

# Wirtschafts- informatik

---

## Grundlagen

Goldammer/Huhn/Picht

Die Wirtschaft



---

# **Wirtschaftsinformatik – Grundlagen**

---

**Gerd Goldammer  
Günter Huhn  
Jochen Picht**



**Verlag Die Wirtschaft Berlin**

---

**Autoren:**

Prof. Dr. sc. oec. Gerd Goldammer:  
Kapitel 1 (ohne 1. 3. 2.) bis 4  
und Gesamtreaktion  
Prof. Dr. sc. oec. Günter Huhn:  
Abschnitt 1.3.2.  
Prof. Dr. sc. oec. Jochen Picht:  
Kapitel 5

Als Lehrbuch für die Ausbildung an Universitäten und Hochschulen der DDR anerkannt.

Berlin, Oktober 1988  
Minister für Hoch- und Fachschulwesen

Goldammer, Gerd:  
Wirtschaftsinformatik – Grundlagen /  
Gerd Goldammer; Günter Huhn; Jochen  
Picht. – 1. Aufl. – Berlin: Verl. Die Wirtschaft, 1989. – 208 S.: 111 Abb., 29 Tab.,  
10 Fotos. –  
NE: 2. Verf.; 3. Verf.:

*Redaktionsschluß:* 15. 6. 1988

ISBN 3-349-00471-7

© Verlag Die Wirtschaft 1989  
Am Friedrichshain 22, Berlin, 1055  
Lizenz-Nr. 122; Druckgenehmigungs-  
Nr. 195/115/89  
LSV 0394  
Einbandgestaltung: Hermann Misersky  
Typographie: Verlag Die Wirtschaft  
Printed in the German Democratic Republic  
Gesamtherstellung: INTERDRUCK Graphischer  
Großbetrieb Leipzig, Betrieb der ausgezeichneten  
Qualitätsarbeit III/18/97  
Bestellnummer 676 2916

01130

**Es haben zugearbeitet:**

Prof. Dr. habil. Siegfried Apelt  
Dr. oec. Klaus Arnold  
Prof. Dr. sc. oec. Kurt Aust  
Prof. Dr. sc. oec. Hans Gernert  
Prof. Dr. sc. oec. Harald Gläß  
Prof. Dr. sc. oec. Manfred Goepel  
Prof. Dr. sc. oec. Heinz Hartlepp  
Doz. Dr. sc. oec. Dieter Kirsten  
Prof. Dr. sc. oec. Hans-Friedrich Meuche  
Dr. sc. oec. Kurt Porkert  
Doz. Dr. sc. oec. Frank Richter  
Dr. oec. Bernd Vieweger  
Doz. Dr. sc. nat. Thilo Weller

Außerdem haben folgende Mitglieder  
der Arbeitsgruppe Informatik  
des Beirates für Wirtschaftswissenschaften  
beim Ministerium für Hoch- und  
Fachschulwesen mitgewirkt:  
Doz. Dr. sc. oec. Dieter Ehrenberg  
Prof. Dr. sc. techn. Peter Gmilkowsky  
Prof. Dr. sc. oec. Siegfried Köhli  
Doz. Dr. sc. oec. Rudolf Müller  
Prof. Dr. habil. Horst Schneider  
Prof. Dr. sc. oec. Lothar Wagner  
Prof. Dr. sc. pol. Reiner Tschirschwitz

**Lektor:**

Dr. Rolf Scherfenberg





# Inhaltsverzeichnis

<b>Vorwort</b> . . . . .	8	2.2.	
<b>1.</b>		Elemente von Rechnersystemen	37
<b>Einführung</b> . . . . .	9	2.2.1.	
1.1.		Zentraleinheit . . . . .	37
Information	9	2.2.1.1.	
1.2.		Bussystem . . . . .	37
Codierung von Information . . . . .	14	2.2.1.2.	
1.2.1.		Zentralprozessor	38
Interne Bitmuster		2.2.1.3.	
und Zahlensysteme . . . . .	14	Hauptspeicher . . . . .	40
1.2.2.		2.2.1.4.	
Codierung nichtnumerischer		Weitere Bestandteile	42
Information	16	2.2.2	
1.2.3.		Periphere Geräte	
Codierung numerischer		zur Ein- und Ausgabe . . . . .	44
Information	19	2.2.2.1.	
1.3.		Ein- und Ausgabetechniken . . . . .	44
Automatisierte		2.2.2.2.	
Informationsverarbeitung . . . . .	23	Lochstreifen- und Lochkartentechnik	44
1.3.1		2.2.2.3.	
Automatisierte		Bildschirmtechnik . . . . .	46
Informationsverarbeitungssysteme . . . . .	23	2.2.2.4.	
1.3.2.		Drucktechnik . . . . .	48
Ökonomische Gesichtspunkte		2.2.2.5.	
von AIV-Systemen . . . . .	29	Optische Lesetechnik . . . . .	51
Übungen		2.2.2.6.	
		Magnetkartentechnik . . . . .	52
		2.2.2.7.	
<b>2.</b>		Sprachein- und	
<b>Rechnersysteme aus Nutzersicht</b> . . . . .	33	Sprachausgabetechnik	54
2.1.		2.2.3.	
Grundstruktur und Klassen		Externe Speicher	56
von Rechnersystemen . . . . .	33	2.2.3.1.	
		Klassifikation . . . . .	56

2.2.3.2.		4.	
Magnetbandtechnik . . . . .	56	<b>Entwicklung von Anwendersoftware</b>	108
2.2.3.3.		4.1.	
Folienspeichertechnik . . . . .	58	Anwendersoftware . . . . .	108
2.2.3.4.		4.2.	
Plattentechnik . . . . .	61	Entwicklungsphasen und	
2.3.		softwaretechnologische Aspekte . . .	112
Rechnerverbundsysteme . . . . .	63	4.2.1.	
2.3.1.		Phasenmodell . . . . .	112
Grundbegriffe		4.2.2.	
der Online-Kommunikation . . . . .	63	Prinzipien, Mittel und Methoden	
2.3.2.		der Softwareentwicklung . . . . .	119
Rechnernetze . . . . .	70	4.2.3.	
2.3.3.		Nutzermitwirkung und Prototyping . .	127
Telekommunikationsdienste	75	4.3.	
Übungen		Programmiersprachen . . . . .	129
<b>3.</b>		4.3.1.	
<b>Betriebssysteme</b>		Programme und	
<b>und Nutzungstechnologien</b> . . . . .	79	Programmiersprachen . . . . .	129
3.1.		4.3.2.	
Basissoftware . . . . .	79	Konzepte höherer	
3.1.1.		Programmiersprachen . . . . .	133
Klassifikation . . . . .	79	4.3.3.	
3.1.2.		Programmierungsumgebung . . . . .	139
Struktur und Arbeitsweise		Übungen	
von Betriebssystemen . . . . .	82	<b>5.</b>	
3.1.3.		<b>Rechnereinsatz für Standardaufgaben</b>	143
Überblick über ausgewählte		5.1.	
Betriebssysteme . . . . .	87	Einsatzgebiete . . . . .	143
3.2.		5.2.	
Nutzungstechnologien	93	Automatisierte Textverarbeitung . . .	143
3.3.		5.2.1.	
Nutzung des Betriebssystems DCP . .	96	Grundbegriffe . . . . .	143
3.3.1.		5.2.2.	
Dateien . . . . .	96	Kommunikation mit dem	
3.3.2.		Textverarbeitungsprogramm . . . . .	144
Residente Kommandos (Auswahl) . . .	100	5.2.3.	
3.3.3.		Textgestaltung . . . . .	145
Transiente Kommandos und		5.2.4.	
Kommandoverbindungen . . . . .	103	Texteingabe und Textkorrektur . . . .	146
Übungen			

5.2.5. Funktionen zur Wiederverwendung von Texten . . . . .	146	5.4.3. Einführung in die Arbeit mit einem Datenverwaltungssystem . . . . .	158
5.2.6. Weitere Textverarbeitungs- funktionen . . . . .	147	5.5. Integrierte Softwarepakete . . . . .	162
5.2.7. Textverarbeitungssysteme . . . . .	147	Übungen	
5.3. Tabellenkalkulation . . . . .	149	<b>Anhang A</b> Metasprachliche Notation in erweiterter, nichtrekursiver Chomsky-Backus-Naur-Form . . . . .	163
5.3.1. Anwendungsbereich . . . . .	149	<b>Anhang B</b> Kommandoübersicht DCP . . . . .	164
5.3.2. Grundschema der Tabellenkalkulation . . . . .	150	<b>Anhang C</b> Kommandos und Funktionen des TABCALC für das Betriebssystem SCP 1700 . . . . .	173
5.3.3. Leistungsmerkmale von Tabellenkalkulationsprogrammen . . . . .	153	<b>Anhang D</b> Befehlsübersicht REDABAS-3 . . . . .	181
5.4. Datenverwaltung . . . . .	154	<b>Literaturverzeichnis</b> . . . . .	199
5.4.1. Grundbegriffe . . . . .	154	<b>Bildnachweis</b> . . . . .	203
5.4.2. Merkmale und Funktionen eines Datenverwaltungssystems . . . . .	156	<b>Sachwortverzeichnis</b> . . . . .	204

# Vorwort

---

Die automatisierte Informationsverarbeitung hat sich unter dem Einfluß der Mikroelektronik als wirksames Instrument für höheres Leistungswachstum und für die Steigerung der Effektivität in der Volkswirtschaft erwiesen. Computer sind in viele ökonomische Prozesse eingedrungen und haben zu neuen Informations- und Kommunikationstechnologien geführt. Automatisierte Informationsverarbeitungssysteme verändern grundlegend den Inhalt, den Charakter und die Organisation der Arbeit. Diese Entwicklung wird sich mit der weiteren Durchsetzung der Wirtschaftsstrategie der DDR noch beschleunigen.

Als Wissenschaft von der Gestaltung, Entwicklung und Nutzung automatisierter Informationsverarbeitungssysteme in der Wirtschaft hat sich die Wirtschaftsinformatik herausgebildet. Die Wirtschaftsinformatik nutzt Wissen verschiedener Disziplinen, darunter der Ökonomie, Mathematik, Kybernetik, Statistik und Informatik. Die Beziehungen zwischen der Wirtschaftsinformatik und der Informatik sind besonders eng. Aus der Sicht der Wirtschaftsinformatik ist die Informatik ihre profilbestimmende Grundlage. Aus der Sicht der Informatik ist die Wirtschaftsinformatik ein Arbeitsgebiet der angewandten Informatik.

Das Lehrbuch beinhaltet mit Ausnahme der Programmiersprachen die Teile der Informatik, die als Grundlagen der Wirtschafts-

informatik angesehen werden. Der Leser findet alle wichtigen Kategorien, Methoden und Zusammenhänge der Informatik, mit denen er bei der Anwendung der Computertechnik in der Wirtschaft konfrontiert ist. Den Besonderheiten der Wirtschaftsinformatik wird durch die Auswahl des Wissens und durch zahlreiche ökonomische Anwendungsbeispiele Rechnung getragen.

Das Lehrbuch dient der Grundlagenausbildung sowie in einigen Teilen der Vertiefungsausbildung auf dem Gebiet der Informatik in den Fachrichtungen der Grundstudienrichtung Wirtschaftswissenschaften an den Universitäten und Hochschulen der DDR. Es eignet sich auch für die selbständige Aneignung von Grundwissen der Informatik, als Nachschlagewerk beim Umgang mit Begriffen und durch Tabellen im Anhang als Arbeitsmittel für die praktische Tätigkeit am Computer.

An der Entwicklung des Lehrbuches haben alle Hochschullehrer der Arbeitsgruppe Informatik des Beirates für Wirtschaftswissenschaften beim Ministerium für Hoch- und Fachschulwesen mitgewirkt. Die Arbeiten wurden außerdem durch Herrn Prof. Dr. sc. E. Menzel und Herrn Prof. Dr. sc. P.-D. Kluge unterstützt. Allen Beteiligten wird auf diesem Wege herzlich gedankt.

Gerd Goldammer

# 1. Einführung

---

## 1.1. Information

Die Informatik als Grundlage der Wirtschaftsinformatik ist die Wissenschaft von der automatisierten Informationsverarbeitung.<sup>1</sup> Sie beschäftigt sich mit den Gesetzmäßigkeiten und Prinzipien der Verarbeitung, Speicherung und Übertragung von Information. Der Begriff Information wird von der Philosophie und einer Reihe von Einzelwissenschaften, wie Biologie, Mathematik, und natürlich auch von der Informatik verwendet. Jede dieser Wissenschaften hat eine spezielle Betrachtungsweise. In der Informatik wird der Informationsbegriff zunehmend über den Begriff Nachricht abgeleitet.

### Nachricht

**Die Nachricht ist eine zur Weitergabe bestimmte Folge von Zeichen mit Bedeutung für einen Empfänger.**

Die Zeichen der Nachricht sind Elemente eines diskreten Zeichenvorrates (Buchstaben, Ziffern, Sonderzeichen) oder bilden eine kontinuierliche Funktion.

Eine Folge von Zeichen ist zum Beispiel: »Das produzierte Nationaleinkommen der DDR betrug 1986 252 210 Millionen Mark.« Eine kontinuierliche Funktion könnte das Erönen der Akustikeinrichtung eines Computers sein.

Nachrichten sind an physikalische Träger gebunden. Sie werden auf Papier geschrie-

ben, gesprochen oder als Frequenz technisch moduliert. Unter diesem physikalischen Aspekt heißen sie *Signal*.

Nachrichten sind zur Weitergabe von einem Sender zu einem Empfänger bestimmt. Sie setzen voraus, daß zwischen Sender und Empfänger Vereinbarungen vorhanden sind über

- die Struktur der Zeichenfolgen,
- die Zuordnung zwischen einer Zeichenfolge und der Erscheinung, die die Zeichenfolge bezeichnet.

In der oben genannten Zeichenfolge ist die deutsche Grammatik die strukturelle Vereinbarung. Die Wirtschaftswissenschaften definieren die Zuordnung zwischen Nachricht und ökonomischer Realität. Für das Erönen der Akustikeinrichtung eines Computers sind Frequenz und Anzahl der Wiederholungen strukturelle Merkmale. Sie werden vom Gerätehersteller im allgemeinen dem Auftreten eines Fehlers zugeordnet.

### Information

Nachricht wird zur Information, wenn beim Empfänger Unbestimmtheit beseitigt wird. Die Unbestimmtheit besteht hinsichtlich konkreter Ausprägungen allgemeiner Erscheinungen und Zusammenhänge.

Die Unbestimmtheit beim produzierten Na-

<sup>1</sup> Belke, W.; Tschoppe, H.: Zur Herausbildung und Bestimmung der Informatik als Wissenschaftsdisziplin im System der Wissenschaften – Thesen. In: GI-Mitteilungen 4/1987, S. 158.

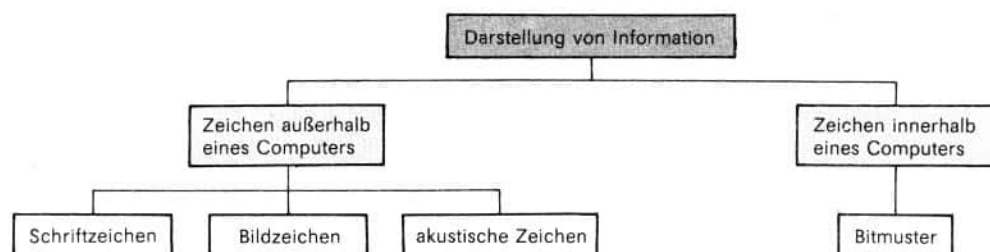


Abbildung 1.1.

Zeichen zur Informationsdarstellung außerhalb und innerhalb des Computers

tionaleinkommen besteht darin, daß es unterschiedliche Umfänge besitzen kann. Der konkrete Wert beseitigt die Unbestimmtheit.

Bei der Arbeit am Computer kann ein Fehler auftreten – oder nicht. Das Akustiksignal beseitigt diese Unbestimmtheit.

Nicht immer wird durch Information die Unbestimmtheit vollständig beseitigt, in jedem Fall aber verringert.

**Information ist eine Nachricht, die Unbestimmtheit verringert.**

In der Informationstheorie wird die Informationsmenge  $I$  einer Nachricht, der Informationsgehalt, als Differenz zwischen der Unbestimmtheit vor ( $V$ ) und nach ( $N$ ) dem Erhalt einer Nachricht gemessen.<sup>2</sup> Allgemein gilt:

$$I_{\text{Nachricht}} = V - N$$

Grundsätzlich besteht die Aufgabe, das Verhältnis zwischen Länge der Nachricht (Anzahl der zur Darstellung benutzten Zeichen) und Informationsmenge optimal zu gestalten, d. h., mit möglichst kurzen Nachrichten ist viel Information zu übermitteln. Der Teil einer Nachricht, der gekürzt werden kann, ohne daß Information verloren geht, wird als *Redundanz* bezeichnet. Es gibt fördernde und leere Redundanz. Leere Redundanz ist zu vermeiden.

Wegen dieser Zusammenhänge ist es nicht treffend, die Informationsmenge als Anzahl

von Informationen (z. B. zwei Informationen) anzugeben. Der Gebrauch des Wortes Information im Plural, also »Informationen«, führt von vornherein zur Gleichsetzung von Nachricht und Information. Deshalb ist der Gebrauch des Wortes Information im Plural nicht eindeutig.

Für die Informatik ist neben den Begriffen Nachricht und Information der Begriff Daten von Bedeutung.

**Daten sind Nachrichten, die so aufbereitet sind, daß sie durch einen Computer identifiziert, verarbeitet, gespeichert und übertragen werden können.**

Bei der Anwendung der Informatik werden die Begriffe Nachricht, Information und Daten nicht immer streng unterschieden.

Beim Einsatz eines Computers wird Information in den Computer gebracht, dort gegebenenfalls unter Verwendung bereits gespeicherter Information verarbeitet, und das Ergebnis wird nach außen bereitgestellt.

Außerhalb des Computers erfolgt die Darstellung der Information mit Schriftzeichen, Bildzeichen oder akustischen Zeichen. Innerhalb des Computers können zur Darstellung von Information auf einer technischen Speichereinheit, dem Bit (bi-

<sup>2</sup> Jaglom, A. M.; Jaglom, I. M.: Wahrscheinlichkeit und Information. Berlin: VEB Deutscher Verlag der Wissenschaften 1967, S. 85.

nary digit), aber nur zwei Zustände verwendet werden:

- Spannungspegel niedrig  
(Beschreibung für nicht initialisiert mit »0« oder »L« (low));
- Spannungspegel hoch  
(Beschreibung für initialisiert mit »1« oder »H« (high)).

Dadurch müssen zur Darstellung eines (externen) Schrift-, Bild- oder akustischen Zeichens intern mehrere Bit verwendet werden. Die symbolische Darstellung einer Folge von Bit wird als *Bitmuster* bezeichnet. Die Abbildung 1.1. gibt einen Überblick über die Informationsdarstellung außerhalb und innerhalb des Computers.

*Schriftzeichen* sind Buchstaben, Ziffern oder Sonderzeichen, wie ;, \*, +, &, {}, (), []. Sie werden im allgemeinen über alphanumerische Tastaturen in den Computer gebracht und über Bildschirm und Drucker ausgegeben.

*Bildzeichen* sind vor allem Grafiken. Grafische Grundsymbole, wie Bildpunkte und Linien, werden mit speziellen Geräten, z. B. Tablett mit elektronischem Stift, Rollkugel oder Maus (vgl. Abschnitt 2.2.2.), in den Computer eingegeben. Ausgabegeräte sind grafikfähige Bildschirme, grafikfähige Drucker und spezielle Zeichengeräte (Plotter).

Zu den Bildzeichen, die für Anwendungen in der Wirtschaft von besonderer Bedeutung sind, gehören Balkencodiendiagramme (vgl. Abbildung 1.2.). Sie werden auf Verpackungen von Waren gedruckt oder zur Regalauszeichnung in der Lagerwirtschaft benutzt. Der Balkencode wird mit speziellen Geräten (Lesepistole, Scanner) optisch gelesen und dient der rationellen Identifikation von Artikeln (vgl. Abschnitt 2.2.2.).

Zunehmende Bedeutung erlangen auch *akustische Zeichen* für die Spracheingabe.



Abbildung 1.2.  
Balkencode

Die Spracheingabe ist besonders dort von Bedeutung, wo die Hände für eine Tastatureingabe während der Computernutzung nicht oder nur eingeschränkt zur Verfügung stehen. Das ist z. B. in der Lagerwirtschaft und im Transportwesen der Fall. Heute existieren zuverlässige Spracheingabesysteme mit einem Wörternvorrat von bis zu einigen hundert Wörtern (vgl. Abschnitt 2.2.2.).

Die Sprachausgabe ist von Bedeutung, wenn keine örtliche Bindung an den Bedienplatz möglich ist. Das ist ebenfalls in der Lagerwirtschaft und im Transportwesen der Fall.

Sprachein- und Sprachausgabe sind von erheblicher Bedeutung für den Anschluß von Computern an ein sprachaktives Kommunikationsnetz (Telefonnetz).

Schriftzeichen, grafische Zeichen und akustische Zeichen müssen bei der Eingabe in den Computer in die elementare computerinterne Form, das Bitmuster, umgewandelt werden. Dieser Vorgang heißt *Codierung*.

**Codierung ist die Umwandlung von Zeichen aus einer (hier computerexternen) Darstellung in eine andere (hier computerinterne) Darstellung.**

Die eindeutige Zuordnung von Zeichen außerhalb des Computers zur computerinternen Darstellung regelt ein *Code*. Codes sind weitgehend standardisiert.

Tabelle 1.1.  
Bitgruppen

Bezeichnung	Anzahl Bit	Anzahl Byte
Halbbyte		
(Tetrade, Nibble)	4	—
Byte	8	1
Wort	16	2
Doppelwort	32	4
Paragraph (Abschnitt)	128	16

**Der Code ist eine eindeutige Vorschrift für die Zuordnung von Zeichen eines externen Zeichenvorrats (Zeichensatz, Alphabet) zu internen Zeichen.**

Für die computerinterne Darstellung von Information reicht ein Bit nicht aus. Bei einem Zeichenvorrat von 2 ermöglicht 1 Zeichen die Darstellung von genau  $2^1 = 2$  verschiedenen Zuständen. Die 52 Groß- und Kleinbuchstaben des lateinischen Alphabets, die Ziffern und die Sonderzeichen ließen sich also nicht darstellen. Deshalb werden zur Darstellung externer Zeichen intern Bitmuster der Länge 8 benutzt. Damit können  $2^8 = 256$  verschiedene externe Zeichen codiert werden. Die Gruppierung zu 8 Bit hat auch gerätetechnische Gründe. Im Speicher eines Computers kann ein einzelnes Bit nicht gesondert gelesen oder geschrieben werden. Die kleinste (adressierbare) Einheit sind 8 Bit. Eine Gruppe von acht zusammengehörenden Bit bildet ein Byte (englisches Kunstwort). Eine Übersicht über diese und weitere wichtige Bitgruppen enthält Tabelle 1.1.

Für die Verarbeitung der Information innerhalb des Computers oder zur Verarbeitung durch einen anderen Computer kann es erforderlich sein, die computerinterne Darstellungsform noch einmal zu verändern. Dieser Vorgang heißt *Konvertierung*.

**Konvertierung ist die Transformation von Information aus einer computerinternen Darstellungsform in eine andere computerinterne Darstellungsform.**

Konvertierung geschieht z. B. für numerische Daten. Die Eingabe von Zahlen erfolgt zunächst ziffernweise. Für arithmetische Operationen ist diese Darstellung jedoch ungünstig. Deshalb erfolgt für arithmetische Operationen eine erneute Umwandlung, die Konvertierung (vgl. Abschnitt 1.3.2.).

### Kommunikation

Information wird von einem System zu einem anderen System übertragen. Die Richtung der Übertragung kann wechseln. Dieser Vorgang heißt Kommunikation.

**Kommunikation ist Austausch von Information zwischen Systemen.**

Kommunikation kann zwischen Menschen, zwischen Mensch und Computer (Mensch-Maschine-Kommunikation), zwischen Computern (Maschine-Maschine-Kommunikation) und auch zwischen Programmen erfolgen. Untersuchungsgegenstand der Informatik sind vor allem die drei letztgenannten Formen.

Der Weg der Information von der Entstehung bis zur Verwertung ist für alle Formen gleich und heißt *Kommunikationskette* (vgl. Abbildung 1.3.).

*Quelle* der Information sind Prozesse in der Natur, Technik, Wissenschaft und Gesellschaft. Der Ablauf dieser Prozesse verringert Unbestimmtheit und erzeugt dadurch Information. Die Information wird durch einen *Sender* zu einer Nachricht formiert. Das geschieht, um ihre Weitergabe in Raum und Zeit zu ermöglichen. Die physikalische Form der Nachricht ist dabei abhängig vom *Kanal*, über den die Nachricht weitergegeben werden soll. Der Kanal ist das Trägermedium, in dem die physikali-



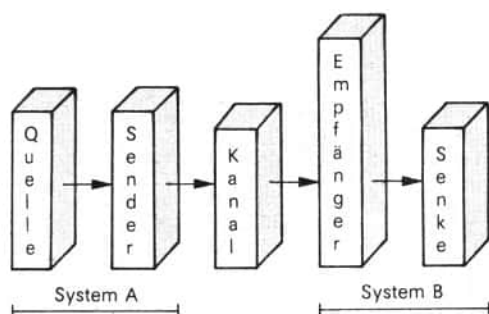


Abbildung 1.3.  
Kommunikationskette

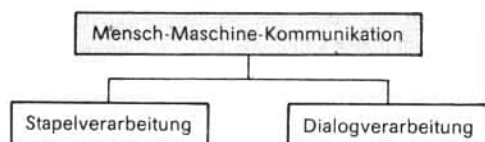


Abbildung 1.4.  
Mensch-Maschine-Kommunikation  
und Nutzungstechnologien

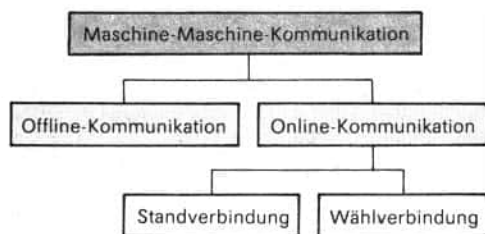


Abbildung 1.5.  
Klassifikation der Maschine-Maschine-Kommunikation

schen Signale für den Transport sorgen. Die Nachricht gelangt über den Kanal zu einem *Empfänger*. Dieser ordnet der Nachricht unter Berücksichtigung der Syntax Bedeutung zu. In der *Senke* erfolgt die Verwertung der Information.

Bei der *Mensch-Maschine-Kommunikation* ist zunächst der Mensch System A. Er übermittelt der Maschine, dem System B, Information in Form von Programm- und Verarbeitungsdaten. Die Maschine verwer-

tet die erhaltene Information unter Verwendung eventuell bereits gespeicherter Information und erzeugt dadurch neue Information. Jetzt wird die Maschine zu System A und sendet die Information an den Menschen. Je nachdem, ob der Zyklus Mensch-Maschine-Mensch für eine Aufgabe einmal oder mehrmals abläuft, werden verschiedene Nutzungstechnologien der Maschine unterschieden (vgl. Abbildung 1.4.).

Bei der *Stapelverarbeitung* wird dem Computer die gesamte Information für eine Aufgabe übergeben, von diesem verarbeitet, und das Ergebnis wird an den Menschen zurückgegeben. Der Mensch greift im allgemeinen nicht in den Verarbeitungsprozeß ein.

Bei der *Dialogverarbeitung* unterbricht der Computer entsprechend dem (dialogorientierten) Programm die Ausführung der Aufgabe, informiert über Teilergebnisse und fordert in Abhängigkeit von der jeweiligen Situation weitere Information an. Der Zyklus Mensch-Maschine-Mensch wiederholt sich, bis die Aufgabe ausgeführt ist. Stapel- und Dialogverarbeitung sind aus der Sicht des Nutzers verschiedene Nutzungstechnologien und werden im Abschnitt 3.2. dargestellt.

Die *Maschine-Maschine-Kommunikation* dient der wechselseitigen Übermittlung von Information zwischen zwei Computern. Es wird zwischen Offline- und Online-Kommunikation unterschieden (vgl. Abbildung 1.5.).

Offline-Kommunikation erfolgt durch Datenträgeraustausch zwischen den Computern. Die Datenträger, wie Magnetband und Magnetplatte, werden durch einen Computer mit Daten beschrieben und dann über eine räumliche Entfernung zu dem anderen Computer transportiert.

Bei der Online-Kommunikation werden nachrichtentechnische Übertragungswege

benutzt, die beide Computer ständig oder vorübergehend verbinden. Standverbindungen sind fest durchgeschaltete Übertragungswege mit ständiger Dienstbereitschaft und hoher Übertragungsrate. Wahlverbindungen werden auf Anforderung durch Vermittlungseinrichtungen hergestellt und nach Beendigung der Kommunikation für andere Nutzer wieder freigegeben.

Bei der Maschine-Maschine-Kommunikation können mehrere Computer miteinander verbunden sein. Im allgemeinen kann in einem solchen *Rechnernetz* jeder Computer mit jedem anderen kommunizieren. Ein Rechnernetz kann sich über kleinere räumliche Distanzen erstrecken, wie z. B. über eine Werkhalle oder ein Gebäude (lokales Netz). Es kann sich jedoch auch über größere Entfernungen ausdehnen (Fernnetz) und selbst Spezialrechner nutzen (vgl. Abschnitt 2.3.).

Die *Kommunikation zwischen Programmen* erfolgt innerhalb eines Computers und wird durch das Betriebssystem oder durch die Programme selbst gesteuert. Sie ist im Zusammenhang mit der Programmiersprache darzustellen<sup>3</sup> und wird deshalb in diesem Buch nicht behandelt.

Die verschiedenen Kommunikationsformen können durch einen Computer gleichzeitig realisiert werden. So kann im Hintergrund einer Mensch-Maschine-Kommunikation eine Maschine-Maschine-Kommunikation stattfinden, und zugleich können verschiedene Programme Information austauschen.

Die Kommunikation beinhaltet technische, psychologische, arbeitswissenschaftliche, organisatorische und ökonomische Aspekte. Sie erfordert die Bereitstellung und den Einsatz von entsprechenden geräte- und programmtechnischen Mitteln. Organisatorisch entstehen ständig qualitativ neue Anwendungen mit erheblich veränderten ökonomischen Parametern.

## 1.2.

### Codierung von Information

#### 1.2.1.

#### Interne Bitmuster und Zahlensysteme

Information wird innerhalb des Computers durch physikalische Zustände dargestellt, die durch Bitmuster beschrieben werden. Ein solches Bitmuster enthält Abbildung 1.6.

Das Muster in Abbildung 1.6. wurde byteweise abgesetzt. Trotzdem ist es schwierig, die Muster der Bytes zu vergleichen und festzustellen, daß zwei Muster gleich sind (Byte 3 und Byte 6). Um eine übersichtlichere Schreibweise zu erreichen, kann der Inhalt eines Bytes als eine Zahl aus einem Zahlensystem mit nur zwei Ziffern, nämlich 0 und 1, aufgefaßt werden. Eine solche Zahl, die von der gewohnten Dezimalzahl mit einem Ziffernvorrat von 0 bis 9 abweicht, heißt *Binärzahl*. Zur Bestimmung des dezimalen Wertes einer Binärzahl ist die allgemeine Zahlenformel zu verwenden:

$$\text{Zahlenwert} = \underline{z}^T * \underline{s}$$

Der Zeilenvektor  $\underline{z}^T$  ist der Vektor der Ziffernwerte. Am Beispiel des ersten Bytes aus Abbildung 1.6. ist:

$$\underline{z}^T = [0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0]$$

Der Spaltenvektor  $\underline{s}$  ist der Vektor der Stellenwerte mit den Komponenten  $B^i$  für  $i = 0$  (1)  $n$ .  $B$  bezeichnet die Basis des Zahlensystems, also 10 für das Dezimalsystem und 2 für das Binärsystem.  $B$  ist zugleich die Anzahl verschiedener Zeichen im Zeichenvorrat des jeweiligen Systems.  $i$  bewegt sich in den Grenzen 0 bis  $n$ , und der Vektor hat so viele Komponenten, wie die Zahl Ziffern hat.

<sup>3</sup> Vgl. Goldammer, G.: PASCAL für die Anwendung in der Wirtschaft. Berlin: Verlag Die Wirtschaft 1987, S. 91.

01001100	01100101	01101001	01110000	01111010	01101001	01100111
----------	----------	----------	----------	----------	----------	----------

Abbildung 1.6.  
Bitmuster

76	101	105	112	122	105	103
01001100	01100101	01101001	01110000	01111010	01101001	01100111

Abbildung 1.7.  
Bitmuster mit dezimaler Interpretation

\$4C	\$65	\$69	\$70	\$7A	\$69	\$67
01001100	01100101	01101001	01110000	01111010	01101001	01100111

Abbildung 1.8.  
Bitmuster mit hexadezimaler Interpretation

$$\underline{s} = \begin{bmatrix} B^n \\ \vdots \\ B^i \\ \vdots \\ B^1 \\ B^0 \end{bmatrix}$$

$i$  ist die Ziffernposition und  $B^i$  der Stellenwert, mit dem eine an der Stelle  $i$  stehende Ziffer zu multiplizieren ist, um ihren Zahlenwert zu bestimmen. Der Stellenwertvektor für alle Bitmuster in der Breite eines Bytes ist:

$$\underline{s} = \begin{bmatrix} 2^7 \\ 2^6 \\ 2^5 \\ 2^4 \\ 2^3 \\ 2^2 \\ 2^1 \\ 2^0 \end{bmatrix}$$

Der Zahlenwert  $W$  für das erste Byte aus Abbildung 1.6. ist:

$$\begin{aligned} W = \underline{z}^T \cdot \underline{s} &= 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 \\ &\quad + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 \\ &\quad + 0 \cdot 2^0 = 76 \end{aligned}$$

Bei dezimaler Interpretation der Bitmuster aus Abbildung 1.6. ergibt sich die Darstellung in Abbildung 1.7.

Die Schreibweise ist jetzt übersichtlicher. Allerdings ist die Zuordnung zwischen Binärzahl und Dezimalzahl nicht offensichtlich. So beginnen alle Binärzahlen mit »01«, die Dezimalzahlen beginnen aber mit »7« oder »1«. Auch die Anzahl der Ziffern ist bei der Binärzahl immer gleich, bei der entsprechenden Dezimalzahl wechselt sie. Es müßte also jedesmal zwischen Binärzahl und Dezimalzahl umgerechnet werden.

Zur einfachen und übersichtlichen Interpretation interner Darstellungen haben sich *Hexadezimalzahlen* durchgesetzt, vor allem bei byteorientierten Rechnern. Die Ziffern des Hexadezimalsystems sind 0 bis 9 und A bis F. Die Basis des Hexadezimalsystems ist 16. Wegen  $16 = 2^4$  korrespondiert eine hexadezimale Ziffer zu jeweils einem Halbbyte (vier Bit) eines Bitmusters. Die Zuordnung zeigt Tabelle 1.2.

Bei hexadezimaler Interpretation des Bitmusters aus Abbildung 1.6. ergibt sich die Darstellung in Abbildung 1.8. Um kenntlich

Tabelle 1.2.

Bitmuster (Halbbyte) mit hexadezimaler und dezimaler Interpretation

Bitmuster	Interpretation		Bitmuster	Interpretation	
	hexadezimal	dezimal		hexadezimal	dezimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Tabelle 1.3.

Wirkung von Steuerzeichen

Steuerzeichen	Wirkung auf den Bildschirm	Wirkung auf den Drucker
BEL (bell)	ohne	Akustiksignal
BS (backspace)	Cursor zurück	Druckkopf zurück
FF (form feed)	Löschen, Cursor auf Position 1,1	Blattvorschub
CR (carriage return)	Cursor zum linken Rand	Druckkopf zum linken Rand

zu machen, daß es sich um hexadezimale Zahlen handelt, wird ein »\$« vorangestellt.

Die Umrechnung der hexadezimalen in die dezimale Darstellung erfolgt mit Tabellen.<sup>4</sup>

Die interpretierten Bitmuster sagen nichts über ihre Bedeutung aus. Grundsätzlich gibt es drei Möglichkeiten. Es kann sich handeln um

- nichtnumerische Information,
- numerische Information oder
- Programmcode.

Welche dieser drei Möglichkeiten für ein konkretes Byte zutreffend ist, ergibt sich aus dem Speicherregime des jeweiligen Computers und entsprechenden Festlegungen in den Programmen. Die Darstellung von Programmcode wird im Ab-

schnitt 4.3.1. behandelt. Die Codierung nichtnumerischer und numerischer Information wird in den Abschnitten 1.2.2. und 1.2.3. dargestellt.

### 1.2.2.

#### Codierung nichtnumerischer Information

Ein erheblicher Teil der Information besteht aus Schrift- und Bildzeichen, die keiner arithmetischen Operation unterliegen. Es werden Zeichen als Byte eingefügt, angehängt, gelöscht oder kopiert. Das sind Textoperationen. Die Zweckbestimmungen solcher Information sind die Identifika-

<sup>4</sup> Goldammer, G.: PASCAL für die Anwendung in der Wirtschaft. Berlin: Verlag Die Wirtschaft 1987, S. 163.

tion anderer Information und die Kommunikation mit dem Menschen. Information dieser Art heißt nichtnumerische Information.

**Nichtnumerische Information ist Information, die vorrangig der Ein- und Ausgabe sowie Textverarbeitungsoperationen unterliegt.**

Die interne Darstellung nichtnumerischer Information muß eine alphabetische (lexikalische) Sortierung ermöglichen. Dabei ist eine exakte Zuordnung zur externen Darstellung dieser Zeichen wichtig. Eine solche Zuordnung wird als Code (vgl. Abschnitt 1.1.) bezeichnet.

Kommuniziert ein Computer mit einem anderen Computer oder mit Geräten, die zur Kommunikation bestimmt sind, wie Tastatur, Drucker, Bildschirm, so müssen die Codes gleich sein. Deshalb kommt der Normung eine große Bedeutung zu. International genormte Codes sind DKOI für Anlagen des ESER, KOI-7H0 (ASCII) nach ST RGW 356 und 8-Bit-Code ESER PC für Personalcomputer.

Die Normung erfolgt in Form einer *Codetabelle*, die die externen Zeichen den internen Bitmustern gegenüberstellt.

Dabei werden zugleich Zeichen für die Kommunikation mit Ein- und Ausgabegeräten, die *Steuerzeichen*, genormt. Steuerzeichen sind keine druckbaren Zeichen, wie »A« oder »B«. In der Codetabelle steht z. B. die Abkürzung LF für line feed. Bei der Übertragung des zu LF gehörenden Bitmusters zu einem Drucker wird der Druckkopf eine Zeile tiefer gestellt. Wird das Zeichen LF zum Bildschirm gesendet, wird dort die Schreibmarke (Cursor) eine Zeile nach unten gesetzt. Die Wirkung der Steu-

erzeichen ist abhängig vom empfangenden Gerät und bei programmgesteuerten Geräten auch vom Betriebssystem (vgl. Kapitel 3.). Beispiele für die Wirkung von Steuerzeichen auf Bildschirm und Drucker enthält Tabelle 1.3.

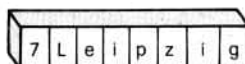
Von besonderer Bedeutung für Personalcomputer ist der Code KOI-7H0, der zugleich eine echte Teilmenge (für alle Zeichen, die mit Bit 7 gleich Null dargestellt werden) des 8-Bit-Code ESER PC ist. Die Codetabelle (HWT – höherwertiger Teil/ linkes Halbbyte, NWT – niederwertiger Teil/rechtes Halbbyte) enthält Tabelle 1.4.

Wird der Code aus Tabelle 1.4. auf das Bitmuster der Abbildung 1.8. angewandt, ergibt sich als nichtnumerische Information:

Leipzig

Im allgemeinen ist es erforderlich, Zeichenfolgen unterschiedlicher Länge im Speicher unterzubringen. In diesem Falle wird das erste Byte dazu benutzt, die Anzahl der Zeichen festzuhalten. Es ergibt sich das unten abgebildete allgemeine Stringformat.

Die interne Darstellung des Textes »Leipzig« ist (symbolisch)



Im Speicher stehen immer nur Bitmuster; für die Zeichen stehen die Bitmuster der Codetabelle, und für die dynamische Länge 7 steht das Muster 00000111. Solche Darstellungen heißen Zeichenketten und innerhalb von Programmiersprachen Strings.

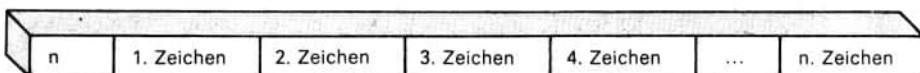


Tabelle 1.4. 7-Bit-Code KOI-7H0(DDR) nach ST RGW 356

HWT DEZ	HEX	NWT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
0	\$0		NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
16	\$1		DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
32	\$2			!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
48	\$3		0	1	2	3	4	5	6	7	8	9		;	<	=	>	?
64	\$4		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	\$5		P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	\$6			a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	\$7		p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Tabelle 1.5. 8-Bit-Code des EC 1834 (erweiterter Teil)

HWT DEZ	HEX	NWT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
128	\$8		Ç	ü	é	â	ä	à	ä	ç	ê	ë	è	ï	î	í	À	Á
144	\$9		É	æ	Æ	ô	ö	ò	û	ü	ý	ÿ	Ö	Ü	Φ	£	¥	℔
160	\$A		á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	¬	½	¼	;	«	»
176	\$B																	
192	\$C		⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂
208	\$D		⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂	⌂
224	\$E		α	β	Γ	π	Σ	σ	μ	τ	Φ	θ	Ω	δ	∞	∅	€	∩
240	\$F		≡	±	≥	≤	∫	∫	÷	≈	°	•	•	•	•	•	•	•

Die möglichen Bitmuster ( $2^8 = 256$ ) sind durch die Codetabelle des KOI-7H0 nicht ausgeschöpft. Die Bezeichnung »7-Bit-Code« bringt das auch zum Ausdruck. Der größte definierte Wert des KOI-7H0 ist \$7F und entspricht dem Bitmuster 01111111, d. h., es werden nur 7 Bit benutzt. 8-Bit-Codes nutzen dagegen die gesamte Brei-

te des Bytes. Dabei wird in der Regel der KOI-7H0, der dem international bevorzugten ASCII-Code entspricht, als echte Teilmenge übernommen und dient der allgemeinen, systemunabhängigen Kommunikation. Gegenüber dem 7-Bit-Code verfügt der 8-Bit-Code auch über die Bitmuster zwischen \$80 und \$FF. Das ist ein Bereich,

der noch einmal so groß ist wie der zwischen \$00 und \$7F. Die Bitmuster dieses Bereiches werden für länderspezifische Codes und für Grafikzeichen (Quasigrafikzeichen) genutzt. Ein Beispiel zeigt die Tabelle 1.5.

Ab \$C0 kann nach ST RGW 358/360 ein deutscher Zeichensatz mit Umlauten definiert sein (8-Bit-Code KOI-7H0). Solche Zusätze werden als alternative Codes bezeichnet.

Die Codierung von Eingabeinformation, d.h. die Umsetzung eines Zeichens aus der externen Darstellung in die interne Darstellung, erfolgt direkt nach der Tastatureingabe. Wird z. B. die Taste mit dem Kleinbuchstaben »a« betätigt, so sendet die Tastatur ein der Taste eindeutig zugeordnetes Bitmuster, das vom Betriebssystem in das Bitmuster 01100001 umgesetzt wird. Steuerzeichen können eingegeben werden, wenn die spezielle Taste <CTRL> und gleichzeitig eine weitere Taste gedrückt werden. In diesem Fall werden die (oberen) Bit 5 bis 7 auf 0 gesetzt. <CTRL> <X> erzeugt z. B. CAN (cancel) und löscht die bisherige Eingabe. Auf einen alternativen Code kann bei vielen Tastaturen mit der Taste <ALT> umgeschaltet werden.

Die Codierung der Ausgabeinformation ist geräteabhängig. Drucker besitzen einen *Zeichengenerator*, der die erhaltenen Bitmuster in Information zur Bewegung des Druckkopfes umformt. Es kann zwischen verschiedenen Zeichensätzen umgeschaltet werden. Für die Bildschirmausgabe ist es ähnlich. Bildschirmsteuereinheiten enthalten im allgemeinen einen festen Zeichengenerator für 128 Zeichen (Bitmuster \$00 bis \$7F, also Bit 7 gleich 0) und einen ladbaren Zeichengenerator für weitere 128 Zeichen (Bitmuster \$80 bis \$FF). Der feste Zeichengenerator realisiert immer den unteren Teil einer 8-Bit-Code-Tabelle, z. B. KOI-7H bzw. ASCII. Dieser Zeichensatz steht auch beim Einschalten eines Ge-

rätes zur Verfügung. Der obere Teil der 8-Bit-Code-Tabelle muß von einer speziellen Speichereinheit oder durch ein Programm geladen werden.

### 1.2.3. Codierung numerischer Information

Die gesamte Eingabeinformation für einen Computer wird zunächst als nichtnumerische Information angesehen und intern entsprechend der Codetabelle als Bitmuster Byte für Byte abgelegt. Das gilt auch für Ziffern. Wird z. B. über eine Tastatur die Ziffernfolge 90 eingegeben, so ist das interne Äquivalent (vgl. Tabelle 1.4.)

00111001 00110000

Diese interne Darstellung benötigt 2 Byte und erweist sich als technisch ungünstig, wenn z. B. die Zahl 90 mit 2 multipliziert, durch 5 dividiert oder einer Addition oder Subtraktion unterzogen werden muß. Ist das der Fall, so geht es intern nicht um eine Folge von Ziffern, sondern um eine Zahl. Es handelt sich um numerische Information.

**Numerische Information ist Information, die vorrangig arithmetischen Operationen unterliegt.**

Arithmetische Operationen werden im Computer auf bitweise logische Operationen zurückgeführt, deren theoretische Grundlagen von George Boole (1815 bis 1869) erstmals formuliert wurden. Inwieweit diese Operationen bereits hardwaremäßig in den Computer eingebaut sind oder programmiert werden müssen, hängt vom Computer ab. In jedem Fall muß numerische Information so dargestellt werden, daß die logischen Operationen zeit- und speicherplatzsparend ausgeführt werden können. Deshalb muß die Darstellung

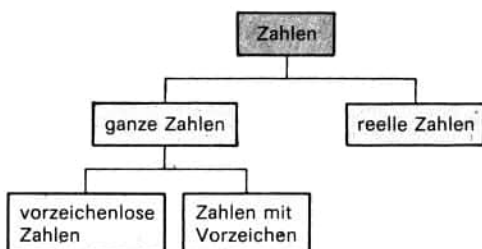


Abbildung 1.9.

Typen numerischer Information

der Eingabeinformation noch einmal verändert werden. Das erfolgt intern. Dieser Vorgang wird als Konvertierung bezeichnet (vgl. Abschnitt 1.1.).

Für die Konvertierung in eine optimale interne Darstellung müssen mehrere Typen numerischer Information unterschieden werden, die vom Wertebereich (also nicht unbedingt von der konkreten Zahl) abhängig sind (vgl. Abbildung 1.9.).

### Vorzeichenlose ganze Zahlen

Vorzeichenlose ganze Zahlen werden als Binärzahl dargestellt. Die allgemeine Zahlenformel (vgl. Abschnitt 1.2.1.) liefert den Zahlenwert, wenn als Basis  $B = 2$  verwendet wird. Die interne Darstellung der Zahl 90 ist in diesem Fall (mit führenden Nullen auf ein volles Byte gefüllt):

01011010

$$(0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 90)$$

Zur Darstellung der numerischen Information »90« ist ein Byte und zur (zeichenweisen) Darstellung der nichtnumerischen Information »90« sind zwei Byte erforderlich. Dem Aufwand der Konvertierung steht hier eine Speicherplatzverringerung gegenüber. Vor allem aber lassen sich arithmetische Operationen ausführen. Für die stellenweise Addition zweier Binärzahlen gilt:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$1 + 1 = 0$  mit Übertrag 1 nach links oder Verlust der 1 (Überlauf).

Ist zu 90 der Wert 3 (mit der binären Darstellung 00000011) zu addieren, so ergibt sich

$$01011010$$

$$+ 00000011$$

$$= 01011101$$

Die Multiplikation wird als mehrfache Addition ausgeführt.

Die kleinste mit einem Byte darstellbare vorzeichenlose ganze Zahl ist 0, die größte:

$$11111111 = 2^8 - 1 = \$FF = 255$$

Dieser Zahlenbereich reicht für viele Anwendungsfälle nicht aus. Deshalb ist die Darstellung von ganzen Zahlen auf 2 oder 4 Byte typisch. Bei einer Länge von zwei Byte ist die kleinste darstellbare vorzeichenlose Zahl 0, die größte:

$$11111111 \ 11111111 = 2^{16} - 1 = \$FFFF = 65535.$$

### Ganze Zahlen mit Vorzeichen

Die Verarbeitung ganzer Zahlen mit Vorzeichen ist computerabhängig. Personalcomputer verwenden das *Zweierkomplement*. Für das Zweierkomplement wird das höchstwertige Bit benutzt. Dadurch schränkt sich der Bereich darstellbarer Zahlen ein. Die auf einem Byte größte darstellbare positive Zahl ist unter diesen Bedingungen

$$01111111 = 127.$$

Für negative Zahlen wird das Einerkomplement durch bitweise Negation ermittelt und danach 1 addiert. Als numerische Information innerhalb des Computers entsteht für -90:



```

01011010
10100101 Komplementation
+ 00000001
10100110 Addition

```

Bei negativen Zahlen ist Bit 7 stets gesetzt. Es wird deshalb auch als Vorzeichenbit bezeichnet. Die kleinste darstellbare Zahl ist binär 10000000 und entspricht der Dezimalzahl  $-128$ .

Der Zahlenbereich der in einem Byte darstellbaren ganzen positiven und negativen Zahlen reicht von  $-128$  bis  $+127$ . Allgemein liegt für ganze Zahlen mit Vorzeichen der Zahlenbereich in den Grenzen

$$-(2^{n-1}) \text{ bis } 2^{n-1} - 1,$$

wobei  $n$  der Anzahl der maximal für die Darstellung zu verwendenden Bitstellen entspricht. Für die gebräuchliche Darstellungsbreite von 16 Bit ergibt sich ein Zahlenbereich von  $-32768$  bis  $+32767$ .

Werden zur Zahlendarstellung mehrere Byte benutzt, so steht aus technischen Gründen (entgegen den gewohnten Zahlensystemen) im Speicher das niederwertige Byte (physisch) vor dem höherwertigen Byte (umgekehrtes Byteformat).

### Reelle Zahlen

Reelle Zahlen sind positive und negative Dezimalbrüche (Dezimalzahlen). Zur internen Darstellung reeller Zahlen werden binär codierte Dezimalzahlen (BCD) oder Gleitkommazahlen bevorzugt.

Für die *BCD-Darstellung* stehen folgende binäre Zifferncodes (Gruppen zu 4 Bit) zur Verfügung:

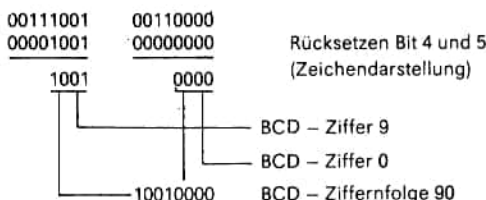
0000 (0)	0101 (5)
0001 (1)	0110 (6)
0010 (2)	0111 (7)
0011 (3)	1000 (8)
0100 (4)	1001 (9)

Das Grundprinzip der BCD-Darstellung besteht darin, jede (externe) Dezimalziffer einzeln zu codieren. Die Anzahl der Zifferncodes richtet sich nach der Anzahl der Stellen der Dezimalzahl. Jedes Byte kann zwei codierte Ziffern aufnehmen. Außerdem werden am Anfang folgende Byte belegt:

- Anzahl der codierten Ziffern,
- Kommaposition (z. B. kann 00000010 = 2 bedeuten: Komma steht links von der zweiten Ziffer),
- Vorzeichen (z. B. 0000 für +, 0001 für -).

So entsteht das unten abgebildete allgemeine BCD-Format.

Die BCD-Darstellung läßt sich aus der (nichtnumerischen) Eingabeinformation einfach gewinnen. Die Vorgehensweise wird an der Dezimalziffer 90 gezeigt. Ausgangspunkt ist der Code in Tabelle 1.4.



Hinzu kommt die Ermittlung der dynamischen Länge (hier 00000010 = 2), der Kommaposition (hier 00000011 = 3, also links der dritten Ziffer) und des Vorzeichens (00000000 = 0 für positiv). Die Gesamtdarstellung belegt 4 Byte:

00000010 00000011 00000000 10010000

Die BCD-Darstellung benötigt, besonders bei größeren Zahlen, relativ viel Speicherplatz. Nachteilig ist außerdem, daß eine Arithmetik technisch realisiert werden muß, die die nichtbelegten (ungültigen)

n	Kommaposition	Vorzeichen	1. Ziffer	2. Ziffer	...	n. Ziffer
---	---------------	------------	-----------	-----------	-----	-----------

vierstelligen »Zifferncodes«, die Pseudotraden, bei einem Bitübertrag behandelt. Der Vorteil der BCD-Arithmetik besteht in einer absolut genauen internen Darstellung von Zahlen. Die BCD-Darstellung ist günstig für die schnelle Konvertierung aus dem nichtnumerischen in das numerische Format und damit für die Ein- und Ausgabe von Massendaten. Beides hat für Anwendungen in der Wirtschaft besondere Bedeutung. Die konkrete interne BCD-Darstellung ist programmabhängig und meist auch mit einigen Einschränkungen versehen.

Das Grundprinzip der *Gleitkommadarstellung* besteht darin, Dezimalzahlen im festen Format darzustellen. Zunächst werden die Zahlen vor und nach dem Komma in eine Binärdarstellung überführt. Die Überführung geschieht so, daß die Bedingungen der allgemeinen Zahlenformel

$$\text{Zahlenwert} = z^T \cdot s$$

erfüllt sind. Der Stellenwertvektor enthält wie für ganze Zahlen die Stellenwerte. Das Komma wird dadurch berücksichtigt, daß Stellen nach dem Komma eine negative Positionsnummer erhalten.  $i$  ist also wieder die Ziffernposition, aber relativ zur Kommastellung. Die Grenzen für  $i$  sind mit  $i = m(1) n$  allgemeiner und können negativ sein. Als binäre Darstellung der Dezimalzahl 90,90 ergibt sich nach der allgemeinen Zahlenformel:

$$01011010,1110011001100110$$

Das Komma wurde symbolisch eingetragen. Die Richtigkeit dieser binären Darstellung läßt sich prüfen, wenn sie als Ziffernvektor aufgefaßt und von rechts mit dem Stellenwertvektor

$$s^T = [2^7 \ 2^6 \dots 2^0 \ 2^{-1} \ 2^{-2} \dots 2^{-16}]$$

multipliziert wird. Das Skalarprodukt ist (Nullelemente werden nicht dargestellt):

$$\begin{array}{r} 64 \\ + 16 \\ + 8 \\ + 2 \\ + 0,5 \\ + 0,25 \\ + 0,125 \\ + 0,015625 \\ + 0,0078125 \\ + 0,0009766 \\ + 0,0004883 \\ \hline = 90,8999024 \end{array}$$

Das ermittelte Bitmuster entspricht nicht der internen Darstellung der Zahl. Es fehlt die Darstellung des Kommas, denn das wurde nur symbolisch eingetragen. Damit die stets feste Anzahl der Bitstellen maximal genutzt werden kann, wird zunächst eine Normalisierung der Binärzahl vorgenommen. Dazu wird sie in Mantisse und Exponent zerlegt. Die Mantisse befindet sich nach der Normalisierung immer in einem festgelegten Bereich.

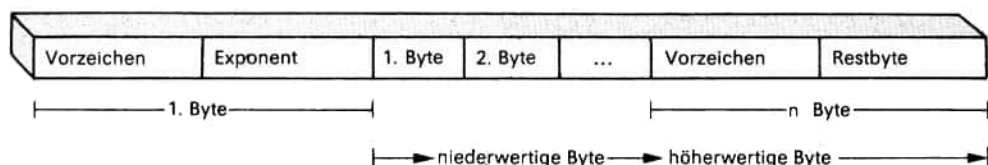
Möglich ist die folgende normalisierte Darstellung (binär):

$$1,011010111001100110011$$

Die Zahl befindet sich jetzt im Bereich größer oder gleich  $2^0$  und kleiner als  $2^1$ . Das Komma wurde um 6 Stellen versetzt, und führende Nullen wurden gestrichen. Die ursprüngliche Kommastellung wird als Exponent zur Basis 2 festgehalten und in einem gesonderten Byte abgelegt. Hier ist der Exponent 6, das Byte ist:

$$00000110$$

Die Darstellung von Mantisse und Exponent erfolgt ohne Berücksichtigung des Vorzeichens. Ist das Vorzeichen negativ, so wird das Zweierkomplement wie bei ganzen Zahlen gebildet. Dadurch werden die höchstwertigen Bit wieder zur Darstellung der jeweiligen Vorzeichen verwendet. Das Grundformat der Gleitkommadarstellung ist also:



Wieviele Bytes zur Darstellung der Mantisse bereitstehen, hängt von den Programmen ab, die die Gleitkommadarstellung steuern. Standardisiert sind Gleitkommadarstellungen mit 4 Bytes (IEEE 754) und 6 Bytes (AMD 9511). Jeweils ein Byte wird für den Exponenten und sein Vorzeichen und 3 oder 5 Bytes werden für die Mantisse einschließlich des Vorzeichens der Mantisse verwendet.

Die Vorteile der Gleitkommadarstellung liegen in einer relativ günstigen Arithmetik und darin, daß auch sehr große Zahlen dargestellt werden können (etwa  $10^{-38}$  bis  $10^{+38}$ ). Ein Nachteil ist die Ungenauigkeit der Darstellung, die sich ergibt, weil eine praktisch unbegrenzte Zahlenmenge (an eine Dezimalzahl lassen sich immer wieder weitere Ziffern anhängen) auf einem in seiner Länge festen Bereich dargestellt werden muß. Die Genauigkeit der approximierten Mantisse hängt von der Anzahl der Bytes ab, die zur Darstellung der Mantisse verfügbar sind. Bei 3-Byte-Mantissen liegt sie zwischen 6 und 7 Stellen, bei 5-Byte-Mantissen zwischen 11 und 12 Stellen.

Die Überführung der externen nichtnumerischen und numerischen Information in die interne Darstellung ist nicht Aufgabe des Nutzers. Entweder ist sie bereits hardwaremäßig realisiert oder es existieren dazu Programme. Der Nutzer muß jedoch die Grundprinzipien kennen, da der Informationstyp dem Computer bei der Programmierung oder bei der Verwendung von Standardsoftware mitgeteilt werden muß (Datentypen BYTE und WORD für

ganze Zahlen ohne Vorzeichen; Datentypen SHORTINT, INTEGER und LONGINT für ganze Zahlen mit Vorzeichen; Datentyp REAL für reelle Zahlen). Auch können diese Kenntnisse bei der Auswahl von Hardware (z. B. Arithmetikcoprozessor) oder von Software (z. B. Compiler mit BCD-Arithmetik) benötigt werden. Der Umgang mit den verschiedenen Datentypen hat direkte Wirkungen für den Nutzer, das betrifft z. B. die Ungenauigkeit der Gleitkommadarstellungen oder einen Überlauf bei der Darstellung positiver ganzer Zahlen.

### 1.3. Automatisierte Informationsverarbeitung

#### 1.3.1. Automatisierte Informationsverarbeitungssysteme

Die Nutzung eines Computers zur Informationsverarbeitung in Arbeitsprozessen erfordert folgende Komponenten:

- das Rechnergrundgerät und einige Zusatzgeräte (Hardware),
- Programme und dazugehörige Dokumentationen (Software),
- Information in maschinenlesbarer Form (Daten), die verarbeitet, gespeichert oder übertragen werden soll,
- organisatorische Festlegungen und Organisationsmittel (Orgware).

#### Hardware

Die wichtigste Komponente der Hardware ist der Rechner.

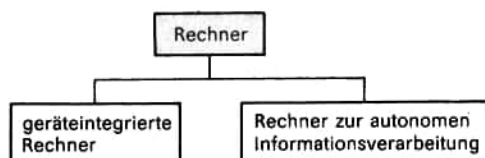


Abbildung 1.10.  
Grundlegende Rechnerklassifikation

**Ein Rechner (Computer im engeren Sinne) ist eine Funktionseinheit zur programmgesteuerten Verarbeitung von Information mittels umformender, strukturierender, übertragender und speichernder Operationen.**

Rechner werden entweder in andere technische Geräte, z. B. Werkzeugmaschinen, für deren Steuerung eingebaut oder bilden den Kern autonomer Gerätesysteme zur ausschließlichen Informationsverarbeitung. Es ergibt sich die in Abbildung 1.10. dargestellte grundlegende Rechnerklassifikation.

Geräteintegrierte Rechner müssen sich in Struktur und Leistung dem eigentlichen Finalprodukt anpassen und dessen Gebrauchswert erhöhen. Geräteintegrierte Rechner werden in diesem Buch nicht behandelt. Rechner zur autonomen Informationsverarbeitung werden mit Zusatzgeräten ausgestattet, um die Kommunikation mit dem Menschen zu ermöglichen und große Informationsmengen zu speichern. Sie bilden ein Rechnersystem.

**Ein Rechnersystem ist die Gesamtheit aus Rechner und peripheren Geräten, wie Bildschirm, Tastatur, Drucker und Speichertechnik.**

Die konkreten Bestandteile und technischen Parameter des Rechnersystems bilden eine konkrete Maschine mit potentieller Leistungsfähigkeit. Sie werden im Kapitel 2. beschrieben.

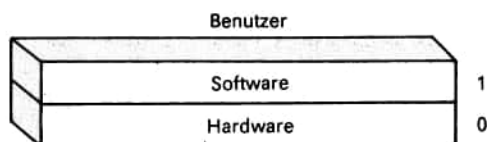


Abbildung 1.11.  
Schichten der Nutzung  
eines Rechnersystems

### Software

In welchem Maße die potentielle Leistungsfähigkeit der Hardware bei der praktischen Anwendung tatsächlich nutzbar gemacht werden kann, hängt von der eingesetzten Software ab.

**Software ist die Gesamtheit der allgemein nutzbaren Programme einschließlich der Dokumentationen.**

Software aktiviert Teile des universellen Leistungsvermögens der sonst nur auf rein technischem Niveau nutzbaren Maschine. Sie bildet unter Nutzung der Hardware scheinbar eine »neue« Maschine mit spezifischen Gebrauchseigenschaften. Der Nutzer steht dieser virtuellen (scheinbaren) Maschine gegenüber. Wechselt die Software, so wechseln die Eigenschaften des Systems, und es ist wieder eine »neue« Maschine entstanden. Auf diese Weise entstehen die in Abbildung 1.11. dargestellten Schichten virtueller Maschinen.

Schichten beschreiben virtuelle Maschinen mit einer Gesamtheit von Funktionen, die an der oberen Schnittstelle unter Nutzung unterer Schichten verfügbar sind. Der Nutzer hat als Schnittstelle die Softwareschicht, die ihm unter Nutzung der Hardware eine definierte Leistung anbietet. Diese Schichtenbetrachtung wird im Kapitel 3. verfeinert.

Software kann nach der Zweckbestimmung (vgl. Abbildung 1.12.) klassifiziert werden.

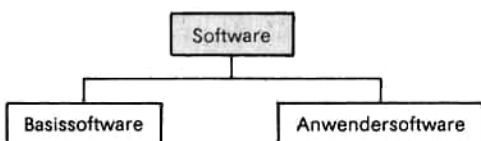


Abbildung 1.12.  
Grobklassifikation von Software

**Basissoftware (Systemsoftware) ist Software zur Steuerung multivalenter, anwendungsunabhängiger (oft gerätebezogener) Operationen.**

Die Basissoftware steuert den allgemeinen Arbeitsablauf innerhalb des Rechnersystems und unterstützt dessen Bedienung. Wichtigster Bestandteil der Basissoftware ist das Betriebssystem (vgl. Kapitel 3.). Das Betriebssystem erschließt die allgemeinen Eigenschaften der Hardware.

**Anwendersoftware ist Software zur Bearbeitung von spezifischen Anwenderproblemen.**

Anwendersoftware wird im Abschnitt 4.1. behandelt

#### Daten

Ein Rechnersystem verarbeitet (transformiert) programmgesteuert Eingabeinformation zu Ausgabeinformation. Die Darstellung der Information erfolgt durch Daten, die aus logisch-inhaltlicher Sicht konkrete Ausprägungen der Eigenschaften von Objekten sind. Aus der Sicht der Verarbeitung, Speicherung und Übertragung sind verschiedene Datentypen zu unterscheiden.

**Ein Datentyp ist eine Menge definierter Werte, die hinsichtlich Darstellungsattribut (numerisch, nichtnumerisch), Struktur und zulässiger Operationen festgelegt ist.**

Für die automatisierte Informationsverarbeitung in der Wirtschaft sind große Datenmengen, Massendaten, charakteristisch. Zur Verarbeitung, Speicherung und Übertragung durch ein Rechnersystem werden Daten zu Datensätzen und Dateien zusammengefaßt.

