Ergebnis: AL = 0 Datensatz gelesen
1 Dateiende, keine weiteren Zeichen gelesen
2 Segmentüberlauf (DTA zu klein)
3 Dateiende erreicht, Teil-Satz wurde gelesen
und mit 00h aufgefüllt**22h Wahlfreies Schreiben mit Satznummer (Random Write)**

Aufruf: AH = 22h

DS:DX = Adresse des geöffneten FCB

Ergebnis: AL = 00h Datensatz geschrieben
01h Diskette voll
02h Segmentüberlauf (DTA zu klein)**23h Ermitteln der Dateigröße**

Aufruf: AH = 23h

DS:DX = Adresse eines ungeöffneten FCB

Ergebnis: AL = 00h Datei gefunden

AL = FFh Datei nicht gefunden

24h Relative Satznummer setzen

Aufruf: AH = 24h

DS:DX = zeigt auf geöffneten FCB

Ergebnis: –

25h Interrupt-Vektor setzen

Aufruf: AH = 25h
 AL = Nummer des Interrupts
 DS:DX = Adresse der neuen Interrupt-Routine
 Ergebnis: –

26h Neuen P.S.P. (Programm Segment Prefix) erstellen

Aufruf: AH = 26h
 DX = Segmentadresse des neuen P.S.P.
 Ergebnis: –

27h Wahlfrei Blöcke lesen (Random Block Read)

Aufruf: AH = 27h
 DS:DX = Adresse des geöffneten FCB
 CX = Anzahl der zu lesenden Datensätze
 Ergebnis: AL = 00h Lesen erfolgreich abgeschlossen
 01h Dateiende
 02h Segmentüberlauf
 03h Dateiende, Teil-Satz wurde gelesen und mit 00h aufgefüllt
 CX = Anzahl der gelesenen Datensätze

28h Wahlfrei Blöcke schreiben (Random Block Write)

Aufruf: AH = 28h
 DS:DX = Adresse des geöffneten FCB
 CX = Anzahl der zu schreibenden Datensätze
 (00h = aktualisiere Dateigröße im FCB)
 Ergebnis: AL = 00h Schreiben erfolgreich abgeschlossen
 01h Disk voll
 02h Segmentüberlauf
 CX = Anzahl der geschriebenen Datensätze

29h Dateiname suchen und FCB erstellen

Aufruf: AH = 29h
 DS:SI = Adresse der zu durchsuchenden Zeichenkette
 ES:DI = Adresse des neuen ungeöffneten FCB
 AL = Bit 0 = 0
 Untersuchung endet, wenn Trennzeichen gefunden
 = 1
 führende Trennzeichen werden ignoriert (z. B. Leerzeichen)
 Bit 1 = 0
 Laufwerksnummer im FCB wird auf 0 (aktuelles Laufwerk) gesetzt, wenn die Zeichenkette keine Laufwerksnummer enthält
 = 1
 Laufwerksnummer im FCB wird nicht verändert, wenn die Zeichenkette keine Laufwerksnummer enthält
 = Bit 2 = 0
 Dateiname im FCB wird auf 8 Leerzeichen gesetzt, wenn die Zeichenkette keinen Dateinamen enthält
 = 1
 Dateiname im FCB wird nicht verändert, wenn die Zeichenkette keinen Dateinamen enthält

Bit 3 = 0

Namenszusatz im FCB wird auf 3 Leerzeichen gesetzt, wenn die Zeichenkette keinen Namenszusatz enthält

= 1

Namenszusatz im FCB wird nicht verändert, wenn die Zeichenkette keinen Namenszusatz enthält

Bits 4–7

werden ignoriert

Ergebnis: AL = 00h keine Wild-Card-Zeichen gefunden

= 01h Wild-Card-Zeichen gefunden

= FFh ungültige Laufwerksbezeichnung

DS:SI = erstes Byte nach gefundenem Dateinamen

ES:DI = Adresse des ungeöffneten neuen FCB

2Ah Systemdatum holen

Aufruf: AH = 2Ah

Ergebnis: CX = Jahr (1980–2099)

DH = Monat (1–12)

DL = Tag (1–31)

AL = Wochentag

(0 = Sonntag, 1 = Montag usw.)

2Bh Systemdatum neu setzen

Aufruf: AH = 2Bh

CX = Jahr (1980–2099)

DH = Monat (1–12)

DL = Tag (1–31)

Ergebnis: AL = 00h Datum war gültig

= FFh Datum war ungültig

Änderung nicht durchgeführt

2Ch Systemzeit holen

Aufruf: AH = 2Ch

Ergebnis: CH = Stunden (0–23)

CL = Minuten (0–59)

DH = Sekunden (0–59)

DL = Hundertstelsekunde (0–99)

2Dh Systemzeit neu setzen

Aufruf: AH = 2Dh

CH = Stunden (0–23)

CL = Minuten (0–59)

DH = Sekunden (0–59)

DL = Hundertstelsekunde (0–99)

Ergebnis: AL = 00h Zeit war gültig

= FFh Zeit war ungültig

Uhrzeit wird nicht geändert

2Eh Setzen bzw. Löschen des Verify-Flag

Aufruf: AH = 2Eh

AL = 00h es wird nicht überprüft (VERIFY OFF)

= 01h es wird überprüft (VERIFY ON)

DL = 00h (nur DOS-Version 1 und 2)

Ergebnis: –

2Fh Adresse des Datenübertragungspuffers DTA holen

Aufruf: AH = 2Fh

Ergebnis: ES:BX = aktuelle Adresse des DTA

30h DOS-Versionsnummer holen

Aufruf: AH = 30h

AL = 00h

Ergebnis: AL = Haupt-Versionsnummer (z.B. 3)

AH = Unter-Versionsnummer (z.B. 14h für 20)

BH = OEM- bzw. Herstellernummer (z.B. IBM=00h)

BL: CX = 24-Bit-Benutzernummer (optional)

31h Programm beenden und resident halten

Aufruf: AH = 31h

AL = Return- bzw. Ausgangs-Code (Fehlercode)

DX = Programmgröße im Speicher in Paragraphen (bei EXE-Programmen +100h für P.S.P.)

Ergebnis: -

32h Absolute Adresse des D.P.B. (Disk Parameter Block) holen

Aufruf: AH = 32h

DL = Laufwerk (aktuelles Laufwerk = 0, A = 1, B = 2 usw.)

Ergebnis: AL = FFh Laufwerk ungültig

= 00h Laufwerk gültig

DS:BX = Adresse des D.P.B.

33h Strg+C-Überprüfung holen/setzen (Break Flag)

Aufruf: AH = 33h

AL = 00h Flag holen

Ergebnis: AL = FFh Fehler beim Funktionsaufruf

= 00h fehlerfreier Ablauf

DL = 00h Strg+C-Flag ausgeschaltet

= 01h Strg+C-Flag eingeschaltet

Aufruf: AH = 33h

AL = 01h Flags setzen

DL = 00h Strg+C-Flag ausschalten

= 01h Strg+C-Flag einschalten

Ergebnis: AL = FFh Fehler beim Funktionsaufruf

= 01h fehlerfreier Ablauf

DL = 00h Strg+C-Flag ausgeschaltet

= 01h Strg+C-Flag eingeschaltet

34h DOS Critical Flag Address holen

Aufruf: AH = 34h

Ergebnis: ES:BX = Adresse des Merkers für DOS-Funktionsunterbrechung

35h Interrupt-Vektor holen

Aufruf: AH = 35h

AL = Nummer des Interrupts

Ergebnis: ES:BX = Adresse der Interrupt-Routine

36h Ermitteln des freien Speicherplatzes auf einem Laufwerk

Aufruf: AH = 36h

DL = Laufwerk (0=aktuelles Laufwerk, 1=A, 2=B, usw.)

Ergebnis: AX = FFFFh Laufwerk ungültig
 oder
 Sektoren pro Cluster
 BX = Anzahl freie Cluster
 DX = Gesamtzahl der Cluster für das Laufwerk
 CX = Byte pro Sektor

37h Parameterseparator holen (Character Operations)

Aufruf: AH = 37h
 AL = 00h holen Einleitungszeichen für Zusatzeingabe
 in DL (normalerweise »/«)
 01h setze Einleitungszeichen für Zusatzeingabe
 auf Wert in DL
 02h zeige gültigen Namen für Gerätetreiber
 in DL
 03h setze Namen für gültigen Gerätetreiber
 DL = Wert des Einleitungszeichens (bei AL = 01h)
 Wert für Gerätetreibernamen (bei AL = 03h)
 Ergebnis: AL = FFh ungültiger Funktionsaufruf
 DL = Wert des Einleitungszeichens (bei AL = 00h)
 gültiger Gerätetreibernamen (bei AL = 02h)

38h Landesspezifische Daten holen

Aufruf: AH = 38h
 AL = 00h aktuelle Landesinformation holen
 01h-FEh Landesinformation für Landescode
 holen (Landescode < 255)
 FFh Landesinformation für Landescode in
 Register BX holen
 BX = Landescode (falls AL = FFh)
 DS:DX = Adresse eines 34-Byte-Speicherbereiches für die
 Landesinformationen
 CF = 0 Kein Fehler
 BX = Landescode
 1 Daten nicht gefunden (AX = 02h)

38h Landesspezifische Daten setzen

Aufruf: AH = 38h
 DX = FFFFh (-1)
 AL = 01h-FEh Landescode
 FFh Landescode in BX, wenn größer 254
 BX = Landescode (falls AL = FFh)
 Ergebnis: CF = 0 Kein Fehler
 1 ungültiger Landescode (AX = 02h)

39h Sub-Directory erstellen

Aufruf: AH = 39h
 DS:DX = Adresse des Pfadnamens (ASCIIZ)
 Ergebnis: CF = 00h kein Fehler
 001 Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht gefunden
 05h Directory voll (Root-Directory)
 Name existiert schon
 Kein Zugriffsrecht für Netzbenutzer

3Ah Sub-Directory-Eintrag löschen

Aufruf: AH = 3Ah
 DS:DX = Adresse des Pfadnames (ASCIIIZ)
 Ergebnis: CF = 00h Eintrag gelöscht
 01h Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht gefunden
 05h Ein Verzeichnisname im Pfad falsch
 Sub-Directory beinhaltet noch Dateien
 06h aktuelle Directory

3Bh Wechseln des aktuellen Directory

Aufruf: AH = 3Bh
 DS:DX = Adresse des Pfad-Namens (ASCIIIZ)
 Ergebnis: CF = 00h kein Fehler
 01h Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht gefunden

3Ch Neue Datei erstellen**Bestehende Datei löschen bzw. überschreiben**

Aufruf: AH = 3Ch
 CX = Datei-Attribut
 DS:DX = Adresse des Pfad-/Dateinamens (ASCIIIZ)
 Ergebnis: CF = 00h Datei angelegt
 AX = Dateinummer (File Handle)
 CF = 01h Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht gefunden
 04h keine Dateinummer frei
 05h Root-Directory voll
 Dateiname existiert und Read-Only
 Kein Zugriffsrecht (Netzwerk)

3Dh Eröffnen einer Datei (File Handle)

Aufruf: AH = 3Dh
 AL = Zugriffsart
 DS:DX = Adresse des Pfad-/Dateinamens (ASCIIIZ)
 Ergebnis: CF = 00h Datei eröffnet
 AX = Dateinummer (File Handle)
 CF = 01h Fehler (Fehlercode in AX)
 AX = 01h falsche Zugriffsart bzw. SHARE nicht
 geladen
 02h Datei nicht gefunden
 03h Verzeichnis nicht gefunden
 04h kein Handle frei
 05h kein Zugriff
 12h nicht erlaubter Zugriff (Netzwerk)

3Eh Schließen einer Datei (File Handle)

Aufruf: AH = 3Eh
 BX = Dateinummer (File Handle)
 Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)
 AX = 06h falsche Dateinummer
 Dateinummer nicht offen

3Fh Lesen aus einer Datei oder von einem Gerät

Aufruf: AH = 3Fh
BX = Dateinummer (File Handle)
CX = Anzahl der zu lesenden Bytes
DS:DX = Zeiger auf Zwischenspeicher

