

Computer 1
Persönlich

CP/M und WORDSTAR



Anwender-Handbuch

Eine leichtverständliche Einführung in die
meistverbreiteten Mikrocomputer-Betriebssysteme
CP/M und MP/M

R. Paul/M. Riedel

Markt & Technik

CP/M und WORDSTAR

Anwender-Handbuch

Deutschesprachiges Handbuch für die
Betriebs-Systeme CP/M und MP/M und das
Textverarbeitungs-System WordStar

Rüdiger Paul · Martin Riedel

Verlag Markt & Technik

Paul, Rüdiger:

**CP-M und WordStar—Anwender-Handbuch/
Rüdiger Paul; Martin Riedel.—Haar bei München:
Markt & Technik Verlagsgesellschaft, 1981.**

ISBN 3-922120-10-5

NE: Riedel, Martin:

© Copyright 1981 bei te-wi Verlag GmbH

LIZENZAUSGABE FÜR:

Markt & Technik Verlagsgesellschaft,
8013 Harr bei München

Alle Rechte vorbehalten. Ohne ausdrückliche, schriftliche Genehmigung der Herausgeber ist es nicht gestattet, das Buch oder Teile daraus in irgendeiner Form durch Fotokopie, Mikrofilm oder ein anderes Verfahren zu vervielfältigen oder zu verbreiten. Dasselbe gilt für das Recht der öffentlichen Wiedergabe.

Die Herausgeber übernehmen keine Gewähr dafür, daß die beschriebenen Schaltungen, Baugruppen, Verfahren usw. funktionsfähig und frei von Schutzrechten Dritter sind.

CP/M und MP/M sind Warenzeichen von
DIGITAL RESEARCH.

WordStar ist Warenzeichen der MicroPro
Intern. Corp.

Z80 ist Warenzeichen der ZILOG Inc.

6502 ist Warenzeichen von MOS-Technology.

8080 und 8085 sind Warenzeichen von INTEL.

GESAMTHERSTELLUNG;

technik marketing, München

tm 3459/1183

Printed in W-Germany

ISBN 3-922120-10-5

Vorwort

Die zunehmende Anwendung der Mikrocomputer-Technologie in der kommerziellen Datenverarbeitung während der letzten Jahre kann nicht darüber hinwegtäuschen, daß die Entwicklung im Systemsoftware-Bereich im Allgemeinen nicht mit der ständigen Steigerung der Hardware-Leistungsfähigkeit Schritt halten konnte.

Nur wenige für den Anwender akzeptable Lösungen konnten sich in diesem problematischen Bereich etablieren.

Eine von ihnen ist das Mikrocomputer-Betriebs-System CP/M. Es ist heute Dank seiner vielfältigen Möglichkeiten und einem großen, darauf abgestimmten Software-Angebot das am weitesten verbreitete Betriebs-System

für Mikrocomputer. Es existieren mittlerweile CP/M-Versionen für fast alle gebräuchlichen Computer-Systeme, die auf den Mikroprozessoren 8080, 8085 oder Z80 basieren, ebenso für manche 6502-Mikrocomputer.

Dem Trend zur Dezentralisierung der Dateneingabe in der EDV wurde Rechnung getragen, indem aus CP/M das Multi-User-Betriebs-System MP/M entwickelt wurde, welches in Verbindung mit geeigneter Hardware einen komfortablen Time-Sharing-Betrieb gewährleistet.

Dem wachsenden Interesse am Einsatz von Mikrocomputern zur bequemen und rationellen Bearbeitung des in steigendem Maße anfallenden Schriftgutes kommt das CP/M-kompatible Textverarbeitungs-System "WordStar" entgegen.

Einfache Handhabung trotz großer Variations-Möglichkeiten in der Textgestaltung, als auch leichte Anpaßbarkeit an verschiedene Drucker und Sichtgeräte kennzeichnen dieses System.

Besonders in Deutschland ist in Analogie zur "Software-Krise" eine "Dokumentations-Krise" entstanden, die bei vielen Anwendern, deren "Muttersprache" nicht Englisch ist, zu Verdruß und begrenzter Ausnutzbarkeit ihrer wertvollen Software führt.

Hier soll dieses Handbuch für CP/M-Anwender Abhilfe schaffen, ohne daß besonderes Fachwissen vorausgesetzt wird.

Der schnellste Weg, mit der Software vertraut zu werden, ist dabei stets das Ausprobieren verschiedener Möglichkeiten in der Praxis am Computer. Das gilt in besonderem Maße für WordStar, das bei der Anwendung viele hilfreiche Benutzerinformationen bietet.

Für die Erprobung der Kommandos stand uns das Mikrocomputersystem CMS 8000 der Firma Computer Modular zur Verfügung.

Hamburg, Mai 1981

Rüdiger Paul
Martin Riedel

Inhalt

	Seite
1. DIE BETRIEBSSYSTEME CP/M UND MP/M	1
1.0 Allgemeine Einleitung und Bedienungshinweise	1
1.0.1 Das Datenverarbeitungs-System	1
1.0.2 Die Zentraleinheit	2
1.0.3 Massenspeicher	5
1.0.4 Die Konsole	6
1.0.5 Der Drucker	7
1.0.6 Die Software (Programme und Daten) ...	8
1.0.7 CP/M und MP/M	11
1.0.8 System-Initialisierung	13
1.0.9 Umgang mit Disketten	13

1.1	CP/M Kurzbeschreibung	18
1.1.1	Leistung und Aufbau des CP/M 2.	18
1.1.2	Die Kommandostruktur des CP/M 2.	19
1.1.3	Bezeichnung von Dateien	20
1.1.4	Die Control-Funktionen	21
1.2	Einführendes Beispiel	23
1.3	Residente Kommandos	
1.3.1	Allgemeines	26
1.3.2	Das "DIR"-Kommando	27
	(Auflisten des Inhaltsverzeichnisses einer Disk)	
1.3.3	Das "ERA"-Kommando	27
	(Löschen von Dateien)	
1.3.4	Das "REN"-Kommando	28
1.3.5	Das "TYPE"-Kommando	29
	(Ausgabe von Dateien auf den Bildschirm)	
1.3.6	Das "SAVE"-Kommando	30
	(Speichern des Programmspeicherinhalts auf Disk)	
1.3.7	Das "USR"-Kommando	32
1.4	Transiente Kommandos	33
1.4.1	Allgemeines	33
1.4.2	Das "PIP"-Kommando	33
	(Kopieren von Dateien)	
1.4.3	Das "STAT"-Kommando	45
	(Informationen und Gerätezuordnung)	
1.4.4	Das "SUBMIT"-Kommando	50
	(serielles Abarbeiten von CP/M- Kommandos)	

1.4.5	Das "ASM"-Kommando	54
	(Aufruf und Ausführung des CP/M-Assemblers)	
1.4.6	Das "LOAD"-Kommando	55
1.5	ED: Der Text-Editor des CP/M-Betriebs-Systems	56
1.5.1	Einführung, Funktion des Editors	56
1.5.2	Text-Übertragung	59
1.5.3	Text-Manipulation	60
1.5.4	Suchen und Ersetzen	64
1.5.5	Einfügen von Dateien	66
1.5.6	Block-Move	67
1.5.7	Mehrfache Kommando-Wiederholung	67
1.5.8	Fehlermeldungen	68
1.5.9	Zusammenfassung	69
1.6	Das Multi-User-Betriebssystem MP/M	73
1.6.1	Allgemeines	73
1.6.2	Das "CONSOLE"-Kommando	74
	(Information über Konsolennummern)	
1.6.3	Das "DSKRESET"-Kommando	75
	(Information der Benutzer über Disk-Wechsel)	
1.6.4	Das "ABORT"-Kommando	75
	(Abbruch laufender Programme)	
1.6.5	Das "SPOOL"-Kommando	76
	(Aufbau von Warteschlägen für Peripherie)	
1.6.6	Das "SCHED"-Kommando	77
	(Programmausführung von vorgegebenen Zeitpunkten)	

1.6.7	Das "TOD"-Kommando	77
	(Initialisierung und Anzeige der Zeit)	
1.6.8	Das "ERAQ"-Kommando	78
	(Löschen von Dateien in MP/M)	
1.6.9	Das "DIR"-Kommando	79
	(Auflisten der Inhaltsverzeichnisse in MP/M)	
1.6.10	Das "TYPE"-Kommando	79
	(Anzeigen von Dateien auf der Konsole)	
1.6.11	Das "ATTACH"-Kommando	79
	(Hintergrundverarbeitung in MP/M)	

	Seite
2. WORDSTAR: EIN MENU-GESTEUERTES VISUELLES TEXTVERARBEITUNGS-SYSTEM FÜR CP/M	81
2.1 Übersicht über die Möglichkeiten	81
2.2 Disk-Initialisierung für WordStar	88
2.3 Starten und NO-FILE-Kommandos	89
2.4 Das Schirmbild	93
2.5 Editier-Kommandos	100
2.5.1 Cursor-Bewegung	100
2.5.2 Schirmbild rollen (Scrolling)	103
2.5.3 Text-Eingabe	104
2.5.4 Text löschen	105
2.5.5 Sichern und Beenden	106
2.5.6 Schirmbild-Textformatierung	107
2.5.7 Suchen und Ersetzen	110
2.5.8 Marken setzen	112
2.5.9 Block-move-Kommandos	113
2.5.10 Zusatz-Files	114
2.5.11 Help-Kommandos	115
2.6 Druck-Funktionen	116
2.6.1 Druck-Kontroll-Zeichen	116
2.6.2 Dot-Kommandos	117
2.7 Fehlermeldungen	121

Die Betriebssysteme CP/M und MP/M

1.0 ALLGEMEINE EINLEITUNG UND BEDIENUNGSHINWEISE

1.0.1 Das Datenverarbeitungssystem

Das Datenverarbeitungssystem läßt sich prinzipiell in Hardware und Software unterteilen. Die Software besteht aus Programmen, während die Hardware alle gerätetechnischen Komponenten eines Rechnersystems umfaßt.

Die minimale Konfiguration für die Hardware eines Datenverarbeitungssystems der Mikrocomputertechnologie besteht aus:

- der Zentraleinheit, oft in Verbindung mit integrierten Laufwerken für flexible Magnetschicht-Datenträger (Disketten).
- der Konsole (auch Terminal genannt), bestehend aus Bildschirm und Eingabetastatur.

Mit diesen Geräten ist, in Kombination mit der geeigne-

ten Software, ein einwandfreier Datenverarbeitungs-Betrieb möglich.

In der Praxis wird die Geräte-Konfiguration meistens um einen Drucker erweitert.

Außerdem besteht in Zusammenhang mit den geeigneten Programmen die Möglichkeit, mehr als eine Konsole mit der Zentraleinheit zu verbinden.

Dem Wunsch nach der Verarbeitung und Speicherung besonders großer Datenmengen wird entsprochen, wenn anstelle der Diskettenlaufwerke externe Laufwerke für feste Magnetplatten ("Hard Disks") verwendet werden. Außerdem besteht insbesondere für die Datensicherung die Möglichkeit, ein Magnetbandgerät für die Speicherung von großen Datenmengen einzusetzen.

Früher gebräuchliche Ein-/Ausgabegeräte, wie Lochstreifenleser/-stanzer oder Lochkartenleser/-stanzer sind heute wenig gebräuchlich.

1.0.2 Die Zentraleinheit

Die Zentraleinheit ist der wesentliche Teil des gesamten Systems. Sie enthält als Herzstück den Mikroprozessor, eine in hochintegrierter Technologie hergestellte, komplette logische Verarbeitungseinheit. Der Mikroprozessor stellt hauptsächlich das Rechen- und das Steuerwerk des Computers dar.

Das Rechenwerk (ALU = Arithmetic Logic Unit) führt arithmetische Operationen, logische Verknüpfungen, Verschieben ("Shiften"), Vergleichen, Rundungen und andere Vorgänge aus. Hier werden Bit-Gruppen parallel verarbeitet. Bei vielen Mikroprozessoren sind dies 8-Bit-Worte, d.h. die Prozessoren sind 8-bit-parallel organisiert.

Das Steuerwerk ist das komplizierteste Teil des Mikroprozessors. Es liefert die Steuersignale, die für den logischen Ablauf der Operationen notwendig sind und bewirkt dadurch die Durchführung der im abgespeicherten Programm niedergelegten Verarbeitungsvorgänge.

Weitere Bestandteile der Zentraleinheit sind Register (schnelle, kleine Speichereinheiten), Taktgenerator (übernimmt die Synchronisation der Operationen des Systems) und die Systemsteuerung, welche die aus dem Steuerwerk kommenden Steuersignale aufbereitet, bevor sie den Prozessor verlassen.

Außerdem gehören zur Zentraleinheit noch in Halbleitertechnologie ausgeführte Speicher. Man unterscheidet zwischen RAM- und ROM-Bausteinen.

ROM-Bausteine (ROM = Read Only Memory) sind Speicher, aus denen gelesen, aber in die nicht geschrieben werden kann. Der Speicherinhalt wird beim Abschalten der Versorgungsspannung nicht gelöscht und kann ebenfalls nicht irrtümlich geändert werden. Diese Speicher heißen deshalb auch "Festwertspeicher".

Anders verhält sich dies bei den RAM-Bausteinen (Random Access Memory): in sie können Daten immer wieder neu eingeschrieben werden, indem z.B. alter Speicherinhalt gelöscht wird. RAM dienen daher vorzugsweise zum Speichern von Programmen und Daten.

Ein weiterer wesentlicher Bestandteil der Zentraleinheit ist die Eingabe-/Ausgabeeinheit die den Datenaustausch mit den peripheren Geräten (siehe auch Kapitel 1.0.3 – 1.0.5) ermöglicht.

Zur Verbindung der einzelnen Komponenten (Prozessor, Speicher und Systemsteuerung) wird das Bus-Prinzip angewandt. Busse sind vieladrige, parallel angeord-

nete Verbindungsleitungen, an welche die Bausteine des Computers gleichberechtigt angeschlossen sind. Die Busse werden wiederum unterteilt, je nachdem, ob sie Daten, Speicheradressen oder Steuersignale transportieren (siehe auch Bild 1).

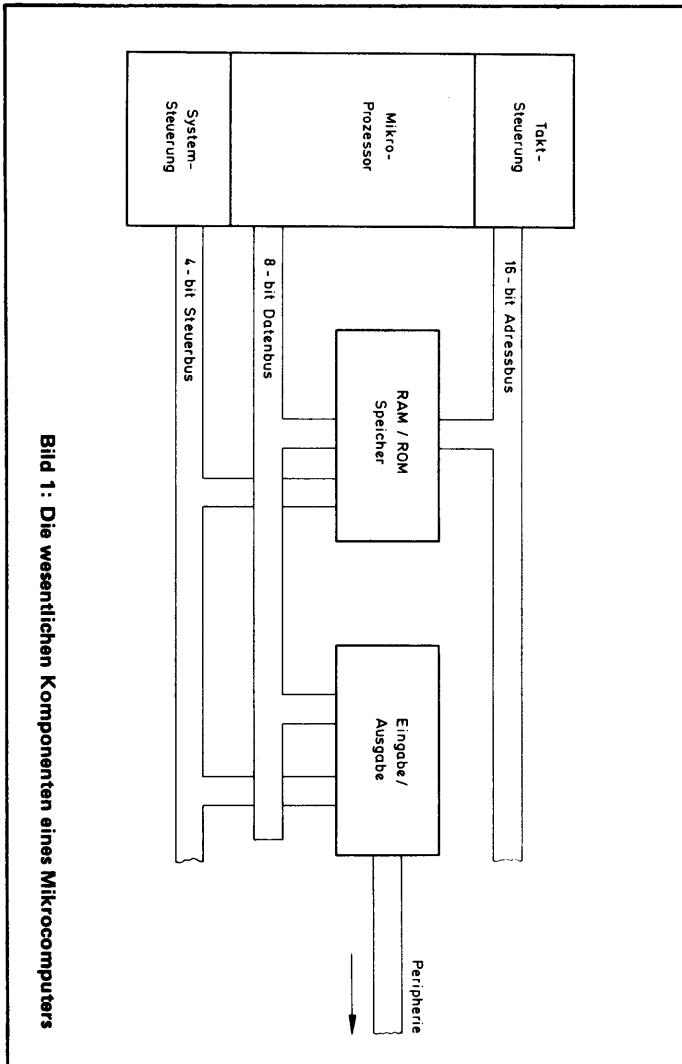


Bild 1: Die wesentlichen Komponenten eines Mikrocomputers

1.0.3 Die Massenspeicher

Da die im vorherigen Kapitel erwähnten Halbleiterspeicher zwar schnell im Zugriff sind, dafür aber eine begrenzte Kapazität besitzen (und damit pro zu speichernder Einheit teuer sind) verwendet man für die Speicherung von Programmen und Daten sogenannte Massenspeicher. So sind, wie später noch näher erläutert wird, nur einige wenige Teile des Betriebssystems CP/M permanent im Hauptspeicher (resident), während ein wesentlicher Teil sich auf den angeführten Massenspeichern befindet und um zur Ausführung gelangen zu können, vom Massenspeicher in den RAM-Bereich geladen werden muß, um dort nur so lange zu verweilen, bis dieser Teil nicht mehr benötigt ist und von anderen Programmteilen überschrieben, d.h. gelöscht wird (transient, d.h. flüchtig im Hauptspeicher).

Massenspeicher sind heute im allgemeinen in Magnetschichttechnik ausgeführt. Die zu speichernde Information wird hier auf einem magnetisierbarem Material aufgezeichnet, das sich gegenüber einem Magnetkopf in Bewegung befindet. Die magnetisierbare Schicht wird von einem Trägermedium gehalten. Dieses Medium kann entweder flexibel (Magnetband, Disketten, siehe auch 1.0.9) oder starr sein (Magnetplatten, sogenannte "Hard Disks", oder auch Magnettrommelspeicher).

Magnetschichtspeicher können sehr große Datenmengen bei relativ geringen Kosten aufnehmen. Die Zugriffszeit ist allerdings mechanisch bedingt (entweder durch die Bandgeschwindigkeit des Magnetbandes oder durch die Bewegung des Zugriffsarms, ähnlich eines Tonarms beim Plattenspeicher, bei Magnetplatten und Disketten).

Da Magnetplatten ("Hard Disks") und Disketten ("flexible Disks") durch die notwendige Möglichkeit des quasi wahlfreien Zugriffs (d.h. eine Information kann direkt über einen Such-Algorithmus gefunden werden) heute bei Mikrocomputern die größte Verwendung finden und Bandgeräte allenfalls zur schnellen, rein physischen Sicherung von Daten dienen, wird in diesem Buch ausschließlich die Speicherung auf den zuerst genannten Speichermedien beschrieben. Hard Disks und Disketten werden im folgenden unter dem Begriff "Disks" zusammengefaßt.

1.0.4 Die Konsole

Die Konsole ist ein peripheres Gerät für den interaktiven Betrieb, d.h. durch sie wird der Dialogbetrieb mit dem Computer ermöglicht. Dazu besteht die Konsole aus einer Ausgabe- und einer Eingabeeinheit.

Die Ausgabereinheit ist das Sichtgerät. Das Sichtgerät ist eine Einrichtung zur digitalen Darstellung von Daten auf einem Bildschirm. Die Daten werden mit einer Kathodenstrahlröhre (CRT = Cathode Ray Tube) auf einer Mattscheibe analog dem Fernsehbild auf dem Fernsehschirm sichtbar gemacht.

Die Eingabeeinheit wird durch eine Tastatur, ähnlich der Schreibmaschinentastatur realisiert. Mit ihr werden Zeichen zunächst eingegeben und sofort auf dem Bildschirm dargestellt. Erst wenn mit Hilfe dieser Darstellung die Richtigkeit der Eingabe überprüft wurde, wird mittels einer speziellen Taste die Eingabe an die Zentraleinheit "abgeschickt". Die Tastatur enthält meistens die Ziffern 0 - 9, das Alphabet (groß, klein), Sonderzeichen (, . / ; : * etc.) und spezielle Funktionstasten,

die durch Programme belegt werden können. Die Stelle auf dem Bildschirm, auf der das nächste eingetastete Zeichen erscheinen wird, markiert in der Regel ein Cursor (oder auch Pointer) genannt, intermittierender Lichtpunkt. Er bewegt sich bei der Zeicheneingabe von Schreibstelle zu Schreibstelle, kann aber auch durch Leertaste ("Blank") oder Tabulator bewegt werden.

1.0.5 Der Drucker

Unter dem Begriff "Drucker" wird hier ein speziell für Datenverarbeitungsanlagen gebautes Druckausgabegerät hoher Leistung (Zeichen/Sekunde) für die Druckausgabe auf Papier verstanden.

In der Mikrocomputertechnologie finden vor allem mechanische Drucker, welche das Druckbild durch Anschlagen oder Andrücken erzeugen, als periphere Geräte Verwendung.

Zwei Arten von Druckern finden hauptsächlich Anwendung:

- der Matrixdrucker (auch Mosaik- oder Nadeldrucker genannt).

Die gedruckten Zeichen werden mosaikartig aus einzelnen Punkten gebildet.

Der Druckkopf besteht aus matrixförmig angeordneten Stiften ("Nadeln", z.B. 7 x 9- oder 9 x 9-Matrix), aus denen jeweils die für die Bildung eines Zeichens notwendigen zunächst gegen das Farbband und dann gegen das Papier geschoben werden.

Die Druckrate liegt zwischen 80 und 200 Zeichen pro Sekunde. Da das Druckbild aus einzelnen Punkten zusammengesetzt ist, sieht man ihm immer an, daß es ein

Computerausdruck ist. Da dieses vor allem in der Textverarbeitung von Nachteil ist (z.B. bei Geschäftsbriefen, wird der langsamere

– Typenraddrucker verwendet. Das vertikal angeordnete Typenrad dieses Druckers besteht aus sternförmig angeordneten Typenhebeln. Soll nun ein Zeichen zu Papier gebracht werden, so wird das Typenrad in Position gedreht, so daß die gewünschte Type vor ein Anschlaghämmerchen kommt, welches dieses gegen Farbband und Papier preßt. Das Druckergebnis ist, aufgrund der Verwendung von Typen, bei guten Druckern nicht von dem einer elektrischen Schreibmaschine zu unterscheiden.

Neben den angeführten Druckern gibt es ausgesprochene Schnelldrucker, bei denen die Typen auf einer schnell rotierenden Walze oder Kette angebracht sind.

1.0.6 Die Software (Programme und Daten)

Die Software des Computers, die in Speicher-ICs (Integrierten Bausteinen), auf Disketten oder anderen Speichermedien konserviert wird, läßt sich grob in Programme und Daten unterteilen. (Siehe Bild 2).

Dabei besteht ein Programm aus einer Folge von mehr oder weniger komplexen Befehlen, aus denen die Zentraleinheit erkennt, was sie mit den Daten zu tun hat. Die Daten werden dazu in den Computer eingegeben, als Zwischen-Ergebnisse gespeichert, oder als Ausgabe dem Benutzer mitgeteilt (siehe Bild 2). Diese Daten können z.B. Zahlen, Buchstaben oder ganze zusammenhängende Texte sein.

Programme sind entweder System-Software oder Anwender-Software (Bild 2). System-Software dient dazu,

dem Benutzer eine möglichst komfortable Bedienung und optimale Nutzung des Computers zu ermöglichen. Z.B. gehört das Betriebssystem CP/M zur System-Software. Ohne ein Betriebssystem gäbe es gar keine Möglichkeit, mit dem Computer in Kontakt zu treten, es ist daher für den Benutzer obligatorisch.

Anwender-Software sind Programme, die spezielle, vom Computer-Anwender gestellte Aufgaben erfüllen. Hier ist gerade in Verbindung mit dem CP/M-Betriebssystem die Auswahl sehr groß. Ein komfortables Anwender-Programm ist z.B. das Textverarbeitungs-System WordStar.

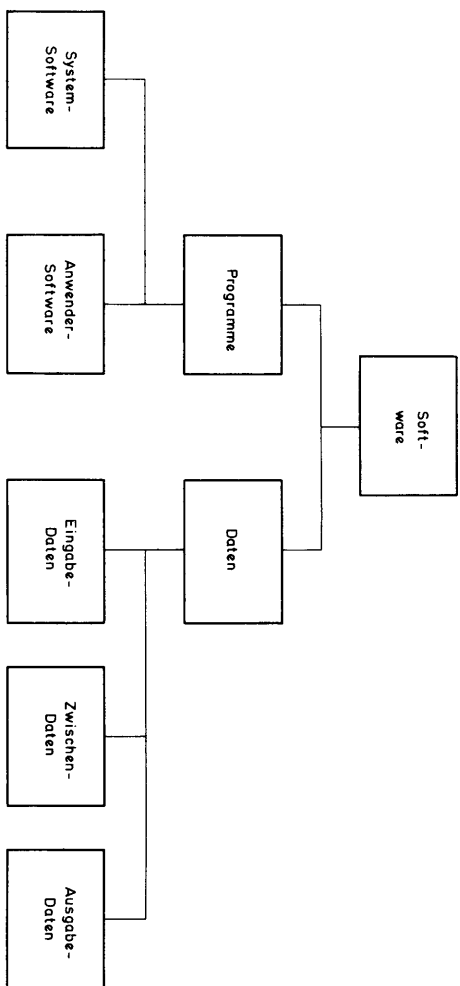


Bild 2

Ein Programm oder eine abgeschlossene Einheit von Daten bezeichnet man als Datei oder File. Unter dem Namen der File tritt sie z.B. im Inhaltsverzeichnis (Directory) einer Diskette in Erscheinung.

1.0.7 CP/M und MP/M

CP/M und MP/M sind Betriebs-Systeme, sie gehören also zur System-Software.

CP/M bedeutet "Control Program for Microprocessors" und MP/M steht für "Multiprogramming Control Program for Microprocessors".

Beide haben innerhalb eines Mikrocomputers vielfältige Aufgaben zu erfüllen:

- Erkennen und Ausführen sämtlicher Benutzer-Kommandos
- Steuerung der Peripherie-Geräte wie Konsole, Drucker etc.
- Verwaltung der Programme, Daten und Freiräume (Space) auf Disketten, Magnetplatten (Hard-Disk) und im Arbeitsspeicher (RAM) der Zentraleinheit.

Zusätzlich dazu muß MP/M noch die Prozessor-Zeit unter den verschiedenen Benutzern aufteilen (Time-Sharing-Verfahren) und jedem die gewünschten Programme und Daten zur Verfügung stellen.

CP/M kann also nur ein Sichtgerät steuern, während es in MP/M möglich ist, von mehreren Konsolen aus gleichzeitig mit dem Computer zu arbeiten (siehe Bild 3).

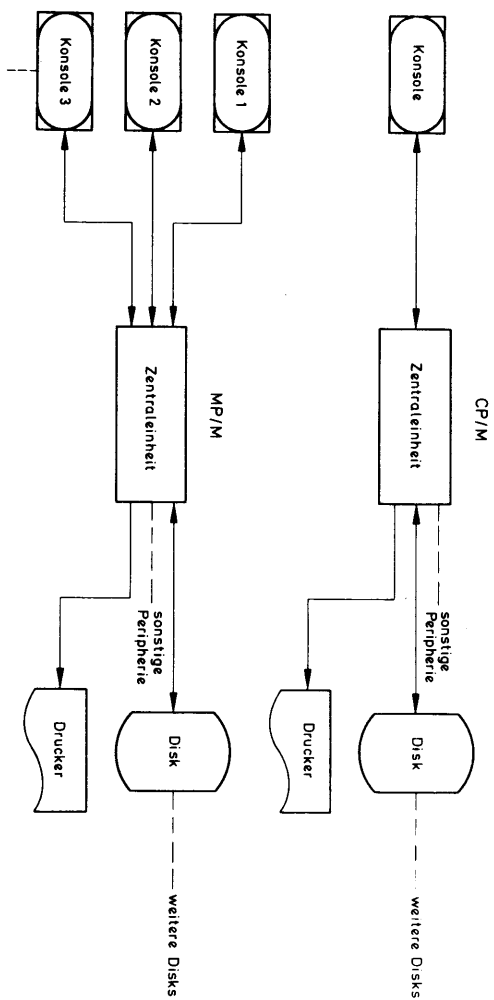


Bild 3

1.0.8 System-Initialisierung

Im Allgemeinen befindet sich das Betriebssystem nach dem Einschalten des Computers und der Peripherie-Geräte noch nicht im Speicher der Zentraleinheit. Diese enthält lediglich, fest gespeichert (ROM), ein winziges Programm, den "Bootstrap Loader". Wird dieses Programm durch Drücken einer Taste (Reset-Taste), Drehen eines Schlüssels oder auch schon durch das Einschalten der Zentraleinheit gestartet, so lädt es automatisch das Betriebssystem von der Disk in den Hauptspeicher (Auto-Load-Vorgang) und startet es. Diesen Vorgang nennt man Kaltstart; dabei werden alle vorher im Arbeitsspeicher vorhandenen Programme und Daten gelöscht.