**Ein Datensatz ist eine Gruppe von Daten mit dem gleichen sachlichen oder funktionellen Bezug.**

In vielen Fällen besteht ein Datensatz aus

- Hauptordnungsmerkmal zur eindeutigen Identifikation des mit dem Datensatz beschriebenen Objekts,
- Nebenordnungsmerkmalen für verschiedene Gruppierungen und Verknüpfungen mit anderen Objekten,
- Auswertungs- und Hinweisdaten, die der Verarbeitung unterliegen.

Als Ordnungsmerkmale werden oft Schlüssel verwendet.

**Schlüssel sind identifizierende Kurzbeschreibungen von Attributen, die nach einer festgelegten Regel, der Schlüsselsystematik, gebildet werden.**

Abhängig von der Schlüsselsystematik kann ein Schlüssel außerdem ordnend und/oder klassifizierend sein. Nach der Darstellungsform werden numerische und alphanumerische Schlüssel unterschieden. Schlüssel sind infolge fehlender Redundanz fehlerkritisch. Sie werden deshalb durch spezielle Erweiterungen besonders geschützt. Beispiele für Schlüssel, die auch zur automatisierten Informationsverarbeitung genutzt werden, sind Kontonummern in Banken und Sparkassen sowie Erzeugnis-, Leistungs- und Handelsschlüsselnomenklaturen, z. B. die Europäische Artikel-nomenklatur (EAN).

Mehrere Datensätze können eine Datei bilden.

**Eine Datei (File) ist eine (sachbezogene) Zusammenfassung von gleichstrukturierten Datensätzen, die im verarbeitungstechnischen Zusammenhang stehen.**

Die Anzahl der Datensätze (Komponenten) einer Datei ist nicht begrenzt. Die Komponenten haben alle die gleiche Länge oder sind durch Steuerzeichen voneinander getrennt (Textdateien). Das Ende einer Datei wird durch Steuerzeichen hinter der letzten Komponente gekennzeichnet oder ergibt sich aus der Nachweisführung durch das Betriebssystem.

Eine Datei ist für den sequentiellen oder für den wahlfreien Zugriff organisiert.

**Sequentieller Zugriff ist eine Zugriffsform, bei der auf Dateikomponenten nur zugegriffen werden kann, wenn vorher auf alle ihre Vorgänger zugegriffen wurde.**

Der Aufwand für die Speicherung von Dateien mit sequentiellem Zugriff ist relativ gering, der Zugriff aber zeitintensiv, wenn die Komponenten in einer anderen als der vorhandenen Reihenfolge zu verarbeiten sind.

**Wahlfreier (direkter) Zugriff ist eine Zugriffsform, bei der auf beliebige Dateikomponenten zugegriffen werden kann, ohne daß vorher auf andere Komponenten zugegriffen werden muß.**

Die Speicherung von Dateien mit wahlfreiem Zugriff ist aufwendiger als für Dateien mit nur sequentiellem Zugriff, die Zeit des Zugriffs bei Verarbeitung in einer anderen als der vorhandenen Reihenfolge ist jedoch bedeutend kürzer.

Die Organisation für den wahlfreien Zugriff schließt die Möglichkeit des sequentiellen Zugriffs ein. Dateien mit wahlfreiem Zugriff setzen Datenträger voraus, die diesen Zugriff auch physisch ermöglichen

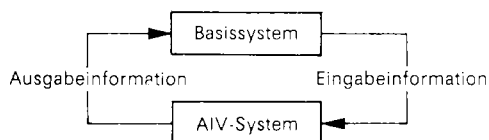


Abbildung 1.13.  
Zusammenwirken von Basissystem und AIV-System

(vgl. Abschnitt 2.2.3.). In der praktischen Organisation der Speicherung von Daten gibt es Zwischenformen.

Bei der komplexen Automatisierung wird nach einer einheitlich organisierten und durch spezielle Programme realisierten Verwaltung aller Daten eines Anwendungsbereiches gestrebt. Eine solche Organisationsform für Daten heißt Datenbank.

**Eine Datenbank ist eine Gesamtheit von Dateien, die von einem Datenbankbetriebssystem verwaltet werden.**

Kennzeichnend für Datenbanken ist die relative Unabhängigkeit von den verschiedenen (Auswertungs-)Programmen, die auf vielseitige Zugriffserfordernisse ausgerichtete interne Struktur (das Datenmodell) und die weitgehende Vermeidung der Mehrfachspeicherung von Daten.

Dateien und Datenbanken müssen gegen unbeabsichtigte Veränderungen und unberechtigten Zugriff gesichert werden.

### **Automatisierte Informationsverarbeitungssysteme**

**Hardware, Software und Dateien bilden in ihrem funktionellen Zusammenwirken für ein konkretes Basissystem ein automatisiertes Informationsverarbeitungssystem (AIV-System).**

Die Eingabeinformation eines AIV-Systems wird Prozessen in Natur, Technik, Wissenschaft und Gesellschaft entnommen. Die

Ausgabeinformation dient dazu, die Prozesse zu beeinflussen. Aus dieser Sicht können die zu beeinflussenden Prozesse als Basisprozesse und ihr materieller Träger als *Basissystem* bezeichnet werden. Basissystem und AIV-System bilden dann in ihrer Gesamtheit ein neues System (vgl. Abbildung 1.13.).

Die Umsetzung der Ausgabeinformation des AIV-Systems in eine zielgerichtete Beeinflussung des Basissystems erfolgt entweder durch Eingriff des Menschen (offline) oder durch spezielle Geräte (online). Die Ergebnisse werden erfaßt und bilden erneut Eingabeinformation. Auch das kann offline oder online erfolgen. Sind Basissystem und AIV-System online verbunden, so wird die Gesamtheit aus Basissystem und AIV-System als automatisches System, sonst als *rechnergestütztes System* bezeichnet.

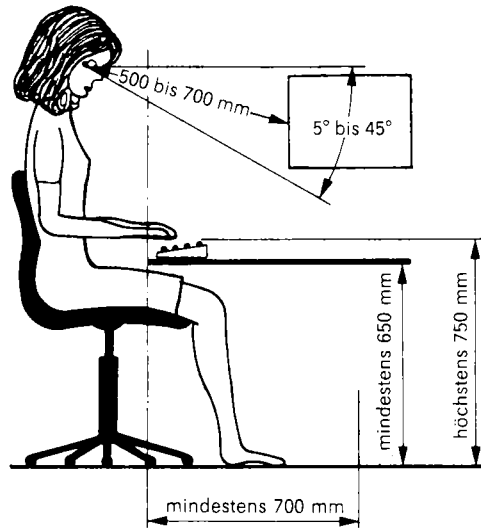


Abbildung 1.14.  
Abmessungen für die Arbeit mit Bildschirm und Tastatur

### Rechnergestützte Arbeitsplätze

Rechnergestützte Systeme können durch einen oder mehrere rechnergestützte Arbeitsplätze (Arbeitsstationen) realisiert werden. Nach Arbeitsaufgaben lassen sich z. B. Arbeitsplatz Materialwirtschaft, Arbeitsplatz Konstruktion und Technologie, Disponentenarbeitsplatz, Arbeitsplatz Kostenrechnung unterscheiden. Hardware und Dateien können zu verschiedenen Zeitpunkten unterschiedlichen Arbeitsplätzen zugeordnet sein.

Bei der Gestaltung von rechnergestützten Arbeitsplätzen sind ergonomische Fragen von Bedeutung. Sie beeinflussen die Akzeptanz und Effektivität der Automatisierungslösung erheblich. Es wird zwischen Hardware-, Software- und Umgebungsergonomie unterschieden.

Parameter der *Hardwareergonomie* sind:

- Helligkeit des Bildschirms und Kontrast zwischen Schrift und Hintergrund,
- Nachleuchtdauer (Persistenz) und Wie-

derhol frequenz der Bildanzeige (Flimmereffekt), Schriftgröße und Bildauflösung,

- Geräuschpegel des Lüfters am Rechnergrundgerät sowie der physischen Zugriffe zu externen Speichern, .
- Lage, Tastenanordnung und optische Gestaltung der Tastatur.

Die *Softwareergonomie* wird wesentlich durch die Qualität der Anwendersoftware beeinflusst (vgl. Abschnitt 4.1.). Parameter der Softwareergonomie sind z. B.:

- Klarheit und Einfachheit der Bedienung und Dialogführung,
- Abstimmung der Rechnerleistung auf Form und Inhalt der Arbeitsaufgabe,
- Reaktionszeiten auf Nutzeranforderungen.

Parameter der *Umgebungsergonomie* sind:

- Arbeitsplatzbeleuchtung,
- Akustik,



Abbildung 1.15.  
Rechnergestützter Arbeitsplatz mit Personalcomputer

- Raumklima,
  - Arbeitsplatzabmessungen.
- Mögliche Arbeitsplatzabmessungen für die Arbeit mit Bildschirm und Tastatur enthält Abbildung 1.14. (TGL 44 690/01...05).

### **Orgware**

Die Bereitstellung von Ein- und Ausgabeinformation, Hardware, Software und Daten in rechnergestützten Systemen bedarf der Organisation.

**Orgware ist die Gesamtheit der dokumentierten Festlegungen und Organisationsmittel zur Gestaltung der Arbeitsabläufe innerhalb eines AIV-Systems und für das Zusammenwirken mit der Umgebung.**

Die Festlegungen der Orgware betreffen die Verfügbarkeit der Ressourcen des AIV-Systems, Sicherheitskonzepte für Programme und Daten, die Datenerfassung

und Datenbereitstellung. Sie regeln alle Abläufe, die der Verarbeitung von Information vor- oder nachgelagert sind.

Die Einführung von AIV-Systemen und ihre effektive Verbindung mit Basissystemen bewirkt eine neue Informationstechnologie in der Wirtschaft. Der Mensch wird bei der Lösung seiner Arbeitsaufgaben in den Produktions- und Zirkulationsprozessen, der Leitung, Planung, Bilanzierung, Abrechnung und Kontrolle des Reproduktionsprozesses durch AIV-Systeme unterstützt. Damit beschäftigt sich die Wirtschaftsinformatik.

**Die Wirtschaftsinformatik ist die Wissenschaft von der Gestaltung, Entwicklung und Nutzung automatisierter Informationsverarbeitungssysteme in der Wirtschaft.**



Wirtschaftsinformatik ist angewandte Informatik im Gegenstandsbereich der Wirtschaftswissenschaften. Sie berücksichtigt die wirtschaftswissenschaftlichen Spezifika des Anwendungsbereiches und nutzt Modelle und Methoden der Mathematik und Kybernetik.

Typisch für die Struktur von AIV-Systemen in der Wirtschaft ist, daß zentral installierte Großrechenanlagen den Kern eines (mehrstufigen) AIV-Systems bilden und ihre Leistung über Personalcomputer bis an den (rechnergestützten) Arbeitsplatz heranbringen. Im Dialog ruft der Mensch früher gespeicherte Information von einer Zentralrechenanlage ab, ergänzt diese und löst die erforderliche Verarbeitung aus. Die Verarbeitungsergebnisse werden auf dem Bildschirm, dem Drucker oder anderen Ausgabegeräten innerhalb kurzer Zeit angezeigt. Dadurch kann die Rechnerleistung direkt in die Arbeitsprozesse integriert werden. Darüber hinaus stehen die Personalcomputer für die weitere autonome Rationalisierung am Arbeitsplatz zur Verfügung (vgl. Abschnitt 2.1.1.). Diese Informationstechnologie verändert den Charakter der Arbeit, die Arbeitsinhalte und die Arbeitsorganisation. Sie ist ein wichtiger Produktivitätsfaktor.

In Anlehnung an den internationalen Sprachgebrauch werden rechnergestützte Systeme auch als CA(Computer Aided)-Systeme, rechnergestützte Arbeitsplätze als CA-Arbeitsplätze oder CA-Stationen und komplexe Softwarelösungen als CA-Lösungen bezeichnet. In bezug auf den Bereich der materiellen Produktion und dessen Vorbereitung sind z.B. folgende CA-Lösungen definiert:<sup>5</sup>

- CAD (Computer Aided Design):  
Rechnergestützte konstruktive Vorbereitung;
- CAM (Computer Aided Manufacturing):  
Rechnergestützte Fertigung;

- CAP (Computer Aided Planning):  
Rechnergestützte technologische Vorbereitung.

Es ist naheliegend, die verschiedenen rechnergestützten Systeme in einem Bereich zusammenzuführen, um so eine durchgängige Automatisierungslösung zu erreichen. Ein System durchgehender rechnergestützter Planung, Vorbereitung, Durchführung und Abrechnung der Produktion trägt die Bezeichnung CIM (Computer Integrated Manufacturing). Das CA-Konzept und das CIM-Konzept werden auch auf nichtproduktive Bereiche der Volkswirtschaft übertragen.

### 1.3.2.

#### Ökonomische Gesichtspunkte von AIV-Systemen

Die Entwicklung und Nutzung von AIV-Systemen ist eine der Grundrichtungen zur Verwirklichung des wissenschaftlich-technischen Fortschritts. Sie ist für die Mitgliedsländer des Rates für Gegenseitige Wirtschaftshilfe bis zum Jahr 2000 konzipiert<sup>6</sup> und in ihrer Umsetzung Bestandteil der Wirtschaftsstrategie der DDR.

#### Ökonomische Ziele

Die grundlegenden ökonomischen Ziele der Entwicklung und Nutzung von AIV-Systemen in der Wirtschaft sind:

- Steigerung der Arbeitsproduktivität durch Einsparung von Arbeitszeit und Arbeitskräften bei gleicher Leistung oder durch Leistungszuwachs bei gleichem Arbeitskräftepotential;
- Senkung des spezifischen Material- und Energieverbrauchs in Produktion und

5 Vgl. Autorenkollektiv: CAD/CAM Schlüsseltechnologie als Intensivierungsfaktor. Berlin: Verlag Die Wirtschaft 1987, S. 31.

6 Komplexprogramm des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW bis zum Jahre 2000. In: Einheit, Berlin 41 (1986) 2, S. 167.

Zirkulation, insbesondere durch verbesserte Konstruktion der Erzeugnisse, verbesserte Fertigungstechnologien und Arbeitsorganisation sowie Verkürzung von Transportwegen; -

- Beschleunigung des Reproduktionsprozesses, z. B. durch Verkürzung der Forschungs-, Entwicklungs- und Überlebenszeiten für neue Erzeugnisse, Verkürzung der Durchlaufzeiten in der Produktion und Erhöhung der Warenumschlaggeschwindigkeit im Handel;
- Erhöhung der Flexibilität des Reproduktionsprozesses zur schnelleren Anpassung von Produktion und Handel an sich verändernde Bedürfnisse der Bevölkerung und an Anforderungen des Weltmarktes.

Hinzu kommen soziale Ziele, besonders die Verbesserung der Arbeits- und Lebensbedingungen der Werktätigen durch Verringerung des Anteils geistiger Routinearbeit, und gesellschaftliche Ziele.

Nutzen, Aufwand und Effektivität der Entwicklung und des Einsatzes von AIV-Systemen sind zu planen und nachzuweisen. Gesetzliche Grundlagen dafür sind die Rahmenrichtlinie für die Ermittlung, Planung, Kontrolle und Abrechnung der Effektivität der Maßnahmen des wissenschaftlich-technischen Fortschritts<sup>7</sup>, Grundsätze für das einheitliche Herangehen (Nutzensanordnung<sup>8</sup>) und ergänzende Regelungen für einzelne Wirtschaftsbereiche.

### Nutzen

Als Nutzen von AIV-Systemen in der Wirtschaft gelten der ökonomische Nutzen, der soziale Nutzen aus der Verbesserung von Arbeits- und Lebensbedingungen und der gesellschaftliche Nutzen durch die Erhöhung von Ordnung und Sicherheit.

**Der ökonomische Nutzen für ein AIV-System ist die Summe aus meßbarem Lei-**

**stungszuwachs und nachweisbaren Einsparungen abzüglich laufender Mehraufwendungen für ein Planjahr. -**

Die Ermittlung des Nutzens aus Leistungszuwachs und aus Einsparungen ist abhängig vom Basisprozeß und deshalb sehr differenziert. Im allgemeinen ist eine Vergleichsbasis erforderlich. Laufender Mehraufwand entsteht durch Druckerpapier, Farbbänder, Vordrucke, Datenträger, Energieverbrauch, Wartung, Reparatur und Lohnkosten.

Zur Darstellung des Nutzens werden Nutzenskennziffern verwendet. Dazu gehören Zuwachs an Produktion, Selbstkostensenkung, Erhöhung des Warenumsatzes, Beschleunigung des Warenumschlages, Einsparung an Material sowie Einsparung an Arbeitszeit und Gewinnung von Arbeitskräften. Die Nutzenskennziffern sind nach Bestätigung der Planung zugrunde zu legen. In vielen Fällen ist der Einsatz von AIV-Systemen untrennbar mit anderen Rationalisierungsmaßnahmen, wie Anwendung mathematischer oder kybernetischer Methoden, betriebsorganisatorischen Maßnahmen, technisch-technologischen Veränderungen in Produktion, Umschlag, Lagerhaltung und Transport, verbunden und nur mit diesen zusammen wirksam. Im allgemeinen kann der Nutzen nur komplex nachgewiesen werden.

Der Nutzen wird zunächst als Nutzen beim unmittelbaren Anwender, also einem Betrieb, Kombinat oder einer staatlichen Einrichtung, berechnet. Außerdem können

7 Anordnung vom 5. 2. 1982 über die Rahmenrichtlinie für die Ermittlung, Planung, Kontrolle und Abrechnung der Effektivität der Maßnahmen des wissenschaftlich-technischen Fortschritts. In: GBl. I/1982 Nr. 8 S. 165.

8 Anordnung vom 19. 12. 1986 über Grundsätze für das einheitliche Herangehen an die Ermittlung, Planung und Nachweisführung des Nutzens und der Effektivität der Maßnahmen des wissenschaftlich-technischen Fortschritts – Nutzensanordnung. In: GBl. I/1987 Nr. 1 S. 1.

die Entwicklung und der Einsatz von AIV-Systemen Nutzen in anderen Bereichen der Volkswirtschaft erzielen. Das ist eine Folge der arbeitsteiligen Verflechtung der Produktion sowie der Verflechtung von Produktion und Handel. Für die Ermittlung des volkswirtschaftlichen Nutzens enthalten die gesetzlichen Bestimmungen Besonderheiten.

### Aufwand

**Der (einmalige) Aufwand für ein AIV-System ist die Summe aller einmaligen materiellen, personellen und finanziellen Aufwendungen zur Schaffung dieses Systems abzüglich einmaliger Erlöse.**

Zu beachten ist, daß laufende Mehraufwendungen für ein AIV-System als Abzug vom Nutzen zu betrachten und nicht Bestandteil des (einmaligen) Aufwandes sind. Der einmalige Aufwand umfaßt vor allem

- die Kosten und/oder Investitionen für den Kauf oder die Nutzung von Hardware und Basissoftware (bei Nutzung durch mehrere AIV-Systeme anteilig);
- die Kosten und/oder Investitionen für den Kauf und/oder die Entwicklung von Anwendersoftware;
- die Kosten und/oder Investitionen für bauliche Veränderungen und die Installation der Hardware einschließlich einmaliger Aufwendungen für den Gesundheits-, Arbeits- und Brandschutz;
- Restbuchwerte und Demontagekosten bei der Aussonderung früher eingesetzter Grundmittel;
- Kosten für die Erprobung und die Einführung des AIV-Systems, einschließlich der Kosten für die Schulung künftiger Nutzer.

Den Vorausberechnungen für Hardware und Basissoftware sind gültige Preisangebote zugrunde zu legen. Der Aufwand an

lebendiger Arbeit (in Vollbeschäftigtenheiten) für die Entwicklung von Anwendersoftware kann unter Verwendung von Schätzwerten für die Anzahl erforderlicher Programmzeilen ermittelt werden.

Einmalige Erlöse können durch den Verkauf von Grundmitteln oder Restmaterial erzielt werden.

### Effektivität

**Die Effektivität eines AIV-Systems ergibt sich aus der Gegenüberstellung von (einmaligem) Aufwand und (jährlichem) Nutzen.**

Zur Darstellung der Effektivität werden die Kennziffern Rückflußdauer und Nutzkoeffizient verwendet.

Die *Rückflußdauer* wird nach der Formel

$$\text{Rückflußdauer} = \frac{\text{Aufwand}_{\text{einmalig}}}{\text{Nutzen}_{\text{jährlich}}}$$

errechnet. Sie gibt an, in wieviel Jahren die einmaligen Aufwendungen für die Schaffung des AIV-Systems durch den laufend entstehenden ökonomischen Nutzen zurückfließen. Die Rückflußdauer hängt stark vom Anwendungsgebiet ab. Es werden Werte von 1 bis 2 Jahren angestrebt.

Der *Nutzkoeffizient* für ein AIV-System ist der Kehrwert der Rückflußdauer.

### Übungen

1. Nennen Sie Eigenschaften, die einen Computer für den Menschen nützlich machen, und nennen Sie ökonomische Ziele, die mit dem Einsatz von Computern in der Wirtschaft verbunden sind!
2. Nennen Sie ein Beispiel für ein automatisiertes Informationsverarbeitungssystem in der Wirtschaft, und erläutern

Sie die vier Bestandteile des Systems in ihren Wechselwirkungen!

3. Für zwei mögliche Realisierungsvarianten eines AIV-Systems liegen folgende Schätzwerte (in 1 000 M) vor:

	Vari- ante 1	Vari- ante 2
Leistungszuwachs (Planjahr)	40	40
Einsparungen (Planjahr)	30	45
Mehraufwendungen (Planjahr)	15	20
Aufwendungen zur Schaffung des AIV-Systems	120	130
Verkaufserlös abzulösender Grundmittel	10	10

Welche Variante ist vorzuziehen?

4. Codieren Sie ihren Nachnamen und Vornamen, getrennt durch Komma und Leerzeichen, in eine byteweise Binärdarstellung!
5. Analysieren Sie, wodurch sich die interne Darstellung der Groß- von der internen Darstellung der Kleinbuchstaben unterscheidet!
6. Nennen Sie mindestens zwei Formate numerischer Information, und werten Sie diese für die Anwendung in der Buchführung!

## 2.

# Rechnersysteme aus Nutzersicht

### 2.1.

#### Grundstruktur und Klassen von Rechnersystemen

##### Grundstruktur

Die typische Grundstruktur eines Rechners ist in Abbildung 2.1. dargestellt. Sie geht auf John v. Neumann (1903 bis 1957) zurück.

Die Ein- und Ausgabe von Information bei der Mensch-Maschine-Kommunikation sowie der Maschine-Maschine-Kommunikation erfolgt über Geräte, die in ihrer Gesamtheit die Ein- und Ausgabeperipherie bilden. Der Zentralprozessor übernimmt die programmgesteuerte Verarbeitung der Information. Information (Programme und Verarbeitungsinformation), die sich in ihrer Gesamtheit nicht in den schnellen Zwischenspeichern des Zentralprozessors, den *Registern*, befinden kann, wird im Hauptspeicher zugriffsbereit zwischengespeichert. Zentralprozessor, Hauptspeicher und die für die Steuerung der Ein- und Ausgabe erforderlichen internen Baugruppen, die Anschlußsteuerung, bilden die *Zentraleinheit*. Der dauerhaften Aufbewahrung sowie der Nachschub- und Archivspeicherung größerer Informationsmengen dienen externe Speicher.

Die Darstellung bezieht sich auf Digitalrechner, die Information in diskreter Darstellung verarbeiten. Außerdem gibt es Analogrechner und Mischformen zwischen Digital- und Analogrechnern (Hybridrechner). Information wird hier durch

kontinuierliche Funktionen dargestellt, und für ihre Verknüpfung gibt es spezielle Funktionseinheiten. Die Analogtechnik wird für ökonomische Aufgabenstellungen nicht angewendet.

##### Klassen von Rechnersystemen

Nach dem Leistungsvermögen und der technologischen Realisierung können Rechnersysteme in die in Abbildung 2.2. dargestellten Klassen eingeteilt werden. Unterhalb der Personalcomputer können Kalkulatoren, z. B. Taschenrechner, sowie Heimcomputer eingeordnet werden. Oberhalb der Großrechenanlagen gibt es Superrechner. Diese Klassen werden hier jedoch nicht behandelt.

**Personalcomputer (Bürocomputer, Arbeitsplatzcomputer, Mikrorechnersysteme) sind Rechnersysteme, deren Kern ein Mikroprozessor bildet und deren Peripherie vor allem auf die Mensch-Maschine-Kommunikation ausgerichtet ist.**

Personalcomputer in der DDR sind die Bürocomputer A 5120/30, Personalcomputer PC 1715/1715 W, Arbeitsplatzcomputer A 7100/7150 und der Personalcomputer EC 1834 des VEB Kombinat Robotron. Einen Personalcomputer EC 1834 zeigt die Abbildung 1.15.

Personalcomputer kommunizieren über Bildschirm und Tastatur mit dem Nutzer, besitzen eine nur für die direkte Kommunikation mit dem Nutzer ausgelegte Druck-

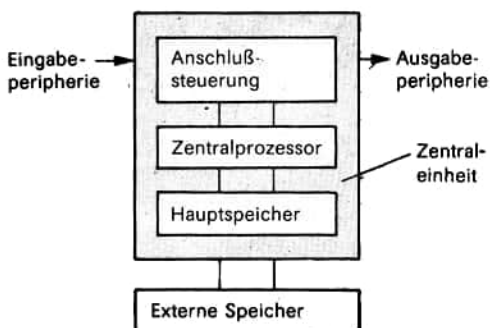


Abbildung 2.1.  
Grundstruktur eines Rechnersystems

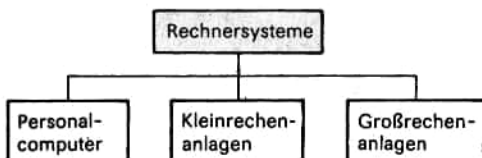


Abbildung 2.2.  
Rechnersystemklassen

technik und sind nicht zur Speicherung von Massendaten (über 100 MByte, vgl. Abschnitt 2.2.1.) ausgestattet. Sie haben sich als autonome Geräte und auch im Verbund mit anderen Rechnersystemen fast alle ökonomischen und technischen Einsatzgebiete erobert.

**Kleinrechenanlagen** (Kleindatenverarbeitungsanlagen, KDVA, Minirechnersysteme) sind Rechnersysteme, deren Kern von einem oder mehreren Mikroprozessoren gebildet wird. Ihre Peripherie ist entweder durch mehrere Bildschirmarbeitsplätze dialogorientiert, durch Eingabetechnik für maschinell lesbare Datenträger und leistungsfähige Drucktechnik stapelverarbeitungsorientiert oder für die Prozeßdatenverarbeitung ausgelegt.

Kleinrechenanlagen besitzen im allgemeinen eine höhere Verarbeitungsgeschwindigkeit als Personalcomputer, eine erheb-

lich leistungsfähigere Ein- und Ausgabetechnik und verfügen über Speichertechnik für Massendaten. Das Produktionsprogramm für Kleinrechner ist zwischen den RGW-Ländern abgestimmt. Einige Systeme tragen CM-Chiffren, wie CM-4, CM-52. Kleinrechenanlagen sind das Basisrechnersystem K 1600 und das Rechnersystem mit virtuellem Speicher (RVS) K 1840 des VEB Kombinat Robotron.

Der Einsatz von Kleinrechenanlagen kann autonom zur Echtzeitsteuerung technischer Prozesse, zur Laborautomatisierung, für Lagerhaltungsprobleme, für Massendatenverarbeitung mittleren Umfangs, für CAD/CAM-Systeme und im Verbund mit anderen Rechnersystemen als Datenkonzentrator, Server und Kommunikationsrechner erfolgen.

**Großrechenanlagen** (universelle Rechenanlagen, elektronische Datenverarbeitungsanlagen, EDVA, mainframes) sind Rechnersysteme, deren Kern im allgemeinen durch mehrere Prozessoren, auch Spezialprozessoren, für intensive Verarbeitung gebildet wird. Großrechenanlagen sind mit umfassender und universeller Peripherie, besonders für umfangreiche Ein- und Ausgabe, ausgestattet.

Der *Durchsatz* (performance), die Rechnerleistung in einer Zeiteinheit, ist bei Großrechenanlagen erheblich größer als bei Personalcomputern oder Kleinrechenanlagen.

Das Erzeugnisprogramm für Großrechenanlagen ist zwischen den Ländern des RGW abgestimmt. Die Geräte tragen die Bezeichnung EC und EC-Chiffren. Großrechenanlagen sind z. B. EC 1025 (ČSSR), EC 1035 (UdSSR), EC 1040 (DDR), EC 1045 (VRP/UdSSR), EC 1057 (DDR) und EC 1065 (UdSSR).

Großrechenanlagen werden für die zentrale Verarbeitung und Verwaltung großer

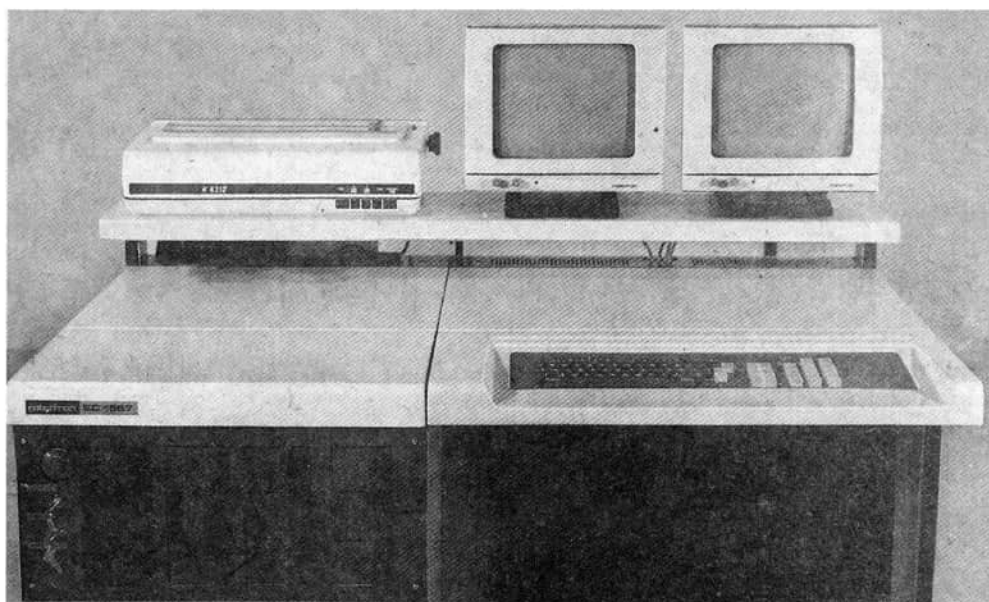


Abbildung 2.3.  
Großrechenanlage EC 1057 des VEB Kombinat Robotron (Bedieneinheit)

Datenmengen, für besonders rechenintensive wissenschaftlich-technische und wissenschaftliche Aufgaben eingesetzt. Über den Verbund mit dezentralen Personalcomputern oder Kleinrechenanlagen kann die Leistung der Großrechenanlage verteilt verfügbar sein und bis an den Arbeitsplatz herangebracht werden.

Die Grenzen zwischen Rechnerklassen sind nicht eindeutig und verschieben sich. Trotzdem werden die Rechnerklassen, die vor allem nach der Ausstattung, der *Konfiguration*, eines Rechnersystems gebildet werden, erhalten bleiben. Die Entwicklung geht zu einer abgestimmten Arbeitsteilung zwischen den Klassen der Rechnersysteme.

### Rechnergenerationen

Die technische Entwicklung der Rechnersysteme ist vor allem gekennzeichnet durch die jeweils verfügbare Bauelementebasis (Elektronenröhren, Transistoren, inte-

griierte Schaltkreise, hochintegrierte Schaltkreise), die Taktfrequenzen, die interne Speicherkapazität, die programmtechnischen Mittel der Programmierung, die Technologie der Ressourcensteuerung, die Leistungsfähigkeit der Peripherie und die Nutzungstechnologien. Eine mögliche Einteilung enthält Tabelle 2.1.

Eine wesentliche Weiterentwicklung der Rechnersysteme wird mit der *fünften Rechnergeneration* erwartet. Grundlage der fünften Rechnergeneration ist die VLSI-Technik (Very Large Scale Integration), die wesentlich kompakter ist als die LSI-Technik (Large Scale Integration).

Wichtige Ziele für die Entwicklung der fünften Rechnergeneration sind:

- Erleichterung des Zugangs eines Nutzers zur Rechnerleistung durch die Ein- und Ausgabe von Information in der Fachsprache des Nutzers und in natürlicher Form (akustische Eingabe, optische Eingabe);

Tabelle 2.1.  
Rechnergenerationen

Generation	Zeit	Bauelemente	Peripherie	Erzeugnis
1	1945–1955	Röhren	Schreibmaschine	ZRA 1
2	1955–1965	diskrete Halbleiter	Lochstreifen/Lochkarte/ Magnetband	R 300
3	1965–1975	kompakte Halbleiter	Magnetplatten/Bildschirmtechnik	EC 1040
4	1975–1985	LSI-Elemente	Rechnersysteme	EC 1057

- objektorientierte Hardware (z. B. Assoziativspeicher, die inhaltsorientiert adressierbar sind) und Möglichkeiten objektbezogener Programmierung;
- Verbesserung der Ressourcensteuerung, so daß dem Nutzer virtuell nahezu unbegrenzter Speicherplatz und hohe Prozessorleistung zur Verfügung stehen;
- Erweiterung der Speicherung von Information in Form von Fakten durch die Möglichkeit, auch Regeln und damit Wissen zu speichern.

Für die fünfte Rechnergeneration sind Sprachen mit einem sehr hohen Niveau zu schaffen. Sie müssen die Formulierung unterschiedlicher Anwendungen durch den Endnutzer (ohne Hilfe eines qualifizierten Programmierers) ermöglichen. Im Komplexprogramm des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW bis zum Jahre 2000<sup>1</sup> sind die wichtigsten Aufgaben zur Entwicklung der fünften Rechnergeneration fixiert. Es soll ein evolutionärer Übergang von den bisherigen Rechnerfamilien des RGW gesichert werden. Das geschieht durch ein Konzept virtueller Maschinen, die die Veränderungen in Hard- und Software vor dem Endnutzer abfangen.

### Rechnerfamilien

Außer nach Rechnergenerationen kann auch nach Rechnerfamilien klassifiziert werden.

**Eine Rechnerfamilie ist eine Gruppe aufeinander abgestimmter und leistungsmäßig abgestufter Rechnersysteme mit gemeinsamem Grundbefehlssatz.**

Durch den gemeinsamen Grundbefehlssatz sind Geräte und Daten verschiedener Rechnersysteme einer Rechnerfamilie verträglich (kompatibel) und Programme übertragbar (portabel). Kompatibilität von Geräten und Portabilität von Programmen gelten in der Regel nur von den weniger leistungsfähigen zu den leistungsfähigeren Rechnersystemen aufwärts (Aufwärtskompatibilität, Aufwärtsportabilität).

In den Mitgliedsländern des RGW werden die Rechnerfamilien EC (kyrillisch für Einheitliches System der Elektronischen Rechenteknik, ESER) sowie CM produziert und eingesetzt. Gegenwärtig wird die *Reihe 3* der EC-Familie produziert. Modelle sind die Großrechenanlagen EC 1036, EC 1046, EC 1066 (Sowjetunion) und EC 1057 (DDR), Kleinrechenanlagen EC 1007 und Personalcomputer EC 1834.

Die Familie CM (kyrillisch für System der Kleinrechenteknik, SKR) umfaßt die in den verschiedenen Ländern des RGW produzierten Kleinrechenanlagen. Gegenwärtig

<sup>1</sup> Komplexprogramm des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW bis zum Jahre 2000. In: Einheit, Berlin 41 (1986) 2, S. 167.



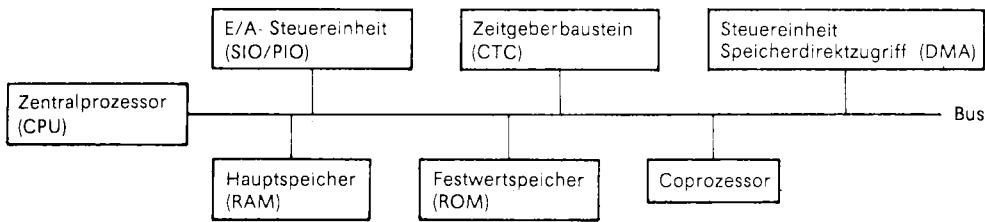


Abbildung 2.4.

Grundstruktur der Zentraleinheit eines Personalcomputers (SIO – Seriell Input Output, PIO – Parallel Input Output, CTC – Counter/Timer Circuit, DMA – Direct Memory Access, CPU – Central Processing Unit, RAM – Random Access Memory, ROM – Read Only Memory)

tig produzierte CM-Modelle sind z. B. CM 1300, CM 1800 für die Steuerung flexibler Fertigungssysteme sowie die Modelle CM 1410 und die 32-Bit-Superkleinrechenanlagen CM 1700 für die Steuerung von technologischen Komplexen und von Fertigungsbereichen. Zur CM-Familie gehört der Arbeitsplatzcomputer A 7150 des VEB Kombinat Robotron.

## 2.2. Elemente von Rechnersystemen

### 2.2.1. Zentraleinheit

#### 2.2.1.1. Bussystem

Die verschiedenen Funktionseinheiten und Baugruppen der Zentraleinheit sind durch Leitungen miteinander verbunden. Dabei kann es sich um Einzelleitungen zwischen den Einheiten handeln, oder es existiert eine Sammelleitung, an die die einzelnen Baugruppen angeschlossen werden.

**Eine Sammelleitung, die von den Funktionseinheiten einer Zentraleinheit gemeinsam genutzt werden kann, heißt Bus.**

Die Verwendung eines Busses ist typisch für Personalcomputer, deren Zentraleinheit z. B. entsprechend Abbildung 2.4. strukturiert ist.

Über den Bus werden Daten, Adressen und Steuerinformation transportiert. Die Daten sind Maschinenbefehle oder Verarbeitungsinformation. Adressen sind Positionsangaben über (nächste) Befehle, Verarbeitungsinformation oder periphere Geräte (vgl. Kapitel 4.). Steuerinformation dient der Kennzeichnung von Übertragungsoperationen und der Synchronisation.

Die Übertragung von Daten, Adressen und Steuerinformation kann seriell im zeitlichen Wechsel über eine Sammelleitung oder parallel über mehrere getrennte Sammelleitungen erfolgen. Bei paralleler Übertragung auf getrennten Leitungen wird zwischen Daten-, Adreß- und Steuerbus unterschieden.

Die Anzahl der auf einem Bus geführten Leitungen (Busbreite) und die Frequenz der Übertragung kennzeichnen die Leistungsfähigkeit eines Busses, der die Leistungsfähigkeit des Personalcomputers erheblich beeinflusst.

Rechnersysteme verwirklichen ein Modulkonzept, um verschiedene Ausbaustufen durch Zusammenschalten von Baugruppen zu ermöglichen. Die Größe des Busses richtet sich dabei nach der Leistung des Mikroprozessors und nach der Ausstattung (der Konfiguration) des Rechnersystems. Auch der Bus ist erweiterbar.

Der Bus verbindet alle Baugruppen über eine einheitliche Schnittstelle. Damit ist

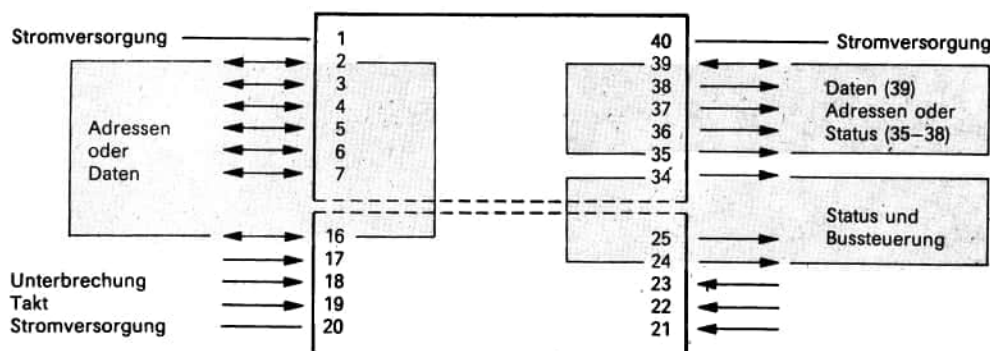


Abbildung 2.5.

Anschlußbild (Teildarstellung) des Prozessors K 1810/WM 86

eine beliebige Aneinanderreihung der Baugruppen möglich. Die Beschränkung der Baugruppenkopplung auf ein einziges Transportsystem senkt die Baugruppenkosten, reduziert aber die Arbeitsgeschwindigkeit im Vergleich zu konventionellen Realisierungsformen von Rechnern, bei denen eine größere Zahl von Kommunikationswegen zur Verfügung steht. Die geringere Arbeitsgeschwindigkeit resultiert daraus, daß zu jedem Zeitpunkt immer nur eine Kommunikation ausgeführt werden kann.

### 2.2.1.2.

#### Zentralprozessor

Das Kernstück des Rechnersystems ist der Zentralprozessor.

**Der Zentralprozessor (CPU) ist eine Funktionseinheit zur Ausführung von Befehlen, die durch ein Programm festgelegt sind.**

Der Zentralprozessor führt die durch ein Programm festgelegten Befehle Schritt für Schritt aus und steuert die Arbeit des gesamten Rechnersystems. Er ist in der Regel technisch auf nur einem Chip (Schaltkreis) realisiert (Einchipprozessor). Die Kommunikation eines Einchipprozessors mit anderen Funktionseinheiten erfolgt über An-

schlußstellen (Pins). Der Prozessor erhält über Pins Information, wandelt diese Information entsprechend der internen Logik um und stellt das Ergebnis an anderen Pins (oder an denselben Pins in einem anderen Takt) bereit (vgl. Abbildung 2.5.).

Der Funktionsablauf ist an ein Taktsignal gebunden, das von einem speziellen Schaltkreis, dem Zeitgeberbaustein (CTC) geliefert wird (Pin 19). Die Häufigkeit des Taktsignals pro Sekunde wird in MHz gemessen (1 Megahertz = 1 Million Taktsignale pro Sekunde). Die Periodendauer eines Taktes oder Taktzyklus ist die kleinste Zeiteinheit für alle im Prozessor ablaufenden Prozesse.

Ein Maschinenbefehl benötigt durchschnittlich 10 bis 20 Taktzyklen. Aus Taktzyklen pro Sekunde und Taktzyklen je Maschinenbefehl läßt sich der mittlere Zeitverbrauch für eine Operation berechnen und zur Charakterisierung des Prozessors verwenden.

#### Verarbeitungsbreite und Adreßbreite

Über die Pins 2 bis 16 und 39 (16 Anschlüsse) werden Daten mit anderen Funktionseinheiten ausgetauscht. Damit können in einem Arbeitstakt 16 Bit übermittelt werden. Die Anzahl der Bit, die in einem Arbeitstakt parallel übermittelt werden kön-

nen, wird als Verarbeitungsbreite bezeichnet. Mögliche Verarbeitungsbreiten für Mikroprozessoren sind 4, 8, 16 und 32 Bit. Sie beeinflussen die Anzahl der Elementaroperationen, in die eine Verarbeitungsaufgabe zerlegt werden muß. Je kleiner die Verarbeitungsbreite ist, um so mehr Operationen werden erforderlich, wenn die Verarbeitungsinformation die vom Prozessor realisierte Breite überschreitet.

Sind auch die anderen Parameter des Prozessors auf die Verarbeitungsbreite von 16 Bit abgestimmt, so handelt es sich um einen 16-Bit-Prozessor. Ein 16-Bit-Prozessor, z. B. der Prozessor K 1810/WM 86, benötigt zum Lesen von Daten nur die Hälfte der Taktzyklen eines 8-Bit-Prozessors. Allerdings gibt es zwischen 8- und 16-Bit-Prozessoren Zwischenformen, weil technisch angestrebt wird, die Anzahl der Pins gering und den Bus einfach zu halten. Die Klassifikation ist dann nicht eindeutig oder hat keinen eindeutigen Bezug zur Leistung. Beispiele sind Prozessoren, die für 16-Bit-Verarbeitungsbreite ausgelegt sind, aber nur Anschlüsse für Verarbeitungsinformation mit einer Breite von 8 Bit besitzen. Die Pins 2 bis 16 und 39 des Prozessors K 1810/WM 86 werden auch benutzt, um Adreßinformation aufzunehmen oder abzugeben. Ob der Prozessor auf Verarbeitungsinformation oder auf Adreßinformation erkennt, hängt vom Takt und von der Statusinformation ab, die an anderen Pins zu dieser Zeit anliegt.

Für das Senden einer Adresse stehen zusätzlich die Pins 35 bis 38 zur Verfügung. Das sind insgesamt 20 Pins (2–16, 35–39) und damit 20 Bit parallel. So erklärt sich die *Adreßbreite*, mit der  $2^{20} = 1$  MByte adressiert werden können.

### Struktur

Die Bestandteile des Prozessors sind Steuerwerk, Rechenwerk und Register. Das *Steuerwerk* holt Maschinenbefehle vom Hauptspeicher, interpretiert sie, sorgt für die Bereitstellung der erforderlichen Information in schnellen, kleinen Zwischenspeichern des Prozessors, den *Registern*, und aktiviert das Mikroprogramm.

**Das Mikroprogramm ist eine Menge hardwaremäßig realisierter Grundoperationen, durch die Maschinenbefehle umgesetzt werden.**

Das Mikroprogramm befindet sich im allgemeinen auf einem speziellen Festwertspeicher, dem Mikroprogrammspeicher, der vom Gerätehersteller fest mit dem Gerät verbunden ist. Kann der Mikroprogrammspeicher durch den Anwender verändert werden, wird das Rechnersystem als mikroprogrammierbar bezeichnet.

Sehen die Maschinenbefehle eine Verknüpfung von Bitmustern vor, z. B. bei arithmetischen Operationen, dann wird das *Rechenwerk* (Arithmetical Logical Unit, ALU) aktiv. Die Operanden der Verknüpfung und die Ergebnisse werden in den Registern bereitgestellt.

### Leistungsparameter

Mikroprozessoren unterliegen einer schnellen Entwicklung, wobei auch weniger leistungsfähige für bestimmte Aufgaben ihre Bedeutung behalten. Die Tabelle 2.2. zeigt einige Mikroprozessoren mit ausgewählten Leistungsparametern.

Der Mikroprozessor U 880 ist der Zentralprozessor des Bürocomputers A 5120/30 sowie des Personalcomputers PC 1715. Der erweiterte Bürocomputer A 5120.16 ist mit dem Prozessor U 8000, die Arbeitsplatzcomputer A 7100 und A 7150 sowie der Personalcomputer EC 1834 sind mit dem Prozessor K 1810/WM 86 ausgestattet.

Tabelle 2.2.  
Ausgewählte Mikroprozessoren und Parameter

Parameter	Mikroprozessor			
	U 808	U 880	U 8000	K 1810/WM 86
Verarbeitungsbreite (bitparallel)	8	8	16	16
Adressierbarer Speicherbereich	64 KByte	64 KByte	8 MByte	1 MByte
Taktfrequenz (MHz)	0,5	2,5	4,0	4,9
Operationszeit (Mikrosekunden/Operation)	2–8,5	1,6–9,6	0,75–9,0	0,4–10,2

Rechnersysteme werden zunehmend mit *Coprozessoren* für spezielle Aufgaben, z. B. die Ein- und Ausgabe, die Grafik oder für arithmetische Operationen, ausgestattet. Ein Coprozessor für arithmetische Operationen (Arithmetikprozessor) kann die Rechnerleistung bis zum Hundertfachen erhöhen.

### 2.2.1.3. Hauptspeicher

Der Hauptspeicher ist zur Ablage von Programmen und Verarbeitungsinformation erforderlich.

**Der Hauptspeicher ist eine Funktionseinheit, die Information aufnehmen, aufbewahren und auf Anforderung wieder bereitstellen kann.**

#### Speicherzellen und Adressen

Der Hauptspeicher besteht aus Speicherzellen. Jede Speicherzelle trägt eine Nummer, über die zum Inhalt der Zelle zugegriffen werden kann (vgl. Abbildung 2.6.). Für die Nummer einer Speicherzelle wird die Bezeichnung *Speicheradresse* verwendet (es gibt außerdem Geräteadressen zur Kommunikation zwischen Funktionseinheiten).

**Speicheradressen sind Nummern von Speicherzellen, über die zum Inhalt der Speicherzelle zugegriffen werden kann.**

Im Hauptspeicher besitzt nicht jedes einzelne Bit eine Speicheradresse. In den Speichern von Personalcomputern sind Gruppen von 8 Bit (ein Byte) adressierbar. In den Speichern von Klein- und Großrechenanlagen sind es 16 Bit (ein Wort, wobei unter einem Wort gegebenenfalls auch die Breite verstanden wird, die der Prozessor in einem Zugriff maximal verarbeiten kann). Diese Systeme heißen byte- oder wortorientiert.

Die Größe des Hauptspeichers ist die *Hauptspeicherkapazität*. Sie wird in einem Vielfachen ihrer Einheit, im allgemeinen aber in Byte angegeben. Dabei gelten folgende binäre Vielfache:

$$\begin{aligned} K &= 2^{10} = 1024 \\ M &= 2^{20} = 1048576 \\ G &= 2^{30} = 1073741824 \end{aligned}$$

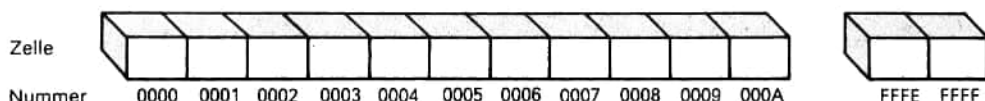


Abbildung 2.6.  
Hauptspeicher mit 65536 (FFFFF) Speicherzellen

In Anlehnung an Standardbezeichnungen zur Basis 10 wird von Kilo, Mega und Giga gesprochen. 512 KByte (Kilobyte) entsprechen also  $512 \cdot 2^{10} = 524\,288$  Byte.

Die Hauptspeicherkapazität ist aus technisch-ökonomischen Gründen relativ eng begrenzt. Hauptspeicher nehmen jeweils nur die gerade zu bearbeitenden Programme und die Verarbeitungsinformation auf. Für die Nachschub- und Archivspeicherung werden externe Speicher genutzt.

### Speichertechnologien

Interne Speicher sind in der Regel Halbleiterspeicher. Nach grundlegender Ausführung und Funktion können die in Abbildung 2.7. dargestellten Speicher unterschieden werden.

Bei *Lese-Schreib-Speichern* (RAM) wird die Information ab einem vorbestimmten, adressierten Speicherplatz geschrieben und bleibt dort erhalten, bis sie mit anderer Information überschrieben wird. Die gespeicherte Information kann unter Angabe der (Anfangs-)Adresse beliebig oft gelesen (und damit z. B. auch kopiert) werden, ohne daß sie zerstört wird.

Hauptspeicher sind stets RAM-Speicher, da ständig geschrieben und gelesen werden muß. Die Speicherzellen in RAM-Schaltkreisen behalten ihren Inhalt nur dann, wenn Betriebsspannung anliegt. Die Information ist also auch nach kurzzeitigem Stromausfall nicht mehr erhalten. Auf dynamischen RAM (im Gegensatz zu statischen RAM) ist die Information sogar so schnell flüchtig (etwa 1 ms), daß während des Betriebs ein ständiges Auffrischen (Refresh) der Information erfolgen muß. Zur Sicherung des Datenerhaltes, z. B. für die interne Uhrzeit- und Datumsangabe, wird in manchen Geräten zusätzlich eine Betriebsspannung für Teile des RAM über Batterien gewährleistet.

*Nur-Lese-Speicher* (ROM) ermöglichen

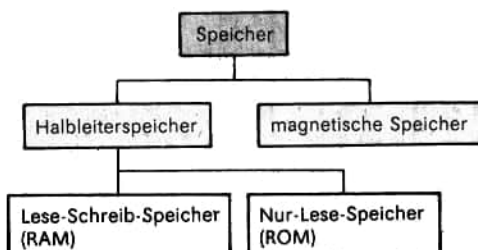


Abbildung 2.7.

Speicherklassifikation nach der Funktion

nur Leseoperationen. Die Information bleibt auch beim Abschalten der Betriebsspannung erhalten. Dieser Speichertyp ist ausschließlich für feststehende Inhalte einsetzbar. In der ROM-Technologie sind der Mikroprogrammspeicher, der Anfangslader (vgl. Kapitel 3.) und die Zeichengeneratoren (vgl. Abschnitt 1.3.2.) ausgeführt. Es gibt auch Betriebssysteme und Compiler, die auf ROM-Speichern realisiert sind.

Der Vorteil der ROM-Technologie liegt in der Beständigkeit der Speicherinhalte, der hohen Sicherheit vor Zerstörung und in der Zugriffszeit. Es gibt ROM-Speicher mit unterschiedlichen Eigenschaften bezüglich der Programmierung (vgl. Tabelle 2.3.).

Das Schreiben von ROM benötigt bedeutend mehr Zeit als das Schreiben von RAM. Deshalb ist für das Schreiben und für das Löschen spezielle Gerätetechnik erforderlich.

### Kenngrößen

Kenngrößen von Speichern sind vor allem Speicherkapazität, Zugriffszeit und Aufrufbreite (Speicherwortbreite).

Die RAM-Schaltkreise werden mit 64 KBit, 256 KBit, 1 MBit und 4 MBit Kapazität produziert. Sie werden entsprechend der vorgesehenen Konfiguration auf Steckkarten zusammengestellt. Gebräuchliche RAM-Kapazitäten für 8-Bit-PC sind 64, 128 oder 256 KByte. 16-Bit-PC verfügen häufig über eine Kapazität von 512 KByte oder 640 KByte.

Tabelle 2.3.  
ROM-Varianten

Bezeichnung	Eigenschaften
ROM	Irreversibler Festwertspeicher, Programmierung einmalig durch den Hersteller.
PROM (programmable ROM)	Irreversibler Festwertspeicher, Programmierung einmalig durch den Anwender.
EPROM (erasable PROM)	Reversibler Festwertspeicher, Programmierung (mehrmals) durch Hersteller oder Anwender, Löschen (gesamt) durch UV-Bestrahlung.
EEPROM (electrically PROM) oder EAROM (electrically alterable ROM)	Reversibler Festwertspeicher, Programmierung (mehrmals) durch Hersteller oder Anwender, Löschen (bitweise) durch elektrische Impulse.

Die Zugriffszeit ist die Zeitdauer zwischen dem Aufruf der Adresse und der Beendigung des Lese- oder Schreibvorganges. Sie ist abhängig von der Speichertechnologie, z. B. MOS-Technik (Metall-Oxid Semiconductor, MOS), Bipolartechnik, und der Arbeitsweise der Speicherschaltkreise (statisch, dynamisch). Die Zugriffszeiten liegen meist im Bereich von Nanosekunden (ns). Alle (internen) RAM und ROM ermöglichen wahlfreien Zugriff im Gegensatz zu einigen externen Speichern.

### 2.2.1.4.

#### Weitere Bestandteile

##### Anschlußsteuerungen

Anschlußsteuerungen dienen der Kommunikation der Zentraleinheit mit peripheren Geräten, mit anderen Rechnersystemen oder mit peripheren Speichern. Für die Steuerung dieser Kommunikation gibt es verschiedene Funktionseinheiten. Dazu gehören die Ein- und Ausgabesteuereinheit sowie die Steuereinheit für den Speicherdirektzugriff (vgl. Abbildung 2.4.). Die grundlegende Aufgabe besteht darin, die Information vom Bus zu nehmen oder auf den Bus zu bringen und für die Anpassung an die peripheren Geräte zu sorgen.

Die *E/A-Steuereinheit* stellt die Verbindung zwischen Zentraleinheit und peripheren Geräten, wie Tastatur, Bildschirm, Drucker, oder anderen Rechnersystemen her. Der Anschluß typischer Peripherie ist über Anschlußbilder standardisiert.

**Ein Anschlußbild (Interface, Schnittstelle) ist die Gesamtheit der logischen und konstruktiven Vorschriften für die Kommunikation zwischen zwei Systemen.**

Der Begriff »Anschlußbild« wird im übertragenen Sinne mit »Nutzerinterface« oder »Nutzerschnittstelle« auch für die Mensch-Maschine-Kommunikation angewendet. E/A(Eingabe/Ausgabe)-Interfaces sind in der Regel national oder international standardisierte Anschlußbilder für Leitungen mit bestimmten elektrischen Kenngrößen, vorgeschriebenen Übertragungsabläufen (Übertragungsprozeduren, Übertragungsprotokollen) und festgelegter konstruktiver Ausführung. Die E/A-Interfaces liegen zwischen den E/A-Steuereinheiten als Bestandteile der Zentraleinheit und den speziellen E/A-Geräten. Somit ist es möglich, unterschiedliche Geräte an die Zentraleinheit anzuschließen, wenn die passenden Interfaces vorhanden sind. Die technische

Realisierung einer E/A-Steuereinheit kann aus der Zwischenspeicherung von Information in Registern bestehen oder bis zum Einsatz von E/A-Prozessoren mit Programmspeicher für die E/A-Steuerung reichen.

Nach der Art der Informationsübertragung werden verschiedene E/A-Interfacetypen unterschieden. Die Haupttypen sind parallele und serielle Interfaces.

Über *parallele Interfaces* werden mehrere Bit gleichzeitig auf verschiedenen Leitungen transportiert (Parallel Input Output, PIO). Parallele Interfaces wirken wie eine direkte Verlängerung des Datenbusses zu einem E/A-Gerät. Die parallele Übertragung ist schnell, stellt aber an das Übertragungsmedium höhere Anforderungen (Leitungsbündel).

Über *serielle Interfaces* werden die Bit nacheinander übertragen (Seriell Input Output, SIO). Dazu wird nur eine Leitung benötigt. Serielle Interfaces werden vor allem zur Überbrückung größerer Entfernungen eingesetzt. Ihr Vorteil sind einfache Übertragungsmedien, ihr Nachteil ist die geringere Übertragungsrate. Die Geschwindigkeit wird auf den langsameren Kommunikationspartner ausgerichtet und ist durch den Übertragungsweg begrenzt. Es entsteht eine erhebliche Verzögerung. Trotzdem wird z. B. vom SIO-Baustein U 856, der zusammen mit dem Prozessor

U 880 eingesetzt wird, eine Rate von 500 KBit pro Sekunde erreicht. Wichtige standardisierte Interfaces enthält Tabelle 2.4.

**Speicherdirektzugriff**

Eine spezielle Aufgabe erfüllt die Steuereinheit für den Speicherdirektzugriff (DMA). Die Steuereinheit ist in der Lage, Information völlig selbständig, d. h. ohne Steuerung durch den Zentralprozessor, zu übertragen. Der Zentralprozessor teilt dem DMA-Prozessor (DMA-Steuereinheit) mit, wieviel Byte von welcher Adresse zu welcher Adresse zu übertragen sind (es kann sich um Hauptspeicher- oder Geräteadressen handeln). Der DMA-Prozessor realisiert die Übertragung und teilt danach dem Zentralprozessor die Beendigung seiner Tätigkeit mit. Der Zentralprozessor kann inzwischen andere Aufgaben (ohne Zugriff auf den Speicher und die adressierten Geräte) realisieren.

**Unterbrechungssystem**

Das Unterbrechungssystem beeinflusst die Leistungsparameter eines Rechnersystems erheblich.

**Das Unterbrechungssystem ist die Gesamtheit der Einrichtungen zur Unterbrechung eines laufenden Programms.**

Tabelle 2.4.  
Standardinterfaces

Bézeichnung	Beschreibung
V.24	(S2) Serielles Interface für Geräte der seriellen Informationsübertragung sowie für die Fernübertragung (TGL 29 077).
IFSS	Seriell Interface für die Informationsübertragung im Nahbereich (bis 20 m) nach TGL 42 886.
IFSP-M	(CENTRONICS, nur Ausgabe) paralleles Interface (byteparallel) für Druckeranschluß (TGL in Vorbereitung).

Tabelle 2.5.  
Übersicht über E/A-Techniken (Auswahl)

EA-Technik	Eingabegeräte	Ausgabegeräte
Lochstreifentechnik	Lochstreifenleser	Lochstreifenstanzer
Lochkartentechnik	Lochkartenleser	Lochkartenstanzer
Bildschirmtechnik	Tastatur, Lichtstift, Maus, Joystick	Bildschirm
Drucktechnik	—	Zeichen-, Zeilen-, Seitendrucker
Optische Lesetechnik	Markierungsleser, Strich- codeleser, Klarschriftleser	—
Magnetkartentechnik	Magnetkartenleser	Magnetkartenschreiber
Sprach-E/A-Technik	Spracheingabegerät, Funkempfänger	Sprachausgabegerät, Funksender

Unterbrechungen haben zur Folge, daß ein Spezialprogramm zur Unterbrechungsbehandlung gestartet, ausgeführt und danach das unterbrochene Programm fortgesetzt wird. Unterbrechungserfordernisse ergeben sich bei jeder Ein- und Ausgabe, bei der zeitgeteilten Verarbeitung mehrerer Programme (vgl. Abschnitt 3.1.2.), bei der Steuerung durch ein Betriebssystem und bei Fehlern. Unterbrechungen, die durch prozessorexterne Vorgänge ausgelöst werden, heißen *Interrupt*. Für ihre Behandlung stehen als Spezialprogramme Interruptbehandlungsroutinen zur Verfügung, die je nach Art der Unterbrechung über einen Adreßvektor (Interruptvektor) aufgerufen werden.

Unterbrechungen durch prozessorinterne Vorgänge (z. B. Nichterkennung eines Maschinenbefehls) werden als *Trap* bezeichnet. Zur Reaktion auf Traps stehen Trapbehandlungsroutinen zur Verfügung, die im Mikroprogrammspeicher abgelegt sind.

## 2.2.2. Periphere Geräte zur Ein- und Ausgabe

### 2.2.2.1. Ein- und Ausgabetechniken

Die Ein- und Ausgabe dient der Kommunikation eines Rechnersystems mit der Umgebung, vor allem mit dem Menschen. Aus der Sicht einer Zentraleinheit ist auch das Zwischenspeichern von Information auf peripheren Speichern Ein- oder Ausgabe (vgl. Abschnitt 2.2.3.).

Die Tabelle 2.5. gibt einen Überblick über Ein- und Ausgabetechniken von Rechnersystemen.

Die Ausstattung mit peripheren Geräten zur Ein- und Ausgabe beeinflußt die Leistungsfähigkeit eines Rechnersystems erheblich. Sie ist der jeweiligen Aufgabe und dem Basissystem angepaßt.

### 2.2.2.2. Lochstreifen- und Lochkartentechnik

#### Lochstreifentechnik

Lochstreifentechnik ist E/A-Technik zur Verarbeitung von Lochstreifen (Papier,



Plastfolie), auf denen Bitmuster für Information in Form von Lochungen (Bit gesetzt) und Nichtlochungen (Bit nicht gesetzt) dargestellt sind.

Ein Lochstreifen ist in Sprossen und Spuren mit genormten Abständen aufgeteilt (vgl. Abbildung 2.8.).

Für die Zuordnung des Bitmusters einer Sprosse zu einem Zeichen existieren Lochstreifencodes.

Vorteil der Lochstreifentechnik ist die Resistenz der gespeicherten Information gegenüber Temperatur, Luftfeuchtigkeit und magnetischem Einfluß. Nachteilig sind der hohe Materialaufwand für den Datenträger (nur einmalige Verwendung) und die visuelle Nichtlesbarkeit. Außerdem sind Einfügungen und Korrekturen unmöglich oder sehr aufwendig.

Haupteinsatzgebiete der Lochstreifentechnik sind die Steuerung von Maschinen und Produktionsanlagen.

### Lochkartentechnik

Der Lochkartentechnik liegt ein ähnliches Prinzip zugrunde wie der Lochstreifentechnik.

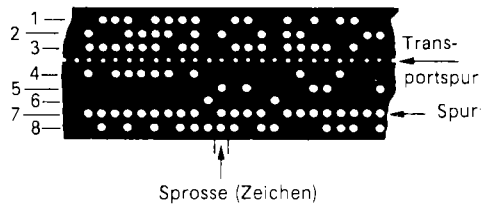


Abbildung 2.8.  
Lochstreifen

**Lochkartentechnik ist E/A-Technik zur Verarbeitung von Lochkarten, auf denen Bitmuster für Information in Form von Lochungen (Bit gesetzt) oder Nichtlochungen (Bit nicht gesetzt) in Zeilen und Spalten dargestellt sind.**

Zusätzlich kann der Inhalt in einer Klarschriftzeile visuell lesbar gemacht und die gesamte Lochkarte wie ein Formular bedruckt sein (vgl. Abbildung 2.9.).