Ergebnis: CF = 0 kein Fehler
AX = Anzahl der gelesenen Bytes
AX = 0 Dateiende erreicht
CF = 1 Fehler (Fehlercode in AX)
AX = 05h kein Zugriff
06h falsche Dateinummer
Dateinummer nicht offen

40h Schreiben in eine Datei oder Übertragung an ein Gerät

Aufruf: AH = 40h
BX = Dateinummer (File Handle)
CX = Anzahl der zu schreibenden Bytes
DS:DX = Zeiger auf Zwischenspeicher

Ergebnis: CF = 0 kein Fehler
AX = Anzahl der geschriebenen Bytes
CF = 1 Fehler (Fehlercode in AX)
AX = 05h kein Zugriff (Netzwerk)
06h falsche Dateinummer
Dateinummer nicht offen

41h Löschen eines Dateieintrags

Aufruf: AH = 41h
DS:DX = Adresse des Pfad-/Dateinamens (ASCIIZ)

Ergebnis: CF = 00h kein Fehler
01h Fehler (Fehlercode in AX)
AX = 02h Datei nicht gefunden
03h Verzeichnis nicht gefunden
05h Datei ist Read-Only

42h Dateizeiger bewegen (Pointer)

Aufruf: AH = 42h
AL = Art der Bewegung
00h Zeiger wird zum Dateianfang plus Zahl der definierten Bytes bewegt
01h Zeiger wird von der momentanen Position um die Anzahl der definierten Bytes weg bewegt
02h Zeiger wird zum Dateiende plus Zahl der definierten Bytes bewegt

BX = Dateinummer (File Handle)
CX:DX = Anzahl der Bytes, um die der Zeiger bewegt werden soll (32-Bit-Offset)

Ergebnis: CF = 0 kein Fehler
DX:AX = neue Zeigerposition
CF = 1 Fehler (Fehlercode in AX)
AX = 01h falsche Funktion (AL)
Unerlaubte Angabe in CX:DX
06h falsche Dateinummer
Datei nicht offen

43h Dateiattribute holen oder ändern

Aufruf: AH = 43h
 AL = 00h Attribut-Byte der Datei holen
 01h Attribut-Byte der Datei setzen
 CX = neues Attribut-Byte
 DS:DX = Adresse des Pfad-/Dateinamens (ASCIIZ)

Ergebnis: CF = 00h kein Fehler
 CX = Attribut-Byte (falls AL = 0)
 CF = 01h Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion in AL
 03h Verzeichnis nicht gefunden
 05h kein Zugriff

44h 00h Daten eines Gerätes holen

Aufruf: AH = 44h
 AL = 00h
 BX = Dateinummer (File Handle)

Ergebnis: CF = 0 kein Fehler
 DX = Geräteinformation (WORD)
 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 05h falsche Dateinummer

44h 01h Daten eines Gerätes setzen

Aufruf: AH = 44h
 AL = 01h
 BX = Dateinummer (File Handle)
 DX = Geräteinformation (WORD)

Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 05h falsche Dateinummer

44h 02h Daten eines Gerätes holen

Aufruf: AH = 44h
 AL = 02h
 BX = Dateinummer
 CX = Anzahl der zu lesenden Bytes
 DS:DX = Adresse eines Zwischenspeichers

Ergebnis: CF = 0 kein Fehler
 AX = Anzahl der gelesenen Bytes
 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 05h falsche Dateinummer

44h 03h Kontroll-Daten zu einem Zeichentreiber senden

Aufruf: AH = 44h
 AL = 03h
 BX = Dateinummer
 CX = Anzahl der zu sendenden Bytes
 DS:DX = Adresse des Zwischenspeichers

Ergebnis: CF = 0 kein Fehler
 AX = Anzahl der gelesenen Bytes
 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 05h falsche Dateinummer

44h 04h Kontroll-Daten von einem Blocktreiber empfangen

Aufruf: AH = 44h
AL = 04h
BL = Laufwerksnummer (0=aktuelles Laufwerk, 1=A usw.)
CX = Anzahl der zu lesenden Bytes
DS:DX = Adresse des Zwischenspeichers
Ergebnis: CF = 0 kein Fehler
AX = Anzahl der gelesenen Bytes
1 Fehler (Fehlercode in AX)
AX = 01h falsche Funktion
05h falsche Laufwerksnummer

44h 05h Daten an einen Blocktreiber senden

Aufruf: AH = 44h
AL = 00h
BL = Laufwerksnummer (0=Arbeitslw, 1=A, usw.)
CX = Anzahl der zu sendenden Bytes
DS:DX = Adresse des Zwischenspeichers
Ergebnis: CF = 0 kein Fehler
AX = Anzahl der gelesenen Bytes
1 Fehler (Fehlercode in AX)
AX = 01h falsche Funktion
05h falsche Laufwerksnummer

44h 06h Eingabestatus prüfen

Aufruf: AH = 44h
AL = 06h
BX = Dateinummer (Handle)
Ergebnis: CF = 0 kein Fehler
Für ein Gerät:
AL = 00h Gerät ist nicht bereit
AL = FFh Gerät ist bereit
Für eine Datei:
AL = 00h Datei-Pointer steht bei EOF
AL = FFh Datei-Pointer nicht bei EOF
CF = 1 Fehler (Fehlercode in AX)
AX = 01h falsche Funktion
05h Zugriff nicht möglich
06h Dateinummer falsch
0Dh Daten falsch

44h 07h Ausgabestatus prüfen

Aufruf: AH = 44h
AL = 07h
BX = Dateinummer (Handle)
Ergebnis: CF = 0 kein Fehler
Für ein Gerät:
AL = 00h Gerät ist nicht bereit
AL = FFh Gerät ist bereit
Für eine Datei:
AL = FFh
CF = 1 Fehler (Fehlercode in AX)

AX = 01h falsche Funktion
 05h Zugriff nicht möglich
 06h Dateinummer falsch
 0Dh Daten falsch

44h 08h Prüfen, ob Medium wechselbar ist

Aufruf: AH = 44h
 AL = 08h
 BL = Laufwerksnummer (0=aktuelles Laufwerk,
 1=A usw.)
 Ergebnis: CF = 0 kein Fehler
 AL = 00h Medium kann gewechselt werden
 AL = 01h Medium kann nicht gewechselt werden
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 0Fh Laufwerksnummer falsch

44h 09h Prüfen, ob Gerät der eigene PC ist, oder das Netzwerk (Remote)

Aufruf: AH = 44h
 AL = 09h
 BL = Laufwerksnummer (0=aktuelles Laufwerk,
 =A usw.)
 Ergebnis: CF = 0 kein Fehler
 DX = Attribut-WORD des Treibers
 Bit 12 = 0 Laufwerk ist Local
 1 Laufwerk ist Remote
 CF = 1 Fehler (Fehlercode in AX)
 X = 01h falsche Funktion
 0Fh Laufwerksnummer falsch

44h 0Ah Prüfen, ob eine Datei (Handle) im PC oder im Netzwerk vorhanden ist

Aufruf: AH = 44h
 AL = 0Ah
 BX = Dateinummer (Handle)
 Ergebnis: CF = 0 kein Fehler
 DX = Attribut-WORD für Datei oder Gerät
 Bit 15 = 0 Local
 1 Remote
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
 06h Dateinummer falsch oder nicht geöffnet

44h 0Bh Wiederholungen eines Zugriffsversuchs ändern

Aufruf: AH = 44h
 AL = 0Bh
 BX = Anzahl der Wiederholungen (Standardwert = 3)
 CX = Wartezeit (Standardwert = 1)
 Ergebnis: CF = 0 kein Fehler
 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion

**44h 0Ch Allgemeine Ein-/Ausgabekontrolle für Zeichentreiber
Zeichensatztabelle wechseln**

Aufruf: AH = 44h
 AL = 0Ch
 BX = Dateinummer (Handle)
 CH = Kategorie-Code
 01h = COM1, COM2, COM3 oder COM4
 03h = CON (Bildschirm und Tastatur)
 05h = LPT1, LPT2 oder LPT3
 CL = Funktions-Code
 45h = Wiederholungszähler setzen
 4Ah = Zeichensatztabelle auswählen
 4Ch = Start Zeichensatztabelle vorbereiten
 4Dh = Ende Zeichensatztabelle vorbereiten
 65h = Wiederholungszähler holen
 6Ah = Abfrage der ausgewählten Zeichensatztabelle
 6Bh = Abfrage der vorbereiteten Liste
 DS:DX = Adresse des Parameterblocks
 Ergebnis: CF = 0 Kein Fehler
 Bei CL = 65h, 6Ah oder 6Bh
 DS:DX = Adresse des Parameterblocks
 CF = 1 Fehler (Fehlercode in Register AX)

44h 0Dh Allgemeine Ein-/Ausgabekontrolle für Blocktreiber

Aufruf: AH = 44h
 AL = 0Dh
 BL = Laufwerk (0 = aktuelles Laufwerk, 1 = A usw.)
 CH = Kategoriecode:
 08h = Disk-Laufwerk
 CL = Funktionscode:
 40h = Geräteparameter setzen
 41h = Spur schreiben
 42h = Spur formatieren und überprüfen
 60h = Geräteparameter holen
 61h = Spur lesen
 62h = Spur überprüfen
 DS:DX = Adresse des Parameterblocks
 Ergebnis: CF = 0 Kein Fehler
 Bei CL = 60h oder 61h
 DS:DX = Adresse des Parameterblocks
 CF = 1 Fehler (Fehlercode in Register AX)

44h 0Eh Logische Laufwerksbezeichnung holen

Aufruf: AH = 44h
 AL = 0Eh
 BL = Laufwerk (0 = aktuelles Laufwerk, 1 = A usw.)
 Ergebnis: CF = 0 Kein Fehler
 AL = Mapping Code
 00h Nur eine Laufwerkbezeichnung ist dem
 Block-Gerät zugeordnet
 01h-1Ah Eine weitere Laufwerkbezeichnung ist
 zugewiesen (01 = A, 02 = B usw.)
 CF = 1 Fehler (Fehlernummer in Register AX)

44h 0Fh Logische Laufwerksbezeichnung setzen

Aufruf: AH = 44h
 AL = 0Fh
 BL = Laufwerk (0 = aktuelles Laufwerk, 1 = A usw.)
 Ergebnis: CF = 0 Kein Fehler
 AL = Mapping Code
 00h Nur eine Laufwerksbezeichnung ist dem Block-Gerät zugeordnet
 01h-1Ah Weitere Laufwerksbezeichnung zugewiesen (01 = A, 02 = B usw.)
 CF = 1 Fehler (Fehlernummer in Register AX)

45h Duplizieren einer Dateinummer

Aufruf: AH = 45h
 BX = Dateinummer (File Handle)
 Ergebnis: CF = 0 kein Fehler
 AX = neue Dateinummer
 CF = 1 Fehler (Fehlercode in AX)
 AX = 04h keine Dateinummer frei
 06h falsche Dateinummer

46h Neue Dateinummer erzwingen

Aufruf: AH = 46h
 BX = erste Dateinummer (File Handle)
 CX = zweite Dateinummer
 Ergebnis: CF = 0 kein Fehler
 CF = 1 Fehler (Fehlercode in AX)
 AX = 06h Datei nicht offen
 Handle nicht vorhanden

47h Aktuellen Pfadnamen holen

Aufruf: AH = 47h
 DL = Laufwerksnummer (0=Arbeitslaufwerk, 1=A usw.)
 DS:SI = Adresse für Pfadnamen (64 Byte)
 Ergebnis: CF = 00h kein Fehler
 01 Fehler (Fehlercode in AX)
 AX = 0Fh Laufwerk falsch

48h Zusätzlichen Arbeitsspeicher anfordern

Aufruf: AH = 48h
 BX = benötigter Speicher in Paragraphen (16 Byte)
 Ergebnis: CF = 0 kein Fehler
 AX = Segmentadresse des angeforderten Speichers
 CF = 1 Fehler (Fehlercode in AX)
 AX = 07h Speicherkontrollblock zerstört
 08h nicht genug freier Speicher vorhanden
 BX = Zahl der maximalen Paragraphen