Für diese Prozedur muß sich natürlich die entsprechende Disk (System-Disk) mit dem Betriebs-System in dem vom Hersteller bestimmten Laufwerk befinden. (Zur Behandlung von Disketten siehe "1.0.9 Umgang mit Disketten"!).

Nach dem Kaltstart kann der Benutzer beliebige Betriebssystem-Kommandos eingeben oder Programme starten.

Im Anschluß an jedes Kommando muß die "RETURN" Taste der Konsole gedrückt werden. Erst dadurch wird der Befehl vom Sichtgerät an die Zentraleinheit abgeschickt.

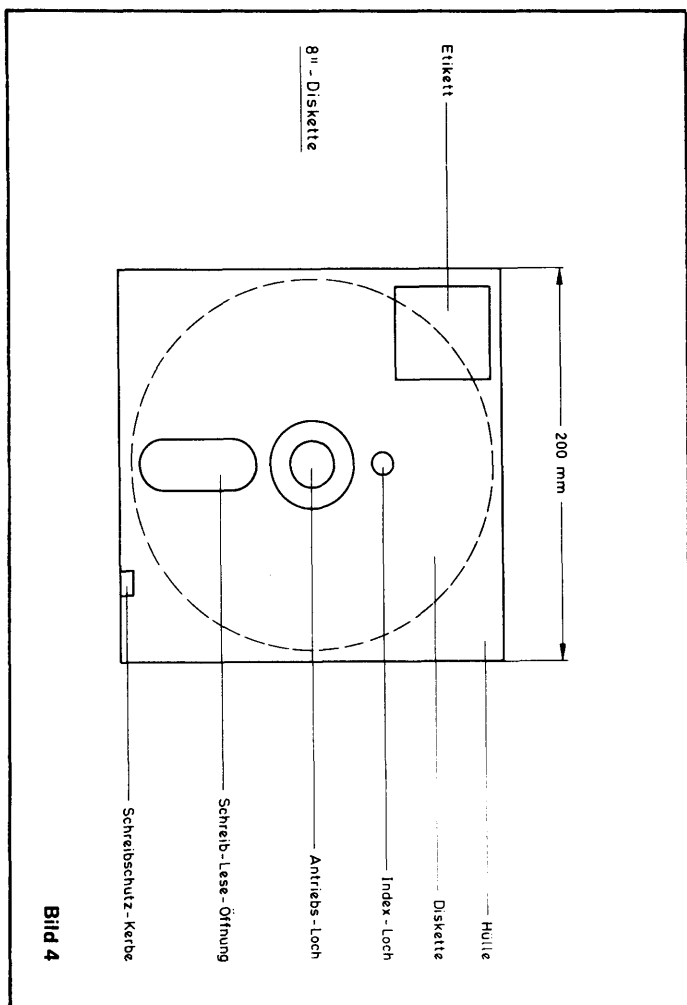
Kann das Kommando von CP/M (MP/M) nicht identifiziert werden, so erscheint es auf dem Bildschirm noch einmal, gefolgt von einem Fragezeichen.

1.0.9 Umgang mit Disketten

Da Disketten (Floppy-Disks) bei Mikrocomputern als preiswertes Massenspeicher-Medium am weitesten ver-

breitet sind, sei hier noch einiges über sie gesagt.

Disketten sind kreisrunde, flexible, magnetisierbare Folien, die sich in einer schützenden Papier- oder Kunststoff-Hülle befinden. Die Hülle hat die für den Zugriff, den Antrieb und das Abtasten eines Index-Loches notwendigen Öffnungen (siehe Bild 4).



Es gibt Disketten mit 8" und solche mit 5,25" Durchmesser. Letztere werden auch Mini-Floppy genannt.

Zur Bearbeitung wird die Diskette in ein spezielles Laufwerk eingelegt, wo sie mit hoher Geschwindigkeit rotiert und von einem Magnetkopf abgetastet wird, der die in magnetischer Form verschlüsselten Informationen liest oder schreibt.

Bei 8"-Disketten ist i.A. eine Kerbe als Schreibschutz-Markierung in der Hülle vorhanden, über der sich eine Aluminiumfolie befindet. Wird die Folie entfernt, so kann die Diskette nicht mehr beschrieben oder gelöscht werden.

Umgekehrtes gilt für die 5,25"-Floppys. Hier muß man die Aluminiumfolie entfernen, um das Beschreiben und Löschen der Diskette zu ermöglichen.

Im Vergleich zu Hard-Disks sind Disketten billig aber relativ langsam. Hat ein Programm große Datenmengen auf Disketten zu bearbeiten, so machen sich die längeren Zugriffszeiten für den Anwender deutlich bemerkbar.

Disketten sind zwar recht robust, aber bei falscher Behandlung durch verschiedene Ursachen zerstörbar. Folgende Hinweise sollten daher unbedingt beachtet werden:

- Diskette nicht magnetischen Feldern aussetzen, wie sie z.B. von Transformatoren, Fernsehern, Lautsprechern, Werkzeugen oder einem Telefon ausgesendet werden können!
- Nicht die Disketten-Folie berühren!
- Nicht knicken!

- Nicht direkt auf die Hülle schreiben, die Disketten-Folie könnte dadurch eingedrückt werden! Besser erst ein Etikett beschreiben und dieses anschließend auf die Hülle aufkleben.
- Disketten vor Staub, Kratzern und Feuchtigkeit schützen!

Je nach dem verwendeten System wird die Diskette von einer oder von beiden Seiten beschrieben. In jedem Fall muß sie immer richtig herum in das Laufwerk eingelegt werden.

Ob sich das Etikett dabei oben oder unten, bzw. rechts oder links befinden muß, ist hersteller-abhängig.

Zum Ausprobieren sollte man auf jeden Fall eine Kopie der wertvollen System-Diskette verwenden, nicht etwa das einzige Original!

Der Computer bzw. die Disketten-Laufwerke dürfen nur ausgeschaltet werden, nachdem alle Disketten entnommen wurden! Ein Ausschalten, während sich noch eine Diskette in einem Laufwerk befindet, könnte die auf ihr gespeicherten Informationen zerstören.

Von jeder erarbeiteten File sollte eine Kopie auf einer anderen Diskette aufbewahrt werden, um bei Verlust der Original-File (z.B. durch Darübergießen von Kaffee oder durch einfachen Verschleiß) auf dieses Doppel zurückgreifen zu können. Diese Datensicherung – also das Kopieren geänderter Daten auf eine zweite Diskette (oder ein Magnetband) – sollte so oft wie möglich erfolgen, um im Zweifelsfalle immer eine möglichst neue Version der betreffenden File zur Verfügung zu haben.

Des weiteren sollten nie Disketten ihren Laufwerken entnommen werden, solange auf ihnen eine Verarbei-

tung stattfindet oder etwas ausgedruckt wird. Es ist ratsam, Disketten nur zu tauschen, wenn sich das Betriebssystem im Ausgangszustand befindet, sich also mit der Bezeichnung des aktuellen (eingeschalteten) Laufwerks gemeldet hat.

Beim Erstellen großer Files ist unbedingt zu beachten, daß eine Diskette nur begrenzten Speicherraum zur Verfügung stellen kann. Das bedeutet z.B. bei einer einseitig (Single Sided) und mit einfacher Schreibdichte (Single Density) zu beschreibenden 8"-Diskette: 64 Files mit insgesamt maximal 241.000 Zeichen.

Man sollte sich daher stets mit Hilfe des STAT-Kommandos (siehe 1.4.3) davon überzeugen, daß noch genügend Speicherplatz zur Verfügung steht; sonst kann es passieren, daß sich eine in den Arbeitsspeicher eingegebene (editierte) File nicht mehr auf die Diskette bringen läßt und verloren geht.

1.1 CP/M-KURZBESCHREIBUNG

1.1.1 Leistung und Aufbau des CP/M 2.

CP/M ist ein weitverbreitetes Betriebssystem für Mikrocomputer. Es unterstützt Hardware, die auf dem Mikroprozessor Z-80 bzw. Z-80A von ZILOG oder dem 8080 von INTEL als Zentralprozessor basiert.

CP/M enthält eine umfassende Dateiorganisation für Disks (Disketten oder Hard-Disks), die einen schnellen Zugriff auf Programme und Daten gestattet. Dateiverarbeitung ist sowohl mit sequentiell als auch wahlfreiem Zugriff möglich. Der Text-Editor des CP/M 2. gestattet es, Dateien zu kreieren, bzw. gegebenenfalls zu ändern.

Das Betriebssystem CP/M 2. läßt sich logisch in vier Segmente aufteilen, welche im Folgenden kurz dargestellt werden.

BIOS – Basic Input/Output-System

BIOS ermöglicht den Zugriff auf die Disks und gewährleistet Funktion und Zusammenspiel der anderen Peripheriegeräte (Konsole, Drucker) mit der Hardware.

BDOS – Basic Disk-Operating-System

BDOS organisiert den Dateiaufbau (Formatierung der Disks). Die Anzahl der Sätze pro Datei ist ausschließlich durch die Kapazität der Disk begrenzt. Eine Single-Density-Diskette kann beispielsweise bis zu 64 Dateien mit 241.000 Zeichen aufnehmen.

CCP – Consol Command Processor

CCP stellt die Verbindung zwischen der Konsole und den weiteren Hardware-Komponenten her.

TPA – Transient Program Area

TPA speichert Programme, die durch CCP von der Diskette in den Arbeitsspeicher geladen werden.

1.1.2 Die Kommandostruktur des CP/M 2.

Die Kommandos des Betriebssystems CP/M 2. lassen sich grundsätzlich in zwei Gruppen unterteilen:

1. Kommandos, welche Bestandteil des Betriebssystems sind und zu Beginn mit dem "Autoload-Vorgang" von der Diskette in den Hauptspeicher geladen werden (Residente Kommandos). Diese Kommandos heißen:

- ERA Löschen von Dateien
- DIR Auflisten von Dateibezeichnungen einer Disk
- REN Umbenennung von Dateien
- SAVE Speicherinhalt in Maschinensprache auf einer Datei sichern
- TYPE Ausgabe des Inhalts einer Datei von der Disk auf die Konsole

Die Kommandos beziehen sich jeweils auf eine zu spezifizierende Datei bzw. auf mehrere Dateien.

2. Kommandos, welche Programme von der Platte in den TPA-Bereich laden, wie beispielsweise "ED", der Aufruf des Editors, oder "STAT", das Kommando, welches die spezifizierten Dateien in bestimmte Zustände versetzt (Read Only, z.B.) oder dem Benutzer Informationen erteilt. Diese Kommandos heißen "Transiente Kommandos".

Die Kommandos können eine maximale Länge von 255 Zeichen haben. Sie sind erst an die Zentraleinheit abgeschickt, wenn die "RETURN"-Taste betätigt wurde.

1.1.3 Bezeichnung von Dateien

Die Dateibezeichnung hat in Verbindung mit Kommandos die Aufgabe, eine Datei oder eine Gruppe von Dateien auf der Disk zu identifizieren.

Die Dateibezeichnung setzt sich aus zwei Teilen zusammen: dem Dateinamen und dem Dateityp. Der Dateiname ist frei zu wählen und darf aus maximal acht Zeichen bestehen. Der Dateityp besteht aus maximal drei Zeichen, welche dem Dateinamen angefügt werden und durch einen Punkt von ihm getrennt sind. Er ist in gewissen Grenzen ebenfalls frei wählbar, doch ist es zweckmäßig, sich an bestimmte Konventionen zu halten.

Für eine Datei, die ein FORTRAN-Programm enthält, sollte man beispielsweise den Dateityp "FOR" verwenden. Der Dateityp kann auch wegfallen. Eine Dateibezeichnung hat daher allgemein folgendes Aussehen:

⟨dateiname⟩ . ⟨dateityp⟩

Soll sich ein Kommando auf eine Datei beziehen, welche nicht auf dem momentan aktivierten Laufwerk vorhanden ist, so stellt man der Dateibezeichnung die Laufwerksbezeichnung (z.B. "A" oder "B") gefolgt von einem Doppelpunkt voran:

d: ⟨dateiname⟩ . ⟨dateityp⟩

Für die gesamte Bezeichnung einer Datei sind alphanumerische Zeichen und Sonderzeichen verwendbar. Eine

Ausnahme bilden die Zeichen < > . , ; : = ? * []. Sie dürfen nicht verwendet werden.

Die Bezeichnung einer Datei könnte daher z.B. folgendermaßen aussehen:

DATEI10.FOR

Die Dateibezeichnung kann in bestimmten Kommandos auch zweideutig gewählt werden, um mit einem Befehl mehrere Dateien ansprechen zu können. Dazu wird an eine oder mehreren Stellen der Bezeichnung ein Fragezeichen oder ein Stern gesetzt. Das Kommando bezieht sich dann auf alle Dateien, die an der Position des oder der Fragezeichen ein bzw. mehrere beliebige Zeichen haben. "DATEI??.FOR" würde im Zusammenhang mit einem Befehl alle Dateien ansprechen, die an der Position der Fragezeichen beliebige Zeichen als Bezeichnung haben, also beispielsweise "DATEI03.FOR" oder "DATEI23.FOR".

Wird der Stern (*) als Dateityp in bestimmten Kommandos verwendet, so werden alle Dateien angesprochen, die den im Befehl verwendeten Dateinamen enthalten. Die Bezeichnung "DATEI03.*" identifiziert alle Dateien mit dem Dateinamen "DATEI03" und beliebigen Dateitypen, also beispielsweise "DATEI03.BAK" oder "DATEI03.FOR". Auf die gleiche Art ist es auch möglich, den Dateinamen durch einen Stern zu ersetzen.

Wird als Dateibezeichnung *.* verwendet, d.h. werden sowohl Dateiname als auch Dateityp durch Sterne ersetzt, so werden alle auf der Diskette vorhandenen Dateien angesprochen. Verwendet man " *.* ." als Dateibezeichnung im Kommando "ERA" (Löschen von Datei-

en), ist Vorsicht geboten, da dann unter Umständen alle Dateien einer Disk gelöscht werden.

1.1.4 Control-Funktionen

Die "CONTROL"-Funktionen dienen u.a. dem Editieren von Kommandozeilen in CP/M 2. Sie werden durch das gleichzeitige Betätigen der "CONTROL"-Taste und einer weiteren realisiert. Folgende Möglichkeiten bestehen für diese Funktionen:

TASTE (in Verbindung mit CONTROL)	Wirkung
U	hebt die Wirkung aller seit dem letzten "RETURN" eingegebenen Zeichen auf. Kommandos können weiter eingegeben werden.
X	Löscht die letzte eingegebene Kommandozeile.
E	Zeilenwechsel ohne Abschicken des Kommandos (kein "RETURN").
C	Warmstart des Betriebssystems wird durchgeführt. Betriebssystem meldet sich mit "A)", "B)" oder Anderen. Diese Funktion ist vor allem anzuwenden, wenn sich das Betriebssystem nach der Ausführung eines Kommandos nicht wieder meldet. Außerdem gibt es die Möglichkeit, die "RESET"-Taste der Zentraleinheit zu betätigen ("Kaltstart").
P	Einschalten der Hardcopy-Funktion. Zeichen, die auf dem Bildschirm ausgegeben werden, werden gleichzeitig ausgedruckt.
S	Die Bildschirmausgabe wird gestoppt. (z.B. beim "TYPE"-Kommando). Die Ausgabe weiteren Dateiinhalts geschieht durch die Eingabe eines beliebigen Zeichens der Tastatur.

1.2 EINFÜHRENDES BEISPIEL

Dieses Beispiel soll die wichtigsten Grundkenntnisse im Umgang mit dem Betriebssystem CP/M 2. vermitteln.

Das Betriebssystem muß geladen sein; es meldet sich dann mit "A>". Diese Zeichen besagen, daß die Disk in Laufwerk A aktiviert ist und CP/M 2. auf die Eingabe von Kommandos wartet. Außerdem muß für die beispielhafte Kommandofolge in Laufwerk B eine Disk eingelegt werden. Alle Kommandos werden mit "RETURN" abgeschlossen und abgesendet. In allen folgenden Kommandos wird dies zur Vereinfachung nicht mehr explizit erwähnt.

Wenn das Betriebssystem sich nach der Eingabe eines Kommandos einmal längere Zeit nicht wieder meldet, so führen Sie einen Warmstart durch, indem Sie die Tasten "CONTROL" und "C" gleichzeitig drücken. Auch das Betätigen der "RESET"-Taste ("Kaltstart") an der Zentraleinheit schafft Abhilfe. Diese sogenannte "Dead-Lock-Situation" des Betriebssystems wird von Zeit zu Zeit wieder auftreten; diese ist aber selbst bei Betriebssystemen für Großrechner üblich und ist auf nicht beseitigte Fehler im Programm zurückzuführen.

Nun das Beispiel:

Kommandos	Wirkung
DIR	Alle auf der aktivierten Diskette vorhandenen Dateien werden auf dem Bildschirm aufgelistet. So bedeutet beispielsweise: A: F80 COM :L80 COM, daß auf Diskette A Dateien mit der Bezeichnung F80.COM und L80.COM vorhanden sind. Wir erinnern uns, daß "F80" den Dateinamen und "COM" den Dateityp darstellt.

Kommandos	Wirkung
B:	Disk "B" wird aktiviert. Das Betriebssystem meldet sich mit "B)" und wartet auf weitere Kommandos.
DIR	Dieses uns schon bekannte Kommando listet alle Dateibezeichnungen der aktivierten Disk (nun also "B") auf. Sollten Sie eine leere Disk in "B" eingelegt haben, so antwortet CP/M 2. mit "NO FILE".
A:	Wir aktivieren wieder Disk "A". Das System antwortet mit "A)".
ED BEISPIEL	Der Text-Editor wird aufgerufen und eine Datei mit dem Namen "BEISPIEL" erzeugt. Dazu muß der Editor (die Datei "ED.COM") auf Disk "A" vorhanden sein. Das Kommando "ED" gehört zu der 2. Art Kommandos, die auf Seite 18 erwähnt wurden (Transientes Kommando). Der Editor meldet sich mit "***". Wenn wir eine Datei eröffnen, so wird die Meldung "NEW FILE" auf der Konsole erzeugt.
I	Dieses Kommando wird in der Editorbeschreibung näher erläutert. Der Editor meldet sich mit "1:"; alle folgenden Zeichen werden in die 1. Zeile unserer Datei "BEISPIEL" geschrieben. Schreiben Sie: "DIESES IST EINE UEBUNG".
ctl-Z (beide Tasten zugleich)	Dieser Befehl bedeutet das Ende der Datei "BEISPIEL". Der Editor meldet sich mit "***".
E	Sie verlassen den Editor. Das Betriebssystem meldet sich wieder mit "A)".
DIR	Es werden alle auf Disk "A" vorhandenen Dateibezeichnungen aufgelistet. Auch die soeben erzeugte Datei "BEISPIEL" ist vorhanden. Außerdem wurde zum Zweck der Datensicherung eine Datei BEISPIEL.BAK erzeugt (Dateityp:BAK).

Kommandos	Wirkung
TYPE BEISPIEL	Der Inhalt der Datei "BEISPIEL" wird auf der Konsole ausgegeben. Es erscheint "DIESES IST EINE UEBUNG" auf dem Bildschirm.
PIP CON:=BEISPIEL	Dieses Kommando hat die gleiche Wirkung wie das vorhergehende. "CON" steht für Konsole.
PIP LST:=BEISPIEL	Dieses Kommando bewirkt, daß der Inhalt der Datei "BEISPIEL" auf den Drucker ausgegeben wird. Dazu muß der Drucker natürlich eingeschaltet sein und auf ON-LINE stehen.
PIP B:=BEISPIEL	Die Datei "BEISPIEL" wird von Disk "A" nach Disk "B" kopiert. Dabei wird sie auf "A" nicht gelöscht.
DIR B:	Die auf Disk "B" vorhandenen Dateien werden aufgelistet, ohne daß man vorher auf Disk "B" umschalten muß, d.h. Laufwerk "A" ist immer noch aktiviert. Das Betriebssystem meldet sich wieder mit "A)". Auf Disk "B" befindet sich jetzt die Datei "BEISPIEL".
REN UEBUNG=BEISPIEL	Die Datei "BEISPIEL" auf Disk "A" wird in "UEBUNG" umbenannt. Der Inhalt der Datei wird nicht berührt.
DIR	Auf Disk "A" befinden sich jetzt die Dateien "UEBUNG" und "BEISPIEL.BAK".
PIP B:=UEBUNG	Kopieren der Datei "UEBUNG" von Disk "A" nach Disk "B".
B:	Umschalten auf Laufwerk "B".
DIR	Auf Disk "B" befinden sich jetzt die Dateien "BEISPIEL" und "UEBUNG".
ERA BEISPIEL	Löschen der Datei "BEISPIEL" auf Disk "B". Vorsicht beim Löschen von Dateien, damit nichts unbeabsichtigt gelöscht wird. Auf Disk "B" befindet sich jetzt nur noch die Datei "UEBUNG".

Kommandos	Wirkung
	Sollte die Datei auf Read/Only geschaltet sein, so gibt das System die Meldung "Bdos Err On A: File R/O" aus; Sie müssen dann auf Laufwerk "A" umschalten und mittels des Kommandos "STAT B:BEISPIEL \$R/W" die Datei wieder zum Schreiben bzw. Löschen freigeben.

Ende des einführenden Beispiels.

Falls man den Mini-Computer ausschalten will, so ist zu beachten, daß die Disketten aus den Laufwerken zuvor entnommen werden müssen, da sonst die Gefahr der Zerstörung des Inhalts der Disketten besteht!

1.3.1 Allgemeines

Die Residenten Kommandos befinden sich durch den Auto-Load-Vorgang des CP/M 2. im Hauptspeicher und sind jederzeit verfügbar.

Das Betriebssystem meldet sich mit "A>". Diese Zeichen besagen, daß das Disklaufwerk "A" aktiviert ist und CP/M 2. auf Kommandos wartet. Geben Sie Ihre Kommandos unmittelbar nach "A>" ein und beenden Sie sie mit "RETURN". Dies ist notwendig, um den Befehl von der Konsole an die Zentraleinheit abzuschicken. Es wird daher im Folgenden nicht mehr explizit erwähnt.

Wenn Sie vom Laufwerk "A" auf das Laufwerk "B" umschalten wollen, so geben Sie das Kommando "B:" (bzw. "C:", "D:" usw.) ein. Das Betriebssystem meldet sich dann mit dem aktivierten Laufwerk "B" (bzw. "C", "D" usw.). Geben Sie wieder "A:" (oder eine andere Laufwerksbezeichnung) ein um auf "A" (oder ein anderes Laufwerk) umzuschalten.

1.3.2 Das "DIR"-Kommando

Dieses Kommando in der Form

`"DIR<dateibezeichnung>"`

bewirkt, daß auf der aktivierten Disk die Datei mit der in dem Kommando angegebenen Dateibezeichnung auf der Konsole aufgelistet wird, falls diese auf der aktivierten Disk vorhanden ist. Ist sie nicht vorhanden, so meldet CP/M 2.: "NO FILE".

Die Dateibezeichnung kann auch mehrdeutig gewählt werden, d.h. an die Position der für die Dateibezeichnung gültigen Zeichen wird das Fragezeichen "?" bzw. der Stern "*" gesetzt. Es werden dann alle Dateien aufgelistet, die in diese mehrdeutige Bezeichnung passen und auf der Disk des aktivierten Laufwerks vorhanden sind (siehe auch Abschnitt 1.1.3).

Das Kommando "DIR *.*" hat die gleiche Wirkung wie "DIR" und bewirkt eine Auflistung der Dateibezeichnungen aller Dateien der momentan aktivierten Disk. Ist keine Datei vorhanden, so meldet das Betriebssystem "NO FILE".

Es besteht die Möglichkeit, den Inhalt der momentan nicht aktivierten Disk aufzulisten (ohne auf das andere Laufwerk umzuschalten). Dieses geschieht z.B. mittels des Kommandos "DIR B:", falls man das Laufwerk "A" zur Zeit aktiviert hat, bzw. "DIR A:", wenn "B" oder andere Laufwerke zur Zeit eingeschaltet sind.

1.3.3 Das "ERA"-Kommando

Das "ERA"-Kommando dient zum Löschen von Dateien.

Die Form des Kommandos lautet:

"ERA d:(dateibezeichnung)"

Für die Dateibezeichnung sind alle im Kapitel 1.1.3 (Bezeichnung von Dateien) erwähnten Regeln anwendbar. Beim "ERA-Kommando" ist Vorsicht geboten, um Dateien nicht unbeabsichtigt zu zerstören. Insbesondere gilt dies für das Kommando "ERA *.*", da dieser Befehl alle Dateien löscht. CP/M 2. trägt diesem Risiko Rechnung, indem es zunächst keine Datei löscht, wenn "ERA *.*" eingegeben wird, sondern die Meldung "ALL (Y/N)?" auf den Bildschirm ausgibt. Die Eingabe von "Y" (Y steht für yes), gefolgt von "RETURN", bewirkt die endgültige Ausführung des Kommandos, also das Löschen aller Dateien. Wird hingegen "N" (für "NO") eingegeben, so wird der Befehl nicht ausgeführt.

1.3.4 Das "REN"-Kommando

Das "REN"-Kommando gestattet es dem Benutzer, die auf der Disk vorhandenen Dateien umzubenennen. Die allgemeine Form des Kommandos ist:

REN <alte dateibezeichnung> = <neue dateibezeichnung>

Wird beispielsweise

REN NEU.FOR=ALT.FOR

eingegeben, so wird die Datei "NEU.FOR" umbenannt in "ALT.FOR". Die Datei "ALT.FOR" muß dafür auf der aktivierten Disk vorhanden sein, sonst meldet das Betriebssystem CP/M 2. "NO FILE". Die beiden Bezeichnungen der Dateien im "REN"-Kommando müssen eindeutig gewählt werden, d.h. es dürfen keine "?" und keine "*" verwendet werden. Die Datei "ALT.

FOR" ist nach der Ausführung des Kommandos nicht mehr auf der Disk vorhanden, ihr Inhalt existiert nur noch unter der Dateibezeichnung "NEU.FOR".

Es besteht die Möglichkeit, die beiden Dateibezeichnungen im "REN"-Kommando auf Dateien im momentan nicht aktivierten Laufwerk zu beziehen. Das folgende Beispiel soll dieses veranschaulichen (Laufwerk "A" sei aktiviert):

— REN B:NEU.FOR=ALT.FOR

Die Datei "ALT.FOR" befindet sich auf Laufwerk "B". Sie wird in "NEU.FOR" umbenannt.

"NEU.FOR" befindet sich nun ebenfalls auf Disk "B".

Bei der Anwendung des "REN"-Kommandos ist zu beachten, daß sich sowohl die alte Dateibezeichnung, als auch die neue auf derselben Disk befinden.

Sollte die neue Dateibezeichnung schon existieren, so erscheint auf dem Sichtgerät die Meldung "FILE EXISTS".

1.3.5 Das "TYPE"-Kommando

Mit dem "TYPE"-Kommando werden Inhalte von Dateien auf dem Sichtgerät ausgegeben.