Die Zuordnung der Lochungen zu Zeichen richtet sich nach einem Lochkartencode. Vorteile der Lochkartentechnik sind die maschinelle und visuelle Lesbarkeit der Datenträger sowie die einfache Korrektur-, Misch- und Sortierfähigkeit. Nachteile sind der hohe Materialverbrauch (einmalige

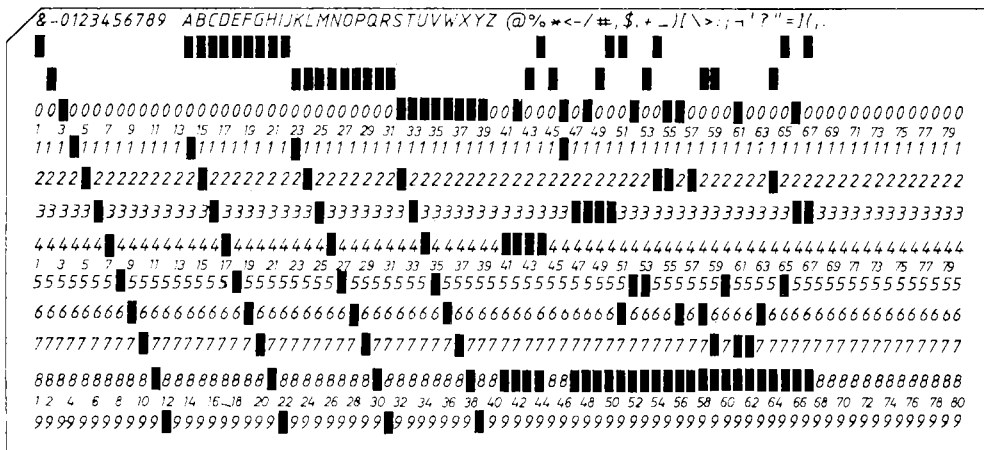


Abbildung 2.9.  
Lochkarte



Abbildung 2.10.  
Tastatur des EC 1834

Verwendbarkeit des Datenträgers) und die Fehleranfälligkeit bezüglich der Reihenfolge der gespeicherten Information. Haupteinsatzgebiete der Lochkartentechnik sind die Kommandosteuerung von Großrechenanlagen und einige Dispositionsprozesse im Ein- und Verkauf sowie in der Lagerwirtschaft. Der Einsatz von Lochstreifen- und Lochkartentechnik ist wegen des Materialaufwandes und des Einsatzes von magnetischen Datenträgern rückläufig.

### 2.2.2.3. Bildschirmtechnik

**Bildschirmtechnik ist E/A-Technik, bei der Information über die Tastatur und Zusatzeingabegeräte eingegeben und über den Bildschirm ausgegeben wird.**

Bildschirmtechnik ist gegenwärtig die Hauptform der Mensch-Maschine-Kommunikation.

#### Tastatur

Das wichtigste Eingabegerät der Bildschirmtechnik ist die Tastatur.

**Die Tastatur ist ein Gerät zur taktilen Informationsübertragung vom Menschen zur Maschine.**

Eine Tastatur besteht aus Tasten, deren Anordnung und Wirkung genormt sind (vgl. Abbildung 2.10.).

Beim Niederdrücken einer Taste wird das zu dieser Taste gehörende Bitmuster zur Zentraleinheit übertragen. Im allgemeinen existiert eine separate Anschlußsteuereinheit für die Tastatur, die auch aus einem gesonderten Prozessor bestehen kann. Es können folgende Tastaturfelder unterschieden werden:

- *alphanumerisches Tastenfeld* mit Buchstaben, Ziffern und Sonderzeichen, übereinstimmend mit dem Zeichensatz (vgl. Abschnitt 1.2.2.). Zwischen Groß- und Kleinbuchstaben kann durch spezielle Tasten zeitweise (SHIFT) oder dauerhaft (CAPS-LOCK) umgeschaltet werden. Die Anordnung der Tasten entspricht etwa der Anordnung von Tasten einer Schreibmaschinentastatur;
- *numerisches Tastenfeld* mit den Ziffern 0 bis 9, Doppelnull, Komma und Minuszeichen;

- *Sondertasten*, z. B. die Tasten zur Bewegung des Cursors, CTRL (erzeugt zusammen mit einer anderen Taste Steuerzeichen), ESC (sendet das Steuerzeichen ESC), ALT (schaltet zwischen Zeichensätzen um), CL (schaltet den Akustikgeber/Tastenklick ein oder aus) und ENTER bzw. RETURN oder ET (sendet die Steuerzeichenfolge CR, LF als Endemitteilung einer Eingabe);
- *Funktionstasten*, die sich mit Zeichenfolgen belegen lassen. Die Zeichenfolgen werden gesendet, wenn die entsprechende Funktionstaste gedrückt wird.

Es gibt eine Vielzahl von Tastaturen, so daß die Symbole auf den Tasten und auch die Anzahl der Funktionstasten verschieden sein können.

### Bildschirm

Das Ausgabegerät der Bildschirmtechnik ist der Bildschirm.

**Der Bildschirm (Display, Monitor) ist ein Gerät zur alphanumerischen oder grafischen Anzeige von Information durch aktivierte Bildpunkte (Pixel).**

Charakteristische Parameter eines Bildschirms sind Bildfeldgröße (Höhe, Breite, Diagonale), Auflösungsvermögen (Anzahl aktivierbarer Bildpunkte), Bildwiederholfrequenz, Nachleuchtdauer (Persistenz), Bildverzerrung (horizontal und vertikal), Kontrast und Farbattribut (monochrom für einfarbig, color für mehrfarbig).

Jedes auf dem Bildschirm dargestellte Zeichen ist zugleich intern in einem *Bildwiederholpeicher* (gegebenenfalls zusammen mit einem Darstellungsattribut) gespeichert. Der Inhalt des Bildwiederholpeichers wird zyklisch gelesen und angezeigt. Über die Tastatur können Zeichen in den Bildwiederholpeicher eingelegt oder aus ihm entfernt werden. Auch

ist es möglich, vorhandene Zeichen zu überschreiben. Das geschieht an der Stelle, die durch eine Schreibmarke, den *Cursor*, für den Nutzer sichtbar ist. Die Änderung wird nach dem nächsten Bildwiederholzyklus sichtbar. Mit Hilfe spezieller Tasten (Cursortasten) kann die Position des Cursors verändert werden, ohne den Inhalt des Bildwiederholpeichers zu verändern.

Für Bildschirme werden zwei Betriebsmodi unterschieden:

- *Text-Modus*. Angezeigt werden die Zeichen eines Zeichensatzes. In speziellen Zeichensätzen können auch einige (Quasi-)Grafikzeichen enthalten sein, aus denen elementare Grafiken formiert werden können. Außer bei sehr einfachen Bildschirmen wird jedes Zeichen intern durch zwei Byte repräsentiert. Das eine charakterisiert das Zeichen selbst, z.B. »A«, »1«, als Repräsentant der vom Zeichengenerator zu aktivierenden Bildpunkte. Das andere Byte charakterisiert die Art, das Attribut, der Darstellung. Attribute bei Verwendung einer Monochromkarte als Bildschirmsteuereinheit sind invers, intensiv und blinkend. Bei Verwendung einer Farbgrafikkarte (Color Graphic Adapter, CGA) als Bildschirmsteuereinheit sind das Farbattribut (z. B. 16 Vordergrund-, 8 Hintergrundfarben) oder Helligkeitsstufen.

- *Grafik-Modus*. Angezeigt werden einzelne Pixel (picture elements), die sich auch einzeln ansteuern lassen. Alle Pixel werden z. B. bei mittlerer Auflösung intern durch zwei Bit repräsentiert. Das eine Bit kennzeichnet die Aktivierung, das zweite Bit das Attribut.

Im Text-Modus wird der Bildschirm in Zeilen und Spalten für die Schrift- oder Quasigrafikzeichen aufgeteilt. Typisch sind 25 Zeilen und 80 Spalten. Der Bildwiederholpeicher benötigt in diesem Falle

$25 * 80 * 2 = 4000$  Byte (technisch realisiert mit 4 KByte). Im Grafikmodus ist dagegen ein weitaus größerer Speicherplatz für die Aufnahme der Bildpunkte erforderlich. Deshalb werden als Anschlußsteuereinheiten Bildschirmsteuereinheiten (gesonderte Prozessoren als Steckkarten) mit eigenem Bildwiederholpeicher eingesetzt, so daß der Hauptspeicher nicht belastet wird.

### Zusatzeingabegeräte

Die Mensch-Maschine-Kommunikation läuft in vielen Fällen darauf hinaus, daß auf dem Bildschirm ein Funktionsangebot erfolgt und daraus eine Funktion auszuwählen ist. Die Auswahl kann erfolgen durch Eingabe eines Funktionscodes (Zeichen, Zeichenfolge) oder durch Bewegung des Cursors auf die Bildschirmposition, die die gewünschte Funktion anzeigt. Die Cursorpositionierung für die Auswahl kann schneller und bequemer durch einen Lichtstift oder eine Maus erfolgen.

**Ein Lichtstift (light pen) ist ein Zusatzeingabegerät, mit dem Punkte auf der Bildschirmoberfläche markiert oder Linien dargestellt werden können.**

Der Lichtstift arbeitet mit einem fotoelektronischen Bauelement, das bei der Berührung mit der Bildschirmoberfläche den Abtaststrahl der Bildröhre aufnimmt und so die Berührungsposition über ein Kabel zur Zentraleinheit überträgt.

**Eine Maus (mouse) ist ein Zusatzeingabegerät, durch dessen Bewegung auf einer festen Unterlage der Cursor (im allgemeinen in Form eines Pfeiles) auf dem Bildschirm bewegt und durch einen Auslösemechanismus als Punkt markiert werden kann.**

Die Bewegung der Maus wird von einer Rollkugel, die sich durch den Druck auf die

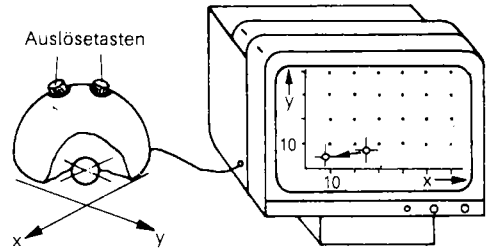


Abbildung 2.11.

Bildschirm mit Maus-Zusatzeinrichtung

Unterlage bewegt, über ein Kabel zur Zentraleinheit übertragen (vgl. Abbildung 2.11.) und dort in die Cursorbewegung umgesetzt. Die Betätigung des Auslösemechanismus durch ein oder zwei Tasten auf der Maus (*Klicken*) wird dann als Auswahl der Position aufgefaßt, an der sich der Cursor befindet.

Zur Unterstützung von CAD gibt es außer Lichtstift und Maus Digitizer (Digitalisiertabletts, Digitalisierungsgeräte). Sie dienen der direkten Eingabe (Digitalisierung) von Koordinaten einer Zeichnung.

Die Hardwareergonomie wird durch solche Zusatzgeräte zur schnellen Cursorpositionierung erheblich verbessert. Voraussetzung ist natürlich, daß die Software diese Technik unterstützt.

### 2.2.2.4. Drucktechnik

Das Hauptausgabegerät bei der Mensch-Maschine-Kommunikation ist der Bildschirm. Als Ergebnis der Arbeit ist jedoch auch häufig eine Ausgabe auf Papier erforderlich.

**Drucktechnik ist E/A-Technik zur Ausgabe von Information durch Kontrasterzeugung auf Papier.**

Die Ausgabe übernehmen Drucker (printer, vgl. Abbildung 2.12.).

Für einige Drucker kann wie beim Bild-

schirm die Arbeit im Text- und im Grafik-Modus unterschieden werden. Im Text-Modus werden Zeichen eines Zeichensatzes (Buchstaben, Ziffern, Sonderzeichen, Quasigrafikzeichen) und im Grafik-Modus Bildpunkte gedruckt.

Drucker werden klassifiziert nach

- dem Ergebnis einer (einzelnen) Druckoperation in Zeichendrucker, Zeilendrucker und Seitendrucker;
- dem verwendeten Druckverfahren in Typenraddrucker, Nadeldrucker, Tintenstrahldrucker, Thermodrucker, elektrofotografische, elektrofotografische und magnetografische Drucker.

Wichtige Kennwerte für Drucker sind die Druckgeschwindigkeit (Zeichen pro Sekunde, Zeilen pro Minute, Seiten pro Minute), die maximale Druckbreite (Zeichen pro Zeile), die Druckqualität (Kontrast, Schärfe, Bildpunkte je Zoll), der Zeichenvorrat (Anzahl der Zeichensätze, Schriftarten, Grafikfähigkeit), die Technik des Papiertransports, der Geräuschpegel, der Energieverbrauch und die Bedienungs-freundlichkeit.

### Zeichendrucker

**Zeichendrucker (character printer) sind Drucker, die (seriell) Zeichen für Zeichen mit einer Geschwindigkeit zwischen 30 und 200 Zeichen pro Sekunde schreiben.**

Drucker am Arbeitsplatz sind vorwiegend Zeichendrucker. Die wichtigsten Zeichendrucker sind Typenraddrucker und Nadeldrucker. Das Druckprinzip eines Typenraddruckers zeigt Abbildung 2.13.

Das Typenrad kann auswechselbar sein. Typenraddrucker realisieren einen vollen Schriftzug in sehr guter Qualität. Sie sind aber nicht grafikfähig. Die Druckgeschwindigkeit beträgt bis zu 60 Zeichen pro Sekunde.

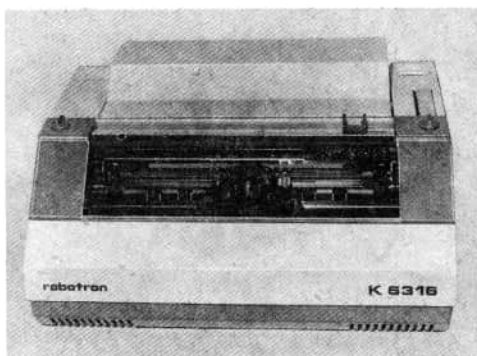


Abbildung 2.12.  
Drucker K 6316 des VEB Kombinat Robotron

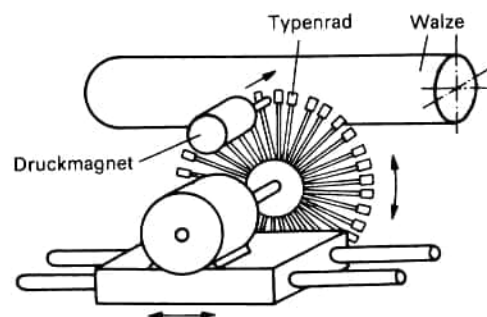
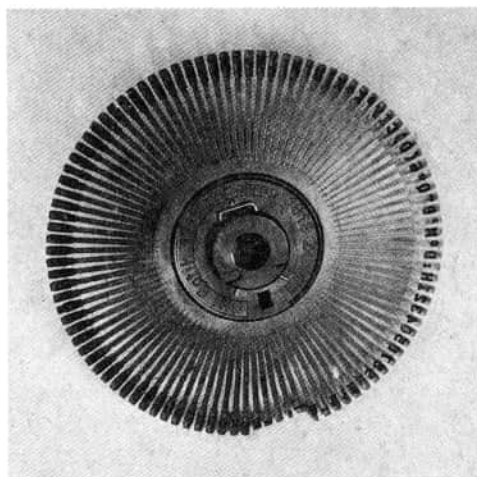


Abbildung 2.13.  
Typenraddruck

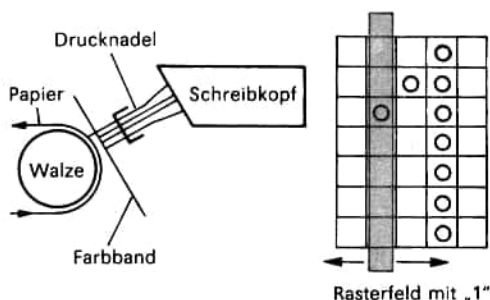


Abbildung 2.14.  
Nadeldruck

Das Druckprinzip der Nadeldrucker geht aus Abbildung 2.14. hervor.

Die Nadeldrucker haben bei etwas geringerer Druckqualität gegenüber den Typenraddruckern den Vorteil einer höheren Geschwindigkeit und größeren Flexibilität der Ausgabe von Zeichen. Bei Nadeldruckern kann der Nutzer den Zeichenvorrat um selbstdefinierte Zeichen erweitern. Nadeldrucker sind grafikfähig. Bei einigen Spezialausführungen kann die Geschwindigkeit von 200 Zeichen/Sekunde überschritten werden.

### Zeilendrucker

**Zeilendrucker (line printer, Paralleldrucker, Schnelldrucker) sind Drucker, die bei jeder Druckoperation alle Zeichen einer Zeile (parallel) mit einer Geschwindigkeit von 150 bis 4000 Zeilen pro Minute schreiben.**

Zeilendrucker sind im allgemeinen Bestandteil jeder Großrechenanlage. Bei Zeilendruckern können Matrixdrucker (Nadeldruck in Zeilenbreite), Trommeldrucker (Drucker mit Typenwalze) und Kettendrucker (Drucker mit einer Typenkette) unterschieden werden. Der Vorteil von Zeilendruckern liegt in ihrer hohen Geschwindigkeit. Der Nachteil von Trommel- und Kettendruckern ist die verminderte Druck-

qualität, vor allem bezüglich der Justierung von Zeichen auf der Linie (tanzende Zeichen).

### Seitendrucker

**Seitendrucker (page printer) sind Drucker, die bei jeder Druckoperation alle Zeichen einer Seite mit einer Geschwindigkeit von 20 bis 300 Seiten pro Minute drucken.**

Von besonderer Bedeutung sind *Laserdrucker*, die mit elektrostatischer Druckbildzeugung arbeiten. Dabei wird eine Trommel mit Selenoberfläche im Dunkeln positiv aufgeladen. Die Oberfläche ist danach nicht leitend. Durch Belichten der zu druckenden Flächen mit einem Laserstrahl (über einen Polygonspiegel) werden diese leitend, und die Oberflächenladung fließt ab. Die entladenen Flächen werden mit positiv geladenem Farbpulver geschwärzt. Dadurch entsteht ein Druckbild. Das Druckbild wird auf das negativ geladene Papier übertragen und durch Wärmebehandlung eingebrannt. Die Druckqualität ist sehr gut. Laserdrucker sind für die grafische Ausgabe geeignet.

### Plotter

Eine Sonderstellung unter den Druckern nehmen Plotter ein.

**Plotter (Kurvenschreiber) sind Ausgabegeräte zur (grafischen) Ausgabe von Information in Form von Kurven und Einzelpunkten in einem Koordinatensystem.**

Plotter dienen vor allem CAD-Anwendungen. Es können unterschiedliche Strichstärken, Farben und Schriftarten gewählt werden.

## 2.2.2.5.

**Optische Lesetechnik**

Mit optischer Lesetechnik werden handschriftliche Markierungen und Klartexte auf Belegen sowie gedruckte Strichsymbole oder OCR-Schrift (Optical Character Recognition, OCR) gelesen.

**Optische Lesetechnik ist Eingabetechnik, bei der Bereiche eines Trägermaterials optisch abgetastet und mit Hilfe eines Referenzmusters Zeichen zugeordnet werden.**

Die Abtastvorrichtungen optischer Leser erzeugen zunächst ein elektrisches Abbild in einem Grundraster. Je Rasterpunkt wird durch fotoelektronische Wandler (Fotозellen) zwischen Hell (0) und Dunkel (1) unterschieden. Diese Werte werden gespeichert (bei Erkennung von Handschriften können auch die Hell/Dunkel-Übergänge und damit die Konturen ermittelt und gespeichert werden). Bei der Abtastung eines Schriftzeichens entsteht z. B. das in Abbildung 2.15. dargestellte Speicherabbild.

Besonders bei Handschriften, aber auch beim Druck, sind durch unterschiedliche Abstände der Zeichen sowie durch unterschiedliche Zeichenbreiten, Zeichenhöhen und Kantenbreiten Abweichungen von Normzeichen vorhanden. Auch ist zu berücksichtigen, daß das Abtastbild horizontal oder vertikal verkantet sein kann, wenn die Leseeinrichtung schräg über das Zeichen geführt wurde. Die Erkennung besteht deshalb nicht nur in einem einfachen Vergleich zwischen Abtastfeld und Referenzmuster, sondern berücksichtigt auch charakteristische Merkmale des Zeichens (z. B. bei der »8« die zwei Nullfelder). In der Regel wird nicht auf Übereinstimmung mit dem Referenzmuster geprüft, sondern eine Wahrscheinlichkeit für jedes Zeichen des Zeichensatzes festgestellt. Als ausge-

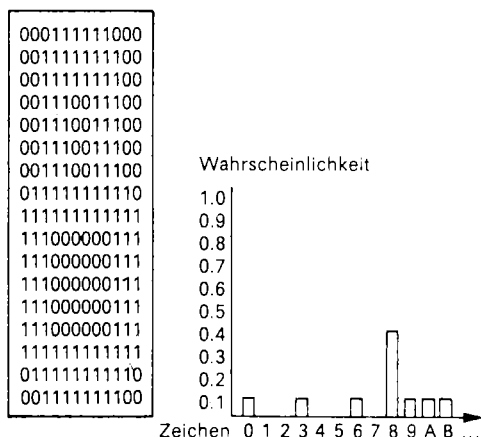


Abbildung 2.15.  
Speicherabbild einer »8«  
und Wahrscheinlichkeitsdiagramm

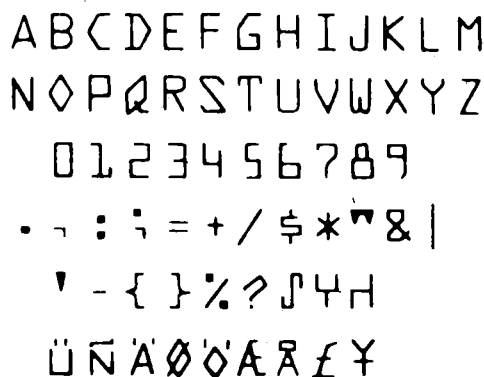


Abbildung 2.16.  
OCR/A-Zeichensatz

wählt zählt das Zeichen mit der höchsten Wahrscheinlichkeit (vgl. Wahrscheinlichkeitsdiagramm in Abbildung 2.15.). Wird dabei ein Grenzwert unterschritten, so wird auf Fehler erkannt und dieser akustisch angezeigt.

**OCR-Schriftarten**

Als Referenzmuster und Druckvorlage für OCR-Schrift werden die Schriftarten OCR/A (vgl. Abbildung 2.16.) und OCR/B unterschieden.

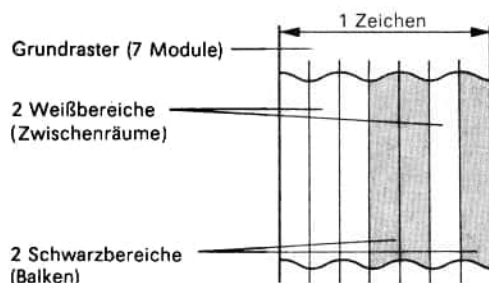


Abbildung 2.17.  
Darstellung einer (rechtsbündigen)  
0 im Balkencode

Die OCR/A-Schrift ist erkenntungssicherer für optische Lesegeräte, denn die Zeichen unterscheiden sich stark voneinander. OCR/B gleicht der gewohnten Zeichendarstellung durch Bildschirm und Drucker. Sie ist günstiger für das menschliche Auge.

Eine andere Vorgehensweise ist erforderlich, wenn ein Balkencode optisch gelesen wird. Die Strichmarkierung für ein Einzelzeichen erfolgt in einem Darstellungsbereich mit sieben Moduln. Ein Modul kann dunkel (Schwarzmodul, Balken) oder hell (Weißmodul, Zwischenraum) sein (vgl. Abbildung 2.17.).

Bei der optischen Abtastung des Balkencodes in Abbildung 2.17. entsteht das Speicherbild 0001101 (0 für Weißmodul, 1 für Schwarzmodul).

Da durch das menschliche Auge nicht die einzelnen Moduln, sondern nur zusammenhängende Schwarz- und Weißbereiche zu erkennen sind, entsteht der Eindruck stärkerer und schwächerer Balken, z. B. 0101111 mit einem schwächeren und einem stärkeren Balken. Die konkrete Zuordnung zwischen Bitmuster und Zeichen ergibt sich aus dem gültigen und vom Strichcodeleser technisch realisierten Code.

### Lesegeräte

Es werden mobile und stationäre optische Lesegeräte unterschieden.

**Mobile optische Lesegeräte (Handler, Lesestift, Lesepistole)** sind Lesegeräte, die manuell über den Lesebereich geführt werden.

Mobile optische Lesegeräte sind mit einem Erkennungsgerät verkabelt, das über ein Standardinterface mit einem Rechnersystem gekoppelt werden kann. Die Erkennungsgeschwindigkeit beträgt bis zu 140 Zeichen pro Sekunde. Einen Klarschrifthandler zeigt Abbildung 2.18.

**Stationäre optische Lesegeräte (Scanner)** sind Lesegeräte, die fest in andere Geräte oder Maschinen eingebaut sind und an denen die Lesebereiche vorbeigeführt werden.

Stationäre optische Lesegeräte sind meist mit den Transporteinrichtungen für Gegenstände, die die zu lesenden Zeichen tragen, verbunden. Die Lesegeschwindigkeit und die Erkennungssicherheit stationärer Leser ist höher als bei mobilen Lesegeräten. Schwierig ist das Vorbeiführen der Lesebereiche. Haupteinsatzgebiete für optische Lesetechnik sind international die Lagerwirtschaft (z. B. Regalauszeichnung), der Einzelhandel (artikelkonkrete Verkaufsdatenerfassung), das Gesundheitswesen und das Bibliothekswesen.

### 2.2.2.6. Magnetkartentechnik

**Magnetkartentechnik ist eine E/A-Technik für Information auf Plastekarten mit Magnetstreifen.**

Plastekarten mit Magnetstreifen sind standardisiert. Sie enthalten visuell lesbare Information in Form von Schriftzeichen und maschinenlesbare Information auf Magnetstreifen (vgl. Abbildung 2.19.).



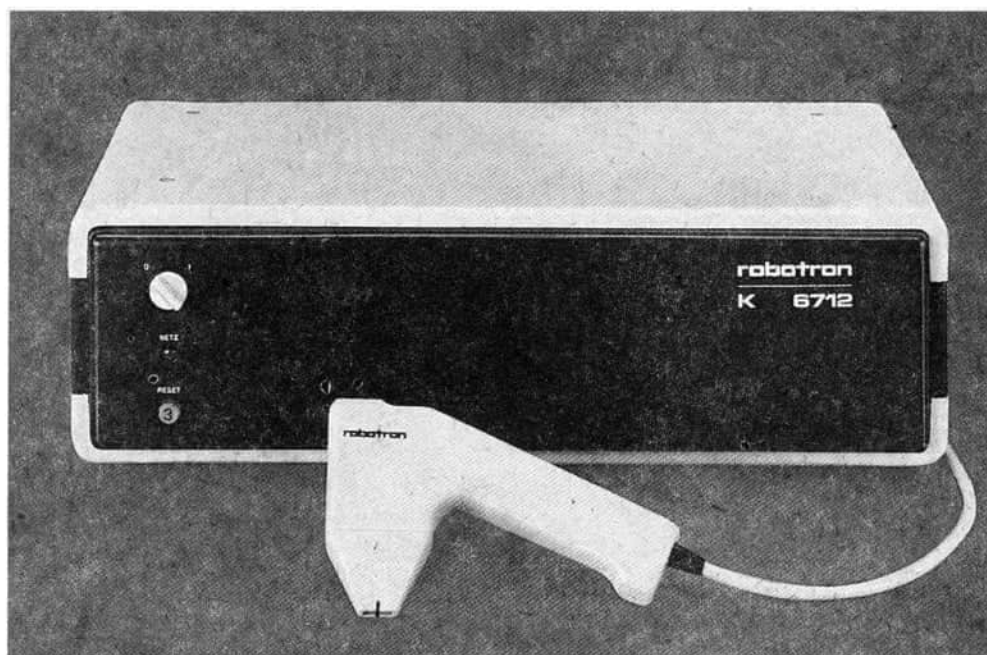


Abbildung 2.18.  
Klarschrifthandler robotron K 6712

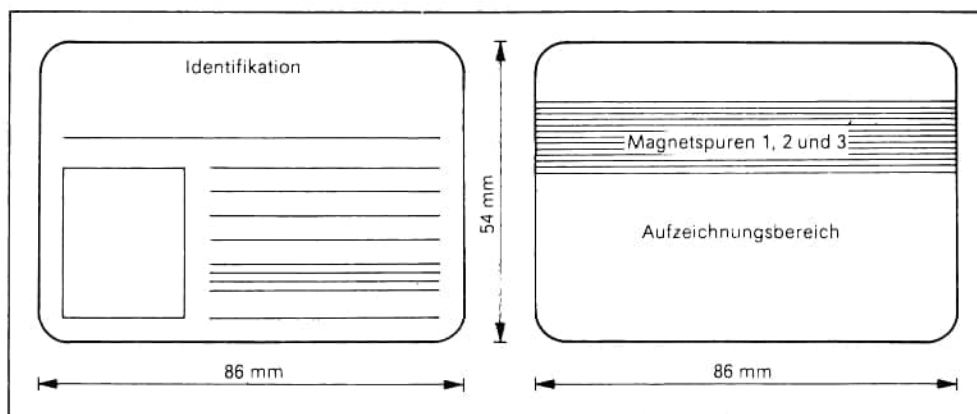


Abbildung 2.19.  
Plastekarte mit Magnetstreifen nach TGL 4909

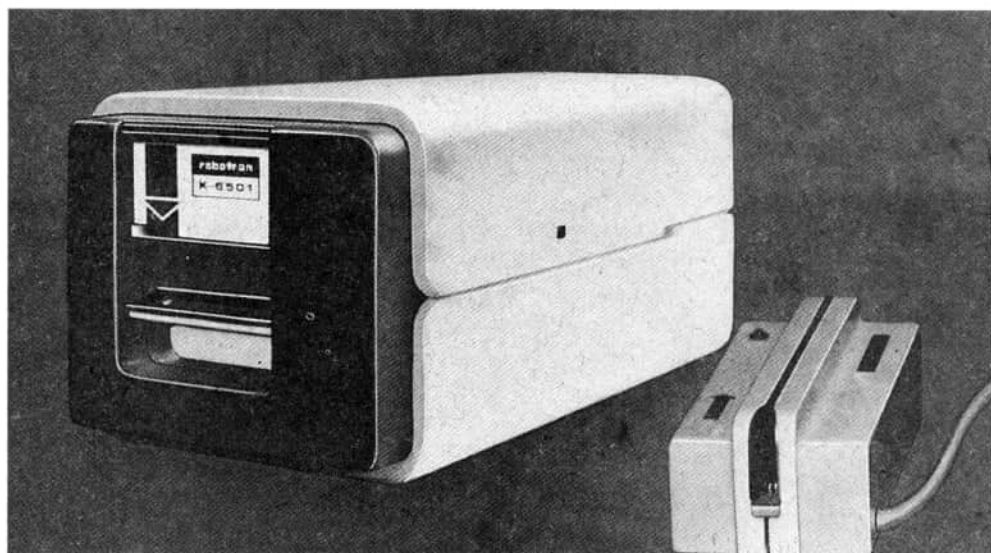


Abbildung 2.20.  
Lese- und Schreibeinheit K 6501

Die visuell lesbare Information (z. B. Paßbild, Personalien) dient im allgemeinen der Identifikation des Karteninhabers. Der Magnetstreifen kann auf den (unabhängigen, parallelen) Spuren 1 bis 3 insgesamt 1394 Bit (Spur 1 und 3 mit je 592, Spur 2 mit 210 Bit) speichern. Es werden besondere Aufzeichnungsverfahren (Wechseltaktschrift) benutzt, die Information wird speziell gesichert (Startbit, Stopbit, Prüfbit je Spur). Als periphere Ein- und Ausgabegeräte stehen Lese- und Schreibeinheiten zur Verfügung (vgl. Abbildung 2.20.).

Die Lesegeschwindigkeit liegt bei 10 bis 100 mm/s. Die Lesegeräte ermöglichen die automatische Lesewiederholung und die Eingabesperre. Zusätzlich können sie mit einer Beschlagnehmungseinrichtung ausgerüstet sein.

Plastekarten mit Magnetstreifen werden vor allem zur Handhabung personengebundener Information benutzt. Sie werden z. B. zur Rationalisierung der Geldwirtschaft, zum bargeldlosen Einkauf und zur Personenidentifikation eingesetzt.

### 2.2.2.7. Sprachein- und Sprachausgabetechnik

Mensch-Maschine-Kommunikation ist in natürlicher Sprache möglich.

**Sprachein- und Sprachausgabetechnik ist eine E/A-Technik, bei der Information durch das gesprochene oder hörbare Wort zwischen Mensch und Maschine übermittelt wird.**

#### Spracheingabe

Bei der Spracheingabe ist eine sprachliche Äußerung physikalisch als eine Folge von Druckschwankungen, dem Sprachsignal, meßbar. Mit einem Mikrofon wird das Sprachsignal in (kontinuierliche) elektrische Spannungsschwankungen umgesetzt. Die Spannungsschwankungen werden in genügend kurzen Zeitabständen gemessen (Abtastraten zwischen 8000 und 10000 Hertz). So entsteht eine Folge von Kennwerten, die das Sprachsignal genü-

gend genau beschreiben. Die Kennwerte werden einer Vorverarbeitung unterzogen (z. B. Filtern von Störgeräuschen). Anschließend werden charakteristische Merkmalswerte für die einzelnen Teilfolgen extrahiert und im Betriebsmodus Lernen gespeichert oder im Betriebsmodus Erkennen mit gespeicherten Werten verglichen. Es werden die in Abbildung 2.21. dargestellten Varianten unterschieden.

Für die Anwendung in der Wirtschaft genügend sicher ist die *sprecherabhängige Einzelworterkennung*. Die Wörter können aus einem Vorrat von etwa 50 (Spracherkennungseinheit K 7821), 100 (Spracherkennungseinheit K 7823) oder einigen hundert Wörtern ausgewählt werden. Beim Sprechen sind die Wörter durch maschinell erkennbare Pausen (z. B.  $> 300$  ms) zu trennen. Die Erkennungssicherheit liegt bei 97 %. Nicht erkannte Wörter werden durch Akustiksignal angezeigt. Die Wörter des Wörternvorrats sind durch den Sprecher in einer einmaligen Lernphase dem Rechnersystem vier- bis zehnmal vorzu-

sprechen und werden dann als Sprecherdatei gespeichert. Die Sprecherdatei wird geladen, wenn sich der Sprecher zur Rechnerbenutzung anmeldet. Der prinzipielle Ablauf geht aus Abbildung 2.22. hervor.

Die Spracheingabe ist vor allem von Bedeutung, wenn die Hände zur Benutzung der Tastatur nicht zur Verfügung stehen (z. B. Lagerwirtschaft) oder Entfernungen zu überbrücken sind (z. B. Ferndisposition, bei der direkt mit dem Rechnersystem telefoniert werden kann). Hinderlich sind hohe Nebengeräuschpegel und oft wechselnde Nutzer.

### Sprachausgabe

Bei der Sprachausgabe wird zwischen der Ausgabe einer natürlichen und der Ausgabe einer synthetischen Sprache unterschieden. Bei Ausgabe natürlicher Sprache werden gesprochene Wörter oder Sätze gespeichert und auf Anforderung so oder in verschiedener Kombination hörbar wiedergegeben. Bei synthetischer Sprachausgabe wird die hörbare Ausgabe aus phonetischen Grundelementen (Phoneme) erzeugt. Der Wörternvorrat ist in diesem Fall unbegrenzt.

Die Anwendung der Sprachausgabe ist von Bedeutung, wenn sich der Empfänger in Bewegung befindet (verbunden mit Funksender) oder wenn mehrere Empfänger die Information gleichzeitig erhalten müssen. Die Sprachausgabe ist zur Zeit weniger bedeutsam als die Spracheingabe. Beide Formen können natürlich getrennt voneinander angewendet werden.

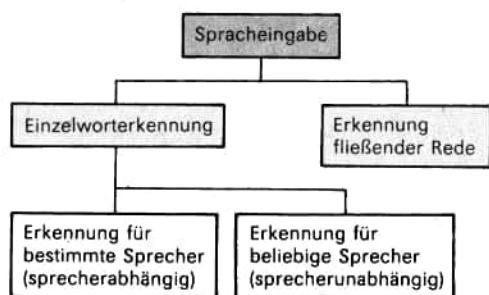


Abbildung 2.21.  
Varianten der Spracherkennung

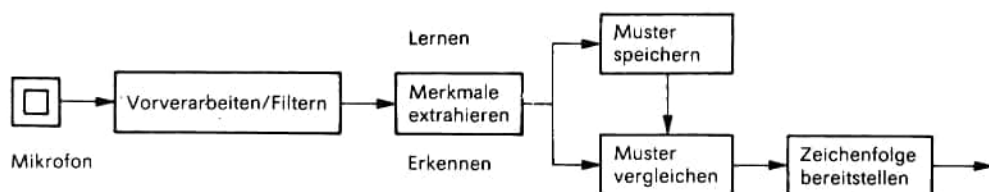


Abbildung 2.22.  
Spracherkennung (Grundprinzip)

## 2.2.3. Externe Speicher

### 2.2.3.1. Klassifikation

Die externen Speicher dienen der Nachschub- und Archivspeicherung außerhalb der Zentraleinheit, aber innerhalb des Rechnersystems. Sie unterscheiden sich nach Beschaffenheit und Aufzeichnungsförmern. Die wichtigsten sind magnetische Speicher und Halbleiterspeicher. Es gibt außerdem optische Speicher, die bisher aber nicht verbreitet sind. Magnetische Speicher lassen sich in Magnetspeicher und Magnetblasenspeicher klassifizieren. Magnetblasenspeicher haben sich noch nicht umfassend durchgesetzt. Nachstehend werden die weitverbreiteten Magnetspeicher und die externen Halbleiterspeicher (Halbleiterplatten) behandelt.

**Magnetspeicher sind (nicht flüchtige) externe Speicher. Die Speicherung der Information erfolgt durch partielle Magnetisierung einer Oberflächenschicht.**

In der Regel wird beim Lesen und Schreiben des Speichers das Speichermedium an einem Schreibkopf elektromotorisch vorbeigeführt. Diese Speicher heißen deshalb (im Unterschied zu den Magnetblasenspeichern) auch magnetmotorische Speicher. Abbildung 2.23. gibt einen Überblick über verschiedene Magnetspeicher.

### 2.2.3.2. Magnetbandtechnik

Zur Archivspeicherung großer Informationsmengen werden Magnetbänder benutzt.

**Das Magnetband (magnetic tape) ist ein bandförmiger Datenträger, auf das Information durch Magnetisierung aufgezeichnet werden kann. Es besteht aus einer nichtmagnetisierbaren Trägerfolie (z. B. Polyester) und einer ferromagnetischen Beschichtung ( $\text{Fe}_2\text{O}_3$ ).**

Das Lesen von und Schreiben auf Magnetbänder erfolgt durch ein Magnetbandgerät (Magnetbandeinheit) oder durch ein Magnetbandkassettengerät (vgl. Abbildung 2.24.). Es gibt verschiedene Aufzeichnungsverfahren (Impulsschrift RZ, Dauerstromverfahren NRZ, Phasenmodulation, Frequenzmodulation). Die Geräte sind stets nur für ein Verfahren ausgelegt.

Es sind unterschieden zwischen:

- ½ Zoll-Magnetband auf kleiner Spule (360 m) oder großer Spule (720 m). Die Aufzeichnung erfolgt in 7–9 Spuren (byteparallel). Die Anordnung ist ähnlich dem Lochstreifen. Die Kapazität beträgt etwa 11 (kleine Spule) oder 22 MByte.
- ½ Zoll-Magnetband in einer Magnetbandkassette. Die Kapazität einer Seite beträgt etwa 350 KByte

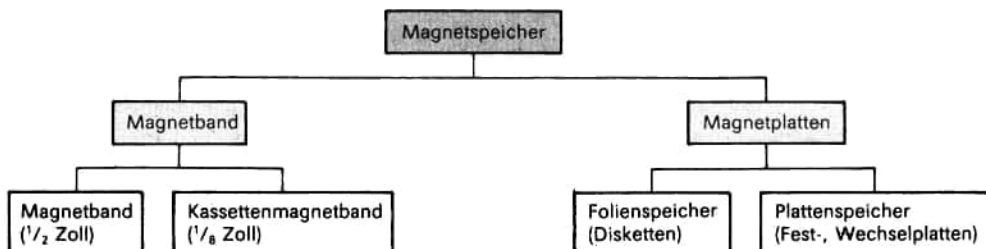


Abbildung 2.23.  
Magnetspeicherklassifikation (Auswahl)

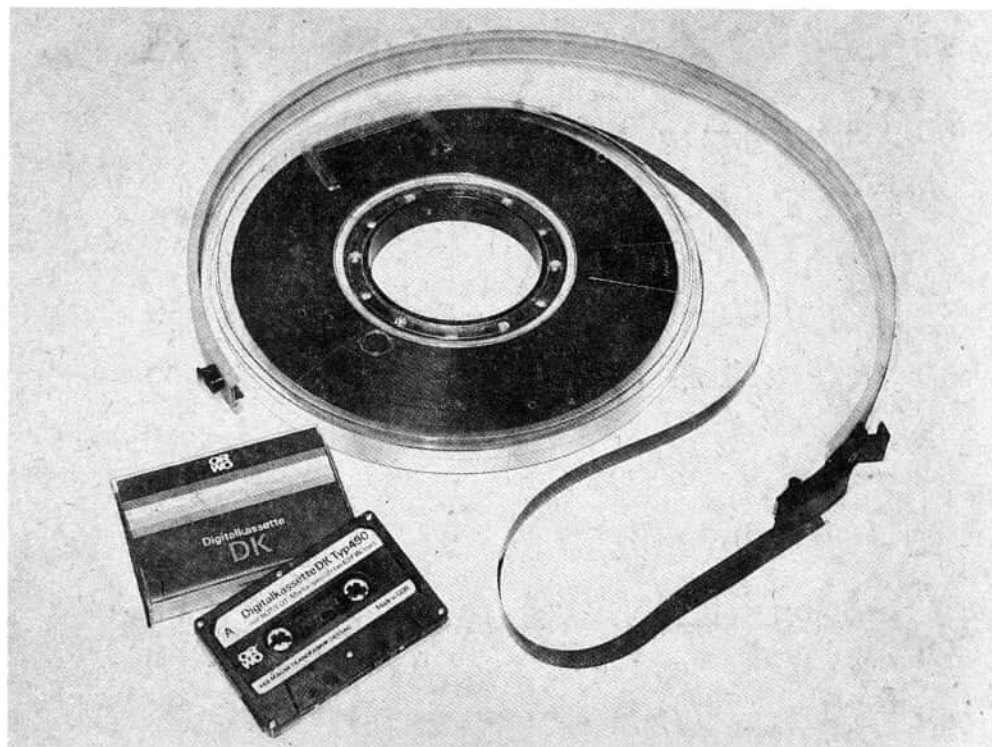


Abbildung 2.24.  
Magnetbandspule und Magnetbandkassette

Der Zugriff (Lesen oder Schreiben) ist beim Magnetband nur sequentiell möglich. Die Zugriffszeiten sind also im Gegensatz zum wahlfreien Zugriff abhängig von der Aufzeichnungsposition der Information.

Die Aufzeichnung erfolgt Datei für Datei. Nach dem Aufzeichnen einer Datei werden stets zwei *Blockmarken* (spezielle Bitmuster) geschrieben, die zugleich als (logisches) Ende des Magnetbandes (Bandmarke) gelten. Folgt die Aufzeichnung einer weiteren Datei am Ende, so wird der Schreibkopf zwischen die beiden Blockmarken positioniert. Bei der folgenden Aufzeichnung wird die letzte der beiden Blockmarken überschrieben, so daß zwei aufeinanderfolgende Dateien durch eine Blockmarke getrennt sind. Am Ende der Aufzeichnung werden wieder zwei Blockmar-

ken und damit eine Bandmarke geschrieben. Diesen für den praktischen Gebrauch von Magnetbändern wichtigen Gesichtspunkt demonstriert die Abbildung 2.25.

Wird eine neue Datei an den Anfang des Magnetbandes geschrieben, so ist die eventuell folgende Information verloren, da unbedingt eine Bandmarke geschrieben wird. Es muß also stets zum Aufzeichnungsende vorgespult und dann ange-

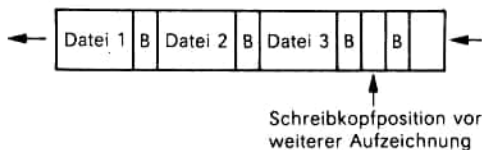


Abbildung 2.25.  
Dateien und Blockmarken  
der Bandaufzeichnung von drei Dateien



Abbildung 2.26.  
Einlegen einer Diskette

hängt werden. Im Beispiel (Abbildung 2.25.) ist ein Ändern (update) der Dateien 1 und 2, das mit Schreiben verbunden wäre, unmöglich, ohne die nachstehenden Dateien zu zerstören. Deshalb muß periodisch eine Reorganisation durch Umspeichern erfolgen.

Die Magnetbandgeräte verfügen über Zusatzseinrichtungen, die den Nutzer vor dem irrtümlichen Überschreiben von Dateien in gewissem Maße schützen (beim Schreiben nach dem Lesen wird Fehler erkannt).

Magnetbänder eignen sich auf Grund des relativ geringen Aufwandes für ihre Herstellung und durch ihre große Speicherkapazität zur Archivspeicherung großer Informationsmengen. Die Kosten pro Bit sind gering, die Zugriffszeiten aber relativ lang.

### 2.2.3.3.

#### Folienspeichertechnik

Die Folienspeichertechnik ist eine Form der Magnetplattentechnik. Es ist wahlfreier Zugriff möglich. Speichermedium ist die Diskette.

**Eine Diskette (floppy disk, Magnetfolie) ist eine kreisrunde, biegsame Folie, die beiderseits mit einer magnetisierbaren Schicht überzogen ist und die zum Schutz in einer (innen beschichteten) Plastekassette untergebracht ist.**

Nach den Abmessungen der Plastekassette wird in 3,5-Zoll-Disketten, 5,25-Zoll-Disketten (Minidisketten) und 8-Zoll-Disketten (Normaldisketten) unterschieden.



### Folienspeicherlaufwerk

Das Schreiben und Lesen von Disketten erfolgt durch ein Folienspeicherlaufwerk (kurz *Laufwerk*), das bei Personalcomputern in das Rechnergrundgerät eingebaut ist und eventuell zusätzlich als Beistellgerät existiert. Das Einlegen einer Diskette in ein Laufwerk zeigt die Abbildung 2.26.

Über die Zentrier- und Treibmechanik des Laufwerkes wird die Folie in der Plastekassette in schnelle Umdrehung versetzt. Das Indexloch dient der physischen Orientierung des Schreib- und Lesekopfes, der sich über der Schreib- und Leseöffnung befindet. Befindet sich die adressierte Position der Diskette während der Umdrehung unter dem Schreib- und Lesekopf, erfolgt der physische Zugriff. Die Aussparung für den Schreibschutz kann verklebt werden (Minidiskette), oder der Klebestreifen kann entfernt werden (Normaldiskette). Dadurch wird das Schreiben unmöglich, und der Disketteninhalt ist geschützt.

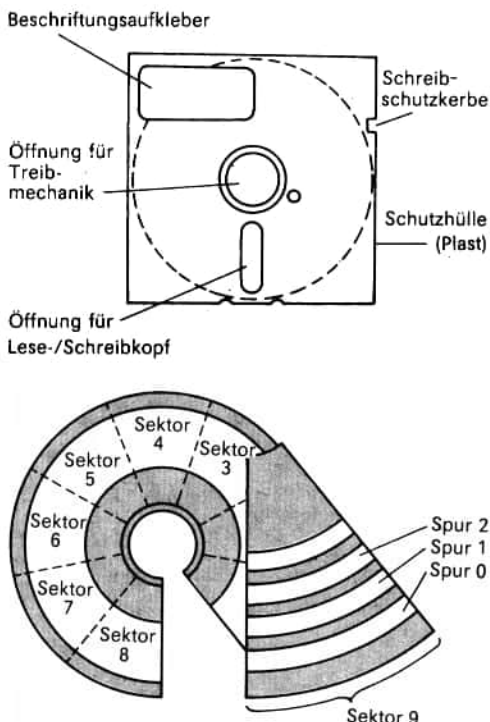


Abbildung 2.27.  
Diskettenhülle / Einteilung einer  
Diskettenoberfläche

### Disketteneinteilung

Die Diskette ist in konzentrische Spuren eingeteilt.

**Eine Spur (track) ist ein für die Informationsaufzeichnung vorgesehener konzentrischer Kreis auf der Folienoberfläche.**

Die Spuren können in einfacher Spurdichte (Single Density, SD) mit 48 Spuren je Zoll (48 TPI, Track Per Inch) oder doppelter Spurdichte (Double Density, DD) geschrieben werden. Die Diskette ist entweder nur auf der Vorderseite (Single Sided, SS) oder auf der Vorder- und Rückseite (Double Sided, DS) benutzbar. Bei doppelseitiger Nutzung bilden Spuren, die sich auf der Ober- und Unterseite gegenüberliegen, einen Zylinder.

Zur DD- und DS-Benutzung müssen Disketten vom Hersteller zugelassen sein. Sie sind entsprechend beschriftet.

Auf Minidisketten werden im allgemeinen bei einfacher Spurdichte 40 Spuren, bei doppelter Spurdichte 80 Spuren angelegt. Die Zählung der Spuren beginnt jeweils mit 0 (0 bis 39, 0 bis 79).

Die Spuren sind in Sektoren unterteilt (vgl. Abbildung 2.27.).

**Ein Sektor ist der Abschnitt zwischen zwei Radialen. Er besitzt einen Kennzeichnungsteil, einen Datenteil und eine Trennlücke.**

Mögliche Sektorgrößen (Sektorierungen) sind

26 Sektoren zu je 128 Byte  
= 3,25 KByte/Spur

16 Sektoren zu je 256 Byte  
= 4 KByte/Spur

- 9 Sektoren zu je 512 Byte  
= 4,5 KByte/Spur
- 4 Sektoren zu je 1024 Byte  
= 4 KByte/Spur
- 5 Sektoren zu je 1024 Byte  
= 5 KByte/Spur

Die Zeichendichte pro Spur nimmt in dieser Aufzählung von oben nach unten zu. Die Kennzeichnung zur Density-Eignung von Disketten bezieht sich auch auf diese Zeichendichte in der Spur.

### Diskettenkapazität

Die Bruttospeicherkapazität  $B$  einer Diskette ergibt sich wie folgt:

$$B = \text{Sektoranzahl} * \text{Sektorlänge} \\ * \text{Spuranzahl} * \text{Seitenanzahl}$$

Kapazitäten bei häufig verwendeten Sektorisierungen sind in Tabelle 2.6. zusammengestellt.

Die Diskettengröße tendiert zur Verringerung des Durchmessers (von 8 Zoll zu 5,25 Zoll oder 3,5 Zoll) bei gleicher oder höherer Kapazität.

Die für den Nutzer verwendbare Diskettenkapazität verringert sich betriebssystemabhängig. Im Betriebssystem SCP z. B. um

- 2 bis 3 Spuren (Spuren 0 bis 2), die zur Speicherung des Betriebssystems vor-

gesehen (Systemspuren) und im allgemeinen dem Nutzer nicht zugänglich sind;

- 2 bis 4 KByte für das Diskettenverzeichnis (Directory), das unmittelbar nach den Systemspuren angelegt wird.

Die Kapazität der Disketten und auch ihre Belegung kann über Dienstprogramme (vgl. Kapitel 3.) angezeigt werden.

Die Aufzeichnung und Wiedergabe von Information auf einer Diskette wird durch das Betriebssystem (vgl. Kapitel 3.) gesteuert. Die dafür im Betriebssystem installierten Parameter müssen mit dem Format der Diskette verträglich sein.

**Das Diskettenformat ist die Gesamtheit der (softwaregesteuerten) Parameter einer Diskette. Das Diskettenformat wird durch Initialisierungs-/Formatierungsprogramme eingestellt und bleibt bis zur erneuten Formatierung unverändert.**

Fabrikneue Disketten müssen vor ihrer ersten Benutzung initialisiert (formatiert) werden. Dabei erfolgt zugleich eine Kontrolle auf einwandfreie Beschaffenheit.

Natürlich muß die Gerätetechnik auch physisch das jeweilige Aufzeichnungsverfahren (FM für SD, MFM für DD) ermöglichen.

Tabelle 2.6.

Bruttodiskettenkapazität

Sektoranzahl	Sektorlänge	Kapazität (KByte)		
		SS-SD	SS-DD/DS-SD	DS-DD
26	128	130	260	520
16	256	160	320	640
9	512	180	360	720
4	1024	160	320	640
5	1024	200	400	800



### Diskettenbehandlung

Disketten sind sorgfältig zu behandeln. Es gelten folgende Regeln:

1. Nicht auf die Lese-/Schreiböffnung fassen oder diese mit Gegenständen berühren.
2. Disketten nicht knicken, verdrehen oder mechanisch auf sie einwirken (z. B. Beschriften mit Kugelschreiber oder Aufbewahren unter schweren Gegenständen).
3. Disketten nicht in Berührung mit magnetischen Objekten (z. B. Büroklammern) oder in die Nähe von Magnetfeldern bringen (Transformatoren, Telefonapparate).
4. Disketten nicht in die Nähe von Staubquellen legen (Lüfter, Aschenbecher, offene Fenster, Gardinen) und für ausreichende Luftfeuchtigkeit sorgen.
5. Disketten stets in den Papierschutzhüllen aufbewahren und nicht, auch nicht kurzzeitig, offen liegen lassen.
6. Disketten keinen Temperaturen außerhalb des Bereiches von 10° bis 52°C aussetzen.
7. Vor dem Ausschalten des Rechners die Laufwerke entriegeln.
8. Disketten nicht während eines Diskettenzugriffs (angezeigt durch das Leuchten einer Laufwerksdiode) aus dem Laufwerk nehmen.

Von wichtigen Disketteninhalten ist stets eine Kopie anzufertigen.

Die Folienspeichertechnik ist die universelle Speichertechnik der Personalcomputer. Disketten können körperlich transportiert werden. Dadurch ist der Transport von Information zwischen Rechnersystemen durch Datenträgertransport möglich. Außerdem lassen sich Disketten in verschlossenen Behältnissen aufbewahren und können so vor unberechtigtem Zugriff geschützt werden.

Die Kosten für die Speicherung von Information auf Disketten sind pro Bit höher als

bei Magnetbändern, aber trotzdem relativ gering. Der Zugriff erfolgt schneller. Disketten eignen sich besonders für die Nachschubspeicherung kleiner Informationsmengen.

### 2.2.3.4. Plattentechnik

Platten sind Festplatten, Wechselplatten oder als Sonderfall Halbleiterplatten.

**Festplatten und Wechselplatten sind kreisrunde, scheibenförmige Datenträger, auf denen Information durch Magnetisierung aufgezeichnet werden kann. Fest- und Wechselplatten bestehen aus einem Trägermaterial (Plast) und einer  $\text{Fe}_2\text{O}_3$ -Beschichtung.**

#### Festplattenspeicher

Festplattenspeicher (hard disk) haben einen Durchmesser von 14; 8; 5,25 oder 3,5 Zoll und sind fest in ein Laufwerk eingebaut. Bei Festplatten in der Winchester-technologie (vgl. Abbildung 2.28.) ist das Gehäuse hermetisch abgeschlossen und mit Edelgas gefüllt. Winchesterlaufwerke realisieren mit höchster Präzision (der Ab-

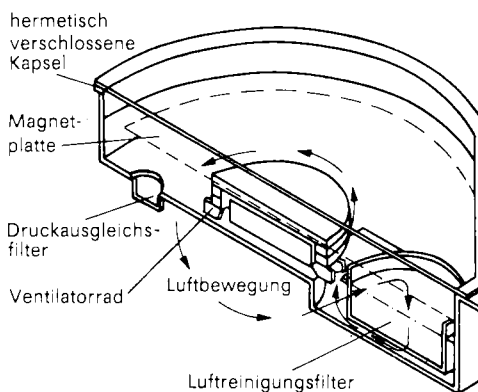


Abbildung 2.28.  
Darstellung einer  
Festplatte in Winchester-technologie

stand zwischen Schreib-/Lesekopf und Plattenoberfläche beträgt während des Betriebes etwa 1 Mikrometer, die Rotation beträgt bis zu 3600 Umdrehungen pro Minute) Aufzeichnungsdichten bis zu 40 000 Bit/mm<sup>2</sup>.

Vor dem Ausschalten des Gerätes ist es zweckmäßig, die Schreib- und Leseköpfe von der Platte abzuheben und in eine Ruhelage zu bringen. Dieser Vorgang wird als *Parken* bezeichnet und erfolgt mit einem Parkprogramm.

Die Kapazität der Festplatten liegt im allgemeinen zwischen 10 und 50 MByte. Es können aber auch weit größere Kapazitäten erreicht werden.

Durch die feste Verbindung mit dem Rechnersystem sind Festplattenspeicher keine Archivspeicher. Außerdem sind die Kosten je Bit hoch.

Festplattenspeicher gehören zur externen Speicherperipherie der Personalcomputer A 7150 und EC 1834. Sie sind in das Grundgerät eingebaut.

### Wechselplattenspeicher

Wechselplattenspeicher bestehen aus einem Satz von 6 bis 12 übereinander angeordneten Magnetplatten (Plattenstapel). Der Plattenstapel als Ganzes kann ausgetauscht werden.

Die Außenflächen der oberen und der unteren Platte werden nicht zur Speicherung verwendet. Es stehen also insgesamt 10 bis 22 (innere) Speicherflächen zur Verfügung.

Zum Schreiben und Lesen wird ein Zugriffskamm verwendet. Er besteht aus vertikal angeordneten Lese- und Schreibköpfen, die als Ganzes radial (in die Räume zwischen den Platten) bewegt und so positioniert werden können. Die Platten sind wie die Disketten in konzentrische Spuren eingeteilt. Die Schreib- und Leseköpfe des Zugriffskamms befinden sich stets über den gleichen Spuren verschiedener Plat-

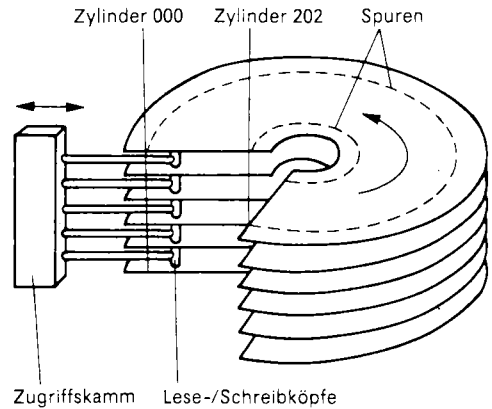


Abbildung 2.29.  
Aufbau eines Wechselplattenspeichers

ten. Die Gesamtheit übereinanderliegender Spuren eines Plattenstapels heißt Zylinder.

Das Lesen und Schreiben der Information erfolgt wegen des schnelleren Zugriffs nach Zylindern (vgl. Abbildung 2.29.).

Die Speicherkapazität je Wechselplattenstapel beträgt im allgemeinen 7,25 bis 200 MByte. Bei speziellen Speichern wird 1 GByte erreicht.

Da die Stapel auswechselbar sind, können sie in begrenztem Maße auch für die Archivspeicherung häufig benutzter Information verwendet werden. Der Aufwand für die Speicherung ist pro Bit jedoch deutlich höher als beim Einsatz von Magnetbandtechnik.

### Halbleiterplatten

**Halbleiterplatten (RAM-Disk) sind extrem schnelle, flüchtige externe Schreib-/Lesespeicher auf RAM-Speicherbauelementen.**

Halbleiterplatten sind wie der Hauptspeicher aufgebaut, werden aber programmtechnisch wie eine Magnetplatte behan-

delt. Die Herstellung von Halbleiterplatten ist relativ aufwendig. Halbleiterplatten sind infolge der fehlenden Mechanik verschleißfrei. Die Kapazität von Halbleiterplatten liegt im allgemeinen zwischen 150 KByte und mehreren MByte.

Einige Betriebssysteme (vgl. Kapitel 3.) bieten dem Nutzer an, einen Teil des Hauptspeichers als RAM-Disk (virtuelles Laufwerk) zu nutzen. In die RAM-Disk werden häufig benötigte Programme und Dateien geladen, um so die Zugriffsgeschwindigkeit zu erhöhen. Zu beachten ist, daß die Information beim Ausschalten des Gerätes verlorengeht.

## 2.3. Rechnerverbundsysteme

### 2.3.1. Grundbegriffe der Online-Kommunikation

Die Online-Kommunikation (vgl. Abschnitt 1.1.) nimmt für die weitere Entwicklung von Informations- und Kommunikationstechnologien eine Schlüsselstellung ein. Sie ermöglicht eine höhere Effektivität der Automatisierung von Informationsprozessen und eröffnet qualitativ neue Anwendungsmöglichkeiten der automatisierten Informationsverarbeitung.

#### Kommunikationsstation

In der allgemeinen Kommunikationskette wird zwischen dem sendenden System A, dem Kanal (Übertragungskanal) und dem empfangenden System B unterschieden. Das sendende System besteht aus der Quelle und dem Sender, das empfangende System aus dem Empfänger und der Senke. Da die Übertragungsrichtung der Information wechseln kann, tragen alle an der Rechnerkommunikation teilnehmenden Systeme die Bezeichnung Kommunikationsstation.

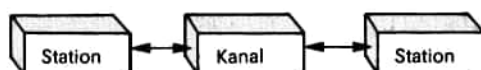


Abbildung 2.30.  
Rechnerkommunikationskette

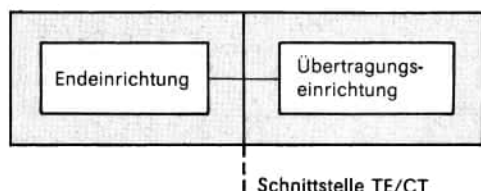


Abbildung 2.31.  
Grundstruktur  
einer Kommunikationsstation

Eine Kommunikationsstation (Datenstation, Station) ist eine Einrichtung, die die Fähigkeit besitzt, Information in maschinenlesbarer Form (Daten) über einen Kanal zu senden oder zu empfangen.

Eine Rechnerkommunikationskette hat die in Abbildung 2.30. dargestellten Bestandteile.

Jede Kommunikationsstation besteht aus zwei Teilen (vgl. Abbildung 2.31.), einer Endeinrichtung (Terminal Equipment, TE) und einer Übertragungseinrichtung (Communication Equipment, CT).

*Endeinrichtungen* können alle Geräte sein, die in der Lage sind, Information zu verarbeiten oder zu speichern.

Dazu gehören:

- Zentraleinheiten,
- Geräte zur Mensch-Maschine-Kommunikation, wie Bildschirm, Tastatur, Drucker,
- Geräte zur externen Speicherung, wie Magnetband- oder Magnetplatteneinheiten.

Die *Übertragungseinrichtung* paßt die rechnerinterne Darstellung der Informa-

tion den Übertragungsbedingungen des Kanals an. Es ist zwischen der Übertragung im Nahbereich (einige 100 m) und im Fernbereich zu unterscheiden.

Für die Übertragung im Nahbereich werden *Repeater* eingesetzt, die eine Signalverstärkung vornehmen. Die Störungen bei der Übertragung nehmen mit der Länge des Übertragungsweges zu und zwingen deshalb zu einer drastischen Begrenzung der Übertragungsentfernungen.

Bei Übertragungen im Fernbereich werden besondere Maßnahmen zur Sicherung der Information unternommen.

Dazu gehören:

- die Codierung mit einem fehlererkennenden oder fehlerkorrigierenden Code. Durch die Übertragung deformierte Bitmuster können als fehlerhaft erkannt und innerhalb bestimmter Grenzen korrigiert werden.
- die Änderung der Informationsdarstellung aus der diskreten in die für die Fernübertragung erforderliche kontinuierliche Form. Dieser Vorgang heißt Modulation, seine Umkehrung heißt Demodulation, und das entsprechende Gerät wird als Modem bezeichnet.

**Ein Modem ist ein kombiniertes Gerät zur Modulation und Demodulation von Signalen für die Fernübertragung.**

Für die Fernübertragung werden auch *Akustikkoppler* eingesetzt. Akustikkoppler transformieren digitale Signale in akustische Signale und umgekehrt. Zur Übertragung wird der Handapparat eines Telefons und als Übertragungsweg wird das Telefonnetz benutzt.

Die Übertragungsleistung eines Akustikkopplers ist deutlich geringer als die Übertragungsleistung eines Modems. Der Einsatz eines Akustikkopplers erfolgt, wenn geringe Informationsmengen zu übertra-

gen sind und dazu das Telefonnetz benutzt werden muß. Einsatzgebiete für Akustikkoppler sind die mobile Datenerfassung und die Ferndisposition.

An einer Rechnerkommunikation können mehr als zwei Stationen teilnehmen. Dann ist eine Station entweder Terminalstation oder Vermittlungsstation.

**Eine Terminalstation ist eine Station, die am Ende einer Rechnerkommunikationskette steht und selbst Information verarbeitet. Die Endeinrichtung einer Terminalstation heißt Terminal (Datenendplatz, Abonnen-tenpunkt, Arbeitsstation).**

Eine Terminalstation verarbeitet Information oder speichert sie dauerhaft. Eine Vermittlungsstation setzt sie lediglich um, erforderlichenfalls mit Zwischenspeicherung.