49h Speicher freigeben (vorher mit 48h angefordert)

Aufruf: AH = 49h
 ES = Segmentadresse des freizugebenden Speichers
 Ergebnis: CF = 0 kein Fehler
 CF = 1 Fehler (Fehlercode in AX)
 AX = 07h Speicherkontrollblock zerstört
 09h falsches Segment benannt

4Ah Angeforderten Speicherbereich in der Größe ändern

Aufruf: AH = 4Ah
BX = Neue Größe des zusätzlichen Speicherbereichs in Paragraphen
ES = Segmentadresse des zu modifizierenden zusätzlichen Speicherbereiches
Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)
AX = 07h Speicherkontrollblock zerstört
 08h nicht genug freier Speicher vorhanden
BX = Zahl der maximalen Paragraphen
 09h falsches Segment benannt

4Bh Programm laden und starten

Aufruf: AH = 4Bh
AL = 00h Laden und Starten
 03h Overlay laden
DS:DX = Adresse des Programmnamens (ASCIIZ)
ES:BX = Adresse des Parameterblocks
Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)
AX = 01h falsche Funktion
 02h Weg oder Datei nicht gefunden
 08h nicht genug freier Speicher vorhanden
 0Ah falscher Environment-Block
 0Bh falscher Aufruf/Format

4Ch Programm beenden

Aufruf: AH = 4Ch
AL = Return-Code bzw. Ausgangskode
Ergebnis: -

4Dh Return-Code holen (siehe auch Funktionen 31h und 4Ch)

Aufruf: AH = 4Dh
Ergebnis: AL = übergebener Wert (Return-Code)
 AH = 00h Normales Ende
 01h durch *Strg+C* abgebrochen
 02h wegen Hardwarefehler abgebrochen
 03h Ende und resident halten

4Eh Ersten Directory-Eintrag finden

Aufruf: AH = 4Eh
CX = Attribut-Byte
DS:DX = Adresse des Pfad-/Dateinamens (ASCIIZ)
Ergebnis: CF = 00h Datei gefunden
 01h Fehler (Fehlercode in AX)
AX = 02h Datei nicht gefunden
 12h keine Datei mehr vorhanden

4Fh Nächsten Directory-Eintrag finden

Aufruf: AH = 4Fh
Ergebnis: CF = 00h Datei gefunden
 01h Fehler (Fehlercode in AX)
AX = 12h keine Datei mehr vorhanden

- 50h Setzen Programm Segment Präfix Adresse (P.S.P.)**
 Aufruf: AH = 50h
 BX = neues Segment für P.S.P.
 Ergebnis: –
- 51h Holen Program Segment Präfix Adresse (P.S.P.)**
 Aufruf: AH = 51h
 Ergebnis: BX = Segment des aktuellen Programmes
- 52h Holen der Adresse des Reentry Flag**
 Aufruf: AH = 52h
 Ergebnis: ES:BX = Adresse des Reentry Flag
- 53h Setzen Diskparameterblock (D.P.B.)**
 Aufruf: AH = 53h
 DS:SI = Adresse eines BIOS-Parameterblocks
 ES:BP = Adresse eines noch nicht beschriebenen D.P.B.
- 54h Hole Verify Flag (Ein-/Ausschalten mit 2Eh)**
 Aufruf: AH = 54h
 Ergebnis: AL = 00h Keine Überprüfung nach dem Schreiben
 01h Überprüfung der Daten nach dem Schreiben
- 55h Duplizieren P.S.P. (Programmkopf)**
 Aufruf: AH = 55h
 DX = Segment des neuen P.S.P.
 Ergebnis: –
- 56h Directory-Eintrag umbenennen oder verschieben**
 Aufruf: AH = 56h
 DS:DX = Adresse des bisherigen Dateinamens (ASCIIZ)
 ES:DI = Adresse des neuen Dateinamens (ASCIIZ)
 Ergebnis: CF = 00h Vorgang fehlerfrei abgeschlossen
 01h Fehler (Fehlercode in AX)
 AX = 02h Datei nicht vorhanden
 03h Pfad nicht gefunden
 05h Zugriff nicht möglich
 11h Laufwerk nicht identisch
- 57h Holen und Setzen von Datum und Uhrzeit einer Datei**
 Aufruf: AH = 57h
 AL = 00h Datum und Zeit holen
 BX = Dateinummer (File Handle)
 Ergebnis: CF = 0 kein Fehler
 CX = Uhrzeit der letzten Modifikation
 DX = Datum der letzten Modifikation
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion (AL)
 06h falsche Dateinummer
 Aufruf: AH = 57h
 AL = 01h Datum und Zeit setzen
 BX = Dateinummer (File Handle)
 CX = neue Uhrzeit
 DX = neues Datum
 Ergebnis: CF = 0 kein Fehler
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion (AL)
 06h falsche Dateinummer

58h Holen/Setzen Speicherverwaltungs-Strategie

Aufruf: AH = 58h
 AL = 00h Art der Speicherzuteilung holen
Ergebnis: CF = 0 kein Fehler
 AX = 00h Suche von unten
 01h Beste Möglichkeit
 02h Suche von oben
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion
Aufruf: AH = 58h
 AL = 01h Art der Speicherzuteilung setzen
 BX = 00h Suche von unten
 01h Beste Möglichkeit
 02h Suche von oben
Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion

59h Erweiterte Fehlermeldung holen (Extended Error)

Beachten Sie Kapitel 3.5 für Fehlercode-Beschreibung
Aufruf: AH = 59h
 BX = 00h
Ergebnis: AX = erweiterter Fehlercode
 BH = Fehlerklasse
 BL = Vorschlag zur Fehlerbeseitigung
 CH = Fehlerort
 ES:DI = Adresse des Diskettennamens (ASCIIZ)
 (bei unerlaubtem Diskettenwechsel
 AX = 0022h)
 Alle anderen Registerinhalte werden zerstört!

5Ah Erstellen einer Arbeitsdatei (temporäre Datei)

Aufruf: AH = 5Ah
 CX = Attribute der neuen Datei
 DS:DX = Zeiger auf Verzeichnisname plus 1 Byte mit Wert
 00h und 13 Byte freiem Speicher
Ergebnis: CF = 0 kein Fehler
 AX = Dateinummer (File Handle)
 CF = 1 Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht vorhanden
 05h Zugriff nicht möglich

5Bh Neue Datei erstellen, wenn noch nicht vorhanden

Aufruf: AH = 5Bh
 CX = Attribute der neuen Datei
 DS:DX = Zeiger auf Dateinamen (ASCIIZ)
Ergebnis: CF = 0 kein Fehler
 AX = Dateinummer (File Handle)
 CF = 1 Fehler (Fehlercode in AX)
 AX = 03h Verzeichnis nicht vorhanden
 04h keine Dateinummer mehr frei
 05h Zugriff nicht möglich
 50h Datei schon vorhanden

5Ch Sperren/Freigeben Dateiteil

Aufruf: AH = 5Ch
 AL = 00h Dateiteil sperren
 BX = Dateinummer (File Handle)
 CX:DX = Anfang der Sperrung (Datei-Offset)
 SI:DI = Länge der Sperrung

Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)

AX = 01h falsche Funktion
 06h falsche Dateinummer
 16h Sperrung nicht möglich

Aufruf: AH = 5Ch
 AL = 01h Sperrung aufheben
 BX = Dateinummer (File Handle)
 CX:DX = Anfang der Freigabe (Datei-Offset)
 SI:DI = Länge der Freigabe

Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)

AX = 01h falsche Funktion
 06h falsche Dateinummer
 16h Freigabe nicht möglich

5Dh Für Netzwerk reserviert**5Eh Computernamen holen und Druckerinitialisierung**

Aufruf: AH = 5Eh
 AL = 00h Namen des eigenen Computers holen
 DS:DX = Adresse eines 16-Byte-Speicherbereichs (ASCII-Z)

Ergebnis: CF = 0 kein Fehler
 CH = 00h Computernamen nicht vorhanden
 > 00h Computernamen vorhanden
 CL = netBIOS Namensnummer (bei CH nicht 00h)
 DS:DX = Adresse des Namens (bei CH nicht 00h)
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion (MS-NET fehlt)

Aufruf: AH = 5Eh
 AL = 02h Setzen Druckerinitialisierung
 BX = Umleitungs-Listenindex
 CX = Länge des Initialisierungsstrings (max. 64 Byte)
 DS:SI = Adresse des Initialisierungsstrings

Ergebnis: CF = 0 kein Fehler
 1 Fehler (Fehlercode in AX)

AX = 01h falsche Funktion (MS-NET nicht geladen)

Aufruf: AH = 5Eh
 AL = 03h Holen Druckerinitialisierung
 BX = Umleitungs-Listenindex
 ES:DI = Adresse für den Initialisierungsstring

Ergebnis: CF = 0 kein Fehler
 CX = Länge des Initialisierungsstrings
 ES:DI = Adresse des Initialisierungsstrings
 CF = 1 Fehler (Fehlercode in AX)
 AX = 01h falsche Funktion (MS-NET nicht geladen)

5Fh »Umleitungsliste« bearbeiten

Aufruf:	AH	= 5Fh
	AL	= 02h Umleitungs-Listen-Eintrag holen
	BX	= Umleitungs Listenindex
	DS:SI	= Adresse des lokalen Gerätenamens (128 Byte)
	ES:DI	= Adresse des Netzwerknamens (128 Byte)
Ergebnis:	CF	= 0 kein Fehler
	BH	= Gerätestatus Flag
	Bit 0	= 0 gültiges Gerät
		1 ungültiges Gerät
	Bit 1	= ungültiges Gerät
	Bit 1-7	= Reserviert
	BL	= 03h Drucker
		04h Laufwerk
	CX	= Gesicherte Parameterdaten
	DX	= Inhalt zerstört
	BP	= Inhalt zerstört
	DS:SI	= Adresse des lokalen Gerätenamens (ASCIIZ)
	ES:DI	= Adresse des Netzwerknamens (ASCIIZ)
		1 Fehler (Fehlercode in AX)
	AX	= 01h MS-NET nicht geladen
		12h Keine Einträge mehr vorhanden
Aufruf:	AH	= 5Fh
	AL	= 03h Gerät »umleiten«
	BL	= 03h Drucker
		04h Laufwerk
	CX	= »private« Parameterdaten
	DS:SI	= Adresse des lokalen Gerätenamens (ASCIIZ)
	ES:DI	= Adresse des (Netzwerknamens (ASCIIZ) und folgendem Kennwort (ASCIIZ)
Ergebnis:	CF	= 0 kein Fehler
		1 Fehler (Fehlercode in AX)
	AX	= 01h MS-NET nicht geladen
		03h Verzeichnis nicht vorhanden
		05h Zugriff nicht möglich
Aufruf:		08h Arbeitsspeicher zu klein
	AH	= 5Fh
	AL	= 04h »Umleitung« löschen
	DS:SI	= Adresse des lokalen Gerätenamens (ASCIIZ)
Ergebnis:	CF	= 0 kein Fehler
		1 Fehler (Fehlercode in AX)
	AX	= 01h MS-NET nicht geladen
		0Fh Network-Server pausiert

60h Holen des tatsächlichen Namens eines Laufwerks oder Directory

Aufruf:	AH	= 60h
	DS:SI	= Adresse des Ursprungsnamens
	ES:DI	= Adresse für das Ergebnis
Ergebnis:	ES:DI	= Text des benutzten Namens

62h Holen Programm Segment Prefix Adresse (P.S.P.)

Aufruf:	AH	= 62h
Ergebnis:	BX	= Segmentadresse des P.S.P.