Das Kommando hat die Form:

TYPE d:(dateibezeichnung)

Es bewirkt, daß der Inhalt der spezifizierten Datei auf das Sichtgerät ausgegeben wird.

Die Bezeichnung der Datei in dem "TYPE"-Kommando muß eindeutig gewählt werden, es dürfen also keine "?" und keine "*" verwendet werden.

Mit dem "TYPE"-Kommando kann erreicht werden, daß der Inhalt der Datei sowohl auf dem Drucker als auch auf dem Bildschirm erscheint ("Hardcopy"). Vor der Anwendung von "TYPE" muß man hierfür "ctl-P" eingeben (CONTROL- und P-Taste gleichzeitig). Mittels nochmaliger Eingabe von "CTL P" wird die Hardvopy-Funktion wieder ausgeschaltet.

Wenn die Datei mehr Zeilen hat als der Bildschirm zur Verfügung hat, so wird nur der letzte Teil der Datei auf dem Sichtgerät sichtbar. Um den durchlaufenden Dateiinhalt zu stoppen, betätigen Sie die Tasten "CONTROL" und "S" (gleichzeitig). Um weitere Zeilen der Datei auszugeben, muß "ctl-S" abermals eingegeben werden.

1.3.6 Das "SAVE"-Kommando

Das "SAVE"-Kommando kopiert den Inhalt einer Datei aus dem Programmspeicher (TPA) auf die aktivierte Disk. Die allgemeine Form des Befehls ist:

SAVE n <dateibezeichnung>

Dabei ist "dateibezeichnung" die Datei, unter welcher der Inhalt des Programmspeichers gesichert werden soll. "n" ist ein ganzzahliger Parameter; er gibt die Anzahl von 256-Byte-Segmenten ("Pages") der zu speichern den Datei an. Das "SAVE"-Kommando läßt sich speziell zur Speicherung eines Programms in Maschinensprache verwenden. Die Ausführung des Programms läßt sich dann durch einfaches Eingeben der Dateibezeichnung, unter der das Programm im Maschinencode gespeichert ist, realisieren. Der zeitraubende Vorgang des Übersetzens und Bindens kann auf diese Art und Weise eingespart werden. Im folgenden sind die hierfür

notwendigen Kommandos mit Erklärung dargestellt:
Erstellen Sie sich zunächst das Programm in einer Datei; für dieses Übungsbeispiel wird die Programmiersprache FORTRAN verwendet, die Programm-Datei wird TEST.FOR genannt. Der FORTRAN-Compiler F80 (d.h. die Dateien F80.COM, L80.COM) muß vorhanden sein.

Der Dateiname (hier: "TEST") ist frei wählbar, während der Dateityp für das FORTRAN-Programm mit "FOR" obligatorisch ist.

F80 = TEST/L	Übersetzen des Programms, unter Umständen werden Fehler in Ihrem Programm angezeigt, diese sollten Sie zunächst korrigieren und dann das Programm nochmals übersetzen. (Nähere Anweisungen hierzu in der Beschreibung Ihres Compilers).
L80	Aufruf des Linkage-Loaders, System meldet sich mit "***".
TEST	Eingabe des Dateinamens.
/N	Befehl im Linkage-Loader
/E	Befehl im Linkage-Loader. Es werden zwei Zeilen auf den Bildschirm ausgegeben: die zweite Zeile enthält mehrere Zahlen in eckigen Klammern. Die letzte dieser Zahlen gibt die Anzahl der 256-Byte-Segmente, also "n" an. Dieses "n" wird später im "SAVE"-Kommando verwendet. Das Betriebssystem meldet sich wieder mit dem aktivierten Laufwerk.
SAVE n TEST.COM	Das Programm wird unter der Dateibezeichnung "TEST.COM" gespeichert. Es kann jetzt jederzeit durch Eingabe von "TEST" gestartet werden.

Wird das "SAVE"-Kommando in der Form

SAVE n D: <dateibezeichnung>

verwendet, so wird die Datei "dateibezeichnung" auf der Disk in Laufwerk "d" gespeichert.

1.3.7 Das "USR"-Kommando

Auch mit CP/M2. als "Single-User-System" ist es möglich, in verschiedenen Benutzerbereichen zu arbeiten. Jeder Benutzer erhält eine Nummer im Bereich von 0 (voreingestellt) bis 15. Durch das Kommando

USER n

wird der gewünschte Benutzerbereich eingestellt. "n" ist dabei die Zahl von 0 bis 15. Im eingestellten Bereich ist ein Zugriff nur auf die Dateien dieses Bereichs möglich. Es kann also beispielsweise nach dem "DIR"-Kommando die Meldung

NO FILE

auf dem Sichtgerät erscheinen, obwohl die Disk keinesfalls leer sein muß. Die Dateien befinden sich dann im anderen Benutzerbereich.

1.4 TRANSIENTE KOMMANDOS

1.4.1 Allgemeines

Transiente Kommandos müssen vor der Ausführung von der Disk in den Hauptspeicher geladen werden. Deshalb ist es notwendig, daß die entsprechenden Programme auf der aktivierten Disk vorhanden sind. Ist dieses nicht der Fall, so meldet das Betriebssystem CP/M 2. "NO FILE".

Überprüfen Sie im Zweifelsfall, ob das benötigte Kommando auf der aktivierten Disk vorhanden ist. Verwenden Sie dazu das "DIR"-Kommando (s. auch Abschnitt 1.3.2).

CP/M 2. antwortet nach der Ausführung der Kommandos mit "A)" bzw. "B)" usw., also genau wie bei den Residenten Kommandos.

Die Befehle werden auch hier mit "RETURN" abgeschickt. Dieses ist zur Ausführung jedes Befehls notwendig und wird bei der Kommandobeschreibung deshalb nicht mehr explizit erwähnt.

Das Umschalten von einem Laufwerk auf das andere erfolgt ebenfalls durch die Eingabe der gewünschten Laufwerkbezeichnung ("A", "B" oder Andere), gefolgt von einem Doppelpunkt.

1.4.2 Das "PIP"-Kommando

Das "PIP"-Programm ist ein Übertragungsprogramm für Dateien. "PIP" steht für "Peripheral Interchange Program". Die Datei "PIP.COM" muß zur Ausführung des Kommandos auf der Disk vorhanden sein. In erster

Linie dient "PIP" dem Kopieren von Dateien. Darüber hinaus gibt es noch weitere Möglichkeiten, die ebenfalls in diesem Kapitel beschrieben werden. "PIP" kann grundsätzlich auf zwei Arten aktiviert werden:

- als ein Kommando, welches sich auf die in dem Kommando enthaltenen Dateien bezieht; die Form des Befehls lautet dann:

PIP "weitere Anweisungen"

- als ein Programmaufruf; die Form des Kommandos ist dann nur:

PIP

Das "PIP"-Programm meldet sich dann mit "*" und wartet auf weitere Anweisungen. Dieser Modus wird solange beibehalten, bis ein einfaches "RETURN" eingegeben wird. Danach meldet sich das Betriebssystem wieder mit "A)", "B)" oder anderen Laufwerken.

Zunächst soll hier das Kopieren von Dateien beschrieben werden.

Das Kommando

A)PIP d:=<dateibezeichnung>

bewirkt, daß die Datei "dateibezeichnung" der momentan aktivierten Disk auf die Disk "d" kopiert wird.

Voraussetzung für die erfolgreiche Ausführung des Kommandos ist, daß die Datei "<dateibezeichnung>" auf der aktivierten Disk vorhanden ist, ansonsten meldet CP/M 2.:

NO FILE:<dateibezeichnung>"

Für die Dateibezeichnung in diesem "PIP"-Kommando

besitzen alle Regeln des Abschnitts 1.1.3 Gültigkeit. So würde beispielsweise das Kommando

PIP B:=*.*

alle Dateien von der aktivierten Disk auf die Disk "B" kopieren. Die ursprünglichen Dateien auf der aktivierten Disk werden von diesem Kommando nicht beeinflusst, d.h. sie bleiben unverändert erhalten.

Es besteht, wie bereits oben bemerkt, die Möglichkeit, das PIP-Kommando als Programm aufzurufen, und im PIP-Programm zu arbeiten. Nach einfacher Eingabe von "PIP" meldet sich das Datei-Übertragungsprogramm mit "*" und ist nun für Kommandos bereit, bis die Eingabe eines "RETURN" erfolgt.

Um eine Datei zu kopieren, gibt man nun das Kommando in der Form

d:= <dateibezeichnung>

ein.

Um einer kopierten Datei eine neue Bezeichnung zu geben, kann man das PIP-Kommando in der Form

PIP d: <dateibezeichnung neu> = <dateibezeichnung alt>

verwenden. In dieser Form ist das Kommando anzuwenden, wenn das Laufwerk "d" aktiviert ist.

In jedem Fall besteht allerdings auch hier die Notwendigkeit, daß die Datei "<dateibezeichnung alt>" auf dem momentan aktivierten Laufwerk vorhanden ist (andernfalls stellt man der Dateibezeichnung den Buchstaben des betreffenden Laufwerks, gefolgt von einem Doppelpunkt voran; siehe auch Kapitel 1.1.3). Die Datei "<dateibezeichnung neu>" befindet sich nach dem

Kopiervorgang auf der Disk des Laufwerks "d", wobei "d" im Kommando zu spezifizieren ist.

Ist es beabsichtigt, ein PIP-Kommando während der Ausführung zu unterbrechen, so ist wahllos ein Zeichen von der Konsole an die Zentraleinheit zu schicken. Das PIP-Programm wird dann unterbrochen und auf dem Bildschirm erscheint die Meldung des Betriebssystems "ABORTET".

Eine weitere Möglichkeit des PIP-Kommandos, neben dem einfachen, oben beschriebenen Kopieren von Dateien besteht darin, Dateien in Bereiche zu kopieren. Unter "Bereichen" sind hier z.B. Drucker, Konsole oder andere an die Zentraleinheit angeschlossene Peripheriegeräte zu verstehen.

Die wichtigste und deshalb hier zunächst beschriebene Funktion soll das Kopieren von Dateien in den Drucker-Bereich, also das Ausdrucken von Dateien sein. Dieses Ziel ist ebenfalls durch das Betätigen der Tasten "CONTROL" und "P" (gleichzeitig!) zu erreichen: dieser Befehl bewirkt, daß alle auf den Bildschirm ausgegebenen Zeichen auch vom Drucker ausgegeben werden ("Hardcopy"). Verwendet man dann das im Kapitel 1.3.5 beschriebene "TYPE"-Kommando, so wird der Inhalt einer Datei nicht nur auf das Sichtgerät ausgegeben, sondern auch ausgedruckt. Diese Form der Druckausgabe hat jedoch den Fehler, daß die Kommandos ebenfalls gedruckt werden. Deshalb ist es die elegantere Lösung, das PIP-Kommando zu verwenden.

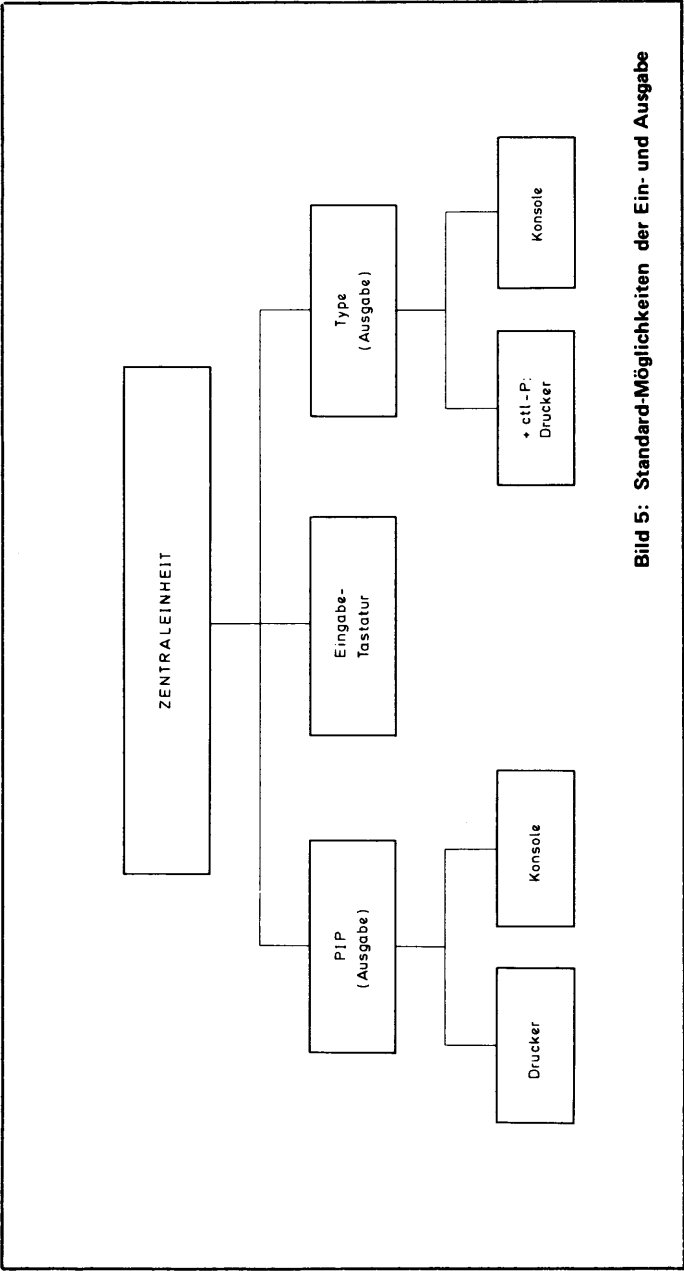


Bild 5: Standard-Möglichkeiten der Ein- und Ausgabe

Der logische Geräte­name für den Drucker ist "LST" (abgeleitet vom englischen Wort "list"). Um den Inhalt einer Datei auszudrucken, muß man das PIP-Kommando in der Form

PIP Ist:= <dateibezeichnung>

eingeben. Der Inhalt der Datei "<dateibezeichnung>" wird dann auf dem Drucker ausgegeben. Voraussetzung ist, daß ein Drucker vorhanden und eingeschaltet ist. Außerdem muß die Datei "<dateibezeichnung>" auf der aktivierten Disk vorhanden sein. Die Bezeichnung der Datei in diesem Befehl muß eindeutig sein, d.h. es dürfen keine "?" und "*" verwendet werden. Der Dateibezeichnung kann auch hier die Laufwerksbezeichnung vorangestellt werden, auf der sich die Datei befindet.

Die Ausgabe des Dateiinhalts auf das Sichtgerät kann sowohl mit dem "TYPE"-Kommando (beschrieben in Kapitel 1.3.5), als auch mit dem "PIP"-Kommando erfolgen. Der logische Geräte­name für die Konsole ist "CON". Sinngemäß zum Druckbefehl lautet also das Kommando für die Ausgabe auf die Konsole

PIP CON:=<dateibezeichnung>

Die Datei "<dateibezeichnung>" wird dann auf das Sichtgerät ausgegeben. Um den über den Bildschirm laufenden Text zu stoppen, sei auf die im Abschnitt 1.3.5 genannte Möglichkeit verwiesen. Die Dateibezeichnung muß auch hier wiederum eindeutig gewählt werden, d.h. es dürfen keine "?" und "*" verwendet werden.

Weitere logische Geräte­namen für das "PIP"-Kommando, die sinngemäß zu den beiden oben genannten Möglichkeiten verwendet werden, sind im Folgenden ge-

nannt. Voraussetzung ist, daß das jeweilige Gerät vorhanden ist:

- RDR Lochkarten- bzw. Lochstreifenleser
- PUN Lochkarten- bzw. Lochstreifenstanzer

Da diese Geräte heute wenig Verwendung finden, sei auf eine eingehendere Beschreibung hier verzichtet.

Anstelle der logischen Gerätenamen können im "PIP"-Kommando auch physische Gerätenamen verwendet werden. Diese sind im Folgenden aufgelistet:

- TTY: Beliebiges Eingabegerät, also Konsole, Kartenleser o.a.
- CRT: Konsole
- PTR: Lochkarten- oder Lochstreifenleser
- PTP: Lochkarten- oder Lochstreifenstanzer
- LPT: Zeilendrucker
- UC1: Konsole (vom Benutzer definiert)
- UR1: Lochkarten- bzw. Lochstreifenleser (vom Benutzer definiert)
- UR2: Weiterer Lochkarten- bzw. Lochstreifenleser (vom Benutzer definiert)
- UP1: Lochkarten- bzw. Lochstreifenstanzer (vom Benutzer definiert)
- UP2: Weiterer Lochkarten- bzw. Lochstreifenstanzer (vom Benutzer definiert)
- UL1: Drucker (vom Benutzer definiert)

Wird der logische Gerätenamen "PRN" für das "PIP"-Kommando verwendet, so wird wie bei der Verwendung von "LST" der Drucker angesprochen. Der Unterschied besteht darin, daß eine Zeilennummerierung vor-

genommen wird. Es erscheint also im Druckbild vor jeder ausgegebenen Zeile eine fortlaufende Nummer. Außerdem wird nach 60 Zeilen ein Seitenvorschub bewirkt und der Inhalt der Datei um acht Spalten nach rechts eingerückt.

Eine weitere Möglichkeit für die Anwendung des Datei-Übertragungsprogramms ist die Verkettung von Dateien. Die einfachste Form für die Eingabe des Kommandos ist:

PIP <datbezneu> = <datbezalt 1>, <datbezalt2>

Die Dateien "<datbezalt1>" und "<datbezalt2>" werden verkettet und sind ansprechbar unter der Dateibezeichnung "<datbezneu>". Der Inhalt der Datei "<datbezalt1>" steht dabei in "<datbezneu>" vor dem Inhalt der Datei "<datbezalt2>". Auf die beiden Dateien "<datbezalt1>" und "<datbezalt2>" kann weiterhin zugegriffen werden. Die Anzahl der verketteten Dateien ist nur durch die Kapazität der Disk begrenzt. Es ist auch möglich, das Kommando in der Form

PIP <datbez1> = <datbez1>, <datbez2>

zu verwenden. Es bewirkt dann, daß der Inhalt von "<datbez2>" an "<datbez1>" gefügt wird. Nach der Ausführung existieren dann weiterhin "<datbez2>" und die Datei "<datbez1>" in modifizierter Form.

Auch bei der Verwendung des "PIP"-Kommandos zum Auflisten von Dateien auf dem Drucker oder dem Sichtgerät ist die Verkettung möglich. Das Kommando

PIP LST:=<datbez1>, <datbez2>, <datbez3>

bewirkt, daß die drei in dem Kommando spezifizierten Dateien nacheinander in der im Kommando genannten Reihenfolge ausgedruckt werden. Sinngemäß gilt das Gleiche für die Ausgabe auf das Sichtgerät.

Das "PIP"-Kommando läßt sich durch eine Reihe von Parametern erweitern. Diese Parameter werden nun dargestellt und erläutert. Sie werden in eckige Klammern gesetzt und dem gesamten Kommando angefügt:

B	Die von "PIP" behandelten Daten werden zunächst in einen Speicher übernommen bis dieser voll ist. Die Kapazität dieses Speichers ist hardware-abhängig. Dieser Zusatz hat Bedeutung bei kontinuierlicher Dateneingabe (z.B. vom Lochstreifenleser).
Dn	Die Zeichen, welche über die Spalte n hinausreichen, werden während des Kopiervorgangs ignoriert.
E	Alle vom Kopiervorgang berührten Zeichen werden auf der Konsole ausgegeben.
F	Während des Kopiervorgangs werden alle Seitenvorschubzeichen ("form feed") gelöscht.
H	Parameter für die Übertragung von hexadezimalen Dateien; alle behandelten Zeichen werden geprüft, ob sie den korrekten Hexadezimal-Code enthalten. Eine hexadezimale Datei vom Typ ".HEX" wird vorausgesetzt.
I	Für Hexadezimal-Dateien: ' :00'-Sätze werden während der Übertragung ignoriert. Der "H"-Parameter wird automatisch gesetzt.
L	Alle Großbuchstaben werden in Kleinbuchstaben umgesetzt.
N	Jede Zeile der übertragenen Datei wird mit einer Zeilennummer versehen. Führende Nullen der Numerierung werden unterdrückt (soll dieses vermieden werden, verwende man den Parameter "N2").
O	Bei der Verkettung von Dateien wird das physische Ende der Datei ignoriert. Dieses ist bei Dateien von Bedeutung, die nicht im ASCII-Code verschlüsselt sind.
Pn	Bei der Übertragung wird in jeder "n-ten" Zeile ein Steuerzeichen für den Seitenvorschub gesetzt. Wird n fortgelassen oder setzt man n gleich 1, so wird n gleich 60 gesetzt. In Kombination mit dem Parameter F kann so eine Datei zum Drucken auf beliebige Papierlängen ausgerichtet werden.

- Q** Dieser Parameter erlaubt das teilweise Kopieren einer Datei. Der Zusatz wird in der Form
- `[Q"string" ^Z]`
- an das "PIP"-Kommando angefügt. Das Zeichen " " in einem Kommando bedeutet das Betätigen der "CONTROL"-Taste bei gleichzeitiger Eingabe des dem " " folgenden Zeichens (hier "Z"). Der Zusatz bewirkt, daß die Datei nur bis zum ersten Auftreten der Zeichenkette "string" kopiert wird. Die Zeichenkette "string" muß dabei in der ursprünglichen Datei vorhanden sein.
- S** Dieser Parameter wird in der Form
- `[S"string" ^Z]`
- benutzt. Er bewirkt, analog zum "Q"-Parameter, daß die Datei beginnend beim ersten Auftreten der Zeichenkette übertragen wird. Weitere Bemerkungen zum teilweisen Kopieren von Dateien sind im Anschluß an diese Parameter-Auflistung gemacht.
- U** Alle Kleinbuchstaben werden während der Übertragung in Großbuchstaben umgesetzt.
- V** Die neue Kopie der Datei wird mit der alten verglichen und auf Fehler überprüft. Dieser Parameter sollte bei allen wichtigen Dateiübertragungen verwendet werden, um sicherzustellen, daß unter Umständen auftretende Übertragungsfehler festgestellt werden (z.B. beim Kopieren des Betriebssystems).

Diese Parameter können auch in einem Kommando kombiniert verwendet werden. Angenommen es existiert eine Datei "TEXT", so bewirkt das Kommando

`PIP LST:=TEXT[ENV]`

das der Inhalt von "TEXT" auf den Drucker mit Zeilennumerierung ausgegeben wird. Außerdem erscheinen die an den Drucker übermittelten Zeichen auch auf dem Sichtgerät. Würde bei der Übertragung ein Fehler auftreten, so würde das Betriebssystem eine Meldung ausgeben.

Wie bereits oben erwähnt, können mit den Parametern "Q" und "S" Dateien teilweise übertragen werden. Mit dem Parameter "Q" kann bis an eine bestimmte Stelle

der Datei übertragen werden, während mit dem Parameter "S" von einer bestimmten Stelle an übertragen werden kann. Die Anwendung von "Q" und "S" soll nun an einigen Beispielen erläutert werden.

Angenommen es existiert die Datei "TEXT" mit folgendem Dateiinhalt:

"CP/M ist ein Betriebssystem für Mikrocomputer. Es unterstützt eine Hardware, die auf dem Mikroprozessor Z-80 bzw. Z-80A von ZILOG sowie dem 8080 von INTEL als Zentraleinheit basiert."

Es soll nun nur der erste Satz und der Zusatz "Es unterstützt eine Hardware" der Datei auf dem Drucker ausgegeben werden. Folgende Kommandos erfüllen diese Aufgabe:

A>PIP

*LST:=TEXT[QHardware ^Z]

(Die "Control"- und "Z"-Tasten müssen gleichzeitig betätigt werden). Es ist von Bedeutung, "PIP" als Programmaufruf auszuführen (s. auch Seite 21). Nur in diesem Falle werden die Zeichen der zu suchenden Zeichenkette (in diesem Fall "Hardware") nicht in große Buchstaben übersetzt.

Soll nun der Teil nach dem Wort "Hardware" übertragen werden, so müssen folgende Kommandos verwendet werden, um diese Aufgabe zu erfüllen:

A>PIP

*LST:=TEXT[SHardware ^Z]

Sinngemäß gilt hier das gleiche wie für den Parameter "Q".

Durch Kombination der Parameter "Q" und "S" können auch Teile der Datei übertragen werden, die nicht

am Anfang oder am Ende der Datei stehen. Folgendes Beispiel mag dies verdeutlichen:

A>PIP

*LST:=TEXT[S8080CTL—ZQINTEL ^Z]

In diesem Fall werden nur die Zeichen "8080 von INTEL" übertragen.

1.4.3 Das "STAT"-Kommando

"STAT" ist das Informations-Programm über Dateien und Disks. Außerdem zeigt es die Zuordnung von logischen zu physischen Bereichen (siehe auch Abschnitt 1.4.2) an und ändert diese unter Umständen.

Wird das Kommando in der Form

d)STAT

einggegeben, so wird der noch zur Verfügung stehende Platz auf der Disk "d" in der Form

d:R/W, SPACE:nnnK

angezeigt. "nnn" ist dabei eine ganze Zahl und gibt den ungenutzten Speicherplatz auf der Disk in K-Bytes an. Das momentan nicht aktivierte Laufwerk ist ebenfalls mit dem "STAT"-Befehl anzusprechen. Der Befehl

d)STAT B:

bewirkt, daß der ungenutzte Speicherplatz der Disk "B" in der Form

BYTES REMAINING ON B: nnnK

angezeigt wird. "nnn" gibt wiederum den Speicherplatz in K-Bytes an.

Eine weitere Möglichkeit der Anwendung des "STAT"-Kommandos ist die Information über Dateien. Die Schreibweise von "STAT" ist dann:

STAT <dateibezeichnung>

Für "<dateibezeichnung>" gelten alle in Abschnitt 1.1.3 genannten Regeln (also auch das Voranstellen der Laufwerksbezeichnung, z.B. B:<dateibezeichnung>). Das Betriebssystem CP/M 2. meldet dann nach erfolgreicher

Suche der Datei(en):

RECS BYTS EXT Acc

Unter dieser Überschrift werden alle zu “<dateibezeichnung>” passenden Dateien aufgelistet. Die einzelnen Spalten bedeuten dabei folgendes:

- RECS: Anzahl der 128-Byte-Sätze der betreffenden Datei.
- BYTS: Von der Datei belegter Speicherplatz in K-Byte (1 K-Byte = 1024 Byte)
- EXT: Anzahl von 16 K-Byte-Bereichen der Datei.
- Acc: Gibt den Status der Datei an. “R/W” bedeutet, daß die betreffende Datei zum Lesen und Schreiben bereit ist (REAR/ WRITE). “R/O” (READ/ONLY) bedeutet dagegen, daß die Datei nur gelesen werden kann. Schreiben oder Löschen ist nicht möglich. Wie von einem Status in den anderen umgeschaltet werden kann, wird in diesem Abschnitt weiter unten erläutert.

Neben diesen Hinweisen steht die Bezeichnung der Datei, für die diese Angaben gültig sind. Ein Beispiel soll die Information über Dateien näher erläutern:

Recs	Bytes	Ext	Acc
25	4k	1	R/W A:TEXT

Bytes Remaining on A: 43k

Diese Meldung von CP/M 2. wird beispielsweise nach dem Kommando

A)STAT TEXT

ausgegeben. Sie enthält folgende Informationen:

Die Datei "TEXT" enthält eine Anzahl von 25 Sätzen à 128 Bytes. Dies entspricht aufgerundet 4 kBytes. Es wird ein Bereich (Ext) belegt. Die Datei ist auf Lesen/Schreiben (R/W) geschaltet und befindet sich auf der Disk "A". Auf dieser verbleibt ein ungenutzter Speicherplatz von 43 K-Bytes.