**Eine Vermittlungsstation (Vermittlungsknoten, Knoten) ist eine Station, die Verbindungen zwischen anderen Kommunikationsstationen aufbaut, steuert oder in anderer Form unterstützt.**

### Verbindungsstrukturen

Nach der Art und Weise, wie Kommunikationsstationen miteinander verbunden sind, wird zwischen Zweipunktverbindungen und Mehrpunktverbindungen unterschieden.

**Eine Zweipunktverbindung (point-to-point-connection) ist eine direkte Verbindung zwischen genau zwei (Terminal-)Stationen. Sie wird nur von diesen beiden Stationen benutzt.**

Bei der Zweipunktverbindung ist keine Vermittlungsstation zwischengeschaltet. Eine Station kann aber mit mehreren anderen Stationen verbunden sein (vgl. Abbildung 2.32.).

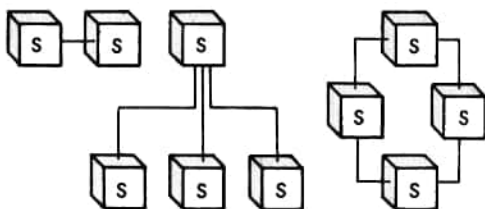


Abbildung 2.32.  
Typische Zweipunktverbindungen  
(S – Station)

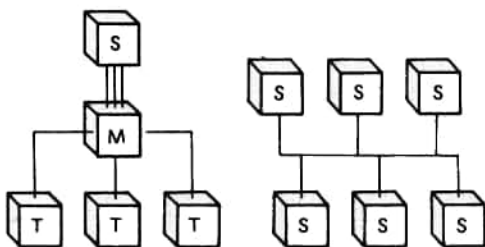


Abbildung 2.33.  
Typische Mehrpunktverbindungen  
(S – Station, T – Terminalstation,  
M – Multiplexer)

Bei einer Zweipunkt-Ringverbindung puffern die Kommunikationsstationen die Information und leiten sie dann weiter. Im allgemeinen führt die Zweipunktverbindung nur zu einer geringen Auslastung des Übertragungsweges. Sind mehr als zwei Stationen zu verbinden, ist der Aufwand für die Vielzahl von Zweipunktverbindungen hoch.

**Eine Mehrpunktverbindung (multi-point-connection) ist eine Verbindung zwischen mehr als zwei Stationen. Die Verbindung kann über Vermittlungsstationen führen. Sie wird von mehreren Stationen genutzt.**

Mögliche Strukturen der Mehrpunktverbindung zeigt die Abbildung 2.33.

Der Vorteil der Mehrpunktverbindung liegt darin, daß verschiedene Kommunika-

tionsstationen gleiche Übertragungskanäle benutzen können. Ein Nachteil ist, daß zu einem gegebenen Zeitpunkt immer nur eine Station senden kann.

Bei Mehrpunktverbindungen kann es erforderlich sein, Übertragungskanäle zum rationelleren Transport zu bündeln oder umgekehrt, gebündelte Kanäle für verschiedene Stationen wieder zu verzweigen. Diese Aufgaben übernehmen Konzentratoren und Multiplexer.

**Konzentratoren sind Vermittlungsstationen, die mehrere Übertragungskanäle (geringerer Kapazität) auf Übertragungskanäle größerer Kapazität führen.**

Das geschieht unter Verwendung eines Pufferspeichers, so daß Kapazitätsengpässe im schnellen Kanal ausgeglichen werden können.

**Multiplexer sind Vermittlungsstationen, die die (höhere) Kapazität eines Übertragungskanals in mehrere Übertragungskanäle (geringerer) Kapazität aufteilen.**

Konzentratoren und Multiplexer benutzen verschiedene Frequenzen eines schnellen (Breitband-)Übertragungsweges (frequenzmultiplex) oder verschachteln die Signale zeitlich (zeitmultiplex).

Verbindungsstrukturen, die eine Kommunikation aller Stationen untereinander ermöglichen, bilden ein Übertragungsnetzwerk. Es kann sich um Zweipunkt-Ringstrukturen oder Mehrpunktstrukturen handeln.

**Ein Übertragungsnetzwerk (Übertragungsnetz) ist eine Verbindungsstruktur zwischen mehr als zwei Kommunikationsstationen, bei der jede Station mit jeder Station kommunizieren kann.**

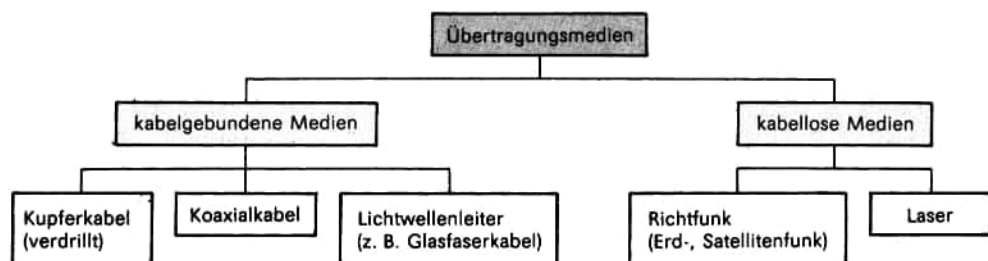


Abbildung 2.34.  
Übertragungsmedien (Auswahl)

Zur Kommunikation im Übertragungsnetz benutzen die Stationen entweder eine Station mit Speicherfunktion oder eine gemeinsame Sammelleitung (globaler Bus). Im ersten Fall wird die Information von der sendenden Station im Pufferspeicher (Briefkasten) abgelegt und von der empfangenden Station dort abgeholt. Im zweiten Fall wird die Information von der sendenden Station auf den Bus gebracht und dafür gesorgt, daß nur die empfangende Station Zugriff zum Bus hat.

### Übertragungsmedien

Die Information wird über Wege (Übertragungswege, Leitungen) transportiert, die physikalisch durch Übertragungsmedien realisiert sind. Die Klassifikation der Übertragungsmedien ist in Abbildung 2.34. dargestellt.

Wichtige Parameter der Übertragungsmedien sind die Bandbreite, die Übertragungsgeschwindigkeit und die Schwingungsdämpfung.

**Die Übertragungsgeschwindigkeit ist eine Kenngröße für die Anzahl der Bit, die pro Sekunde durch das verwendete Medium übertragen werden können (Bit/s).**

Die Übertragungsgeschwindigkeit wird von der Bandbreite des Übertragungsmediums beeinflusst.

**Die Bandbreite ist der in Schwingungen pro Sekunde (eine Schwingung pro Sekunde ist gleich 1 Hertz) gemessene Bereich, in dem die Übertragung in diesem Medium möglich ist.**

Ein breitbandiges Medium (größer oder gleich 5 MHz) läßt sich in mehrere (parallele) Übertragungskanäle (Frequenzbänder) aufteilen, die dann multiplex (Frequenzmultiplexbetrieb) genutzt werden können. Die mögliche Anzahl der Kanäle eines Übertragungsmediums wirkt als Multiplikator der möglichen Übertragungsgeschwindigkeit des Gesamtmediums.

Werden in einem Medium mehrere Übertragungskanäle (im Multiplexbetrieb) genutzt, so handelt es sich um Breitbandübertragung, sonst um Basisbandübertragung. Breitbandübertragung erfordert einen bedeutend höheren Aufwand.

Bei der Übertragung werden die Schwingungen mit zunehmender Übertragungsentfernung gedämpft.

**Die Dämpfung ist das (logarithmierte) Verhältnis zwischen dem Abfall der Übertragungsleistung auf einer bestimmten Übertragungsentfernung zu dieser Übertragungsentfernung. Sie wird in Dezibel (dB) pro Streckeneinheit ausgedrückt (z. B. dB/km).**

Tabelle 2.7.

Parameter von Übertragungsmedien (Auswahl)

Übertragungs- medium	Übertragungs- geschwindigkeit	Maximale Entfernung	Erforderlicher Aufwand
Kupferkabel	150 KBit/s Basisband	5 km	niedrig
Koaxialkabel	10 MBit/s Basisband	3 km	niedrig
	300 MBit/s Breitband	3 km	hoch
Glasfaserkabel	600 MBit/s Breitband	30 km	hoch
Erdfunk	4 MBit/s	10 km	niedrig
Satellitenfunk	10 GBit/s	10 000 km	hoch

Von der Dämpfung ist die Distanz abhängig, die ein Signal ohne Zwischenverstärkung im Übertragungsmedium zurücklegen kann. Eine Übersicht über wichtige Parameter ausgewählter Übertragungsmedien enthält Tabelle 2.7.

Zur Beurteilung eines Trägermediums sind weitere Gesichtspunkte heranzuziehen, z. B. die Empfindlichkeit gegenüber äußeren Einflüssen, die Abstrahlung (und damit die Abhörsicherheit) und bei kabelgebundenen Medien auch die Verlegbarkeit der Kabel.

### Betriebsarten

Übertragungswege zwischen Kommunikationsstationen können unterschiedlich genutzt werden. Das betrifft die Übertragungsrichtung, die Übertragungsbreite, die Synchronisation der Übertragung und die Art der Vermittlung bei Mehrpunktverbindungen.

Bei der *Übertragungsrichtung* wird zwischen Simplex-, Halbduplex- und Duplexbetrieb unterschieden (vgl. Abbildung 2.35.).

**Simplexbetrieb (Einwegbetrieb)** ist Übertragung in eine Richtung, **Halbduplexbetrieb (Wechselbetrieb)** ist zeitversetzte Übertragung in zwei Richtungen und **Duplexbetrieb (Zweiwegbetrieb)** ist gleichzeitige Übertragung in zwei Richtungen.

Im Simplexbetrieb kann eine Station entweder nur senden oder nur empfangen. Diese Betriebsart ist in Ausnahmefällen anwendbar, oder es müssen zwei (gegenläufige) Übertragungswege benutzt werden.

Die *Übertragungsbreite* ergibt sich aus der Anzahl der Bit, die in einem Schritt übertragen werden. Wird in einem Schritt nur ein Bit übertragen, heißt die Übertragung bitseriell, werden mehrere Bit gleichzeitig übertragen, heißt sie bitparallel.

Beim bitseriellen Betrieb ist zwischen den Stationen nur eine Leitung erforderlich. Die (intern parallelen) Bitmuster des Busses werden über eine SIO-Steuereinheit (vgl. Abschnitt 2.2.1.) in einen Strom aufeinanderfolgender Bit umgeformt und auf den Übertragungsweg gebracht. Auf der Empfängerseite erfolgt die Rückverwandlung.

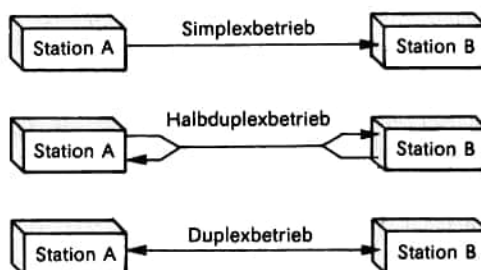


Abbildung 2.35.  
Übertragungsrichtungen

Bei einer bitparallelen Übertragung sind mehrere Leitungen (oder Frequenzbänder) erforderlich. Die Information wird über die PIO-Steuereinheit in Byte- oder Wortbreite zusammengestellt und gleichzeitig übertragen.

Die bitparallele Übertragung ist schneller als die bitserielle, erfordert aber materialaufwendige Übertragungswege. Dieser Nachteil fällt vor allem bei langen Wegen (größeren Entfernungen) ins Gewicht. Deshalb wird für große Entfernungen die bitserielle Übertragung bevorzugt.

Die eindeutige Erkennung der zu übertragenden Information auf der Empfängerseite erfordert eine Abstimmung zwischen Sender und Empfänger. Es wird zwischen Asynchronübertragung und Synchronübertragung unterschieden.

**Asynchronübertragung ist ein Übertragungsverfahren, bei dem die Synchronisation zwischen den Stationen durch Übertragungssteuerzeichen innerhalb des Bitstroms erfolgt.**

Bei der Asynchronübertragung wird der Bitstrom eines Zeichens in ein Startbit (Start of Text, STX) und ein Stoppbit (End of Text, ETX) eingeschlossen und dann gesendet. Diese Übertragung heißt deshalb Start-Stopp-Betrieb. Sie wird bei Übertragungsgeschwindigkeiten bis zu 1200 Bit/s angewendet.

**Synchronübertragung ist ein Übertragungsverfahren, bei dem die Synchronisation zwischen den Stationen außerhalb des Bitstroms durch Taktinformation erfolgt.**

Die zu übertragende Information ist in größeren Übertragungsblöcken zusammengefaßt. Es sind keine Start- und Stoppbit und keine Unterbrechungen zu ihrer Erkennung erforderlich. Die Übertragung ist also

schneller als die Asynchronübertragung, aber technisch aufwendiger.

Bei Mehrpunktverbindungen kann die *Art der Wählvermittlung* (vgl. Abschnitt 1.1.) verschieden sein, in der die Verbindung zwischen zwei Stationen hergestellt wird. Es ist zwischen Leitungs- und Paketvermittlung zu unterscheiden.

**Leitungsvermittlung (Durchschaltevermittlung) ist eine Vermittlungsart, bei der ein Übertragungsweg zwischen zwei Stationen physisch durchgeschaltet wird. Der (geschaltete) Übertragungsweg ist für die Dauer der Verbindung reserviert, unabhängig davon, ob in dieser Zeit Information übertragen wird oder nicht.**

Der (geschaltete) Übertragungsweg ist bei Leitungsvermittlung für andere Teilnehmer blockiert. Wird Information übertragen, so gelangt sie direkt und ohne Zeitverzug zum Empfänger. Allerdings wird im allgemeinen der Übertragungsweg nicht optimal ausgelastet. Dieses Vermittlungsprinzip wird vorwiegend angewendet, wenn wenige Stationen vorhanden sind und große Informationsmengen übertragen werden.

**Paketvermittlung (Speichervermittlung) ist eine Vermittlungsart, bei der die zu übermittelnde Information in (mehrere) Einheiten (Pakete aus Adreß-, Steuer- und Nutzinformation) auf den Übertragungsweg gebracht und dort von Vermittlungsstationen, eventuell nach Zwischenspeicherung, an die Empfängerstation weitergeleitet wird.**

Sende- und Empfangsstation sind bei der Paketvermittlung nur scheinbar (virtuell) verbunden (vgl. Abbildung 2.36.). Dieser Unterschied fällt aber durch die hohe Geschwindigkeit im allgemeinen nicht ins Ge-



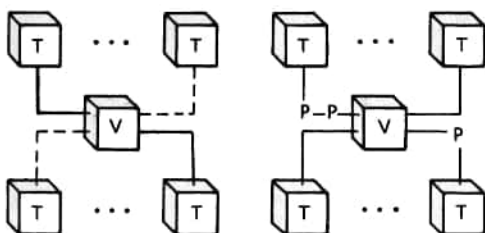


Abbildung 2.36.  
Leitungs- und Paketvermittlung  
(T–Terminalstation, V–Vermittlungsstation,  
– feste Verbindung, P – Paket)

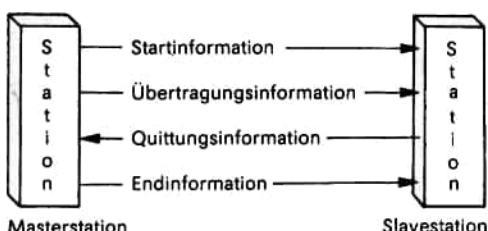


Abbildung 2.37.  
Ablauf einer Master-Slave-Übertragung

wicht. Die Paketvermittlung ermöglicht eine hohe Auslastung des Übertragungsweges. Der technische Aufwand für Übertragungswege, die paketvermittelnd arbeiten, ist höher als für Wege, die nur mit Leitungsvermittlung arbeiten.

### Kommunikationsprotokolle

Ein **Kommunikationsprotokoll** (Übertragungsprotokoll) ist die Beschreibung der Regeln für die Kommunikation zwischen zwei Stationen. Die Regeln umfassen Festlegungen zum Interface (einschließlich der Codierung von Übertragungsinformation) und zu den Übertragungsverfahren.

Das *Interface* (vgl. Abschnitt 2.2.1.4.) wird durch die Übertragungseinrichtung einer Station (vgl. Abbildung 2.31.) realisiert. Es folgt entsprechenden CCITT-Empfehlun-

gen (Comité Consultatif International Télégraphique et Téléphonique, Ausschuß der internationalen Fernmeldevereinigung). Es werden unterschieden:

- V-Empfehlungen für die Übertragung im Fernsprechnet und
- X-Empfehlungen für die digitale Übertragung in Datennetzen.

Eine Norm für die Paketvermittlung ist die CCITT-Empfehlung X.25, die Pakete mit einer Länge von 1024 Byte vorsieht.

Die *Übertragungsverfahren* sind abhängig von den verschiedenen Aufgaben. Im einfachsten Fall dient die Übertragungsverfahren der Kommunikation mit einem peripheren Gerät.

Eine (programmtechnisch realisierte) **Übertragungsverfahren für ein spezielles Gerät** heißt **Gerätetreiber** (Treiber, Driver).

Bei der Master-Slave-Übertragung verläuft die Kommunikation nach den in Abbildung 2.37. dargestellten Schritten.

Die **Master-Slave-Übertragung** ist eine **Kommunikationsverfahren**, bei der eine **Station als Leitstation (Master)** eine andere Station als **Unterstation (Slave)** führt.

Eine Station bewirbt sich bei der Master-Slave-Übertragung um einen Übertragungsweg und wird, nachdem sie ihn von einer Vermittlungsstation erhalten hat, zur Master-Station. Die Master-Station sendet die Adresse einer anderen Station, die zur Slave-Station wird, und danach entweder Übertragungsinformation (wenn die Master-Station an die Slave-Station sendet) oder eine Sendeaufforderung (wenn die Slave-Station an die Master-Station senden soll). Die Slave-Station nimmt je nach Übertragungsrichtung die Information vom Übertragungsweg ab oder bringt sie auf

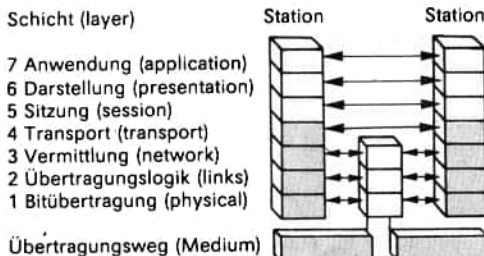


Abbildung 2.38.

ISO/OSI-Schichtenmodell  
 (vereinfachte, nichtgenormte  
 deutsche Bezeichnungen)

den Übertragungsweg. Es werden Nutzinformation, Steuerinformation und Prüfinformation übertragen. Das Ende einer Übertragung wird von der Master-Station mit Steuerinformation (End Of Transmission, EOT) mitgeteilt, danach wird der Übertragungsweg freigegeben. Eine andere Station kann sich als Master-Station anmelden.

In den Kommunikationsablauf sind Zeitkontrollen (Time Out, TO) eingebaut, die Fehler in der Kommunikation mitteilen und die Übertragung unterbrechen können (z. B. wenn eine Station nicht eingeschaltet ist).

Die Kommunikationsprotokolle, nach denen zwei Stationen Information austauschen, müssen exakt übereinstimmen. Zu ihrer Beschreibung und als Empfehlung für Gerätehersteller und Softwareproduzenten wurde ein allgemeines Modell für Übertragungsnetze geschaffen, die mehreren Nutzern offenstehen (öffentliche Netze). Es ist das ISO-Basis-Referenzmodell.

**Das ISO-Basis-Referenzmodell für die Kommunikation offener Systeme (Open Systems Interconnection, OSI) ist ein Schichtenmodell zur Beschreibung der Kommunikation zweier Stationen.**

Im ISO/OSI-Basis-Referenzmodell werden sieben Schichten formuliert (vgl. Abbildung 2.38.).

Die Schichten 1 bis 4 regeln die reine Informationsübertragung (Transportschichten), darunter mechanische und elektrische Parameter, Prozeduren der Übertragung, Kontrollmechanismen und Paketformate. Die Schichten 5 bis 7 beschreiben die Nutzung der Transportmöglichkeit (Anwendungsschichten), darunter die Errichtung und den Abbau von Verbindungen sowie die Steuerung der Kommunikation.

Die Festlegungen des ISO-Basis-Referenzmodells beschränken sich auf die exakte Beschreibung der Schnittstellen. Die technische Realisierung wird offengelassen. Die Standardisierung der Schichten hat den Vorteil, daß Schichten, z. B. die Transportschichten, austauschbar werden, ohne daß sich andere Schichten verändern müssen. Bei einem vollständig entwickelten Übertragungsnetz tritt der Nutzer eines Übertragungsweges nur mit der obersten Schicht in Wechselwirkung.

### 2.3.2. Rechnernetze

Ein System aus mindestens zwei Kommunikationsstationen und einem Übertragungskanal wird als Verbundsystem bezeichnet. Ein Rechnernetz ist eine spezifische Form des Verbundsystems.

**Ein Rechnernetz ist ein Verbundsystem, in dem mindestens zwei Kommunikationsstationen autonome Rechnersysteme sind und in dem jede Station mit jeder Station kommunizieren kann.**

#### Verbundfunktionen

Rechnernetze können folgende Verbundfunktionen realisieren:

- Datenverbund. Der Datenverbund ermöglicht den Zugriff auf alle räumlich

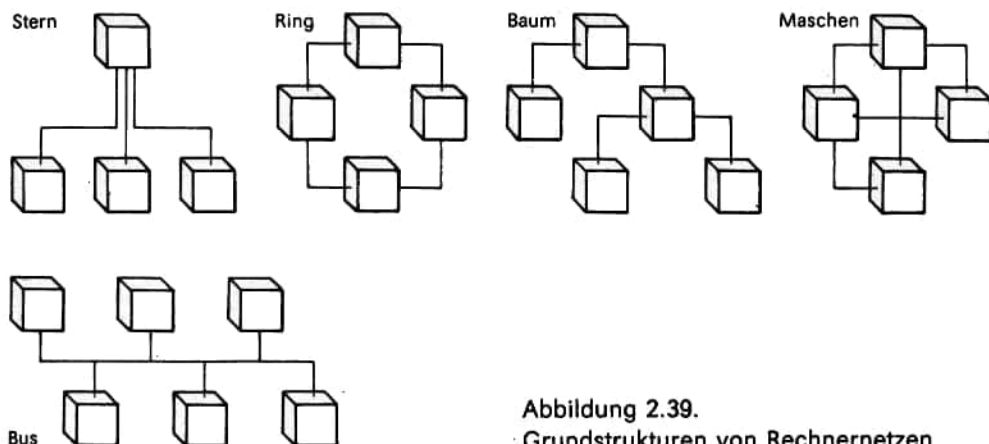


Abbildung 2.39.

Grundstrukturen von Rechnernetzen

im Netz verteilten Datenbestände. Der Vorteil eines Datenverbundes ist die hohe Verfügbarkeit der Daten bei relativ geringem Aufwand für die Speicherung.

- **Ressourcenverbund** (Betriebsmittelverbund). Der Ressourcenverbund ermöglicht die Benutzung aller im Netz vorhandenen Betriebsmittel (Hard- und Software). Alle Terminalstationen besitzen die Möglichkeit, auf spezielle periphere Geräte, wie Plotter, Schnelldrucker, zuzugreifen, obwohl sie davon räumlich getrennt sind.
- **Lastverbund**. Der Lastverbund ermöglicht die Verteilung einer Arbeitsbelastung auf mehrere Rechnersysteme mit den Vorteilen, die sich aus der Arbeitsteilung ergeben. Außerdem ist gesichert, daß die Rechnerleistung auch bei einer Havarie oder während der Wartung zur Verfügung steht.
- **Kommunikationsverbund**. Der Kommunikationsverbund ermöglicht die Mensch-Mensch-Kommunikation zwischen Nutzern verschiedener räumlich getrennter Stationen.

Die Verbundfunktionen bedingen und ergänzen sich teilweise untereinander.

### Topologie von Rechnernetzen

Die Leistungsfähigkeit von Rechnernetzen und der für die Installation und den Betrieb erforderliche Aufwand hängen wesentlich von der räumlichen Anordnung der Kommunikationsstationen, der Netztopologie, ab.

**Die Netztopologie ist die durch die räumliche Anordnung der Kommunikationsstationen und ihre möglichen Direktverbindungen gegebene Struktur eines Rechnernetzes.**

Mögliche Grundstrukturen zeigt Abbildung 2.39.

Bei einem *Sternnetz* sind alle Kommunikationsstationen über eine Zentralstation, die zusätzlich eine Vermittlungsfunktion realisiert, miteinander verbunden. Vorteile dieser topologischen Grundstruktur sind die leichte Erweiterbarkeit des Netzes um Terminalstationen. Der Ausfall einer Terminalstation beeinflusst die anderen Stationen nicht. Nachteilig ist, daß beim Ausfall der Zentralstation die gesamte Kommunikation im Netzwerk unterbrochen wird.

Charakteristisch für ein *Ringnetz* ist, daß jede Kommunikationsstation genau einen Vorgänger und genau einen Nachfolger besitzt. Die Informationsübertragung im Ring ist gerichtet. Die Information wird von Station zu Station geschickt, bis die adressierte Station erreicht ist. Vorteile der Ringtopologie sind die leichte Erweiterbarkeit des Netzes und der geringe Verkabelungsaufwand. Nachteilig sind die höhere Übertragungszeit und die geringere Zuverlässigkeit. Das Netzwerk fällt bereits aus, wenn eine der Stationen ausfällt. Deshalb werden die Stationen durch zusätzliche Verkabelung überbrückt. Im Havariefall wird die Brücke benutzt.

In einem *Baumnetz* erfolgt die Übertragung von Information zwischen zwei Kommunikationsstationen der gleichen Hierarchieebene über Stationen der höheren (vorgelagerten) Hierarchieebene. Das Baumnetz ist ein mehrschichtiges Sternnetz. Es hat die gleichen Vor- und Nachteile wie ein Sternnetz.

In einem *Maschennetz* ist jede Kommunikationsstation mit mindestens zwei, gegebenenfalls sogar mit allen anderen Stationen, verbunden. Die Vorteile eines Maschennetzes sind hohe Übertragungsgeschwindigkeit und hohe Zuverlässigkeit. Der Nachteil besteht in einem hohen Auf-

wand für die Verkabelung und für die Vermittlungssteuerung.

Das *Busnetz* verfügt über eine Sammelleitung (Bus) in Linienform, an die alle Stationen über eine Mediumzugriffseinheit (Medium Access Unit, MAU) angeschlossen sind. Die Information wird von der sendenden Station zusammen mit einer Adresse auf den Bus gelegt. Der Bus wird von allen Stationen »abgehört«. Die adressierte Station nimmt die Information vom Bus, alle übrigen Stationen bleiben passiv.

Rechnernetze werden nach der geographischen Entfernung zwischen den Kommunikationsstationen und dem Übertragungsweg in Fernnetze und lokale Netze klassifiziert.

### Fernnetze

**Fernnetze (globale Netze, Weitverkehrsnetze, Wide Area Network, WAN)** sind **Rechnernetze**, in denen Kommunikationsstationen geographisch weit voneinander entfernt sind (mehr als 10 km) und die öffentliche (aus der Sicht einer Institution externe) Übertragungswege benutzen.

Öffentliche Übertragungswege sind das Fernsprechnetz, das Telexnetz, das Datennetz und perspektivisch ein integriertes

Tabelle 2.8.

Parameter öffentlicher Netze (L – Leitungsvermittlung, P – Paketvermittlung)

Parameter	Telexnetz	Fernsprechnetz	Datennetz	
			L	P
Übertragungsgeschwindigkeit	50 Bit/s	max. 4,8 KBit/s	max. 64 KBit/s	max. 48 KBit/s
Signalform	digital	analog	digital	digital
Typische Betriebsart	Halbduplex/asynchron	Duplex/synchron	Duplex/synchron	Duplex/synchron
Bitfehlerwahrscheinlichkeit	$10^{-5} \dots 10^{-4}$	$10^{-4} \dots 10^{-2}$	$10^{-6} \dots 10^{-5}$	$10^{-6} \dots 10^{-5}$

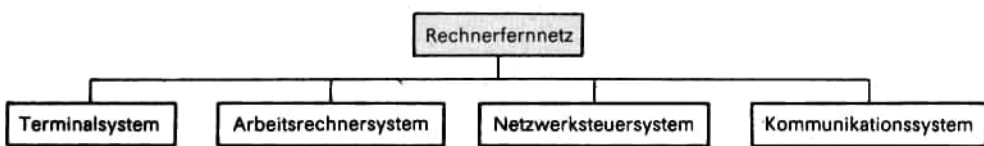


Abbildung 2.40.  
Komponenten eines Fernnetzes

Netz, das dem Nutzer zugleich Dienstleistungen bietet (Integrated Service Digital Network, ISDN). ISDN ist ein integriertes digitales Netz, das einheitliche digitale Vermittlungseinrichtungen und Übertragungswege benutzt sowie Dienste zur Übertragung von Sprache, Daten, Bild und Text anbietet.

Öffentliche Übertragungswege werden auch grenzüberschreitend genutzt. Für die Online-Kommunikation wichtige Parameter öffentlicher Netze enthält Tabelle 2.8.

Rechnerfernnetze lassen sich unter logisch-funktionellen Gesichtspunkten in Netzkomponenten gliedern (vgl. Abbildung 2.40.).

Das *Terminalsystem* realisiert die (externe) Kommunikation mit der Umgebung. Es umfaßt die Gerätetechnik zur Ein- und Ausgabe von Information sowie Schnittstellenwandler zu den anderen Komponenten des Rechnernetzes.

Das *Arbeitsrechnersystem* verarbeitet Information. In Fernnetzen sind Arbeitsrechner in der Regel leistungsfähige Großrechner, die die vom Nutzer gestellten Aufträge arbeitsteilig realisieren. Ein Arbeitsrechner entspricht einer Endeinrichtung, die vorwiegend Verarbeitungsfunktionen wahrnimmt. Es können aber auch Spezialrechner für bestimmte Serviceleistungen, wie Datenverwaltung und Drucksteuerung, im Rechnernetz sein.

Das *Netzwerksteuersystem* steuert das Zusammenwirken der Komponenten eines Rechnernetzes. Es reserviert die erforderlichen Ressourcen und steuert die Priorität sowie die Ordnungsmäßigkeit der Ressour-

cennutzung. Die Festlegungen werden dem Netzwerksteuersystem mit Hilfe einer Netzwerksteuersprache (Network Control Language, NCL) mitgeteilt, die dann durch die (Netz-)Software interpretiert und ausgeführt werden.

Das *Kommunikationssystem* realisiert den Informationsaustausch zwischen den Kommunikationsstationen. Es umfaßt die Übertragungseinrichtungen der Terminalstationen, die Vermittlungsstationen und die Übertragungswege. Vermittlungsstationen können selbst leistungsfähige Rechner sein. Sie werden als Kommunikationsrechner (Vermittlungsrechner, Netzknoten, Vorrechner) bezeichnet.

Neben Fernnetzen gewinnen auch lokale Netze an technischer und wirtschaftlicher Bedeutung.

### Lokale Netze

**Lokale Netze (Local Area Network, LAN)** sind Rechnernetze, in denen unabhängige Kommunikationsstationen auf eng begrenztem geographischem Gebiet (bis 10 km) einen nichtöffentlichen (innerbetrieblichen) Übertragungsweg hoher Bandbreite nutzen.

Ein lokales Netz mit verschiedenartigen Kommunikationsstationen zeigt Abbildung 2.41.

Die Endeinrichtungen der Kommunikationsstationen in lokalen Netzen sind (vgl. Abschnitt 2.3.1.) Zentraleinheiten verschiedener Rechner, Ein- und Ausgabegeräte sowie externe Speicher. Die Übertragungseinrichtung wird durch eine Netzzugangs-

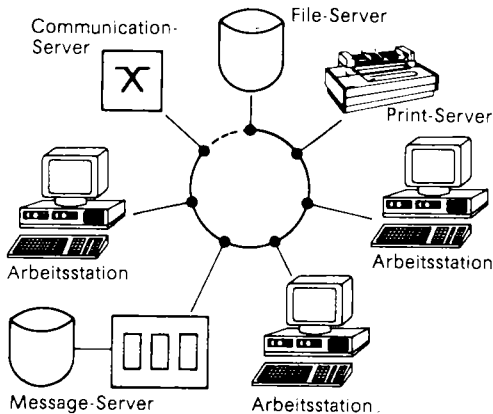


Abbildung 2.41.  
Mögliche Konfiguration  
eines lokalen Netzes

einheit gebildet, die als *LAN-Controller* bezeichnet wird. Der Controller wird in einen Erweiterungsschleitz (Slot) des Personalcomputers gesteckt. Der Anschluß an das Übertragungsmedium erfolgt im allgemeinen durch einen *Transceiver* (zusammengesetzt aus den Wörtern *TRANSMITTER*, Sender, und *RECEIVER*, Empfänger), der die Mediumzugriffseinheit bildet. Kennzeichnend für lokale Netze sind Übertragungsgeschwindigkeiten zwischen 100 KBit/s und 300 MBit/s bei einer niedrigen Fehlerwahrscheinlichkeit von  $10^{-9}$ . Die Terminalstationen eines lokalen Netzes sind entweder Arbeitsstationen oder Serverstationen.

**Arbeitsstationen sind (funktionelle) Komponenten lokaler Netze, die zur Erfüllung von Nutzeraufträgen Dienste des Netzes anfordern und die Kommunikation mit dem Nutzer realisieren.**

Intelligente (d. h. mit einem eigenen Prozessor ausgestattete) Arbeitsstationen sind z. B. Personalcomputer unterschiedlicher Leistungsklassen mit eigener peripherer Ausstattung.

**Serverstationen (Server) sind (funktionelle) Komponenten lokaler Netze, die spezifische Dienste des Netzes für die kollektive Nutzung durch Arbeitsstationen bereitstellen.**

Arbeitsstationen erteilen Aufträge, Server realisieren sie. Es gibt folgende Arten von Servern:

- File-Server für die Verwaltung und den Zugriff auf große externe Speicher,
- Print-Server zur leistungsfähigen Druckerausgabe,
- Message-Server zur Annahme, Sammlung und Quittierung von Mitteilungen,
- Communication-Server für den Übergang zu anderen lokalen Netzen oder zum Fernnetz,
- Application-Server, besonders für Hochgeschwindigkeitsrechnungen,
- Authentification-Server zur Überwachung des Zugangs zum lokalen Netz.

Bei Communication-Servern wird zwischen Gateways und Bridges unterschieden. Gateways (interconnect server) ermöglichen den Anschluß an Netze mit anderer Funktionsweise, Bridges (Brücken) übernehmen den Anschluß an andere Netze mit gleicher Funktionsweise.

Server sind entweder ausschließlich für den jeweiligen Zweck ausgelegte Geräte (dedizierte Server), oder diese Aufgabe wird bevorzugt durch ein speziell konfiguriertes Rechnersystem ausgeführt.

Es gibt eine Vielzahl von Konzepten für lokale Rechnernetze. Sie unterscheiden sich hinsichtlich

- des verwendeten Übertragungsmediums, wie verdrehtes Kupferkabel, Koaxialkabel (Basis- und Breitband) oder Lichtwellenleiter,
- der verwendeten Signalforn, digital in Basisbandsystemen oder analog in Breitbandsystemen,

- der Topologie (Bus-, Baum-, Ring- oder Sternnetz),
- des Mediumzugangsverfahrens (z. B. Carrier Sense Multiple Access with Collision Detection, CSMA/CD).

Beim CSMA/CD-Verfahren »hören« die Controller der einzelnen Stationen passiv den Transfer auf dem Übertragungsmedium ab. Die Information wird auf dem Übertragungsmedium in Form von Blöcken (Frames) übertragen. Erkennt ein Controller am Anfang eines Frames seine Stationsadresse, wird das Frame empfangen und der Station übergeben. Ist Information zu senden, übergibt die Station diese an den Controller. Der Controller wartet, bis das Übertragungsmedium frei ist und sendet dann. Eine eventuelle Kollision wird durch die Mediumzugriffseinheit erkannt.

Lokale Netze sind in der Lage, relativ viele Stationen miteinander zu verbinden. Die Zahl ist herstellerabhängig.

ROLANET 1, ein lokales Netz des VEB Kombinat Robotron, ermöglicht bis zu 1024 Anschlüsse bei einer Übertragungsgeschwindigkeit bis zu 500 KBit/s. Die tatsächliche Anzahl der Anschlüsse richtet sich nach den Anforderungserfordernissen. Bekannt sind lokale Netze mit 5 bis 100 Anschlüssen.

Die breite Nutzung lokaler Netze ist eng mit der zunehmenden Produktion und dem Einsatz von Personalcomputern verbunden. Lokale Netze sind eine günstige Möglichkeit, die Vorzüge der Großrechenstechnik (umfassende Betriebsmittel, speziell ausgebildetes Bedienpersonal, hoher Durchsatz) mit denen der Personalcomputer (kurze Antwortzeiten im direkten Dialog zwischen Mensch und Maschine, Zugschnitt der Automatisierungslösung auf den Arbeitsplatz, autonome Nutzbarkeit für die arbeitsplatzspezifische Aufgabe) zu verbinden.

Einsatzgebiete lokaler Netze sind vor allem:

- Leitung, Büro und Verwaltung,
- Vermittlung von Dienstleistungen (z. B. Platzreservierung),
- komplexe CAD/CAM-Anwendungen,
- Rechenzentren.

Mit den Verbundfunktionen Daten-, Ressourcen- und Lastverbund wird durch lokale Netze vor allem erreicht:

- Erhöhung der Datenkonsistenz, d. h. der inhaltlichen Übereinstimmung der Daten durch Sicherungsmaßnahmen in zentralen Datenbanken,
- schnelle Verfügbarkeit von Daten durch Wegfall des Datenträgertransports zwischen autonomen Rechnersystemen,
- geringerer Bedarf an Betriebsmitteln, speziell hochwertiger Ein- und Ausgabe- sowie Speicherperipherie,
- Zugang zu öffentlichen Netzen.

Fernnetze und lokale Netze ermöglichen im Zusammenwirken die *verteilte Informationsverarbeitung*. Ziel ist die umfassende Automatisierung der Verarbeitung, Übertragung und Speicherung von Information bei der Planung und Durchführung des volkswirtschaftlichen Reproduktionsprozesses.

### 2.3.3. Telekommunikationsdienste

Telekommunikation unterstützt die verteilte Informationsverarbeitung.

**Telekommunikation ist die Übertragung von Information zwischen Kommunikationsstationen unter Nutzung von Dienstleistungsfunktionen (Telekommunikationsdiensten) eines (öffentlichen) Fernnetzes.**

Die Information kann als Folge von Schrift-, Bild- oder Akustikzeichen vorliegen. Entsprechend den volkswirtschaftlichen Aufgabenstellungen in der DDR und in Durchsetzung des Komplexprogramms

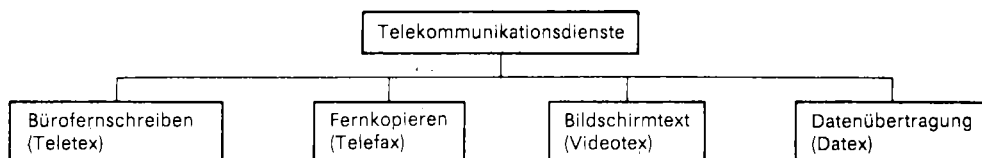


Abbildung 2.42.

Einteilung der Telekommunikationsdienste  
(CCITT-Bezeichnungen in Klammern, »x« für exchange)

des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW bis zum Jahre 2000 werden die in Abbildung 2.42. dargestellten Telekommunikationsdienste eingesetzt.

Zu den Telekommunikationsdiensten zählen außerdem der Fernschreibdienst (Telex) und der Fernsprechdienst. Sie sind jedoch überwiegend nicht Mittel der automatisierten Informationsverarbeitung. Die Bereitstellung von Fernsprech- und Telexleitungen für die Informationsübertragung zwischen Rechnersystemen wird nicht als gesonderter Telekommunikationsdienst angesehen.

Videotex ist von Videotext zu unterscheiden. Videotext nutzt die Austastzeilen des Elektronenstrahls zur Übermittlung von Information in eine Richtung, nämlich vom Fernsehsender zum Fernsehempfänger (die elektronische Zeitung). Im Unterschied zu Videotext ist Videotex interaktiv, d. h., es erfolgt Informationsübertragung in beiden Richtungen.

### Bürofernschreiben

**Bürofernschreiben ist ein Telekommunikationsdienst zur zeichenweisen Übertragung von Textnachrichten.**

Bürofernschreiben ist ein weiterentwickelter Telex-Dienst. Es sichert eine schnellere Übertragung (etwa 2400 Bit/s) in fernsprechtypischen Leitungen. Die Übertragungszeit für einen Brief im Format A4 mit

1500 Anschlägen beträgt etwa 5 s und ist damit wesentlich kürzer als im Telex-Dienst. Die Übertragung erfolgt vom Speicher einer Kommunikationsstation A zum Speicher einer Kommunikationsstation B (elektronischer Briefkasten, mailbox). Der Übertragungsinformation ist eine codierte Adresse des Teilnehmers B vorangestellt.

Kommunikationsstationen sind Personalcomputer oder Textverarbeitungssysteme, die mit einem zusätzlichen Empfangsspeicher ausgestattet sind. Das Eintreffen der Information wird auf dem Bildschirm der Kommunikationsstation angezeigt, ohne daß deren laufender Betrieb unterbrochen wird. Der Empfänger kann zwischen Bildschirm- oder Druckerausgabe wählen.

Der Vorteil des Bürofernschreibens liegt in der höheren Übertragungsgeschwindigkeit, in der Übertragungsmöglichkeit bis zum Arbeitsplatz, in besserer Integration in die Arbeitsprozesse und in der möglichen Papiereinsparung.

### Fernkopieren

**Fernkopieren (Faksimileübertragung) ist ein Telekommunikationsdienst zur Übertragung von Schrift- und Grafikvorlagen.**

Die Vorlagen (Grafiken, Zeichnungen, Formulare, unterzeichnete Verträge, Geschäftsbriefe) werden in der sendenden Kommunikationsstation optisch abgetastet



und die Bildpunkte an die empfangende Kommunikationsstation übertragen.

Die Kommunikationsstationen müssen über spezielle Geräte (Fernkopiergeräte) verfügen. Es gibt verschiedene Gerätetypen, deren Übertragungsgeschwindigkeit für eine A4-Seite zwischen 4 bis 6 Minuten (analog arbeitende Geräte) und 20 bis 60 Sekunden (digital arbeitende Geräte) liegt. Die Geräte arbeiten mit Spezialpapier.

Fernkopieren ist aufwendiger und bedeutend langsamer als Bürofernreiben, da zahlreiche Bildpunkte zu übertragen sind. Es wird vor allem für CAD/CAM-Anwendungen und im Vertragswesen eingesetzt.

### Bildschirmtext

**Bildschirmtext (Btx)** ist ein Telekommunikationsdienst, bei dem ein Teilnehmer Textinformation abrufen, Verarbeitungsleistungen von Rechnersystemen oder Dienstleistungen von Anbietern in Anspruch nehmen oder Textinformation an andere Teilnehmer weitergeben kann.

Die Komponenten eines Bildschirmtextsystems zeigt Abbildung 2.43.

Der Grundgedanke besteht in der Bereitstellung zentral gespeicherter Information (in Textform) und von Dienstleistungen von Anbietern für andere Institutionen oder private Haushalte (Teilnehmer).

Anbieter sind z. B.

- Geld- und Kreditinstitute mit Auskünften über Kontostände und Zahlungsdiensten,
- Verkehrsbetriebe, Reisebüros und Hotels mit Fahrplänen, Leistungsangeboten und Buchungsmöglichkeiten,
- Kultur- und Gastspieldirektionen mit Spielplänen, Platz- und Buchungsangeboten,
- Buch- und Patentbibliotheken, Rechts-

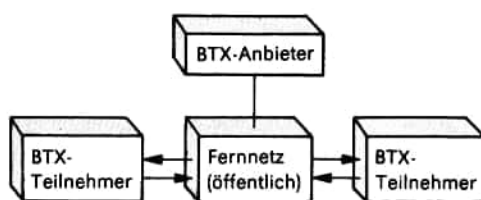


Abbildung 2.43.

Grundstruktur eines Bildschirmtextsystems

träger von Programm- und Methodenbanken mit Auskunfts- und Nachnutzungsangeboten,

- Hochschulen, Universitäten mit Weiterbildungs- oder Fernstudienkursen.

Die Teilnehmer benötigen ein Rechnersystem (z. B. Personalcomputer), ein lokales Netz mit Communication-Server oder ein Sichtgerät (z. B. Fernsehbildschirm) mit Zusatzeinrichtungen (Telefon, Decoder, Tastatur) und den Zugang zum Fernnetz.

Die Nutzung von Bildschirmtext in der DDR erfolgt zunächst vor allem für zentrale Informationsdienste, wie Patentauskünfte, Standards, Datenbanken über Materialkennzahlen, Softwareangebote, CAD-Modelldaten und ähnliches.

### Datenübertragung

**Datenübertragung ist ein Telekommunikationsdienst zur Übertragung von Information in rechnerinterner Darstellung.**

Kommunikationsstationen sind Rechnersysteme oder (lokale) Netze von Betrieben und Institutionen, die im Rahmen der verteilten Informationsverarbeitung Information austauschen. Datenübertragung erfolgt z. B. zur zentralen, territorialen oder betrieblichen staatlichen Leitung und zur staatlichen Berichterstattung. Dieser Dienst wird erheblich an Bedeutung gewinnen.

Die Weiterentwicklung der Telekommunikationsdienste ist mit umfangreichen Investitionen in der gesamten Volkswirtschaft

verbunden, da sie umfassend nur über ISDN (vgl. Abschnitt 2.3.2.) verwirklicht werden können. Das Komplexprogramm des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW sieht die Schaffung eines einheitlichen Systems der digitalen Informationsübertragung und eine Vereinheitlichung der technischen Mittel zu seiner Nutzung vor. Bis zur Jahrtausendwende werden alle Dienste, die in unterschiedlichen Netzen ablaufen, über ein universelles Breitbandkommunikationsnetz digitalisiert vermittelt. Der Einsatz eines solchen digitalen Systems bringt erhebliche Vorteile. Die Kosten für Modems entfallen, die Übertragungsgeschwindigkeit wird größer, und gleichzeitig wird die Übertragungsqualität verbessert. Außerdem ist der Anschluß einer Kommunikationsstation ohne zusätzliche Verkabelung und mit einem Minimum an Hardware möglich. Sprachkommunikation zwischen Menschen und Online-Kommunikation zwischen Rechnersystemen erfolgen gleichzeitig über eine Zweidrahtleitung, die nur einen Anschluß der Anlage benötigt. Die Online-Kommunikation ist dann ein zusätzliches Leistungsmerkmal des Fernnetzes, das volkswirtschaftlich nur wenig Mehraufwand verursacht.

### Übungen

1. Welches Rechnersystem (mit welcher Konfiguration) empfehlen Sie zur Verwaltung einer Bibliothek mit einem Buchbestand bis zu 50 000 Exemplaren?
2. Welche Kriterien benutzen Sie bei der Auswahl zwischen den Ausgabegeräten und Datenträgern
  - Bildschirm und Drucker,
  - Diskette und Magnetbandkassette?
3. Welche Perspektive ordnen Sie der Tastatur in der Mensch-Maschine-Kommunikation zu? Wie beurteilen Sie Ergänzungen und Alternativen?
4. Rechnen Sie in KByte um: 8 192 Bit, 256 KBit, 1 MBit, 65 535 Byte, 1 MByte!
5. Schildern Sie (ökonomische) Erfordernisse und Möglichkeiten zur Nutzung der Verbundfunktionen eines lokalen Rechnernetzes in einem Betrieb!
6. Wie schätzen Sie die (laufenden) volkswirtschaftlichen Aufwands- und Nutzensfaktoren der Telekommunikation ein?

### 3.

# Betriebssysteme und Nutzungstechnologien

---

## 3.1. Basissoftware

### 3.1.1. Klassifikation

Aus der Sicht des Nutzers sind Rechnersysteme universelle Arbeitsmittel zur Automatisierung von Informationsprozessen. Um sie den konkreten Einsatzerfordernissen anpassen zu können, sind die Rechnersysteme vom Gerätehersteller mit der Eigenschaft ausgestattet, Befehle zu interpretieren und auszuführen. Die Befehle sind Bestandteil der Software.

Für das Leistungsvermögen eines Rechnersystems spielt die Basissoftware eine besonders wichtige Rolle (vgl. Abschnitt 1.3.1.). Eine Klassifikation der Basissoftware nach der Zweckbestimmung in Anlehnung an gesetzliche Regelungen zur Planung, Abrechnung und Bilanzierung von Software<sup>1</sup> zeigt Abbildung 3.1.

Der Entwicklung von Basissoftware liegt zugrunde, daß ein Teil der Vorgänge im Rechnersystem während der Ausführung von Anwendersoftware und bei der Schaffung derselben unabhängig vom konkreten Anwendungsfall sind. Zu solchen Vorgängen gehören die Überwachung von Ein- und Ausgabeoperationen, der Transport innerhalb der Zentraleinheit, das Kopieren von Datenbeständen und das Übersetzen eines in einer höheren Programmiersprache geschriebenen Programms in ein durch den Computer ausführbares Maschinenprogramm (vgl. Abschnitt 4.3.1.).

## Betriebssysteme

Innerhalb der Basissoftware nehmen Betriebssysteme eine zentrale Stellung ein.

**Das Betriebssystem (Operating System, OS) ist die Gesamtheit der Basissoftware zur Organisation und Steuerung der Programmausführung, der Ressourcen- und Dateiverwaltung sowie für allgemeine Dienstleistungen.**

Das Betriebssystem wird vom Gerätehersteller mit dem Rechnersystem geliefert und ist nur für dieses System bestimmt. Es besteht mindestens aus einem Steuerprogramm und einigen Dienstprogrammen.

Das *Steuerprogramm* überwacht die Ausführung aller (anderen) Programme und stellt ihnen Befehlsfolgen für grundlegende Aufgaben zur Verfügung. Das Steuerprogramm ist nach dem Ladevorgang im Hauptspeicher ständig anwesend (resident), andere Software wechselt entsprechend der Aufgabe (transient). Das Steuerprogramm schiebt sich als Schicht zwischen Hardware und das jeweils auszuführende Programm (vgl. Abbildung 3.2.).

Für spezielle Aufgaben kann das Steuerprogramm durch Zusatzkomponenten erweitert sein (z. B. SCP-GX Grafikerweiterung SCP 1700). Durch seine zentrale Stellung wird das Steuerprogramm als Be-

<sup>1</sup> Anordnung vom 27. 1. 1989 über die Planung, Bilanzierung und Abrechnung von Software. In: GBl. I/1989 Nr. 6 S. 100.

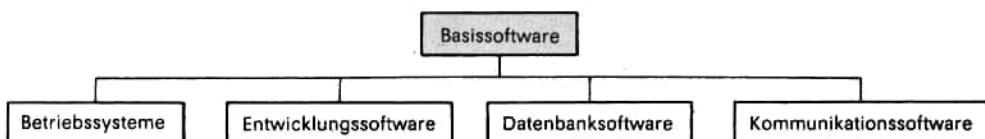


Abbildung 3.1.  
Klassifikation von Basissoftware (Auswahl)

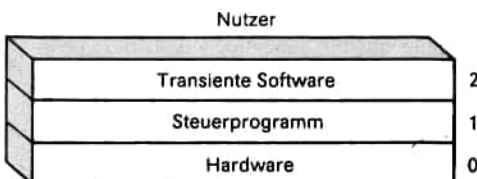


Abbildung 3.2.  
Erweitertes Schichtenmodell  
der Nutzung eines Rechnersystems

etriebssystemkern oder Exekutive bezeichnet und in der Praxis begrifflich mit Betriebssystem gleichgesetzt. Die Verwendung des Begriffs Betriebssystem ist nicht einheitlich.

#### Dienstprogramme

Dienstprogramme führen allgemeine Dienstleistungen aus. Dazu gehören

- Dateidienste, wie Dateien anlegen, löschen, kopieren, vergleichen, sortieren, mischen, schützen,
- Datenträgerdienste, wie Disketten formatieren, Datenträger kopieren, Datenträger prüfen,
- Konfigurationsdienste, wie Generieren verschiedener Zeichensätze (z. B. lateinisches und kyrillisches Alphabet), Speicheraufteilung, Anlegen eines virtuellen Laufwerks (RAM-Disk), Steuerung des Ladevorgangs beim Einschalten des Rechnersystems,
- Dienste zur Systemverwaltung, wie Nachweisführung der Ressourcennutzung, Auslastungsstatistik.

Die innere Struktur eines Betriebssystems wird im Abschnitt 3.1.2. behandelt.

#### Entwicklungssoftware

Entwicklungssoftware rationalisiert und erleichtert die Entwicklung von Software (vgl. Kapitel 4.). Ihre Klassifikation ist aus Abbildung 3.3. ersichtlich.

**Entwicklungssoftware ist die Gesamtheit programmtechnischer Mittel zum Spezifizieren, Entwerfen, Implementieren und Testen von Software.**

*Programmentwicklungssysteme* (Program Development System, PDS) sind meistens sprachorientiert. Zu den Programmentwicklungssystemen gehören z. B. Entwurfsgenerator, Editor, Übersetzer (Assembler; Compiler, Interpreter), Linkage Editor, Debugger und Hilfsprogramme (vgl. Kapitel 4.).

*Programmbibliotheken* beinhalten Bibliotheksmoduln oder Boxen für die Mehrfachverwendung.

**Bibliotheksmoduln sind vorgefertigte Teilprogramme im Quell-, Objekt- oder Maschinencode, die vom Entwickler (unverändert) in die Software übernommen werden können.**

Moduln gibt es z. B. für mathematische Methoden und Modelle, statistische Verfahren, nichtnumerische Operationen und für CAD/CAM-Funktionen.

**Boxen sind vorgefertigte Programme im Quellcode, die über eine definierte Schnittstelle der Aufgabe angepaßt werden.**

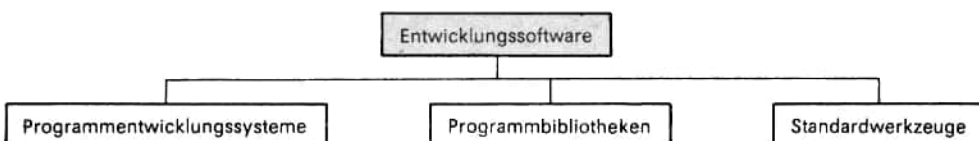


Abbildung 3.3.  
Klassifikation von Entwicklungssoftware

Boxen enthalten Programmteile, die sich entsprechend der jeweiligen Aufgabe ändern. Die Veränderung des Quellcodes der Boxen bezieht sich in der Regel auf die Datenschnittstelle. Boxen gibt es zum Beispiel für Sortieraufgaben (SORTBOX), das Anlegen und Auswerten von Dateien (DATEI-BOX) sowie die Arbeit mit Bildschirm- und Druckergrafik (GRAFIKBOX).

Der Einsatz von Programmiersystemen und Programmbibliotheken zur Entwicklung von Basis- oder Anwendersoftware erfordert spezielles Wissen. Im Unterschied dazu lassen sich Standardwerkzeuge ohne umfangreiche Kenntnisse vom Endnutzer verwenden.

*Standardwerkzeuge* (Endnutzerwerkzeuge, enduser tools) sind programmtechnische Mittel, die es dem Nutzer ohne Programmierkenntnisse gestatten, ein Rechnersystem für arbeitsplatzbezogene Aufgaben zu nutzen. Standardwerkzeuge gibt es jeweils einzeln für die Textverarbeitung, die Tabellenkalkulation, die Datenverwaltung und die Grafikausgabe oder als integriertes Paket. Beispiele sind die vom VEB Kombinat Robotron vertriebenen Erzeugnisse TP, RE-DABAS, ARIADNE und MULTICOMP. Einige Softwarewerkzeuge bilden Software-rahmen.

Ein **Softwarerahmen (Shell)** ist ein Standardwerkzeug mit gleichbleibender Nutzerschnittstelle, das durch Eingabeinformation aufgabenbezogen gefüllt werden kann und definierte Leistungen anbietet.

Shells existieren z. B. für Expertensysteme. In einen Rahmen für Expertensysteme lassen sich für unterschiedliche Aufgaben Fakten und Regeln eingeben sowie Fragen stellen. Es ist keine Programmierung erforderlich.

Die Abgrenzung zwischen Standardwerkzeugen und Standardsoftware (vgl. Abschnitt 4.1.) ist nicht immer eindeutig. Wichtige Standardwerkzeuge werden im Kapitel 5. behandelt.

### Datenbanksoftware

Für die Verwaltung großer Datenbestände werden Datenbanken genutzt.

**Datenbanksoftware ist Basissoftware zum Aufbau, zur Manipulation, zur Sicherung und zur Kontrolle einer Datenbank.**

Datenbanksoftware für ein konkretes System wird als Datenbankbetriebssystem bezeichnet. Häufig benutzte Datenbankbetriebssysteme in der DDR sind z. B. DBS/R, TOPAS und MIMER. Datenbankbetriebssystem und Datenbasis bilden in ihrer Gesamtheit ein Datenbanksystem.

### Kommunikationssoftware

**Kommunikationssoftware ist Software zur Steuerung der Informationsübertragung sowie zur Steuerung von Rechnernetzen.**

Zur Kommunikationssoftware gehören Gerätetreiber und Softwarepakete, wie z. B. SCP-NET für das lokale Netz ROLANET 1.

### 3.1.2. Struktur und Arbeitsweise von Betriebssystemen

Betriebssysteme beeinflussen wesentlich, in welchem Maße die potentiellen Eigenschaften eines Rechnersystems für den Anwender nutzbar werden. Die Betriebssysteme verschiedener Rechnersysteme für allgemeine und für spezielle Aufgaben unterscheiden sich erheblich. Das Grundmodell eines Betriebssystems in Schichtendarstellung zeigt Abbildung 3.4.

Das Steuerprogramm besteht aus der (physischen) Ein- und Ausgabesteuerung peripherer Geräte (Schicht 1.1), der Dateiverwaltung (Schicht 1.2) und der Kommandoverwaltung (Teilschicht 1.3, bei Großrechenanlagen die Auftrags- und Ablaufsteuerung).

#### Ein- und Ausgabesteuerung

Die Ein- und Ausgabesteuerung setzt unmittelbar auf die Hardwareebene auf und ist deshalb in jedem Betriebssystem abhängig von der Architektur und Konfiguration des konkreten Rechnersystems. Gegebenenfalls ist auch die Hauptspeicheraufteilung zu steuern.

Wichtiger Bestandteil des Ein- und Ausgabesystems sind Gerätetreiberprogramme, die die Übertragung von Information von und zu peripheren Geräten oder einer

Schnittstelle für die Fernübertragung steuern. Ändert sich die Peripherie (z. B. statt Drucker mit serieller Schnittstelle Drucker mit paralleler Schnittstelle) oder die Konvention für die Fernübertragung, so kann eine Änderung des Betriebssystems (des Steuerprogramms) erforderlich sein. Programme zur Ein- und Ausgabesteuerung benötigen ständig Speicherplatz. Deshalb enthält ein Steuerprogramm nur die Komponenten, die tatsächlich benötigt werden. Die Anpassung an eine konkrete Konfiguration des Rechnersystems wird als *Installation* bezeichnet. Die Gerätehersteller stellen dazu spezielle Programme zur Verfügung, die die Installation unterstützen. Außerdem werden geräteabhängige Module geliefert, die mit den Installationsprogrammen in das Steuerprogramm eingebunden werden können.

#### Dateiverwaltung

Die Dateiverwaltung steuert das Lesen, Schreiben und Verwalten von Dateien (vgl. Abschnitt 1.3.1.) auf externen Datenträgern. Die Dateien erhalten Namen. Die Restriktionen für Dateinamen richten sich nach dem Betriebssystem. Sie bestehen z. B. bei Betriebssystemen für Personalcomputer aus einem Basisnamen mit maximal acht Zeichen und einem Typnamen mit maximal drei Zeichen, z. B.:

```
ARTIKEL.DAT
SPEZIAL.TXT
INVENTUR.IST
```

Der Basisname von Dateien kann vom Nutzer unter Beachtung einiger Restriktionen festgelegt werden. Da auch Programme, darunter Teile des Betriebssystems, in Dateien verfügbar gehalten werden, sind einige Typnamen für das Betriebssystem reserviert (z. B. COM, EXE, BAK, SUB, BAT). Basisname und Typname sind durch einen Punkt getrennt.

Auf Datenträgern mit nur sequentiell

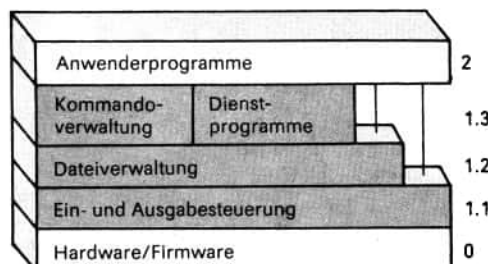


Abbildung 3.4.  
Schichtenmodell eines Betriebssystems  
(1.1–1.3)

Zugriff (z. B. Magnetbänder) werden die Dateien nacheinander gespeichert. Am Anfang des Aufzeichnungsbereiches steht eine Dateikennung, die auch den Namen enthält. Dateien auf Datenträgern mit möglichem wahlfreiem Zugriff (z. B. Disketten) werden über ein Verzeichnis (directory) verwaltet. Es befindet sich auf einem für den Nutzer nicht direkt zugänglichen Teil der Diskette.

Das Verzeichnis kann hierarchisch organisiert sein. In diesem Fall ist es möglich, Verzeichnisse in Unterverzeichnisse (sub-directories) zu unterteilen. Unterverzeichnisse können erneut Unterverzeichnisse enthalten. Jedes Verzeichnis/Unterverzeichnis erhält einen Namen. Dadurch, daß das Gerät bezeichnet wird, in dem sich der Datenträger befindet, die hierarchischen Verzeichnisnamen nacheinander aufgezählt werden und der Name der Datei genannt wird, kann jeder Datenbestand exakt identifiziert werden. Die Dateiorganisation des gesamten Datenträgers bildet eine Baumstruktur.

Das Betriebssystem kann durch die Art und Weise der Eintragung des Dateinamens in ein Verzeichnis zugleich Schutzmaßnahmen für eine Datei realisieren. Dazu werden die Bitmuster von Namenszeichen oder ein spezielles Attributbyte mit Dienstprogrammen manipuliert. Schutzattribute sind z. B. R/O (Read Only), SYSTEM und HIDDEN. Die Manipulation von Dateien mit Schutzattributen unterliegt Einschränkungen. Eine Datei mit R/O-Attribut kann z. B. nur gelesen werden. Das (irrtümliche) Überschreiben ist unmöglich. Grundsätzlich lassen Betriebssysteme auch einen Zugriffsschutz mit einem vom Nutzer festgelegten Codewort zu. Außerdem können Zeit- und Datumseintragungen erfolgen, die die letzte Dateimanipulation festhalten.

### Laden des Steuerprogramms

Nach dem Einschalten eines Rechnersystems wird ein fest mit der Hardware verbundenes (PROM-realisiertes) Umladeprogramm (Anfangslader) gestartet. Ein solches hardwarerealisiertes Programm ist eine Zwischenform zwischen Hardware und Software und wird als *Firmware* bezeichnet.

Der Anfangslader aktiviert ein Systemladeprogramm (Systemlader), das dann das Steuerprogramm von einem Datenträger (Systemdatenträger, bei Personalcomputern Systemdiskette) in den Hauptspeicher transportiert und startet. Dabei überschreibt sich der Systemlader selbst (Bootstrap-Verfahren). Der Systemlader gilt allgemein als Teil des Betriebssystems.

Nach dem Bootstrap meldet sich das Steuerprogramm auf dem Bediengerät mit seinem Namen, der Schutzeintragung des Herstellers (Copyright) und einer Versionsnummer (vgl. Abbildung 3.5.).

Nach dem Start ist die Schicht Kommandoverwaltung aktiv.

### Kommandoverwaltung

Die Kommandoverwaltung erwartet ein Kommando des Nutzers und steuert danach dessen Ausführung. Kommando ist ein Grundbegriff der Kommunikation zwischen Nutzer und Betriebssystem.

### Das Kommando ist ein Auftrag des Nutzers an das Betriebssystem zur Ausführung einer definierten Befehlsfolge.

Das Kommando kann ein Kommandowort mit Kommandoparametern oder ein Steuerzeichenkommando sein.