65h Erweiterte Landesinformationen holen

Aufruf: AH = 65h
 AL = Unterfunktion
 01h Holen Landesinformationen
 02h Holen Adresse der Umsetzungstabelle für 80h bis FFh
 04h Holen der Adresse der Dateinamen-Umsetzungstabelle
 06h Holen Adresse der Vergleichstabelle für das Sortieren
 BX = Zeichensatztabellennummer (-1 = aktive für CON)
 CX = Speichergröße für die zu holenden Informationen
 DX = Landesnummer (-1, wenn Standardwert)
 ES:DI = Adresse des Speichers für die Informationen
 Ergebnis: CF = 0 Kein Fehler
 ES:DI = Adresse der erweiterten Landesinformationen
 1 Fehler (Fehlercode in Register AX)
 AX = 01h Angabe in CX war kleiner als 5
 02h Landesnummer und Zeichensatztabellennummer nicht verträglich
 oder
 eine oder beide Angaben sind falsch

66h Zeichensatztabelle holen/setzen

Aufruf: AH = 66h
 AL = 01h Zeichensatztabelle holen
 Ergebnis: CF = 0 Kein Fehler
 BX = Nummer der aktiven Zeichensatztabelle
 DX = System-Zeichensatztabelle (beim Bootup)
 CF = 1 Fehler (Fehlercode in Register AX)
 Aufruf: AH = 66h
 AL = 02h Zeichensatztabelle setzen
 BX = Nummer der Zeichensatztabelle
 Ergebnis: CF = 0 Kein Fehler
 1 Fehler (Fehlercode in Register AX)
 AX = 02h Zeichensatztabelle für Land nicht verfügbar oder Zeichensatztabelle nicht gefunden
 AX = 65h Ein Gerät kann mit der neuen Zeichensatztabelle nicht arbeiten

67h Maximale Anzahl der Handles setzen

Aufruf: AX = 67h
 BX = Anzahl der maximal offenen Handles
 Ergebnis: CF = 0 Kein Fehler
 1 Fehler (Fehlercode in Register AX)

68h Zwischengespeicherte Daten (Puffer) einer Datei speichern

Aufruf: AX = 68h
 BX = Dateinummer (Handle)
 Ergebnis: CF = 0 Kein Fehler
 1 Fehler (Fehlercode in Register AX)

Anhang IX Übersicht DOS-Fehlercodes

*Fehlercodes
(Register AX)*

Nummer	Fehlerbezeichnung
01h	Ungültige Funktionsnummer
02h	Datei nicht gefunden
03h	Pfad/Weg nicht gefunden
04h	Kein Handle bzw. keine Dateinummer verfügbar
05h	Zugriff nicht möglich
06h	Ungültige Dateinummer
07h	Speicherkontrollblöcke zerstört
08h	Zu wenig Speicher
09h	Speicherblockadresse ungültig
0Ah	Ungültiger Environment-Bereich
0Bh	Format ungültig
0Ch	Zugriffsscode ungültig
0Dh	Daten ungültig
0Eh	Unbenutzt
0Fh	Laufwerkbezeichnung falsch
10h	Versuch, das Standardverzeichnis zu verlagern
11h	Anderes Gerät
12h	Weitere Dateien nicht gefunden

*Erweiterte Fehlercodes
(Register AX)*

zur Funktion 59h

Nummer	Fehlerbezeichnung
00h	Kein Fehler
01h	Falscher Funktionsaufruf
02h	Datei nicht vorhanden
03h	Weg (Verzeichnis, Pfad) nicht vorhanden
04h	Keine Dateinummer (File Handle) mehr frei
05h	Zugriff nicht möglich
06h	Falsche Dateinummer
07h	Kontrollblock für Speicherzuteilung zerstört
08h	Arbeitsspeicher zu klein (Insufficient Memory)
09h	Adresse des Speicherkontrollblockes falsch
0Ah	Systemspeicher (Environment) fehlerhaft
0Bh	Falsches Format
0Ch	Falscher Zugriffscode
0Dh	Falsche Daten
0Eh	RESERVIERT (zur Zeit nicht benutzt)
0Fh	Falsche Laufwerksangabe
10h	Versuch, das aktuelle Directory zu löschen
11h	Peripheriegerät ist nicht gleich
12h	Keine weitere(n) Datei(en)
13h	Diskette ist schreibgeschützt
14h	Falsche Disketteneinheit
15h	Laufwerk nicht bereit
16h	Falscher Festplattenbefehl
17h	CRC-Fehler (Cyclic Redundancy Check)
18h	Falsche Sektorgröße (Anforderungsblock-Länge)
19h	Suchfehler
1Ah	Keine DOS-Diskette
1Bh	Sektor nicht gefunden
1Ch	Kein Papier mehr (Papierende beim Drucker)
1Dh	Schreibfehler
1Eh	Lesefehler
1Fh	Allgemeiner Fehler
20h	Datei ist gesperrt
21h	Datei oder Datensatz ist gesperrt
22h	Falsche Diskette (unerlaubter Diskettenwechsel)
23h	FCB nicht mehr vorhanden
24h bis 31h	RESERVIERT
32h	Netzwerkanfrage wird nicht unterstützt
33h	Zielcomputer im Netzwerk antwortet nicht
34h	Name ist im Netzwerk doppelt
35h	Netzwerkname nicht gefunden
36h	Netzwerk ist aktiv
37h	Netzwerkgerät ist nicht mehr vorhanden
38h	Befehl ist für NET-BIOS zu groß
39h	Fehler der Netzwerkkarte (Hardwarefehler)
3Ah	Antwort des Netzwerkes ist falsch
3Bh	Unerwarteter Netzwerkfehler
3Ch	Netzwerkkarte des Zielcomputers ist nicht kompatibel
3Dh	Warteliste für das Drucken ist voll
3Eh	Warteliste für das Drucken ist nicht voll

Nummer	Fehlerbezeichnung
3Fh	Speicher für die zu druckende Datei zu klein
40h	Name des Netzwerkes wurde gelöscht
41h	Zugriff nicht möglich
42h	Typ des Netzwerkgerätes ist falsch
43h	Name des Netzwerkes nicht gefunden
44h	Name des Netzwerkes ist zu lang
45h	NET-BIOS-Kapazität ist überschritten
46h	Vorübergehend unterbrochen
47h	Netzwerk hat Anfrage nicht akzeptiert
48h	Drucker- oder Laufwerksumleitung vorübergehend unterbrochen
49h bis 4Fh	RESERVIERT
50h	Datei schon vorhanden
51h	RESERVIERT (zur Zeit nicht benutzt)
52h	Kann nicht bearbeiten
53h	Interrupt 24h Fehler
54h	Struktur ist falsch
55h	Bereits zugeteilt
56h	Falsches Paßwort (Kennwort)
57h	Falsche Zusatzeingabe (Parameter)
58h	Schreibfehler des Netzwerkes

*Fehlerklassen
(Funktion 59h,
Register BH)*

Nummer/Klasse	Fehlerbezeichnung
01h	Ressourcen (Speicher, Kanäle usw.) voll bzw. belegt
02h	Vorübergehender Fehler, z.B. Datei- oder Datensatzsperrung
03h	Zugriffsberechtigung nicht vorhanden
04h	Interner Fehler der Systemsoftware
05h	Hardwarefehler
06h	Fehler in der Systemsoftware, der nicht durch den aktiven Prozeß hervorgerufen wurde, sondern z.B. durch falsche Konfiguration
07h	Fehler des Anwendungsprogramms
08h	Datei oder Daten nicht gefunden
09h	Datei oder Daten haben falsches Format, Typ usw.
0Ah	Datei oder Daten gesperrt
0Bh	Falsche Diskette eingelegt, Diskette beschädigt oder Festplattenfehler
0Ch	Name usw. existiert bereits
0Dh	Anderer Fehler

Fehlerbeseitigung
(Funktion 59h,
Register BL)

Nummer/Vorschlag	Fehlerbezeichnung
01h	Wiederholen, anschließend Anzeige an Bediener
02h	Nach einer Pause erneut versuchen
03h	Sollte der Bediener Daten eingegeben haben (z.B. Laufwerksbezeichnung oder Dateiname), erneut zur Eingabe auffordern
04h	Beenden und alle Daten löschen
05h	Sofort beenden (ohne Dateien zu schließen und Indexe zu aktualisieren). Der Fehler ist vom Programm nicht zu beheben
06h	Fehler kann nicht dargestellt werden / ignorieren
07h	Bediener auffordern, eine Aktion durchzuführen (z.B. Diskette tauschen), dann erneut versuchen

Fehlerort
(Funktion 59h,
Register CH)

Nummer/Ort	Fehlerbezeichnung
01h	Unbekannt
02h	Peripherie mit wahlfreiem Zugriff (z.B. Laufwerk)
03h	Netzwerk
04h	Peripherie mit serielltem Zugriff (z.B. Drucker)
05h	Arbeitsspeicher (RAM)

Stichwortverzeichnis

A

Adreßregister 143
 Akustikkoppler 13 f., 283
 American National Standard Institute 283
 – Standard Code for Information Interchange 283
 Anschlüsse 14
 ANSI 283
 ANSI-Steuersequenzen 265
 ANSI.SYS 265
 Anwendungsmöglichkeiten 15
 Anzahl der Laufwerke 262
 APPEND 74, 97
 Arbeitsspeicher 285
 Archivbyte 102
 ASCII 283
 ASCII-Zeichentabelle 185
 Assembler 283
 ASSIGN 100
 AT-Tastaturbezeichnungen 27
 ATTRIB 102
 Aufbau eines Directorys 52
 Ausstattung des PC 35
 AUTOEXEC.BAT 40, 45

B

Back-Slash 74
 BACKUP 61, 105
 Bandlaufwerk 16
 Basiszeiger 143
 BAT-Dateien 29
 BATCH-Befehle 245
 BATCH-Datei 109, 157, 245
 – abbrechen 245
 – unterbrechen 207
 baud-rate 199
 Bedingungserfüllung in
 BATCH-Dateien 188
 Befehle, externe 29, 32
 –, interne 29 f.
 Befehlsanforderungs-Zeichen 31
 Befehlsausführung 29
 Befehlsebene 120
 Befehlseingabe 47
 Befehlseingabe-Bereitschafts-
 meldung 17
 Befehls-Funktionsübersicht 31

Befehlsinterpretierer 30, 119
 Befehlskette 89
 Befehlsverkettung 87
 Befehlswiederholung 24
 Betriebssystem 14, 17
 Betriebssystem-Anweisungen 29
 Betriebssystem-Kopien
 herstellen 37
 Bildschirm löschen 118
 Bildschirmausgabe seitenweise 203
 Bildschirmbetriebsart
 bestimmen 197
 Bildschirminhalt auf
 Drucker 58
 Bildschirmmodus auswählen 268
 Bildschirmsteuermöglich-
 keiten 265
 Bit 65
 Blocktreiber 93
 Booten 284
 BOOT-LOADER 18
 Boot-Loader 178
 Bootup 17
 BREAK 108, 258
 Btx-Anschluß 13
 BUFFERS 258
 Byte 65, 284

C

CALL 109, 253
 CD (CHDIR) 112
 Centronics-Schnittstelle 284
 Charaktertreiber 93
 CHCP 111, 256
 CHDIR (CD) 73, 77, 112
 CHKDSK 114
 CLS 118
 Codesegment 143
 COM-Dateien 29
 COMMAND 119
 COMMAND.COM 18, 29 f.
 COMMAND.COM-Prozessor
 verlassen 169
 COMP 122
 Compiler 284
 Composite Video 284
 COMSPEC-Angabe 225
 Concatenation 127
 CONFIG.SYS 40, 46, 257
 COPY 47, 52, 54, 82, 125
 COUNTRY 36, 259
 CTTY 131
 Cursor 284
 Cursorposition 266