Eine weitere Möglichkeit der Anwendung des "STAT"-Kommandos besteht darin, den Status einer Datei festzulegen. Unter dem "Status" einer Datei wird hier verstanden, ob die Datei nur zum Lesen freigegeben ist oder ob ebenfalls die Möglichkeit des Schreibens besteht. Das Kommando

STAT <dateibezeichnung> \$R/O

bewirkt, daß die Datei "<dateibezeichnung>" nur noch gelesen werden kann. Es ist dann nicht mehr möglich in sie zu schreiben oder sie mit dem "ERA"-Kommando zu löschen. Um auch schreiben und löschen zu können, wird das Kommando in der Schreibweise

STAT <dateibezeichnung> \$R/W

einggegeben. Für die Bezeichnung der Dateien in diesem Kommando gelten alle in Abschnitt 1.1.3 genannten Möglichkeiten.

Das Kommando

STAT DSK:

bewirkt eine Information über die Formatierung der Disks. Es ist möglich, mit diesem Kommando auch das momentan nicht aktivierte Disketten-Laufwerk anzusprechen, indem man beispielsweise

A)STAT B:DSK:

eingibt. In diesem Fall wird vom Betriebssystem CP/M 2. eine Information über die Disk in Laufwerk "B" erteilt.

Im Betriebssystem CP/M 2. muß unterschieden werden zwischen Dateien, welche durch das "DIR"-Kommando (s. auch Abschnitt 1.3.2) aufgelistet werden und Dateien, die zwar auf der Disk vorhanden sind, aber nicht aufgelistet werden (sogenannte System-Dateien). Mit dem "STAT"-Befehl ist es möglich, einer Datei den Status "System-Datei" zu- bzw. abzuerkennen. Der Befehl

STAT <dateibezeichnung> \$SYS

bewirkt, daß die Datei "dateibezeichnung" nicht mehr durch das "DIR"-Kommando aufzulisten ist. Sie bleibt aber erhalten und es ist weiterer Zugriff auf sie möglich. Den Zugriff auf eine System-Datei durch das "DIR"-Kommando erreicht man, indem man folgendes Kommando eingibt:

STAT <dateibezeichnung> \$DIR

Die Datei "<dateibezeichnung>" wird bei Anwendung des "DIR"-Kommandos aufgelistet. Für beide Kommandos "DIR" und "SYS" dürfen ebenfalls die Regeln aus Abschnitt 1.1.3 angewendet werden. Wie bereits bemerkt, ist es möglich, die Zuordnung von logischen zu physischen Gerätenamen mit dem "STAT"-Kommando zu beeinflussen. Bei Eingabe des Kommandos

STAT VAL:

listet das Betriebssystem CP/M 2. unter der Überschrift "Iobyte Assign:" die möglichen Zuordnungen logischer und physischer Geräte auf. Diese Liste hat die folgende Form:

```

CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PIP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:

```

Die linke Spalte zeigt die logischen Gerätenamen an (abgekürzt, wie schon in Abschnitt 1.4.2 erläutert). Auf der rechten Seite des "="-Zeichens stehen die physischen Gerätenamen, die den logischen zuzuordnen sind (auch diese wurden bereits in Abschnitt 1.4.2 beschrieben). Außerdem werden in der vom Betriebssystem ausgegebenen Liste weitere Informationen über Möglichkeiten des "STAT"-Befehls gegeben.

Mit dem CP/M 2.-Kommando

```
STAT DEV:
```

wird die aktuelle Zuordnung von logischen und physischen Geräten angezeigt. Dieses geschieht in der Form

```

CON: is TTY:
RDR: is TTY:
PUN: is TTY:
LST: is TTY:

```

Die linke Spalte zeigt auch hier die logischen Gerätenamen an, während rechts von "is" die physischen Gerätenamen angezeigt werden.

Diese aktuelle Zuordnung ist durch den Anwender zu beeinflussen. Dieses geschieht mittels des Befehls

```

STAT <logname1>:=<physname1>;
    <logname2>:=<physname2>; . . .

```

"logname1" und "logname2" sind dabei die logischen Gerätebezeichnungen (CON, RDR, PUN, LST) und "physname1", sowie "physname2" die im CP/M 2.

möglichen physischen Gerätebezeichnungen (durch das Kommando "STAT VAL:" abzufragen). Die Zuordnungen können in diesem Kommando beliebig aneinander gereiht werden. Wird ein unbekannter Gerätenamen verwendet, so meldet CP/M 2.:

Invalid Assignment

Voraussetzung für die Anwendung der neuen Zuordnung ist selbstverständlich, daß ein der physischen Bezeichnung entsprechendes Gerät vorhanden und funktionsfähig ist.

Eine weitere Möglichkeit des "STAT"-Kommandos besteht in der Information über den Benutzerbereich (Abschnitt 1.3.7). Das Kommando

STATUSR:

bewirkt die Ausgabe folgender beispielhafter Meldung:

Active User: 0

Active Files: 0 1

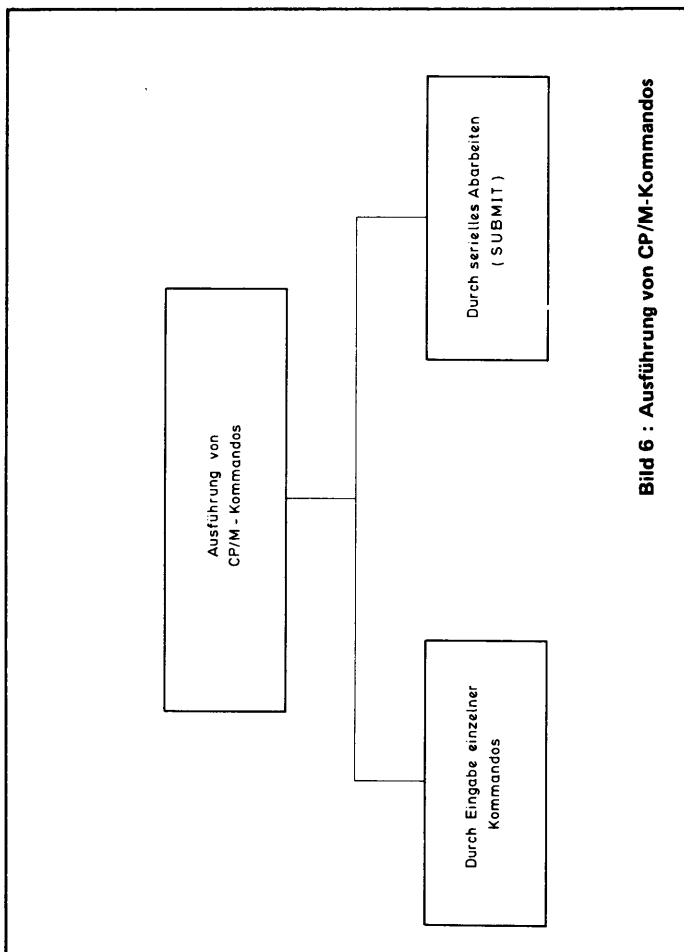
Diese bedeutet, daß momentan der Benutzerbereich 0 aktiviert ist und Dateien unter den Bereichen 0 und 1 existieren.

1.4.4 Das "SUBMIT"-Kommando

Das "SUBMIT"-Kommando dient zur seriellen Ausführung von CP/M 2.-Kommandos. Für die Ausführung dieses Befehls muß eine Datei vom Typ ".SUB" existieren. Diese Datei enthält die in einer Reihenfolge auszuführenden Kommandos. Sie stehen zeilenweise in der Anordnung in dieser Datei, in der sie abzuarbeiten sind.

Ein Kommando besteht in der Regel aus der Anweisung (wie STAT, TYPE, PIP) und der Dateibezeichnung,

auf die sich diese Anweisung bezieht. Die Ausnahmen wurden in den Kommandobeschreibungen bisher erwähnt. In der ".SUB"-Datei, welche die auszuführenden Kommandos enthält, werden diese Dateibezeichnungen durch Parameter ersetzt. Diese Parameter haben eine ähnliche Wirkung wie Variablen in einem Programm; sie werden bei der Ausführung der Kommandos durch Dateibezeichnungen ersetzt.



Die Anwendung des "SUBMIT"-Kommandos soll an einem Beispiel verdeutlicht werden:

Zunächst wird mit dem Text-Editor die Datei der auszuführenden Befehle vom Typ ".SUB" erstellt, sie darf keine Leerzeilen enthalten. Die Datei heißt "BEFEHL.SUB". Sie hat folgenden Inhalt:

```
TYPE $1
PIP LST:=$1
PIP CON:=$2
```

"\$1" und "\$2" sind die Parameter, welche die Dateibezeichnungen in den Kommandos ersetzen. Im "SUBMIT"-Kommando werden die anzusprechenden Dateien aufgeführt. Vorausgesetzt, es existieren die Dateien "TEST" und "BEISPIEL", so kann "SUBMIT" wie folgt verwendet werden:

SUBMIT BEFEHL TEST BEISPIEL

Die Kommando-Datei BEFEHL.SUB
wird als erstes aufgeführt.
(nur "BEFEHL" angeben)

Die nächste Datei bezieht sich auf den Parameter "\$1", d.h. bei Ausführung der Befehle wird dort "TEST" eingesetzt, wo "\$1" steht.

Die Datei "BEISPIEL" wird überall an Stelle des Parameters "\$2" eingesetzt.

Die Verwendung weiterer Parameter und damit das Ansprechen weiterer Dateien ist möglich.

Sollen sich die im "SUBMIT"-Kommando angegebenen Dateien auf die momentan nicht aktivierte Disk beziehen, so ist der jeweiligen Dateibezeichnung die Lauf-

werksbezeichnung und der Doppelpunkt voranzustellen
Das Kommando hätte dann beispielsweise folgende Form:

SUBMIT B:BEFEHL TEST B:BEISPIEL

Die Dateien "BEFEHL" und "BEISPIEL" müssen sich dann auf Disk "B" befinden. Es ist ebenfalls möglich, die "\$"-Parameter in der Befehlsdatei (vom Typ "SUB") mit den Dateibezeichnungen zu kombinieren. Folgendes Beispiel mag dieses verdeutlichen:

Die Datei "BEFEHL.SUB" hat jetzt folgenden Inhalt:

**TYPE \$1ST
PIP LST:=\$1ST
PIP CON:=BEI\$2**

Unter der Voraussetzung, daß die Dateien "TEST" und "BEISPIEL" weiterhin verfügbar sind, wird durch das folgende "SUBMIT"-Kommando das gleiche wie beim Beispiel auf Seite 35 bewirkt:

SUBMIT BEFEHL TEST BEISPIEL

Das "SUBMIT"-Kommando bezieht sich wiederum auf die Datei "BEFEHL.SUB".

Die Angabe "TE" wird mit den Zeichen "ST" nach dem Parameter "\$1" in den Befehlen "TYPE \$1ST" und "PIP LST:=\$1ST" kombiniert.

Die Angabe "SPIEL" wird mit den Zeichen "BEI" vor dem Parameter "\$2" im Befehl "PIP CON:=BEI\$2" kombiniert.

Soll die Abarbeitung der einzelnen Kommandos der

„SUB“-Datei abgebrochen werden, so ist die Taste „DELETE“ zu betätigen. CP/M 2. meldet sich dann wieder für die Eingabe von Kommandos bereit (durch „A“) „B“ o.a.).

Eine weitere Möglichkeit des „SUBMIT“-Kommandos besteht darin, in der Datei vom Typ „SUB“ das „SUBMIT“-Kommando selbst als Befehl zu verwenden.

1.4.5 Das „ASM“-Kommando

Das „ASM“-Kommando dient zum Laden und Ausführen des CP/M 8080 Übersetzer-Programms. Es übersetzt Kommandos der Assemblersprache in die Maschinensprache. An dieser Stelle sei auf den Assembler-Teil des Hersteller-Handbuches verwiesen, der zum Verständnis der maschinenorientierten Sprache ASSEMBLER notwendig ist.

Voraussetzung für die Anwendung des „ASM“-Kommandos ist, daß eine Datei vom Typ „ASM“ mit Assembler-Befehlen existiert.

Existiert beispielsweise die Datei „DAT.ASM“, so bewirkt das Kommando

ASM DAT

daß der Inhalt von „DAT“ assembliert wird. Treten während der Übersetzung Fehler auf, so werden diese auf dem Sichtgerät gemeldet. Es wird automatisch die Datei „DAT.PRN“ erzeugt, welche das Quellprogramm und die entsprechenden Befehle in der Maschinensprache enthält. Außerdem werden hier auch eventuelle Fehlermeldungen aufgelistet. Darüberhinaus wird die Datei „DAT.HEX“ erzeugt. Sie enthält das Programm in der Maschinensprache des Prozessors.

Das "ASM"-Kommando kann auch Dateien ansprechen, die nicht auf dem aktuell aktivierten Laufwerk vorhanden sind. Dieses geschieht z.B. in der Form

ASM B:DAT

Dieses Kommando spricht die Datei "DAT.ASM" an, auch wenn diese sich auf Laufwerk "B" befindet und "A" aktiviert ist. Die Dateien "DAT.PRN" und "DAT.HEX" befinden sich dann ebenfalls auf Disk "B".

1.4.6 Das "LOAD"-Kommando

Die durch das "ASM"-Kommando erzeugte Datei vom Typ "HEX" (siehe auch Abschnitt 1.4.5) wird durch das "LOAD"-Kommando in eine "COM"-Datei umgesetzt. Die Datei (sie bestand ursprünglich als Datei vom Typ "ASM" aus Assembler-Befehlen) wird dadurch zu einem ausführbaren Programm. Sie hat damit den gleichen Typ ("COM"), wie die Transienten-Kommandos. Das bedeutet, daß die Befehle (und damit der Dateiinhalt) durch Aufrufen des Dateinamens auszuführen sind.

Existiert beispielsweise die Datei "DAT.HEX" (siehe auch Abschnitt 1.4.5), so bewirkt die Ausführung des Kommandos

LOAD DAT

daß eine Datei "DAT.COM" kreiert wird. Diese Datei enthält den ursprünglichen Inhalt der Datei "DAT.ASM" (in Assembler-Sprache) im Maschinen-Code. Die Befehle können nun durch einfaches Aufrufen des Dateinamens

DAT

ausgeführt werden. Es ist dann kein Übersetzungslauf mehr notwendig.

1.5 ED: DER TEXTEDITOR DES CP/M-BETRIEBS-SYSTEMS

1.5.1 Einführung in den Editor, Aufbau des Editors

“ED” ist der Texteditor des CP/M-Betriebs-Systems. Er wird benutzt, um Dateien (FILES) zu erstellen oder zu ändern, die Daten, Programme oder beliebigen Text aus Großbuchstaben enthalten. Er enthält außerdem die Möglichkeit:

- Bestimmte Zeichenketten (STRINGS) zu suchen und automatisch durch andere zu ersetzen
- “Block-move” (Textblöcke kopieren)
- Beliebige Dateien in den bearbeiteten Text einzufügen
- Kommandos automatisch mehrfach ausführen zu lassen.

Die Datei, in der sich ED befindet, und die vorhanden sein muß, heißt “ED.COM”. Man initialisiert den Editor, indem man

ED <dateiname>

oder

ED <dateiname>, <dateityp>

oder

ED D:<dateiname>.<dateityp>

eingibt.

Der Inhalt der Datei kann dann mit entsprechenden Kommandos (siehe 1.5.2 Text-Übertragung) in den Arbeitsspeicher geladen, angezeigt und verändert werden. Wenn die angegebene Datei bisher nicht existierte, wird

sie vom Editor automatisch als leere Datei kreiert und kann beschrieben werden. Es können maximal 6000 Zeichen einer Datei gleichzeitig zur Änderung in den Arbeitsspeicher geladen werden.

Bei der Arbeit mit dem Editor treten folgende Versionen einer Datei auf:

Form:	Beispiel:	Bedeutung:
<dateiname>.<dateityp>	TEST.FOR	"Haupt-Datei"
<dateiname>.BAK	TEST.BAK	"Hintergrund-Datei"
<dateiname>.\$\$\$	TEST.\$\$\$	"Kurzzeitige Datei"

Der Editor kann nur Text bearbeiten, der im Arbeitsspeicher steht. Dazu wird zunächst die zu bearbeitende Haupt-Datei von der Disk in den Arbeitsspeicher geladen und kann dort verändert werden. Der geänderte Text wird in der Kurzzeitigen Datei <dateiname>.\$\$\$ abgelegt (siehe Bild 7).

Nach Verlassen des Editors wird automatisch der Inhalt des Arbeitsspeichers in die Kurzzeitige Datei geladen, gefolgt von restlichem ungelesenen Text, der noch in der Haupt-Datei steht. Die ursprüngliche Haupt-Datei wird zur Hintergrund-Datei und umbenannt in <dateiname>.BAK und als solche auf der Disk abgespeichert, so daß sie noch jederzeit zur Verfügung steht.

Neue Hauptdatei wird die Kurzzeitige Datei <dateiname>.\$\$\$, die jetzt den Namen der ursprünglichen Haupt-Datei <dateiname>.<dateityp> erhält. Sie ist das eigentliche Ergebnis des Editier-Vorganges.

Auf der Disk erscheinen jetzt also:

<dateiname>.BAK	Ursprüngliche Haupt-Datei, die zu bearbeiten war, noch verfügbar.
-----------------	---

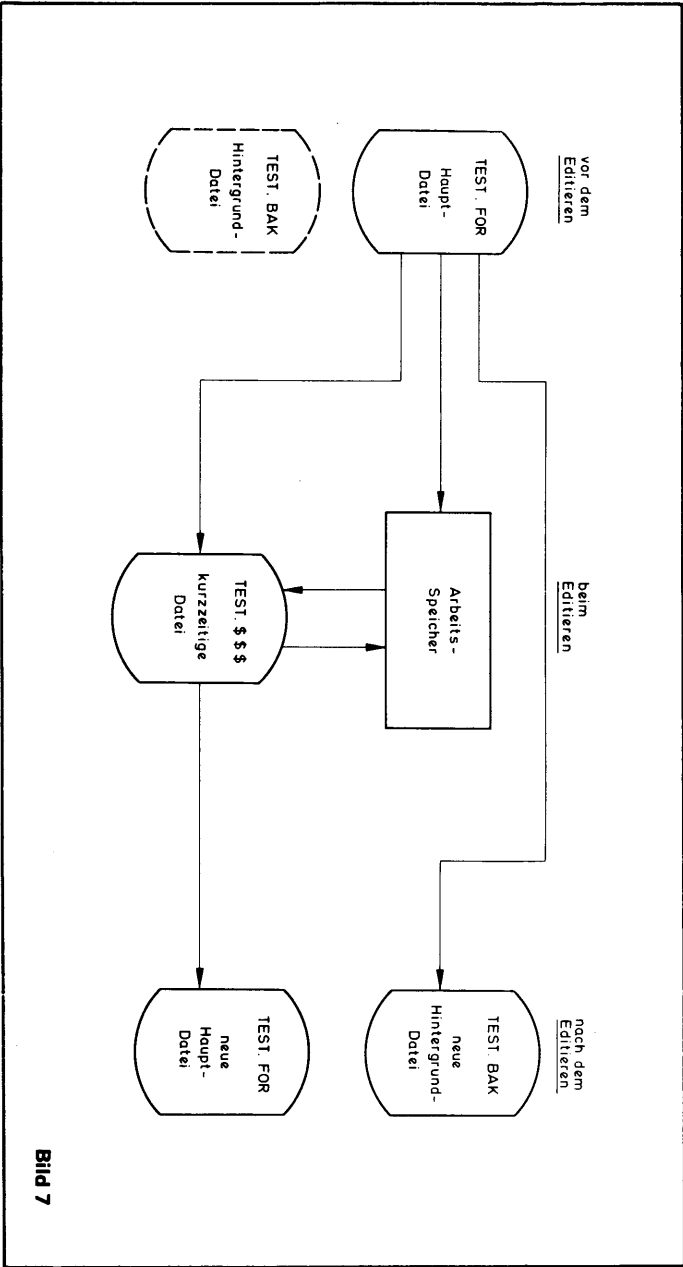


Bild 7

⟨dateiname⟩.⟨dateityp⟩ Neue bearbeitete Datei. Haupt-
Datei für weitere Verarbeitung.

Wurde eine Datei erst durch das ED-Kommando kreiert,
so enthalten die beiden nach Abschluß der Bearbeitung
entstandenen Dateien den gleichen Inhalt.

Die Kurzzeitige Datei ⟨dateiname⟩.\$\$\$ tritt normaler-
weise nicht in Erscheinung, es sei denn, der Editier-
Vorgang wurde anormal unterbrochen, z.B. durch Ent-
nehmen der Disk, auf der gerade die Bearbeitung statt-
findet.

1.5.2 Text-Übertragung

Folgende Kommandos stehen im Editor zur Verfügung,
um Zeichen von der Haupt-Datei in den Arbeitsspei-
cher und zur Kurzzeitigen (und im Abschluß endgül-
tigen) Datei zu übertragen.

Dabei ist n eine Zahl von 1 bis 65535.

Wird für n “#” eingesetzt, so nimmt der Editor die
höchstmögliche Zahl (65535) an, das Kommando gilt
dann also für alle Zeilen.

Wird n weggelassen, so setzt der Editor n=1.

(Jedes Kommando muß mit der “RETURN”-Taste ab-
geschlossen werden.):

Kommando	Bedeutung
nA	Die nächsten (oder ersten) n unbearbeiteten Zeilen werden von der Haupt-Datei in den Arbeitsspeicher hinter dort eventuell schon vorhandene Zeilen gela- den. I.A. beginnt man die Arbeit im Editor, nach dessen Initialisierung und Meldung mit “*”, indem man “#A” eingibt. Damit werden alle Zeilen der zu edi- tierenden Datei in den Arbeitsspeicher geladen und können verändert werden.

Kommando	Bedeutung
	"OA" füllt den Arbeitsspeicher bis zur Hälfte aus der Haupt-Datei.
nW	Die ersten n Zeilen des Arbeitsspeichers werden in die Kurzzeitige Datei hinter dort eventuell schon vorhandene Zeilen geladen. "#W" lädt den gesamten Arbeitsspeicher-Inhalt in die Kurzzeitige Datei. "OW" entleert den Arbeitsspeicher zur Hälfte in die Kurzzeitige Datei.
E	Der Editier-Vorgang wird abgeschlossen. Der Inhalt des Arbeitsspeichers und restliche nicht geänderte Zeilen der Haupt-Datei werden in die Kurzzeitige Datei geladen. Die Dateien werden wie oben beschrieben umbenannt. Es erfolgt ein Rücksprung ins Betriebssystem.
H	Der Editier-Vorgang wird automatisch abgeschlossen und mit der Kurzzeitigen Datei als neue Haupt-Datei gestartet. Der Arbeitsspeicher wird gelöscht und eine neue Kurzzeitige Datei wird kreiert. (Entspricht dem E-Kommando, gefolgt von einer Neuinitialisierung des Editors mit der editierten Datei als neue Haupt-Datei.)
O	Zurück zur Original-Datei. Der Arbeitsspeicher und die Kurzzeitige Datei werden gelöscht. Alle vorherigen Editor-Kommandos werden eliminiert.
Q	Verlassen des Editors ohne Veränderung einer Datei. Rücksprung ins Betriebssystem.

Wird das Fassungsvermögen des Arbeitsspeichers überschritten, so tritt eine Fehlermeldung auf. Dann muß ein Kommando eingegeben werden, das den Inhalt des Arbeitsspeichers verringert (z.B.: "OW").

1.5.3 Text-Manipulation, Speichern und Ändern

Speicher- und Änderungsvorgänge können nur an Text vorgenommen werden, der sich im Arbeitsspeicher be-

findet. Dazu wird eine bestimmte Anzahl von Zeilen der zu editierenden Datei mit dem "A"-Kommando, wie beschrieben, in den Arbeitsspeicher geladen.

"ED" arbeitet cursor-orientiert, d.h. Änderungen und Einfügungen sind nur dort möglich, wo sich ein "imaginärer Cursor" (CP, character pointer) befindet, der mit verschiedenen Kommandos an die gewünschte Stelle geschoben werden kann. Er befindet sich immer vor dem ersten oder hinter dem letzten Zeichen einer Zeile oder zwischen zwei Zeichen. Nach dem Initialisieren des Editors steht CP vor dem ersten Zeichen der ersten Zeile.

Die aktuelle Zeile (CL, current line) ist die Zeile, in der sich CP befindet.

CP wird, falls er sich im Bildfeld befindet, durch "*" dargestellt.

Der Editor gibt für jede Zeile eine maximal fünfstellige Zeilennummer an, die nicht mit abgespeichert wird, sondern nur als Orientierungshilfe dient.

Folgende Erklärungen sind für das Verständnis der unten aufgeführten Editor-Kommandos notwendig:

n ist eine Zahl von 1 bis 65535. Läßt man n weg, so wird automatisch n=1 eingesetzt. Setzt man für n "*" ein, so entspricht das der höchstmöglichen Zahl. Das Kommando gilt dann für alle Zeilen bzw. Zeichen.

+/- gibt die Richtung innerhalb des Arbeitsspeichers bzw. Textes an. + ist die Richtung zum Ende des Textes hin, also bei Zeilen weiter nach unten, innerhalb einer Zeile weiter nach rechts. - ist die entgegengesetzte Richtung, also zurück in Richtung des Textan-

fangs. (Ausnahme: siehe B-Kommando). Das Plus-Zeichen kann auch weggelassen werden. Mit folgenden Kommandos kann nur CP bewegt werden:
(Jedes Kommando mit der "RETURN"-Taste abschließen!)

Kommando	Bedeutung
+/-B	Setzen von CP an den Anfang des Textes mit "+B" bzw. "B" oder Setzen von CP an das Ende mit "-B".
+/-nC	Bewegen von CP um n Zeichen. Richtung wie oben erklärt. ("carriage return" und "line feed" werden als zwei einzelne Zeichen gewertet.)
+/-nL	Wenn n=0, setzen von CP an den Anfang der CL. Sonst setzen von CP an den Anfang der CL und bewegen von CP um n Zeilen nach oben oder unten (Vorzeichen wie oben erklärt). "L" kann weggelassen werden "+/-n". Es genügt also Eingabe einer Zahl n, um CP um n Zeilen weiter vor zu bewegen. Die n-te Zeile wird dann ausgegeben.
n:	Setzen von CP an den Anfang der Zeile n.
:n	Setzen von CP an den Anfang der Zeile n und Ausgeben dieser Zeile.
RETURN	(Drücken der Taste "RETURN".) Setzen von CP um eine Zeile weiter vor und Ausgeben der Zeile.

Mit folgenden Kommandos können nun an der Stelle, an der sich CP befindet, Veränderungen vorgenommen werden:

Kommando	Bedeutung
+/-nD	Löschen von n Zeichen vor oder hinter CP. (Vorzeichen wie oben erklärt.)
+/-nK	Löschen von n Zeilen vor oder hinter CP, incl. der Zeichen vor oder hinter CP in der CL (aktuelle Zeile).
+/-nT	Wenn n=0: Ausgeben des Inhaltes der CL vor CP.

K(string) ^Z

Wenn n=1: Ausgeben des Inhaltes der CL hinter CP.
Wenn n>1: Ausgeben von CL und n-1 davor oder dahinter stehenden Zeilen. (Vorzeichen wie oben erklärt.)

Einfügen von beliebig vielen Zeichen (string) oder Zeilen. Mit ctl-Z abschließen! CP steht anschließend hinter dem letzten eingefügten Zeichen. Alle folgenden Zeichen werden entsprechend weiter nach vorn verschoben.

Alle diese Kommandos können kombiniert und in einer oder mehreren Zeilen zusammengefaßt werden (ohne "RETURN" dazwischen). Dabei können folgende CONTROL-Kommandos benutzt werden. (Diese werden nicht mit "RETURN" abgeschlossen):

- BACKSPACE** (Drücken der Taste "BACKSPACE").
Löschen des zuletzt eingegebenen Zeichens.
- ^U** (Drücken der Tasten "CONTROL" und "U" gleichzeitig). Löschen der Zeile, in der sich gerade der Cursor befindet.
- ^C** Rücksprung ins Betriebssystem.
- ^E** Neue Zeile, ohne die vorherige Zeile abzusenden. (Für lange Kommando-Kombinationen bis zu maximal 128 Zeichen Länge).