Ein *Kommandowort mit einem Kommandoparameter* ist z. B.:

DIR b:

»DIR« ist das Kommandowort, »b:« ein Kommandoparameter. Dieses Kommando

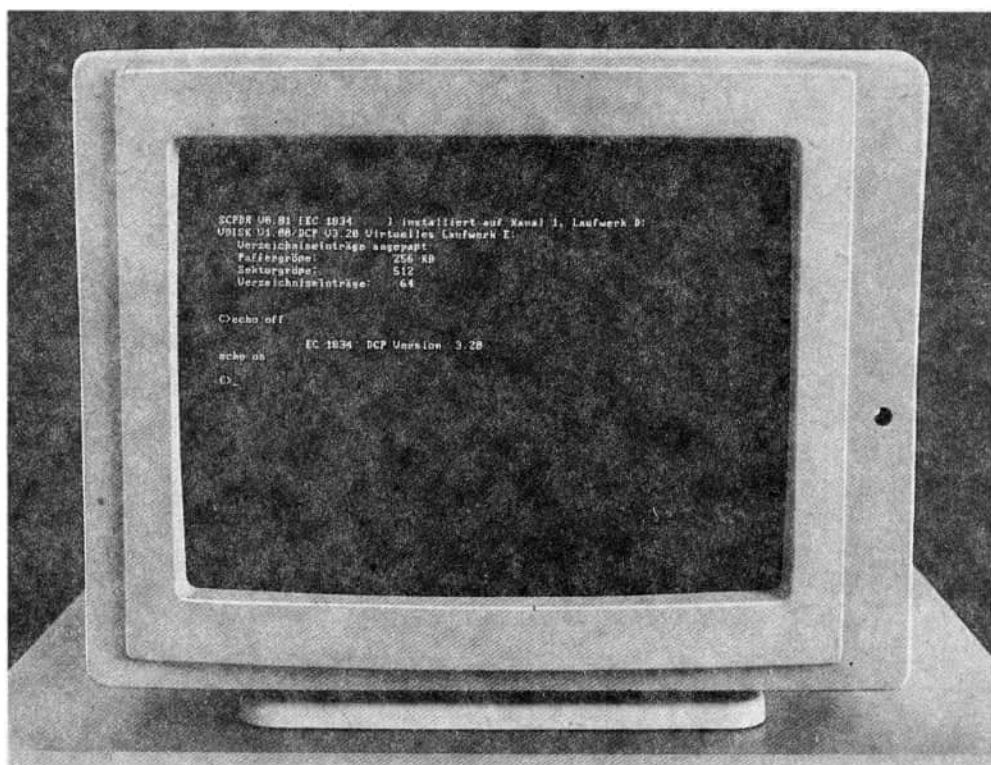


Abbildung 3.5.  
Eröffnungsbild des Steuerprogramms DCPX

bewirkt in Betriebssystemen für Personalcomputer die Anzeige des Dateiverzeichnisses (directory) im Laufwerk B. Kommandoparameter sind Bezeichner von Objekten (z. B. Laufwerk, Datei) oder Modifikationen für die Kommandoausführung. Modifikationen für die Kommandoausführung werden als *Schalter* bezeichnet.

*Steuerzeichenkommandos* sind Kommandos, die mit der CTRL-Taste und einer gleichzeitig betätigten weiteren Taste oder mit Sondertasten (z. B. Cursortasten) erzeugt werden, z. B.:

⟨CTRL⟩ C

Dieses Steuerzeichenkommando setzt im Betriebssystem SCP (vgl. 3.1.2.) das Diskettensystem nach Diskettenwechsel zurück.

Kommandos werden dem Rechnersystem entweder direkt mitgeteilt oder in einer Datei (Stapeldatei) abgelegt und später zur Ausführung aufgerufen.

Die direkte Eingabe eines Kommandos kann nur erfolgen, wenn das Rechnersystem dazu bereit ist. Diese Bereitschaft wird durch ein Systembereitschaftszeichen (Promptzeichen) für den Nutzer sichtbar angezeigt. In Abbildung 3.5. erfolgt das mit:

C) \_

»C« gibt das Laufwerk an, mit dem die Dateiverwaltung aktuell verbunden ist. Das Zeichen »)« ist das Systembereitschaftszeichen und »\_« zeigt die aktuelle Cursorposition an. Das Rechnersystem befindet sich



nach der Ausgabe des Systembereitschaftszeichens in einem Wartezustand, der erst mit dem Kommando und einer Kommandoendemitteilung <ET> (Taste ENTER in verschiedener Ausführung, wie RETURN, ET, ET1, REBOUT) beendet wird.

Die Gesamtheit der zulässigen Kommandos einschließlich ihrer Darstellungsregeln (Syntax) bilden die Kommando- oder Auftragssprache (für Betriebssysteme von Großrechenanlagen: Job Control Language, JCL). Der Nutzer muß sich exakt an die Kommandosprache des Betriebssystems halten. Zur Beschreibung von Kommandosprachen werden Metasprachen genutzt. Die Regeln für Metasprachen werden im Anhang A dargelegt.

Ein Kommando wird entweder durch die Kommandoverwaltung mit residenten programmtechnischen Mitteln ausgeführt oder es wird ein anderes Programm, eine Kommandodatei, in den Hauptspeicher transportiert und gestartet. Ein- und Ausgabesteuerung sowie Dateiverwaltung des Steuerprogramms bleiben als Schichten aktiv. Ist das Programm ausgeführt, geht die Steuerung an die Kommandoverwaltung zurück. Die Kommandoverwaltung führt ein bereits vorliegendes nächstes Kommando aus oder zeigt das Systembereitschaftszeichen an und wartet auf ein neues Kommando. Diese Technik bei der Ausführung von Kommandos führt zu einer Unterteilung in

- residente oder eingebaute (built in) Kommandos, die von der Kommandoverwaltung mit eigenen (residenten) Mitteln direkt ausgeführt werden und
- transiente Kommandos, die zur Ausführung an (transiente) Programme übergeben werden.

Transiente Programme sind Dienstprogramme oder Anwenderprogramme. Dienstprogramme und Anwenderprogramme werden also programmtechnisch gleich behandelt. Sie unterscheiden sich

aber funktionell wie Basissoftware und Anwendersoftware.

Der Abbildung 3.4. ist zu entnehmen, daß Betriebssysteme keine geschlossene Oberfläche besitzen. Anwenderprogramme können eine direkte Verbindung zur Schicht Dateiverwaltung und zur Schicht Ein- und Ausgabesteuerung besitzen. Betriebssysteme bieten auf diese Weise Anwenderprogrammen die Mitbenutzung von Befehlsfolgen des Steuerprogramms an. Das erfolgt über genau definierte Schnittstellen. Das Anwenderprogramm muß dazu die Register des Zentralprozessors auf genau definierte, in der Dokumentation beschriebene Weise laden und die Befehlsausführung zu einer festgelegten Adresse verzweigen (Softwareinterrupt). Das Steuerprogramm führt danach die Befehlsfolge aus und kehrt zurück in das Anwenderprogramm.

Ein Beispiel ist die Einstellung des Laufwerks B: als aktuelles Laufwerk unter Nutzung von Befehlsfolgen des Betriebssystems SCP.1520 von einem Anwenderprogramm aus. Zu realisieren ist vom Anwenderprogramm:

- Laden des Registers A (niederwertiger Teil) mit der Zahl 1 (Laufwerk A=0, B=1),
- Sprung zur Adresse 5 (mit Rückkehrabsicht).

Danach ist B das aktuelle Laufwerk.

Solche Leistungen eines Betriebssystems unterstützen die Entwicklung leistungsfähiger Anwenderprogramme erheblich.

### **Mehrprogrammbetrieb**

Eine wichtige Forderung an Betriebssysteme ist, daß sie den Einsatz der Betriebsmittel (Prozessoren, Hauptspeicher, Kanäle und periphere Geräte) optimieren, also pro Zeiteinheit eine hohe Verarbeitungsleistung (Durchsatz) erzielen. Dieser Forderung stehen z. B. große Unterschiede in der Arbeitsgeschwindigkeit der Betriebs-

mittel entgegen. Die schnellen Prozessoren müssen z. B. in einen Wartezustand versetzt werden, während ein Zeichendrucker aktiv ist. Auch bei Eingaben im Dialog, die direkt von der Arbeitsgeschwindigkeit des Menschen abhängen, ergeben sich erhebliche Wartezeiten für den Zentralprozessor. Diesen Konflikten in der Nutzung von Betriebsmitteln kann durch ein Betriebssystem mit Mehrprogramm- oder Mehrnutzerbetrieb entgegen gewirkt werden.

**Mehrprogrammbetrieb (multiprogramming mode)** ist eine Betriebsart, bei der das Betriebssystem die Ausführung mehrerer Programme zeitlich (zeitmultiplex) verzahnt.

Der alternative Begriff ist Einprogrammbetrieb (single user mode).

Wenn ein Rechnersystem über nur einen Prozessor verfügt, kann auch nur ein Programm befehlsweise ausgeführt werden. Die Ausführung mehrerer Programme wird erreicht, indem die Programme ganz oder teilweise in gesondert ausführbare Einheiten, *Task*, zerlegt werden. Die Task werden im Hauptspeicher bereitgehalten (reentrancy) oder von einem Archivspeicher auf Anforderung im Wechsel dorthin transportiert (swapping). Das Betriebssystem ordnet im zeitlichen Wechsel den Task verschiedener Programme Betriebsmittel zu und führt die Task aus (Multitaskbetrieb). Dieser Wechsel vollzieht sich sehr schnell und ist nach außen nicht sichtbar. So entsteht der Eindruck einer gleichzeitigen Bearbeitung. Es erfolgt aber eine Zeitteilung (time sharing, vgl. Abbildung 3.6.).

Der Teil des Hauptspeichers, der einem Programm während der Ausführung vom Betriebssystem zugeordnet wird, heißt *Partition*. Die Priorität und die Größe der Partition wird dem Betriebssystem durch Kom-

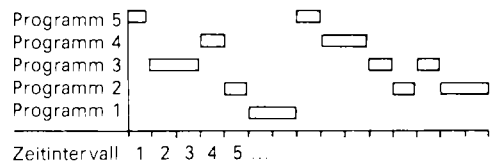


Abbildung 3.6.

Zeitteilung für aktive Task  
der Programme 1 bis 5

mandos mitgeteilt. Die Partition mit der geringsten Priorität heißt Hintergrund (Background)-Partition, alle anderen Partitionen heißen Vordergrund (Foreground)-Partitionen.

Vorteile des Mehrprogrammbetriebes ergeben sich, wenn in einem Programm eine Ein- oder Ausgabeanforderung oder ein Fehler auftritt. Dann entzieht das Betriebssystem sofort diesem Programm die Betriebsmittel und setzt sie für andere Programme ein. Der Zeitgewinn ist erheblich, wenn spezielle Ein- und Ausgabeprozessoren vorhanden sind, die eine vom Zentralprozessor unabhängige Ausgabe (Simultaneous Peripheral Operations On Line, SPOOL) ermöglichen.

Für das Betriebssystem ergeben sich beim Mehrprogrammbetrieb eine Reihe von Aufgaben über die Kommandoverwaltung hinaus. Dazu gehören das Umwandeln der Programme in Task, die Zuordnung der Betriebsmittel entsprechend den Prioritäten, Unterbrechungssteuerung sowie Reentrancy oder Swapping. Die entsprechende Komponente des Betriebssystems wird deshalb als Scheduler, Supervisor oder Monitor bezeichnet.

Der Begriff Mehrprogrammbetrieb wird vom Begriff Mehrnutzerbetrieb tangiert.

**Mehrnutzerbetrieb (multiuser mode)** ist Multitaskbetrieb im Dialog mit mehreren Nutzern.

Die Task können zu einem oder zu mehreren Programmen gehören, und es kann auf

mehrere oder eine gemeinsame Datenbasis zugegriffen werden (vgl. Abschnitt 3.2.). Je nachdem, ob der Nutzer dabei Zugang zu allen Betriebsmitteln des Rechnersystems hat oder nicht, handelt es sich um Teilnehmerbetrieb oder Teilhaberbetrieb.

*Teilnehmerbetrieb* ist die charakteristische Nutzungstechnologie bei der Programmentwicklung im Dialog mit Kleinrechenanlagen und Großrechenanlagen.

*Teilhaberbetrieb* mit gemeinsamer Daten- und Programmbasis wird für Platzbuchungssysteme in Hotels und Reisebüros, Mehrplatzdatenerfassungssysteme und Auskunftssysteme genutzt.

3.1.3. Überblick über ausgewählte Betriebssysteme

Eine Übersicht häufig eingesetzter Betriebssysteme enthält Tabelle 3.1. Betriebssysteme werden ständig weiterentwickelt. Deshalb erhalten die einzelnen Ausgaben Versionsnummern (z. B. OS/EC Ausgabe 6.1, DCP 1834 Version 3.30). Bei der Entwicklung von Betriebssystemen wird nach internationalen Standards gestrebt, um die Nutzung verbreiteter Entwicklungs-, Standard- und Anwendersoftware zu ermöglichen. Die Betriebssysteme sind weitgehend herstellerabhängig und verbreiten sich mit der Hardware. Für Anlagen des ESER werden die Betriebssysteme OS/EC und DOS/EC bevorzugt. Nachstehend wird ein Überblick über den Einsatz, die Nutzerschnittstelle, das Datenverwaltungs- und Steuerkonzept der häufigsten Betriebssysteme gegeben.

Tabelle 3.1.  
Betriebssysteme des VEB Kombinat Robotron und ihre Zuordnung zu Rechnerklassen (Auswahl)

Rechnerklasse	Betriebssystem
Personalcomputer ( 8 Bit)	Single user Control Program (SCP 1520 / SCP 1715 / SCP 3.0)
	(16 Bit) Single user Control Program (SCP 1700)
	Disk oriented Control Program (DCP)
Kleinrechenanlagen	Multi User Time sharing Operating System (MUTOS 1600)
	MODule Operating System (MOOS 1600)
	Operating System (OS/EC)
Großrechenanlagen	Disk Operating System (DOS/EC)

Tabelle 3.2.

Residente Kommandos des Betriebssystems SCP (Auswahl)

Kommandowort	Kommandoparameter	Wirkung
DIR	Laufwerksbezeichner	Directory anzeigen
ERA	Dateiname	Datei löschen
REN	neuer Name = alter Name	Datei umbenennen
TYPE	Dateiname	Textdatei anzeigen
USER	Nutzerbereich 0...15	Nutzerbereich wählen

Tabelle 3.3.

Steuerzeichenkommandos des SCPX (Auswahl)

Kommandozeichen	Wirkung
^C	Rücksetzen des Diskettensystems nach Diskettenwechsel
^P	Druckerausgabe parallel zum Bildschirm EIN/AUS
^S	Unterbrechung der Bildschirmausgabe
^X	Eingabezeile und Zeilenpuffer löschen

Tabelle 3.4.

Transiente Kommandos des Betriebssystems SCP (Auswahl)

Kommandowort	Kommandoparameter	Wirkung
INIT	—	Disketten initialisieren
PIP	Dateinamen und Optionen	Dateien kopieren, verketteten, ausgeben
STAT	Datei- und Gerätenamen	Dateiattribute anzeigen und ändern
SUBMIT	—	Stapeldatei ausführen

### Betriebssystem SCP

Das Betriebssystem SCP wird vom VEB Kombinat Robotron als Grundausstattung aller 8-Bit-Personalcomputer (Bürocomputer A 5120/30, PC 1715/1715 W) und des 16-Bit-Computers A 7150 geliefert.

Das Steuerprogramm SCPX besteht aus den Komponenten BIOS (Basic Input Output System) für die Ein- und Ausgabesteuerung, BDOS (Basic Disk Operating System) zur Dateiverwaltung und CCP (Console Command Processor) zur Kommandoverwaltung.

Die *Nutzerschnittstelle* ist kommandogesteuert. Wichtige residente Kommandos und ihre Wirkungen enthält Tabelle 3.2.

Einige SCP-Steuerzeichenkommandos (Zeichen ^ für Taste <CTRL>) zeigt Tabelle 3.3.

Wichtige transiente Kommandos können der Tabelle 3.4. entnommen werden.

Die *Dateiverwaltung* des Betriebssystems SCP unterstützt Dateien mit sequentiellem und wahlfreiem Zugriff. Die Dateien kön-

nen einen Schreibschutz (R/O-Attribut) erhalten und für die Anzeige mit DIR gesperrt werden (SYS-Attribut). Außerdem lassen sie sich in separaten Nutzerbereichen (Userbereiche 0 bis 15) unterbringen. Der Nutzer hat immer nur Zugriff zu dem aktuellen Nutzerbereich.

Das *Steuerkonzept* des Betriebssystems SCP sieht nur den Einprogramm- und Ein-nutzerbetrieb vor.

Fehler in der Ein- und Ausgabe oder bei Speicheroperationen werden durch die Komponenten BIOS und BDOS nach dem Grundschemata

BIOS/BDOS ERROR ON <Laufwerk>:(Fehlerursache)

mitgeteilt. Als Fehlerursache werden angegeben:

- BAD SECTOR bei Disketten- oder Plattenfehlern (auch falsche Formate),
- SELECT bei nicht existierenden Laufwerken,
- READ ONLY bei nicht zulässigem Schreibvorgang (z. B. Schreiben nach Diskettenwechsel, ohne daß das Diskettensystem mit "C zurückgesetzt wurde),
- FILE R/O wenn eine schreibgeschützte Datei mit R/O-Attribut überschrieben werden soll.

### Betriebssystem DCP

Das Betriebssystem DCP des VEB Kombinat Robotron gehört zur Grundausstattung des Personalcomputers EC 1834 und des Arbeitsplatzcomputers A 7150. Das Steuerprogramm DCPX besteht aus den Komponenten IO.SYS für die Ein- und Ausgabe-steuerung, DOS.SYS für die Dateiverwaltung und COMMAND.COM für die Kommandoverwaltung.

Es stehen eine kommando- und eine menü-gesteuerte (grafische) *Nutzerschnittstelle* zur Verfügung. Da der Kommandoprozessor COMMAND.COM nicht ständig im Hauptspeicher resident ist, werden die durch ihn realisierten Kommandos nicht als residente, sondern als eingebaute (built in) Kommandos bezeichnet. Es besteht die Möglichkeit, Kommandos in einer Stapel-datei abzulegen und geschlossen zur Ausführung aufzurufen. Der Kommandovorrat dafür ist im Vergleich zum Betriebssystem SCP erheblich erweitert.

Ein wesentlicher Unterschied zum Betriebssystem SCP ergibt sich durch die hierarchische *Dateiverwaltung*. Ist eine Datei auf einem Datenträger zu identifizieren, so ist der Weg vom Wurzelverzeichnis, dem Ausgangspunkt der Baumstruktur, bis zu dem Verzeichnis zu beschreiben, das diese Datei enthält. Dieser Weg heißt Pfad. Laufwerk, Pfad und Dateiname heißen in ihrer Gesamtheit Dateispezifikation. Die Dateispezifikation in der Abbildung 3.7. identifiziert eine Datei GETDIR.BIB im Laufwerk A:, die sich im Verzeichnis LIBRARY befindet. Das Verzeichnis LIBRARY ist Unterverzeichnis des Verzeichnisses PASCAL. Die hierarchische Verwaltung der Dateien ist vor allem bei der Verwendung von Festplatten unentbehrlich.

Als Besonderheit gegenüber SCP ermöglicht DCP die gleichzeitige Eingabe mehrerer Kommandos getrennt durch den Operator »|« (Pipe-Operator). Dabei übergibt das linke vom Operator stehende (erzeugende) Kommando während seiner Ausführung Information an das rechts ste-

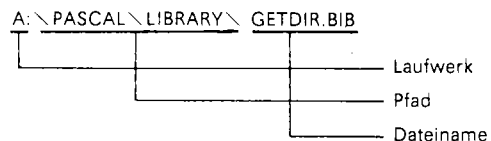


Abbildung 3.7.  
Dateispezifikation im Betriebssystem DCP

Tabelle 3.5.  
Steuerzeichenkommandos  
des Betriebssystems DCP (Auswahl)

Kommando- zeichen	Wirkung
^⟨ALT⟩⟨DEL⟩	Bootstrap (Reset, Neustart)
⟨PRTSC⟩⟨I⟩	Druckerausgabe des Bildschirm- schirminhalts (Hardcopy)
^⟨BREAK⟩	Programmunterbrechung

hende (konsumierende) Kommando und veranlaßt dessen Ausführung. Außerdem kann mit Hilfe der Operatoren >, < und >> die Ein- und Ausgabe zu verschiedenen Geräten oder Dateien umgeleitet werden.

Wichtige Steuerzeichenkommandos des Betriebssystems DCP sind (mehrere Tasten müssen gleichzeitig gedrückt werden) in Tabelle 3.5. aufgeführt.

Das *Steuerkonzept* des Betriebssystems DCP unterstützt den Einprogramm- und Einnutzerbetrieb. Durch das Dienstprogramm PRINT ist auch ein Ausdrucken von Textdateien während der Ausführung anderer Programme möglich (SPOOL-Betrieb). Die Textdateien und der Drucker stehen, während PRINT aktiv ist, nicht für andere Aufgaben zur Verfügung.

Die Fehlerausschriften des Betriebssystems DCP erfolgen im Klartext. Die Kommandos des Betriebssystems DCP werden im Abschnitt 3.3.2. in der Anwendung gezeigt. Anhang B enthält eine vollständige Kommandoübersicht.

### Betriebssystem MUTOS

Das Betriebssystem MUTOS wird vom VEB Kombinat Robotron als Grundausstattung der Bürocomputer A 5120.16 (MUTOS 8000) und der Kleinrechenanlagen K 1600 (MUTOS 1600) bereitgestellt. MUTOS besteht zunächst aus einem Kern

(Nukleus), der die Ein- und Ausgabesteuerung, die Dateiverwaltung sowie die Steuerung der Programmausführung übernimmt. Der Kern enthält außer den Befehlsfolgen zur E/A-Steuerung und Datenverwaltung die Befehlsfolgen zur Kommandoausführung. Die Befehlsfolgen können über Systemrufe in Anspruch genommen werden. Der MUTOS-Kern enthält keine Kommandoverwaltung, weil die Systemrufe nur durch transiente Programme ausgelöst werden können. Dadurch gibt es keine residenten Kommandos wie in den Betriebssystemen SCP und DCP. Um dem Nutzer einen direkten Zugang zu den leistungsfähigen Systemrufen zu ermöglichen, stellt das Betriebssystem MUTOS ein gesondertes transientes Programm zur Verfügung. Es wird als Kommandointerpreter oder *Shell* bezeichnet.

Der Kommandointerpreter realisiert einen wesentlichen Teil der *Nutzerschnittstelle*. Ein einfaches (Shell-)Kommando hat die Grundform (Schreibweise vgl. Anhang A):

⟨Kommandowort⟩ {⟨Parameter⟩} ⟨ET⟩

Der Kommandointerpreter prüft das Kommando auf syntaktische Richtigkeit und löst die zu seiner Realisierung erforderlichen Prozesse aus. Nach der Beendigung dieser Prozesse meldet sich der Kommandointerpreter und erwartet das nächste Kommando.

Wird eine Kommandozeile mit »&« abgeschlossen, geht das Betriebssystem zum Mehrprogrammbetrieb über. Die Ausführung des Kommandos erfolgt im Hintergrund, der Kommandointerpreter ist sofort wieder zur Entgegennahme eines neuen Kommandos bereit.

Wie im Betriebssystem DCP kann mit Pipe-Operatoren Information von einem zum anderen Kommando übergeben und Ein- und Ausgaben können umgeleitet werden. Dem Kommandointerpreter können auch Kommandogruppen in der Form

Tabelle 3.6.

Kommandos des Betriebssystems MUTOS (Auswahl)

Kommandowort	Kommandoparameter	Wirkung
CHMOD	Zugriffsoption, Pfad, Dateispezifikation	Änderung von Zugriffsrechten für Datei/Subdirectory
CP	Quell- und Zielspezifikation	Datei kopieren
DATE	Datum und Uhrzeit, Dateispezifikation	Datum und Uhrzeit ausgeben oder setzen
LOGIN	Nutzername, Paßwort	Anmelden des Nutzers
LS	Optionen, Dateispezifikation	Anzeige Directory
PASSWD	altes und neues Paßwort	Paßwort ändern
PR	Optionen, Dateispezifikation	Ausgabe einer Datei
RMDIR	Datei- oder Subdirectoryspezifikation	Datei oder Subdirectory löschen

<Kommandowort> {<Parameter>}  
 {;<Kommandowort> {<Parameter>}}  
 übergeben werden. Die (durch Semikolon getrennten) Kommandos werden nacheinander ausgeführt. Ausgewählte Shell-Kommandos sind in Tabelle 3.6. zusammengefaßt.

Beim Betriebssystem MUTOS können Kommandos auch in einer programmiersprachenähnlichen Form erteilt werden. Diese Kommandoprogrammiersprache enthält Steuerstrukturen (IF-, CASE-, WHILE-, FOR-Konstruktionen), vordefinierte und nutzerdefinierte Variablen sowie vordefinierte Prozeduren.

Die *Dateiverwaltung* des Betriebssystems MUTOS ist wie im Betriebssystem DCP hierarchisch aufgebaut. Dabei werden neben Text- und Binärdateien sogenannte Gerätedateien verwaltet. Gerätedateien sind Geräte (Bildschirm, Tastatur, Plattenspeicher), die beim Zugriff wie Dateien behandelt werden.

Der Zugriffsschutz für Binär- und Textdateien ist im Betriebssystem MUTOS erweitert. Beim Anlegen einer Datei wird der Datei eine Nutzeridentifikation aus Namen, Nutzernummer und Gruppennummer zugeordnet. Der Nutzer kann durch ein Kommando oder von einem Programm aus die

Zugriffsrechte für verschiedene Nutzerklassen festlegen. Eine Systemkomponente (der System-Administrator oder Super-User) hat unbeschränkten Zugriff.

Das *Steuerkonzept* des Betriebssystems MUTOS unterstützt den Mehrprogramm- und Mehrnutzerbetrieb.

### Betriebssysteme OS/EC und DOS/EC

Die Betriebssysteme OS/EC und DOS/EC werden ausschließlich auf Großrechenanlagen des ESER eingesetzt. Entsprechend anspruchsvoll und leistungsfähig sind die Konzepte zur Nutzerschnittstelle, zur Datenverwaltung und zur Steuerung der Programmausführung. Die Komponenten des Steuerprogramms heißen Supervisor, Jobverwaltung und Datenverwaltung.

Die *Nutzerschnittstelle* berücksichtigt Kommandos durch Bediener von der Konsole (dem Bediengerät, im allgemeinen Tastatur und Bildschirm) und Kommandos auf Datenträgern (Jobsteueranweisungen). Einen Einblick in Bedienkommandos des OS/EC, Ausgabe 6.1, gibt die Tabelle 3.7. auf Seite 92.

Grundsätzlich können die meisten Bedienkommandos (hier außer DUMP, SWAP) auch auf Datenträgern bereitgestellt und

Tabelle 3.7.

Bedienkommandos des Betriebssystems OS/EC (Auswahl)

Kommandowort	Kommandoparameter	Wirkung
CANCEL	Jobname, Jobkennzeichen, Geräteadresse	Abbruch der Verarbeitung
DUMP	Kennzeichnungstext, Bereichsangaben	Hauptspeicherabzug erzeugen
LOG	Text	Eintragung in Systemprotokoll
RESET	Jobname, Priorität, Klasse	Priorität und Klasse des Jobs ändern
SWAP	alte und neue Geräteadresse/Schalter	Datenträger/Geräte austauschen

Tabelle 3.8.

Jobsteueranweisungen des Betriebssystems OS/EC (Auswahl)

Kommandowort	Kommandoparameter	Wirkung
JOB	Abrechnungsinformation, Name des Nutzers, Zeitbegrenzung, Nachrichtenumfang, Jobklasse	Kennzeichnung des Anfangs eines Jobs
EXEC	Programmname, Übergabeparameter	Definition des auszuführenden Programms
DD	Dateiart, Dateistruktur, Geräte, Dateinamen, Datenträger	Definition der zu verwendenden Dateien
//		Jobendeerkennung

mit den Programmen und Daten als Jobstrom eingegeben werden. In diesem Fall werden die Bedienkommandos als Kommandoanweisung bezeichnet. Der überwiegende Teil der Kommandos für die Stapelverarbeitung wird mit Hilfe der Jobsteuersprache erteilt. Die Jobsteueranweisungen befinden sich auf Datenträgern. Jede Jobsteueranweisung beginnt mit zwei Schrägstrichen, denen ein Name mit maximal acht Zeichen folgen kann (//<Name>). Wichtige Jobsteueranweisungen sind in der Tabelle 3.8. zusammengefaßt.

Jobsteueranweisungen können als Anweisungsfolge katalogisiert und durch eine EXEC-Anweisung geschlossen zur Ausführung aufgerufen werden.

Zur Steuerung einer Nutzerschnittstelle für die Dialogverarbeitung (vgl. Abschnitt 3.2.) stehen im Betriebssystem OS/EC spezielle Systemkomponenten zur Verfügung, dar-

unter die Komponente TSO (Time Sharing Option).

Die *Dateiverwaltung* des OS/EC unterstützt den sequentiellen und direkten Zugriff. Die Sätze der Dateien können eine feste, variable oder unbestimmte Länge besitzen.

Das *Steuerkonzept* der Betriebssysteme DOS/EC und OS/EC realisiert Mehrprogramm- und bei der Dialogverarbeitung auch Mehrnutzerbetrieb in verschiedenen Ausbaustufen. Ausbaustufen sind z. B. Mehrprogrammbetrieb mit einer festen Anzahl von Task (Multiprogramming with a Fixed number of Task, MFT), mit einer variablen Anzahl von Task (Multiprogramming with a Variable number of Task, MVT) oder mit virtueller Speichertechnik (System Virtual Storage, SVS).



### 3.2. Nutzungstechnologien

Zur Verarbeitung von Information können durch das Betriebssystem unterschiedliche Nutzungstechnologien unterstützt werden. Eine Übersicht enthält Abbildung 3.8.

#### Stapelverarbeitung

**Stapelverarbeitung (batch processing) ist eine Technologie, bei der die für die Verarbeitung erforderliche Information (Programme und Verarbeitungsinformation) vor der Ausführung des Programms als Auftrag (Job) vollständig bereitgestellt wird.**

In der Regel werden bei der Stapelverarbeitung mehrere Jobs unmittelbar nacheinander ausgeführt (Jobstrom). Die Komponenten jedes Jobs, die Reihenfolge und die Prioritäten der Ausführung werden durch Jobsteuerkommandos festgelegt. Die Jobsteuerkommandos sind in maschinenlesbarer Form Bestandteil des Jobs, um die Übergangszeit zwischen den Jobs kurz zu halten.

Die Stapelverarbeitung gewährleistet einen wesentlich höheren Durchsatz als die interaktive Verarbeitung. Sie ist die günstigste Nutzungstechnologie, wenn die jeweilige Aufgabe das zuläßt.

Stapelverarbeitung ist gegenwärtig die charakteristische Nutzungstechnologie für Großrechenanlagen. Stapelverarbeitung ist auch für Personalcomputer möglich und in einigen Fällen zweckmäßig. Im Betriebssystem SCP gibt es dazu das Dienstprogramm SUBM. Dieses Dienstprogramm sucht nach dem Start die als Kommandoparameter angegebene Stapeldatei und führt die in ihr abgelegten Kommandos nacheinander aus. Den möglichen Inhalt einer Stapeldatei zeigt Abbildung 3.9.

Hat die Stapeldatei den Namen ARTIKEL

.SUB (der Typname SUB ist vorgeschrieben), so erfolgt der Start mit dem Kommando:

SUBM b:Artikel b

Das Dienstprogramm SUBM.COM (es muß sich auf der Diskette im aktuellen Laufwerk befinden) sucht die Stapeldatei ARTIKEL.SUB im Laufwerk B und führt die in dieser Datei enthaltenen Kommandos nacheinander aus. Dabei werden

- Zeilen, die mit »;« beginnen, als Mitteilungen auf den Bildschirm geschrieben,
- \$1 (möglich ist \$1 bis \$9) als formaler Parameter interpretiert und vor der Ausführung durch den im Startkommando angegebenen (ersten) aktuellen Parameter »b« ersetzt.

Im Beispiel (Abbildung 3.9.) werden zunächst die Mitteilungen

Stapel: Aktualisierung der Artikeldatei  
Start Aktualisierung



Abbildung 3.8.  
Nutzungstechnologien  
eines Rechnersystems

```

; Stapel: Aktualisierung der Artikeldatei
; Start Aktualisierung
$1:AKTUELL
$1:SORT $1:Artikel.DAT
; Artikeldatei aktualisiert
  
```

Abbildung 3.9.  
Inhalt der SCP-Stapeldatei ARTIKEL.SUB

```

ECHO
REM Stapel: Aktualisierung der Artikeldatei
REM Start Aktualisierung
$1:AKTUELL
$1:SORT $1:Artikel.DAT
REM Artikeldatei aktualisiert

```

Abbildung 3.10.  
Inhalt der DCP-Stapeldatei ARTIKEL.BAT

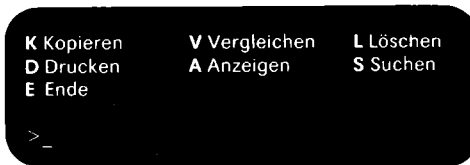


Abbildung 3.11.  
Dialogmenü

auf den Bildschirm geschrieben, und danach wird das Programm AKTUELL im Laufwerk B: gestartet. Ist die Ausführung von AKTUELL beendet, wird das Programm SORT im Laufwerk B: gestartet. SORT wird der Name einer (zu sortierenden) Datei übergeben. Hat SORT die Arbeit beendet, erscheint auf dem Bildschirm die Mitteilung »Artikeldatei aktualisiert«, und die Steuerung geht an das Steuerprogramm SCPX zurück.

Im Betriebssystem DCP sind die Möglichkeiten der Stapelverarbeitung gegenüber dem Betriebssystem SCP erleichtert und erweitert. Im Betriebssystem DCP tragen Stapeldateien den Typnamen BAT. Die Stapeldatei aus Abbildung 3.9. hat im Betriebssystem DCP den in Abbildung 3.10. dargestellten Inhalt.

Der Start der Datei ARTIKEL.BAT erfolgt direkt mit

A>Artikel b<ET>

Das Steuerprogramm erkennt am Typnamen (der beim Start nicht angegeben wird), daß es sich um eine Stapeldatei handelt und führt die Kommandos unter Substitution der formalen Parameter aus. Der

Basisname darf kein Name eines transienten Programms sein, denn bei einem Namenskonflikt haben transiente Programme die Priorität. Ein Dienstprogramm mit der Funktion des SUBM.COM ist unter DCP nicht erforderlich.

Zur Formulierung von Kommandofolgen im Betriebssystem DCP steht zusätzlich zu den Möglichkeiten des Betriebssystems SCP eine einfache Kommandosprache mit FOR-, GOTO- und IF-Befehlen zur Verfügung. Außerdem gibt es für DCP-Stapeldateien die Möglichkeit, formale Parameter (Umgebungsparameter) zu benutzen, die vor dem Start durch ein spezielles Kommando (SET) aktuell festgelegt werden.

### Interaktive Verarbeitung

**Die interaktive Verarbeitung (interactive processing) ist eine Technologie, bei der die für die Verarbeitung erforderliche Information ganz oder teilweise während der Ausführung eines Programms bereitgestellt wird.**

Träger der interaktiven Verarbeitung sind das Rechnersystem und entweder ein Nutzer oder ein mit dem Rechnersystem gekoppelter technisch-physikalischer Prozeß. Nach den beteiligten Trägern wird in Dialogverarbeitung und Prozeßdatenverarbeitung unterschieden.

**Dialogverarbeitung (conversational mode) ist eine Technologie, bei der während der (interaktiven) Programmausführung eine Kommunikation zwischen Mensch und Maschine erfolgt.**

Während der Dialogverarbeitung sind Mensch und Rechnersystem (Maschine) wechselseitig aktiv (interaktiv) und realisieren jeweils einen Dialogschritt (Aktion). Jeder Dialogschritt ist eine Funktion des vorhergehenden Dialogschritts. Nach der

Gestaltung des Dialogs kann zwischen kommando- und menügesteuertem Dialog unterschieden werden.

Beim *kommandogesteuerten Dialog* sendet das Rechnersystem eine Eingabeaufforderung im allgemeinen über den Bildschirm. Danach befindet sich das Rechnersystem im Wartezustand. Der Mensch sendet im allgemeinen über die Tastatur ein Kommando als Antwort. Nach Erkennung der Endemittlung (z. B. Tasten ET, ENTER) wird das Kommando auf syntaktische Richtigkeit und Ausführbarkeit geprüft. Anschließend werden die zur Ausführung des Kommandos gehörenden Befehlsfolgen aktiviert, oder es wird ein Fehler angezeigt und die Bereitschaft zur Aufnahme eines neuen Kommandos hergestellt.

Im *menügesteuerten Dialog* unterbreitet das Rechnersystem ein Angebot in Form eines Dialogmenüs (vgl. Abbildung 3.11.).

**Ein Dialogmenü (kurz Menü) ist ein Angebot möglicher Kommandos durch das Rechnersystem, aus dem der Nutzer ein Kommando zur Ausführung auswählt.**

Nach der Anzeige des Menüs, das stets mit einer Eingabeaufforderung endet, wird das Rechnersystem in den Wartezustand versetzt. Für die Eingabe sind nur Zeichen zugelassen, die dem konkreten Angebot entsprechen. Das gewählte Kommando wird ausgeführt.

Die Gestaltung von Menüs ist vielfältig. Es sind Textdarstellungen und grafische Symbole (sogenannte Ikonen) auf grafikfähigen Bildschirmen möglich. Die Auswahl grafischer Symbole erfolgt, indem ein speziell gestaltetes Zeichen, z. B. ein Pfeil, mit Hilfe von Cursortasten oder Spezialgeräten (Maus, Joystick, Lichtstift) über der auszuwählenden Ikone positioniert wird.

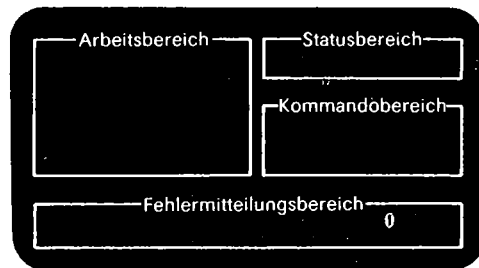


Abbildung 3.12.  
Bildschirmbereiche (Fenster)

Der Vorzug des menügesteuerten Dialogs liegt in der Einfachheit der Nutzerschnittstelle und in der Sicherung gegen fehlerhafte Kommandos. Die Gestaltung des Dialogs liegt in hohem Maße bei der Maschine.

Der kommandogesteuerte Dialog erfordert umfangreichere Kenntnisse des Nutzers, ermöglicht aber die Wahl zwischen einer größeren Zahl von Varianten und benötigt, besonders bei geübten Nutzern, weniger Zeit als der menügesteuerte Dialog. Beide Formen behalten ihre Bedeutung.

Bei der äußeren Gestaltung des Dialogs spielt die Fenstertechnik (Window-Technik) eine zunehmende Rolle. Fenster sind definierte Bildschirmbereiche. Diese Bereiche werden wie gesonderte Bildschirme gehandhabt. Im Dialog ist eine Einteilung des Bildschirms in einen Statusbereich, Arbeitsbereich, Kommando- und Fehlermitteilungsbereich typisch (vgl. Abbildung 3.12.).

Der Statusbereich kann Information über das aktive Programm, die gewählte Funktion, das Datum und die Uhrzeit enthalten. Im Kommandobereich werden die für den kommando- oder menügesteuerten Dialog erforderliche Information und das Echo der Nutzereingaben angezeigt. Der Arbeitsbereich enthält die Information, die der eigentliche Gegenstand der Programmausführung ist. Bei einem Anwenderprogramm zur Verkaufsdisposition sind

das z.B. Angaben zum verkaufsbereiten Artikel (Bezeichnung, Schlüsselnummer, Gebrauchseigenschaften, Bestand, Preis, Lieferzeit). Der Fehlermitteilungsbereich enthält Information, die das aktive System unabhängig vom Nutzer anzeigt. Das sind vor allem Eingabefehler.

Leistungsfähige Betriebssysteme ermöglichen auf geeigneten Rechnersystemen gleichzeitig Dialog- und Stapelverarbeitung. Da die Dialogverarbeitung durch die relativ langsame Arbeitsgeschwindigkeit des Menschen zur häufigen Unterbrechung der Prozessortätigkeit führt, erhöht sich der Durchsatz bei gleichzeitiger Stapelverarbeitung erheblich.

Auch die *Prozeßdatenverarbeitung* ist interaktive Verarbeitung.

**Prozeßdatenverarbeitung (process control mode)** ist eine Technologie, bei der während der (interaktiven) Programmausführung ein ständiger Informationsaustausch mit einem gekoppelten technisch-physikalischen Prozeß erfolgt.

Das Rechnersystem erhält ständig Eingabedaten, die im wesentlichen Meßdaten des technisch-physikalischen Prozesses darstellen, über spezielle, mit dem Rechnersystem direkt verbundene Meßgeräte. Die Meßdaten werden mit Sollwerten verglichen (Überwachung), und in Abhängigkeit vom Vergleichsergebnis wird Information über erforderliche Veränderungen technisch-physikalischer Parameter ermittelt. Die Information wird durch spezielle periphere Geräte (Stelleinrichtungen) zur Steuerung des Prozesses verwendet. In Ausnahmefällen wird die Programmausführung unterbrochen und der menschliche Eingriff angefordert.

Die Prozeßdatenverarbeitung stellt hohe Anforderungen an das Rechnersystem. Es muß ständig eingabebereit sein und Reak-

tionszeiten (Antwortzeiten) realisieren, die einen rechtzeitigen Eingriff in den laufenden Prozeß ermöglichen. Deshalb wird die Prozeßdatenverarbeitung auch als Echtzeitverarbeitung (real time processing) bezeichnet. Zur Erfüllung der Anforderungen für die Prozeßdatenverarbeitung werden Rechnersysteme hardwareseitig speziell ausgerüstet. Sie besitzen eine besonders leistungsfähige Unterbrechungstechnik und eine spezielle Ein- und Ausgabeperipherie zur Umsetzung von Meßdaten (Analog/Digital- und Digital/Analog-Umsetzer). Rechnersysteme mit dieser Konfiguration werden auch als Prozeßrechner (process computer) bezeichnet. Mehrprogrammbetrieb auf Prozeßrechnern kann zu einem Konflikt mit Echtzeitanforderungen führen.

### 3.3.

#### Nutzung des Betriebssystems DCP

##### 3.3.1.

##### Dateien

DCP verwaltet Dateien hierarchisch (vgl. Abschnitt 3.1.2.). Basisnamen und Namen von Verzeichnissen bestehen aus maximal 8 Namenszeichen. Als Namenszeichen sind alle Buchstaben und Ziffern sowie folgende Sonderzeichen zugelassen:

! @ # \$ % & ( ) - \_ { } ' `

Groß- und Kleinschreibung werden nicht unterschieden. Die Zeichen »?« und »\*« haben eine Sonderfunktion.

Die folgende Dateispezifikation bezeichnet eine Datei PASCAL.TXT, die sich auf einem Datenträger des Laufwerks A im Verzeichnis COMPILER befindet:

A:\COMPILER\PASCAL.TXT

Die in der Tabelle 3.9. aufgeführten Typnamen sind vom Betriebssystem oder von anderer Basissoftware reserviert. Vom Betriebssystem reservierte Typnamen (»\*\*«)

Tabelle 3.9.

Reservierte Typnamen (Auswahl) unter DCP

Typname	Reservierung	Bemerkung
ASM	*	Quellprogramm der Assemblersprache
BAK	*	Sicherungstextdatei
BAS	*	Quellprogramm der Programmiersprache BASIC
BAT	**	Steuerkommandodatei zur Stapel-(Batch)verarbeitung
COM	**	ausführbares Programm (Kommandodatei)
EXE	**	ausführbares Programm (Executedatei)
PAS	*	Quellprogramm der Programmiersprache PASCAL
SYS	**	Bestandteil des Betriebssystemkerns

dürfen nicht vom Nutzer für eigene Dateien vergeben werden. Von anderer Basissoftware reservierte Typnamen (»\*)«) sollten nicht vergeben werden, wenn auf dem Rechnersystem diese Basissoftware eingesetzt wird.

Verwendet werden sollten die Typnamen TXT für allgemeine Textdateien, DOK oder DOC für Textdateien mit Dokumentationen, TMP oder \$\$\$ für temporäre Arbeitsdateien.

Jede Datei besitzt Attribute, die ihre Manipulierbarkeit beeinflussen und durch den Nutzer verändert werden können (Kommando ATTRIBUT). Das sind:

- das Zugriffsattribut mit den Zuständen R/W (read/write, Lesen und Schreiben möglich) und R/O (read only, Datei ist schreibgeschützt); ohne Festlegungen des Nutzers ist das Attribut R/W gültig (Standard, default);
- das Anzeigeattribut mit den Zuständen DIR-Anzeige möglich und HIDDEN (verborgen, versteckt); Standard ist: DIR-Anzeige möglich;
- das Systemattribut mit den Zuständen Systemdatei und keine Systemdatei; Dateien mit Systemattribut werden wie Dateien mit dem Attribut HIDDEN behandelt; zusätzlich können Dateien mit den Typnamen COM, EXE und BAT nicht aufgerufen werden;

- das Archivattribut; das Setzen des Archivattributes bewirkt, daß mit dem Kommando XCOPY rationell eine Sicherheitskopie der Datei erstellt werden kann.

Dateiattribute werden in einem zusätzlichen Byte gespeichert, das zusammen mit dem Namen der Datei, dem Datum und der Uhrzeit der Dateierstellung, der Dateigröße in Byte und anderen Parametern im Directory abgelegt ist.

Vom Betriebssystem DCP werden periphere Geräte wie Dateien behandelt, von denen Daten gelesen oder auf die Daten ausgegeben werden können. Für solche Gerätedateien werden die in Tabelle 3.10. aufgeführten festen Namen reserviert.

Beim Systemstart sind COM1 und AUX sowie PRN und LPT1 gleichgesetzt. Der Nutzer kann sie trennen oder andere Zuweisungen treffen (Kommando MODE). Soll z. B. ein serieller Drucker angeschlossen werden, können ihm die Schnittstelle und damit der Gerätedateiname COM2 zugewiesen und seine Übertragungsparameter geändert werden.

Die in Tabelle 3.10. aufgeführten Gerätedateinamen können in Kommandos zur Ein- und Ausgabe von Dateien genutzt werden. Es ist nicht möglich, Gerätenamen als Namen von Speicherdateien (Diskette, Festplatte) zu verwenden.

Tabelle 3.10.  
Gerätedateinamen im Betriebssystem DCP

Gerätedateiname	Geräteschnittstelle
CON (consol)	Tastatur und Bildschirm
AUX (auxiliary)	
COM1 (communication)	1. serielle Schnittstelle, z. B. für den Anschluß eines Modems
COM2	2. serielle Schnittstelle
PRN (printer)	
LPT1 (lineprinter)	1. parallele Schnittstelle (CENTRONICS) für Druckeranschluß
LPT2	2. parallele Schnittstelle

Bei der Eingabe von Kommandos, die Speicherdateinamen benutzen, kann durch Verwendung eines Gruppennamens eine Gruppe von Dateien angesprochen werden.

**Gruppennamen sind Dateinamen, die im Basis- oder im Typnamen die Zeichen »?« oder »\*« (Jokerzeichen, Wildcardzeichen) enthalten.**

Ein »?« in einem Gruppennamen steht für jedes beliebige Zeichen in einem Dateinamen. Der Gruppename BEISPIEL.?A? steht also z. B. für die Gruppe:

BEISPIEL.PAS  
BEISPIEL.BAK

Beide Dateinamen unterscheiden sich lediglich im 1. und 3. Zeichen des Typnamens. Genau diese wurden im Gruppennamen mit dem Jokerzeichen »?« belegt und so als nicht signifikant für die Identifikation bezeichnet.

Das Jokerzeichen »\*« steht für einen beliebigen Basis- oder Typnamen. Ein Gruppename wäre:

BEISPIEL.\*

Zur Gruppe der bezeichneten Dateien gehören jetzt z. B.:

BEISPIEL.PAS  
BEISPIEL.BAK  
BEISPIEL.EXE

Die Namen BEISPIEL.\* und BEISPIEL.??? bezeichnen die gleiche Gruppe. Jokerzeichen können auch im Basisnamen verwendet werden. Der Gruppename »\*.COM« bezeichnet alle Dateien mit dem Typnamen »COM« (Kommandodateien), »\*.?\*« oder »????????.???« bezeichnet alle Dateien, und zwar unabhängig von ihrem tatsächlichen Dateinamen.

Die Verwaltung der Dateien auf einem Datenträger erfolgt durch DCP dynamisch anhand einer Belegungstabelle. Die Belegungstabelle unterteilt den Datenträger nach Aufzeichnungseinheiten (Aufzeichnungsblöcke, Cluster) und überwacht, welche Blöcke frei und welche belegt sind. Verzeichniseinträge werden wie Dateien behandelt; deren Komponenten Dateinamen oder weitere Namen von Verzeichnissen sind. Nach außen erscheint eine Baumstruktur, die auch bei Datenträgern großer Kapazität, auf denen einige hundert Dateien gespeichert sein können, dem Nutzer eine übersichtliche Organisation ermöglicht.

Jeder Datenträger besitzt zunächst ein Wurzelverzeichnis (root, Stammverzeichnis). Wenn der Nutzer keine weitere Einteilung in (Unter-)Verzeichnisse vornimmt (Kommando MAKEDIR, kurz MD), werden alle Dateien des Datenträgers nur in dieses Verzeichnis eingetragen. Besonders bei Festplatten erhöht aber die Einteilung in Unterverzeichnisse die Übersichtlichkeit (vgl. Abbildung 3.13.).

Das Wurzelverzeichnis hat keinen Namen. Bei der Angabe eines Pfades zu einem (Unter-)Verzeichnis ist das Wurzelverzeichnis jedoch durch einen Linksschrägstrich (backslash) anzugeben. Die Namen der Verzeichnisse werden vom Nutzer festgelegt. Dabei gelten die gleichen Regeln wie für Dateinamen. Es ist aber üblich, keinen Typnamen zu benutzen.

Befindet sich im Verzeichnis LIBRARY des PASCAL-Zweiges eine Datei GETDIR.BIB, so ist die vollständige Spezifikation auf einer Festplatte (Laufwerk C:) folgende:

C:\PASCAL\LIBRARY\GETDIR.BIB

Der Weg zu einer Datei heißt Zugriffspfad (vgl. Abschnitt 3.1.2.). Der Zugriffspfad schließt den führenden Linksschrägstrich als Zeichen für das Wurzelverzeichnis und den abschließenden Linksschrägstrich zur Öffnung des bezeichneten Verzeichnisses ein. Im Beispiel ist der Zugriffspfad:

\PASCAL\LIBRARY\

Ein Pfad hat eine maximale Länge von 63 Zeichen.

Wie die Datei in einem Kommando zu spezifizieren ist, hängt noch davon ab, an welcher Position des Suchpfades sich der Nutzer zum Zeitpunkt des Kommandos befindet. Diese Position wird vom Steuerprogramm DCPX teilweise oder optional vollständig angezeigt.

Nach einem Systemstart meldet sich DCPX mit der Eingabeaufforderung:

A>\_

Die Kommandoposition wird vor dem Promptzeichen »>« angezeigt. Dort steht als aktueller Parameter des Pfades der Laufwerksbezeichner »A«. Aktuelle Parameter des Pfades können in einem Kommando entfallen.

Da sich GETDIR.BIB im Beispiel aber im Laufwerk C befindet, muß in einem Kommando die vollständige Dateispezifikation angegeben werden.

Das aktuelle Laufwerk kann durch Angabe eines gültigen anderen Laufwerksbezeichners, gefolgt von einem Doppelpunkt, gewechselt werden. Das Kommando

A>C:

wählt nach Verwendung der Endetaste das Laufwerk C als aktuelles Laufwerk. Die Reaktion des Systems ist:

C>\_

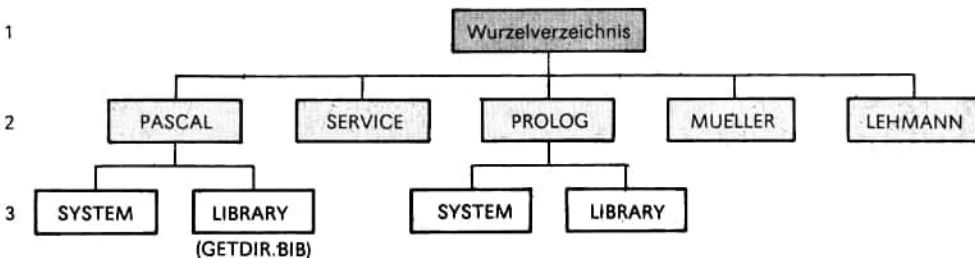


Abbildung 3.13.

Beispiel einer hierarchischen Verzeichnisstruktur (Ebenen 1, 2 und 3)

Zur Spezifikation von GETDIR.BIB in einem Kommando genügt jetzt:

```
\PASCAL\LIBRARY\GETDIR.BIB
```

Der Nutzer kann sich aber auch in Verzeichnisse (den Zugriffspfad) einwählen. Das geschieht mit dem Kommando CHDIR oder CD, gefolgt von einem Verzeichnisnamen, z. B.:

```
C>CD PASCAL
```

Die aktuelle Kommandoposition ist jetzt Laufwerk C:, Verzeichnis PASCAL. Zur Spezifikation von GETDIR.BIB ist nur noch

```
LIBRARY\GETDIR.BIB
```

erforderlich. Wird in das Verzeichnis LIBRARY eingewählt, genügt der Dateiname als Dateispezifikation in einem Kommando. Die Angabe des Zugriffspfades kann entfallen, weil die aktuelle Kommandoposition mit dem Zugriffspfad übereinstimmt.

Die aktuelle Kommandoposition erfährt der Nutzer mit dem Kommando CD (ohne Kommandoparameter). Es ist aber auch möglich, die ständige Anzeige von Laufwerk und Pfad links vom Promptzeichen zu veranlassen. Dazu ist das Kommando

```
PROMPT $P$G
```

erforderlich. Die Anzeige des Zugriffspfades nach der Auswahl des Laufwerkes C: sowie der Unterverzeichnisse PASCAL und LIBRARY ist folgende:

```
C:\PASCAL\LIBRARY>_
```

Durch die Aufnahme des PROMPT-Kommandos in die Steuerkommandodatei AUTOEXEC.BAT, die DCP beim Start auswertet, kann diese Anzeige eingestellt werden.

### 3.3.2.

#### Residente Kommandos (Auswahl)

##### Verzeichnis anzeigen

Die Einträge eines aktuellen oder durch einen Zugriffspfad definierten Verzeichnisses können auf dem Bildschirm mit dem Kommando DIR, z. B.

```
C:\MUELLER >DIR
```

angezeigt werden. Die vollständige Beschreibung dieses Kommandos enthält Anhang B.

Angezeigt werden Namen der Eintragung (Datei oder Unterverzeichnis), Größe einer Datei in Byte, Datum und Uhrzeit des letzten Zugriffs. Abschließend wird der freie Platz des Datenträgers angezeigt (vgl. Abbildung 3.14.).

(Unter-)Verzeichnisse sind mit <DIR> gekennzeichnet.

Als Kommandoparameter kann nach DIR eine Dateispezifikation angegeben werden. Auch Gruppennamen sind zulässig. Das Kommando

```
C:\MUELLER > *.AR
```

zeigt alle Einträge mit dem Typnamen AR an.

Wichtige Optionen (Wahlmöglichkeiten) des DIR-Kommandos sind:

- Anhalten der Ausgabe nach jeder Bildschirmseite; es ist der Kommandopara-

#### Verzeichnis von C:\MUELLER

BERICHT	TXT	368	14-11-87	9:25
BERICHT	BAK	351	14-11-87	9:12
DATEN	AR	11152	14-11-87	20:24
TABELLE	AR	1648	14-11-87	10:29
TABKALK	(DIR)		11-11-87	14:04
GRAFIK	AR	6896	15-11-87	18:39
PLAN	AR	624	14-11-87	12:05
KONZEPT	AR	8384	15-11-87	10:18
8 Datei(en)		3296	16 Byte frei	

Abbildung 3.14.  
Verzeichnisausgabe nach DIR



meter (Schalter) »/P« (page) zu setzen, z. B. DIR A:/P

- Übersichtsausgabe, in der lediglich die Namen angezeigt werden; es ist der Schalter »/W« zu verwenden, z. B. DIR A:/W.

### Dateien löschen

Mit dem Kommando DEL (delete) werden einzelne Dateien oder Dateigruppen auf einer Diskette oder Platte gelöscht. Anstelle von DEL kann auch ERASE verwendet werden. Hinter dem Kommandowort ist eine Dateispezifikation erforderlich. Zum Beispiel löscht das Kommando

```
C>DEL \MUELLER\*.BAK
```

alle BAK-Dateien im Laufwerk C, Verzeichnis MUELLER. Mit dem Kommando

```
C>DEL.\MUELLER\*.*
```

werden alle Einträge des Verzeichnisses MUELLER gelöscht. Bei solchen Kommandos erfolgt eine Rückfrage des Systems, um schwerwiegende Fehler zu vermeiden. Schreibgeschützte Dateien und Verzeichnisse können auf diese Weise nicht gelöscht werden.

### Dateien umbenennen

Zur Reaktivierung von Sicherungsdateien und aus anderen Gründen kann es erforderlich sein, einmal gewählte Dateinamen zu verändern. Dazu wird das Kommando REN (rename) zur Verfügung gestellt. Als Kommandoparameter sind der alte Dateiname und danach der neue Dateiname anzugeben. Die Kommandos

```
C:\MUELLER > DEL BERICHT.TXT<ET>
```

```
C:\MUELLER > REN BERICHT.BAK  
BERICHT.TXT<ET>
```

löschen die Datei BERICHT.TXT und reaktivieren die Sicherungsdatei BERICHT.BAK. Für REN kann auch das Kommandowort RENAME verwendet werden. Als Komman-

doparameter sind auch Gruppennamen zulässig, z. B.:

```
A> REN BEISPIEL.* MUSTER.*<ET>
```

In diesem Fall werden die oben genannten Dateien BEISPIEL.PAS, BEISPIEL.BAK und BEISPIEL.EXE in MUSTER.PAS, MUSTER.BAK und MUSTER.EXE umbenannt.

Verzeichnisnamen können nicht umbenannt, sondern nur gelöscht und danach unter neuem Namen angelegt werden.

### Kopieren von Dateien

Zum Kopieren von Dateien steht das Kommando COPY zur Verfügung. Kopiert wird von einer Quelle (source) zu einem Ziel (destination). Entsprechend sind eine Quell- und eine Zieldateispezifikation erforderlich.

Eine einfache Form ist z. B.:

```
C:\MUELLER > COPY GRAFIK.AR A:
```

Die Datei GRAFIK.AR aus dem (aktuellen) Verzeichnis MUELLER wird in das Wurzelverzeichnis der Diskette im Laufwerk A kopiert. Das Steuerprogramm DCPX erkennt das Kommando COPY durch das Leerzeichen nach dem Kommandowort. Der erste Kommandoparameter GRAFIK.AR reicht bis zum nächsten Leerzeichen und wird als Quellspezifikation angesehen. Danach folgt A: als zweiter Kommandoparameter. Er wird von DCPX als Zielspezifikation interpretiert. Als Quellspezifikation sind auch Gruppennamen erlaubt. Dann werden alle zur Gruppe gehörenden Dateien kopiert. Quell- und Zielspezifikation können auch Gerätedateinamen sein. Das Kommando

```
C>COPY \PASCAL\LIBRARY\GETDIR  
.BIB PRN
```

kopiert das Bibliotheksprogramm GETDIR.BIB zum Drucker, d. h., die (Text-)Datei wird über den Drucker ausgegeben. Der Abschluß eines Kopiervorganges wird mit

1 Datei(en) kopiert

auf dem Bildschirm quittiert.

Mit COPY können auch mehrere Dateien zu einer vereinigt (gekettet) werden.

Das residente Kommando COPY darf nicht mit dem transienten Kommando DISK-COPY verwechselt werden. Die vollständige Beschreibung des Kommandos COPY enthält Anhang B.

### **Unterverzeichnisse anlegen, ändern und löschen**

Für die Benutzung der Unterverzeichnisse stehen drei Kommandos zur Verfügung:

MD (oder MKDIR, make directory) zum Anlegen eines Verzeichnisses,

CD (oder CHDIR, change directory) zum Wechsel des aktuellen Verzeichnisses. Mit »CD\« wird das Wurzelverzeichnis eingestellt,

RD (oder RMDIR, remote directory) zum Entfernen (Löschen) eines Verzeichnisses, wenn es keine Dateien mehr enthält.

Werden die Kommandowörter ohne Parameter benutzt, erfolgt jeweils die Anzeige der aktuellen Kommandoposition auf dem Zugriffspfad. Sonst ist der Name eines Verzeichnisses als Kommandoparameter zu benutzen. Die nächstehende Kommandofolge erzeugt den PASCAL-Zweig in der Verzeichnisstruktur der Abbildung 3.13.:

```
C>MD \PASCAL
C>MD \PASCAL\SYSTEM
C>MD \PASCAL\LIBRARY
```

Direkt in das Verzeichnis LIBRARY kann eingewählt werden durch:

```
C>CD \PASCAL\LIBRARY
```

Bei vollständiger Pfadanzeige ist die Reaktion des Systems:

```
C:\PASCAL\LIBRARY>_
```

Mit »CD C:\« kann zum Wurzelverzeichnis

zurückgekehrt werden. Das Wechseln in das übergeordnete Verzeichnis ist möglich mit den Kommandos »CD\« oder »CD ..«.

Die Wirkung von RD bezieht sich ebenfalls stets auf das als Kommandoparameter spezifizierte Verzeichnis. Das Wurzelverzeichnis kann nicht gelöscht werden.

### **Festlegen von Suchpfaden**

Der Start von Programmen kann nur erfolgen, wenn sich das startbare Programm (startbare Programme tragen die Typnamen EXE, COM oder BAT) im aktuellen Verzeichnis befindet oder der Zugriffspfad im Kommando angegeben wird. Mit dem Kommando PATH besteht die Möglichkeit, diesen Vorgang zu erleichtern. Mit PATH wird ein Suchpfad definiert, der von DCPX in den Suchvorgang nach dem startbaren Programm einbezogen wird, wenn sich die Programmdatei nicht im aktuellen Verzeichnis befindet.

Befindet sich z. B. ein Programm PLUS.COM im Verzeichnis SYSTEM des PASCAL-Zweiges (vgl. Abbildung 3.13.), dann legt das Kommando

```
C:\LEHMANN> PATH C:\PASCAL\SYSTEM
```

einen Suchpfad. Der Programmstart

```
C:\LEHMANN> PLUS
```

wird vom System ausgeführt, obwohl sich PLUS.COM in einem anderen Verzeichnis befindet. Sollte sich doch ein Programm PLUS.COM im aktuellen Verzeichnis befinden, wird dieses Programm gestartet. Auch das Laufwerk kann in den Suchpfad einbezogen werden. Wird das Kommandowort PATH ohne Kommandoparameter benutzt, wird ein früher definierter Suchpfad angezeigt. Durch das Kommando

```
C> PATH;
```

können alle mit PATH getroffenen Vereinbarungen gelöscht werden.

### 3.3.3. Transiente Kommandos und Kommandoverbindungen

#### Formatieren von Datenträgern

Disketten und Festplatten müssen vor der Erstbenutzung formatiert werden. Beim Formatieren werden alle auf dem Datenträger vorhandenen Dateien zerstört. Es wird eine Einteilung in Spuren und Sektoren vorgenommen, die Directories, die Dateizuordnungstabelle und der Systemlader werden initialisiert. Gleichzeitig wird das Speichermedium auf Fehler untersucht. Eine Möglichkeit des Formatierens ist:

C>FORMAT A: /S

#### ACHTUNG!

Niemals das FORMAT-Kommando ohne Laufwerksangabe benutzen. Obwohl das möglich ist, besteht die Gefahr einer unbeabsichtigten Zerstörung des Inhalts einer Diskette oder Festplatte.

Da FORMAT ein transientes Kommando ist, muß sich das Dienstprogramm FORMAT.COM im aktuellen Verzeichnis oder auf einem mit PATH definierten Suchpfad befinden. Die Kommandoausführung beginnt mit der Ausschrift:

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

Es ist zu prüfen, ob sich die richtige Diskette im Laufwerk befindet. Mit ^C kann im Fehlerfall noch abgebrochen werden.

Die Wirkung des Formatbefehls kann durch Schalter beeinflusst werden. Möglich sind z. B. die folgenden Schalter:

/S (system). Die DCPX-Systemkomponenten (BIO.SYS, DOS.SYS und COMMAND.COM) werden als verborgene Dateien auf den neuformatierten Datenträger kopiert,

/V (volume label). Der Datenträger erhält einen Namen, der nach dem Kommando DIR angezeigt wird.

Die Formatierung dauert einige Minuten. Der Nutzer wird über den Fortgang der Arbeiten laufend informiert. Nach der Formatierung gibt das Dienstprogramm z. B. den folgenden Text aus:

Formatieren beendet.

362496 Byte Gesamtspeicherbereich

362496 Byte auf Diskette/Platte verfügbar.

Falls fehlerhafte Sektoren auf der Diskette gefunden werden, wird das angezeigt. Defekte Bereiche wurden vom System pseudobelegt und so für die Benutzung gesperrt.

#### Duplizieren von Disketteninhalten

Der gesamte Inhalt einer Diskette kann mit dem Kommando DISKCOPY vollständig auf eine andere Diskette übertragen werden. Ein Beispiel ist:

A> DISKCOPY A: B:

Das Dienstprogramm DISKCOPY.COM muß im aktuellen Verzeichnis oder auf einem mit PATH definierten Suchpfad sein.