D

DATE 36, 132
 Datei 284
 – löschen 56
 – umbenennen 55
 Dateiattribut 103
 Datei-Einträge 53
 Dateien erneuern 214, 217
 – kopieren 82, 125
 – löschen 147
 – schützen 102
 – sichern 105
 – zurückholen 219
 Dateiinhalte anzeigen 237
 Dateiinhalte vergleichen 122
 Dateiname 48
 Dateinamen ändern 216
 Dateinamen-Aufbau 47
 Dateisicherung bei
 Festplatte 61
 Dateisteuerblöcke 260
 Dateizuordnungstabelle 272
 Daten sortieren 230
 Datenfernübertragung 13
 Datensegment 143
 Datensicherung 16
 Datenumleitung 87
 Datenumleitungs-Zeichen 87
 Datenverarbeitung 284
 Datum 36, 132
 DEBUG 134
 DEBUG-Befehle 138
 Debugging 284
 DEL 47, 52, 56, 81, 147
 DEVICE 264
 Device Driver 264
 Dienstprogramme 18, 29
 DIN-Norm 26
 DIR 47, 52, 79, 148
 Directory 46, 284
 – anzeigen 52, 79, 148
 – bearbeiten 74
 – »seitenweise« 53
 – wechseln 73, 77, 112
 Directory-Aufbau 52, 72
 Directory-Einträge 53
 Directory-Hierarchie 71
 Directory-Struktur
 – anzeigen 73
 – kopieren 241
 Directory-Weg angeben 73
 – definieren 97
 Disk Operating System 18
 DISKCOMP 152

DISKCOPY 37, 60, 155
Diskette 63, 66
– formatieren 178
– kopieren 155
– prüfen 114
Diskettenarten 63
Diskettenbehandlung 67
Disketten-Kompatibilität 65
Diskettenlaufwerk 13, 14
Diskettennamen ändern 195
– anzeigen 240
– erstellen 195
– löschen 195
Diskettenvergleich 152
DISPLAY.SYS 265, 270
DOS 17
– installieren 223
DOS-Befehl wiederholen 177
DOS-Befehle 29, 95
– eingeben 22
DOS-Fehlercode-Übersicht 327
– Funktionen 305
– Interrupts 303
DOS-Partition 44, 172
DRIVER.SYS 265, 275
DRIVER.SYS-Parameter 276
Drucken von Textdateien 208
Drucker 13
– einstellen 57
– konfigurieren 197
– umleiten 57, 197

E
EBCDIC 284
ECHO 157, 253
Editor 285
EDLIN 159
EDLIN-Befehle 160
Eingabegeräte 87
Einheitentreiber 264
Einschubkarten 19
Environment-Bereich 119, 121, 225
EOF-Markierung 123
EPROM 285
ERASE 52, 56, 81, 147
ERRORLEVEL 188, 252
EXE-Dateien 29
EXE2BIN 167
EXIT 169
Expanded Binary Coded Decimal
Interchange Code 284
Externe Befehle 29, 32
Extrasegment 143

F
Farbwahl 267
FASTOPEN 170
FCBS 260
FDISK 43, 45, 171
Fehler in DOS 3.30 301
Fehlercode-Übersicht 327
Festplatte 13, 16, 63, 68
– einrichten 43
– formatieren 178
– prüfen 114
Festplatten-Bereiche aufteilen 44
Festplatten-Einrichtungs-
programm 43
Festplatten-Installation prüfen 46
Festplattenname anzeigen 240
Festplatten-Vorbereitung 171
File-Allocation-Table (FAT) 178
File-Name 47
FILES 261
File-Sharing 105
File-Sharing-Support 227
File-Typ 47
Filter 87, 89
FIND 89, 174
Firmware 285
FOR 177, 253
FORMAT 39, 45, 178
FORMAT-Parameter 179
Formatieren 39
Funktionen-Übersicht 305
Funktionstasten 21 f.
Funktionstastenbelegung 23

G
Gerätenamen 93
Gerätetreiber 93
GOTO 183, 253
Grafik-Mode 185
Grafikzeichenausgabe auf
Drucker 186
Grafikzeichensatztabelle 185
GRAFTABL 58, 185, 256
GRAPHICS 58, 186

H
Hardcopy 285
Hardcopy-Funktion 24, 58
Harddisk 285
Hardware 13
Hardwareerweiterungen 19
Hardware-Kompatibilität 19
Hexadezimal 285

I
IBM-Token-Ring
(Netzwerk) 281
IBMBIO.COM 18, 234
IBMDOS.COM 18, 244
IF 188, 253
Indexregister 143
Informationseinheiten 65
Inhaltsverzeichnis 46, 71
– anzeigen 52, 79, 148
Initialisierungsteil 30
Installation von DOS 37
Interne Befehle 29 f.
Interrupt-Übersicht 303

J
JOIN 113, 190

K
KEYB 192, 256
KEYBOARD.SYS 192
Kommandoaufbau 95
Kommando-Interpreter 263
Kommentarzeilen in BATCH-
Dateien 215
Kommunikation 13
Kompatibilität 19
Konfiguration von DOS 257
Kopieren 54
– von Dateien 82, 125
– von Disketten 155
Kopierrichtungen 62
Kopiervorgang 38
Künstliche Intelligenz 285

L
LABEL 52, 195
Landescode 45
Landescode-Tabelle 260
Landeskennziffer 259
LASTDRIVE 233, 262
Laufwerk anwählen 63
– wechseln 49
Laufwerkbuchstabe 31
Laufwerks-Kompatibilität 65
Laufwerks-Name 53
Laufwerksbezeichnung für
Sub-Directory 232
– vertauschen 190
– zuordnen 275
Laufwerksbezeichnungen 277
Laufwerkseinteilung 96

Laufwerksumleitung 100
 Laufwerkzugriff beschleunigen 170
 Löschen von Dateien 81, 147

M

Maschinencode 134
 Maschinensprache-Befehle 135
 MD (MKDIR) 196
 Mehrplatzfähigkeit 18
 Menüprogramm 250
 MF-Tastatur 21, 26
 MF-Tastaturbezeichnungen 27
 MKDIR (MD) 76, 196
 MODE 47, 57, 197, 256
 Modem 14
 Monitor 15
 MORE 89, 91, 203
 MS-NET (Netzwerk) 281

N

Namenszusatz 47
 Netzwerk 227, 281
 Netzwerkbetrieb 105
 Netzwerk-Server 281
 NLSFUNC III, 204, 256

O

OS/2 18

P

Parallele Schnittstelle 14
 parity 199
 Partition 44, 172
 PATH 73 f., 205
 PAUSE 207, 253
 PC-/XT-Tastaturbezeichnungen 27
 PC-Hardware 13
 Peripherie 13, 285
 Peripherie-Anpassung 57
 PIPEs 89
 Plotter 14
 PRINT 59, 208
 PRINTER.SYS 265, 278
 Program Counter 143
 Programmumwandlung
 EXE-COM 167
 Programmunterbrechung 108, 258
 Programmzähler 143
 Prompt 18, 31, 37
 PROMPT 212, 265
 – ändern 269
 PROMPT-Befehl 23

Prozessor (CPU) 143
 Prozeßsteuerung 17
 PS/2-Hardware 13
 PS/2-Hardware-Änderungen 19

R

RAM 30, 63, 285
 RAM-Disk 271, 274
 RAM-Speicher 121
 Random Access
 Memory 30, 63, 285
 RD (RMDIR) 222
 Read-Only 102
 Read Only Memory 285
 Read-Write 102
 Record-Locking 281
 RECOVER 214
 Register 135
 REM 215, 253
 RENAME 47, 52, 55
 – (REN) 216
 REPLACE 217
 RESTORE 61, 219
 RMDIR (RD) 79, 222
 ROM 285
 ROOT-Directory 53, 71 f.
 RS232C-Schnittstelle 14
 RTC (Real-Time-Clock) 132

S

Schnittstellen 57
 Schreib-Lese-Kopf 65
 Schreibmaschinentastatur 21
 Schreiboperationen prüfen 239
 Segment 139
 Seitenweise Bildschirmaus-
 gabe 203
 Sektoren 64, 214
 Sektorenanzahl 272
 Sekundär-Prozeß 121
 SELECT 38, 45, 223, 256
 Serielle Schnittstelle 14
 SET 225
 SGR-Befehl 268
 SHARE 227
 SHELL 263
 SHIFT 228, 253
 Sicherungsdiskette 107
 Sicherungskopien erstellen 59
 Software 13 ff., 285
 Software-Kompatibilität 19
 Software-Probleme 19

SORT 89, 230
 Sortieren von Daten 230
 Speicher 285
 – (RAM) 17, 29
 Speicheradresse 139
 Speichererweiterungsbereich 272
 Speicherinhalt 139
 Speicherkapazität 53
 Speicherplatzberechnung 274
 Speicherstruktur 134
 Speicherzelle 139
 Sprungbefehl in BATCH-Datei 183
 Spuren 64
 STACK-Segment 167
 STACKS 264
 Stapelsegment 143
 Stapelzeiger 143
 Starten von DOS 36
 Startvorgang von DOS 257
 Status-Report bei CHKDSK 115
 Statusregister 143
 Sub-Directories löschen 222
 Sub-Directory 45, 71
 – anlegen 76, 196
 – löschen 79
 SUBST 101, 113, 232
 Suchreihenfolge von Dateien 30
 SYS 234
 Systemanforderungszeichen 212
 Sytem-Bereitschaftsmeldung 37
 Systemdateien kopieren 234
 Systemdatum 132
 Systemlader 178
 Systemzeit 235

T

Tagesdatum 36, 132
 Tageszeit 37
 Tape-Streamer 60
 Tastatur 15, 21
 Tastaturanpassung 192, 224
 – nationale 38
 Tastaturbezeichnungen 27
 Tastaturcode 39
 Tastaturcode-Tabelle 192
 Tastatureingabespeicher 22
 Tastaturroutine 45
 Tastatursteuermöglichkeiten 265
 Tastenkombinationen 26
 Teletex 13
 Testhilfeprogramm 134
 Text anzeigen 25
 Textbearbeitungsprogramm 159

Textdateien drucken 208

erstellen 159

TIME 37, 235

Tracks 64

TREE 73, 91, 236

Treiber 87

Treiber-Namen 93

TYPE 91, 237

U

Uhrzeit 37, 235

Umrechnungstabellen Hexa-
dezimal/Dezimal 298

Unix 18

Unterbrechung einer BATCH-
Datei 207

Unterschiede der DOS-
Versionen 295

Unterverzeichnis anlegen 196

Unterverzeichnisse löschen 222

Fehlermeldungen

»(w)rite error, no destination
defined« 138

»Angeg. COMMAND-Verzeichnis
fehlerhaft« 121

»Bad command or filename« 30

»Datei hat Status Nur-Lesen« 160

»Datei kann nicht umgewandelt
werden 167

»Datei nicht gefunden« 76, 162

»Dateien haben unterschiedliche
Größe« 124

»Dateifehler während Datenüber-
gabe (PIPE)« 90

»Dateiname muß angegeben
werden« 159

»Dateisteuerblock (FCB) nicht
verfügbar« 261

»Disketten-/Laufwerksart nicht
kompatibel« 15

»Doppelter Dateiname oder Datei
nicht gefunden«

V

V.24-Schnittstelle 14

Variablenzahl in BATCH-
Dateien 228

VDISK.SYS 265, 271

VER 238

VERIFY 239

Versionsnummer anzeigen 238

Verzeichnisstruktur

anzeigen 236

Verzeichniswege festlegen 205

VOL 240

W

Wege der Directories
festlegen 205

Wildcardzeichen 50

Winchester 286

Wurzel-Verzeichnis 71

X

XCOPY 39, 61, 82, 83, 241

Z

Zählregister 143

Zeichendarstellungs-Tabellen 289

Zeichenfolgen suchen 174

Zeichensätze 185, 255

Zeichensatztabelle III, 255

– anzeigen 201

– bearbeiten 256

– laden 204

– verwalten 197

– vorbereiten 200

– wechseln 270, 278

Zeit 235

Zentraleinheit 14

Zifferntastefeld 21

Zugriffszeit 68

Zwischenspeicher 258

Zylinder 286

»Eingabefehler« 159

»EXE and HEX files cannot be
written« 146

»Falscher Befehl oder Datei-
name« 30, 49, 75

»Fehler beim Erstellen der
Datei« 54

»File cannot be con-
verted« 167

»Kein APPEND« 97

»Kein Platz mehr im Umgebungs-
bereich« 226

»Laufwerksbuchstabe muß ange-
geben werden« 179

»Name nicht gefunden« 184

»Nicht genug Platz auf
Diskette/Platte« 84

»Rekursive Kopie nicht mög-
lich« 242

»Schreibschutz« 67

»SHARE bereits installiert« 227

»SUBST für Netzwerklaufrwerk
nicht möglich« 232

»Tastaturroutine nicht gefun-
den« 224

»Ungültige Anzahl Para-
meter« 104

»Ungültige Datumseingabe« 132

»Ungültige Laufwerksangabe« 64

»Ungültige Umgebungsgröße
angegeben« 121

»Ungültiger Parameter« 101

»Ungültiges Verzeichnis« 78, 112

»Vergleichsfehler auf Seite x,
Spur x« 152

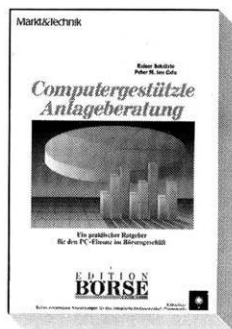
»Verzeichnis kann nicht angelegt
werden« 77