Beispiele für Kommando-Kombinationen im Editor:

- B#T** Setze CP an den Anfang des Textes und gib den ganzen Text aus.
- 5COT** Bewege CP um 5 Zeichen nach rechts und gib die Zeile bis dahin aus.
- 2LOT** Setze CP um 2 Zeilen weiter vor und gib die dortige Zeile aus.
- L#K** Setze CP um eine Zeile zurück und lösche alle dann folgenden Zeilen.
- IABC^Z3D** (" ^Z" bedeutet ctl-Z.) Einfügen des Textes "ABC" und löschen der drei Zeichen, die vorher an dieser Stelle standen.

1.5.4 Suchen und Ersetzen von Text im Editor

Im Editor des CP/M gibt es das F-Kommando, um beliebige Zeichenketten im Arbeitsspeicher zu suchen.

Es hat die Form:

`nF<string>`

und wird mit `ctl-Z` abgeschlossen, wenn in der gleichen Zeile weitere Kommandos folgen. Sonst kann direkt dahinter `"RETURN"` gedrückt werden.

Dabei gibt `"<string">` eine beliebige Zeichenkette an, nach der gesucht werden soll. In ihr muß ein `zu suchendes "RETURN"` als `ctl-L` dargestellt werden. (`"<string">` sind max. 100 Zeichen).

`n` gibt an, nach der wievielten Zeichenkette `"<string">`, ausgehend von der jetzigen CP-Position, gesucht werden soll.

Direkt hinter der gefundenen Zeichenkette wird CP plziert. Ist die Suche nicht erfolgreich, so verbleibt CP an der ursprünglichen Position.

Beispiele für das F-Kommando:

(`" ^Z"` bedeutet `ctl-Z`, Befehle mit `"RETURN"` abschließen!)

`FHANS`

Suche `"HANS"` und plziere CP direkt dahinter.

`FJA ^Z-2DINEIN ^Z`

Suche `"JA"`, lösche es und füge `"NEIN"` ein.

`3F ^Z-DI-^Z`

Suche den dritten Blank und ersetze ihn durch einen Bindestrich.

Für die Ersatzvorgänge, bei denen gesucht und eingefügt werden soll, gibt es ein spezielles, einfacheres Kom-

mando, das S-Kommando:

nS<string1>-^Z<string2>

Es ersetzt die nächsten n, hinter der CP-Position folgenden "<string1>" durch "<string2>". Die "<string1>" werden automatisch gelöscht. Die Länge der beiden Zeichenketten muß nicht übereinstimmen, darf aber wieder nicht je 100 Zeichen überschreiten.

Beispiele für das S-Kommando: (Mit "RETURN" abschließen!).

10SJETZT ^ZBALD Ersetze die nächsten zehn "JETZT"
durch "BALD"!

B#S1979 ^Z1980 ^ZB#T Gehe an den Anfang, ersetze alle "1979"
im gesamten Arbeitsspeicher durch
"1980", gehe wieder an den Anfang und
gib den ganzen Text aus.

Das nun folgende N-Kommando ermöglicht es, nicht nur, wie mit dem F-Kommando, nach Zeichenketten im Arbeitsspeicher zu suchen, sondern die Suche gegebenenfalls automatisch auf die ganze Haupt-Datei auf der Disk auszudehnen. Es hat folgende Form:

nN<string>

Wird das n-te "<string>" im Arbeitsspeicher nicht gefunden, wird automatisch #W und OA ausgeführt, bis die entsprechende Zeichenkette gefunden wird, oder die gesamte Haupt-Datei gelesen ist.

Desweiteren gibt es ein sogenanntes Nebeneinanderstellungs-Kommando, das J-Kommando. (Juxtaposition command.) Es hat folgende Form:

nJ<string1> ^Z<string2> ^Z<string3>

und bezieht sich wieder nur auf den Inhalt des Arbeitsspeichers. Dabei wird "`<string1>`" gesucht, "`<string2>`" direkt dahinter eingefügt und alle Zeichen bis zum nächsten "`<string3>`" werden gelöscht. Wird "`<string3>`" nicht gefunden, findet keine Löschung statt. Der ganze Vorgang wird n mal durchgeführt.

Beispiel für das J-Kommando:

JHAUS^Z. ^Z ^L

Suche das nächste "HAUS", setze einen Punkt und einen Blank (Leerstelle) dahinter und lösche alles bis zum folgenden "RETURN".

1.5.5 Einfügen von Dateien in den bearbeiteten Text

Der Editor erlaubt auch das Einfügen ganzer Dateien in den Text, der gerade editiert wird, direkt hinter CP. Dafür verwendet man das R-Kommando, das folgende Form hat:

R<dateiname>

und wieder mit `ctl-Z` oder "RETURN" abgeschlossen wird, je nachdem, ob in der gleichen Zeile noch weitere Befehle folgen oder nicht. "`<dateiname>`" ist der Dateiname der einzufügenden File, deren Bezeichnung folgende Form haben muß:

<dateiname>.LIB

und die auf der gleichen Disk, auf der sich auch ED.COM befindet, gespeichert sein muß. Der Dateityp "LIB" ist obligatorisch.

Beispiel für das R-Kommando:

RTEST

Die Datei "TEST.LIB" wird direkt hinter CP in den Arbeitsspeicher eingefügt. Der Text hinter CP wird entsprechend weitergeschoben. Befindet sich "TEST.LIB" nicht auf der gleichen Disk, erfolgt eine Fehlermeldung.

1.5.6 Block-Move

(Textblöcke kopieren durch Schreiben und Lesen kurzfristiger Dateien)

ED enthält auch eine "Block-Move"-Möglichkeit, mit der sich ganze Textblöcke von einer Stelle des Textes an beliebige andere Stellen kopieren lassen: Das X-Kommando. Es hat die Form:

nX

und überträgt die hinter CP folgenden n Zeilen, ohne sie im Arbeitsspeicher zu löschen, in eine kurzfristige Datei mit der Bezeichnung

X\$\$\$\$\$\$\$.LIB

Diese "Bibliotheks"-Datei (Typ LIB) besteht nur während des Editier-Prozesses und wird durch das X-Kommando kreiert und durch das E-Kommando oder den Befehl

OX

automatisch gelöscht.

Diese Datei, bzw. die in ihr durch X-Kommandos angesammelten Zeilen, kann durch ein einfaches R-Kommando:

R

an beliebigen Stellen komplett wieder in den Arbeitsspeicher eingefügt und somit kopiert werden.

1.5.7 Automatische mehrfache Kommando-Wiederholung

Das M-Kommando erlaubt dem Benutzer des Editors, beliebige Editor-Kommandos (das M-Kommando natürlich ausgenommen) zusammenzufassen und automa-

tisch in mehrfacher Wiederholung abarbeiten zu lassen.
Die Form ist folgende:

nMkkk

wobei "n" die Zahl angibt, wie oft die Kommandofolge "kkk" ausgeführt werden soll. Ist n=0 oder n=1, wird die Ausführung so oft wiederholt, bis ein Fehler auftritt (z.B. wenn bei einem F-Kommando das Ende des Arbeitsspeichers erreicht ist). Die Folge "kkk" kann aus mehreren, durch ^Z abgegrenzten Befehlen bestehen, darf aber nicht länger als 128 Zeichen sein. (Der Zeilenwechsel innerhalb der Folge muß dann mit ctl-E durchgeführt werden.)

Beispiele für das M-Kommando:

MS1979 ^Z1980 ^Z0TT Ersetze jedes "1979" durch "1980" und gib jede geänderte Zeile aus.

7MF. ^ZI ^L ^ZB#T Füge hinter die nächsten sieben Punkte, die von einem Blank gefolgt werden, ein "RETURN" ein und gib den ganzen Inhalt des Arbeitsspeichers aus.

1.5.8 Fehlermeldungen des Editors

Fehlermeldungen werden vom Editor im allgemeinen in der Form:

BREAK "x" AT k

ausgegeben, wobei "x" das fehlerhafte (nicht entschlüsselbare) Zeichen ist, und "k" das Kommando, in dem der Fehler auftrat.

Für "x" können folgende Zeichen mit besonderer Bedeutung erscheinen:

? Ein falsches Kommando wurde eingegeben.

> Der Arbeitsspeicher ist voll, oder eine F-, N- oder S-Folge war zu lang.

- # Das Kommando kann nicht in der gewünschten Häufigkeit wiederholt werden.
- O Die in einem R-Kommando angegebene Datei existiert nicht als LIB-Datei auf der gleichen Disk.

Des weiteren kann die Fehlermeldung

PERM ERR DISK d

erscheinen, die bedeutet, daß beim Lesen der Disk "d" ein Lesefehler aufgetreten ist. (Wird durch einen automatischen "cyclic redundancy check" (CRC) erkannt.)

Der Benutzer kann den Fehler ignorieren, indem er irgend ein Zeichen eingibt. Besser ist es allerdings, die editierte Datei aus der Hintergrund-Datei neu herzuleiten. Tritt zum Beispiel beim Editieren der Datei "TEST.FOR" ein Lesefehler auf, so könnte man sie nach einem Rücksprung ins Betriebssystem (Warmstart) folgendermaßen erneuern:

TYPE TEST.BAK	Ausgeben der Hintergrund-Datei auf den Bildschirm, um festzustellen, ob sie die richtige Information enthält.
ERA TEST.FOR	Löschen der verunglückten Datei.
REN TEST.FOR=TEST.BAK	Umbenennen der Hintergrunddatei in die neue Hauptdatei. Danach neuer Editier-Prozeß mit TEST.FOR.

1.5.9 Zusammenfassung der CONTROL- und Editor-Kommandos

Im Editor sind folgende CONTROL-Kommandos möglich: (" ^ Z" bedeutet z.B. Drücken der Z-Taste bei niedergedrückter CONTROL-Taste.)

Kommando	Bedeutung
^C	Rücksprung ins Betriebssystem.

Kommando	Bedeutung
^E	Neue Zeile, ohne die vorherige Zeile abzusenden (anstatt "RETURN" in langen Kommandokombinationen bis maximal 128 Zeichen).
^I	Logischer Tabulator (Spalte 1, 8, 15, ...).
^L	Logisches "RETURN" in einem zu suchenden oder einzufügenden string (Zeichenkette).
^U	Löschen der Zeile, in der sich der Cursor befindet.
^Z	Ende eines zu suchenden oder einzufügenden Textstückes, wenn weitere Kommandos in der gleichen Zeile folgen (Kommando-Kombination).
BACKSPACE	(Drücken der Taste "BACKSPACE".) Löschen des zuletzt eingegebenen Zeichens. Cursor wandert um eine Spalte nach links.

Die folgende Tabelle ist eine Zusammenfassung und Kurzbeschreibung aller Editor-Kommandos:

Kommando	Bedeutung
nA	Laden von n Zeilen der Haupt-Datei in den Arbeitsspeicher. (S.59)
+/-B	CP an den Anfang/das Ende des Textes. (S.62)
+/-nC	CP um eine Spalte nach rechts/links. (S.62)
+/-nD	Löschen von n Zeichen hinter/vor CP. (S.62)
E	Abschließen des Editors und der Dateien. (S.60)
nF	Suchen einer Zeichenkette. (S.64)
H	Abschließen und Neustarten des Editors. (S.60)
I	Zeichen einfügen. (S.62)
nJ	Nebeneinanderstellen von Textteilen. (S.65)
+/-nK	Löschen von n Zeilen hinter/vor CL. (S.62)
+/-nL	CP um n Zeilen vor/zurück. (S.62)
nM	n-fache automatische Kommando-Wiederholung. (S.68)
nN	Suche des n-ten Strings in der ganzen Haupt-Datei. (S.65)
O	Zurück zur Original-Datei. (S.60)

Kommando	Bedeutung
Q	Verlassen des Editors ohne Veränderung. (S.60)
R	Lesen aus Bibliotheks-Dateien. (S.66)
RETURN	CP eine Zeile vor und Ausgeben der Zeile. (S.62)
nS	Ersetzen von Zeichenketten. (S.64)
+/-nT	Ausgeben von n Zeilen hinter/vor CP. (S.62)
nW	Laden von n Zeilen in die Kurzzeitige Datei. (S.60)
nX	Schreiben von n Zeilen in eine Bibliotheks-Datei. (S.67)
nZ	Pause, abhängig in der Länge von n.
+/-n	Bewegen von CP und Ausgeben der Zeile n. (S.62)
n:	CP an den Anfang der Zeile n. (S.62)
:n	CP an den Anfang der Zeile n und Ausgeben der Zeile. (S.62)

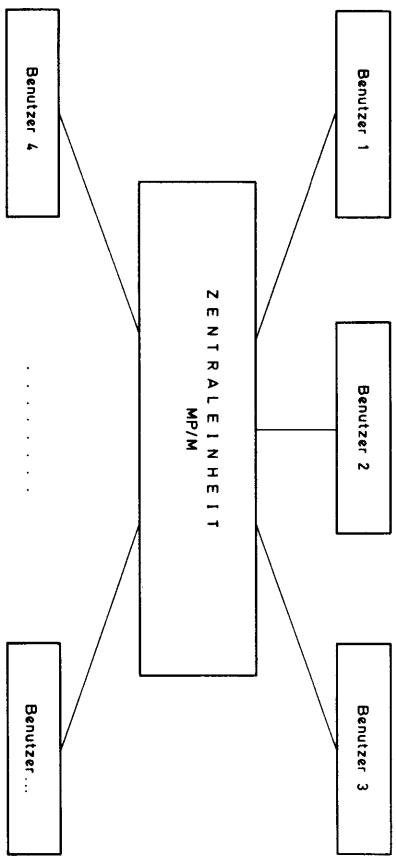


Bild 8: Das Multi-User - Betriebssystem MP/M

1.6 DAS BETRIEBSSYSTEM MP/M FÜR MULTI-USER-BETRIEB

1.6.1 Allgemeines

MP/M ist ein Betriebssystem für den Multi-User-Betrieb, d.h. es können an einer Zentraleinheit mehrere Konsolen gleichzeitig betrieben werden. Es stellt den verschiedenen Benutzern (in Verbindung mit der geeigneten Hardware) jeweils eine eigene virtuelle Maschine zur Verfügung. Dieses Betriebssystem ist aber ebenfalls für den Single-User-Betrieb, d.h. für den Betrieb mit einem Terminal geeignet.

Die Funktion, die MP/M gegenüber CP/M auszeichnet, ist die Verteilung der Hardware-Zeit an die von den verschiedenen Benutzern gestellten Aufgaben. Das damit verbundene Ein- bzw. Auslagern von aktiven und blockierten Prozessen ist dabei eine der Hauptaufgaben des Multi-User-Betriebssystems.

Um die oben genannten Anforderungen erfüllen zu können, besitzt MP/M gegenüber CP/M einige Erweiterungen, welche den Anwender unmittelbar betreffen.

Folgende Kommandos sind zu nennen:

- das "CONSOLE"-Kommando (Abschnitt 1.6.2)
- das "DSKRESET"-Kommando (Abschnitt 1.6.3)
- das "ABORT"-Kommando (Abschnitt 1.6.4)
- das "SPOOL"-Kommando (Abschnitt 1.6.5)
- das "SCHED"-Kommando (Abschnitt 1.6.6)
- das "TOD"-Kommando (Abschnitt 1.6.7)
- das "ERAQ"-Kommando (Abschnitt 1.6.8)
- das "DIR"-Kommando (Abschnitt 1.6.9)
- das "TYPE"-Kommando (Abschnitt 1.6.10)
- das "ATTACH"-Kommando (Abschnitt 1.6.11)

Das Laden des Betriebssystems geschieht bei MP/M auf die gleiche Weise wie schon bei CP/M beschrieben ("AUTOLOADVORGANG"). Nach der Ausgabe einiger Systemmeldungen meldet sich MP/M z.B. mit 0A). Dieses bedeutet, daß das Laufwerk "A" aktiviert, und der Benutzer dieser Konsole die Benutzerkennziffer "0" hat (siehe auch "USER"-Kommando im CP/M-Teil). Diese Benutzerkennziffer kann entsprechend der Beschreibung des "USER"-Kommandos im CP/M-Teil individuell verändert werden.

Alle in den folgenden Kommando-Beschreibungen nicht erwähnten Befehle können aus der CP/M-Beschreibung sinngemäß auf das Arbeiten mit dem Multi-User-Betriebssystem MP/M übertragen werden.

1.6.2 Das "CONSOLE"-Kommando

Wie bereits im vorhergehenden Kapitel geschildert, besteht die Aufgabe dieses Betriebssystems darin, das Arbeiten an mehreren Terminals zu gewährleisten. Dabei hat jedes Sichtgerät eine eigene Nummer (0 bis $n-1$), wobei n die Anzahl der aktiven Terminals ist. Wird das "CONSOLE"-Kommando in der Form

0A)CONSOLE

einggegeben, so gibt MP/M die Nummer der Konsole, an der dieses Kommando eingegeben wurde, z.B. in der Form

0A)CONSOLE = 1

an.

Dabei ist wichtig, nicht die Bedeutung der Benutzerkennziffer ("USER"-Kommando) mit der Terminalnummer zu verwechseln. Wie das obige Beispiel verdeutlicht, können beide Kennziffern durchaus unterschiedlich sein. Während die Benutzerkennziffer vom Anwender modifiziert werden kann, ist dieses bei der Terminalnummer nicht möglich.

1.6.3 Das "DSKRESET"-Kommando

Beim Multi-User-Betrieb kann der Wechsel einer Disk dann problematisch werden, wenn ein Benutzer den anderen hierüber nicht informieren kann. Ein nicht angekündigter Wechsel der Disk kann dazu führen, daß der oder die anderen Anwender in ihrem Ablauf gestört werden. Es kann z.B. ein bereits compiliertes und gebundenes Programm während des Laufes unterbrochen und damit ein nochmaliger Start des Programms notwendig werden. Um dies zu verhindern, sollte man vor jedem Diskwechsel das "DSKRESET"-Kommando anwenden. Dieses geschieht in der Form

```
0A>DSKRESET
Confirm reset disk system(Y/N)?
```

Diese Meldung erscheint auf jedem angeschlossenen Terminal. Jeder Benutzer muß dann mit "Y" (YES) quittieren. Erst dann kann der Diskwechsel erfolgen. Sollte ein Benutzer mit "N" (NO) antworten, so wird die "DSKRESET"-Anfrage verneint.

1.6.4 Das "ABORT"-Kommando

Mit dem "ABORT"-Kommando können laufende Programme abgebrochen werden. Das Kommando wird z.B. in der Form

2B)ABORT TEST2

eingegeben. Es bewirkt, daß das Programm mit dem Namen "TEST" (zur Konsole "2" gehörig) abgebrochen wird. Ebenso ist es möglich, diese Wirkung von einer anderen Konsole aus zu erzielen, indem man beispielsweise

4A)ABORT TEST2

eingibt. Jetzt wird das Programm "TEST", laufend auf der Konsole "2" vom Terminal "4" aus abgebrochen.

1.6.5 Das "SPOOL"-Kommando

Mit dem "SPOOL"-Kommando können Warteschlangen für periphere Geräte aufgebaut werden. Es findet Anwendung, wenn mehrere Benutzer zur gleichen Zeit Dateien ausdrucken wollen oder wenn ein Benutzer mehrere Dateien nacheinander an den Druck-Bereich senden will. Es können nur Text-Dateien durch das "SPOOL"-Kommando angesprochen werden. Das Kommando wird in der Form

0A)SPOOL <dateibezeichnung> <dateibezeichnung2>

eingegeben. Es können weitere Dateien als die oben angeführten angesprochen werden. Durch das angeführte Kommando-Beispiel werden die Dateien "dateibezeichnung1" und "dateibezeichnung2" in eine Warteschlange zum Drucker eingereiht.

Durch das Kommando

0A)STOPSPLR

wird die angelegte Warteschlange geleert.

1.6.6 Das "SCHED"-Kommando

Das "SCHED"-Kommando dient dem Ausführen von Programmen zu bestimmten Zeitpunkten (als Grundlage dienen die im System gespeicherten Daten Zeit und Datum; siehe auch "TOD"-Kommando). Das "SCHED"-Kommando wird z.B. in der Form

0A>SCHED MM/TT/JJ HH:MM <programmname>

einggegeben. Dabei bedeuten die Daten im einzelnen folgendes:

- | | |
|------------------|---|
| – MM | Ausführungsmonat |
| – TT | Ausführungstag |
| – JJ | Ausführungsjahr |
| – HH | Ausführungsstunde |
| – MM | Ausführungsminute |
| – <programmname> | Name des zum oben angeführten Zeitpunkt auszuführenden Programms. |

1.6.7 Das "TOD"-Kommando

Für Programme in der kommerziellen Anwendung ist es von Bedeutung, das aktuelle Tagesdatum und die Uhrzeit abrufbereit zu haben. Im Betriebssystem MP/M ist für die Initialisierung dieses Datums und der Uhrzeit ein Kommando vorhanden, das "TOD"-Kommando.

Mit der einfachsten Form dieses Befehls wird das (zuvor eingegebene, s.u.) Datum mit der Uhrzeit auf dem Bildschirm angezeigt. Das "TOD"-Kommando wird z.B. in der Form

0A>TOD

eingegeben. Daraufhin wird das aktuelle Datum und die Uhrzeit in der Form

Wochentag MM/TT/JJ HH:MM:SS

ausgegeben.

Um das Datum und die Uhrzeit zu initialisieren, werden diese Werte unmittelbar mit dem "TOD"-Kommando zusammen eingegeben. Dieses geschieht beispielsweise in der Form

0A>TOD 07/25/81 19:32:00

Das Betriebssystem meldet dann:

Strike any key to set time

Wird danach eine beliebige Taste betätigt, so werden die zu initialisierenden Daten vom System übernommen und können auf die oben beschriebene Weise abgerufen werden.

1.6.8 Das "ERAQ"-Kommando

Mit dem "ERAQ"-Kommando lassen sich Dateien löschen. Die Dateibezeichnung braucht dabei nicht eindeutig verwendet werden, d.h. "?" und "*" sind anwendbar. Das aus der CP/M-Version bekannte "ERA"-Kommando kann weiterhin in der gewohnten Form angewendet werden.

Das "ERAQ"-Kommando wird beispielsweise in der Form

0A>ERAQ TEST.*

eingegeben. Auf diese Weise können alle Dateien gelöscht werden, die den Dateinamen "TEST" besitzen, unabhängig vom jeweiligen Dateityp. MP/M fragt bei

Ausführung des ERAQ-Kommandos im Gegensatz zum ERA-Kommando nochmals an, ob der Löschvorgang endgültig durchgeführt werden soll. Der Anwender kann dann eine positive (Eingabe: Y) oder eine negative Antwort (Eingabe: N) erteilen und damit das unbeabsichtigte Löschen noch verhindern.

1.6.9 Das "DIR"-Kommando

Das "DIR"-Kommando des Multi-User-Betriebssystems MP/M bietet gegenüber dem der CP/M-Version eine Erweiterung:

Mit dem Parameter "S" ist es möglich, auch System-Dateien auflisten zu lassen.

Dieses läßt sich z.B. auf folgende Weise realisieren:

```
0A>DIR *.*S
```

1.6.10 Das "TYPE"-Kommando

Das "TYPE"-Kommando ist ebenfalls gegenüber der CP/M-Version erweitert. Durch den Parameter "P15" werden jeweils nur die folgenden 15 Zeilen der angesprochenen Datei auf dem Sichtgerät angezeigt. Durch Betätigen der "RETURN"-Taste erreicht man, daß die nächsten 15 Zeilen ausgegeben werden.

Das Kommando ist beispielsweise in der Form

```
0A>TYPE TEST P15
```

einzugeben.

1.6.11 Das "ATTACH"-Kommando

Im Betriebssystem MP/M besteht die Möglichkeit, Programme im Hintergrund ablaufen zu lassen. Vorausset-

zung dafür ist, daß der Programmname vom System gelesen wurde. Betätigt man während des Ablaufs des Programms die Tasten "CONTROL" und "D" (gleichzeitig!), so wird die Verarbeitung im Hintergrund fortgesetzt. Die Konsole ist dann frei für weitere Eingaben.

Will man das Hintergrund-Programm wieder auf der Konsole sichtbar machen, so ist das "ATTACH"-Kommando zu verwenden. Dies ist nur auf Programme anwendbar, die zuvor mittels "CONTROL D" der Hintergrundverarbeitung zugewiesen wurden. Die Form des Kommandos ist:

0A>ATTACH <programmname>

Das Programm kann nur an der Konsole wieder aktiviert werden, an der es mit "CONTROL D" vom Benutzer zum Hintergrund-Programm wurde.

WordStar: Ein menu- gesteuertes visuelles Textverarbeitungs- System für CP/M

2.1 ÜBERSICHT ÜBER DIE MÖGLICHKEITEN VON WORDSTAR

WordStar ist ein CP/M-kompatibles, bildschirmorientiertes Textverarbeitungssystem mit integrierten Druckfunktionen. Sowohl die Eingabe von neuem, als auch die Änderung von schon vorhandenem Text finden direkt auf dem Bildschirm statt. Die meisten Funktionen resultieren augenblicklich in einer Änderung des Schirmbildes, das mit dem später ausgedruckten Text übereinstimmt.

Die Editier-Funktionen werden verwendet, um Text-Dateien zu erzeugen oder zu ändern, die auf Disk (Diskette oder Magnetplatte) abgespeichert werden:

VIDEO EDITING (Editieren auf dem Bildschirm):

Ein Teil des eingetragenen oder korrigierten Textes ist ständig auf dem Bildschirm zu sehen; Einfügungen oder Korrekturen resultieren augenblicklich in einer entsprechenden Änderung des Schirmbildes. Man fügt Text

ein, indem man ihn einfach an die gewünschte Stelle schreibt. Funktionen wie Cursor-Bewegung und Löschen werden sofort mit sichtbarem Ergebnis ausgeführt.

AUTOMATIC DISK BUFFERING (automatische Disk Pufferung):

Die Größe des zu bearbeitenden Dokuments (Text-Dateien) ist nicht von der Größe des vorhandenen Arbeitsspeichers (RAM) abhängig, sondern von der Disk-Kapazität. Der Text wird während der Bearbeitung automatisch ohne Hinzutun des Benutzers auf die Disk geschrieben oder von ihr gelesen.

ON-SCREEN TEXT FORMATTING (Textformatierung auf dem Bildschirm):

Der Text wird auf dem Bildschirm genauso dargestellt, wie er beim Ausdrucken aussehen wird, und kann also vorher noch umgestaltet werden. Die Textformatierung auf dem Bildschirm wird durch die folgenden drei Funktionen unterstützt:

WORD WRAP (Wort-Umschlag):

Es können ganze Absätze sehr schnell eingegeben werden, ohne die "RETURN"-Taste zu benutzen. Wenn ein eingegebenes Wort die rechte Grenze der Zeile überschreitet, wird es von WordStar automatisch an den Anfang der nächsten Zeile gesetzt, und die vorherige Zeile wird so gestaltet, daß sie links- oder rechtsbündig ist. Die "RETURN"-Taste wird nur benutzt, um das Ende eines Absatzes, eine Leerzeile oder ein anderes wichtiges Zeilenende zu kennzeichnen. Daher braucht der Benutzer nicht seinen Rhythmus bei der Texteingabe zu unterbrechen oder auch nur von seinem Material aufzuschauen.

AUTOMATIC MARGINATION, JUSTIFICATION, LINE SPACING, CENTERING (automatische Einstellung der Ränder, Rand-Ausrichtung, Leerstellen-Einfügung, Zentrierung):

Bei eingeschaltetem Word Wrap wird jede vollständige Zeile automatisch dem gesetzten linken und rechten Rand angepaßt (wenn nicht "ragged-right"-Format eingestellt ist) und – zusätzlich möglich – jedes Wort vom anderen durch doppelte oder dreifache Leerstellen (double-spaced, triple-spaced) getrennt. Mit einem einzigen Kommando kann ein Text zwischen dem linken und rechten Rand in die Mitte positioniert werden.