Der erste Kommandoparameter gibt das Quellaufwerk (»A:«), der zweite Kommandoparameter das Zielaufwerk (»B:«) an. Fehlt der (zweite) Kommandoparameter mit der Zielaufwerksangabe, wird das aktuelle Laufwerk als Zielaufwerk angesehen. Es kann auch mit einem Laufwerk kopiert werden. In diesem Fall sind entweder zwei gleiche Laufwerksbezeichner anzugeben oder die Kommandoparameter sind wegzulassen. Beim Kopieren der Disketten in einem Laufwerk wird der Disketteninhalt abschnittsweise im Hauptspeicher zwischengespeichert, und es wird wiederholt zum Diskettenwechsel aufgefordert. Ist die Diskette im Zielaufwerk eine neue Diskette, so wird zugleich formatiert (vgl. FORMAT-Kommando).

Mit dem Kommando DISKCOPY kann nicht von oder zu einer Festplatte und nicht dateiweise kopiert werden.

### Setzen von Dateiattributen

Das Anzeigen, Setzen oder Rücksetzen von Dateiattributen (vgl. Abschnitt 3.3.1.) erfolgt mit dem Kommando ATTRIB. Nach dem Kommandowort wird der Parameter »R+« verwendet, wenn eine Datei das R/O-Attribut (Schreibschutz) erhalten soll. Dateien mit diesem Attribut können nicht verändert und nicht gelöscht werden. Mit »R-« wird der Schreibschutz zurückgesetzt. Mit dem zweiten Kommandoparameter ist die Datei zu spezifizieren, die das im ersten Parameter aufgeführte Attribut erhalten soll. Das Kommando

```
C>ATTRIB R+ A:\PASCAL\LIBRARY\
GETDIR.BIB
```

setzt den Schreibschutz für GETDIR.BIB in der Verzeichnishierarchie der Abbildung 3.13. Es können auch Gruppennamen verwendet werden. Dann wird das Attribut für alle Dateien der Gruppe gesetzt oder zurückgesetzt.

Enthält das Kommando keinen Parameter, der das Attribut kennzeichnet, z. B.

```
C>ATTRIB A:\PASCAL\LIBRARY\*.BIB
```

so werden die Dateiattribute, hier aller Dateien mit dem Typnamen BIB im Verzeichnis LIBRARY, mit ihren aktuellen Attributen angezeigt. Da ATTRIB ein transientes Kommando ist, muß sich das Dienstprogramm ATTRIB.EXE bei dieser Form des Kommandos im aktuellen Verzeichnis oder auf einem mit PATH definierten Suchpfad befinden. Sonst ist vor dem Kommandowort der Suchpfad anzugeben.

### Datenumleitung

Das Steuerprogramm DCPX verwendet die Tastatur und den Bildschirm als Standardgeräte der Ein- und Ausgabe. Die Standard-

gerätezuweisung kann durch Datenumleitung verändert werden. Zur Datenumleitung sind folgende Zeichen zu verwenden, deren Spitze auf das Ziel und deren Öffnung auf die Quelle der Information zeigt:

```
< Umleitung der Standardeingabe
> Umleitung der Standardausgabe
>> Umleitung der Standardausgabe an das
    Ende einer Datei
```

Im Kommando ist die Umleitung rechts vom Umleitungszeichen zu spezifizieren. Es kann ein Speicherdateiname oder ein Gerätedateiname (außer CON, denn das ist bereits die Standardzuweisung) sein. Eine Möglichkeit ist:

```
C>DIR \PASCAL\LIBRARY\*. *
>A:INHALT.LIB
```

Das Verzeichnis der PASCAL-Bibliothek wird nicht auf dem Bildschirm gelistet, sondern als Datei INHALT.LIB auf der Diskette im Laufwerk A (Wurzelverzeichnis) abgelegt.

Bei Verwendung des Zeichens »>>><<< statt »>><<< ist es möglich, die Ausgabe an das Ende einer bereits existierenden Datei INHALT.LIB anzufügen. Existiert die Datei noch nicht, ist die Wirkung des Zeichens »>>><<< wie die des Zeichens »>><<<.

Durch das Kommando

```
C>DIR \PASCAL\LIBRARY\*. * >PRN
```

kann die Ausgabe auch an den Drucker gesendet werden.

Das Umleitungszeichen »<<< dient der Umleitung der Eingabe. Wird das Kommando

```
C>PLANUNG < STANDARD.TST
```

erteilt, so erfolgen alle Eingaben für das Programm PLANUNG, die sonst von der Tastatur kommen, aus der Datei STANDARD.TST. Die Datei muß alle Eingaben für PLANUNG enthalten, weil keine Rück-

schaltung zur Tastatur erfolgt, bevor das Kommando ausgeführt ist. Bei fehlenden Eingaben kommt es zu einem Wartezustand, der nur mit den Tasten <CTRL><BREAK> beendet werden kann.

**Befehlsverkettung mit Filtern**

**Filter sind transiente Kommandos, die Information vom Standardeingabegerät lesen, entsprechend den Kommandoparametern bearbeiten und das Ergebnis an das Standardausgabegerät weitergeben.**

Filter können allein oder in Verbindung mit anderen Kommandos verwendet werden. Es sind folgende Filter vorhanden:

**SORT** sortiert eine Textdatei in steigender oder fallender Reihenfolge der Zeilen. Nach SORT kann mit dem Schalter »/R« die fallende Reihenfolge gefordert werden. Mit Schalter »/+10« wird erreicht, daß erst ab zehnter (allgemein /+n) Position der Zeile sortiert wird (Standard ist n = 1).

**FIND** durchsucht eine Textdatei nach einer angegebenen Zeichenkette und stellt die Zeilen zur Verfügung, in denen die Zeichenkette auftritt. Kommandoparameter ist eine Zeichenkette, z. B. »PROGRAM«; es können Schalter gesetzt werden, die die Ausgabe beeinflussen.

**MORE** übergibt eine Textdatei in Abschnitten zu je einer Bildschirmseite. Nach jeder Seite wird pausiert und zur Fortsetzung mit einer beliebigen Taste aufgefordert (blättern).

Zur Verkettung der Filter mit anderen Kommandos (piping) steht der Pipe-Operator »|« zur Verfügung. Das Zeichen »|« wird durch die Tastenkombination <ALT>1 2 4 mit dem numerischen Tastenfeld erzeugt. Ein Beispiel ist:

```
C:\MUELLER > PATH A:
C:\MUELLER > DIR | SORT
```

Die Definition eines Suchpfades ist notwendig, weil sich das Diebstprogramm SORT.EXE als transientes Kommando im Wurzelverzeichnis auf der Diskette des Laufwerks A befindet. Die Dateien des Verzeichnisses MUELLER (vgl. Abschnitt 3.3.2.) werden in alphabetischer Reihenfolge angezeigt (vgl. Abbildung 3.15.). Es sind auch mehrfache Kommandoverkettungen möglich, z. B.:

```
C:\MUELLER > DIR | FIND »AR« |
SORT | MORE
```

Das Verzeichnis wird diesmal nach der Zeichenkette »AR« durchsucht, sortiert und seitenweise auf dem Bildschirm angezeigt. Dabei sind die Einträge des Verzeichnisses MUELLER Eingabeinformation für FIND. FIND filtert alle Einträge mit dem Typnamen »AR« heraus und übergibt sie an den Filter SORT. SORT übergibt die sortierten Einträge an MORE, das die seitenweise Anzeige der Einträge auf dem Bildschirm vornimmt (vgl. Abbildung 3.16.).

BERICHT	BAK	351	14-11-87	9:12
BERICHT	TXT	368	14-11-87	9:25
DATEN	AR	11152	14-11-87	20:24
GRAFIK	AR	6896	15-11-87	18:39
KONZEPT	AR	8384	15-11-87	10:18
PLAN	AR	624	14-11-87	12:05
TABELLE	AR	1648	14-11-87	10:29
TABKALK	(DIR)		11-11-87	14:04

Abbildung 3.15.  
Sortierte Verzeichnisausgabe nach DIR

DATEN	AR	11152	14-11-87	20:24
GRAFIK	AR	6896	15-11-87	18:39
KONZEPT	AR	8384	15-11-87	10:18
PLAN	AR	624	14-11-87	12:05
TABELLE	AR	1648	14-11-87	10:29

Abbildung 3.16.  
Gefilterte Verzeichnisausgabe

### Weitere Möglichkeiten

Das Betriebssystem DCP bietet die Möglichkeit, Kommandofolgen als Zeilen einer Textdatei mit dem Typnamen »BAT« abzu-legen. Die Ausführung aller Kommandos wird durch Eingabe des Basisnamens der BAT-Datei veranlaßt. Den Übergang von einem zum anderen Kommando realisiert das Steuerprogramm DCPX, so daß kein Eingreifen des Nutzers erforderlich ist. In BAT-Dateien können alle residenten und transienten Kommandos, Umleitungen und Filter verwendet werden. Zu ihrer Verknüpfung stehen die folgenden speziellen (residenten) Stapelverarbeitungskommandos (Unterkommandos) zur Verfügung, die mit formalen Parametern notiert und mit wechselnden aktuellen Parametern ausgeführt werden können (vgl. Abschnitt 3.1.2.):

- ECHO** aktiviert (ECHO ON) oder reaktiviert (ECHO OFF) die Anzeige des Kommandos während der Kommandoausführung auf dem Bildschirm. Stehen andere Zeichen als ON oder OFF hinter dem Unterkommando ECHO, so werden diese auf dem Bildschirm angezeigt.
- FOR** ermöglicht die wiederholte Ausführung von Kommandos, gegebenenfalls mit wechselnden Variablen.
- GOTO** ermöglicht die Fortsetzung der Kommandoausführung mit einem anderen als dem in der BAT-Datei folgenden Kommando. Nach GOTO wird eine Marke notiert. Die Kommandoausführung wird mit der Kommandozeile fortgesetzt, der diese Marke vorangestellt ist.
- IF** gestattet es, die Ausführung eines Kommandos an eine Bedingung zu binden.
- PAUSE** stoppt die Kommandoausführung und fordert den Nutzer auf, mit

einer beliebigen Taste die Fortsetzung zu veranlassen.

**REM** zeigt den nach REM notierten Text bei ECHO ON auf dem Bildschirm an.

REM <Kommentar> bietet die Möglichkeit, bis zu 123 Zeichen anzuzeigen (nur bei ECHO ON).

**SHIFT** verschiebt die Zuordnung aktueller Kommandoparameter zu eventuell notierten formalen Parametern um eine Position.

Stapeldateien lassen sich mit jedem Texteditor oder direkt von der Tastatur nach dem Kommando

COPY CON <Basisname>.BAT

eingeben (Kopieren der Tastatureingabe in die angegebene Datei). Die Eingabe ist mit <CTRL> Z oder mit der Funktionstaste F6 abzuschließen. <Basisname> darf jeder beliebige Basisname sein, außer AUTOEXEC. Die Stapeldatei AUTOEXEC.BAT wird beim Systemstart des Betriebssystems automatisch ausgeführt. Der Nutzer kann durch Manipulation dieser Datei den Systemstart an seine Anwendungsbedingungen anpassen.

### Übungen

1. Beschreiben Sie Stellung und Funktion von Betriebssystemen bei der Entwicklung und Ausführung von Anwenderprogrammen!
2. Vergleichen Sie Stapelverarbeitung und Dialogverarbeitung, und begründen Sie, warum Stapelverarbeitung für geeignete Aufgaben vorzuziehen ist!
3. Tragen Sie in das folgende Schema ein, welche Programme sich in der jeweiligen Arbeitsphase im internen Speicher des Rechnersystems befinden müssen!

Arbeitsphase	Programm- bezeichnung
Vor dem Einschalten des Gerätes	
Nach Abschluß des Umladevorgangs	
Nach Meldung des Steuerprogramms	
Während der Ausführung des Dienstprogramms FORMAT	
Während der Ausführung des Anwenderprogramms BEISPIEL	
Nach Ausführung des Steuerzeichenkommandos <CTRL> <ALT> <DEL>	

Dialog, und wählen Sie selbst die Dialogtechnik für folgende Aufgaben:

- Materialdisposition,
- Buchführung,
- Transportoptimierung,
- Bedarfsvorhersage!

5. Notieren Sie alle DCP-Kommandos, die zum Anlegen der folgenden Verzeichnisstruktur auf einer neu formatierten Diskette im Laufwerk A erforderlich sind:



6. Notieren Sie DCP-Kommandos, die bei der in Übung 5 gegebenen Dateistruktur von der Kommandoposition des Wurzelverzeichnisses aus die Datei ABSATZ.TXT aus dem Verzeichnis MONAT in das Verzeichnis ARCHIV kopieren und im Verzeichnis MONAT löschen!

4. Erläutern Sie Vor- und Nachteile des menügesteuerten Dialogs! Nennen Sie Kriterien für die Auswahl zwischen menü- oder kommandogesteuertem

## 4. Entwicklung von Anwendersoftware

---

### 4.1. Anwendersoftware

Anwendersoftware ist Software zur Bearbeitung spezifischer Anwenderprobleme (vgl. Abschnitt 1.3.1.). Im erweiterten Schichtenmodell der Nutzung eines Rechnersystems (vgl. Abschnitt 3.1.1.) setzt Anwendersoftware als transiente Software auf dem Steuerprogramm (Betriebssystemkern) auf und bildet die Schnittstelle zum Nutzer (vgl. Abbildung 4.1.).

Nach dem Start des Anwenderprogramms vollzieht sich die Nutzung des Rechnersystems ausschließlich über die Anwendersoftware. Die Anwendersoftware realisiert damit den Gebrauchswert des gesamten Rechnersystems. Die Klassifikation der Anwendersoftware geht aus Abbildung 4.2. hervor.

**Standardsoftware (packaged software, Standardanwendersoftware, bereichsübergreifend nutzbare Software) ist Anwendersoftware, die (unverändert oder nach Generierung) zur Mehrfachnutzung für unterschiedliche Basisprozesse geeignet ist.**

Der Aufwand für Standardsoftware ist infolge der Mehrfachnutzung relativ gering. Standardsoftware zeichnet sich in der Regel durch hohe Zuverlässigkeit aus, da die Entwicklung durch Spezialisten erfolgt. Ein Nachteil der Standardsoftware ist, daß sie nicht an den einzelnen Anwendungsfall

vollständig angepaßt sein kann. Das ist oft mit Restriktionen für die Automatisierungslösung verbunden und kann zu Problemen in der Akzeptanz von Standardsoftware führen.

**Individualsoftware (custom software, bereichsspezifische Software) ist Anwendersoftware, die für einen speziellen Basissystem entwickelt und eingesetzt wird.**

Die Softwarenomenklatur der DDR klassifiziert die Individualsoftware auch nach volkswirtschaftlichen Bereichen, z. B.

- Leichtindustrie,
- Verkehrswesen,
- Chemie,
- Handel und Versorgung.

Der Vorteil der Individualsoftware liegt in der genauen Anpassung an die Bedingungen und Erfordernisse der Basissysteme. Ihr Nachteil ist der relativ hohe Aufwand für Entwicklung und Wartung.

Die Entwicklung von Anwendersoftware hat grundsätzlich so zu erfolgen, daß eine Mehrfachnutzung als Ganzes oder bestimmter Teile möglich ist. Das geschieht, um den gesellschaftlichen Aufwand zu reduzieren. Diese Vorgehensweise führt zu dem Zwischenbegriff *Typsoftware* (Typprojekt, Software mit Typcharakter). Er bezeichnet Anwendersoftware, die innerhalb eines Bereiches mehrfach genutzt wird.

Die Zusammenführung der Vorteile von Standardsoftware und Individualsoftware ist über Programmbibliotheken möglich, in



denen Moduln mit Anwendungscharakter zur Verfügung stehen. Diese Moduln können durch Programmierer entsprechend den speziellen Bedingungen individuell zusammengefügt und komplettiert werden. Diese Vorgehensweise ist eng mit der Vervollkommenung der Technologie der Entwicklung von Anwendersoftware verbunden.

Entwickler von Anwendersoftware haben vor dem Beginn der Arbeiten zu recherchieren, ob die Nutzung von Standardsoftware oder die Nachnutzung von Individualsoftware möglich ist. Entwicklungsvorhaben sind bilanzierungspflichtig.

### Qualität von Anwendersoftware

Eine Automatisierungslösung wird maßgeblich von der Qualität der Anwendersoftware beeinflusst.

**Die Qualität der Anwendersoftware ist die Gesamtheit der Gebrauchseigenschaften, die die potentiellen Leistungsmöglichkeiten eines Rechnersystems für den speziellen Anwendungszweck erschließen.**

Qualität der Anwendersoftware kommt in Merkmalen des jeweiligen Softwareerzeugnisses zum Ausdruck. Es werden entwicklungs- und nutzungsorientierte Merkmale unterschieden (vgl. Abbildung 4.3.). Einige Merkmale sind in der TGL 44 530<sup>1</sup> verbindlich festgelegt.

Die *Verwertbarkeit* der Ausgabeinformation ist der eigentliche Sinn der Nutzung eines Rechnersystems und deshalb das wichtigste Qualitätsmerkmal eines Softwareerzeugnisses. Wege zur Beurteilung der Verwertbarkeit führen über den Begriff des Verhaltens eines Informationsempfängers. Informationsempfänger verfügen über mehrere mögliche Verhaltensvarianten. Zielgerichtetes Verhalten setzt die Verringerung von Unbestimmtheit bezüglich möglicher Verhaltensvarianten voraus.

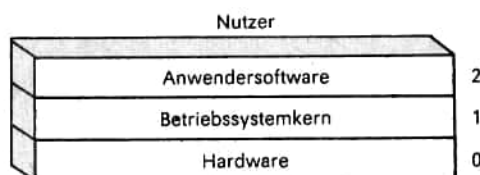


Abbildung 4.1.  
Schichtenmodell  
der anwendungsspezifischen Nutzung  
eines Rechnersystems

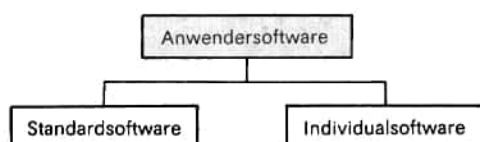


Abbildung 4.2.  
Klassifikation von Anwendersoftware

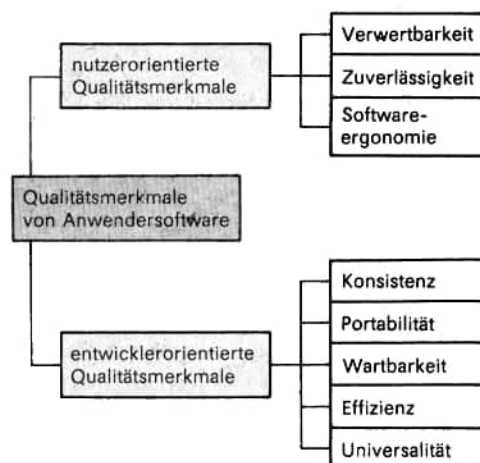


Abbildung 4.3.  
Ausgewählte Qualitätsmerkmale  
von Anwendersoftware

<sup>1</sup> Fachbereichsstandard TGL 44 530/28: Informationsverarbeitung / Termini und Definitionen für Softwarequalitätssicherung

Diese Verringerung erfolgt durch Information (vgl. Abschnitt 1.1.). Erheblicher Mangel an Information führt zu einer zunehmend zufälligen Auswahl von Verhaltensvarianten. Die Verwertbarkeit von Information kann beurteilt werden, indem bestimmt wird, in welchem Maße Information dazu beiträgt, mögliche Verhaltensvarianten zu reduzieren.

Beim Menschen wird das Verhalten nicht nur durch laufende Information aus dem Rechnersystem, sondern auch durch Vorinformation, darunter Information aus anderer Quelle, und durch Erfahrungen wesentlich beeinflusst.

*Zuverlässigkeit* bezeichnet die Eigenschaft eines Softwareerzeugnisses, Fehler und unzulässige Zugriffe zu vermeiden oder gezielt darauf zu reagieren. Diese Eigenschaft der Anwendersoftware wird erreicht durch Plausibilitätskontrollen für Eingabedaten, Sicherungsmaßnahmen während der Verarbeitung, Speicherung und Übertragung von Information, Maßnahmen gegen unberechtigten Zugriff sowie ein System von Fehlermitteilungen und Fehlerbehandlungsmaßnahmen.

Die *Ergonomie eines Softwareerzeugnisses* (Softwareergonomie) charakterisiert die menschlichen Arbeitsbedingungen für die Nutzung des Softwareerzeugnisses. Bei Anwendersoftware für die Stapelverarbeitung reduzieren sich die ergonomischen Gesichtspunkte weitgehend auf den Drucklistenaufbau. Bei Dialogverarbeitung sind sie wesentlich vielfältiger. Zu den ergonomischen Gesichtspunkten bei Dialogverarbeitung gehören z. B. die Dialogtechnik, der Bildschirmaufbau, die Zeichendarstellung für Texte und Grafikelemente und Unterstützungen des Nutzers im Fall einer Hilfeanforderung (HELP-Funktion). Ein sehr wichtiger Gesichtspunkt ist auch die ausführliche, für den Nutzer verständliche Anwenderdokumentation. Die Softwareergo-

nomie entscheidet wesentlich über die Akzeptanz einer Automatisierungslösung.

Die *Konsistenz* (Beständigkeit) eines Softwareerzeugnisses ist eine Eigenschaft, die die Arbeitsfähigkeit auch dann gewährleistet, wenn sich Veränderungen in unteren Schichten (Betriebssystemkern, Hardware) oder im Basissystem (vgl. Abschnitt 1.3.1.) vollziehen. Diese Eigenschaft wird während der Entwicklung des Softwareerzeugnisses erreicht, indem hardware- und betriebssystemabhängige Teile minimiert, eine interne Anpassung (Adaption) an sich verändernde Bedingungen programmiert oder eine Installation (mit einem gesonderten Installationsprogramm) in Abhängigkeit von konkreten Bedingungen vorgesehen wird. Die Konsistenz kann an der Zeitspanne gemessen werden, in der kein ändernder Eingriff in das Softwareerzeugnis erfolgen muß. Allerdings kann kein Normwert für ein Softwareerzeugnis vorgegeben werden, da die tatsächliche Konsistenz immer nur in bestimmten Grenzen möglich ist, die von äußeren Einflüssen abhängig sind.

Die *Portabilität* (Übertragbarkeit) ist von erheblicher Bedeutung, wenn die Konsistenzgrenzen überschritten werden. Diese Eigenschaft kennzeichnet, ob oder in welchem Maße ein Softwareerzeugnis verändert werden muß, wenn die darunterliegende Schicht (Betriebssystemkern, Hardware) wechselt. Die Portabilität wird während der Softwareentwicklung erreicht durch die Wahl einer perspektivisch gesicherten, maschinenunabhängigen (höheren) Programmiersprache und möglichst wenigen Einbindungen betriebssystem- oder maschinenabhängiger Sprachelemente. Vollständige Portabilität ist in der Regel nicht möglich, da dann auf die volle Ressourcenausnutzung weiterentwickelter Maschinen verzichtet werden müßte. Es ist

aber dafür zu sorgen, daß nichtportable Teile separat und leicht auswechselbar sind. Angesichts des hohen Tempos des wissenschaftlich-technischen Fortschritts ist Portabilität eine wichtige Forderung an Anwendersoftware.

Die *Wartbarkeit* eines Softwareerzeugnisses ist gekennzeichnet durch Änderbarkeit und Erweiterbarkeit. 50 bis 70 % des Aufwandes für Anwendersoftware entfallen auf die Wartung (vgl. Abschnitt 4.2.1.). Die Wartung erfordert einen kontrollierten Eingriff in das Softwareerzeugnis. Es muß gesichert sein, daß dadurch mögliche (ungewollte) Nebeneffekte unbedingt auf einen kleinen Bereich beschränkt bleiben. Voraussetzung der Wartbarkeit sind daher transparenter, gut lesbarer Programmcode sowie ein modularer Aufbau des Erzeugnisses mit klarer Funktionsabgrenzung und definierten Schnittstellen. Dabei sind die Anzahl der Funktionen je Modul und die Verbindungen zwischen den Modulen gering zu halten. Die Wartung setzt eine vollständige und aktuelle Programmdokumentation voraus.

Mit *Effizienz* wird das Speicherplatz- und Laufzeitverhalten eines Softwareerzeugnisses bezeichnet. Effizienz tritt mit fortschreitender Entwicklung der Hardware hinter Konsistenz, Portabilität und Wartbarkeit zurück. Allerdings kann Speicherplatzbedarf bei begrenzten Ressourcen die Ausführung eines Programms unmöglich machen, und die Laufzeit kann zu nicht vertretbaren Reaktionszeiten im Dialogbetrieb führen. Innerhalb bestimmter Grenzen besteht die Möglichkeit, entweder Speicherplatz zu Lasten der Laufzeit zu verringern oder Laufzeit zu Lasten des Speicherplatzbedarfs zu verkürzen. Lassen sich trotzdem Speicher- oder laufzeitkritische Engpässe nicht beseitigen, erhält die Effizienz die Priorität vor anderen Qualitätsmerkmalen.

Die *Universalität* beschreibt die Anwendungsbreite und damit die Nachnutzbarkeit eines Softwareerzeugnisses. Diese Eigenschaft muß, wenn potentiell weitere Anwender vorhanden sind, bereits vor und während der Entwicklung berücksichtigt werden. Es genügt nicht, nach der Fertigstellung die Nachnutzbarkeit zu prüfen. Bei potentiellen Nachnutzungsmöglichkeiten sind in der Regel einige Teile variabel zu halten, da die Anwendungsbereiche niemals völlig gleich sind. Nachnutzbarkeit erfordert also in der Regel auch geringen Mehraufwand bei der Entwicklung. Nachnutzung wird gesondert vergütet.

### Qualitätssicherung

Qualitätsforderungen an ein Softwareerzeugnis sind bereits bei der Formulierung der Aufgabenstellung (vgl. Abschnitt 4.2.1.) festzulegen und während der gesamten Softwareentwicklung regelmäßig zu kontrollieren. Der Aufwand zur Beseitigung von erkannten Qualitätsmängeln ist um so größer, je später sie festgestellt werden. Bei der Formulierung von Qualitätsforderungen und zur Qualitätskontrolle haben sich *Checklisten* bewährt. Sie sind nach Qualitätsmerkmalen und/oder Entwicklungsphasen differenziert. Jede Checkliste enthält eine Reihe von Kriterien, auf die es jeweils drei mögliche Antworten gibt (vgl. Abbildung 4.4. auf Seite 112):

INDIFFERENT	Das Kriterium ist für die Qualität des Softwareerzeugnisses nicht von Bedeutung oder nachweisbar technisch nicht realisierbar.
JA	Das Kriterium ist nicht INDIFFERENT und für das Erzeugnis eingehalten.
NEIN	Das Kriterium ist nicht INDIFFERENT und für das Erzeugnis nicht eingehalten.

Frage	Antwort		
	I	J	N
1. Wird der Dialog mit einem Eröffnungsbild begonnen?			
2. Erfolgt vor jeder Eingabe eine Eingabeaufforderung?			
3. Wird Statusinformation angezeigt?			
4. Ist der Kommandovorrat einschließlich Endemöglichkeit sichtbar?			
5. Ist der Bildschirm in Teilbereiche (Fenster) aufgeteilt?			
6. Wird Groß- und Kleinschreibung benutzt?			
7. Wird vollständig auf Abkürzungen verzichtet?			
8. Werden die Möglichkeiten zur Hervorhebung von Teiltexen genutzt (invers, intensiv, Farbe)?			
9. Werden weniger als 50 % der Bildschirmoberfläche beschrieben?			
10. Wird unterschiedlichen Fertigkeiten verschiedener Nutzer entsprochen?			
11. Wird eine Helpfunktion angeboten?			
12. Ist zu jeder Zeit eine Beendigung des Dialogs möglich?			
13. Werden Möglichkeiten zur Reduzierung der Eingabe genutzt (Standards, Eingabeangebote über Menü)?			
14. Kann der Bildschirminhalt auf dem Drucker (hardcopy) protokolliert werden?			

Abbildung 4.4.

Checkliste Softwareergonomie / Dialogbetrieb (Auszug)

Es empfiehlt sich, die INDIFFERENT-Spalte zum Zeitpunkt der Planung auszufüllen und als Bestandteil eines Pflichtenheftes aufzufassen.

Es gibt weitere Methoden für die Qualitätssicherung (TGL 44 530). Wichtig ist, daß die Qualitätssicherung als ein durchgängiges Prinzip in allen Entwicklungsphasen von Anwendersoftware durchgesetzt wird.

## 4.2. Entwicklungsphasen und softwaretechnologische Aspekte

### 4.2.1. Phasenmodell

Ein Softwareerzeugnis durchläuft von der Entstehung bis zur Aussonderung charakteristische Phasen, die in ihrer Gesamtheit den *Softwarelebenszyklus* bilden.

**Eine Phase im Softwarelebenszyklus (Softwarephase) ist ein inhaltlich, technologisch, organisatorisch und zeitlich abgegrenzter Prozeß mit einem charakteristischen Ergebnis.**

Die Softwarephasen und ihre grundsätzliche Aufeinanderfolge bilden das *Phasenmodell* nach Fachbereichsstandard TGL 44 545 in Abbildung 4.5.

Nach dem Phasenmodell können die Phasen Analysieren, Entwerfen, Implementieren, Testen und Fertigstellen als Entwicklungsphasen und die Phasen Einführen, Betreiben und Warten als Anwendungsphasen bezeichnet werden.

### Analysieren

**Analysieren (Spezifizieren) ist eine Softwarephase, in der die Basisprozesse untersucht, die grundlegenden Funktionen des Softwareerzeugnisses spezifiziert und Einsatzbedingungen festgelegt werden.**

Beim Analysieren werden die ökonomischen, technologischen sowie arbeitsorganisatorischen Abläufe und Bedingungen im künftigen Anwendungsbereich untersucht, die Möglichkeiten ihrer Automatisierung durch den Einsatz verfügbarer Rechnersysteme geprüft sowie die erforderliche Leistung des Softwareerzeugnisses abgegrenzt. Diese Tätigkeit erfolgt in Kollektiven, denen neben den Entwicklern der Anwendersoftware unbedingt Leitungskader und direkte Nutzer des Anwenderbereiches angehören sollten. Das Ergebnis dieser Tätigkeit ist die Spezifikation, für die das in Abbildung 4.6. dargestellte Deckblatt verwendbar ist.<sup>2</sup>

Für die Entwicklung von Anwendersoftware gilt folgender Grundsatz:

**Der Detaillierungsgrad verbindlicher Aussagen innerhalb einer Entwicklungsphase ist so gering zu halten, wie es der Übergang zur folgenden Phase gerade zuläßt.**

Details sind also so spät wie möglich zu entscheiden. Dadurch wird die vorzeitige Einengung des Entwicklers vermieden, der Aufwand für Änderungen gesenkt und die konkreten Tätigkeiten in die Entwicklungsphasen verlagert, in denen dafür die zweckmäßigsten Mittel zur Verfügung stehen. So ist es z. B. nicht günstig, bereits in der Spezifikation detailliert Datenstrukturen oder Programmabläufe zu beschreiben. Die Darstellung sollte grob und für alle Partner verständlich sein. Das gilt natürlich nicht für solche Teile, die nur in dieser Phase bearbeitet werden, wie Aufwands- und Nutzenskalkulationen.

Beim Analysieren nehmen Aussagen zur Einordnung der Automatisierungslösung in ein Gesamtsystem eine besondere Stellung ein. Sie sind Grundlage für die Abgrenzung zur Umgebung, besonders die Bereitstellung von Daten, Zugangsbedingungen

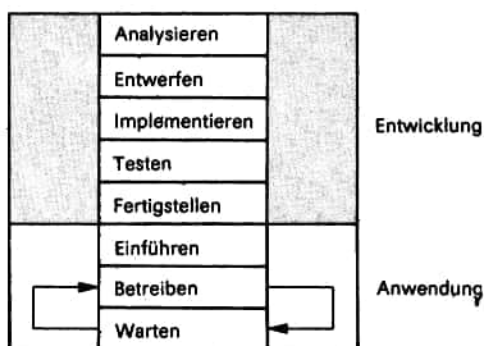


Abbildung 4.5.  
Phasenmodell des Softwarelebenszyklus

SPEZIFIKATION zum Erzeugnis ..... Stand: .....
1. Bestätigungsvermerk 2. Kurzbeschreibung 3. Ziel und Ablauf der Softwareentwicklung 3.1. Auftrag und Begründung 3.2. Inhaltliche Aufgabenstellung 3.3. Aufwand und Nutzen 3.4. Kopierrecht und Datensicherheit 3.5. Kooperation, Entwicklungssetappen und Termin 4. Schnittstellen des Softwareerzeugnisses 4.1. Einordnung in das Gesamtsystem 4.2. Hardware und Betriebssystem 4.3. Nutzerschnittstelle 4.4. Kaderentwicklung und Qualifikation 4.5. Softwaretechnologische Vorgaben 5. Grundeigenschaften des Softwareerzeugnisses 5.1. Funktionen 5.2. Datenstrukturen 5.3. Qualitätsparameter Anhang (Beispiele, Begriffe)

Abbildung 4.6.  
Deckblatt einer Spezifikation

<sup>2</sup> Vgl. Rothhardt, G.: Praxis der Softwareentwicklung. Berlin: VEB Verlag Technik 1987, S. 163.

zum Rechnerbetrieb sowie Schnittstellen zu anderen Automatisierungslösungen (Organisationskonzept). Die Erarbeitung des Organisationskonzepts ist nicht Bestandteil der Softwareentwicklung.

Um unnötigen Aufwand für Fehlentwicklungen zu vermeiden, ist es erforderlich, daß die Spezifikation von Auftraggeber und Auftragnehmer als verbindliche Grundlage der Softwareentwicklung *schriftlich* vereinbart wird und jede Änderung an die Schriftform zu binden ist. Unkontrollierte Änderungsforderungen können die Produktivität der Softwareentwicklung erheblich mindern.

### Entwerfen

**Entwerfen ist eine Softwarephase, in der der Funktionsumfang des künftigen Softwareerzeugnisses, die Elemente und die Struktur sowie die Kommunikation mit der Umgebung erarbeitet werden.**

Durch das Entwerfen wird ein Modell der programmtechnischen, datenorganisatorischen und arbeitsorganisatorischen Lösung, der Entwurf, geschaffen. Der Entwurf beinhaltet:

1. Übersicht über die Elemente (Moduln) des künftigen Softwareerzeugnisses und die Beziehungen zwischen ihnen (Strukturübersicht);
2. Beschreibung jedes Moduls hinsichtlich Eingabe- und Ausgabeinformation sowie des Algorithmus für die Überführung der Eingabe- in die Ausgabeinformation (Modulbeschreibung);
3. Beschreibung der Daten in abstrakter Form (Datenkapsel), d. h. noch nicht detailliert nach Typ (Wertevorrat) und innerer Struktur (Datenbeschreibung);
4. Konzept der Erkennung, Mitteilung und Behandlung von Ein- und Ausgabefehlern sowie von Laufzeitfehlern (Fehlerkonzept);

5. Konzept zur Gestaltung der Schnittstelle zum Nutzer, darunter Bildschirmaufteilung, Eingabeaufforderungen, Abbruchrealisierung und Unterstützungsfunktionen im Dialogbetrieb (Dialogkonzept) oder im Stapelbetrieb (Drucklistenentwurf).

Die Erarbeitung des Entwurfs erfolgt in den Schritten fachlich-logisches Entwerfen (Grobentwurf) und programmtechnisches Entwerfen (Feinentwurf).

Das *fachlich-logische Entwerfen* hat zum Ziel, aus der problemorientierten Sicht der Softwareanwender die Funktionen durch Ausgabeinformation, Algorithmen und Eingabeinformation sowie hinsichtlich ihrer Einordnung in die Gesamtstruktur zu beschreiben. Jede Funktion entspricht dabei einem problemorientierten Modul. Außerdem ist das Organisationskonzept fertigzustellen.

Das *programmtechnische Entwerfen* verfeinert den fachlich-logischen Entwurf. Dabei werden Vorgänge, die sich vom Algorithmus her in den problemorientierten Moduln wiederholen, herausgelöst, und es werden Querschnittsmoduln gebildet, die von allen problemorientierten Moduln aktiviert werden können. Vorgänge mit Querschnittscharakter sind z. B. die Nutzereingabe mit Erzeugung der Eingabeaufforderung, Fehlererkennung und Abbruchrealisierung, die Eröffnung und der Zugriff auf Dateien, die Erzeugung von Bildschirmmasken, mathematische oder statistische Verfahren.

Es gilt immer der Grundsatz, daß der Detaillierungsgrad möglichst grob gehalten wird, da eine Verfeinerung in der Folgephase stets günstiger ist.

Die Form des Entwurfs hängt von den verwendeten Darstellungsmitteln ab (vgl. Abschnitt 4.2.2.).

## Implementieren

**Implementieren ist eine Softwarephase, in der der Entwurf in ein lauffähiges Programm umgesetzt wird.**

Durch Implementieren wird der Feinentwurf in lauffähige Programmeinheiten überführt. Das geschieht durch Programmierung (vgl. Abschnitt 4.3.) oder den Einsatz von Standardwerkzeugen (vgl. Kapitel 5.) Das Rechnersystem, auf dem die Implementation erfolgt, wird als *Wirtsrechnersystem* bezeichnet. Die erzeugte Software ist in der Regel nicht nur für dieses Rechnersystem, sondern auch für viele andere bestimmt, die unter diesem Aspekt *Zielrechnersysteme* genannt werden.

Beim Implementieren werden alle Details festgelegt, die in vorherigen Entwicklungsphasen nur grob umrissen wurden. Dazu gehören die (programmtechnischen) Modulschnittstellen einschließlich der Form des Datenaustausches, die konkreten Datenstrukturen, die Auswahl günstiger Ablaufstrukturen und die konkreten Texte der Mensch-Maschine-Kommunikation.

Sind die zu implementierenden Programmeinheiten Hauptprogramme, so werden statt der tatsächlichen Moduln zunächst sogenannte leere Moduln (Dummys) eingebunden. Dummys ermöglichen die Kontrolle der Lauffähigkeit des Hauptprogramms.

Sind die zu implementierenden Programmeinheiten Unterprogramme, so werden kurze Rahmenprogramme geschrieben, mit denen die Lauffähigkeit der sonst nicht ausführbaren Programmeinheiten getestet werden kann. Sind Zugriffe auf Dateien erforderlich, so können leere Dateien (Stups) verwendet werden.

Implementieren ist in der praktischen Durchführung eine wiederholte Aufeinanderfolge von Editieren und Übersetzen.

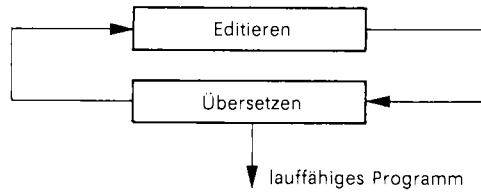


Abbildung 4.7.

Zyklus des Implementierens

**Editieren ist das Anlegen und Ändern einer Textdatei, deren Inhalt aus dem Quellcode eines Programms besteht.**

Die Quellcodedatei wird von einem Übersetzungsprogramm (vgl. Abschnitt 4.3.1.) gelesen und auf syntaktische Fehler geprüft. Werden Fehler erkannt, so werden diese mitgeteilt, sonst wird durch die Übersetzung das ausführbare Programm oder aus technischen Gründen eine Zwischenform erzeugt (vgl. Abbildung 4.7.).

Ist das Ergebnis eine Zwischenform (relativer Code), so ist ein weiterer Vorgang, das Verbinden und Laden (vgl. Abschnitt 4.3.1.), erforderlich, ehe das Programm tatsächlich ausgeführt werden kann.

Die Phase Implementieren endet entsprechend dem Fachbereichsstandard mit syntaktisch richtigen, aber inhaltlich nicht geprüften Programmeinheiten. Eine Qualitätskontrolle ist nicht möglich. In der praktischen Programmentwicklung werden deshalb Implementieren und Testen als Einheit betrachtet.

## Testen

**Testen ist eine Softwarephase, in der das (lauffähige) Softwareerzeugnis auf Übereinstimmung mit der spezifizierten Funktion geprüft wird.**

Der Test erfolgt zunächst getrennt für implementierte Hauptprogramme mit Dummys, Moduln mit Rahmenprogrammen un-

ter Verwendung von Testdaten (Einzeltest). Nach getesteter Funktionsfähigkeit aller Teile wird bei schrittweiser Integration der Programmeinheiten (Integrationstest) der Systemtest durchgeführt.

Als Ergebnis der Testphase entstehen das Programm/Programmsystem und die *Entwicklerdokumentation* (Programmdokumentation) entsprechend TGL 44 535<sup>3</sup>. Die Entwicklerdokumentation enthält alle Angaben, die für den Entwickler von Bedeutung sind, darunter verwendete Algorithmen, den Quellcode (Listings), die Beschreibung von Daten und Dateien, Ablaufprotokolle für Tests. Bei der Verwendung von Programmiersprachen (vgl. Abschnitt 4.3.1.) bildet der Quellcode den Kern der Entwicklerdokumentation.

### Fertigstellen

**Fertigstellen ist eine Softwarephase, in der das Softwareerzeugnis als Programmsystem und als Dokumentation bereitzustellen und im Anwendungsbereich unter realen Bedingungen zu erproben ist.**

In der Phase Fertigstellen ist vor allem die *Anwenderdokumentation*, die in den vorgelagerten Phasen in groben Konturen entworfen wurde, entsprechend dem Fachbereichsstandard 44 535 fertigzustellen. Für Anwendersoftware enthält sie:

- Programmkenndaten, darunter Deskriptoren zur Kurzbeschreibung von Funktionen, Anwendungsbereichen, Gerätebedarf, Betriebssystem, Programmgröße, Speicherplatzbedarf, verwendeter Programmiersprache und Übersetzer, Dateien und Dateioorganisation, Kopier- und Nutzungsrechten sowie der Organisation des Änderungsdienstes;
- Beschreibung der fachlichen Anwendung für den Nutzer (Nutzerhandbuch) einschließlich der Fehlermitteilungen und des Verhaltens bei Fehlern;

- Beschreibung des geräte- und programmtechnischen Betriebs sowie die Anpassung an unterschiedliche Bedingungen (Systemhandbuch) für die Systemverantwortlichen.

Ist die Anwendersoftware Teil eines komplexen Software-Anwendungssystems, so wird die Anwenderdokumentation Teil der Dokumentation des Software-Anwendungssystems.

Zur Dokumentation des Software-Anwendungssystems gehört die Beschreibung-

- der Programmbasis,
- der Datenbasis,
- des organisatorischen Systems und
- des technischen Systems.

Die Entwicklungsphasen Spezifizieren, Entwerfen, Implementieren, Testen und Fertigstellen gehen in der praktischen Umsetzung fließend ineinander über oder treten parallel zueinander auf. Oft werden sie auch aufgrund neuer Erkenntnisse wiederholt durchlaufen.

Die Entwicklung von Anwendersoftware und ihre Einführung werden in den zentralen und örtlichen Staatsorganen, in den Kombinat und Betrieben sowie in den anderen Einrichtungen der volkseigenen Wirtschaft auf der Grundlage gesetzlicher Bestimmungen<sup>4</sup> geplant, bilanziert und abgerechnet.

Die Planung und Abrechnung als wissenschaftlich-technische Aufgabe kann in der Nomenklatur der Arbeitsstufen und Lei-

3 Fachbereichsstandard TGL 44 535: Informationsverarbeitung / Dokumentation von Programmen, Programmsystemen und Softwareanwendungssystemen.

4 Anordnung vom 27. 1. 1989 über die Planung, Bilanzierung und Abrechnung von Software. In: GBl. I/1989 Nr. 6 S. 100.

5 Anordnung vom 28. 5. 1975 über die Nomenklatur der Arbeitsstufen und Leistungen von Aufgaben des Planes Wissenschaft und Technik. In: GBl. I/1975 Nr. 1 S. 426.

Anordnung Nr. 2 vom 18. 12. 1986 über die Nomenklatur der Arbeitsstufen und Leistungen von Aufgaben des Planes Wissenschaft und Technik. In: GBl. I/1987 Nr. 1 S. 7.



stungen des Planes Wissenschaft und Technik<sup>5</sup> erfolgen. Für die Entwicklung von Software gilt allgemein die E-Nomenklatur, im Rahmen spezieller Forschungsleistungen auch z. B. die A-Nomenklatur. Die Leistungsstufen der E-Nomenklatur und ihre Zuordnung zu den Softwarephasen sind in der Tabelle 4.1. zusammengestellt.

Die A-Nomenklatur unterscheidet 4 Entwicklungsstufen für Ergebnisse der angewandten Forschung. Die Stufe E 1 entspricht der Stufe A 1, E 2 und E 3 werden als Stufe A 2 umgesetzt, und die Stufen E 4 und E 5 entsprechen der Stufe A 3. Das Ergebnis der Stufe A 3 ist der experimentelle Nachweis der Prinziplösung und ihrer Reproduzierbarkeit. Die Entwicklung nach der A-Nomenklatur endet mit der Vorbereitung der Anwendung (A 4). Welche der Nomenklaturen zu benutzen ist, wird im Pflichtenheft festgelegt.

Die Neuentwicklung der Software ist nach den Stufen E 1 bis E 5 oder A 4 abgeschlossen. Die Arbeitsstufe E 6 bezieht sich auf die Nutzung der Software und umfaßt die Anwendungsphasen Einführen und Betreiben (vgl. Abbildung 4.5.7).

### Einführen und Betreiben

**Einführen ist eine Softwarephase, in der die (meist schrittweise) Integration der Softwareanwendung in den Anwendungsbereich erfolgt.**

Das Einführen verläuft nach einem Einführungsplan und endet, wenn ein stabiler Dauerbetrieb für das gesamte Erzeugnis erreicht ist. Nach Abschluß der Phase Einführen müssen die erreichten technisch-ökonomischen Parameter mit den Vorgaben verglichen werden.

**Betreiben ist eine Softwarephase, in der der stabile Dauerbetrieb des Softwareerzeugnisses erfolgt.**

Die Länge der Phase Betreiben ist die Nutzungsdauer der Anwendersoftware. Es wird der Nutzen erwirtschaftet (vgl. Abschnitt 1.3.2.). Die Analyse des Verhaltens der Software während der Anwendung erlaubt es, Funktionsabweichungen, Mängel und Ansätze für Verbesserungen zu erkennen.

Während des Betriebes erfolgt die Wartung.

### Wartung

**Wartung ist eine Softwarephase, in der Software modifiziert wird, ohne daß sich die Grundfunktion ändert.**

Im Durchschnitt entfallen 50 bis 70 % des Gesamtaufwandes für Anwendersoftware auf die Wartung. Bereits während des Entwurfs und beim Implementieren ist zu gewährleisten, daß der spätere Aufwand für

Tabelle 4.1.

E-Nomenklatur und Phasen (ergebnisorientiert)

Stufe	Bezeichnung	Phase
E 1	Pflichtenheft	Spezifikation
E 2	Projektkonzeption	Grobentwurf
E 3	Problemlösung	Feinentwurf
E 4	Programmierung und Test	Implementation/Test
E 5	Erprobung und Freigabe	Fertigstellung
E 6	Dauerbetrieb	Einführung/Betrieb/Wartung

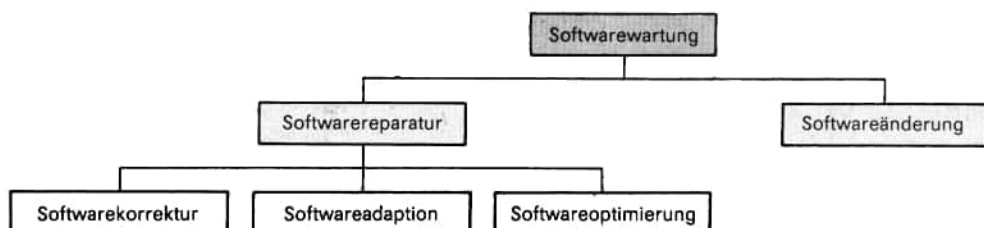


Abbildung 4.8.  
Klassifikation der Softwarewartung

die Wartung minimiert wird. Wartbarkeit ist ein Qualitätsmerkmal der Anwendersoftware (vgl. Abschnitt 4.1.). Die Klassifikation der Wartung geht aus Abbildung 4.8. hervor.

Kennzeichnend für die *Softwarereparatur* ist, daß sich der Funktionsumfang des Softwareerzeugnisses nicht ändert.

Korrigierende Wartung dient der Beseitigung von Fehlern, die erst während des Dauerbetriebes bemerkt werden. Fehler dieser Art sind keineswegs ungewöhnlich, da besonders komplexe Programmsysteme nur mit einem nichtvertretbaren Aufwand vollständig getestet werden könnten. Fehler dürfen aber nicht die Hauptfunktionen betreffen und nicht unter normalen Betriebsbedingungen auftreten.

Adaptierende Wartung paßt das Softwareerzeugnis veränderten Prozeß- und Anwendungsbedingungen, wechselnden Betriebssystemen oder wechselnder Hardware an. So ist zum Beispiel der Übergang vom Betriebssystem SCP zum Betriebssystem DCP für ein PASCAL-Programm mit einer Reihe von Änderungen verbunden. Einige Sprachelemente (Betriebssystemrufe, Dateioperationen) sind syntaktisch anzupassen, und die meist größeren Möglichkeiten des neuen Systems sind zu nutzen (z. B. erweiterter Dateischutz, Grafikelemente). Auch kann sich die Peripherie verändern (z. B. Festplatte) und zu einer veränderten Lösung führen.

Optimierende Wartung dient der Verbes-

serung der Qualitätsparameter eines Softwareerzeugnisses. Sie ergibt sich aus Forderungen des Anwendungsbereiches, z. B. nach kürzeren Reaktionszeiten, und durch wachsende Erfahrungen und softwaretechnologische Kenntnisse des Softwareentwicklers.

*Softwareänderung* führt zu einer Änderung der Funktionen des Softwareerzeugnisses. Es können auch weitere Funktionen hinzukommen. Im Unterschied zu einer Neuentwicklung bleibt jedoch der Anteil des geänderten Quellcodes unter 50 %.

Die unterschiedlichen Formen der Wartung treten oft kombiniert auf. Modifikationen werden durch Versionsnummern für den Nutzer der Software sichtbar gemacht. Entsprechend ändern sich auch die Dokumentationen. Die Freigabe modifizierter Versionen und die Übergabe an den Nutzer erfolgt über einen Änderungsdienst, dessen Modalitäten bereits in der Spezifikation festgelegt werden.

Inwieweit eine rationelle und effektive Softwareentwicklung erreicht wird, hängt entscheidend von den in diesem Prozeß eingesetzten Mitteln und Methoden ab.

### 4.2.2.

#### Prinzipien, Mittel und Methoden der Softwareentwicklung

Anwendersoftware ist wie alle Erzeugnisse in hoher Qualität und mit möglichst geringem Aufwand herzustellen. Großen Einfluß auf den Aufwand hat die Softwaretechnologie.

**Softwaretechnologie ist die Gesamtheit der Prinzipien, Mittel und Methoden, die für die Entwicklung eines Softwareerzeugnisses angewendet werden.**

#### Prinzipien der Softwareentwicklung

Prinzipien sind allgemeine Grundsätze, die in Theorie und Praxis der Softwareentwicklung durchgängig zu beachten sind und als Leitfaden dienen. Außer Qualitätssicherung und Aufwandsoptimierung zählen zu den Prinzipien der Softwareentwicklung:

1. Prinzip der Nachnutzung bereits entwickelter Software. Vor jeder Neuentwicklung ist zu prüfen, ob Software mit der erforderlichen Funktion als Programm oder Modul bereits existiert. In diesem Fall ist nachzunutzen. Bei Eigenentwicklungen ist möglichst die Nachnutzbarkeit zu gewährleisten.
2. Prinzip der Arbeitsteilung mit dem Ziel, die Entwicklungszeit für das Erzeugnis zu verkürzen. Bei Entwicklungszeiten von über zwei Jahren kann durch das hohe Tempo des wissenschaftlich-technischen Fortschritts eine solche Änderung gegenüber Spezifikation und Entwurf eintreten, die das gesamte Erzeugnis unbrauchbar macht.
3. Prinzip der Modularität. Die Modulen (Programmeinheiten) sind relativ abgeschlossen und austauschbar. Modularität ist Voraussetzung für sinnvolle Nachnutzung und Arbeitsteilung.
4. Prinzip der begleitenden Dokumentation und ständigen Aktualisierung der Entwicklungsunterlagen.

Unter Beachtung dieser Prinzipien werden Methoden der Softwareentwicklung eingesetzt.

#### Methoden der Softwareentwicklung

Eine Methode ist ein System detaillierter Regeln (Verfahren). Für die Softwareentwicklung haben sich zwei grundlegende (strategische) Methoden herausgebildet, die TOP-DOWN-Methode und die BOTTOM-UP-Methode.

**Die TOP-DOWN-Methode ist eine Methode der Softwareentwicklung, bei der vom Ganzen zum Teil (von der Spitze abwärts) fortgeschritten wird.**

Bei der TOP-DOWN-Methode werden zwei Arbeitsabschnitte unterschieden:

1. Das zu entwickelnde System wird als Ganzes dargestellt, ohne seine Elemente und die innere Struktur zu betrachten. Beschrieben werden die Eingabe- und Ausgabeinformation sowie die grundlegende Transformation der Eingabe in die Ausgabeinformation. Das System wird als virtuelle Maschine aufgefaßt, die die gestellte Aufgabe erfüllt.
2. Die Darstellung des Systems wird schrittweise durch Zerlegung in Elemente und Beschreibung der Struktur verfeinert.

Bei der Verfeinerung im zweiten Arbeitsabschnitt gelten folgende Regeln:

- Die Darstellung des Systems im Verfeinerungsschritt  $i+1$  ergibt sich durch Zerlegung der Darstellung des Schrittes  $i$  in Elemente. Jedes Element realisiert Eingabe und/oder Ausgabe von Information.
- Die bei jeder weiteren Verfeinerung definierten Elemente gliedern die Funktion des vorhergehenden Systems (oder Elements) vollständig auf. Es entsteht eine Baumstruktur (vgl. Abbildung 4.9.).
- Detailbeschreibungen der Elemente

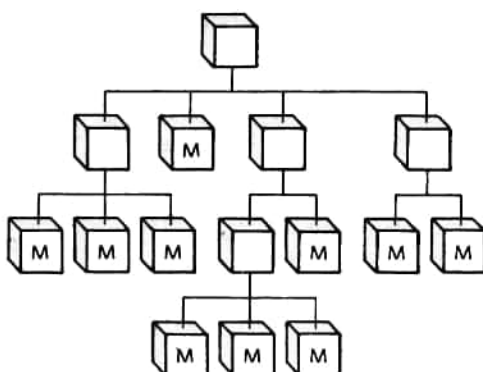


Abbildung 4.9.  
Baumstruktur mit Moduln (M)  
nach der Verfeinerung

hinsichtlich Ein- und Ausgabeinformation, Algorithmus der Transformation und Schnittstellen werden solange wie möglich aufgeschoben (vgl. Abschnitt 4.2.1.).

- Die Verfeinerung ist abgeschlossen, wenn jedes Element genau eine Funktion realisiert und alle Entscheidungen zur rechentechnischen Realisierung getroffen sind. Die letzte Verfeinerung hat damit stets die Form des Quelltextes eines lauffähigen Programms. Bei Verwendung höherer Programmiersprachen sollte ein Modul aus weniger als 50 Quelltextzeilen bestehen

Die Endstellen (Blätter) einer (vollständig verfeinerten) Baumstruktur entsprechen den Programmeinheiten (Moduln), die in ihrer Gesamtheit das Softwareerzeugnis bilden.

Die entgegengesetzte Vorgehensweise wird durch die BOTTOM-UP-Methode realisiert.

**Die BOTTOM-UP-Methode ist eine Methode der Softwareentwicklung, bei der vom Teil zum Ganzen (vom Fuße aufwärts) fortgeschritten wird.**

Es entstehen zunächst ausführbare Module, die anschließend zu einem System zusammengefügt werden. Alle für die Programmierung erforderlichen Entscheidungen sind stets sofort zu treffen.

Die BOTTOM-UP-Methode als Ganzes hat für die Softwareentwicklung wenig Bedeutung. Sie kann aber für kritische Module die TOP-DOWN-Methode ergänzen (HARDEST-FIRST-Test), um überhaupt die Realisierbarkeit eines Systems zu prüfen oder um sehr früh über getestete Module zu verfügen.

#### Technologische Mittel der Softwareentwicklung

Zur Softwareentwicklung werden technologische Mittel benötigt.

**Technologische Mittel der Softwareentwicklung sind Darstellungsmittel und/oder programmtechnische Mittel (Werkzeuge, Tools).**

Technologische Mittel dienen der rationalen Umsetzung einer Methode. Darstellungsmittel dienen zugleich der transparenten Entwicklerdokumentation.

Es gibt eine außerordentlich große Vielzahl von Darstellungsmitteln, besonders für den Entwurf, z. B. Pseudocode, Actigramm, Datagramm, Datenstrukturdiagramm, Datenflußplan, Programmablaufplan. Diese Darstellungsmittel werden in der Fachliteratur beschrieben. Viele haben sich in der Breite jedoch bisher nicht durchgesetzt. Die Tabelle 4.2. enthält den Teil der Darstellungsmittel, der als besonders wichtig für den Entwurf von Software für ökonomische Anwendungen angesehen wird und eine gewisse Verbreitung gefunden hat. Textprozessoren (Textverarbeitungssysteme) eignen sich, um die Textteile der Dokumentation zu bearbeiten.

Die bevorzugte Zuordnung der Darstel-

Tabelle 4.2.  
Softwaretechnologische Mittel für Phasen (Auswahl)

Entwicklungsphase	Definierte Darstellungsmittel	Programmtechnische Mittel
Analysieren	Pflichtenheftnachweis (Vordruck)	Textprozessor
Entwerfen	Strukturdiagramm, EVA-Tabelle, Struktogramm, Entscheidungstabelle	Entwurfssystem, Entwurfsgenerator, Textprozessor,
Implementieren	Programmiersprache	Programmiersystem, Texteditor, Assembler, Compiler, Interpreter, Linkage Editor, Lader
Testen	Entscheidungstabelle	Debugger, Tracer
Fertigstellung	Dokumentation TGL 44 535	Textprozessoren

lungsmittel zu den Bestandteilen des Entwurfs (vgl. Abschnitt 4.2.1.) ist folgende:

- Strukturdiagramm für Strukturübersichten,
- EVA-Tabelle zur Daten- und Funktionsbeschreibung auf der Ebene der Module,
- Struktogramm zur Beschreibung des Algorithmus der Arbeitsweise von Modulen,
- Entscheidungstabelle zur Beschreibung des Fehler- und des Dialogkonzepts.

### Strukturdiagramm

**Das Strukturdiagramm ist ein Darstellungsmittel für hierarchische Gliederungen. Die Elemente des Strukturdiagramms sind Funktionen oder Daten.**

Formen des Strukturdiagramms sind vor allem das Blockdiagramm und die Dezimalgliederung. Ein Blockdiagramm als Strukturübersicht über die Funktionen des Robotron-Computersystems für das Berliner Grand-Hotel zeigt Abbildung 4.10<sup>6</sup>.

Eine solche Strukturübersicht kann auch als dezimalgegliederte Liste dargestellt werden (vgl. Abbildung 4.11.).

Bei Strukturdiagrammen gilt, daß die Schnittmenge der durch die Elemente der gleichen Ebene realisierten Funktionen leer ist und in der Hierarchie folgende Elemente die Funktion vollständig auflösen. Damit ist das Strukturdiagramm ein geeignetes Arbeitsmittel zur schrittweisen Zerlegung eines Systems in Elemente.

Bei der Zerlegung wird in jeder Ebene in zwei Schritten vorgegangen.

- Die Gesamtfunktion wird in Einzelfunktionen zerlegt. Die Zerlegung ist am Problem orientiert. Es entstehen Problemmoduln. Nach der vollständigen Auflösung sollte ein Problemmodul genau eine Funktion realisieren.
- Aus den Problemmoduln werden in der gleichen Ebene Querschnittsaufgaben, Aufgaben, die in mehr als einem Pro-

<sup>6</sup> Manski, H.; Marschner, M.; Schenke, N.: robotron-Computersystem für das Berliner Grand-Hotel. In: Neue Technik im Büro, Berlin 31 (1987) 4, S. 98.

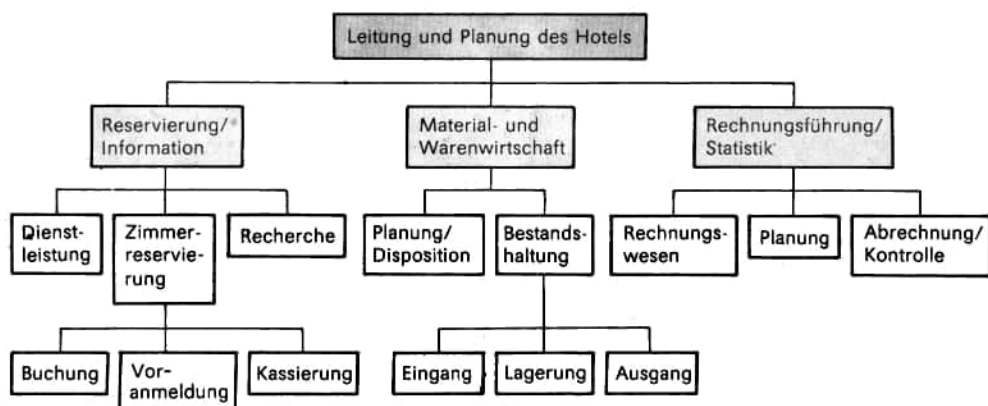


Abbildung 4.10.

Verfeinerungsstufe des Robotron-Computersystems für das Berliner Grand-Hotel

0.	Leitung und Planung des Hotels
1.	Reservierung/Information
1.1.	Dienstleistung
1.2.	Zimmerreservierung
1.2.1.	Buchung
1.2.2.	Voranmeldung
1.2.3.	Kassierung
1.3.	Recherche
2.	Material- und Warenwirtschaft
2.1.	Planung/Disposition
2.2.	Bestandshaltung
2.2.1.	Eingang
2.2.2.	Lagerung
2.2.3.	Ausgang
3.	Rechnungsführung/Statistik
3.1.	Rechnungswesen
3.2.	Planung
3.3.	Abrechnung/Kontrolle

Abbildung 4.11.

Dezimalgliederung  
einer Verfeinerungsstruktur

blemmodul zu lösen sind, ausgegliedert. Sie bilden Systemmoduln.

Die Systemmoduln sind in der Regel verfahrensorientiert oder realisieren rechen-technisch bedingte Abläufe (in den Abbildungen 4.10. und 4.11. wurden noch keine Systemmoduln ausgegliedert). Die Ausgliederung von Systemmoduln kann sich auf jeder Verfeinerungsebene wiederholen. Die Problemmoduln nutzen die Leistung der Systemmoduln. Dadurch wird Mehr-

fachprogrammierung gleicher Abläufe, die die Qualität eines Programmsystems vermindern würde, vermieden.

#### EVA-Tabelle

**Die EVA-Tabelle (Eingabe/Verarbeitung/Ausgabe-Tabelle) ist ein Darstellungsmit-  
tel zur Beschreibung eines Moduls hin-  
sichtlich Ein- und Ausgabeinformation  
(auch unter Verwendung von Datenfluß-  
symbolen) sowie der grundlegenden Ver-  
arbeitungsoperationen.**

Ein Beispiel für die EVA-Tabelle zeigt Abbildung 4.12.

In die Blöcke können Text oder Symbole für Datenträger (im Eingabe- oder Ausgabeblock) oder Verarbeitungsoperationen (im Verarbeitungsblock) eingetragen werden. EVA-Tabellen können zur Beschreibung jedes Elements der hierarchischen Struktur (Baumstruktur) benutzt werden, im allgemeinen werden sie jedoch nur für die Endstellen (Blätter) erarbeitet. In diesem Fall ist die EVA-Tabelle eine Verfeinerung des Strukturdiagramms auf unterster Ebene.

Bei der Anfertigung einer EVA-Tabelle wird mit der Beschreibung der Ausgabein-

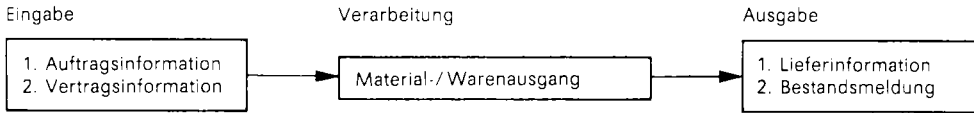


Abbildung 4.12.  
EVA-Tabelle zum Modul 2.2.3./Ausgang (Auszug)

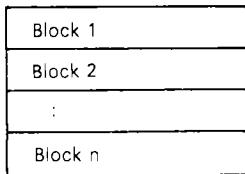


Abbildung 4.13.  
Darstellung einer Sequenz

formation begonnen und danach die erforderliche Eingabeinformation ermittelt. Es genügt die Angabe von Informationskomplexen. Außerdem wird die Verarbeitung als Transformation der Eingabe- in die Ausgabeinformation beschrieben.

Die EVA-Tabelle beschreibt die Kopplungsstruktur für die Moduln und ist die Grundlage für die Datenstruktur, die jedoch noch von den Möglichkeiten der verwendeten Programmiersprache beeinflusst wird. Im allgemeinen genügt auch die Beschreibung der Verarbeitung noch nicht für die programmtechnische Umsetzung. Die EVA-Tabelle kann durch ein Struktogramm weiter verfeinert werden.