»x Vergleichsfehler – Vergleich
beendet« 123

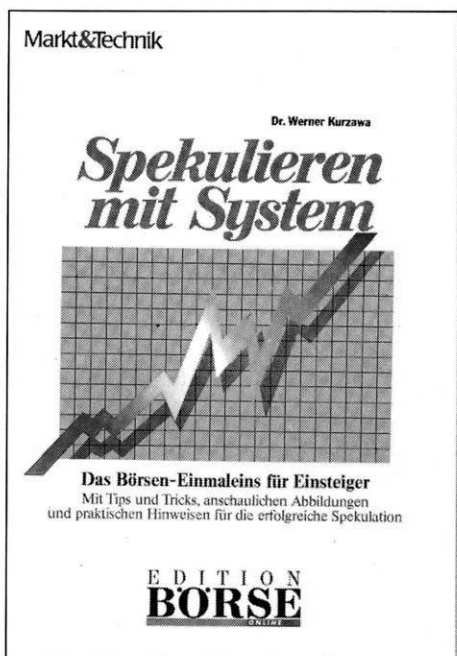
»Zieldiskette ist schreib-
geschützt!« 156

»Zugriff verweigert« 102

Spekulieren mit System

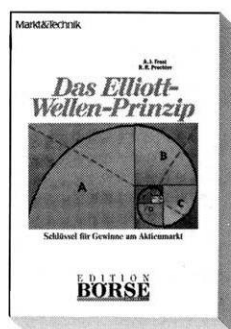


R. Schätzle/P.M. ten Cate
Computergestützte Anlageberatung
 Börsenerfolg ist im wesentlichen eine Frage von Informationsvorsprung und Prognosequalität. Bei beiden Faktoren bietet der PC-Einsatz einen hervorragenden Zeit- und Wissensvorsprung! Auf der beigefügten Diskette finden Sie fertige Programme für Framework II und III zum sofortigen Einsatz. Die behandelten Themen: Online-Kurse, Analyse und Prognose, Portfolio-Management, Adreß-Datenbank und Serienbrief, Zinsarbitrage und künstliche Intelligenz.
 1989, 140 Seiten, inkl. Diskette
 ISBN 3-89090-603-6
DM 69,-
 (sFr 63,50/öS 538,-)



Dr. W. Kurzawa
Spekulieren mit System
 Der Hauptgedanke dieses Buches ist es, den Lesern, die noch nichts oder nur wenig mit der Spekulation an der Börse zu tun

gehabt haben, den Weg dahin zu ebnen. Für Börsenneulinge und fortgeschrittene Börsianer.
 1989, 220 Seiten
 ISBN 3-89090-735-0
DM 49,-
 (sFr 45,10/öS 382,-)

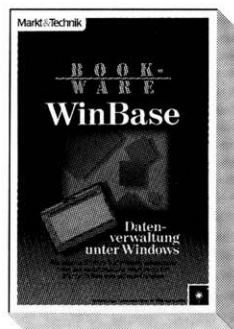


R. R. Prechter/A. J. Frost
Das Elliott-Wellen-Prinzip
 Das nach seinem Entdecker Ralph Nelson Elliott benannte Elliott-Wellen-Prinzip ermöglicht präzise und erfolgreiche Aktienkursprognosen. Sowohl Anfänger, die sich erstmalig mit der technischen Aktienanalyse beschäftigen, als auch erfahrene Analytiker werden nach der Lektüre dieses Werkes die Funktionsweise der Börse in einem neuen Licht sehen. Zahlreiche Illustrationen tragen dazu bei, dem Leser diese komplexe Theorie verständlich zu machen.
 1989, 224 Seiten
 ISBN 3-89090-719-9
DM 79,-
 (sFr 72,70/öS 616,-)



Markt&Technik-Produkte erhalten Sie bei Ihrem Buch- oder Computerfachhändler

BOOK- WARE



WinBase – das Datenbanksystem unter Windows

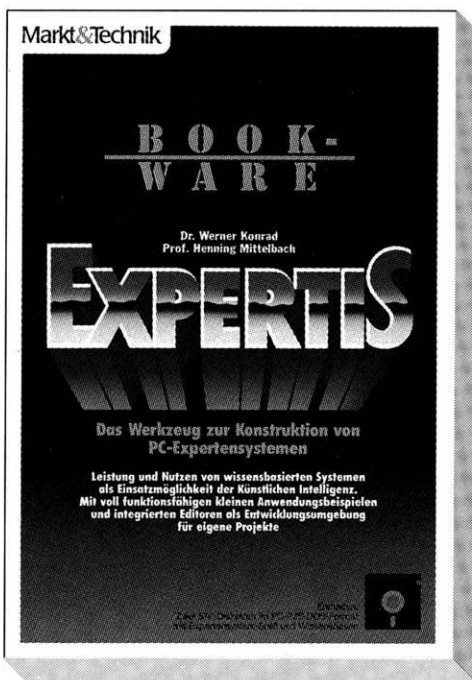
Nutzen Sie die Annehmlichkeiten von Windows auch für Ihre dBase-II-, dBase-III-, FoxBase- und Clipper-Dateien. WinBase ist ein eigenständiges Datenbanksystem, das unter Microsoft Windows angewendet wird und daher sehr komfortabel zu bedienen ist. Vergessen Sie lange, umständliche Kommandos, und benutzen Sie die Maus, um Ihre Datensätze zu sortieren, zu löschen, auszuwählen oder neu anzulegen.

Hardware-Anforderungen: PC/XT, PC/AT, PS/2 oder Kompatibles mit mindestens 512 Kbyte RAM.

Software-Anforderungen: Microsoft Windows 386 oder Windows 2.0 und höher.

1989, 115 Seiten,
inkl. Programmdiskette
ISBN 3-89090-801-2

DM 98,-*
(sFr 90,20*/6S 834,-*)



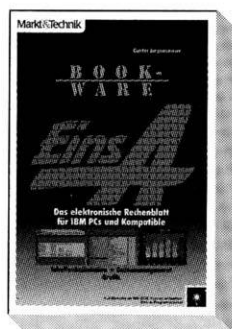
Dr. W. Konrad/
Prof. H. Mittelbach

Expertis

Expertis vermittelt Ihnen praktische Kenntnisse in der Konstruktion von Expertensystemen. Durch »Learning by Doing« üben Sie den Einsatz leistungsfähiger Wissensrepräsentationen und erstellen mit Hilfe einer professionellen Entwicklungsumgebung selbst Wissensbasen für Bereiche Ihrer Kompetenz.

1989, 147 Seiten,
inkl. 2 Disketten
ISBN 3-89090-750-4

DM 98,-*
(sFr 90,20*/6S 834,-*)



G. Jürgensmeier

Eins A

Eins A ist ein integriertes Programmpaket mit vielen Leistungsmerkmalen: Tabellenkalkulation mit 1024 Zeilen, 64 Spalten, 51 Rechenfunktionen und Tastaturmakros, Grafikprogramm mit Linien-, Kreis-, Balken- und XY-Diagrammen, Grafikdruck direkt im Programm und auch gemischt mit Tabellen, Datenmanagement mit Auswertungs-, Such- und Selektionsfunktionen, Makrosprache zur Automatisierung von Abläufen und zum Schreiben von Anwendungen, datei- und makrokompatibel zu Lotus 1-2-3 Version 1A, Eins A steht immer komplett im Speicher und ist auch mit nur einem Diskettenlaufwerk sinnvoll einsetzbar.

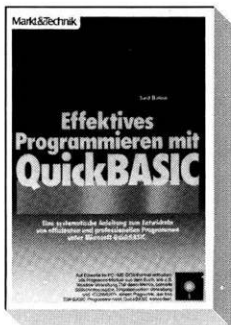
1989, 416 Seiten,
inkl. Programmdiskette
ISBN 3-89090-761-X

DM 98,-*
(sFr 90,20*/6S 834,-*)

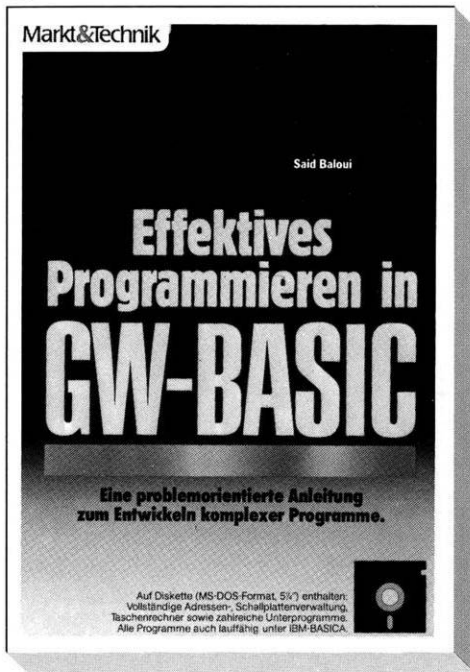


Markt & Technik-Produkte erhalten Sie bei Ihrem Buch- oder Computerfachhändler

Bücher zu Programmiersprachen

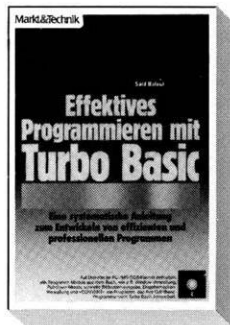


S. Baloui
Effektives Programmieren mit QuickBASIC
1988, 328 Seiten, inkl. Disk.
Eine systematische Anleitung zum Entwickeln von effizienten Programmen unter Microsoft QuickBASIC. Anhand von zahlreichen Beispielen wird der Umgang mit statistischen Variablen, dynamischen Arrays und den User-Libraries von QuickBASIC erläutert. Mit dem Programm Convert können Sie Ihre GW-Basic-Programme in QuickBasic-»Format« konvertieren. Alle entwickelten Programme sind auf der beiliegenden Diskette im Quell- und Objektcode vorhanden.
Bestell-Nr. 90532
ISBN 3-89090-532-3
DM 69,-
(sFr 63,50/öS 538,20)



S. Baloui
Effektives Programmieren in GW-Basic
1987, 420 Seiten, inkl. Disk.
Eine problemorientierte Anleitung für den Anwender mit Basic-Erfahrung. Sie werden in die Entwicklung komplexer Programme

eingeführt: von einfacher Eingaberoutine über professionelle Pull-down-Menüs bis zur Dateiverwaltung.
Bestell-Nr. 90464
ISBN 3-89090-464-5
DM 69,-
(sFr 63,50/öS 538,20)

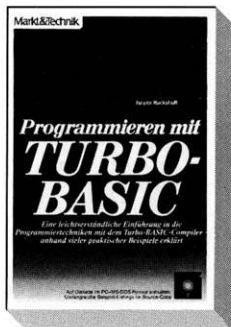


S. Baloui
Effektives Programmieren mit Turbo Basic
1988, ca. 300 Seiten, inkl. Diskette
Anhand umfangreicher Beispiele erläutert Ihnen der Autor detailliert den Umgang mit statischen Variablen, dynamischen Arrays und vor allem den INCLUDE-Libraries von Turbo Basic. Den Schwerpunkt des Buches bildet eine umfangreiche Modul-Sammlung: von komfortabler Eingaberoutine über die Verwaltung von Eingabemasken bis zur Programmierung von Windows und Pull-down-Menüs.
Bestell-Nr. 90636
ISBN 3-89090-636-2
DM 69,-
(sFr 63,50/öS 538,20)