REFORM PARAGRAPH COMMAND (Absatz-Neugestaltung):

Mit nur einem Befehl kann der Text von der Cursor-Position bis zum Ende des Absatzes (z.B., wenn noch etwas eingefügt wurde) "reformiert" werden. Das heißt, er wird wieder an den linken und rechten Rand angepaßt, Leerstellen werden eingefügt oder von WordStar eingefügte Leerstellen werden wieder gelöscht. Es kann ein rechtsbündiger (justified) Absatz zu einem rechts offenen (ragged-right) ohne füllende Leerstellen zwischen den Worten – und umgekehrt – umgestaltet werden.

Zusätzliche Editier-Kommandos:

Zusätzlich zu den genannten Editier-Funktionen gibt es im WordStar noch: Setzen und Löschen variabler Tabulator-Positionen, block move (Verschieben ganzer Textteile), block copy, block delete, find, replace, Schreiben auf und Lesen von anderen Dateien und Setzen von und Zurückkehren zu Textstellen-Markierungen.

FLEXIBLE FIND AND REPLACE COMMAND (flexible Such- und Ersatz-Kommandos):

Such- und Ersatz-Funktionen können einmal, n-mal oder für den ganzen Text, für ganze Wörter, Groß-Klein-Schreibung-ignorierend oder selektiv (der Benutzer wird vor jedem Ersatz-Vorgang gefragt) durchgeführt werden.

HELP SYSTEM (Hilfs-System, Menu):

Im oberen Teil des Bildschirms ist auf Wunsch eine Aufzählung der gerade verfügbaren Kommandos (Menu) mit einem wählbaren Grad an Information (help level) dargestellt. Bei Kommandos aus zwei Zeichen wird nach der Eingabe des ersten Zeichens und einer kurzen Pause ein weiteres Menu ausgegeben, das über alle nun möglichen zweiten Zeichen und deren Funktion informiert. Man kann jedoch auch beide Zeichen schnell hintereinander eingeben, wenn man bereits etwas fortgeschritten ist.

DYNAMIC PAGE BREAK DISPLAY (Darstellung des Seitenendes):

Die beim Drucken auftretenden Seitenwechsel werden beim Editieren des Textes auf dem Bildschirm durch eine durchgehende Linie dargestellt und jeder Löschung oder Einfügung angepaßt. So brauchen beim Drucken der Seitenwechsel und das Seiten-Layout nicht weiter berücksichtigt werden.

USER CONTROL OF TEXT FORMATTING (Kontrolle des Benutzers über das Text-Format):

WordStar merkt sich alle vom Benutzer eingegebenen und automatisch z.B. durch Word. Wrap eingefügten Leerstellen unabhängig voneinander. Eine eingegebene Leerstelle (hard-space) bleibt auf jeden Fall erhalten

und wird nicht durch Word Wrap oder das Reform-Kommando gelöscht. Im Gegensatz dazu werden die von WordStar eingesetzten Leerstellen (soft-space) bei einer Reformierung des Textes gegebenenfalls gelöscht (oder erweitert), um den Text dem linken und rechten Rand anzupassen. Alle automatischen Formatierungsfunktionen können durch spezielle Editier-Kommandos rückgängig gemacht oder überhaupt nach Belieben ein- oder ausgeschaltet werden.

PRINT FUNCTION (Druck-Funktion):

WordStar's Druck-Funktion wird verwendet, um die mit Editier-Funktionen eingegebenen und auf Disk gespeicherten Texte auszudrucken. Dabei kann eine Datei ausgedruckt werden, während gerade die andere editiert wird. Es stehen folgende Druck-Format-Funktionen zur Verfügung:

PAGE FORMATTING (Seiten-Format):

Jede Seite wird durch einen oberen und unteren Rand sowie zusätzlich mögliche Kopf- und Fuß-Texte und Seitenzahlen formatiert. Die Seitenlänge und der obere und untere Rand werden durch "Dot-Kommandos" festgelegt, die direkt in den Text eingefügt sind und mit ihm abgespeichert werden. Für normale Fälle machen die eingestellten Vorgaben solche besonderen Angaben jedoch unnötig. Das Seiten-Format kann innerhalb eines langen Textes beliebig häufig wechseln. Die Seitenzahl kann an verschiedene Stellen der Seite platziert oder auch ganz unterdrückt werden.

PAGINATION CONTROL (Seitenvorschub-Kontrolle):

Seitenwechsel erfolgt, wann immer eine Seite voll ist, oder es von besonderen Kommandos bestimmt wird. Es kann auch eine Gruppe von Zeilen als zusammen-

hängend definiert werden, so daß sie gegebenenfalls auf die nächste Seite gedruckt wird, falls sie nicht mehr komplett auf die vorherige Seite paßt. So wird ein unerwünschter Seitenvorschub vermieden. Das Seitenende wird beim Editieren auf dem Bildschirm angezeigt.

PRINT ENHANCEMENTS (zusätzliche Druckfunktionen):

Unterstreichungen, Doppeldruck, Fett-Druck und Durchstreichungen können durch besondere Zeichen vor und hinter dem betreffenden Text gesteuert werden. Mit entsprechenden Druckern sind auch Tief-Schrift, Hoch-Schrift, verschiedene Typen-Formen, variable Zeilen-Höhe und unterschiedliche Druck-Farben möglich. Alle diese Effekte können beliebig miteinander kombiniert werden.

MICROSPACE JUSTIFICATION (Mikro-Leerstellen-Ausgleich):

Wird randbündliger Text über einen Daisy Wheel-Drucker ausgegeben, so wird der Leerraum zwischen den Worten einer Zeile exakt stufenlos aufgeteilt. Auf dem Bildschirm und auf Druckern ohne diese Möglichkeit wird der Freiraum durch ganze Leerstellen ausgeglichen.

OPERATOR OPTIONS (zusätzliche Benutzer-Kommandos):

Beim Initiieren der Druck-Funktion kann der Benutzer noch spezielle Parameter bestimmen: Anfangs-Seiten-Nummer, Formular-Vorschub-Zeichen, Pausen zwischen dem Ausdruck einzelner Seiten zum Papier-Einfädeln, Datei-Ausdruck ohne Formatierung und Ausgabe auf Disk, anstatt den Drucker.

Druck-Operationen können jederzeit unterbrochen

werden, auch wenn gerade ein anderer Text editiert wird.

Weitere WordStar-Fähigkeiten sind:

KEYBOARD BUFFERING (Eingabe-Puffer):

Der Benutzer kann beliebig schnell Text und Kommandos hintereinander eingeben, ohne das Ergebnis auf dem Bildschirm abzuwarten. Alle Zeichen werden gespeichert und die jeweils neueste Text-Form wird auf dem Sichtgerät produziert.

Eine Ausnahme hiervon bildet die Disk-Ein- und Ausgabe, während der kein Zeichen auf der Tastatur eingegeben werden darf. Sie wird durch eine Mitteilung auf dem Bildschirm angezeigt.

CONVENIENT DEFAULTS (zweckmäßige Vorgaben):

Da WordStar sehr viele mögliche Parameter hat, sind alle schon bei der Initialisierung auf einen zweckmäßigen Wert voreingestellt. Zum Beispiel ist der linke Rand in die erste, der rechte Rand auf die 65. Spalte voreingestellt. Jeweils fünf Spalten liegen zwischen zwei Tabulator-Stops. Die Formular-Länge ist auf 66 Zeilen eingestellt.

So kann man WordStar praktisch ohne Vorkenntnis über die Änderung dieser Parameter benutzen und es dürfte nur selten notwendig werden, mehr als wenige von ihnen neu einzustellen.

FILE - KOMPATIBILITÄT:

Alle Text-Formatierungs-Funktionen können durch Einstellen eines "non-document"-mode unterdrückt werden, so daß WordStar ebenso geeignet ist, Programme oder Daten-Dateien zu bearbeiten.

BILDSCHIRM-KOMPATIBILITÄT:

WordStar kann mit jedem CP/M-kompatiblen Video-Terminal mit mindestens 16 x 64 Zeichen und freier Cursor-Positionierung betrieben werden. Möglichkeiten, Zeichen durch besonders helle oder Negativ-Darstellung hervorzuheben, werden von WordStar genutzt.

DRUCKER-KOMPATIBILITÄT:

WordStar kann fast jeden Drucker (Teletype- oder Daisy Wheel-Typ) steuern, der von CP/M-Ausgabefunktionen oder direkt von WordStars Hardware-Ein/Ausgabe-Befehlen erreicht werden kann. Bei Daisy Wheel (Typenrad) und ähnlichen Druckern werden sämtliche Möglichkeiten der Textgestaltung genutzt.

INSTALLATION:

Mit Hilfe des INSTALL-Programms kann WordStar an ca. ein Dutzend üblicher Terminals und eine der folgenden Drucker angepaßt werden:

- Diablo 1610/1620/1640/1650
- Qume Sprint 5
- NEC Spinwriter 5510/5520
- Teletype

Ungewöhnliche Terminals und Drucker werden durch kleine Programm-Änderungen installiert.

2.2 DISK-INITIALISIERUNG FÜR WORDSTAR

Um mit WordStar zu arbeiten, sollte auf der entsprechenden Diskette oder Magnetplatte folgendes gespeichert sein:

1. Das Betriebssystem (system image), das jedesmal bei Druck auf die Reset-Taste in den RAM-Bereich des

Computers geladen wird. Das ist keine eigentliche Datei, sondern ein besonders reservierter Disk-Bereich. Er wird mit einem speziellen Kommando (z.B. "DCOPY SYSTEM" – siehe Hersteller-Manual – auf die Disk im B-Laufwerk kopiert. Dazu gehören auch noch die CBIOS-Files, die man mit "PIP B:=CBIOS*.*[V]" transportiert.

2. Die WordStar-Files:

WSV.COM	(das installierte WordStar-Programm)
WSMSG.S.COM	(die File mit den Benutzer-Mitteilungen)

3. Files mit besonderen gewünschten CP/M-Kommandos, wie

PIP.COM	(zum Transportieren von Datei-Inhalten)
STAT.COM	(um sich jederzeit über den verbleibenden Platz auf der Disk zu informieren)

Mit dieser Disk in Laufwerk A können jederzeit das Betriebssystem und WordStar initialisiert und Texte gespeichert werden. Im B-Laufwerk oder einem anderen sollte eine Disk mit ausreichend Platz zur Datensicherung laufen. Um WordStar zu initialisieren, gibt man nun nur ein:

WSV	(gefolgt von "RETURN")
-----	------------------------

2.3 STARTEN VON WORDSTAR UND NO-FILE-KOMMANDOS

Nachdem das Betriebssystem gestartet wurde (abhängig

vom Typ des Computers) und sich mit A) gemeldet hat, gibt es drei Möglichkeiten, WordStar zu initialisieren:

(jeweils mit "RETURN" abschließen)

WSV Einstieg in WordStar, ohne daß schon eine Datei editiert wird. Es erscheint kurz eine copyright-Information und dann das NO-FILE-Menü, wie es weiter unten beschrieben wird.

WSV d:<dateibezeichnung>

zum Beispiel:

WSV B:TEXT.DOC Einstieg in WordStar direkt, um eine Datei zu editieren. Die Dateibezeichnung muß den üblichen Formvorschriften entsprechen (siehe "1.1.3 Bezeichnung von Dateien") Die Wirkung ist die gleiche wie bei Eingabe des Kommandos "Bearbeitung einer Text-Datei" aus dem NO-FILE-Menü, wie unten beschrieben.

WSV d:<dateibezeichnung> d:

zum Beispiel:

WSV A:TEXT.001 B: Einstieg in WordStar direkt zum Editieren einer Datei und Speichern der editierten Datei auf einem anderen Disk-Laufwerk. Dies ist für sehr lange Dateien gedacht. Im obigen Beispiel wird die Datei TEXT.001 von

Disk A editiert und die editierte Version wird auf Disk B gespeichert. (Die ursprüngliche Version auf Disk A heißt dann TEXT.BAK). Wird das "Sichern und neu editieren"-Kommando verwendet (^KS), so wird nun TEXT.001 von Disk B editiert und das Ergebnis wiederum auf Disk A gespeichert usw.

Sollte nach dem Starten von WordStar die Meldung kommen:

"You are trying to run an unINSTALLED WordStar.
Please run INSTALL first"

dann muß das WordStar erst mit Hilfe des INSTALL-Programms dem Terminal und dem Drucker angepaßt werden.

Nach dem Start von WordStar ohne eine Dateibezeichnung und immer wenn das Editieren einer Datei abgeschlossen ist, erscheint das NO-FILE-Menü auf dem Bildschirm, das die nun möglichen Kommandos aufzählt und erklärt:

Kein Text in Bearbeitung

D= Bearbeiten einer Text-Datei
N= Bearbeiten einer Programm-Datei
X= Rückkehr in das Betriebssystem
H= HILFS-Funktion setzen
P= Datei drucken
K= Datei löschen

#

(Das Symbol # stellt die Cursor-Position dar)

Der Benutzer gibt nun einen einzelnen Buchstaben ein und wählt dadurch eine der Funktionen. Der Buchstabe kann groß oder klein oder bei gedrückter CONTROL-Taste eingegeben werden. Die Wirkung ist stets die gleiche. Nicht definierte Buchstaben und Zeichen werden ignoriert. Nach dem Buchstaben wird nicht "RETURN" gedrückt! Er wird sofort akzeptiert und erscheint in der linken oberen Ecke des Bildschirms.

Die eingegebenen Buchstaben starten folgende Funktionen:

Buchstabe	Funktion	Beschreibung
D	Bearbeitung einer Text-Datei	Es wird nach einer Dateibezeichnung gefragt, dann wird die entsprechende Datei kreiert (wenn sie noch nicht existiert) oder von der Disk geladen und editiert (wenn sie bereits vorhanden ist). Um die neue editierte Version der Datei auf einer anderen als der "aktuellen" Disk zu speichern, gibt man die Dateibezeichnung, eine Leerstelle, das Laufwerk (A, B,C oder D) und einen Doppelpunkt ein. Wurde eine neue Datei kreiert, so erscheint für wenige Sekunden die Meldung "NEW FILE". Falls man sich vertippt hatte, kann man mit ^KQ wieder zum NO-FILE-Menü zurückkehren, ohne die versehentlich erzeugte Datei zu speichern.
N	Bearbeiten einer Programm-Datei	Wie D, jedoch für eine "non-document"-File ohne Seiteneinteilung und Parameter-Vorgaben. Wird benutzt um Programm- oder Daten-Files zu erstellen.
X	Rückkehr in das Betriebssystem	Rücksprung in das Betriebssystem (z.B. CP/M). Wird benutzt, um die Arbeit mit WordStar abzuschließen und wieder ein Betriebssystem-Kommando eingeben zu können. Das Betriebssystem meldet sich mit der Bezeichnung der aktivierten Disk, z.B. "A".
H	Hilfs-Funktion setzen	Es wird nach einem neuen "help level" (0 bis 3) gefragt, der die Menge der Menu-Informationen und anderer Erklärungen be-

Buchstabe	Funktion	Beschreibung
		stimmt. Wenn der help level nicht 0 ist, wird die Erklärung der help levels immer angezeigt. Wird gerade eine Datei editiert, hat das Kommando ^JH die gleiche Wirkung. Für Fortgeschrittene bringt der niedrige help level den Vorteil, daß mehr Platz für den zu editierenden Text auf dem Bildschirm zur Verfügung steht.
P	Datei drucken/ Druck abbrechen	Das P-Kommando hat drei mögliche Effekte, Je nachdem, ob es schon aktiv ist und ob gerade gedruckt wird oder nicht, wird nach dem Namen der zu druckenden Datei gefragt, oder der Druck wird unterbrochen oder fortgesetzt. Die P-Zeile im Menu zeigt den gerade vorhandenen Zustand an.
K	Datei löschen	Es wird nach dem Namen der zu löschenden File gefragt. Die Wirkung ist dieselbe wie beim CP/M ERASE-Kommando.

Beim Editieren eines Textes kann jederzeit mit ^KD zum NO-FILE-Menu zurückgekehrt werden.

2.4 DAS SCHIRMBILD VON WORDSTAR ZUM EDITIEREN VON FILES

Das Schirmbild von WordStar beim Editieren besteht aus einer Status-Zeile, dem Menu, der Tabulator-Zeile und dem File-Sektor mit den Flag-Zeichen. z.B.:

	Seite 1Zeile	3 COL 21	
Zeiger:	^A=Wort links	^S=Zeichen links ...	
	^E=Zeile hoch	^X=Zeile tief ...	
ROLLEN:	^Z=Zeile hoch	^W=Zeile tief ...	
LÖSCHEN:	DEL=Z. links	^G=Zeichen rechts	
	L — — — — — — — R		<
Probetext			<
Dies ist ein Text #			:

STATUS-ZEILE:

In der ersten Zeile, der Status-Zeile, werden folgende Informationen gegeben:

- Das Kommando, das eingegeben wurde oder gerade ausgeführt wird (gegebenenfalls).
- Die Seiten-Nummer (auf der man gerade arbeitet).
- Die Zeilen-Nummer (in der sich der Cursor befindet)
- Spalten-Nummer (COL) (in der sich der Cursor befindet).
- "Einf. EIN", falls ^V eingegeben wurde, um eine Einfügung zu machen.
- Andere zusätzliche Informationen.

Die Angabe der eingegebenen Kommandos erfolgt in der linken oberen Ecke des Bildschirms in der Reihenfolge der Eingabe. Sehr schnell auszuführende Kommandos, wie Cursor-Bewegungen werden nicht extra angegeben.

Die Seiten-Nummer bezieht sich auf die Druck-Ausgabe und beginnt bei 1. Auf dem Bildschirm wird der Wechsel der Seite durch einen durchgehenden waagrechten Strich angedeutet.

Die Zeilen-Nummer bezieht sich ebenfalls nur auf die ausdruckbaren Zeilen. Spezielle Druck-Funktions-Zeilen (dot-Kommandos), die im Druckbild nicht erscheinen, werden nicht mitgezählt.

Die Spalten-Nummer ignoriert ebenfalls Druck-Kommandos und kann auch größer sein als die Bildschirmbreite, wenn der rechte Rand entsprechend eingestellt wurde.

Wurde WordStar für eine Programm-Datei initialisiert,

so werden im Schirmbild Seiten-Nummer und Zeilen-Nummer ersetzt durch:

FC=nnnnn FL=nnnn

wobei FC die Anzahl der Zeichen zwischen der Cursor-Position und dem Anfang der File +1 angibt, und FL die Nummer der Zeile vom Anfang der File aus, in der sich der Cursor befindet (file character und file line).

So kann man z.B. die Anzahl der Zeichen in einer Programm-Datei feststellen, indem man den Cursor ganz ans Ende der File bewegt (^QC) und dann FC abließt.

Zusätzlich werden in der Status-Zeile gegebenenfalls noch folgende Informationen gegeben:

- | | |
|------------------|---|
| – WORKING | Falls von einer Disk gelesen oder auf eine geschrieben wird. Man sollte solange keine weiteren Eingaben machen. |
| – Randaus | Falls die Rand-Begrenzung gelöst wurde (^OX-Kommando). |
| – Einf. EIN | Falls "Einfügung" eingeschaltet ist (^V-Kommando) um etwas zwischen nebeneinanderstehende Zeichen einzufügen, anstatt sie zu überschreiben. Wird mit einem weiteren ^V wieder abgeschaltet. |
| – LINE SPACING n | Falls der Zeilenabstand nicht auf 1 eingestellt ist, wird der geänderte (^OS) Abstand angegeben. |
| – PRINT PAUSED | Falls der Druck durch bestimmte Steuer-Kommandos oder -Zeichen unterbrochen wurde (z.B. zwecks Papierwechsels). |
| – REPLACE (Y/N): | Falls ein Kommando zum selektiven Ersetzen bestimmter Zeichen eingegeben wurde, wird jeweils gefragt, ob ein Textteil ersetzt werden soll oder nicht. |

MENU

Das Menu unter der Status-Zeile gibt einige der möglichen Kommandos an. Nach dem Initialisieren von WordStar zeigt es die Cursor-Bewegungs-, Lösch- und Scroll-Kommandos. (Scroll heißt rollen des Schirmbildes nach oben oder unten.) Alle Kommandos innerhalb WordStar, mit Ausnahme der NO-FILE-Kommandos, die schon erklärt wurden, müssen bei gleichzeitigem Drücken der CONTROL-Taste eingegeben werden. Darauf wird durch das Zeichen " ^ " hingewiesen.

Bei Benutzung von Kommandos aus zwei Buchstaben (beginnend mit ^J, ^K, ^O, ^P oder ^Q) erscheint nach der Eingabe des ersten Buchstabens ein neues Menu, das die nun möglichen Zweit-Buchstaben angibt. Der Inhalt des Menus ist vom eingestellten help level (siehe Seite 84, H-Kommando) abhängig. Der help level kann mit dem ^JH-Kommando geändert werden. Im Menu kann unter bestimmten Bedingungen eine Reihe verschiedener Erklärungen oder Warnungen erscheinen, z.B. wenn der Druck unterbrochen wurde oder ein Wort so lang ist, daß es nicht in die Randbegrenzungen einer Zeile paßt. Diese Informationen sind selbsterklärend und verschwinden meist schon durch den nächsten Tastendruck.

TABULATOR-ZEILE

Die Tabulator-Zeile unter dem Menu zeigt den linken Text-Rand (L), den rechten Text-Rand (R) und diverse verschiebbare Tabulator-Stop-Positionen (!). Dazwischen sind Striche (—). Die Tabulator-Zeile zeigt immer die neueste Einstellung der entsprechenden Parameter. Tabulator-Stops außerhalb des linken und rechten Randes werden nur angezeigt, wenn die Ränder "gelöst"

wurden (^OX) oder Word Wrap abgeschaltet ist (^OW). Befindet sich der rechte Rand außerhalb des Schirmbildes, so werden zwei oder mehr Tabulator-Zeilen angezeigt. Die Anzeige der Tabulator-Zeile kann durch das ^OT-Kommando unterdrückt und wieder initiiert werden.

FILE-SEKTOR

Der Rest des Schirmbildes zeigt einige Zeilen des zu bearbeitenden Textes. Der dargestellte Textteil schließt immer die Cursor-Position mit ein, so daß Bildschirm-Cursor und File-Cursor übereinstimmen.

Alle Spalten des Bildschirmes, außer der ganz rechten, werden für die Text-Darstellung verwendet. Z.B: Auf einem Bildschirm mit 80 Zeichen pro Zeile werden 79 Spalten für die Text-Darstellung benutzt. Die letzte Spalte ist leer oder enthält eines der Flag-Zeichen, die noch erklärt werden.

Normalerweise zeigt jede Bildschirm-Zeile das, was in einer Zeile ausgedruckt wird; mit folgenden Ausnahmen:

Fortsetzungszeilen: Wenn eine Text-Zeile zu lang ist, um auf einer Bildschirm-Zeile dargestellt zu werden, wird sie in der nächsten Bildschirm-Zeile fortgesetzt. In diesem Fall erscheint in der letzten Spalte der ersten Zeile ein + Zeichen als Flag, um zu kennzeichnen, daß die Zeile noch weiter geht. Wenn eine Text-Zeile sehr lang ist, kann sie über drei, vier oder noch mehr Bildschirm-Zeilen gehen, die alle, außer der letzten ein + Zeichen enthalten.

Wenn sich der Cursor auf einer der Fortsetzungs-Zeilen befindet, zeigt die Spalten-Nummer in der Status-Zeile die entsprechende Druck-Spalte an. Das sind z.B. in der

ersten Fortsetzungszeile auf einem 80-Zeichen-Bildschirm die Spalten 80 bis 158.

Überdrucken von Zeilen: Für besondere Effekte ist es möglich, mehrere Zeilen der Datei nacheinander auf dieselbe Druck-Zeile ausdrucken zu lassen, so daß zwei verschiedene Zeichen in der gleichen Position gedruckt werden, um eine besondere Graphik zu bilden. Auf dem Bildschirm erscheint dann das — Zeichen als Flag in der letzten Spalte. Es deutet an, daß die folgende Bildschirm-Zeile in dieselbe Druckzeile ausgegeben werden wird. Der entsprechende Zeilenwechsel muß beim Eingeben des Textes mit ^P und RETURN erzeugt werden.

Seitenwechsel-Zeilen: Das Ende einer Druck-Seite wird auf dem Bildschirm durch eine Zeile von — Zeichen mit einem P in der letzten Spalte als Flag markiert:

—————P

Diese Zeile wird nur zur Information auf dem Bildschirm erzeugt und natürlich nicht mit ausgedruckt. Die Anzeige der Seitenwechsel-Zeilen kann durch das ^OP-Kommando unterdrückt oder erzeugt werden.

Dateiende-Zeilen: Wenn ein Text vor dem unteren Bildschirm-Rand endet, ist der Rest des Bildschirms mit leeren Zeilen — mit jeweils einem Punkt als Flag in der letzten Spalte — gefüllt. Eine neu kreierte Datei besteht ganz aus solchen Zeilen.

Dateianfangs-Zeilen: Wird der Text so weit nach unten gescrollt, daß der Anfang der Datei unter dem oberen Rand des File-Sektors liegt, so ist der dazwischenliegende Raum mit leeren Zeilen und einem Doppelpunkt als Flag gefüllt.

Normale Text-Zeilen, die nicht länger sind als die Bildschirm-Zeilen und die nicht überdruckt werden, können mit einem "hard" oder "soft" carriage return abschließen. Ersteren erzeugt der Benutzer mit der RETURN-Taste; er wird durch ein < Zeichen als Flag dargestellt. Letzterer entsteht bei der Benutzung von word wrap, wenn am Ende einer Zeile einfach weiter geschrieben wird. Word wrap paßt dann die Zeile dem linken und rechten Rand an und es erscheint in der letzten Spalte eine Blank (Leerzeichen) als Flag.

FLAG-ZEICHEN

Die meisten Flag-Zeichen, die in der letzten Spalte des File-Sektors erscheinen, wurden bereits erklärt. Hier ist noch einmal eine Zusammenfassung:

Zeichen	Bedeutung
(blank)	Die Zeile endet mit einem "soft"-carriage return. Das Zeilenende wurde von word wrap oder einem Reform-Kommando (^B) erzeugt und kann weiterhin durch diese geändert werden.
<	Die Zeile endet mit einem "hard"-carriage return, der vom Benutzer mit der RETURN-Taste eingegeben wurde. Der Zeilenwechsel kann nicht von word wrap oder vom Reform-Kommando geändert werden. Der Benutzer kann ihn jedoch wie ein einfaches Text-Zeichen mit ^G löschen.
+	Die nächste Zeile ist die Fortsetzung der markierten Zeile.
-	Die nächste Zeile wird über die markierte Zeile gedruckt werden.
.	Diese Bildschirmzeile gehört nicht mehr zum Text oder es handelt sich um die letzte Text-Zeile, die nicht mit RETURN abgeschlossen wurde.
:	Diese Bildschirmzeile gehört noch nicht zum (befindet sich über dem) Text.

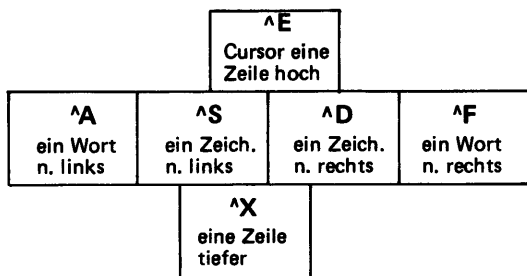
Zeichen	Bedeutung
P	Die folgende Bildschirmzeile wird auf eine neue Seite gedruckt werden.
?	Die Zeile enthält ein undefiniertes "dot-Kommando" (siehe Abschnitt 2.2.6) oder es wird gerade ein dot-Kommando eingegeben.
J	Die Zeile endet mit einem "line feed"-Zeichen ohne carriage return. Dieses ungewöhnliche Format wird nicht von WordStar erzeugt.

2.5 EDITIER-KOMMANDOS IN WORDSTAR

2.5.1 Cursor-Bewegung

Die Cursor-Bewegungs-Kommandos werden benutzt, um den Cursor innerhalb eines Textes an eine gewünschte Stelle zu setzen, um dort eine Korrektur oder Einfügung vorzunehmen. Befindet sich diese Stelle außerhalb des augenblicklichen Schirmbildes, so wird automatisch entsprechend gescrolled, um den Cursor in der entsprechenden Textstelle sichtbar zu machen.