### Struktogramm

**Das Struktogramm (Nassi-Shneiderman-Diagramm) ist ein grafisches Darstellungsmittel zur strukturorientierten Beschreibung des grundlegenden Algorithmus der Verarbeitung von Information.**

In Struktogrammen wird zwischen Sequenz, Selektion und Iteration von Blöcken unterschieden. Blöcke beschreiben Operationen.

**Eine Sequenz ist eine Aufeinanderfolge von zwei oder mehreren Blöcken, die unbedingt und in der notierten Reihenfolge auszuführen sind.**

Die Darstellung der Sequenz im Struktogramm erfolgt durch eine einfache Reihung der Blöcke (vgl. Abbildung 4.13.). Jeder Block kann in sich strukturiert sein, also selbst wieder Strukturkomponenten besitzen. Die Eintragungen in die Blöcke erfolgen im allgemeinen in Textform.

**Eine Selektion ist die Auswahl eines auszuführenden Blocks durch eine Bedingungsprüfung (Alternative) oder durch den Wert einer Variablen (Fallunterscheidung).**

Im Struktogramm wird die Alternative durch Notieren der Bedingung, ihrer möglichen Werte (TRUE, FALSE oder JA, NEIN) und den zugeordneten Block dargestellt (vgl. Abbildung 4.14.). Wurde für den Wert einer Bedingung kein Block notiert (unvollständige Alternative), so erfolgt keine Operation.

Die Darstellung der Fallunterscheidung zeigt Abbildung 4.15.

Auch hier können Blöcke selbst strukturiert sein, also z. B. eine Sequenz oder eine Selektion enthalten.

**Eine Iteration (Schleife) ist die Wiederholung eines Blocks in Abhängigkeit von einer Bedingungsprüfung vor (geschlossene, abweisende Iteration) oder nach Ausführung des Blocks (offene Iteration).**

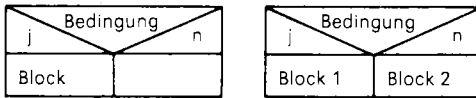


Abbildung 4.14.  
Darstellung einer Alternative

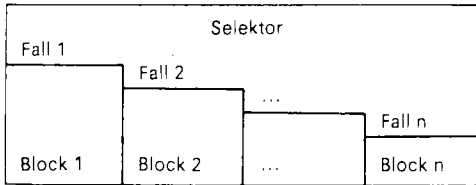


Abbildung 4.15.  
Darstellung einer Fallunterscheidung

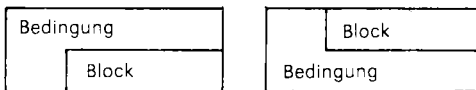


Abbildung 4.16.  
Darstellung einer geschlossenen und einer offenen Iteration

Die Darstellung der Iteration im Struktogramm enthält die Abbildung 4.16.

Bei der geschlossenen Iteration wird die Ausführung des Blocks wiederholt, solange die Bedingung erfüllt ist. Erst wird die Bedingung geprüft, dann erfolgt die Ausführung. Ist die Bedingung beim Eintritt in diese Strukturkomponente nicht erfüllt, wird der Block nicht ausgeführt.

Bei der offenen Iteration wird die Ausführung des Blocks wiederholt, bis die Bedingung erfüllt ist. Erst wird der Block ausgeführt, dann die Bedingung geprüft. Der Block in einer offenen Iteration wird also mindestens einmal ausgeführt.

Ein Beispiel für die Anwendung der Struktogramntechnik zeigt die Abbildung 4.17.

Beim Einsatz von Struktogrammen ist darauf zu achten, daß die Detailliertheit deutlich geringer ist als bei Ausdrucksmitteln der Programmiersprache (Statements). Der mittlere Auflösungsfaktor für eine Struktur-

komponente sollte mehr als 7 Statements betragen. Sonst werden Entscheidungen, die besser mit der Programmiersprache ausgedrückt werden können, vorweggenommen.

Grundsätzlich kann ein Struktogramm in allen Verfeinerungsebenen eingesetzt und selbst verfeinert werden. Verbreitet erfolgt der Einsatz jedoch für die Endstellen (Blätter) der vorher genügend verfeinerten Baumstruktur.

### Entscheidungstabelle

**Die Entscheidungstabelle ist ein Darstellungsmittel zur Beschreibung der Zuordnung zwischen möglichen Situationen und erforderlichen Maßnahmen.**

Ein Beispiel für die Anwendung einer Entscheidungstabelle zum Entwurf eines Fehlerkonzeptes zeigt Abbildung 4.18. Die Entscheidungstabelle besteht aus vier Quadranten, die hier mit römischen Ziffern bezeichnet werden. Im Quadranten I werden Bedingungen notiert. Dieser Quadrant besteht nur aus einer Spalte. Im Quadranten II werden mögliche Werte der Bedingungen notiert:

- J (JA, YES, TRUE) für Bedingung erfüllt
- N (NEIN; NO, FALSE) für Bedingung nicht erfüllt
- (INDIFFERENT, I) für Erfüllung der Bedingung indifferent, nicht signifikant.

»J«, »N« oder »–« werden in derselben Zeile eingetragen, in der die Bedingung notiert wurde. Die entstehenden Spalten charakterisieren jeweils eine *Situation*. Bei  $n$  Bedingungen gibt es  $2^n$  verschiedene Situationen. Es müssen jedoch nicht alle möglichen Zusammenstellungen notiert werden. Es genügt, solche Situationen aufzunehmen, die sich hinsichtlich der erforderlichen Maßnahmen (Reaktion) unter-



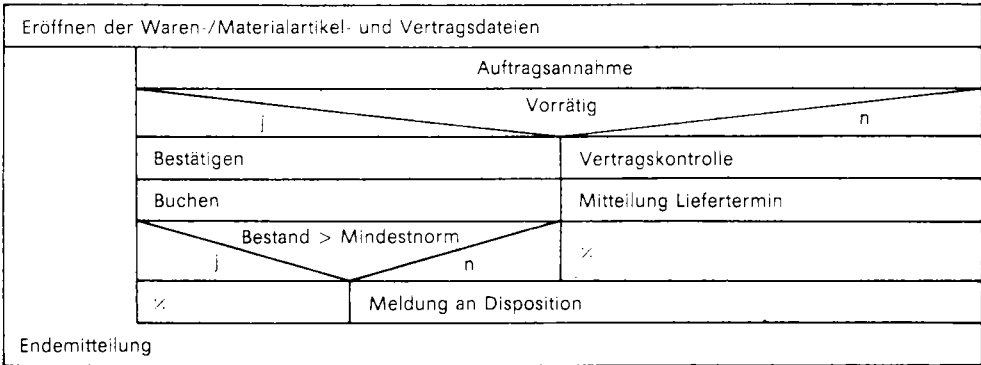


Abbildung 4.17.  
Struktogramm zum Modul 2.2.3. (vereinfacht)

Bestandwert < 0	J	N	N	N
Eingabewert nicht plausibel	—	J	N	N
(I) Mindestnorm = 0	—	—	J	N (II)
kein Vertrag vorhanden	—	—	—	J
Fehlermitteilung	X	X	X	
Warnung				X
(III) Korrektur erzwingen		X		(IV)
Pause bis Tastendruck			X	
Abbruch	X			

Abbildung 4.18.  
Entscheidungstabelle für ein Fehlerkonzept  
des Moduls 2.2.3./Ausgang

scheiden. Diese Maßnahmen werden im Quadranten III beschrieben und im Quadranten IV durch das Zeichen »X« den Situationen zugeordnet. Im Beispiel (Abbildung 4.18.) werden der Situation [J — — —] (»Bestand < 0«, unabhängig von den anderen Bedingungen) die Maßnahmen »Fehlermitteilung« und »Abbruch« zugeordnet. Die Zuordnung zwischen Situation und Maßnahme heißt *Regel*. Die Spalten aus den Quadranten II und IV beschreiben also Regeln.

Die für die Softwareentwicklung wichtige Vollständigkeit der Situationen kann geprüft werden. Die Anzahl der möglichen Zusammenstellungen ist  $2^n$ , im Beispiel  $2^4 = 16$ . Sie reduziert sich, wenn

- x Elemente einer Situation indifferent sind, um  $2^x - 1$ . Zum Beispiel steht in

der Spalte 1 des Quadranten II die notierte Situation [J — — —] für folgende Zusammenstellungen:

J J J J J J J J  
N N N N J J J J  
N N J J N N J J  
N J N J N J N J

Das sind insgesamt  $2^3 = 8$  (allgemein  $2^x$ ) Situationen, von denen mit [J — — —] nur eine wirklich im Quadranten II notiert ist. In Spalte 2 des Quadranten II sind 2 Elemente indifferent, so daß die Spalte zusätzlich 3 mögliche Situationen ( $2^2 - 1 = 3$ ) erfaßt. Entsprechend steht Spalte 3 für eine weitere Situation ( $2^1 - 1 = 1$ ).

- Situationen vorhanden sind, zu denen keine Maßnahmen gehören. Im Beispiel ist das die Situation [N N N N]. In diesem Fall liegt kein Fehler vor, so daß die Fehlerkonzeption keine Maßnahme vorsehen muß.

Die Beschreibung der Regeln ist vollständig, wenn die folgende Bedingung erfüllt ist:

$$2^n - P - \sum_{i=1}^m (2^x - 1)_i = M$$

n

Anzahl der in Quadrant I notierten Bedingungen

P

Anzahl der Situationen ohne Maßnahme

- m Anzahl der Spalten in Quadrant II
- x Anzahl der Indifferenzzeichen in der Spalte i
- M Anzahl der Maßnahmen (Zeilen in den Quadranten III und IV).

Im Beispiel ist die Beschreibung wegen

$$2^4 - 1 - ((2^3 - 1) + (2^2 - 1) + (2^1 - 1) + (2^0 - 1)) = 4$$

vollständig.

Mit diesen Überlegungen können auch tri-viale Regeln erkannt und aus dem Entwurf entfernt werden.

Die Möglichkeiten zur Darstellung in Entscheidungstabellen lassen sich erweitern, wenn im Quadranten I eine Variable notiert wird und im Quadranten II ihre möglichen Werte (oder Intervalle). In diesem Fall handelt es sich um eine erweiterte Entscheidungstabelle.<sup>7</sup>

### Entwurfssysteme

Entwurfssysteme sind aufeinander abgestimmte Methoden und Darstellungsmittel über alle Verfeinerungsstufen. Sie unterscheiden sich nach ihrer konzeptionellen Basis in datenorientierte und funktionsorientierte Entwurfssysteme.

Bei *datenorientierten Entwurfssystemen* werden Datenstrukturen, Datentypen oder Datenflüsse zum Ausgangspunkt genommen. Ein datenorientiertes Entwurfssystem geht z. B. von abstrakten Datentypen aus. Logische Datenstrukturen werden algebraisch spezifiziert, indem von ihrer physischen Repräsentation abstrahiert wird. Es geht dabei um die Definition des äußeren Verhaltens einer Klasse von Datenobjekten, der ein abstrakter Datentyp entspricht. Dieses Verhalten wird mit einer Menge von Operationen oder Funktionen beschrieben, die über Exemplare der Klasse erlaubt und erklärt sind. Wie jedes Exemplar repräsentiert ist und wie die Operationen intern wirken, bleibt dem Benutzer verborgen. Ein abstrakter Datentyp wird

als *Datenkapsel* bezeichnet. Die Datenkapsel verkörpert ein Modul, dessen Datenstruktur vor unkontrolliertem Zugriff geschützt ist. Änderungen dieser Struktur betreffen nur die Datenkapsel selbst. Auf die zugreifenden Moduln wirkt sich die Änderung nicht aus. Damit fördert der Entwurf abstrakter Datentypen und ihre Implementation als Datenkapsel eine konsequente Modularisierung der Software.

Bei *funktionsorientierten* Entwurfssystemen wird in den Verfeinerungsebenen jeweils von der Funktion ausgegangen, und die Überlegungen zu Daten werden nachträglich ausgeführt. Eine Übersicht über Entwurfssysteme vermittelt Tabelle 4.3.

Die Entwurfssysteme als Ganzes haben sich bisher nicht durchgesetzt. Es werden Teile als Regeln und Darstellungsmittel zu »eigenen« Entwurfsmethoden kombiniert. Wichtig ist, daß die Entwurfsmethoden innerhalb einer softwareentwickelnden Einrichtung einheitlich sind. Sie sollten deshalb in Arbeitsanweisungen festgelegt sein.

### Programmtechnische Mittel der Softwareentwicklung

Zur Unterstützung der Softwareentwicklung werden in zunehmendem Maße Rechnersysteme eingesetzt. Software mit dieser Funktion heißt Entwicklungssoftware, die Softwareerzeugnisse heißen *Werkzeuge* (Tools). Besonders ausgeprägt sind Werkzeuge für die Implementierungsphase. Zur Unterstützung des Entwurfs existieren noch keine so ausgereiften Werkzeuge wie für die Implementierung.

Besonders effektiv sind Werkzeuge für verschiedene Phasen, wenn ihnen ein einheitliches Konzept zugrunde liegt. In diesem Fall wird von einem *Softwareentwicklungss-*

<sup>7</sup> Freitag, G.; Gode, W.; Jacobi, H.; Lautz, H.; Simon, J.; Spittel, U.: Einführung in die Entscheidungstabellentechnik. Berlin: VEB Verlag Technik 1976, S. 11.

Tabelle 4.3.  
Entwurfssysteme (Auswahl)<sup>8</sup>

Bezeichnung	Einsatzgebiete	Orientierung	Darstellungsmittel
HIPO (Hierarchy plus Input-Process- Output)	kommerzielle Informations- verarbeitung	Funktion	Strukturdiagramm, EVA-Tabelle
JDM (Jackson Design Method)	kommerzielle Stapelverarbeitung	Datenstruktur	Strukturdiagramm, Pseudocode
SDM. (Structured Design Method)	kommerzielle Dialogverarbeitung	Datenfluß	Strukturdiagramm, Pseudocode, Tabellen
ARS (Allgemeine Rekursive Strukturierung)	Prozeßsteuerung	Datenfluß	Petri-Netze, Entscheidungs- tabellen

system (vgl. Abschnitt 3.1.1.) gesprochen. Ein Softwareentwicklungssystem ist dadurch gekennzeichnet, daß mehrere Entwicklungsphasen durch Werkzeuge unterstützt werden, die über eine Ergebnisweitertege untereinander verknüpft sind und eine einheitliche Nutzerschnittstelle haben. Beispiele für Softwareentwicklungssysteme sind TESYS und PSU des VEB Kombinat Robotron.

### 4.2.3. Nutzermitwirkung und Prototyping

Für die Qualität der Software sowie für die Effektivität ihres Einsatzes ist die Einbeziehung der künftigen Nutzer in den Entwicklungsprozeß unerläßlich. Die Nutzermitwirkung muß dazu beitragen, daß die Software

- den Erfordernissen des Basissystems entspricht,
- die nutzerorientierten Qualitätsmerkmale einhält und
- vom Nutzer akzeptiert wird.

Der Grad der Mitwirkung des Nutzers ist in den einzelnen Phasen des Lebenszyklus verschieden. Die Mitwirkung ist besonders

ausgeprägt beim Analysieren, Entwerfen, Fertigstellen und während der Anwendung. Inhaltliche Schwerpunkte der Nutzermitwirkung in den Softwarephasen sind:

- |               |  |
|---------------|--|
| Analysieren   | Bestimmung der Hauptfunktionen und Abgrenzung des Systems von der Umgebung sowie die Kalkulation von Aufwand und Nutzen. |
| Entwerfen     | Bewertung des fachlich-logischen Entwurfs hinsichtlich Grundstruktur und Vollständigkeit.                                |
| Fertigstellen | Erprobung der Funktionen, der Betriebssicherheit sowie Beurteilung der Anwenderdokumentation.                            |
| Anwendung     | Einführung, Aufdeckung von Unzulänglichkeiten beim Betreiben und Veranlassung von Wartungsmaßnahmen.                     |

Formen der Mitwirkung des Nutzers sind Zuarbeiten und Stellungnahmen zu vorgelegten Ergebnissen, Konsultationen, Über-

<sup>8</sup> Vgl. Rothhardt, G.: Praxis der Softwareentwicklung. Berlin: VEB Verlag Technik 1987, S. 172.

prüfung von Ergebnissen, z. B. mit Checklisten, Teilnahme an Verteidigungen oder direkte Einbeziehung in das Entwicklerkollektiv. Besonders günstig für die Zusammenarbeit zwischen Entwickler und Nutzer ist das Prototyping.

**Prototyping ist Softwareentwicklung mit frühzeitiger Demonstration durch ein lauffähiges Programm (Prototyp), das als unvollständiges Produkt bewertet und im allgemeinen geändert und weiterentwickelt wird.**

Der Prototyp dient vor allem der Kommunikation mit dem Nutzer. Die Schwierigkeit der praktischen Zusammenarbeit zwischen Nutzer und Entwickler bei der Softwareentwicklung besteht darin, daß der Entwickler die fachliche Spezifik des Anwendungsbereiches nicht genügend kennt und der Nutzer die mögliche Rechnerleistung im Detail nicht einschätzen kann. Zum Teil gibt es auch Schwierigkeiten in der Wahl der Begriffe für die Verständigung. Prototyping kann diese Schwierigkeiten am konkreten Objekt beseitigen.

Für den Entwickler ist Prototyping zugleich ein frühzeitiges Mittel zur Überprüfung auf rechentechnische Realisierbarkeit.

Es gibt verschiedene Ansätze des Prototyping, die sich in der praktischen Anwendung vermischen und ergänzen:

- *exploratives* (erforschendes) Prototyping, bei dem nicht ein Lösungsvorschlag im Mittelpunkt steht, sondern (ausgehend von Hauptkriterien) verschiedene Varianten gezeigt werden;
- *experimentelles* Prototyping zur Überprüfung einer konzipierten Lösung durch den Nutzer;
- *evolutionäres* Prototyping, bei dem der Prototyp zugleich ein für den Nutzer handhabbares Erzeugnis und Teil des späteren Endprodukts ist.

Den grundlegenden Zyklus des Prototyping in den Entwicklungsphasen Analysieren

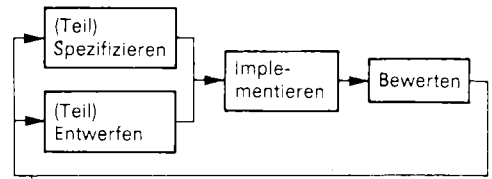


Abbildung 4.19.

Zyklus des Prototyping

(mit dem Schwerpunkt Spezifizieren) und Entwerfen zeigt die Abbildung 4.19.

Durch *Prototyping beim Analysieren/Spezifizieren* kann in einem sehr frühen Stadium bereits die Nutzerschnittstelle für die Hauptfunktionen vorgeführt werden. Der Entwickler hat in der modernen Softwareentwicklung dem Nutzer nach der Analyse des Basisprozesses die Hauptfunktionen des künftigen Softwareerzeugnisses vorzuschlagen. Prototyping erleichtert diese Aufgabe erheblich. Der Nutzer kann sachkundig mitwirken, z. B. bei der Festlegung der Funktionen, bei der Auswahl der Dialogtechnik, dem Bildschirmaufbau und der Drucklistengestaltung. Spezifikationsfehler, deren Behebung in späteren Phasen besonders aufwendig ist, werden weitgehend vermieden. Der Prototyp ersetzt nicht die Spezifikation (vgl. Abschnitt 4.2.1.), sondern ergänzt sie. Prototyping in der Spezifikationsphase ist Spezifizieren am Beispiel.

*Prototyping in der Entwurfsphase* kann mit Blick auf die Baumstruktur eines Entwurfs horizontales Prototyping über eine Verfeinerungsstufe sein oder als vertikales Prototyping einen Zweig der Baumstruktur erfassen. Häufig vermischen sich beide Formen. Das Ziel ist wie beim Prototyping in der Spezifikationsphase die Beteiligung des künftigen Nutzers mit seinen Fachkenntnissen an der Softwareentwicklung. Der Prototyp simuliert in dieser Phase im allgemeinen zunächst die spezifizierte Benutzeroberfläche und fortschreitend auch die Arbeitsweise der Moduln.

Grundsätzlich ist es möglich, den Prototyp nach der Bewertung durch den Nutzer entweder wegzuerwerfen und die Erkenntnisse in dem gesondert entwickelten Softwareerzeugnis zu berücksichtigen (rapid prototyping, quick and dirty) oder zu ändern und zum Enderzeugnis weiterzuentwickeln. Die Weiterverwendung ist anzustreben. In diesem Fall entfallen Implementieren und Testen in traditioneller Form, weil sie bereits (in Schritten) vorweggenommen wurden. Das Prototyping durchbricht die Entwicklungsphasen nicht grundsätzlich, die Veränderungen sind aber bei der Planung und Abrechnung zu berücksichtigen.

Für Prototyping wurden spezielle Werkzeuge entwickelt. Dazu gehören insbesondere Generatoren für Bildschirmmasken und Listengeneratoren. Mit ihnen können der Bildschirm- oder Listenaufbau rationell entworfen und gezeigt werden.

Prototyping wirkt auch motivierend auf den Softwareentwickler, da frühzeitig mit dem Rechnersystem gearbeitet wird und sichtbare Ergebnisse vorliegen.

### 4.3. Programmiersprachen

#### 4.3.1. Programme und Programmiersprachen

##### Maschinenbefehl

Alle in einem Rechnersystem ablaufenden Operationen richten sich nach Maschinenbefehlen.

**Ein Maschinenbefehl ist ein Auftrag an das Steuerwerk, eine elementare Operation auszuführen.**

Maschinenbefehle werden beim Implementieren erzeugt. Ein Maschinenbefehl hat die in Abbildung 4.20. dargestellten Teile.

Der *Operationscode* ist ein Bitmuster aus dem Befehlssatz eines Prozessors. Wird das Bitmuster dem Prozessor als Befehlsinformation übermittelt, also nicht als numerische oder nichtnumerische Information (vgl. Abschnitt 1.2.2.), so reagiert der Prozessor mit einer Operation auf den Operanden. Es kann sich z. B. um Transportoperationen innerhalb eines Computers, arithmetische und logische Operationen oder Ein- und Ausgabeoperationen handeln. Die Gesamtheit der definierten Operationscodes bildet den *Befehlssatz* des Prozessors. Er ist im Mikroprogrammspeicher abgelegt und wird bei der Befehlsausführung aktiviert. Unterschiedliche Prozessoren haben stets auch einen unterschiedlichen Befehlssatz.

Der *Operand* ist entweder Verarbeitungsinformation, die selbst Gegenstand der Operation ist, oder Adreßinformation. Adreßinformation zeigt direkt oder indirekt auf Verarbeitungsinformation, die für die eigentliche Operation erst bereitgestellt werden muß, oder auf Operationscode, der als nächstes zur Ausführung gelangt.

##### Maschinensprache

Ein Befehl umfaßt je nach Art der Operation ein oder mehrere Byte. Die Gesamtheit der Befehle bildet das Maschinenprogramm.

**Ein Maschinenprogramm (Objektprogramm) ist eine geordnete Folge von Maschinenbefehlen, die nacheinander oder in gesteuerter Reihenfolge ausgeführt werden.**

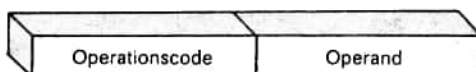


Abbildung 4.20.  
Grundbestandteile  
eines Maschinenbefehls

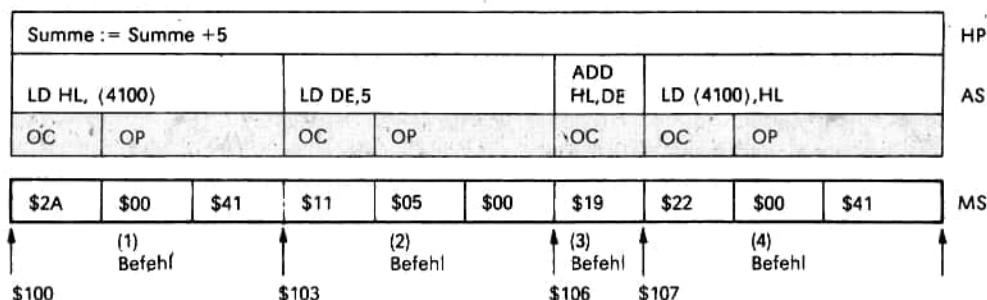


Abbildung 4.21.

Programm in Maschinensprache (MS), Assemblersprache (AS) und höherer Programmiersprache (HP), OC-Operationscode, OP-Operand

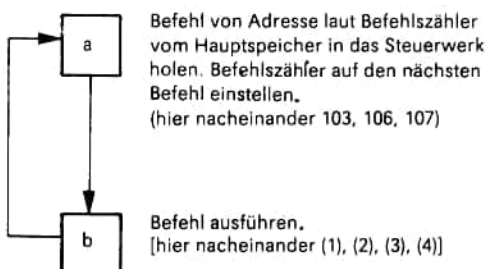
Ein Prozessor und damit ein Rechnersystem kann immer nur Maschinenprogramme (Programme in Maschinencode) ausführen. Wird ein Programm in einer solchen elementaren Form geschrieben, so heißt diese Form *Maschinensprache*. Abbildung 4.21. zeigt ein Programm in Maschinensprache (MS), das einen auf der Adresse \$4100 (\$ kennzeichnet die hexadezimale Schreibweise) befindlichen Wert um 5 erhöht. Dazu werden der Wert und die Zahl 5 in Register (HL, DE) geladen, die Registerinhalte addiert und der neue Wert (steht durch die Befehlskonstruktion im HL-Register) auf die Adresse zurückgeschrieben. Das Programm ist ab Adresse \$100 gespeichert.

Ein Programm in Maschinensprache kann (bytwweise) in den für Befehlscode vorgesehenen Teil des Hauptspeichers eingegeben werden, und die Ausführung kann veranlaßt (gestartet) werden. Bei der Ausführung wird ein spezielles Register, der Befehlszähler (Program Counter, PC), benutzt. Der Befehlszähler enthält am Anfang die Adresse des ersten Befehls (hier \$100) und wird nach jeder Befehlsausführung auf die Adresse des nächsten Befehls eingestellt (inkrementiert). Der Grundzyklus für diese Arbeitsweise ist folgender:

1. Nach einem Startkommando (von außen) wird das Maschinenprogramm in

den Hauptspeicher an eine vom Betriebssystem festgelegte Adresse (z. B. \$100) transportiert, und der Befehlszähler wird auf die Anfangsadresse eingestellt.

2. Das Maschinenprogramm wird zyklisch wie folgt ausgeführt:



Nach der Ausführung des Maschinenprogramms kehrt der Prozessor in den Ausgangszustand zurück und wartet auf ein neues Startkommando.

Ist das Programm in einer anderen Sprache als in der Maschinensprache geschrieben, so muß es vor der Ausführung in ein Maschinenprogramm überführt werden.

**Ein Programm, das nicht in der Maschinensprache geschrieben ist, heißt Quellprogramm (Programm im Quellcode).**

Die Bezeichnung Quellprogramm wird mit der Bezeichnung der jeweiligen Program-

miersprache verbunden, z. B. Assembler-Quellprogramm, PASCAL-Quellprogramm.

### Programmiersprachen

Die Implementierung mit einer Maschinensprache würde darauf hinauslaufen, ausschließlich hexadezimale Zahlen hintereinander zu schreiben und in den Computer zu bringen. Das wäre aufwendig (mittlere Programme benötigen mehrere Millionen solcher Zahlen) und sehr fehleranfällig. Zur Rationalisierung dieses Vorganges wurden Programmiersprachen entwickelt.

**Eine Programmiersprache ist ein Hilfsmittel zur Notation von Befehlsfolgen in einer der Arbeitsweise des Menschen angepaßten Form.**

Es werden die in Abbildung 4.22. dargestellten grundlegenden Arten von Programmiersprachen unterschieden.

**Eine maschinenorientierte Programmiersprache (Assemblersprache, niedere Programmiersprache) ist eine Programmiersprache, die vom Befehlssatz des Rechnersystems bestimmt wird und an den Befehlssatz des Rechnersystems gebunden (maschinenabhängig) ist.**

In Abbildung 4.21. (AS) sind mit gleicher Wirkung wie in der Maschinensprache folgende Elemente einer Assemblersprache notiert:

```
LD HL,<4100>
LD DE,5
ADD HL,DE
LD <4100>,HL
```

Eine Assemblersprache unterscheidet sich von der Maschinensprache durch:

- Verwendung von Bezeichnungen für den Operationscode, die zugleich auf deren Wirkung hinweisen (mnemotech-



Abbildung 4.22.  
Grobklassifikation von  
Programmiersprachen

nischer Operationscode). Im Beispiel steht für \$19 der Maschinensprache das Symbol ADD für Addition.

- Verwendung symbolischer Adressen. Im Beispiel wäre es bei entsprechender Organisation möglich, »LD HL,Summe« zu schreiben, wenn der Adresse \$4100 die symbolische Bezeichnung »Summe« zugeordnet ist. Das erleichtert den Umgang mit Speicherplatz erheblich, da im allgemeinen umfangreiche Verarbeitungsinformation zu manipulieren ist.
- Möglichkeit zum Einbringen bereits früher programmierter Befehlsfolgen (Makros), so daß viele Abläufe nicht stets neu erarbeitet werden müssen.

Kennzeichnend ist, daß jeder Assemblerbefehl genau einem Maschinenbefehl entspricht. Ein in der Assemblersprache geschriebenes Programm wird durch einen Assembler übersetzt.

**Ein Assembler ist ein Programm, das ein in maschinenorientierter Sprache geschriebenes Programm (Quellprogramm) in ein Maschinenprogramm (Objektprogramm) mit verschieblichen Adressen (relokativer Code, Objektcode) übersetzt.**

Maschinencode mit verschieblichen Adressen ist erforderlich, weil die künftige Stelle, die das Programm im Hauptspeicher einnimmt, zum Zeitpunkt der Bearbeitung mit dem Assembler im allgemeinen noch nicht bekannt ist.

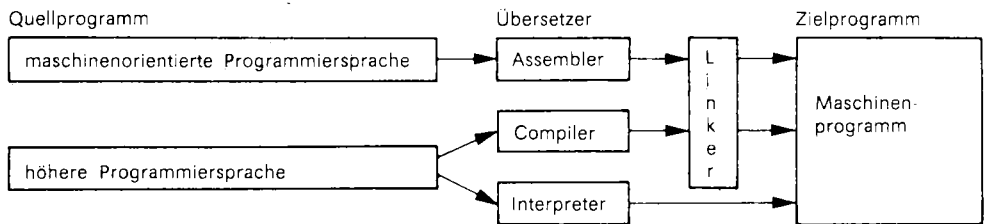


Abbildung 4.23.

Übersetzung von Quellprogrammen verschiedener Programmiersprachen in ein Maschinprogramm

Tabelle 4.4.

Grobe Übersicht über Vor- und Nachteile von Programmiersprachen

Sprache	Speicher/ Laufzeitfaktor	Programmier- aufwand	Portabilität
Maschinenorientierte Programmier-sprache	1	6–10	0
Höhere (prozedurale) Programmier-sprache	2–4	1	0,95

Die Überführung von Maschinenprogrammen mit verschieblichen Adressen in ausführbare Maschinenprogramme mit realen Adressen erfolgt durch ein gesondertes Programm, den Linker.

**Ein Linker (linkage editor) ist ein Programm, das relocativen Code unter Verwendung von Maschinencodebibliotheken in ausführbare Maschinenprogramme umwandelt.**

Linkvorgänge sind auch bei der Programmierung mit höheren Programmiersprachen erforderlich.

**Eine höhere Programmiersprache ist eine Programmiersprache, die vom Befehlssatz eines Prozessors weitgehend unabhängig (maschinenunabhängig) ist und eine kompakte, leichtverständliche Notation ermöglicht.**

In Abbildung 4.21. (HP) ist mit gleicher Wirkung wie in Maschinen- und Assemblersprache in einer höheren Programmiersprache (PASCAL, PL/1, ALGOL) folgendes programmiert:

Summe: = Summe + 5

Ein solcher (komprimierter) Befehl in einer höheren Programmiersprache heißt **Instruktion** oder **Statement**. Die Notation ist einfacher als in Maschinen- oder Assemblersprache. Sie ähnelt der mathematischen Schreibweise.

Quelltexte höherer Programmiersprachen werden durch Compiler oder Interpreter übersetzt.

**Ein Compiler ist ein Programm, das ein in einer höheren Programmiersprache geschriebenes Programm als Ganzes in Maschinencode mit verschieblichen Adressen übersetzt.**



Die Weiterbearbeitung erfolgt wie beim Einsatz eines Assemblers durch einen Linker. Es gibt auch Compiler mit integriertem Linker. Kennzeichnend ist, daß ein Programm als Ganzes in ein ausführbares Programm übersetzt und eine Kommandodatei erzeugt wird. Erst dann erfolgt die (wiederholte) Ausführung.

Interpreter erzeugen keine Kommandodateien.

**Ein Interpreter ist ein Programm, das ein in einer höheren Programmiersprache geschriebenes Programm schrittweise (Statement für Statement) in ausführbaren Maschinencode übersetzt und diesen sofort ausführt.**

Der Interpreter muß im Hauptspeicher anwesend sein, wenn das in einer höheren Programmiersprache notierte Programm ausgeführt wird, der Compiler nicht. Durch die ständige, auch bei wiederholter Ausführung immer wieder notwendige Übersetzung ist die Laufzeit von Programmen, die durch einen Interpreter ausgeführt werden, erheblich länger. Dafür sind aber Änderungen des Programms sehr einfach.

Die verschiedenen Wege zu einem ausführbaren Programm zeigt die Abbildung 4.23.

Den Vorteilen, die höhere Programmiersprachen für die Programmierung mit sich bringen, stehen Nachteile in Laufzeit- und Speichereffizienz sowie in der Portabilität gegenüber (Tabelle 4.4.).

Zu beachten ist, daß Assemblersprachen alle Ressourcen des jeweiligen Rechnersystems erreichen. Deshalb und wegen der aus der Tabelle 4.4. ersichtlichen Vorteile bieten höhere Programmiersprachen die Möglichkeit, mit speziellen Statements Assemblersprachelemente oder Maschinencode zu benutzen. Grundsätzlich kann damit stets in einer höheren Sprache pro-

grammiert werden. Für lauffzeit- oder speicherplatzkritische Operationen (im allgemeinen weniger als 5 %) sind Assemblersprachelemente einzubinden.

#### 4.3.2.

#### Konzepte höherer Programmiersprachen

Die Klassifikation höherer Programmiersprachen geht aus Abbildung 4.24. hervor.

#### Prozedurale Sprachen

**Eine prozedurale Programmiersprache ist eine höhere Programmiersprache, die sich bei definierten Datenstrukturen an einem vorgegebenen Algorithmus orientiert.**

Ein Algorithmus ist eine Verfahrensvorschrift, die Inhalt und Aufeinanderfolge von Operationen zur Lösung einer Aufgabe festlegt. Der Algorithmus hat drei grundlegende Anforderungen zu erfüllen:<sup>9</sup>

1. Für jede Situation, die zu irgendeinem Zeitpunkt eintreten kann, gibt es eine detaillierte Vorschrift über die nächste Operation.
2. Die Vorschrift muß wenigstens einen Parameter enthalten, dessen Belegung wechseln kann (im einfachsten Fall also z. B. nicht »5 \* 6«, weil hier das Ergebnis feststeht, sondern »A \* 6«).

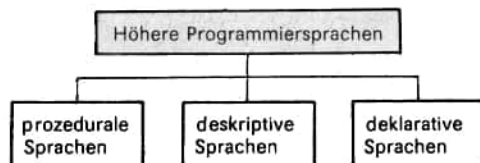


Abbildung 4.24.  
Klassifikation von höheren Programmiersprachen

<sup>9</sup> Kämmerer, W.: Einführung in mathematische Methoden der Kybernetik. Berlin: Akademie-Verlag 1971, S. 413.

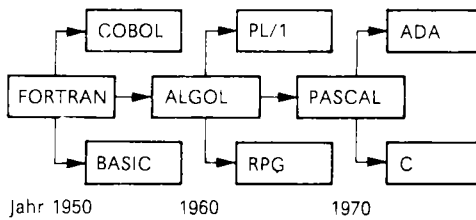


Abbildung 4.25.

Entwicklung wichtiger prozeduraler Programmiersprachen (vereinfacht, die Jahreszahlen beziehen sich vor allem auf die Hauptlinie FORTRAN-ALGOL-PASCAL)

3. Die Vorschrift muß nach einer endlichen Anzahl von Schritten zu einer eindeutig resultierenden Information führen.

Prozedurale Programmiersprachen setzen einen Algorithmus in eine für Compiler oder Interpreter verständliche Form um. Diese erzeugen ausführbare Programme, die den Algorithmus auf dem Rechnersystem technisch realisieren. Prozedurale Sprachen werden seit den 50er Jahren entwickelt. Abbildung 4.25. gibt eine Übersicht über verbreitete prozedurale Sprachen.

Diese Programmiersprachen werden nicht nacheinander abgelöst, sondern weiterentwickelt. Dabei werden Programmiersprachen entwickelt für

- ökonomische Probleme durch besondere Unterstützung der Behandlung umfangreicher Datenmengen,
- technische Probleme durch besondere Unterstützung mathematischer Operationen.

Programmierersprachen gelten als universell, wenn sie beides ermöglichen. Außerdem werden auch Programmiersprachen entwickelt für spezielle Aufgaben, wie RPG (Report Program Generator) für die Programmierung der Listenverarbeitung.

Sprachen, die z. B. für technische Probleme entwickelt wurden, können auch zur Lösung ökonomischer Probleme eingesetzt werden. Überhaupt lassen sich unterschiedliche Probleme mit verschiedenen Programmiersprachen lösen. Das ist anzutreffen, wenn die erforderlichen programmtechnischen Mittel nicht zur Verfügung stehen oder Sprachkenntnisse fehlen. Der Aufwand für die Programmierung und die Effizienz der Programme ist jedoch erheblich ungünstiger, wenn Programmiersprachen verwendet werden, die nicht für die Problemlösung geeignet sind.

*FORTRAN* (FORmula TRANslation) wurde bereits 1957 als Programmiersprache eingeführt. Sie ist besonders zur Lösung technischer Probleme geeignet. Für die breite Anwendung ist ausschlaggebend, daß im Laufe der Jahre umfangreiche FORTRAN-Bibliotheken geschaffen wurden, viele technische Anwendungen in FORTRAN programmiert sind und die Sprachkenntnisse weit verbreitet sind. Vorteilhaft ist

- die Möglichkeit, für Teilaufgaben separate Unterprogramme zu schaffen, gesondert zu übersetzen und später in andere Programme einzubinden.
- die Unterstützung iterativer Ein- und Ausgaben von Daten.

Das Datentypkonzept von FORTRAN unterstützt keine komplexen Datenstrukturen und nicht die Behandlung von Zeichenketten. Unter softwaretechnologischem Gesichtspunkt ist nachteilig, daß kein Blockkonzept mit genau definierten Gültigkeitsbereichen von Variablen und Unterprogrammen existiert und nur Iterationen mit bekannter Häufigkeit unterstützt werden. FORTRAN ist die bevorzugte Programmiersprache der technischen Bereiche und steht in der DDR für alle Rechnersysteme zur Verfügung.

**BASIC** (Beginners All-Purpose Symbolic Instruction Code) wurde durch Vereinfachung von FORTRAN Anfang der 60er Jahre mit dem Ziel leichter Erlernbarkeit geschaffen. In der Regel erfolgt die Übersetzung durch einen Interpreter, der zugleich eine einfache Programmierumgebung schafft. Die Vor- und Nachteile sind bis auf die wenig entwickelte Unterprogrammtechnik die gleichen wie bei FORTRAN. Allerdings sind BASIC-Programme weniger effizient als FORTRAN-Programme. BASIC wird besonders auf Computern eingesetzt, deren Leistungsparameter unter denen der Personalcomputer liegen. Ein praktisches Problem besteht darin, daß es viele BASIC-Versionen gibt, die oft erhebliche Unterschiede aufweisen und an das Rechnersystem gebunden sind.

**COBOL** (COmmon Business Oriented Language) wurde Ende der 50er Jahre geschaffen. Kennzeichnend ist:

- die Unterstützung von Datentypen des kommerziellen Bereichs, wie sie z. B. in der Lohn-, Grundmittel- oder Finanzbuchhaltung erforderlich sind,
- die gute Lesbarkeit durch ausführliche Notation unter Verwendung der (natürlichen) englischen Sprache,
- die (begrenzte) Möglichkeit der Anpassung eines Programms an den jeweiligen Zielcomputer durch ein Maschinenteil (environment deviation), zusätzlich zum Erkennungs-, Daten- und Prozedurteil.

Die Nachteile der Sprache liegen im fehlenden Blockkonzept und der aufwendigen Notation. Außerdem werden numerische Operationen unzureichend unterstützt.

**ALGOL** (ALGOL 60, Algorithmic Language) brachte 1960 erhebliche Fortschritte hinsichtlich der Elemente und Strukturen von Programmiersprachen und der Definition ihrer Syntax.

Zur Syntaxdefinition wurde die Backus-Naur-Form eingeführt.

**Die Backus-Naur-Form (BNF, auch EBNF für Extended BNF) ist eine Sprache (Metasprache) zur syntaktischen Beschreibung von Programmier- und Kommandosprachen.**

Die Backus-Naur-Form benutzt Terminalsymbole, Nichtterminalsymbole und einige Sonderzeichen (vgl. Anhang A).

ALGOL 60 war die erste Programmiersprache mit einem Blockkonzept für die Gültigkeit von Variablen und Unterprogrammen. Die Sprache ist wissenschaftlich-technisch orientiert. In der praktischen Programmierung hat sie keine große Bedeutung erlangt. Sie wurde vor allem Veröffentlichungssprache für mathematische Algorithmen und Grundlage für die Programmiersprachen ALGOL 68 und PL/1.

**PL/1** (Programming Language 1) wurde etwa 1964 entwickelt. Sie nutzt die Erfahrungen, die mit FORTRAN und COBOL auf der theoretischen Basis von ALGOL gesammelt wurden, und zeichnet sich aus durch:

- Eignung für ökonomische und technische Probleme,
- Möglichkeiten (begrenzt) einer maschinennahen Programmierung,
- relative Großzügigkeit für Notationen, wenn die Absicht des Programmierers eindeutig erkennbar ist.

Die Nachteile dieser sehr leistungsfähigen Sprache liegen in ihrem Umfang und in der Effizienz der durch PL/1-Compiler erzeugten Programme.

PL/1 ist die in der DDR auf Großrechenanlagen am meisten genutzte Programmiersprache.

**PASCAL** (nach dem französischen Mathematiker und Philosophen Blaise Pascal,

1623–1662) entstand um 1970 als einfache, auf gute Strukturierung und Übersichtlichkeit der Programme gerichtete Sprache für universelle Aufgaben. Kennzeichnend sind:

- ein leistungsfähiges Datentypkonzept für einfache und strukturierte Variablen sowie für den Umgang mit Adressen,
- das Blockkonzept mit globalen oder in ihrer Gültigkeit begrenzten (lokalen) Variablen und Unterprogrammen,
- leistungsfähige Kontrollstrukturen für die Iteration, die es gestatten, Blöcke mit nur einem Eintritts- und einem Austrittspunkt zu programmieren und so die Modularität des Programms zu gewährleisten. Es wird möglich, auf fehleranfällige GOTO-Statements zu verzichten.

Nachteilig sind relativ starre Grenzen zwischen den Datentypen und die fehlende Möglichkeit, separate Unterprogramme zu handhaben. Diese Nachteile wurden durch die Weiterentwicklung unter dem Namen MODULA (MODULA 2) beseitigt. Inzwischen ist diese Weiterentwicklung auch in den PASCAL-Systemen für Personalcomputer und Kleindatenverarbeitungsanlagen realisiert.

Die Programmiersprache C wurde Anfang der 70er Jahre für universelle Anwendungen entwickelt. Sie hat Ähnlichkeit mit PASCAL, hat aber vor allem ein anderes Datentypkonzept. Kennzeichnend sind:

- Datentypen, die den Objekten entsprechen, mit denen das Rechnersystem selbst umgeht; das sind Zeichen, Zahlen und Adressen;
- leistungsfähige Kontrollstrukturen mit nur einem Eingang und einem Ausgang;
- gering gehaltener Sprachumfang, der durch eine nicht zur Sprache gehörende Funktionsbibliothek umfassend ergänzt wird.

Die Beschränkungen der Sprache verringern den Aufwand für Compiler und ermöglichen die Erzeugung von Objektprogrammen sehr hoher Effizienz. Dadurch kann im allgemeinen auf Assemblerprogrammierung völlig verzichtet werden.

Nachteilig ist die fehleranfällige Syntax. C steht in der DDR für Personalcomputer und Kleindatenverarbeitungsanlagen zur Verfügung.

ADA (nach Augusta Ada Byron, 1815–1852, Mitarbeiterin des englischen Mathematikers und Konstrukteurs von Rechenanlagen, Charles Babbage, 1792–1871) ist ebenfalls eine PASCAL-ähnliche Programmiersprache. Sie wird seit 1983 vornehmlich auf Großrechenanlagen eingesetzt.

ADA realisiert einen ähnlich universellen Anspruch wie PL/1. Sie unterstützt dabei besonders parallele Prozesse in Rechnersystemen mit mehreren Zentralprozessoren. Der Sprachumfang ist außerordentlich umfangreich und wird von vielen Compilern nicht voll realisiert.

Neben den genannten prozeduralen Programmiersprachen gibt es eine Reihe weiterer Sprachen für spezielle Probleme, beispielsweise der Simulation oder der Planung.

### Deskriptive Sprachen

**Eine deskriptive Programmiersprache ist eine höhere Programmiersprache, die vorwiegend auf vorgefertigte, komplexe Funktionen zurückgreift und zur Lösung einer speziellen Klasse von Aufgaben geeignet ist.**

Deskriptive Sprachen entstanden aus der Notwendigkeit, die Entwicklung von Anwendersoftware erheblich zu beschleunigen. Dabei traten Fragen der Ausnutzung

von Ressourcen des Rechnersystems und der Qualität der Anwendersoftware zurück.

Kennzeichnend für eine deskriptive Sprache ist, daß nicht die Beschreibung des Algorithmus im Vordergrund steht, sondern die Beschreibung der Aufgabe, des Problems.

Ein einfaches Beispiel ist die Bildung der Summe eines vorher definierten Datenfeldes UMSATZ mit  $n$  Komponenten. In PASCAL wird programmiert:

```
Summe: = 0;
FOR i: = 1 TO n DO
Summe: = Summe + Umsatz[i]
```

Das entsprechende Statement in einer deskriptiven Sprache kann lauten:

SUM UMSATZ TO SUMME

Der Algorithmus der Summierung wird nicht beschrieben, sondern ist vorgefertigt.

Allgemein können folgende Merkmale deskriptiver Sprachen genannt werden:

- Es gibt viele vordefinierte Funktionen, z. B. für die Ein- und Ausgabe, für Datenkonvertierungen, Drucklistenaufbereitung, Dateioperationen (Sortieren, Mischen).
- Neben Interpretern oder Compilern existieren eine Reihe weiterer, unterstützender (externer) Komponenten, wie Bildschirmmasken, Dialogabläufe.
- Die Sprache ist mit einem definierten System zur Datenspeicherung (Datenbanksystem) verbunden, selbst aber unabhängig von diesem.
- Die Sprachen sind dialogorientiert, d. h. in interaktiver Umgebung einsetzbar.

Es haben sich zwei Richtungen für den Einsatz deskriptiver Sprachen herausgebildet. Die eine dient dem Programmierer zur Rationalisierung bestimmter Tätigkeiten, besonders der Datenbankarbeit, wenn dazu noch keine Programmbausteine zur Verfü-

gung stehen. Die andere gibt Endnutzern einfache Instrumente zur individuellen, arbeitsplatzbezogenen Programmierung in die Hand.

Mit deskriptiven Sprachen lassen sich für den nicht geübten Programmierer Zeitgewinne von 5:1 oder mehr bei der Programmentwicklung erzielen. Die Speicherplatz- und Laufzeiteffizienz verringert sich dabei allerdings um 100 % oder mehr. Spezielle Ressourcen des Rechnersystems können im allgemeinen nicht erreicht werden. Das beeinträchtigt die Qualität der Lösung zum Teil erheblich. Sie fällt jedoch nicht ins Gewicht, wenn Entwickler und Nutzer dieselbe Person sind.

Deskriptive Sprachen stehen in der DDR vor allem für Personalcomputer als Teil der Softwareerzeugnisse REDABAS, MULTICOMP, ARIADNE (vgl. Kapitel 5.) und als Datenbanksprachen auf Großrechenanlagen zur Verfügung.

### Deklarative Sprachen

**Deklarative Programmiersprachen sind höhere Programmiersprachen, die es gestatten, Wissen in Form von Fakten und Regeln zu deklarieren, Fragen zu formulieren und ihre Beantwortung zu veranlassen.**

Der Algorithmus wird vom Programmierer nicht festgelegt und auch nicht deskriptiv aktiviert. Grundlage der internen Arbeitsweise des Rechnersystems ist die Prädikatenlogik.

Diese Programmiersprachen entwickelten sich neben funktions- und objektorientierten Programmiersprachen bei den Bemühungen, Software zu entwickeln, die Elemente menschlicher Denkfähigkeit auf einem Rechnersystem realisiert. Das Arbeitsgebiet der Informatik, das sich mit der Nachbildung von Intelligenzleistungen des Menschen auf der Maschine befaßt, heißt

*künstliche Intelligenz* (KI; Artificial Intelligence, AI).

Auf entsprechend leistungsfähigen Rechnersystemen dienen deklarative Programmiersprachen unter anderem folgenden Zielen:

- Wissen von Experten durch ein Rechnersystem zu speichern und vielen Nutzern (Nichtexperten) zugänglich zu machen (Expertensysteme);
- Lehr- und Lernsysteme für die Aus- und Weiterbildung zu entwickeln und dem Menschen am Ort seiner beruflichen Tätigkeit nutzbar zu machen;
- Kommunikation des Menschen mit dem Rechnersystem in der für den Menschen gewohnten Form, also akustisch unter Verwendung der natürlichen Sprache oder visuell, zu realisieren;
- Schranken verschiedener Landessprachen in Wort und Schrift unter Nutzung von Computern zu überwinden.

Natürlich können solche Aufgaben auch mit prozeduralen Programmiersprachen gelöst werden. Der Aufwand ist entsprechend hoch. Der Grundgedanke von Programmiersprachen liegt aber darin, einen möglichst großen Teil der Lösung dem Computer zu überlassen.

Zu den deklarativen Programmiersprachen gehört PROLOG (PROgramming in LOGic). PROLOG wurde bereits Anfang der 70er Jahre entwickelt, hat aber erst seit Anfang der 80er Jahre praktische Bedeutung erlangt.

Die Programmierung in PROLOG besteht in folgenden Grundschritten:

1. Deklarieren von Fakten über Objekte.  
Die Fakten werden durch Klauseln repräsentiert. Eine Klausel besteht aus einem Prädikat, das eine Beziehung ausdrückt und Argumente besitzt, z. B.:  
  
mittel (fortran, technisch, mittel, effizient)  
mittel (cobol, oekonomisch, mittel, effizient)  
mittel (pascal, universell, leicht, effizient)

2. Deklarieren von Regeln. Die Regeln werden durch Klauseln dargestellt, die aus zwei Teilen bestehen, der Folgerung und der Prämisse. Die Folgerung wird durch ein Faktum dargestellt. Die Prämisse setzt sich aus mehreren durch logische Operatoren verbundenen Fakten zusammen. Die Notation

```
sprache(pascal,X,leicht,effizient)
if mittel (_,X,_,_)
```

kennzeichnet, daß die universelle Sprache PASCAL für ökonomische und technische Anwendungen geeignet ist.

3. Formulierung von Fragen (Zielen). Die Anfrage ist in der einfachsten Form wie ein Faktum aufgebaut. Ein Beispiel ist:

```
sprache(X,oekonomisch,leicht,effizient)
```

Mit dieser Notation wird eine Programmiersprache gesucht, die für ökonomische Aufgaben anwendbar, leicht erlernbar und effizient ist. X ist eine Variable, die durch Suchen und Verknüpfen (Backtracking) unter Verwendung gespeicherter Fakten und Regeln einen Wert erhält. Im Beispiel antwortet das System mit:

X = pascal

### Sprachgenerationen

Ähnlich wie bei der Hardware wird auch bei Programmiersprachen der Begriff Generation verwendet. Die erste Sprachgeneration bezeichnet Maschinensprachen, die zweite Sprachgeneration Assemblersprachen, die dritte Sprachgeneration höhere prozedurale Programmiersprachen, die vierte Sprachgeneration höhere deskriptive Sprachen und die fünfte Sprachgeneration Programmiersprachen, die für Anwendungen der künstlichen Intelligenz benutzt werden.

Zu den Sprachen der fünften Generation gehört neben PROLOG auch die funktionsorientierte Sprache LISP (LISt Processing

language). Sprachen der fünften Sprachgeneration werden besonders mit Rechnersystemen der fünften Rechnergeneration in Verbindung gebracht.

Die (unscharfe) Einteilung in Sprachgenerationen assoziiert, daß die Sprachen nacheinander abgelöst werden. Das ist keineswegs so. Hochleistungsfähige Basissoftware, darunter z. B. Compiler für PROLOG, werden in Assembler geschrieben. Effiziente Programme in höheren Programmiersprachen enthalten in der Regel ebenfalls Assemblerteile.

Prozedurale, deklarative und deskriptive Sprachen unterscheiden sich hinsichtlich ihres Einsatzbereiches und ihres Nutzerkreises, so daß sie nicht gegenseitig ersetzbar sind.

#### 4.3.3. Programmierungsumgebung

Die Eignung einer Programmiersprache für die Softwareentwicklung hängt nicht nur von der definierten Sprache als solche ab.

Wichtige Einflußfaktoren sind die Qualität des Compilers und die Qualität des Linkers. Zwar sind die wichtigsten Programmiersprachen durch ISO-Standard (International Standard Organization) genormt, doch weichen alle implementierten Compiler mehr oder weniger von diesem Standard ab, um in hohem Maße die Ressourcen einer Rechnerklasse zu nutzen. Durch die Implementation des Compilers ergeben sich bereits erhebliche Leistungsunterschiede.

Für die Programmierung sind außer Compiler und Linker weitere programmtechnische Mittel erforderlich. Solche Mittel sind der Editor zur Bearbeitung des Quelltextes, Druckprogramme (Lister) für Quelltexte, Crossreferenzen (Variablenverzeichnisse) und Programmstrukturen, Debugger und Tracer zur Fehlersuche, Dienst- und Hilfs-

programme für die Arbeit mit Dateien, Werkzeuge zur Unterstützung des Programmmentwurfs und der Dokumentation, Bibliotheken und Boxen. Diese Programme bilden die Programmierungsumgebung.

**Die Programmierungsumgebung (programming environment) ist die Gesamtheit programmtechnischer Mittel zur Unterstützung der Programmierung.**

Sind alle programmtechnischen Mittel aufeinander abgestimmt und verbunden, wird die Programmierungsumgebung als integrierte Umgebung bezeichnet.

Die Programmierungsumgebung beeinflusst erheblich die Produktivität der Softwareentwicklung und wird deshalb bei der Auswahl einer Sprache mit gleichem Gewicht berücksichtigt wie die Sprache selbst.

Eine weitgehend integrierte Programmierungsumgebung ist auch in den PASCAL-Programmiersystemen des VEB Kombinat Robotron (z. B. PASCAL 880/886, PASCAL PLUS) für Personalcomputer realisiert.

Jedes dieser Systeme besteht aus Systemkern und Systemservice. Nach dem Start des Systemkerns, der aus Compiler, Editor und Directory-Anzeiger besteht, werden dem Nutzer die in Abbildung 4.26. dargestellten Funktionen angeboten.

Der Aufruf der Funktionen erfolgt über die intensiv dargestellten Kommandobuchstaben L (Laufwerk wechseln), V (Verzeichnis wechseln), A (Arbeitsdatei festlegen und laden), H (Hauptdatei festlegen bei Verarbeitung mehrerer Quelltextmoduln), E (Editor starten und Inhalt anzeigen), C (Compiler mit eingestellten Optionen starten), O (Optionen des Compilers über ein zweites Menü ändern), T (Übersetzung des Programms, Ablage des Codes im Hauptspeicher und Ausführung des übersetzten Programms), S (Sicherung der editierten Datei auf Diskette) D (Inhalt des aktuellen Verzeichnisses anzeigen) und B (Programm beenden und Rückkehr zu DCP).

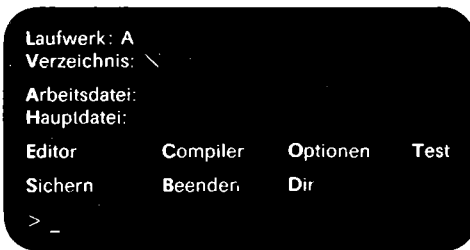


Abbildung 4.26.  
Kerngrundmenü PASCAL 886

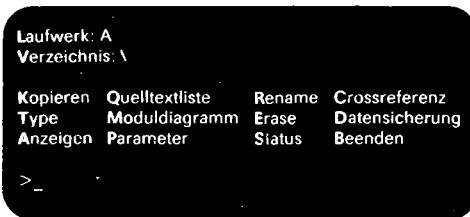


Abbildung 4.27.  
Servicegrundmenü PASCAL 886

Der Systemservice bietet die in Abbildung 4.27. erkennbaren Dienste an. Mit den Kommandobuchstaben L und V können wie im Kerngrundmenü Laufwerk und Verzeichnis gewechselt werden. B (Programm beenden) führt in das Betriebssystem zurück. Mit P (Parameter) können Blattformate eingestellt, Festlegungen über die automatische Trennung überlanger Zeilen getroffen sowie Kopier- und Listfunktionen für unbefugte Nutzer gesperrt werden.

Die anderen Kommandobuchstaben führen zu einer alphabetischen Anzeige aller Dateinamen im aktuellen Laufwerk/Verzeichnis. Nach Verwendung des Kommandobuchstabens A (Anzeige) werden zusätzlich die Größe der einzelnen Dateien, die Diskettenkapazität sowie der freie und belegte Platz auf dem Datenträger angezeigt. Werden die Kommandobuchstaben K (Dateien kopieren), Q (Quelltext des Programmes drucken), R (Umbenennung von Dateien), C (Crossreferenzliste drucken oder

anzeigen), T (Anzeige von Dateien und Hauptspeicherbereichen in hexadezimaler Darstellung und versuchter Zeichendarstellung), M (Anzeige des Strukturdiagramms der Moduln), E (Dateien löschen) und S (Read-Only-Status setzen oder rücksetzen)\* verwendet, so ist eine Auswahl der angezeigten Dateien über Nummern, Namen oder Gruppennamen erforderlich. Die Funktionen werden dann für die ausgewählten Dateien ausgeführt. Vor der Ausgabe von Quelltext- und Crossreferenzlisten werden dem Nutzer Optionen angeboten, die zugleich die Leistungsfähigkeit der Programmierumgebung zeigen.

Für die Ausgabe von Quelltexten können die aus Abbildung 4.28. ersichtlichen Optionen gewählt werden.

Die von der Programmiersprache PASCAL reservierten Wörter werden durch Großschreibung und Fettdruck markiert. Das Einrücken erfolgt entsprechend der logischen Schachtelungstiefe im PASCAL-Programm. Die entstehenden Listen dienen sowohl der Fehlersuche (z. B. durch die paarweise Numerierung von zusammengehörenden BEGIN/END-Klammern) als auch der Dokumentation. Mit K kann Ergänzungsinformation (z. B. der Name des Programmierers, das Datum) auf jedes Blatt gedruckt werden. Nach P pausiert der Drucker nach jedem Blatt.

Wird die Funktion C (Crossreferenzliste drucken) gewählt, so stehen die in Abbildung 4.29. dargestellten weiteren Möglichkeiten zur Auswahl.

K und P wirken wie die Listoptionen im Menü der Abbildung 4.28. Mit den anderen Optionen (alle können beliebig kombiniert werden) wird folgendes erreicht:

- A Die Crossreferenzliste wird nicht gedruckt, sondern auf dem Bildschirm angezeigt.
- E Es wird zur Eingabe von Bezeichnern aufgefordert. Die Crossreferenzliste wird nur für diese Variable gedruckt



**Listenformatierung:**

Zeilen numerieren  
 Absätze zusammenhalten  
 Reservierte Worte markieren  
 Schachtelungstiefe anzeigen  
 Einrücken  
 Inklusive Includedateien / laufende Numerierung  
 Mit Includedateien / gesonderte Numerierung  
 Kopfzeile ergänzen  
 BEGIN/END paarweise numerieren  
 Quelldateiname unterdrücken  
 Pause nach jeder Seite

Auswahl (+ oder (ET) für ZREB): \_

Abbildung 4.28.  
Menü der Quelltextformatierung  
unter PASCAL 886

**Optionen für Crossreferenzliste:**

Kopfzeile ergänzen  
 Anzeige über Bildschirm  
 Einzelne Bezeichner auswählen  
 Definition / Deklaration mit ":" markieren  
 Wertzuweisung mit "+" kennzeichnen  
 Pause nach jeder Seite

Auswahl oder (ET): \_

Abbildung 4.29.  
Menü der Crossreferenzoptionen  
unter PASCAL 886

oder angezeigt. Die Anzahl der Variablen ist unbegrenzt.

D In der Crossreferenzliste wird hinter der Zeilennummer das Zeichen »:« geschrieben, wenn der Bezeichner in dieser Zeile eingeführt (definiert oder deklariert) oder zur Definition anderer Bezeichner benutzt wird.

W In der Crossreferenzliste wird hinter der Zeilennummer das Zeichen »+« vermerkt, wenn der Bezeichner in dieser Zeile durch eine Ergibtanweisung seinen Wert verändert.

Zusammen mit einem sehr schnellen Compiler, einer direkten Verbindung von Compiler und Editor, so daß bei einem Syntax-

fehler der Editor gestartet, der Quelltext angezeigt und der Cursor hinter den Fehler gesetzt wird, ergibt sich für die Programmierung in PASCAL eine einfache und leistungsfähige Programmierumgebung.

Ähnlich leistungsfähige Programmierumgebungen für Personalcomputer gibt es z. B. auch für die Sprachen C, PROLOG und BASIC.

**Übungen**

1. Ordnen Sie den Begriff Individualsoftware in die Softwareklassifikation ein, und betrachten Sie ökonomische Gesichtspunkte des Verhältnisses Individualsoftware und Standardsoftware bei der Bereitstellung von Anwendersoftware!
2. Erläutern Sie an Beispielen, warum die Qualitätssicherung für Anwendersoftware besonders in den frühen Phasen der Softwareentwicklung erfolgen muß!
3. Welche Forderungen stellen Sie an die Nutzerschnittstelle eines dialogorientierten Anwenderprogramms?
4. Schildern Sie Inhalt und Aufeinanderfolge aller Phasen des Softwarelebenszyklus bei der Entwicklung von Anwendersoftware mit Prototyping!
5. Fertigen Sie ein Struktogramm des folgenden (Euklidischen) Algorithmus zur Bestimmung des größten gemeinschaftlichen Teilers zweier natürlicher Zahlen an!
  - a) Belege A und B mit den beiden natürlichen Zahlen! Führe b) aus!
  - b) Prüfe  $A < B$ ! Ist die Bedingung erfüllt, führe c), sonst d) aus!
  - c) Vertausche die Belegungen von A und B! Führe d) aus!
  - d) Prüfe die Bedingung  $B = 0$ ! Ist die Bedingung erfüllt, führe f) aus, sonst e)!

- e) Bestimme den Rest, der sich aus der Division A durch B ergibt! Belege A mit dem Wert von B und B mit dem Rest der Division! Führe d) aus!
  - f) Lies den Wert von A! Es ist der größte gemeinschaftliche Teiler. Beende die Ausführung des Algorithmus!
6. Erläutern Sie die Nachteile des Struktogramms für den Entwurf eines Programms zur Ermittlung des größten gemeinschaftlichen Teilers, und schlagen Sie eine andere Darstellungsmöglichkeit vor!
6. Entwerfen Sie den Mensch-Maschine-Dialog (einschließlich Fehlermitteilungen) für den rechnergestützten Verkauf von Möbeln im Einzelhandel unter Verwendung einer Entscheidungstabelle!

## 5. Rechnereinsatz für Standardaufgaben

### 5.1. Einsatzgebiete

Für die Nutzung von Rechnersystemen im Anwendungsbereich haben im wesentlichen zwei Arten transienter Software unmittelbare Bedeutung:

- Anwendersoftware für Aufgaben in komplexen Automatisierungslösungen,
- Standardwerkzeuge für autonome Aufgaben kleineren Umfangs am Arbeitsplatz.

*Anwendersoftware* für komplexe Aufgaben wird im allgemeinen von Spezialisten für die Softwareentwicklung unter Mitwirkung des Nutzers hergestellt. Der Einsatz von Standardwerkzeugen für autonome Aufgaben erfolgt selbständig durch den Nutzer. Eine Programmierausbildung ist nicht erforderlich.

Autonome Aufgaben sind z. B.