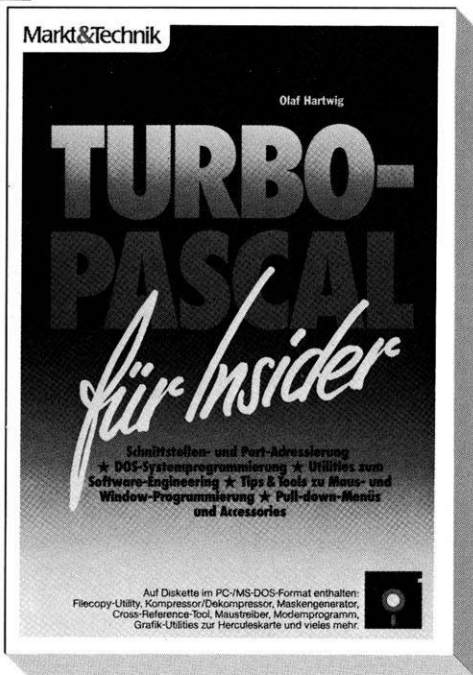
Markt & Technik
Zeitschriften · Bücher
Software · Schulung

Markt & Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

Bücher zu Programmiersprachen

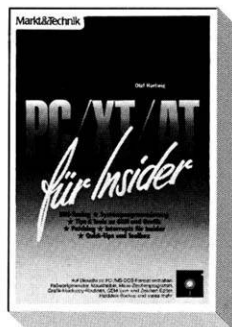


J. Hückstädt
Programmieren mit Turbo-BASIC
1987, 300 Seiten,
inkl. Diskette
Jeder BASIC-Programmierer, der bereits praktische Erfahrung mitbringt, wird mit diesem Buch schnell in Turbo-BASIC eingearbeitet. Sie finden hier nicht nur die allgemein bekannten Programmierertechniken, sondern werden auch in das strukturierte Programmieren eingeführt. Selbst wenn Sie Ihre Programme weiterhin in gewohnter Weise schreiben und mit Turbo-BASIC kompilieren, ist Ihnen diese Lektüre ein wertvoller Begleiter.
Bestell-Nr. 90542
ISBN 3-89090-542-0
DM 59,-
(sFr 54,30/6S 460,20)



O. Hartwig
Turbo-Pascal für Insider
1987, 274 Seiten,
inkl. Diskette
Eine wahre Fundgrube an Insiderwissen für jeden Turbo-Pascal-Anwender! Schwerpunkt-mäßig werden die Adressierung des Betriebs-

systems, direkte Schnittstellen- und Portadressierung, Utilities zum Software-Engineering, Grafik, Color und Sound, Tips & Tools behandelt.
Bestell-Nr. 90540
ISBN 3-89090-540-4
DM 59,-
(sFr 54,30/6S 460,20)

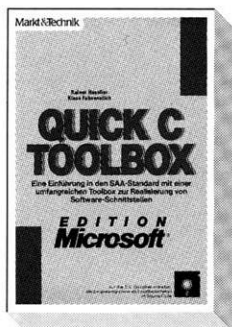


O. Hartwig
PC/XT/AT für Insider
1987, 272 Seiten,
inkl. Diskette
Sind Sie engagierter PC- und AT-Anwender? Interessieren Sie sich für Insiderwissen zu Ihrem System? Dann ist dieses Buch genau das Richtige für Sie! Schwerpunkt-mäßig werden das DOS-Tuning, die System- und Interruptprogrammierung, die direkte Modifizierung des GEM-Systems, universelle Grafik-Tools und die PC- und AT-Toolbox mit zahlreichen weiteren Utilities behandelt. Mit speziellen Tips zu Schneider-PC und -AT.
Bestell-Nr. 90541
ISBN 3-89090-541-2
DM 59,-
(sFr 54,30/6S 460,20)

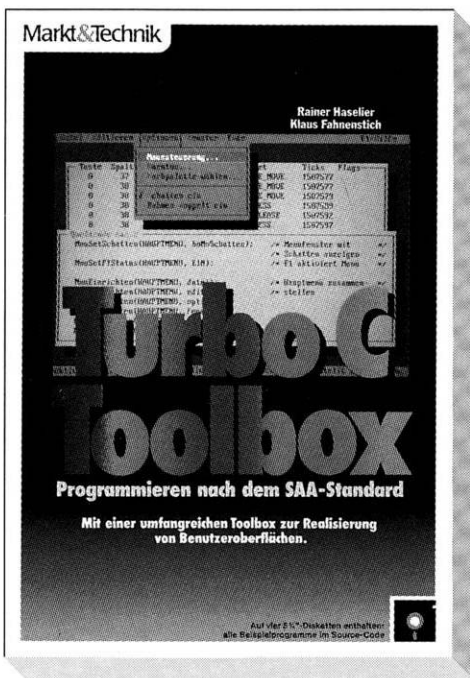


Markt & Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

Bücher zu Programmiersprachen

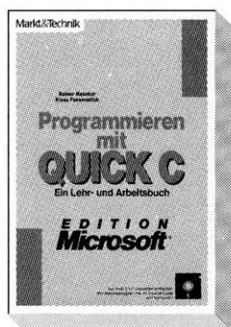


R. Haselier/K. Fahrenstich
QuickC Toolbox
Neben einer Beschreibung des Standards finden Sie alles, was Sie schon immer in Ihrer C-Bibliothek vermisst haben. Auf der Diskette finden Sie den C- und Assembler-Quellcode aller Funktionen und fertige Einzelbibliotheken für sämtliche Speichermodelle. Für QuickC Version 1.01 und Microsoft C Version 5.1.
• Für den fortgeschrittenen QuickC-Programmierer.
1989, 289 Seiten, inkl. vier 5 1/4"-Disketten
ISBN 3-89090-674-5
DM 98,-*
(sFr 90,20*/öS 834,-*)



R. Haselier/K. Fahrenstich
Turbo C Toolbox
Programmieren nach dem SAA-Standard. Mit einer umfangreichen Toolbox zur Realisierung von Benutzeroberflächen. Mit allen Bei-

spielen auf Diskette.
1989, 295 Seiten, inkl. vier 5 1/4"-Disketten
ISBN 3-89090-677-X
DM 98,-*
(sFr 90,20*/öS 834,-*)



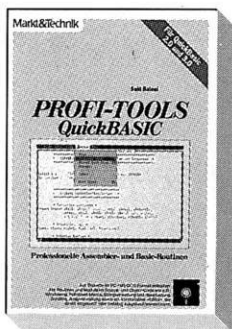
R. Haselier/K. Fahrenstich
Programmieren mit QuickC
Dieses Buch zeigt Ihnen, wie Sie mit QuickC schnell und komfortabel eigene Programme erstellen. Die Möglichkeiten der Hardware-nahen Programmierung werden Ihnen vorgestellt, und Sie erfahren u. a., wie Sie Interrupts von C aus aufrufen und Ihren Bildschirmspeicher direkt ansprechen können.
• Für den C-Einsteiger ebenso wie für den Fortgeschrittenen.
1988, 412 Seiten, inkl. zwei 5 1/4"-Disketten
ISBN 3-89090-609-5
DM 69,-
(sFr 63,50/öS 538,-)

*Unverbindliche Preisempfehlung



Markt & Technik-Produkte erhalten Sie bei Ihrem Buch- oder Computerfachhändler

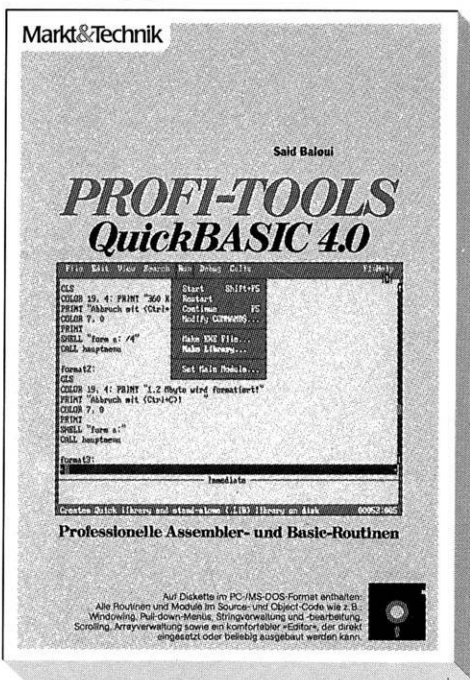
Bücher für jeden PC-Programmierer



S. Baloui

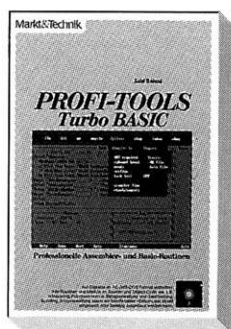
Profi-Tools QuickBasic Version 2.0/3.0

Dieses Buch beinhaltet 35 Routinen, die jeder QuickBasic-Programmierer haben sollte, der professionelle Ansprüche an seine Programme stellt. Schnelligkeit, Leistungsfähigkeit, Sicherheit, optimale optische Gestaltung und einfache Handhabung sind die Schlagwörter, die diese Tools charakterisieren. Alle Routinen und Module sind im Source- und Objekt-Code für QuickBasic Version 2.0/3.0 auf Diskette enthalten. 1988, 139 Seiten, inkl. Diskette
Bestell-Nr. 90615
ISBN 3-89090-615-X
DM 98,-*
(sFr 90,20*/6S 833,90*)



S. Baloui, Profi-Tools QuickBasic Version 4.0

35 professionelle Assembler- und Basic-Routinen. Die Routinen dieses Buches wurden ausschließlich für QuickBasic Version 4.0 entwickelt und sind alle auf der beigefügten Diskette enthalten. 1988, 152 Seiten, inkl. Diskette
Bestell-Nr. 90655, ISBN 3-89090-655-9
DM 98,-* (sFr 90,20*/6S 833,90*)



S. Baloui

Profi-Tools Turbo Basic

35 professionelle Assembler- und Basic-Routinen für jeden TurboBASIC-Programmierer. Alle Routinen und Module sind im Source- und Objekt-Code auf Diskette im PC-/MS-DOS-Format enthalten: z.B. Windowing, Pull-down-Menüs, Stringverwaltung und -bearbeitung, Scrolling, Arrayverwaltung sowie ein komfortabler »Editor«, der direkt eingesetzt oder beliebig ausgebaut werden kann. 1988, 141 Seiten, inkl. Diskette
Bestell-Nr. 90633
ISBN 3-89090-633-8
DM 98,-*
(sFr 90,20*/6S 833,90*)
* Unverbindliche Preisempfehlung



Markt & Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

Bücher zu Betriebs- systemen



C. Meyer

VMS leichtgemacht

Diese Einführung erspart dem Anfänger die Fülle der Informationen der Original-DEC-Dokumentation und führt ihn in die wesentlichen Grundlagen von VMS ab Version 4.0 ein. Ausgehend von den Funktionen des Terminals wird dargestellt, wie man Zugang zu einem VAX-System erlangt, um effektiv damit zu arbeiten. Sie erhalten Kenntnisse über die Organisation von Daten, wesentliche Kommandos der DCL, die Bedeutung von logischen Namen, Symbolen, Kommando-prozeduren und von Batch-Jobs. Zudem erfahren Sie, welche Hilfen das System zur Verfügung stellt und wie man diese am besten nutzt. 1988, 219 Seiten
ISBN 3-89090-662-1
DM 69,-
(sFr 63,50/öS 538,-)



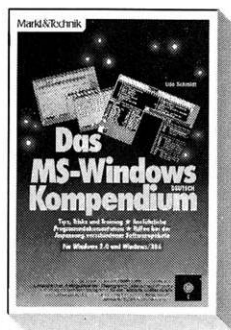
A. Deikman

Unix-Programmierung auf dem 80286/80386

Wenn Sie nach einem Schnellkurs für Unix auf einem 80286/80386-System suchen, dann ist

das vorliegende Buch von unschätzbarem Wert für Sie.