Die wichtigsten Cursor-Kommandos werden schon durch das Menu beim Editieren erklärt. Außerdem sind die vier Grund-Funktionen auf der Tastatur in Form eines Kreuzes angelegt und daher leicht zu merken:



**Komplette Liste der Cursor-Bewegungs-Kommandos:
(alle Kommandos bei gedrückter CONTROL-Taste!)**

Kommando	Funktion	Beschreibung
^S oder ^H oder BACKSPACE	Cursor ein Zeichen nach links	Bewegt den Cursor um ein Zeichen weiter nach links, bzw. an das Ende der vorherigen Zeile, wenn er am Anfang der aktuellen Zeile steht. Wird meist verwendet um ein eben eingegebenes Zeiche zu korrigieren.
^D	Cursor ein Zeichen nach rechts	Bewegt den Cursor in gleicher Weise um ein Zeichen nach rechts, bzw. in die nächste Zeile.
^E	Cursor eine Zeile nach oben	Bewegt den Cursor um eine Zeile nach oben, möglichst in der gleichen Spalte (wenn es keine Leerzeile ist).
^X	Cursor eine Zeile nach unten	Bewegt den Cursor um eine Zeile nach unten, so weit möglich in derselben Spalte.
^QE	Cursor hoch	Bewegt den Cursor in die oberste Zeile des dargestellten Text-Teiles, so weit möglich in der gleichen Spalte.
^QX	Cursor runter	Desgleichen nach unten.
^QS	Cursor links	Bewegt den Cursor in die "linkeste" erlaubte Spalte.
^QD	Cursor rechts	Bewegt den Cursor nach rechts hinter das letzte Zeichen der Zeile.
^A	Cursor ein Wort nach links	Bewegt den Cursor zum ersten Zeichen des vorhergehenden Wortes.
^F	Cursor ein Wort nach rechts	Bewegt den Cursor an den Anfang des folgenden Wortes.
^QR	Cursor File-Anfang	Bewegt den Cursor an den Anfang des bearbeiteten Textes (incl. des nicht sichtbaren Teiles).

Kommando	Funktion	Beschreibung
^QC	Cursor File-Ende	Bewegt den Cursor hinter das letzte Zeichen des Textes.
^Q0 bis ^Q9	Cursor zur Marke	Bewegt den Cursor zu einer der zehn Text-Marken, die vom Benutzer mit den Kommandos ^K0 bis ^K9 gesetzt werden. (Siehe "2.5.8 Marken setzen"!)
^QB	Cursor z. Block- anfang	Bewegt den Cursor an den Anfang eines markierten Blocks und zeigt die dortige Markierung, falls sie unterdrückt ist.
^QK	Cursor z. Block- ende	Desgleichen zum Blockende.
^QP	Cursor-Pos. vor Kom- mando	Bewegt den Cursor auf die Position, auf der er sich vor der Eingabe des vorherigen Kommandos befand.
^QV	Cursor vor Suchen od. Block- Ursprung	Bewegt den Cursor auf die Position, auf der er sich vor dem letzten Such- oder Ersatz-Kommando befand, oder zum Ursprung des Blockes, der zuletzt verschoben, kopiert oder gelöscht wurde.
^QF	Suche String	Fragt nach einem zu suchenden Text-Teil und bewegt den Cursor dorthin. (Siehe "2.5.7 Suchen und Ersetzen".)
^I oder TAB	Tabulator	Bewegt den Cursor zur nächsten Tabulator-Position; fügt Leerzeichen ein, falls kein Text bis dorthin.
RETURN	"hard"- carriage return	Die RETURN-Taste beendet eine bearbeitete Zeile mit einem von word wrap oder Reform nicht veränderbaren Zeilenwechsel. Der Cursor wird an den Anfang der nächsten Zeile bewegt.

Anmerkungen:

Der Cursor kann nur an Stellen bewegt werden, die sich tatsächlich in der bearbeiteten Text-File befinden. Z.B. kann er nicht mehr als eine Stelle hinter das letzte Zeichen einer Zeile bewegt werden, wenn dort keine

Blanks eingefügt wurden, sondern die Zeile mit RETURN abgebrochen wurde. Dann sind nämlich in dieser Zeile hinter dem RETURN keine weiteren Zeichen existent, auf die der Cursor gesetzt werden könnte.

Außerdem kann der Cursor nicht auf Positionen innerhalb von Marken und auch nicht auf die Seitenwechsel-Zeile (mit einem P als Flag) gesetzt werden.

2.5.2 Schirmbild rollen (Scrolling)

Die Scrolling-Kommandos verändern den auf dem Bildschirm sichtbaren Textausschnitt, nicht unbedingt die Cursor-Position. Mit diesen Kommandos kann der Benutzer den gewünschten Textteil sichtbar machen.

Die vorher beschriebenen Cursor-Kommandos können auch ein Scrolling verursachen, wenn es notwendig ist, um die Cursor-Position sichtbar bleiben zu lassen.

Liste der Scrolling-Kommandos:

Kommando	Funktion	Beschreibung
^Z	eine Zeile hoch	Rollt das Schirmbild um eine Zeile nach oben. Die oberste Zeile verschwindet und eine neue unterste Zeile wird sichtbar. Der Cursor wird nur bewegt, wenn er sich in der obersten Zeile befindet, die verschwindet.
^W	eine Zeile runter	Desgleichen nach unten.
^C	Schirmbild hoch	Das sichtbare Schirmbild wird um 3/4 nach oben gerollt, d.h. es wird mit einer gewissen Überlappung im Text "geblättert". Dieses Kommando ermöglicht schnelles Wandern über lange Texte.
^R	Schirmbild runter	Desgleichen nach unten.

Werden die Tasten bei einem Kommando längere Zeit oder zusammen mit der REPEAT-Taste oder nach ^QQ eingegeben, so wird das Kommando automatisch entsprechend oft wiederholt (nach ^QQ solange, bis das Kommando wiederholt wird). Dabei unterbricht jeder Scrolling-Vorgang die Darstellung des vorherigen, und beim Loslassen der Tasten erscheint sofort das neueste Schirmbild.

2.5.3 Text-Eingabe

Normalerweise wird Text eingegeben, indem man ihn einfach an die entsprechende Stelle des Schirmbildes schreibt. Es gibt jedoch noch Kommandos, um zwischen Text-Einfügen und Text-Überschreiben zu unterscheiden sowie um die Tabulator-Positionen und word wrap zu kontrollieren:

Kommando	Funktion	Beschreibung
^V	Einfüg. ein/aus	Bestimmt, ob einzugebende Zeichen den vorhandenen Text überschreiben (Einf. aus) oder die vorhandenen Zeichen auseinanderschieben und sich zwischen sie einfügen (Einf. ein, wird durch entsprechende Mitteilung im Menu gekennzeichnet).
RETURN	Absatzende, "hard"- carriage return	<p>RETURN bei "Einf. ein" zerteilt eine Zeile in zwei Zeilen, wenn sich der Cursor mitten auf der Zeile befindet, oder erzeugt eine Leerzeile, wenn sich der Cursor am Zeilenanfang oder Zeilenende befindet. Es wird ein "hard"-carriage return eingefügt.</p> <p>RETURN bei "Einf. aus" kann benutzt werden um einen "soft" c.r. in einen "hard" c.r. zu ändern, z.B. wenn die letzte Zeile eines Absatzes gelöscht wurde.</p> <p>RETURN sollte normalerweise nicht innerhalb eines Absatzes benutzt werden. Hier läßt man die Zeilen vorteilhaft von word wrap gestalten.</p>

Kommando	Funktion	Beschreibung
^N	"hard"- c.r. einfügen	Einfügen eines "hard" carriage return an der Cursor-Position, ohne diese zu verändern. Text rechts und unterhalb des Cursors wird nach unten bewegt.
^I	Tabulator	Bewegt den Cursor zur nächsten Tabulator-Position innerhalb des linken und rechten Randes. Bei "Einf. aus": Der Cursor bewegt sich über den vorhandenen Text hinweg, die Zeile wird zwischen den vorhandenen Zeichen mit Leerzeichen, bzw. die File mit Leerzeilen aufgefüllt, wenn dies notwendig ist, um die nächste Tabulator-Position zu erreichen. Bei "Einf. ein": In den vorhandenen Text werden Blanks bis zur nächsten Tabulator-Position eingesetzt.
^OI	Setzen von Tabulator-Positionen (siehe 2.5.6 Schirmbild-Textformatierung)	
^ON	Löschen von Tabulator-Positionen (a.a.O.)	
^OF	Anpassen der Tabulator-Zeile an die Form der Cursor-Zeile. (a.a.O.)	
^Px	Control-Zeichen einfügen	Der hinter ^P folgende Buchstabe wird als Control-Zeichen in der File gespeichert, z.B. um besondere Druckformen zu steuern.

2.5.4 Text löschen

Liste der Kommandos, um Text oder einzelne Zeichen aus dem bearbeiteten Text zu löschen:

Kommando	Funktion	Beschreibung
^G	Lösche Zeichen rechts	Löscht das Zeichen auf der Cursor-Position. Am Ende einer Zeile: Löschen des carriage returns, dadurch Verbindung von zwei Zeilen.

Kommando	Funktion	Beschreibung
DELETE oder ^ _	Löschen Zeichen links	Löscht das Zeichen links von der Cursor-Position. Am Anfang der Zeile: Löschen des carriage returns, dadurch Verbinden mit der vorherigen Zeile.
^T	Löschen Wort rechts	Löscht das Wort, bzw. den Wortteil rechts vom Cursor. Wenn Cursor außerhalb des Wortes: Löschen auch der Blanks und carriage returns bis zum Wort. Wenn Cursor mitten im Wort: Löschen auch der Blanks danach.
^Y	Löschen Zeile	Löschen der Zeile, in der sich der Cursor befindet (inkl. Fortsetzungs-Zeilen und Überdruck-Zeilen).
^QY	Löschen Zeile rechts	Löschen einer Zeile von der Cursor-Position ab nach rechts (ohne carriage return oder Überdruck-Zeile).
^QDELETE oder ^Q^ _	Löschen Zeile links	Derselben links vom Cursor.
^KY	Löschen Block	Löschen eines markierten Blockes. (Siehe "2.5.9 Block Move-Kommandos"!)

2.5.5 Sichern und Beenden

Ein neu erstellter oder gerade geänderter Text wird nicht von selbst ständig auf die Disk gespeichert. Man muß ihn selbst von Zeit zu Zeit sichern. Außerdem kann nur eine gesicherte Text-Datei ausgedruckt werden.

Man sollte während der Verarbeitung langer Texte schon jeweils einen Teil sichern, um ihn ausdrucken zu können oder damit er bei Stromausfall, System-Crash, Kommando-Fehlern oder voller Disk nicht verloren geht.

Liste der Sicherungs- und Beendigungs-Kommandos:

Kommando	Funktion	Beschreibung
^KD	Datei- editieren beendet	Sichern des editierten Textes auf die Disk und Rückkehr zum NO-FILE-Me- nu. Von dort nächste Datei aufrufbar.
^KX	Editieren beendet	Sichern und Rückkehr zum Betriebs- system (WordStar verlassen).
^KS	Sichern und neu editieren	Sichern und weiterarbeiten mit demsel- ben Text. Cursor kann mit ^QP an die vorherige Position zurückgesetzt wer- den. (Entspricht ^KD gefolgt von D sowie der Datei-Bezeichnung und RETURN).
^KQ	Abbrechen	Zurück zum NO-FILE-Menu, ohne die gerade erstellte Text-Version zu si- chern. Fragt nach ja/nein-Entscheid, falls eine Änderung im Text vorgenom- men wurde.

2.5.6 Schirmbild-Textformatierung

Die folgenden Kommandos werden verwendet, um Text auf dem Bildschirm in eine bestimmte Form zu bringen und um die Funktionen von word wrap und Reform (^B) zu kontrollieren:

Kommando	Funktion	Beschreibung
^OC	Zeile zentrieren	Die Zeile, in der sich der Cursor befin- det, wird zwischen dem linken und rechten Rand zentriert (in die Mitte ge- gesetzt). Blanks am Anfang oder Ende der Zeile werden ignoriert.
^OL	linken Rand einstellen	Fragt nach dem linken Rand. Es ist eine Zahl von 1 bis 240 einzugeben, gefolgt von RETURN. Oder durch Drücken der Taste ESCAPE wird die aktuelle Cursor-Spalte als linker Rand übernom- men.
^OR	rechten Rand einstellen	Derselben für rechten Rand.

Kommando	Funktion	Beschreibung
		Der linke und rechte Rand sollten immer auf den gewünschten Wert eingestellt sein, wenn Text mit Hilfe von word wrap eingegeben oder das Reform-Kommando (^B) verwendet wird.
^OF	Ränder und Tab's von der Text-Zeile übernehmen	Setzt den linken Rand auf die Spalte des ersten Zeichens der Cursor-Zeile, das kein Blank ist und den rechten Rand auf das letzte Zeichen ungleich Blank. Überall, wo sich in der Zeile ein "I" befindet, wird eine Tabulator-Stop-Position gesetzt, und wo ein "-" steht, wird eine event. vorhandene Tab.-Pos. gelöscht. Damit kann in eine Datei eine Kopf-Zeile zum Setzen der Tab's eingebaut werden.
^OS	Zeilenabstand setzen	Fragt nach der Zahl der Zeilen, die zwischen zwei Textzeilen frei bleiben sollen und immer von word wrap, Reform oder RETURN eingefügt werden.
^B	Reform-Absatz	"Reformiert" einen Text-Absatz von der Cursor-Position bis zum nächsten "hard"-carriage return. Paßt den Text dem linken und rechten Rand an und löscht Blanks oder fügt Blanks ein, um die Worte gleichmäßig über die Zeilen zu verteilen. Verschiebt auch Worte über mehrere Zeilen, bis der Text zwischen die Ränder paßt. Der eingestellte Zeilenabstand wird berücksichtigt. Der Cursor steht anschließend rechts von dem, dem Absatz folgenden "hard"-carriage return, so daß der nächste Absatz wieder einfach durch ^B reformiert werden kann.
^OX	Rand lösen/ setzen	Linker und rechter Rand werden "gelöst" (beschränken den Cursor und den Text nicht mehr), bis der Cursor einmal außerhalb der Ränder war und wieder zurückgekehrt ist. Im Menu erscheint die Information "Randaus". Kann durch ein weiteres ^OX rück-

Kommando	Funktion	Beschreibung
		gänglich gemacht werden. Gleiche Wirkung, wie kurzzeitiges Ausschalten von word wrap.
^OW	word wrap aus/ein	Schaltet word wrap aus, bzw. wieder ein.
^OJ	Blocksatz aus/ein	Schaltet "soft blanks", die von word wrap oder Reform eingefügt werden — order wurden — aus, bzw. wieder ein. Ein folgendes Reform-Kommando (^B) löscht alle "soft blanks" und macht so aus rechtsbündigem Text rechtsoffenen Text, wenn Blocksatz ausgeschaltet wurde. Umgekehrtes bei eingeschaltetem Blocksatz. Das ^O-Menü, das nach Eingabe von ^O und kurzem Warten erscheint, informiert über den eingestellten Zustand.
^OI	Tab's setzen	Fragt nach der Spalte, in der eine Tabulator-Position gesetzt werden soll. In der Tabulator-Zeile erscheint an entsprechender Stelle ein "I".
^ON	Tab's löschen	Fragt nach der Spalte, in der die Tabulator-Position gelöscht werden soll.
^I	Tab	(siehe "2.5.1 Cursor-Bewegung")
^OV	Tab's aus/ein	Schaltet die Wirkung der Tabulator-Positionen aus, bzw. ein. Diese werden aber weiterhin in der Tabulator-Zeile angezeigt.
^OT	Tab.-Zeile aus/ein	Schaltet die Anzeige (und nur diese) der Tabulator-Zeile aus, bzw. ein.
^OP	Seitenw.-Zeile aus/ein	Schaltet die Anzeige der Seitenwechsel- (mit P-Flag) aus, bzw. ein.
^PO	Druck-Blank	Fügt an gewünschten Stellen ein ^O in den Text ein. Das bewirkt, daß an dieser Stelle ein Blank gedruckt wird, der aber im Schirmbild nicht die Zeilenlänge etc. beeinflusst.

Kommando	Funktion	Beschreibung
^PRETURN oder ^PM	über- drucken Zeile	(Siehe "Überdrucken von Zeilen")
^O	Menu	Das ^O-Menu zeigt an, ob folgende Funktionen ein- oder ausgeschaltet sind: word wrap (Wortumbruch) Blocksatz ("soft blanks") Tab's (variabler Tabulator) Tab.-Zeile anzeigen (Lineal) Seitenanzeige (Seitenwechsel)

2.5.7 Suchen und Ersetzen

Zum Suchen und Ersetzen von Strings (Zeichenketten) stehen folgende Kommandos zur Verfügung: (Jede String- oder Option-Eingabe mit RETURN abschließen!)

Kommando	Funktion	Beschreibung
^QF	suche String	Fragt nach dem zu suchenden String und nach Options. Wenn keine Options angegeben werden (einfach RETURN drücken), wird der Cursor vorwärts bewegt, bis zum ersten Auftreten exakt dieser Zeichenkette. (Options und Spezial-Zeichen im String siehe unten!)
^QA	suche und ersetze String	Fragt nach dem zu suchenden und dem dafür einzusetzenden String und nach Options. Ohne Options wird das nächste Auftreten des Strings hinter der Cursor-Position angezeigt und der Benutzer wird gefragt, ob das Ersetzen stattfinden soll, oder nicht: REPLACE (Y/N)
^L	suche/ ersetze nochmal	Das zuletzt eingegebene Such- oder Ersatz-Kommando wird wiederholt (von der neuen Cursor-Position aus), so als ob es mit den gleichen Parametern noch einmal eingegeben worden wäre.

Kommando	Funktion	Beschreibung
^QV	Cursor auf vorherige Position	Nach einem Such-, Ersatz- oder ^L -Kommando bringt dieser Befehl den Cursor auf die Position zurück, die er vor dem Kommando, bzw. vor der letzten Ausführung des mehrfach auszuführenden Kommandos innehatte. Sinnvoll z.B. nach der Fehlermeldung "NOT FOUND".

Liste der Options (auch mehrere gleichzeitig):

- n** Eine Zahl von 1 bis 65535. Suchen oder Ersetzen wird n mal durchgeführt, z.B. um die Stelle zu finden, an der ein String zum n-ten mal im Text auftaucht, oder um die nächsten n gleichartigen Strings zu ersetzen. "NOT FOUND"-Fehlermeldung erscheint, wenn die Funktion nicht entsprechend of ausgeführt werden kann. Sie wird aber so oft wie möglich ausgeführt.
- G** Beim Suchen: Suchen des letzten gleichartigen Strings.
Beim Ersetzen: Alle gleichartigen Strings in der ganzen File (von ihrem Anfang an!) werden ersetzt. Dabei wird der Benutzer jedesmal gefragt, ob ersetzt werden soll oder nicht. "NOT FOUND"-Fehlermeldung erscheint nur, falls der String kein einziges Mal im Text auftaucht.
- N** Ersetzen ohne zu fragen, sonst wie G.
- B** Suchen bzw. Ersetzen rückwärts von der Cursor-Position ab zum File-Anfang hin. Zusammen mit G: Beginn am File-Ende.
- U** Bewirkt, daß der Unterschied von Groß- und Kleinbuchstaben in den Strings ignoriert wird. Z.B. entspricht dann ein "Haus" auch einem gesuchten "haus".
- W** Bewirkt, daß nur nach passenden ganzen Wörtern gesucht wird, also entsprechenden Strings, die vor oder hinter sich keinen Buchstaben und keine Ziffer haben. Z.B. befriedigt "Haus" dann nicht die Suche nach "aus".

Die Länge des zu suchenden Strings ist auf 30 Zeichen beschränkt. Der Unterschied zwischen "hard space" und "soft space" wird beim Suchen ignoriert.

Die Frage "REPLACE (Y/N)" muß mit y, Y oder **^Y**

beantwortet werden, falls der Ersatz stattfinden soll; sonst kann man jede beliebige andere Taste drücken damit nicht ersetzt wird).

Wird der String nach "SUCHE" beim Such-Kommando bzw. der String nach "AUSTAUSCHEN ?" beim Ersatz-Kommando mit der ESCAPE-Taste abgeschlossen, so wird nicht nach Options gefragt.

Der String, den man nach "SUCHE" eingibt, kann einige spezielle Zeichen enthalten, um die Suche zu verallgemeinern:

- ^A** Jedes einzelne Zeichen befriedigt diese Stelle im Such-String.
- ^S** Paßt zu jedem Zeichen, das kein Buchstabe und keine Ziffer ist. (Muß mit **^P ^S** in den String eingefügt werden.)
- ^Ox** Paßt zu jedem Zeichen außer "x", welches ein beliebiges Zeichen darstellt.
- ^N** Entspricht der Folge carriage return, line feed (die normalerweise zwischen zwei Zeilen in der File steht). Der Unterschied zwischen "hard"- und "soft"-carriage return wird ignoriert.
^N kann auch im "AUSTAUSCH ?"-String verwendet werden und fügt dann einen "hard"-carriage return ein.

2.5.8 Marken setzen

Eine Stellen-Markierung (Marke) kann an die Cursor-Position gesetzt werden, indem man **^K** eingibt, gefolgt von einer Ziffer von 0 bis 9. Die Markierung wird dann an dieser Stelle durch die Ziffer kenntlich gemacht.

Der Cursor kann zu jeder gesetzten Marke bewegt werden, indem man **^Q** eingibt, gefolgt von der entsprechenden Ziffer (der normale Weg, um sich über weite Strecken durch den Text zu bewegen).

Die Anzeige einer Marke kann mit **^K** gelöscht werden, wenn sich der Cursor an ihrer Position befindet. Die Position wird trotzdem gespeichert und bei **^Q**, gefolgt

von der passenden Ziffer, wieder angezeigt.

Liste der Markierungsbefehle:

Kommando	Beschreibung
^K0 bis ^K9	Marke an Cursor-Position setzen. Falls sich dort schon eine Marke befindet, wird sie nicht mehr dargestellt.
^Q0 bis ^Q9	Bewegt den Cursor zur angegebenen Markierung. Falls sie nicht existiert, erscheint eine Fehlermeldung. Eine gelöschte Marke wird erneut angezeigt und verwendet.

2.5.9 Block move-Kommandos

Mit den Block move-Kommandos kann ein bestimmter Text-Block markiert und verschoben, kopiert oder gelöscht werden.

Liste der Block move-Kommandos:

Kommando	Funktion	Beschreibung
^KB	markiere Block-Anfang	Setzt die Block-Anfangs-Markierung, dargestellt durch "(B)", auf die Cursor-Position oder löscht die Markierung, wenn sie schon dort steht.
^KK	markiere Block-Ende	Setzt die Block-End-Markierung an die Cursor-Position.
^KV	verschiebe Block	Der markierte Text-Block wird zusammen mit den Markierungen an die Cursor-Position verschoben (er beginnt dann an der Cursor-Position).
^KC	kopiere Block	Der markierte Text-Block wird an die Cursor-Position kopiert. Der Original-Text bleibt unverändert. Die Markierungen bewegen sich mit der Kopie.
^KY	lösche Block	Der markierte Text-Block wird gelöscht.

Kommando	Funktion	Beschreibung
^KW	kopiere Block auf andere Datei	Der markierte Block wird auf eine andere File kopiert, nach der gefragt wird. Die Eingabe der entsprechenden Dateibezeichnung muß mit "RETURN" abgeschlossen werden. Dort eventuell schon vorhandener Text wird gelöscht. Die Ausgangs-File bleibt unverändert.
^KH	unterdr./anzeigen Marken	Die Block-Markierungen werden unterdrückt (nicht mehr angezeigt), bzw. falls sie schon unterdrückt sind, wieder erzeugt.
^QB	Cursor Block-Anfang	Der Cursor wird auf die Block-Anfangs-Markierung gesetzt.
^QK	Cursor Block-Ende	Setzt den Cursor auf die Block-End-Markierung.
^QV	Cursor Block-Quelle	Setzt den Cursor auf die Position der Block-Anfangs-Markierung von der Quelle eines Blockes, der kopiert, verschoben oder gelöscht wurde. Gut, um dort zu kontrollieren, wo der Text herkam.

2.5.10 Zusatz-Files

Mit folgenden Kommandos kann nicht nur die gerade von WordStar bearbeitete, sondern auch jede andere File beeinflusst werden:

Kommando	Funktion	Beschreibung
^KJ	File löschen	Fragt nach einer File, die gelöscht werden soll. Gut z.B., um eine nur kurzzeitig zum Text-Verschieben (zwischen verschiedenen Dateien) benötigte File zu löschen, wenn sie nicht mehr gebraucht wird. Gut auch, um unnötige Files zu löschen, wenn die bearbeitete File nicht mehr auf die volle Disk paßt.

Kommando	Funktion	Beschreibung
^KR	File lesen	Fragt nach einer zu lesenden File, die daraufhin vollständig an die Cursor-Position kopiert wird.
^KW	in File schreiben	Fragt nach einer File, in die ein markierter Block geschrieben werden soll (siehe "2.5.9 Block move").
^R	dieselbe File	Als Antwort auf die Frage nach einer Zusatz-File, aus der gelesen, oder in die geschrieben werden soll. Bewirkt die Ausführung des Kommandos mit derselben File, die beim vorherigen Zusatz-File-Kommando eingegeben wurde. (Anschließend RETURN.)

2.5.11 Help-Kommandos

Kommando	Beschreibung
^JH	Festlegen des Help-Levels (0 bis 3), der die Menge der dargestellten Hilfs-Information bestimmt. Der gerade eingestellte Help-Level wird angezeigt und nach einem neuen gefragt.
^P	Darstellung des Menus für besondere Druck-Kontroll-Kommandos.
^O	Anzeigen der eingestellten Parameter, wie z.B. word wrap ein/aus.

2.6 DRUCK-FUNKTIONEN

2.6.1 Druck-Kontroll-Zeichen

Die Druck-Kontroll-Zeichen sind einzelne CONTROL-Zeichen, die während des Editierens in eine File geschrieben werden, um Funktionen wie Unterstreichen, Hochschrift etc. zu steuern. Die Zeichen werden in die File geschrieben, indem man davor ein ^P eingibt (CONTROL-Taste und "P"!).

Die Druck-Kontroll-Zeichen werden bei der Spalten-Numerierung nicht mitgezählt und nicht gedruckt. Eine Zeile mit solchen Zeichen erscheint daher nur auf dem Bildschirm zu lang.

Das Druck-Kontroll-Zeichen muß vor und hinter dem zu beeinflussenden String stehen. Z.B. ergibt der String

das Wort ^SUnterstreichung ^S ist unterstrichen
beim Ausdruck:

das Wort Unterstreichung ist unterstrichen

Zeichen (mit ^P!)	Funktion
^S	Unterstreichen. (Blanks werden nicht unterstrichen.)
^B	Fettdruck (gleicher Buchstabe, etwas versetzt).
^D	Doppeldruck. Schreibt jeden Buchstaben zweimal an derselben Stelle und hebt ihn dadurch hervor.
^X	Durchstreichen des Strings mit einer Folge von "-" Zeichen.
^V	Tiefschrift. Schreibt den String etwas tiefer als die sonstige Zeile. Maß der Tiefe kann mit dem .SR-Dot-Kommando eingestellt werden.
^T	Hochschrift. Schreibt den String höher als die übrige Zeile.
^Y	Farbwechsel. (Farbband-Umschaltung.)

Zeichen (mit ^PI)	Funktion
^C	Unterbricht den Ausdruck an dieser Stelle, z.B. zwecks Farbbandwechsels.
^A	Umschaltung der Druck-Weite (Zeichenabstand). Normalerweise nur auf Daisy Wheel-Druckern.
^N	Zurückschalten der Druck-Weite.
^K	Strings für Dot-Kommandos.
^F	Druckt auf Daisy Wheel-Drucker das Zeichen mit dem Hex-Code 20, normalerweise das "cents"-Zeichen.
^G	Druckt das Zeichen mit dem Hex-Code 7F.
^O	Druckt ein Blank, das keinen Einfluß auf die Bildschirm-Zeilenlänge hat.
^H	Das folgende Zeichen wird über das vorherige gedruckt.

2.6.2 Dot-Kommandos (Punkt-Kommandos)

Die Dot-Kommandos sind spezielle Zeilen, die in die File geschrieben werden, um z.B. die Papierlänge oder eine bestimmte Form des Seiten-Kopfes festzulegen. Alle Dot-Kommando-Parameter sind auf gebräuchliche Werte voreingestellt, so daß man mit WordStar arbeiten kann, ohne Dot-Kommandos zu benutzen. Dot-Kommandos werden nicht mit ausgedruckt, kontrollieren aber das Druck-Bild.

Ein Dot-Kommando besteht aus einem Punkt in der ersten (!) Spalte der Zeile, einem Zwei-Buchstaben-Code und häufig noch einer Zahl oder einem Text. Die Dot-Kommandos werden wie normaler Text eingegeben, von WordStar erkannt und bei der Zeilen-Nummerierung nicht mitgezählt.

Die Bezeichnungen der Dot-Kommandos beziehen sich auf eine bestimmte Form der Seitenaufteilung.(Bild 9)

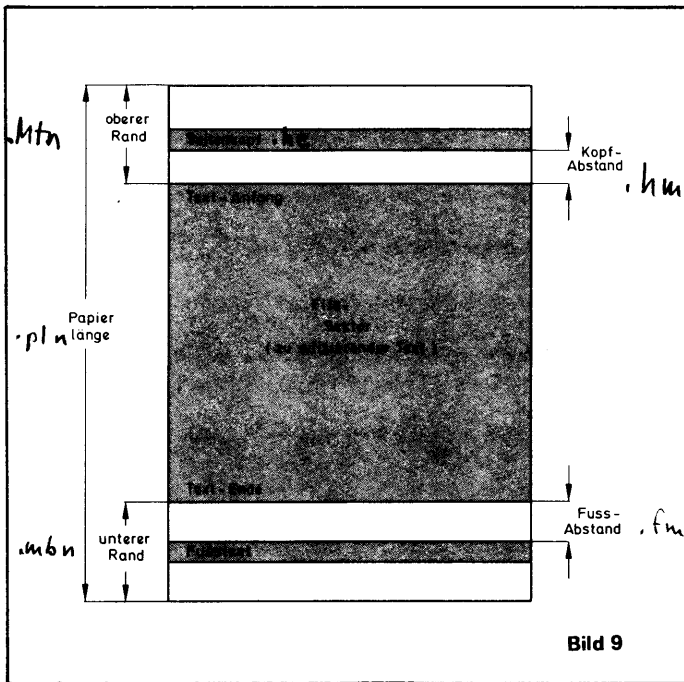


Bild 9

Liste der Dot-Kommandos:

Kommando	Funktion	Beschreibung
.LH n	Zeilen-Abstand	Einstellen des Zeilen-Abstandes in 1/48". ".LH 8" ergibt 6 Zeilen per inch, ".LH 6" ergibt 8 Zeilen per inch usw. Die Voreinstellung beträgt 6 Zeilen per inch. Dieses Dot-Kommando muß am Anfang der File vor allem Text stehen.
.PL n	Papier-Länge	n ist die Anzahl der Zeilen je Papier-Seite, inkl. oberem und unterem Rand. Vorgabe: 66 Zeilen pro Seite (= 11 inch) Muß am Anfang der File stehen.
.MT n	oberer Rand	Anzahl der Zeilen vom oberen Papier-Rand bis zum Textanfang. In diesen Raum wird gegebenenfalls der Seiten-Kopf und der Kopf-Abstand gedruckt.

Kommando	Funktion	Beschreibung
		Vorgabe: 3 Zeilen. Muß am Anfang der File stehen.
.MB n	unterer Rand	Anzahl der Zeilen zwischen Text und unterem Papier-Ende. In diesen Raum wird gegebenenfalls die Seiten-Nummer oder der Fuß-Text gedruckt. Vorgabe: 8 Zeilen. Muß am Anfang der File stehen.

Die maximal für den Text verfügbare Zeilen-Anzahl berechnet sich aus der Papier-Länge minus oberem Rand minus unterem Rand:

$$\text{Zeilen-Anzahl} = \text{Papierlänge} - \text{oberer Rand} - \text{unterer Rand}$$

Die obigen Kommandos können auch innerhalb einer File eingegeben werden, um die Seitenaufteilung o.ä. während des Ausdrucks zu ändern.

Weitere Dot-Kommandos:

Kommando	Funktion	Beschreibung
.HM n	Kopf-Abstand	Anzahl der Leerzeilen zwischen dem Seiten-Kopf (siehe .HE-Kommando) und dem Text-Anfang. Vorgabe: 2 Zeilen.
.FM n	Fuß-Abstand	Anzahl der Zeilen zwischen dem Text-Ende und dem Fußtext bzw. der Seiten-Nummer (siehe .FO-Kommando).
.PC n	Seiten-Nummern-Spalte	Spalte, in der die Zeilen-Nummer ausgedruckt werden soll, wenn kein Fußtext eingegeben und der Seiten-Nummer-Druck nicht abgeschaltet ist (siehe .OP-Kommando). Vorgabe: 1/2 Text-Breite (eingestellter rechter Rand).
.PO n	linker Druck-Rand	Anzahl der Spalten zwischen der ersten Text-Spalte und der ersten Drucker-Spalte. Der Text kann damit beliebig von der Randlochung bei Endlos-Papier abgesetzt werden. Vorgabe: 8 Spalten.

Kommando	Funktion	Beschreibung
.PA	neue Seite	Sofortiger Vorschub auf eine neue Seite.
.CP n	event. neue Seite	Seitenwechsel, falls noch weniger als n Zeilen auf der Papier-Seite zur Verfügung stehen.
.HE . . .	Seiten-Kopf	Der in dieser Zeile folgende Text wird über jede der folgenden Druck-Seiten als Seiten-Kopf ausgedruckt, bis erneut ein .HE-Kommando erfolgt. Vorgabe: kein Text (Blank). Spezielle Zeichen: siehe "Fuß-Text".
.FO . . .	Fuß-Text	Der in der Zeile folgende Text wird auf der aktuellen Seite und allen folgenden Seiten (bis zum nächsten .FO) als Fußnote bzw. Fuß-Text ausgedruckt. Vorgabe: Ausdruck der Seiten-Nummer. Sonst wie .HE-Kommando. Folgende Zeichen können speziell im Seiten-Kopf- und Fuß-Text verwendet werden: # Ausdruck der aktuellen Seiten-Nummer an dieser Stelle. \ Das folgende Zeichen wird nicht als Kopf/Fuß-Text interpretiert. ^K Die folgenden Blanks werden bis zum ersten Nicht-Blank-Zeichen ignoriert, wenn die Seiten-Nummer gerade ist. Z.B. um für Druck-Vorlagen, die gebunden werden sollen, den Kopf oder Fuß abwechselnd in die linke oder rechte Ecke zu schreiben.
.OP	keine Seiten-Nummer	Die Vorgabe "Ausdruck der Seiten-Nummer als Fuß-Text" wird unterdrückt. Ein eingegebener Fuß-Text würde ebenso die Seiten-Nummer unterdrücken. Sie müßte durch ein #Zeichen neu festgelegt werden.
.PN n	Seiten-Nummer	Festlegen der Seiten-Nummer, bei der weitergezählt werden soll. Vorgabe:

Kommando	Funktion	Beschreibung
		Anfang bei "1". Einschalten des Seiten-Nummer-Drucks, falls .OP-Kommando gegeben wurde: mit n = Blank.
.CW n	Zeichen-Abstand	Einstellen des Zeichen-Abstandes bei Daisy Wheel-Druckern in 1/120". Vorgabe: 10 Zeichen per inch (entspricht ".CW 12". Vorgabe des alternativen Zeichen-Abstandes: 12 Zeichen per inch ("CW 10").
.SR n	Hoch-/Tiefschrift Vorschub	Einstellen des Hochschrift-/Tiefschrift-Vorschubs bevor Text gedruckt wird in 1/48 inch-Schritten. Vorgabe: 3/48 inch.
.IG ... oder	ignorieren	Der in dieser Zeile folgende Text wird ignoriert und nicht ausgedruckt. Z.B.: Kommentar-Zeile.

2.7 FEHLERMELDUNGEN

Für WordStars-Fehlermeldungen und Mitteilungen ist es notwendig, daß sich auf der Disk, auf der gerade gearbeitet wird, die File WSMGS.COM befindet.

Viele Fehlermeldungen verlangen vom Benutzer das Quittieren der Information mit der ESCAPE-Taste, um sicherzustellen, daß WordStar nicht schon das nächste Kommando bearbeitet, das der Benutzer vielleicht eingegeben hat, ohne die Fehlermeldung zu sehen. Diese Fehlermeldungen haben meist folgendes Form:

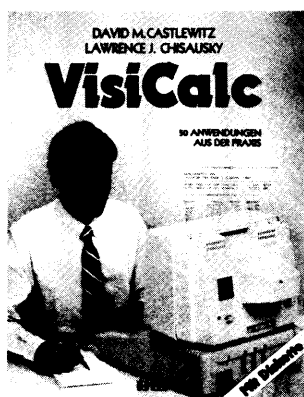
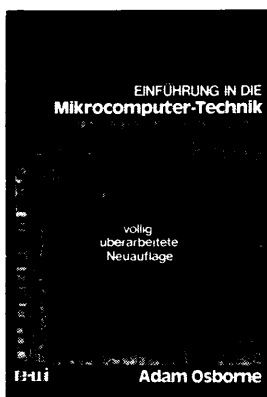
*** ERROR n: (spezielle Mitteilung) *** Esc drücken #

Sollte die File WSMGS.COM nicht verfügbar sein, so lautet die Mitteilung:

*** ERROR n: @@@@ *** Esc drücken #

Das @@@@ wird anstelle der Mitteilung ausgegeben, die sonst von WSMGS.COM gelesen worden wäre. n bedeutet die Fehlernummer.

Die Mitteilungen sind im allgemeinen selbst-erklärend und weisen den Benutzer eindeutig auf den jeweiligen Fehler hin. Die Fehler-Nummern können, falls die Texte nicht erscheinen, in einer speziellen WordStar-Fehlerliste des Herstellers nachgesehen werden.



Einführung in die Mikrocomputer-Technik — jetzt völlig neu!

von Adam Osborne

Das schon legendäre Standardwerk des Autors wurde in dieser Ausgabe restlos überarbeitet. Es spiegelt den allerletzten Stand dieser bedeutenden Technologie wieder. Geblieben ist aber die Osborne-typische Darstellung, in der auch vielleicht komplizierte Vorgänge bildhaft verständlich gemacht werden. Wer dieses Buch gelesen hat, kennt sich aus in der Welt der Mikrocomputer-Technik.

488 Seiten, farbiger Schutzumschlag, Hardcover, DM 66,—

VISICALC

von David M. Castlewitz und Lawrence L. Chisauksy

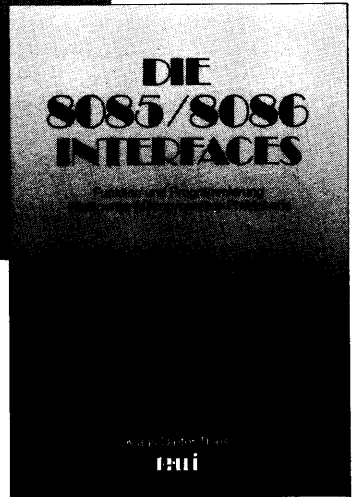
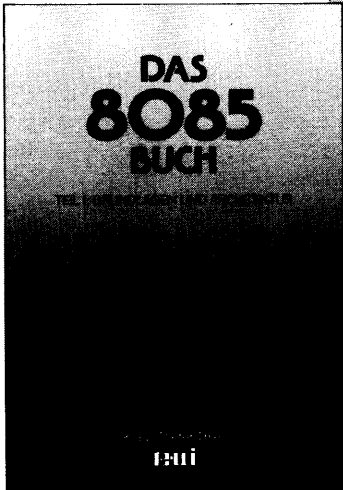
Mit diesem Buch verwandeln Sie Ihren Computer in einen elektronischen Berechnungsbogen, der Ihnen Anwendungen in Wirtschaft und Privatbereich eröffnet. Erstellen Sie Ihre Abschreibungsberechnungen oder Bilanzen an Hand von insgesamt 51 VisiCalc-Modellen, die auf der dem Buch beigelegten Diskette vorhanden sind. Zur direkten Einlesung in Ihren Computer. Erfolgreich getestet auf APPLE, CBM, IBM.

180 Seiten, Diskette, DM 69,—

Die Preise sind die Ladenpreise.

Das abgeschlossene Schulungsprogramm für die Mikroprozessoren 8085/8086 von Klaus-Dieter Thies.

Als Unterrichtsmaterial vom Hersteller des SAB 8085A und SAB 8086 -der Siemens AG - an den firmeneigenen Schulungszentren eingeführt.



DAS 8085 BUCH

TEIL I: Grundlagen und Architektur
278 Seiten, DIN A4, Paperback, DM 49,--

TEIL II: Software
310 Seiten, DIN A4, Paperback, DM 49,--

Dieses Werk führt in vorbildlicher Form in Begriffe und Anwendungen der Mikroprozessortechnik ein — demonstriert am 8-Bit-Mikroprozessor 8085. **Lehrer** können leicht Projektionsfolien aus den DIN A4-Seiten des Buches herstellen. **Entwickler** können erste Erfahrungen mit dem SAB 8085A und die Entwicklung eines Prototyps miteinander verbinden: über die Experimentiersysteme ECB 85 und SITRAIN 701 der Siemens AG — in deren Begleitmaterial DAS 8085 BUCH liegt.

DIE 8085/8086 INTERFACES

ca. 600 Seiten, DIN A4, Paperback, DM 89,--

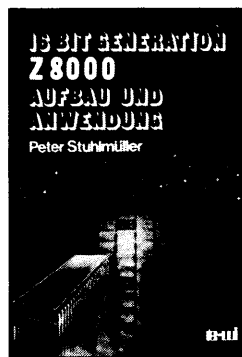
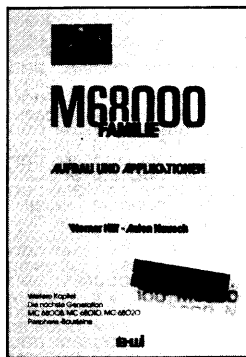
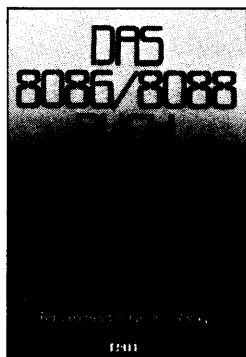
In diesem Werk werden Funktion und Anwendung der Peripheriebausteine zu den Mikroprozessoren 8085 und 8086 beschrieben. Nur mit genauer Kenntnis dieser Interfaces lassen sich Mikroprozessorsysteme erfolgreich entwerfen und programmieren. Auch dieses Werk versteht durch seine überwiegend bildhaften Darstellungen, selbst komplexe Systemzusammenhänge beim Zusammenwirken von Prozessor und Interfacebausteinen leichtverständlich zu erläutern.

Die angegebenen Preise sind die Ladenpreise.

te-wi Verlag GmbH
Telefon 089/1292090

te-wi

Theo-Prosel-Weg 1
8000 München 40



Mikrocomputer der 16-Bit-Generation

Mit diesen drei Titeln über klassische Vertreter der 16-Bit-Mikrocomputer erhält der Leser äußerst umfangreiche Informationen über die komplexen Bausteine. Die Bücher befassen sich grundlegend mit der Architektur sowie der Software und Peripherie zu diesen CPUs.

Bei dem Band "M68000-Familie" wird sogar auch ein echter 32-Bit-Mikrocomputer behandelt.

Die Autoren – alles bekannte Insider der Mikroelektronik – vermitteln ihren Lesestoff in didaktisch hervorragender Weise. Die Bücher sind randvoll mit Funktionsbeschreibungen, Programmierbeispielen und Diagrammen.

Das 8086/8088-Buch, von R. Rector und G. Alexy,
560 Seiten, Paperback, Preis DM 69,—

M68000-Familie, von Werner Hilf und Anton Nausch,
ca. 500 Seiten, Paperback, Preis DM 69,—
(Erscheinung 4.Q83 / 1.Q.84)

Z 8000 – Aufbau und Anwendung, von Peter Stuhlmüller,
464 Seiten, Hardcover, farbiger Schutzumschlag, Preis DM 69,—

Die angegebenen Preise sind die Ladenpreise



CBM COMPUTER HANDBUCH

Dieses Handbuch hilft Ihnen, Ihren CBM-Computer erst richtig zu verstehen! Beim Studium werden keine Vorkenntnisse erwartet. Es ist randvoll mit kurzen Programmbeispielen. Auch die Peripherie wird eingehend behandelt.

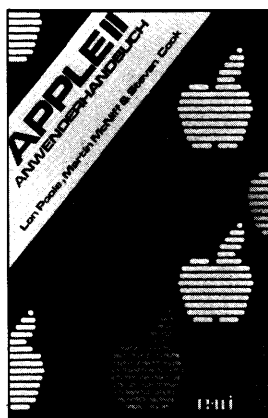
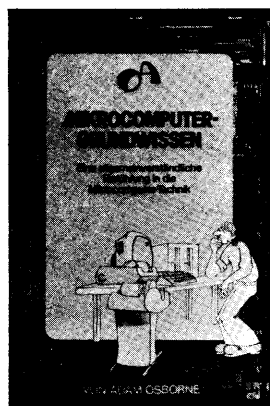
Von Adam Osborne und Carroll S. Donahue
544 Seiten, DM 56.—

MIKROCOMPUTER-GRUNDWISSEN

Eine allgemeinverständliche
Einführung in die
Mikrocomputer-Technik —
vom Mikrocomputer-Papst

Dr. Adam Osborne. Optimal als Einstieg für Elektronik-Laien;
zur Kontaktaufnahme mit diesem die Technik
und unsere Umwelt revolutionierenden Gebiet.

Von Adam Osborne
304 Seiten, DM 36,—



APPLE II PASCAL — EINE PRAKTISCHE EINFÜHRUNG

Dieses Buch ist unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen. Vorkenntnisse werden nicht erwartet. An Hand von Beispielen und Übungen lernen Sie, selbst Programme in PASCAL zu entwickeln und auszutesten.

Von Arthur Luehrmann und Herbert Peckham
544 Seiten, DM 56.—



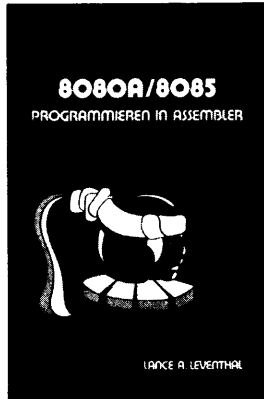
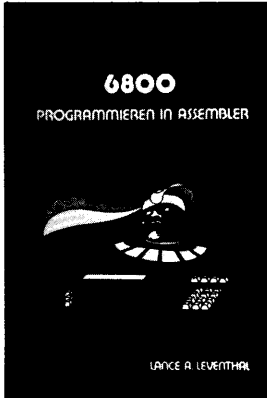
APPLE II ANWENDERHANDBUCH

Der richtige Leitfaden für Ihren APPLE II! Mit Hilfe dieses Buches werden Sie Ihren APPLE-Computer erfolgreich einsetzen. Es beschreibt den Computer als solchen und gibt ausführlich Auskunft über die gesamte Peripherie. Die deutsche Übersetzung entstand in enger Zusammenarbeit mit der Herstellerfirma.

Von Lon Poole und Martin McNiff
400 Seiten, DM 49.—

PROGRAMMIEREN IN ASSEMBLER

In dieser Buch-Serie wird die Programmierung der Mikroprozessoren 6800, 8080A/8085 und 6502 in Assemblersprache beschrieben. Die Titel enthalten eine große Auswahl von praktischen Programmen in Standardformat einschließlich Flußdiagrammen, Quellprogrammen, Objektcodes und erläuternden Texten. Jeder Befehl der entsprechenden CPU wird detailliert erklärt.



Von Lance A. Leventhal
600 bzw. 512 Seiten, DM 59.—
("6502") bzw. DM 49.—

MEIN ATARI-COMPUTER

Dieses Buch macht die Möglichkeiten, die in Ihrem ATARI®-Computer stecken, auf leicht verständliche Art transparent. In einfachen Schritten wird der Anwender mit der Bedienung der Geräte und der Software vertraut gemacht. Ein in sich abgeschlossener ATARI-BASIC-Kurs führt schnell in diese Programmiersprache ein. Der Umgang mit Farbe, Grafik und Ton wird Ihnen anhand von Text und Programmen vermittelt.

Von Lon Poole, Martin McNiff
& Steven Cook
500 Seiten, DM 56,—



77 BASIC-PROGRAMME

Eine Sammlung von 77 praktischen Kurzprogrammen in BASIC, die mathematische, finanztechnische, statistische und verschiedene allgemeine Aufgaben behandeln. Die ausführlich erläuterten Befehle lassen sich leicht direkt anwenden oder sie dienen als Übungen.

Von Lon Poole und Mary Borchers
208 Seiten, DM 39.—

Die angegebenen Preise sind die Ladenpreise.

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40
Telefon 089/1292090

te-wi

LEXIKON DER MODERNEN ELEKTRONIK

von Markt & Technik

Lexikon der modernen Elektronik

Das ideale Nachschlagewerk
für Beruf, Ausbildung und Hobby

Markt & Technik

Mehr als 2.000 aktuelle Fachbegriffe aus den Gebieten Allgemeine Elektronik, Mikroelektronik, Mikrocomputer-Technik und Software. Zusammengetragen von Profis in diesen Sparten. Suchbegriff ist jeweils der englische Ausdruck, dem die deutsche Übersetzung und eine ausführliche Erläuterung folgt. Zahlreiche Abbildungen und eine Zusammenstellung der Begriffe runden den hohen Informationswert dieses bewährten Nachschlagewerkes ab. Bereits über 10.000 verkaufte Bücher.

Der Elektroniker im Beruf, als Student in der Ausbildung oder der Computerhobbyist findet in diesem Lexikon "seinen" Sprach-

schatz an Fachbegriffen, den er beherrschen muß, oder zumindest griffbereit haben sollte.

Es enthält alles, was die Voraussetzung schafft, um verbal mit der rapiden Entwicklung schritthalten zu können.

232 Seiten
Hardcover
4-farbiger Efalin-
einband
Preis DM 48,-*

DAS UNENTBEHRLICHE NACHSCHLAGEWERK

mit über 2.000 Fachbegriffen aus dem Englischen übersetzt
und ausführlich erklärt

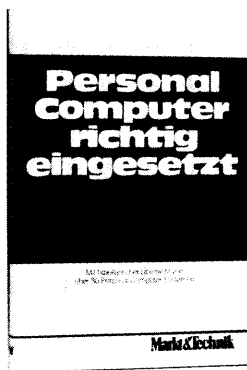
Markt & Technik

Hans-Pinsel-Straße 2

8013 Haar bei München

*Preise inkl. 7% MwSt., zuzüglich Versandkosten

Die »unentbehrlichen 3«



Bestellnummer MT 300 DM 29,—*



Bestellnummer MT 340 DM 29,—*



Bestellnummer MT 350 DM 29,—*

In »Personal Computer richtig eingesetzt« beschreiben Anwender ausführlich und leicht verständlich in 40 Einzelbeiträgen, welche Erfahrungen sie mit Personal Computer-Systemen gemacht haben. Die Beiträge wurden so ausgewählt, daß möglichst viele verschiedene Personal Computer berücksichtigt werden und die Anwendungen aus den unterschiedlichsten Bereichen kommen. Die Beiträge umfassen 16 kaufmännische (Kalkulation, Fakturierung, Angebotserstellung, Textverarbeitung, Buchhaltung, Provisionsabrechnung u.a.) und 24 technisch-wissenschaftliche Anwendungen (NC-Programmierung, CAD-Anwendung, Brand-/Diebstahlschutz, Schulung, Temperaturregelung, psychologische und physiologische Messungen, grafische Darstellung, Faserlängenmessung, Meßdatenerfassung u.a.). Ergänzt wird das Buch durch eine tabellarische Übersicht von mehr als 80 Personal Computer-Systemen mit ihren wichtigsten Daten inkl. Preisen und ein Lieferantenverzeichnis.

»Software-Auswahl leicht gemacht« gibt eine Übersicht über die in Deutschland angebotenen Software-Pakete, die für berufliche oder betriebliche Anwendungen bestimmt sind.

Personal Computer, ursprünglich für einen Einsatz in den eigenen vier Wänden gedacht, haben sich inzwischen in semi- und vollprofessionellen Bereichen bestens bewährt. Diesem Umstand hat Markt & Technik Rechnung getragen und in sein Fachbuchprogramm drei Bücher aufgenommen, die sowohl dem potentiellen als auch dem versierten Anwender wertvolle Anregungen für neue Problemlösungen vermitteln. Die nachfolgend beschriebenen Bände sind unentbehrlich für alle, die auf der Suche nach dem richtigen Personal Computer-System, seiner Peripherie und der dazugehörigen Software sind.

Markt & Technik

Hans-Pinsel-Str. 2
8013 Haar b. München
Telefon 46 13-220

*inkl. MwSt. zuzügl. Versandkosten

Das Buch enthält 800 Programmbeschreibungen und gliedert sich in die Hauptgebiete: branchenneutrale Programme für Rechnungswesen und Verwaltung, Branchenpakete für Rechnungswesen und Verwaltung, Programme für Technik und Wissenschaft und Systemsoftware. Bei jeder Programmbeschreibung befinden sich außerdem Angaben über die Hardware, den Preis, die Zahlungsweise, den Autor und die Bezugsquelle. Mit Hilfe eines Hardware- bzw. Betriebssystemregisters kann der Leser für bestimmte Rechner oder Betriebssysteme die dazu geeigneten Programme auffindig machen. Weitere Hilfestellung geben ein alphabetisches Programmverzeichnis und ein Lieferantenverzeichnis.

»Hardware-Auswahl leicht gemacht« schließt den Reigen dieser Buchreihe und gibt dem Leser erschöpfend Auskunft über Personal Computer und ihre Peripherie. Aktuelle Beiträge zu den Themen Personal Computer, Drucker, Platten-/Floppylaufwerke, Kassettenslaufwerke und Plotter werden durch Marktübersichten über die derzeit auf dem Markt befindlichen Systeme ergänzt. Auch hier erleichtert ein Bezugsquellenverzeichnis die Qual der Wahl.



UNIX-Anwenderhandbuch

von Rebecca Thomas und Jean Yates

UNIX hat sich bereits schon in großem Maße so bewährt, daß die Fachwelt darin das Betriebssystem der Zukunft sieht. Dieses Buch ist der richtige Leitfaden dazu. Es vermittelt Computer-Neulingen und Interessierten alle Kenntnisse zum erfolgreichen Arbeiten mit diesem Betriebssystem. Das Werk ist so praxisnah erstellt, daß der Leser bereits nach kurzer Zeit die Arbeit mit seinem Computer aufnehmen kann. Aber auch ohne eine Anlage wird durch zahlreiche Bildschirmbeispiele ein Wissen vermittelt, das Einblick in die Arbeitsweise von UNIX verschafft. Ein eigenes Kapitel befaßt sich mit der wichtigen Rolle, die UNIX als zentrales System eines automatisierten Büros der Zukunft spielen kann. Der richtige Einstieg in das Betriebssystem UNIX.

478 Seiten , Paperback, Preis DM 79,—

Der Preis ist der Ladenpreis.

te-wi Verlag GmbH
Telefon 089/1292090

te-wi

Theo-Prosel-Weg 1
8000 München 40

Computer 1

Persönlich

CP/M und WORDSTAR **Anwender-Handbuch**

Markt & Technik

ISBN 3-922120-10-5