1989, 250 Seiten
ISBN 3-89090-724-5
DM 79,-
(sFr 72,70/öS 616,-)



U. Schmidt

Das MS-Windows-Kompendium

Tips, Tricks und Training, eine ausführliche Programm-dokumentation und Hilfen bei der Anpassung verschiedener Softwarepakete finden Sie in diesem umfangreichen Kompendium. 1988, 228 Seiten, inklusive 2 Disketten
ISBN 3-89090-558-7
DM 69,-
(sFr 63,50/öS 538,-)

H. Drees

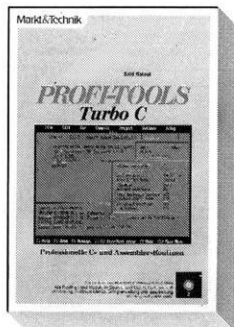
Unix-Handbuch

Dieses Unix-Handbuch beschreibt kein spezielles Unix, sondern Unix schlechthin und stützt sich dabei auf den Leistungs-standard, der durch Unix V repräsentiert wird. 1988, 570 Seiten
ISBN 3-89090-494-7
DM 89,-
(sFr 81,90/öS 694,-)

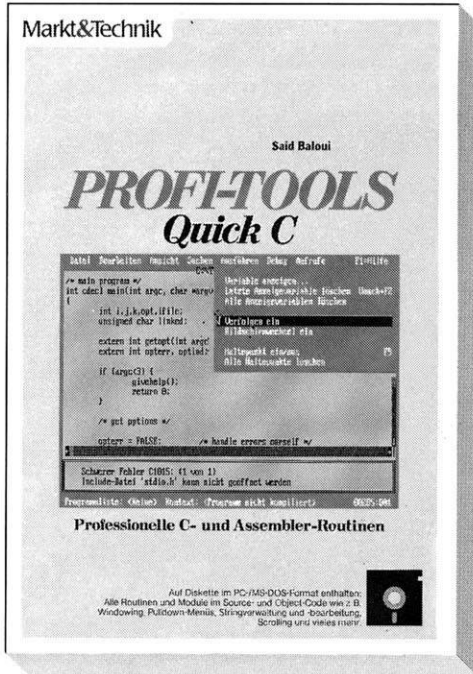


Markt&Technik-Produkte erhalten Sie bei Ihrem Buch- oder Computerfachhändler

PROFI-TOOLS

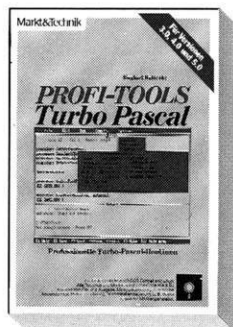


S. Baloui
Profi-Tools Turbo C
 Rund 60 Funktionen für den professionellen Programmierer: MYSTRING enthält häufig benötigte Stringfunktionen. Modul MYSTDLIB enthält Assembler-Routinen zur schnellen Stringausgabe, Bildschirmausschnitte in Variable kopieren, Errorhandling, komfortable Dateneingabe und vieles mehr. Modul MYFRAME bietet umfangreiche Funktionen zu Windowing und Menüverwaltung bis hin zur interaktiven Dateiauswahl in Directories. MYMNEU verwaltet Menüs, die baumartig aufgebaut sind und erlaubt die Einbindung eigener kontextsensitiver Hilfstexte.
 1988, 208 Seiten,
 inkl. Diskette
 ISBN 3-89090-691-5
DM 98,-
 (sFr 90,20/6S 834,-)
 Unverbindliche Preisempfehlung



S. Baloui
Profi-Tools QuickC
 Rund 60 Funktionen, die jeder QuickC-Programmierer, der professionelle Ansprüche an sein Programm stellt, haben sollte.

1988, 209 Seiten,
 inkl. Diskette
 ISBN 3-89090-692-3
DM 98,-
 (sFr 90,20/834,-)
 Unverbindliche Preisempfehlung



S. Heinecke
Profi-Tools Turbo Pascal 3.0/4.0
 Die zahlreichen Routinen aus diesem Buch erleichtern Ihnen die effizientere Programmentwicklung und modulare Gestaltung von Programmen. Die Toolbox enthält Routinen zur komfortablen Ein- und Ausgabe, zur Mengenrechnung, zu statistischen Anwendungen, zur Rechnung mit komplexen Zahlen, zur Matrizenrechnung, zur Steuerung der Schnittstellen sowie vollständige Anwendungen aus den Bereichen Mathematik und Statistik.
 1988, 211 Seiten,
 inkl. Diskette
 ISBN 3-89090-665-6
DM 98,-
 (sFr 90,20/6S 834,-)
 Unverbindliche Preisempfehlung



Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser



Computerliteratur und Software vom Spezialisten

Vom Einsteigerbuch für den Heim- oder Personalcomputer-Neuling über professionelle Programmierhandbücher bis hin zum Elektronikbuch bieten wir Ihnen interessante und topaktuelle Titel für

• Apple-Computer • Atari-Computer • Commodore 64/128/16/116/Plus 4 • Schneider-Computer • IBM-PC, XT und Kompatible

sowie zu den Fachbereichen Programmiersprachen • Betriebssysteme (CP/M, MS-DOS, Unix, Z80) • Textverarbeitung • Datenbanksysteme • Tabellenkalkulation • Integrierte Software • Mikroprozessoren • Schulungen. Außerdem finden Sie professionelle Spitzen-Programme in unserem preiswerten Software-Angebot für Amiga, Atari ST, Commodore 128, 128 D, 64, 16, für Schneider-Computer und für IBM-PCs und Kompatible!

Fordern Sie mit dem nebenstehenden Coupon unser neuestes Gesamtverzeichnis und unsere Programmservice-Übersichten an, mithilfe freier Utilities, professionellen Anwendungen oder packenden Computerspielen!

Adresse:

Name

Straße

Ort

Bitte schicken Sie mir:

- ☐ Ihr neuestes Gesamtverzeichnis
☐ Eine Übersicht Ihres Programmservice-Angebotes aus der Zeitschrift

- ☐ Außerdem interessiere ich mich für folgende/n Computer:

(P.S. Wir speichern Ihre Daten und verpflichten uns zur Einhaltung des Bundesdatenschutzgesetzes)



709005

Markt & Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon (089) 46 13-0

Markt & Technik Verlag AG
- Unternehmensbereich Buchverlag -
Hans-Pinsel-Straße 2
D-8013 Haar bei München

Bücher zu

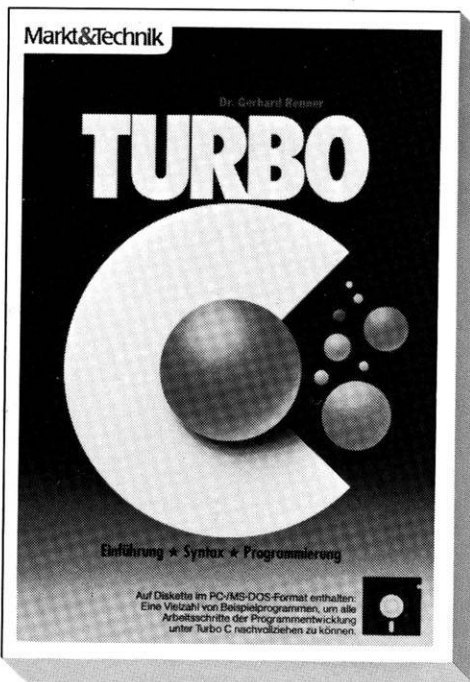


A. Holub

C für Kenner

Dr. Dobb's Journal of Software Tools, eine amerikanische Fachzeitschrift steht mit diesem Buch jetzt auch dem deutschen Leser zur Verfügung. Es wendet sich an alle, die mit der Sprache C vertraut und daran interessiert sind, die Möglichkeiten zur Schaffung einer umfangreichen und mächtigen Software-Bibliothek für fast jeden Zweck zu nutzen. Das Buch ist nicht nur eine wertvolle Hilfe für C-Programmierer, es hebt den Leser in Sachen Programmierertechnik um Klassen höher. Die Qualität amerikanischer Software ist bekannt. Bücher wie dieses sind mit ein Grund für solchen programmtechnischen Vorsprung.

1988, 609 Seiten,
inkl. 2 Disketten
ISBN 3-89090-639-7
DM 98,-
(sFr 90,20/öS 764,-)



Dr. G. Renner

Turbo C

Eine schrittweise Einführung in die Programmierumgebung von Turbo C.

1988, 312 Seiten,
inkl. Diskette
ISBN 3-89090-536-6
DM 59,-
(sFr 54,30/öS 460,-)



D.E. Krekelberg/

G.E. Sobelman

C für Profis

Mit diesem Buch liegt ein Werk vor, das weit über das übliche Handbuchwissen hinausgeht und anhand von professionellen Anwendungen in die fortgeschrittene Arbeit mit C einführt. Ausgehend von den in C zur Verfügung stehenden Sprachelementen werden Themen wie Programmierstil, Zeiger, Rekursion, Strukturen, Listen und Bäume behandelt. Der Leser kann seine Programmierertechniken anhand vieler Beispiele gezielt vertiefen. Die grundlegenden Programmierkonzepte werden anschließend bei der Entwicklung von zwei umfangreichen Unterprogramm-Bibliotheken in die Praxis umgesetzt. 1987, 293 Seiten, inkl. Diskette
ISBN 3-89090-426-2
DM 69,-
(sFr 63,50/öS 538,-)


Markt & Technik
Zeitschriften · Bücher
Software · Schulung

Markt & Technik-Produkte erhalten Sie bei Ihrem Buch- oder Computerfachhändler

DOS 3.3 für PCs und Personal System/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.

IBM liefert mit ihren Personalcomputern und dem neuen Personal System/2 die aktuelle Betriebssystem-Version DOS 3.3 aus. Derzeit werden über 90 Prozent aller PCs weltweit mit DOS betrieben. Zu diesem leicht erlernbaren Betriebssystem steht Ihnen die größte Auswahl an Programmen aus den verschiedensten Bereichen zur Verfügung. Zielsetzung dieses Buches ist es, Ihnen eine verständliche Einführung in das Betriebssystem DOS 3.3 zu bieten. In kompakter Form werden Ihnen die wichtigsten Bestandteile von DOS übersichtlich mit vielen Übungsbeispielen vorgestellt. Dabei wendet sich der Autor nicht nur an den unerfahrenen Anfänger, sondern ebenso an den Fortgeschrittenen und Profi. Dazu wurde das Buch in drei Abschnitte gegliedert:

1. Im ersten Teil finden Sie eine Einführung in DOS, die dem unerfahrenen Anwender grundlegende Kenntnisse vermittelt. Weiterhin enthalten sind eine kurze Einweisung in die Hard- und Software Ihres Computers sowie Hinweise zu Anwendungsmöglichkeiten.
2. Der zweite Teil enthält eine alphabetisch geordnete Übersicht aller DOS-Befehle mit ausführlicher Beschreibung und vielen Übungen. Durch seine übersichtliche Gestaltung wird dieser Abschnitt für den fortgeschrittenen Anwender zu einem hilfreichen Nachschlagewerk für die tägliche Arbeit am PC.
3. Im dritten Teil erhalten Sie Hinweise und Ratschläge zur Anpassung von DOS an Ihr Computersystem (vor allem CONFIG.SYS mit ANSI.SYS)

und führt Sie in die Programmierung von Stapelprogrammen (Batch-Dateien) ein.

Abgerundet wird das Buch durch einen umfangreichen Anhang mit Begriffserklärungen, Zeichencode-Tabellen, einer Übersicht zu Unterschieden und Neuerungen zwischen den DOS-Versionen 2.11, 3.0 bis 3.3, einer Übersicht bekannter Fehler in DOS 3.3 sowie jeweils einer Übersicht zu den »DOS Function Calls« und den »DOS Interrupt Calls«.

Hardware-Anforderungen:

Personalcomputer PC/XT/AT oder ein Modell der Personal System/2-Serie.

Software-Anforderungen:

Betriebssystem PC-DOS von IBM oder MS-DOS von Microsoft in Version 3.3.

ISBN N 3-89090-547-1



DM 69,-
sFr 63,50
öS 538,20