- Verwalten von Daten (Erfassen, Aktualisieren, Auswerten), die am Arbeitsplatz entstehen und auch nur dort verwendet werden),
- Rechnen in Tabellen (Bilanzieren, Kalkulieren, Varianten berechnen),
- organisatorische Gewährleistung der Leitung und Planung, wie Terminplanung, Terminkontrolle, Projektplanung, Projektleitung,
- Textbearbeitung für Leitung, Planung, Analyse, Kontrolle und Schriftverkehr,
- Kommunikation mit anderen Partnern oder Rechnersystemen.

Bei der Auswertung von Daten, beim Rechnen in Tabellen und bei organisatorischen Aufgaben werden mathematische, kybernetische und statistische Verfahren sowie grafische Darstellungsmittel genutzt.

*Standardwerkzeuge* stehen vor allem für Personalcomputer zur Verfügung. Die folgende Darstellung gibt einen Einblick in ihren Leistungsumfang. Für die konkrete Arbeit mit dem Tabellenkalkulationsprogramm und mit REDABAS enthalten Anhang C und D Kommandoübersichten.

### 5.2. Automatisierte Textverarbeitung

#### 5.2.1. Grundbegriffe

Der Einsatz von Rechnersystemen für die Aufbereitung und Speicherung von nicht-numerischer Information in Form von Schriftzeichen (Text, vgl. Abschnitt 1.2.2.) wird als automatisierte Textverarbeitung bezeichnet.

**Automatisierte Textverarbeitung ist Aufbereitung, Verarbeitung, Speicherung, Vielfältigung und Übertragung von Texten unter Nutzung von Rechnersystemen.**

Die automatisierte Textverarbeitung gewinnt an Bedeutung, da 25 % bis 40 % der Bürotätigkeiten aus Textverarbeitungsaufgaben bestehen.

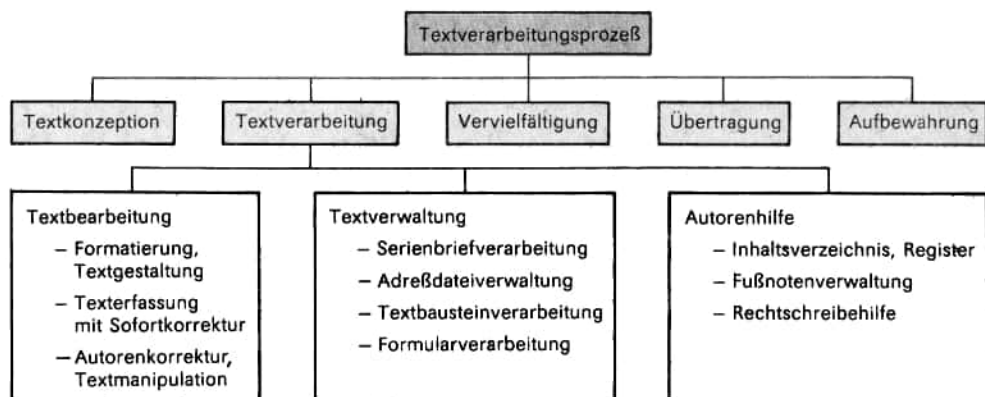


Abbildung 5.1.  
Aufgaben der Textverarbeitung

Der Textverarbeitungsprozeß kann in Teilprozesse gegliedert werden (vgl. Abbildung 5.1.).

Automatisierte Textverarbeitung mit Personalcomputern erfolgt vor allem für:

- Texte mit hoher Überarbeitungsquote (z. B. Leitungsvorlagen, Analysen),
- Texte, die aus bereits vorgefertigten Texten oder Textteilen hergestellt werden können,
- Programmtexte (Quelltexte).

Je höher das Textausgabevolumen gegenüber dem Texteingabevolumen ist, desto effektiver ist automatisierte Textverarbeitung. Neben der Verbesserung der Arbeitsbedingungen für die Schreibenden und der Verbesserung der Qualität der Textgestaltung sind vor allem die Reduzierung der Schreibarbeit und die Beschleunigung der Bürokommunikation Ziele der automatisierten Textverarbeitung.

An Textverarbeitungsprogramme werden folgende Anforderungen gestellt:

- einfache Handhabung und geringer Lernaufwand,
- Erleichterungen bei der Erfassung und Korrektur von Texten (Editorfunktionen),
- Gestaltungsmöglichkeiten von Schriftstücken (Formatierung),

- Wiederverwendung und Mischung von Texten (Textbausteinverarbeitung, Serienbriefverarbeitung, Formularverarbeitung),
- Verwaltung von Verzeichnissen und Fußnoten sowie Rechtschreibhilfe (Autorenhilfe),
- Integration mit anderen Programmen (Grafik, Datenverwaltung, Tabellenkalkulation),
- Rechnerkommunikation (in lokalen Netzen oder Fernnetzen),
- Rechnung im Text (z. B. Rechnungsbearbeitung),
- Programmierbarkeit (Speicherung von Kommandofolgen für die automatische Ausführung).

## 5.2.2. Kommunikation mit dem Textverarbeitungsprogramm

### Menüsteuerung

Die Menüsteuerung arbeitet mit Funktionsangeboten, die auf Anforderung mit unterschiedlichem Unterstützungsumfang versehen werden.

Das Textverarbeitungsprogramm TEXT 30 arbeitet z. B. mit vier Stufen der Anzeige

angebotener Funktionen. Die Stufe 0 schaltet alle Menüs ab. Der gesamte Bildschirm steht für die Aufnahme des Textes zur Verfügung. Anderenfalls (Stufe 3) ist bereits ein Drittel der Schreibfläche mit Funktionsangeboten als Ausschnitt aus dem Gesamtangebot angefüllt.

Komfortable Textverarbeitungsprogramme bieten etwa 150 Funktionen.

- Die Art der Nutzerkommunikation beeinflusst die Handhabbarkeit und das Erlernen der Arbeit mit dem Textverarbeitungsprogramm wesentlich. Ergonomisch günstig ist ein kontextabhängiges, gestuftes Angebot an Funktionen und Hilfsinformation über die Fenstertechnik. Der Nutzer kann (meist über eine Funktionstaste) weitere Funktionen oder Hilfsinformation anfordern.

### Kommandosteuerung

Bei der Kommandosteuerung werden Kommandos über Tastenkombinationen eingegeben. Die Direkteingabe von Kommandos hat gegenüber der Menüsteuerung den Vorzug einer höheren Geschwindigkeit, aber den Nachteil eines höheren Lernaufwandes.

Häufig wird eine wahlweise Nutzung der Menü- und der Kommandotechnik durch das Programm ermöglicht, so daß geübte Nutzer die Kommandosteuerung und ungeübte Nutzer die Menüsteuerung wählen können.

### 5.2.3. Textgestaltung

Bei der Textgestaltung (Formatierung) wird die Form des Textes festgelegt.

Zur Gestaltung des Textes mit einer Schreibmaschine wird das Papier im entsprechenden Format eingelegt und der Abstand eingerichtet. Bei automatisierter Textverarbeitung müssen dem programm-

gesteuerten Gerät alle Formate (linker Rand, rechter Rand, Zeilenabstand, Zeichenabstand, Schriftart) mitgeteilt werden. Bei leistungsfähigen Textverarbeitungsprogrammen bieten sich sehr viel mehr Gestaltungsmöglichkeiten an, als mit einer normalen Schreibmaschine möglich sind. Durch Schriftartenwechsel (Normalschrift oder Near Letter Quality, NLQ), Wechsel der Schreibdicke (Schriften PICA oder ELITE) und der Zeilenabstände, durch Kopf- und Fußzeilen, Abstufungen der Schreibintensität (z. B. Fettdruck), den einfachen oder mehrspaltigen Blocksatz, Zentrierung und die Einbeziehung von Grafik kann die Form des Textes vielseitig variiert und verbessert werden. Häufig genutzte Gestaltungsmittel sind Blocksatz und Proportionalschrift. Blocksatz bedeutet im Unterschied zum Flattersatz, daß linker und rechter Rand durch entsprechenden Leerstellenausgleich ausgerichtet sind. Bei der Proportionalschrift wird im Gegensatz zur Normalschrift der Zeichenabstand nicht gleichmäßig, sondern nach der Zeichenbreite der Buchstaben ausgerichtet. Dadurch entsteht ein geschlossenes Schriftbild.

Ein wesentlicher Unterschied zur Arbeit mit der Schreibmaschine besteht darin, daß das Ergebnis der Schreibarbeit zunächst nur auf dem Bildschirm sichtbar wird. Die Übertragung des Bildschirminhaltes auf Papier kann in zwei Varianten erfolgen. Bei der ersten Variante stimmt die Darstellung auf dem Bildschirm mit der Darstellung auf dem Papier überein. Bei der zweiten Variante werden Steuerzeichen im Text verwendet, die bei der Druckausgabe zur Textgestaltung ausgewertet werden. Diese Variante hat den Vorteil, daß die Steuerzeichen korrigiert werden können. Nachteilig ist, daß die Wirkung erst auf dem Papier erkennbar wird.

### 5.2.4. Texteingabe und Textkorrektur

Texteingabe ist die Erfassung von Texten über die Tastatur. Die Korrektur erfolgt nach der Fehlererkennung oder bei der Überarbeitung des Textes. Texteingabe und Korrektur werden mit einem Editor ausgeführt. Standardfunktionen sind:

#### Bewegung im Text

Die Bewegung im Text wird vom Cursor angezeigt. Sie kann zeichenweise, wortweise, zeilenweise und bei einigen Programmen auch satzweise und absatzweise erfolgen. Weiterhin sind Sprünge zum Textanfang, Textende und zu bestimmten Markierungen möglich.

Von der Einfachheit und Schnelligkeit der Bewegungen im Text wird die Bequemlichkeit der Handhabung des Programms wesentlich bestimmt.

#### Einfügen und Überschreiben

Durch Einfügen können entweder Zeichen, Wörter oder auch längere Passagen eingefügt werden. Dabei werden die ursprünglich platzierten Zeichen nach rechts verschoben. Beim Überschreiben werden die vorhandenen Zeichen durch die Eingabezeichen ersetzt.

#### Suchen und Ersetzen

Die Funktion Suchen und Ersetzen erlaubt das Suchen nach ganzen Wörtern oder nach Wortbestandteilen. Ist der gesuchte Text gefunden, wird er angezeigt und gegebenenfalls gegen ein Wort mit oder ohne Rückfrage ausgetauscht.

#### Verschieben und Kopieren

Textblöcke können markiert und danach zur gewählten Cursorposition verschoben oder kopiert werden. Textblöcke können auch als gesonderte Datei geschrieben, gelesen oder eingefügt werden. Dadurch ist

der Austausch von Blöcken zwischen verschiedenen Dateien möglich.

#### Umbruch und Silbentrennung

Umbruch und Silbentrennung erhöhen die Eingabegeschwindigkeit. Automatischer Zeilenumbruch bewirkt den (automatischen) Übergang auf eine neue Zeile beim Erreichen der eingestellten Zeilenlänge. Dabei wird innerhalb der Randzone (nach Möglichkeit an einem Leerzeichen) auf die neue Zeile gesprungen und der rechte Rand ausgerichtet. Bei automatischer Silbentrennung erfolgt die Trennung nach den Regeln der Grammatik. Die Trennung ist zu 95 % richtig. Halbautomatische Trennverfahren unterbreiten einen Trennvorschlag, der bestätigt oder verändert werden kann.

Seitenumbruch ist ein (automatischer) Übergang zu einer neuen Seite beim Erreichen einer eingestellten Seitenlänge (Zeilenzahl). Beim Seitenumbruch wird die Seitennumerierung vorgenommen.

### 5.2.5. Funktionen zur Wiederverwendung von Texten

#### Textbausteinverarbeitung

Bei der Textbausteinverarbeitung werden Texte vollständig oder teilweise aus Textbausteinen mit hoher Wiederverwendungsmöglichkeit rechnergestützt zusammengestellt. Die Textbausteine sind Bestandteile einer *Textbausteindatei*. Die Zusammenstellung der Textbausteine eines Sachgebietes bildet ein *Texthandbuch*. Der Nutzer kann über einen Bausteinnamen den Textbaustein in den fortlaufenden Text einordnen.

### **Serienbriefverarbeitung**

Serienbriefverarbeitung ist die Zuordnung variabler Texte, wie Adressen, Anreden, Termine, zu einem Standardbrief.

Die einzufügenden Texte können von einer Datei gelesen oder während des Ausdrucks durch den Nutzer eingegeben werden. Das Einfügen von Variablen aus einer Variablendatei ist der typische Fall für die Serienbriefverarbeitung, z. B. wenn ein Brief gleichen Inhalts an unterschiedliche Empfänger zu richten ist. Die unterschiedlichen Adressen werden aus einer Adreßdatei bereitgestellt.

Einige Textverarbeitungsprogramme enthalten auch Selektierfunktionen, mit denen eine Auswahl der Adreßdaten für Serienbriefe erfolgen kann. Mit der gleichen Grundfunktion können Textbausteine aus einer Textbausteindatei eingefügt werden.

Eine andere Form der Serienbriefverarbeitung ist das Einfügen von Variablen durch Tastatureingabe. Diese Form kann genutzt werden, wenn die Anzahl der Variablen gering und das Anlegen einer Datei nicht zweckmäßig ist. Beim Schreiben des Serienbriefes sind die Stellen zu markieren, an denen das Programm beim Ausschreiben pausiert und auf die Eingabe der Variablen wartet. So kann z. B. die Anrede, die Anzahl der zu bestellenden Einheiten, das Lieferdatum oder ein bestimmter Textbaustein eingefügt werden.

Die Arbeit mit Textbausteinen und Adreßdateien bietet sich besonders bei Liefererkunden-Beziehungen an. Dabei sind vielfach ähnliche Texte zu verschiedenen Zeitpunkten an verschiedene Adressaten zu senden (z. B. Bestellungen, Verträge, Mahnungen, Rechnungen, Einladungen und Mitteilungen).

### **Formularverarbeitung**

Die Formularverarbeitung automatisiert die Positionierung des Cursors auf die Schreib-

stellen des Formulars für die Eingabe variabler Texte.

Es werden Tabulatorstellungen und Randbegrenzungen festgelegt. Dies geschieht z. B. für Vertragsvordrucke und Auftragspapiere. Dabei können weiterhin konstante Texte (Textkonstanten), wie z. B. der Betriebsname, die Betriebsnummer, an den vorgesehenen Stellen in die Formulardatei bereits eingetragen werden, während variable Angaben zum Zeitpunkt des Ausdrucks eingegeben werden.

#### **5.2.6. Weitere Textverarbeitungsfunktionen**

Für umfangreiche Texte sind folgende Zusatzfunktionen von Interesse:

- Rechtschreibprüfung (Vergleich der Schreibweise mit einem gespeicherten Lexikon),
- Verwaltung von Verzeichnissen und Fußnoten (Zusammenstellung von Inhaltsverzeichnissen, Abbildungsverzeichnissen und Sachwortregistern mit Seitenangabe sowie Einordnung und Numerierung von Fußnoten),
- Sortieren von Texten,
- Zeichensätze für wissenschaftliche Texte.

Außerdem ist das Zusammenwirken mit anderen Standardwerkzeugen möglich

#### **5.2.7. Textverarbeitungssysteme**

**Textverarbeitungssysteme sind Rechnersysteme, die zur Realisierung von Textverarbeitungsaufgaben eingesetzt werden.**

Eine Einteilung der Textverarbeitungssysteme unter dem Aspekt der rechentechnischen Basis, des Spezialisierungsgrades

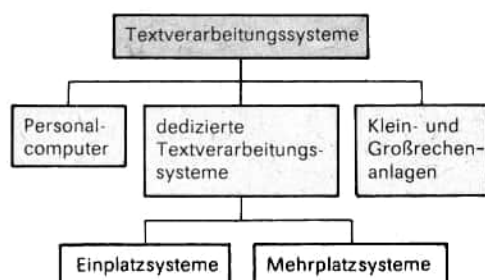


Abbildung 5.2.  
Klassifizierung  
von Textverarbeitungssystemen

des Arbeitsplatzes und der Ressourcenaufteilung ist in Abbildung 5.2. dargestellt. Elektronische Schreibmaschinen gehören nicht zu den Textverarbeitungssystemen. Sie können über eine V.24-Schnittstelle für die Informationsübertragung verfügen. In diesem Fall können elektronische Schreibmaschinen als Drucker über ein lokales Netz oder direkt mit einem Rechnersystem zusammenarbeiten.

#### Personalcomputer als Textverarbeitungssystem

Die Nutzung von Personalcomputern als Textverarbeitungssystem ist sehr verbreitet. Für die Textverarbeitung existiert eine Vielzahl von Textverarbeitungsprogrammen. Für die Textausgabe werden überwiegend Matrixdrucker verwendet.

Als Problem erweist sich die Mehrfachbelegung der Tastaturen mit internationalem und deutschem Zeichensatz (Klammern/Umlaute, Anordnung der Zeichentasten). Durch veränderte Anordnung und Belegung von Tasten ist ein Blindschreiben in gewohnter Weise nur bedingt möglich. Für die normalen Textverarbeitungsaufgaben im Büro genügen Personalcomputer in Normalausstattung mit kleinen und/oder mittleren Textverarbeitungsprogrammen. Für 8-Bit-Personalcomputer wird das Programm TEXT 30/Textprozessor TP genutzt. Dieses Programm paßt sich mit einer Über-

lagerungsstruktur den geringen Ressourcen des 64-KByte-RAM-Hauptspeichers an. Insbesondere werden die Textformatierung, das Editieren, die Textbaustein- und Serienbriefverarbeitung unterstützt. Es fehlen die Funktionen der Verzeichnis- und Registererstellung sowie die Fußnotenverwaltung, die Rechtschreibhilfe und das Rechnen im Text.

Für 16-Bit-Personalcomputer steht das Textverarbeitungsprogramm TEXT 40 zur Verfügung. Dieses System enthält eine klare Nutzerführung und eine gute Strukturierung des Funktionsangebots über Menüs. Darüber hinaus gibt es gegenüber TEXT 30 Funktionserweiterungen, wie automatische Silbentrennung (98 % Treffsicherheit), Funktionen zur Erzeugung von Verzeichnissen und Registern, Fußnotenverwaltung, Titelbearbeitung, eine größere Beweglichkeit im Text (Bildschirmblättern, Bildschirmrollen vertikal und horizontal), Löschen von Bereichen, Dezimaltabulation (dezimalstellengerechte Eingabe von Zahlen), Textübernahme und Textübergabe an andere Programmsysteme. Als neue Funktion ist die Terminverwaltung hinzugekommen. Damit können Termine (Datum, Uhrzeit, Ereignis) erfaßt und abgefragt werden (Terminliste für ein bestimmtes Datum bzw. einen Vorschauzeitraum). Eine Makrofunktion erlaubt die Speicherung bestimmter wiederkehrender Formate für die bequeme Auswahl. Es werden für alle Parameter Standardwerte angeboten, die in spezielle Parameter umgewandelt werden können.

#### Dedizierte Textverarbeitungssysteme

*Einzelplatz-Textverarbeitungssysteme* sind Rechnersysteme mit einem speziellen Betriebssystem zur ausschließlichen Nutzung für die Textverarbeitung. Als Schreibfläche dient ein Halb- oder Ganzseiten-Bildschirm. Als externes Speichermedium werden Disketten oder Festplatten benutzt. Die



Textverarbeitungssysteme können die grafische Darstellung unterstützen und mit typographischen Geräten verbunden sein. Dadurch eignen sie sich für die Bearbeitung typographischer Vorlagen (Desktop Publishing, DP).

Bei *Mehrplatz-Textverarbeitungssystemen* sind mehrere Arbeitsstationen mit einem zentralen Rechnersystem verbunden. Im Dateiverbund werden Textbausteindateien, Adreßdateien, ein zentrales Wörterbuch für die Rechtschreibunterstützung und im Ressourcenverbund schnelle Drucker genutzt. Durch den Mehrnutzerbetrieb sind spezielle Schutzmaßnahmen gegen unbefugten Zugriff auf Dokumente erforderlich.

#### **Textverarbeitung auf Klein- und Großrechenanlagen**

Die Nutzung von Klein- oder Großrechenanlagen für die Textverarbeitung ist sinnvoll, wenn umfangreiche Texte große Speicher erfordern oder spezielle Ausgabemedien, wie Mikrofiches (Computer Output of Microfiches, COM), Licht- und Fotosatz oder schnelle Drucker (z. B. Laserdrucker) zu nutzen sind. Eventuell ist auch eine zentrale Führung von Textdateien (Texthandbücher, Adreßdateien) in einem Mehrnutzerbetrieb zweckmäßig.

Auf Kleinrechenanlagen ist die Textverarbeitungskomponente NROFF und auf Großrechenanlagen ist das Textverarbeitungsprogramm TEPROS verfügbar. NROFF und TEPROS sind im Gegensatz zu den typischen Textverarbeitungsprogrammen nicht seitenorientiert.

Seitenorientierte Textverarbeitung realisiert sofortige Textformatierung beim Erfassen von Daten und beim Ändern gespeicherter Textdateien. Damit müssen sowohl Editor- als auch Textformatierungsfunktionen gleichzeitig verfügbar sein und wechselseitig ausgeführt werden.

### **5.3. Tabellenkalkulation**

#### **5.3.1. Anwendungsbereich**

Das Rechnen in Tabellen ist ein wesentlicher Bestandteil der Arbeit des Ökonomen. Die Anwendung der Personalcomputer im ökonomischen Bereich ist verknüpft mit der Bereitstellung einfach zu handhabender Programme für die Arbeit mit einem elektronischen Arbeitsblatt (worksheet, spreadsheet), das die Grundstruktur einer Tabelle hat.

**Ein Tabellenkalkulationsprogramm ist ein Standardwerkzeug zur Gestaltung autonomer Programme der Aufgabenklasse Rechnen in Tabellen.**

Mit dem Tabellenkalkulationsprogramm wird eine Tabelle vorgegeben, deren Elemente (Felder, Zellen) durch Zeilen- und Spaltenkoordinaten definiert sind (Matrix). Die Elemente der Tabelle können entweder Dateneingaben aufnehmen oder Berechnungsformeln, deren Operanden sich durch Koordinatenangaben auf Eingabefelder oder Ergebnisfelder beziehen. Die Eingabungen können für die spätere Nutzung auf Disketten gespeichert werden. Außerdem können die Ergebnisse als Tabelle oder bei einigen Programmen auch als Grafik ausgegeben werden. Bei der Tabellenkalkulation besteht die Möglichkeit der Variantenrechnung für die Ermittlung der Auswirkung von Veränderungen einzelner Daten.

Die Benutzung von Formeln ist nicht auf ökonomische Probleme beschränkt. Tabellenkalkulationsprogramme sind immer einsetzbar, wenn kleinere Datenmengen nach vorgegebenen Algorithmen auszuwerten sind. Beispiele für Anwendungen aus dem Bereich der Ökonomie sind:

– Kosten-/Preiskalkulation,

- Finanzplanung, -abrechnung,
- Bilanzierung,
- Leistungsanalyse,
- Lohn- und Gehaltsabrechnung.

### 5.3.2.

#### Grundschema der Tabellenkalkulation

##### Aufbau der Tabelle

Das Arbeitsblatt der Tabellenkalkulation enthält einen Tabellenrahmen, dessen Zeilen mit Ziffern und dessen Spalten mit Buchstaben bezeichnet sind (vgl. Abbildung 5.3.).

Bei kleineren Kalkulationsprogrammen (z. B. KP) besteht das Arbeitsblatt aus einer Matrix mit maximal 63 Spalten und 254 Zeilen.

Von der Tabelle wird stets nur ein Ausschnitt auf dem Bildschirm sichtbar. In der leeren Arbeitstabelle sind es die Spalten A bis H und die Zeilen 1 bis 20. Darunter befinden sich Kommunikationszeilen. Die obere Zeile, die Statuszeile, enthält Zustandsinformationen über das aktive Feld und über die gesamte Tabelle. Die Zeile darunter, die Anforderungszeile, enthält Programmnachrichten und Aufforderungen an den Nutzer. Die Eingabezeile dient der Eingabe von Kommandos. Wenn andere Ausschnitte der Tabelle benötigt werden, kann dies durch Verschiebung des

Bildschirmausschnitts geschehen. Dabei wird das Arbeitsblatt unter dem Bildschirm wie an einem Fenster vorbeigeschoben. Kleinere Verschiebungen werden durch Cursorstaste, größere durch Sprungbefehle oder Fenstertechniken realisiert.

##### Feldtypen und -formate

Eine Tabelle soll sowohl Texte als auch Eingabewerte und Formeln für die Berechnung der Ergebnisse aufnehmen. Diese Unterscheidung muß dem Programm durch ein spezielles Kommando zur Festlegung von Feldtypen mitgeteilt werden.

**Der Feldtyp ist eine durch Zuordnung festgelegte Eigenschaft einer Zelle (Matrixelement), einen bestimmten Datentyp aufzunehmen. Datentypen sind Texte, Zahlen und Formeln.**

*Textfelder* werden beim Kalkulationsprogramm KP durch Eingabe eines Anführungszeichens (») als erstes einzugebendes Zeichen deklariert. Die Standardspaltenbreite beträgt 9 Zeichen. Die Spaltenbreite kann von 1 bis 116 Zeichen festgelegt werden.

*Zahlenfelder* nehmen Eingabedaten für die Berechnung auf. Sie können in verschiedenen Feldformaten auftreten.

**Feldformate sind Festlegungen zur Art der Zahlendarstellung.**



Abbildung 5.3.  
Ausschnitt eines leeren Arbeitsblatts

Im Programm KP gibt es die Zahlendarstellungen Normalform mit beliebig vielen Kommastellen, Normalschreibweise mit 2 Dezimalstellen, Exponentialform und die ganzzahlige Darstellung (gerundet).

*Formelfelder* nehmen wie Textfelder bis zu 116 Zeichen auf. Es wird die in der Mathematik übliche Schreibweise benutzt. Allerdings entfällt das Gleichheitszeichen, da als Ergebnisfeld das gleiche Feld wie für die Formeleingabe benutzt wird. Formelfelder besitzen zwei Zustände. Sie enthalten die angegebene Formel und andererseits den Ergebniswert. Diese Inhalte können über ein Kommando alternativ sichtbar gemacht werden.

### Bearbeitungsschritte

Die folgenden Grundschrirte sind für die Anwendung von Tabellenkalkulationsprogrammen typisch:

- (0) Aufgabenanalyse, Bestimmung der Anwendungsziele,
- (1) Entwurf des Grundaufbaus der Tabelle,
- (2) Einrichtung des Grundrasters der Tabelle,
- (3) Eintragung der Formeln,
- (4) Eingabe der Daten und Überprüfung des Layouts,
- (5) Korrektur von Feldformaten,
- (6) Ausgabe der Ergebnistabelle,
- (7) Speicherung des Arbeitsblatts,
- (8) Wiederaufruf zur Ergänzung oder Neuberechnung der Tabelle.

Die Bearbeitungsschritte werden an einem einfachen Beispiel der Planerfüllung eines Kombines gezeigt. Es dient ausschließlich der Demonstration, denn in vielen Bereichen erfolgt die Bereitstellung dieser Information als Teil komplexer Automatisierungslösungen.

#### (0) Aufgabenanalyse

Es sollen die Ergebnisse der Planerfüllung von Betrieben eines Kombines monatlich und kumulativ (anteilige Planerfüllung, bezogen auf den Jahresplan) zum Ergebnis des Kombines für die Kennziffer industrielle Warenproduktion verdichtet werden.

#### (1) Entwurf des Grundaufbaus der Tabelle

- Tabellenkopf: Planerfüllung . IWP (1000 M);
- Spaltenüberschriften: Betrieb, Plan (Monat), Ist (Monat), % Ist/Plan (Monat), Plan (kumulativ), Ist (kumulativ), % Ist/Plan (kumulativ);
- Zeileninhalte: Betriebskennziffern, Gesamtzeile Kombinat.

#### (2) Eingabe der Texte als Grundraster der Tabelle

In die Felder der Tabelle werden die zeitunabhängigen (konstanten) Teile des künftigen Ausgabeblatts eingetragen. Dazu wird der Cursor auf das entsprechende Feld positioniert. Ist das Feld im Format für die Eintragung nicht ausreichend, wird mit dem Format-Befehl /F ein anderes Format angefordert.

#### (3) Eintragung von Formeln

Im Beispiel sind Berechnungsvorschriften für die Ermittlung der prozentualen Erfüllung einzugeben. Dazu wird an die entsprechende Position, an der die prozentuale Erfüllung erscheinen soll, die Formel eingetragen (Formelfeld). Im Beispiel ist an der Position D7 die Formel  $B7/C7 * 100$  einzutragen.

Die Eintragung der Formeln wird durch eine *Kopierfunktion* mit automatischer Anpassung an die Zellenposition erleichtert. Im Beispiel wird das Mehrfachkopierkommando (/W) genutzt. In der Anforderungszeile erscheint die Aufforderung, einen

Quelltabellebereich anzugeben. Nach Eingabe des Quellbereiches (D7) wird die Eingabe des Zielbereiches verlangt (D8:D12). Beim Kopieren erfolgt, soweit nicht anders verlangt, die Umrechnung von Feldadressen auf die jeweilige Position (Kommado: /WG5, G6:G10). Es wird bis einschließlich D12 kopiert. Damit ist mit einem Befehl für die ganze Spalte die Durchschnittsberechnung programmiert. Solange keine Eingabewerte für die Planerfüllung eingegeben sind, erscheint die Ausschrift ERROR (Fehler), da eine Division durch Null versucht wird. Für die Summation der Plan-/Ist-Werte steht die Standardfunktion SUM zur Verfügung. Für die Planwerte des Kombines je Monat ist SUM(B7:B10) anzugeben.

Die Anzeige der eingetragenen Formeln erfolgt entweder je Feld über die Statuszeile [Form = SUM(B7:B10)] oder in globaler Formeldarstellung der gesamten Tabelle mit dem Globalkommando /GF. Dabei reicht allerdings die Feldbreite zur Darstellung der gesamten Formel häufig nicht aus. Für die Formeldarstellung wird im Beispiel die Feldbreite auf 12 festgelegt, allerdings wird der Tabellenausschnitt damit verkleinert. Das Ergebnis ist in Abbildung 5.4. dargestellt.

#### (4) Eingabe der Daten

Mit der Eingabe wird eine Tabelle für einen speziellen Monat aus der Grundtabelle angelegt. Die Grundtabelle und später die aktuelle Arbeitstabelle wird auf

I	A	II	B	III	C	IV	D	V	E	VI	F	I
			Planerfüllung Kennziff Monat		IWP (1000 M) April				kumulativ April			
Betrieb			Plan		Ist		%		Plan		Ist	
Erfurt							$C7/B7 \cdot 100$					
Jena							$C8/B8 \cdot 100$					
Suhl							$C9/B9 \cdot 100$					
Meiningen							$C10/B10 \cdot 100$					
Kombinat			SUM(B7:B10)		SUM(C7:C10)		$C12/B12 \cdot 100$		SUM(E7:E10)		SUM(F7:F10)	

Abbildung 5.4.  
Arbeitsblatt im Formelstatus

Betrieb	Planerfüllung Kennziff Monat	IWP (1000 M) April	%	kumulativ April Plan	Ist	%
Erfurt	234.4	236.1	100.73	950.3	952.1	100.19
Jena	133.6	130.1	97.38	531.9	532.1	100.04
Suhl	334.7	332.9	99.46	1335.7	1336.1	100.03
Meiningen	111.5	110.9	99.46	445.2	445.1	99.98
Kombinat	814.2	810	99.48	3263.1	3265.4	100.07

Abbildung 5.5.  
Ergebnistabelle auf dem Bildschirm

einem externen Speicher aufbewahrt und bei erneuter Nutzung von dort geladen (Schritt 8). Die Eingabe der Plan- und Ist-Daten je Betrieb führt zu einer sofortigen Durchrechnung der Ergebnisse. Die Durchrechnung verzögert die Eingabe, da während der Berechnung die Eingabe gesperrt ist. Es besteht auch die Möglichkeit, die Berechnung erst auf Anforderung durchführen zu lassen.

#### *(5) Korrektur von Feldformaten*

Im Beispiel sind die Tabellenüberschriften nicht günstig plazierte. Es sind deshalb die Feldformate zu korrigieren. Um die Spalteneintragungen mit den Spaltenbezeichnungen rechtsbündig abzuschließen, werden die Standardeintragungen für Texte von linksbündig in rechtsbündig umgewandelt (/F, Z, 5, TR). Danach muß das Feld A5 mit dem Inhalt »Betrieb« auf linksbündig zurückgesetzt werden.

#### *(6) Ausgabe der Ergebnistabelle*

Für die endgültige Darstellung auf dem Bildschirm oder auf dem Papier kann auf den Tabellenrahmen verzichtet werden. Dazu wird ein Schalter im Globalkommando gesetzt (/G, G). Die Tabelle kann über den Drucker ausgegeben werden (Kommando /P,B,A1:G16,P). Das Ergebnis auf dem Bildschirm zeigt die Abbildung 5.5.

#### *(7) Speicherung des Arbeitsblatts*

Bisher ist die Tabelle ausschließlich im Arbeitsspeicher vorhanden. Beim Ausschalten des Rechnersystems geht der Tabelleninhalt verloren. Für die wiederholte Verwendung muß die Arbeitstabelle deshalb auf einem externen Speicher abgelegt werden. Die Speicherung der gesamten Tabelle, d. h. einschließlich der Formeln und Daten (wahlweise), erfolgt unter einem Dateinamen (z. B. PLANERF) auf Diskette (Kommando /R, PLANERF). Im Beispiel ist

es zweckmäßig, die Grundtabelle und die Ergebnistabelle getrennt zu speichern.

#### *(8) Wiederaufruf*

Für die Vervollständigung der Tabelle um weitere Kennziffern (oder Betriebe) muß die Datei mit dem Arbeitsblatt in den Arbeitsspeicher geladen werden (Kommando:/U). Nach Aufruf des Kommandos wird der Dateiname verlangt. Es besteht die Möglichkeit, die gesamte Tabelle (All) oder Teile davon (Part) zu laden.

In das Arbeitsblatt können Zeilen und Spalten eingefügt werden. Dabei wird die Bildschirmanzeige nach rechts oder nach unten gerollt. Es ist möglich, die Titelzeilen/-spalten beim Rollen sichtbar zu halten. Während z. B. weitere Kennziffern rechts eingetragen werden, bleibt die Spalte der Betriebe erhalten. Die anderen Spalten rollen aus dem Bild. Diese Vorgehensweise ist auch für die Titelzeile möglich.

Teile der Tabelle können vor dem Überschreiben geschützt werden.

### **Kommandoprozeduren**

Eine weitere Möglichkeit für die kommandogesteuerte Arbeit bietet die Arbeit mit Kommandoprozeduren. Die Kommandoprozedur ist eine Zusammenstellung von Kommandos. Die Folge der Kommandos wird unter einem Namen auf der Diskette abgelegt. Sie kann aufgerufen werden und wird dann ohne Eingriff des Nutzers Kommando für Kommando ausgeführt.

### **5.3.3.**

### **Leistungsmerkmale von Tabellenkalkulationsprogrammen**

Tabellenkalkulationsprogramme führen zu erheblicher Zeitersparnis. Dazu tragen Standardfunktionen bei, durch die z. B. die Eingabe einer umfangreichen Formel auf die Eingabe des Namens einer Standardfunktion reduziert wird.

Tabellenkalkulationsprogramme entwickeln sich mit den Ressourcen der Personalcomputer. Die ersten Programme entstanden für 8-Bit-Personalcomputer (z. B. Tabellenkalkulationsprogramm KP). Tabellenkalkulationsprogramme für 16-Bit-Personalcomputer bieten Funktionserweiterungen für die Datenverwaltung (Sortieren, Suchen), die Zusammenfassung von Tabellen (Konsolidieren) und Verbesserungen der Nutzerschnittstellen. Ein Vertreter dieser Tabellenkalkulationsprogramme ist TABCALC des VEB Kombinat Robotron. Die Leistungsfähigkeit von TABCALC ist aus dem Kommandovorrat und den Standardfunktionen ersichtlich (vgl. Anhang C).

Die weitere Entwicklung der Leistungsmöglichkeiten von Tabellenkalkulationsprogrammen besteht z. B. in

- Verwendung von Namen für Felder, Bereiche anstelle von Feldreferenzen,
- Übertragung der Kalkulationsdaten in Grafik oder Text, d. h., Änderungen in den Tabellen führen zu Änderungen in den abgeleiteten Darstellungen,
- Vergrößerung des Tabellenbereiches, z. B. 256 Spalten und 8192 Zeilen,
- Darstellung der Ergebnisse in Balken-, Kreis- oder Liniendiagrammen,
- Anwendung der Fenstertechnik, z. B. für parallele Darstellung von Ausschnitten aus derselben oder aus verschiedenen Tabellen,
- gleichzeitiger Verfügbarkeit von Kalkulationsprogrammen, Texten und Adressen im Arbeitsspeicher,
- Nutzung von Zusatzeingabegeräten (z. B. Maus) zur Erhöhung der Eingabegeschwindigkeit beim großflächigen Positionieren, teilweise mit grafikorientierter Benutzeroberfläche,
- Verfügbarkeit weiterer Standardfunktionen, wie Regressionsanalyse, Determinantenberechnung, Lösung von linearen Gleichungssystemen.

## 5.4.

### Datenverwaltung

#### 5.4.1.

#### Grundbegriffe

Neben der Tabellenkalkulation spielt die Nutzung von Datenverwaltungsprogrammen eine wichtige Rolle beim Einsatz von Personalcomputern. Datenverwaltungsprogramme dienen zum Anlegen und Auswerten von Dateien am Arbeitsplatz. Dazu sind keine Spezialkenntnisse der Programmierung erforderlich.

Die größte Verbreitung in der DDR hat das Datenverwaltungssystem REDABAS (RELationales DATenBANKsystem) des VEB Kombinat Robotron gefunden.

Typisch für relationale Datenbanken ist die Organisation der Daten in Tabellen. Die Auswertung der Dateien kann mit Hilfe von Relationen formuliert werden.

**Ein Datenverwaltungssystem ist ein komplexes Softwareerzeugnis (Basissoftware) zum Anlegen, Ändern und elementaren Auswerten von Datenbeständen.**

Ein Datenverwaltungssystem erweitert die Datenverwaltungsfunktionen des Betriebssystems. Es bietet Leistungen eines Datenbankbetriebssystems für Einzelplatzsysteme, wie Datenunabhängigkeit der Programme, redundanzarme Speicherung, Arbeitsweise im Kommando- und Programm-Modus, mächtige Kommandos, integrierte Masken- und Reportgeneratoren.

Die möglichen Varianten des Einsatzes eines Datenverwaltungssystems zeigt Abbildung 5.6.

#### Kommando-Modus

Beim Einsatz im Kommando-Modus erfolgt keine Speicherung der Kommandos in einem Programm. Diese Arbeitsweise ist nur bei Auswertungsproblemen mit ganz wenigen Anweisungen sinnvoll.

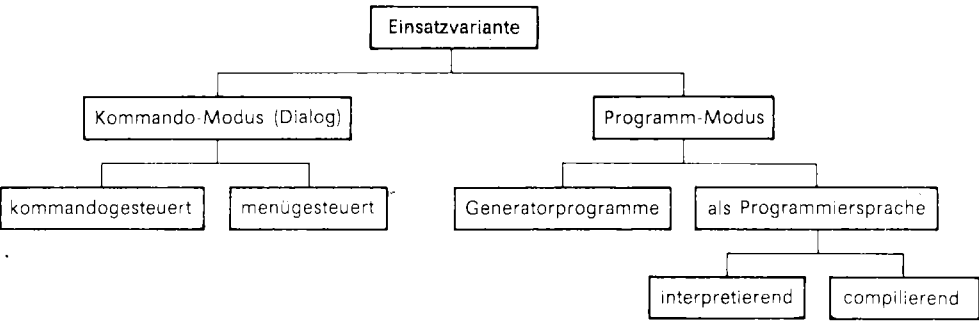


Abbildung 5.6.  
Einsatzvarianten eines Datenverwaltungssystems

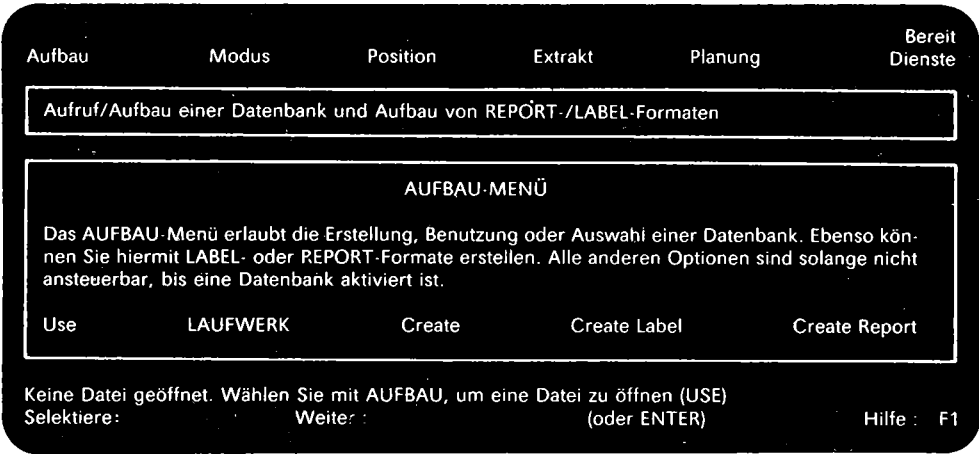


Abbildung 5.7.  
Grundmenü REDABAS-3

Die Arbeit im Kommando-Modus kann in zwei Formen der Dialogführung erfolgen. Das Datenverwaltungssystem REDABAS bietet nur eine *kommandogesteuerte Dialogführung*. Das System zeigt lediglich das Promptzeichen an. Der Nutzer muß den Kommandovorrat kennen, um ein Kommando eingeben zu können.

Bei der *menügesteuerten Dialogführung* wird das Menü der Funktionen angezeigt. Dabei werden die Funktionen nach Grundfunktionen geordnet und nach der Auswahl der Grundfunktion durch weitere An-

gebote untersetzt. Diese Dialogform wird bei den 8-Bit-Versionen durch Zusatzprodukte ermöglicht. Bei den 16-Bit-Versionen ist sie als wahlweise zu nutzende Komponente enthalten. Das Grundmenü des Datenverwaltungssystems REDABAS-3 enthält Abbildung 5.7.

**Programm-Modus**

Der Programm-Modus ist die Ausführung einer gespeicherten Kommandofolge in Form eines Programms für die wiederholte Ausführung. Die Arbeit im Programm-Modus ist die Normalarbeitsweise. Die dafür

Tabelle 5.1.  
Leistungsscharakteristik von REDABAS

Merkmal	REDABAS	REDABAS-3
Maximale Anzahl von Dateien	beliebig	beliebig
Anzahl der eröffneten Dateien	max. 15	max. 15
Anzahl der eröffneten Datenbankdateien	max. 2	max. 10
Sätze je Datei	65 535	max. 1 Milliarde
Zeichen je Satz	max. 1 000	max. 4 000
Felder je Satz	max. 32	max. 128
Feldtypen, Feldlänge	alphanumerisch, max. 254 numerisch, max. 10 logisch, 1	alphanumerisch, max. 254 numerisch, max. 19 logisch, 1 Datum, 8 MERK, max. 10, 4 096 Zeichen je Merkfeld in Merkteil mit REDABAS-3-Texteditor
Sortierkriterien	1	max. 32
Prozeduren je Prozedurdatei	—	max. 32
Anzahl der Speichervariablen	64 (global)	256 (global und lokal)

erforderlichen Komponenten können generiert oder programmiert werden.

Das *Generieren* von Programmen erfolgt mit Hilfe eines Programmgenerators.

**Ein Programmgenerator ist ein Programm, das aus einfachen Kommandofolgen Anwenderprogramme erzeugt.**

Für die Programmgenerierung muß der Nutzer auf Anforderung Parameter mitteilen. Auf diese Weise ist ohne Spezialwissen die Herstellung kleiner, aufgabenspezifischer Programme möglich.

Zur *Programmierung* verfügt das Datenverwaltungssystem REDABAS über eine Programmiersprache. Mit dieser Programmiersprache lassen sich durch die Nutzung von Kommandos als Sprachelemente, die Verfügbarkeit von Generatoren zum Aufbau von Bildschirmmasken (Maskengenerator) und die Generierung von Drucklisten (Reportgenerator) am Arbeitsplatz leistungsfähige Lösungen schaffen.

#### 5.4.2.

#### **Merkmale und Funktionen eines Datenverwaltungssystems**

##### **Komponenten**

Das Datenverwaltungssystem REDABAS besteht aus Moduln, die im allgemeinen interpretativ ausgeführt werden. Mit den Moduln ist der Dialogbetrieb im Kommando-Modus möglich. Zusätzlich zu diesen Moduln kann das Datenverwaltungssystem durch vom Nutzer selbst geschriebene Programmdateien ergänzt und mit nutzerspezifischen Verwaltungsfunktionen ausgestattet werden. Mit diesen Programmen ist die Arbeit im Programm-Modus möglich.

##### **Leistungsmerkmale**

Die Leistungsmerkmale von REDABAS und REDABAS-3 enthält Tabelle 5.1.

Die Leistungsgrenzen von REDABAS ergeben sich durch die Anzahl der Felder je Satz und die für viele Aufgaben langsame Arbeitsweise. Bei REDABAS-3 werden alle Grenzen hinausgeschoben.



**Datenbestände**

Die durch das Datenverwaltungssystem zu verwaltenden Datenbestände bestehen aus folgenden Dateiarten (in Klammern die Typnamen der Dateien):

- Daten(bank)dateien (.DBD). Sie enthalten die Nutzerdaten und deren Strukturbeschreibung.
- Indexdateien (.IDX). Sie dienen dem beschleunigten Zugriff auf Datensätze über ein Ordnungskriterium. Die Indexdatei enthält zu jedem Satz einer Daten-datei den Schlüssel und einen Zeiger.
- Variablendateien (.VAR). Sie dienen der Aufnahme variabler, temporärer Information, z. B. Aufnahme von Eingabewerten und von Zwischenergebnissen.
- Befehlsdateien (.PRG). Sie enthalten die Verarbeitungsprogramme für die Arbeit im Programm-Modus.
- Reportdateien (.DEF). Sie enthalten die Parameter zur Beschreibung von Auswertungslisten und können wiederholt für eine Ausgabe aufgerufen werden.

- Maskendateien (.MSK). Sie beschreiben den Bildschirmaufbau für die programmierte Dateneingabe/-ausgabe.
- Sicherungsdateien (.BAK). Sie werden durch das Datenverwaltungssystem beim Editieren bestehender Befehlsdateien angelegt.
- Textdateien (.TXT). Sie sind für den Datenaustausch mit anderen Programmen oder Zeichenfolgen für die Datenausgabe bestimmt.

**Funktionsübersicht**

Ein großer Teil der Funktionen kann im Kommando-Modus und als Bestandteil von Programmen (Programm-Modus) genutzt werden. Eine Übersicht über die Zuordnung wichtiger Funktionen zu Aufgabenklassen zeigt Tabelle 5.2. Die Funktionen sind in der alphabetischen Kommandoübersicht im Anhang D erläutert.

Tabelle 5.2.

Zuordnung von Kommandos (Auswahl)  
des Datenverwaltungssystems REDABAS-3 zu Aufgabenklassen

Aufgabenklasse	Kommando REDABAS-3
Nutzerunterstützung	HELP, ASSIST
Dateioperationen mit einer Datei	CREATE, USE, CLOSE, DELETE FILE, ERASE, COPY FILE, MODIFY STRUCTURE, DISPLAY, LIST, LIST STRUCTURE, DISPLAY STRUCTURE, RENAME, SORT, INDEX, REINDEX
Aktualisierung von Dateien/Sätzen	APPEND, DELETE, PACK, RECALL, INSERT, EDIT, BROWSE, CHANGE, REPLACE
Auswertung von Dateien	FIND, SEEK, LOCATE, DISPLAY, LIST, SET FILTER, SUM, COUNT, AVERAGE, TOTAL, REPORT, LABEL
Bearbeitung mehrerer Dateien	APPEND FROM, JOIN, UPDATE,
Programmablaufsteuerung	CANCEL, CLEAR, DO, QUIT, RETURN, RUN, MODIFY COM- MAND, DO WHILE, IF, ELSE, DO CASE, PROCEDURE, ACCEPT, INPUT, READ,

### 5.4.3.

#### Einführung in die Arbeit mit einem Datenverwaltungssystem

Als Prototyp für ein Datenverwaltungssystem wird das System REDABAS-3 zur Erläuterung der Handhabung benutzt. Das System REDABAS (für 8-Bit-Personalcomputer) hat einige Einschränkungen, besitzt aber meist gleiche oder ähnliche Kommandos.

#### Start, Nutzerführung und Verlassen des Systems

Durch Eingabe des Namens (REDABAS-3) wird das Datenverwaltungssystem geladen. Eine große Unterstützung des Anfängers bietet die menügesteuerte Kommandoführung ASSIST, die als Kommando durch die Eingabe des Namens aufgerufen wird. Auf die Erläuterung der einzelnen Nutzerhandlungen im Zusammenhang mit der Menüsteuerung wird verzichtet. Das Verlassen des Systems erfolgt mit dem Kommando QUIT.

#### Datendatei anlegen

Für das Anlegen einer neuen Datei muß zunächst die Struktur, d. h. der Aufbau der Datensätze, festgelegt werden. Die Struktur besteht aus einer namentlichen Feldbezeichnung (mit Buchstaben beginnend, maximal 10 Stellen), einer Angabe des Feldtyps und einer typabhängigen Angabe. Feldtypen können alphanumerisch, numerisch, logisch, Datum oder ein Verweisfeld für separate Textdateien (MERK) sein. Für das Beispiel zur Auswertung der Planerfüllung (vgl. Abschnitt 5.3.) ist die aus Abbildung 5.8. ersichtliche Strukturbeschreibung möglich. Es entsteht die Strukturbeschreibung der Datei LEIST.DBD.

Der Feldname IWP1P steht für die Kennziffer industrielle Warenproduktion/Plan des ersten Betriebes.

Feldname	Typ	Länge	Dezimalstellen
BETRIEB	C	15	
IWP1P	N	9	1
IWP1I	N	9	1
IWP2P	N	9	1
IWP2I	N	9	1
IWP3P	N	9	1
IWP3I	N	9	1
IWP4P	N	9	1
IWP4I	N	9	1

Abbildung 5.8.

Maske der Strukturbeschreibung

Satz Nr.	
Betrieb:	
IWP1P	:
IWP1I	:
IWP2P	:

Abbildung 5.9.

Maske für die Datenerfassung (Auszug)

#### Datensätze erfassen

Nach der Beschreibung der Dateistruktur (Datensatzaufbau) wird vom Datenverwaltungssystem die Frage gestellt, ob sich direkt eine Dateneingabe anschließen soll. Wird die Frage bejaht, erscheint die Standardeingabemaske (vgl. Abbildung 5.9.).

Für den Fall einer Datenerfassung, unabhängig von der Erstellung der Strukturbeschreibung, ist die gewünschte Datei über das Kommando USE einzugeben.

Damit wird die Datei eröffnet. Das Kommando APPEND bewirkt die Bereitstellung der Maske zur Erfassung weiterer Sätze.

#### Datensätze ändern

Mit dem Kommando BROWSE kann ein Ausschnitt aus der Datenbank (17 Sätze, entsprechend dem Aufnahmevermögen einer Bildschirmseite) angezeigt werden. Durch Angabe von Feldnamen ist feldweises Ändern möglich (z. B. Eingabe der Planwerte eines bestimmten Monats). Zur Anzeige kann der sichtbare Ausschnitt wie

bei der Tabellenkalkulation über die Datei horizontal und vertikal verschoben werden. Dabei können Vorspalten fixiert werden. Die Positionierung auf einen gewünschten Satz kann durch sequentielle Suche erfolgen, z. B.

LOCATE FOR ORT = 'Suhl'

#### Datensätze löschen

Ist ein Satz aus der Datei zu entfernen, so kann dies logisch (DELETE) oder physisch (PACK) erfolgen. Das logische Löschen (Löschmarkierung setzen) erlaubt auch das Reaktivieren (RECALL).

#### Datensätze auflisten

Das Anzeigen aller Sätze oder der Sätze, die einer bestimmten Bedingung genügen, ist sehr einfach. Wenn z. B. alle Betriebe gefordert werden, die den Monatsplan April in der Kennziffer IWP nicht erfüllt haben, so kann formuliert werden:

LIST FOR IWP11 / IWP1P < 1

Es werden alle zutreffenden Sätze angezeigt. Die Anzeige kann auf bestimmte Felder eingeschränkt werden.

Die Formulierung von Bedingungen kann logische Operatoren (.AND., .OR. und .NOT.) sowie weitere Vergleichsoperatoren enthalten. Es werden z. B. alle Betriebe mit einer IWP > 120 und einer Planuntererfüllung angezeigt durch:

LIST BETRIEB FOR IWP11 > 120 .AND.  
IWP11/IWP1P < 1

#### Auswertungen

Einige Funktionen für die Auswertung von Daten sind vorgefertigt, wie die Bildung von Durchschnittsn (AVERAGE), die Bildung von Summen (SUM) und die Zählung von Datensätzen (COUNT).

Für die Bestimmung der industriellen Warenproduktion des Kombinats kann folgende Angabe benutzt werden:

LOCATE FOR BETRIEB = 'Kombinat'  
REPLACE IWP11 WITH SUM IWP11

Für die Ermittlung der Anzahl der Betriebe mit einer IWP > 120 im Monat April kann die folgende Zählung erfolgen:

COUNT ALL FOR IWP11 > 120

#### Sortieren und Indizieren

Meist werden Auswertungen in einer bestimmten Sortierfolge verlangt. So kann die Beispieldatei nach Betrieben in alphabetischer Reihenfolge der Namen erwartet werden. Nach vorheriger Eröffnung der Datei (USE) gilt das Kommando:

SORT TO PLANERF ON BETRIEB/A

PLANERF.DBD ist der Name der sortierten Datei. Es kann nach bis zu 32 Sortierbegriffen sortiert werden. Sortieren bedeutet stets das Anlegen einer neuen Datei in der geforderten Sortierfolge. Ein Problem ist dabei die Sicherung der Sortierfolge bei dem Hinzufügen und Löschen von Datensätzen. Die Arbeit mit sortierten Dateien wird empfohlen bei sequentieller Verarbeitung, d. h., wenn alle Sätze einer Datei verarbeitet werden. Für den Fall, daß eine wahlfreie Verarbeitung überwiegt, wenn z. B. Angaben zu bestimmten Sätzen gefordert werden, ist die Arbeit mit Indexdateien vorteilhafter.

*Indexdateien* sind Sortierfolgen, die aus dem Sortierschlüssel und der Satznummer als Verweis auf die Position in der Datendatei gebildet werden. Indexdateien existieren zusätzlich zu den Datendateien. Sie gestatten einen schnellen Zugriff über die ihrer Sortierung entsprechenden Sortierbegriffe.

Das Kommando

INDEX ON BETRIEB TO PLANN

bewirkt das Indizieren der Plandatei nach dem Feld BETRIEB. Um eine bereits indi-

zierte Datei über den Index zu verarbeiten, ist es erforderlich, sie zu eröffnen mit dem Kommando:

USE PLANERF INDEX PLANN

Die Anweisung SEEK 'Erfurt' positioniert unmittelbar auf den gewünschten Datensatz. Die Suche mit SEEK (= Finden) ist deswegen besonders schnell, weil nur das eine indizierte Feld abgefragt wird und nicht der gesamte Satz wie etwa bei der Suche mit dem Befehl LOCATE. Der Satz kann nun bearbeitet und z. B. mit LIST angezeigt werden.

### Listengenerator

Die bisherigen Ausgaben im Beispiel verzichteten auf jede Listengestaltung (Überschriften, Summen, Zwischensummen). Eine einfache Form, Standardformulare auf dem Bildschirm oder auf Papier (Druckausgabe) zu gestalten, bietet der integrierte Listengenerator. Der Nutzer gibt menügesteuert eine Spezifikation des Aufbaus und des Inhalts der Liste an. Mit dem Kommando

CREATE REPORT ERFUEL

wird eine Listenbeschreibungsdatei ERFUEL.DEF entworfen. Sie kann mit dem Kommando

MODIFY REPORT ERFUEL

modifiziert werden. Zunächst ist es möglich, eine Überschrift der Liste anzugeben. Weiter können die Zeilenlänge, der linke und rechte Rand sowie die Seitenlänge von einer Standardannahme bestätigt oder korrigiert werden. Außerdem wird nachgefragt, ob bzw. welche Zwischensummen im Report angegeben werden sollen. Für Zwischensummen sind Zwischenüberschriften zu benennen. Schließlich wird der eigentliche Inhalt festgelegt.

Ein Report besteht aus Zeilen und Spalten. Die Spalten entsprechen den Datensätzen.

Zur Hilfestellung bei der Auswahl der in die Liste zu übernehmenden Felder wird die Strukturdatei eingeblendet. Für die gewünschten Spalten sind der Feldname, die Spaltenbreite und eine Überschrift anzugeben. Eine Spalte kann aber statt durch ein Feld auch erst durch einen Ausdruck berechnet werden (z. B. die Planerfüllung). Es können einschränkende Bedingungen für die Ausgabe gesetzt werden.

Im Beispiel soll ein Bericht erzeugt werden, der die kumulative Planerfüllung ausweist. Die Ermittlung der Kumulativwerte für den anteiligen Jahresplan und für die Ist-Werte sowie die prozentuale Erfüllung kann im Report selbst erfolgen. Damit brauchen diese abgeleiteten Daten nicht in der Datei gespeichert zu werden. Die einmal in dem Report definierte Auswertungs- und Formatierungsvorschrift ist für spätere Anwendungen stets verfügbar. Es sind lediglich die Ist-Daten zu ergänzen, und der Report ist für den jeweiligen Monat zu spezifizieren. Einen mit

CREATE REPORT ERFUEL HEADING April

erzeugten Report zeigt Abbildung 5.10.

Die Anweisung TO PRINT bewirkt die Ausgabe des Reports über den Drucker.

Ein besonderer Vorzug besteht darin, daß ein Report mit jeder Datendatei erzeugt werden kann, welche die geforderten Felder enthält. Mit der Modifikations- und der Kopierfunktion können sehr leicht Variationen einer Liste erzeugt werden.

### Filtertechnik

Mit einer Filteranweisung können die durch eine Bedingung ausgewählten Datensätze für eine weitere Bearbeitung bereitgestellt werden. Die Anweisung

SET FILTER TO IWP4P > 150

stellt alle Sätze zur Bearbeitung bereit, bei denen die industrielle Warenproduktion den angegebenen Wert übersteigt. Durch

Seiten-Nr. 1	April					
Betrieb	Kumulative Planerfüllung					
	Plan	Ist	%	Plan kum.	Ist kum.	%
Erfurt	234.4	236.1	100.73	950.3	952.1	100.19

Abbildung 5.10.  
Report (Auszug)

das Filtern verhält sich eine Datei bei allen Operationen, als würde sie nur aus den selektierten Datensätzen bestehen.

**Auswerten von Datumsfeldern**

REDABAS-3 ermöglicht durch Datumsfunktionen, z. B. Tage und Monate aus dem Datumsfeld in ausgeschriebener Form darzustellen. Mit dem Datum können auch arithmetische Operationen ausgeführt werden, z. B. zur Berechnung von Fälligkeiten oder Vertragsstrafen.

**Arbeit mit Zeichenketten**

Bei bestimmten Aufgaben ist die Umwandlung zwischen numerischen Ausdrücken und alphanumerischen Zeichenketten erforderlich. Weiterhin ist es möglich, mit Ausschnitten aus Zeichenketten zu arbeiten, z. B. mit dem Geburtsdatum aus der Personenkennzahl.

**Datensatzstrukturen anzeigen und ändern**

Ein besonderer Vorzug eines Daten(bank)verwaltungssystems ist die Möglichkeit, Datenstrukturen zu verändern, ohne daß dabei bereits bestehende Daten-dateien unbrauchbar werden. Die Struktur einer Datei kann mit den Anweisungen DISPLAY STRUCTURE oder LIST STRUCTURE sichtbar gemacht werden. Mit dem Kommando MODIFY STRUCTURE wird eine Korrektur ermöglicht (Änderung von Feldattributen, z. B. Feldlänge, Streichen oder Hinzufügen von Feldern). Zuerst wird

durch das System eine Sicherheitskopie angelegt. Anschließend bekommt der Nutzer die Editiermaske der Struktur (wie bei CREATE) für eine Veränderung angeboten. Nach dem Editiervorgang (⌘CTRL)W lädt das System die Datei aus der Sicherheitskopie. Dabei werden nur die Felder, die vor und nach der Strukturänderung die gleichen Namen haben, übernommen.

**Arbeit mit mehreren Dateien**

Bei redundanzfreier Speicherung von Daten in Dateien ist jede Information physisch nur an einer Stelle in der Datenbank gespeichert. Dadurch wird Speicherplatz gespart. Vor allem kann die Pflege der Datenbestände relativ einfach und sicher gehandhabt werden (Sicherung der Datenkonsistenz). Damit verbunden ist die Forderung nach einer hohen Flexibilität des Zugriffs zur Information. Es muß von jedem Programm aus auf sie zugegriffen werden können. Insbesondere müssen bei Bedarf auch Daten aus verschiedenen Dateien gleichzeitig verfügbar sein.

Zur simultanen Bearbeitung mehrerer Dateien werden Arbeitsbereiche angelegt (bis zu 10). In jedem Arbeitsbereich kann eine Datei aktiviert werden. Es können bis zu 10 Dateien eröffnet werden. Unmittelbar im Zugriff befinden sich aber nur jeweils 2 Dateien. Die übrigen Dateien behalten nach dem Umschalten [SELECT (Bereichsnummer)] ihren augenblicklichen Zustand, d. h., der Datensatzzeiger zeigt auf den zuletzt bearbeiteten Datensatz. Die Dateien

sind auf diese Weise synchronisiert, d. h., paarige Sätze (Paarigkeit der Ordnungsbegriffe) können bearbeitet werden.

Typische Anwendungen der Bearbeitung mehrerer Dateien sind:

- Aktualisierung einer Stammdatei mit Hilfe einer Änderungsdatei, z. B. Aktualisierung der Leistungsdaten
- Fortschreibung einer Bestandsdatei, z. B. Buchungen auf Sparkonten,
- Verarbeitung einer Bewegungsdatei mit Hilfe einer Stammdatei, z. B. Bewertung von Verkäufen beim Ausschreiben von Rechnungen.

Die Arbeit mit mehreren Dateien ist typisch im Programm-Modus. Für die Programmierung mit REDABAS-3 ist die Programmablaufsteuerung zu nutzen.

### 5.5. Integrierte Softwarepakete

Wachsende Anforderungen an die Nutzer entstehen durch die Integration verschiedener Standardwerkzeuge. Ausgangspunkt der Integration war die Erweiterung der Tabellenkalkulationssysteme, zunächst um grafische Auswertungsmöglichkeiten der Daten und schließlich um Datenverwaltungsfunktionen. Das Ergebnis sind integrierte Standardwerkzeugpakete. Die Vorteile für den Nutzer liegen in der Vereinheitlichung der Nutzeroberfläche und der Integration von Anwendungsrichtungen in einem Paket. In der weiteren Entwicklung wurden weitere Funktionen vereinigt:

- Textverarbeitung,
- Tabellenkalkulation,
- Datenbank- oder Datenverwaltung,
- Geschäftsgrafik,
- Kommunikationsfunktionen.

Die Integration vieler Funktionen in einem Paket führt zu einem höheren Lernaufwand bei der Einarbeitung. Hinzu kommt, daß die Komponenten der integrierten Pakete

vielfach nicht den Leistungsumfang von Einzelwerkzeugen erreichen. Integrierte Pakete benötigen auf Grund ihrer Komplexität einen Arbeitsspeicher von mindestens 512 KByte. Sie haben meist eine Orientierung auf einen speziellen Schwerpunkt. Vom VEB Kombinat Robotron werden zwei integrierte Pakete angeboten.

Das Paket ARIADNE hat den Schwerpunkt auf der konzeptionellen Arbeit und der analytischen Aufbereitung. Es besteht aus den Komponenten Textkonzeption, Textverarbeitung, Tabellenkalkulation, Datenverwaltung und Geschäftsgrafik.

Das Paket MULTICOMP enthält neben diesen Funktionen zusätzlich eine Schreibumgebung, wie Adressenverwaltung, Kalender, Notizblock, Taschenrechner und Umrechnungstabellen.

Der Trend der Weiterentwicklung der Pakete geht zur Verstärkung der Leistungsfähigkeit der Einzelkomponenten und zur Erleichterung der Benutzung.

### Übungen

1. Entwickeln Sie ein Beispiel zur Anwendung der automatisierten Textverarbeitung für die Serienbriefverarbeitung!
2. Nennen Sie Möglichkeiten der automatisierten Textverarbeitung, die sich für die Ausarbeitung einer (ökonomischen) Jahresanalyse einsetzen lassen!
3. Verbessern Sie die Ausgangstabelle des Beispiels im Abschnitt 5.3.2., so daß sich die einzugebenden Werte verringern!
4. Erweitern Sie das Beispiel für die Tabellenkalkulation aus Abschnitt 5.3. um Betriebe und Kennziffern!
5. Ergänzen Sie die Strukturdatei der Plandatei um die Kennziffer Nettoproduktion! Notieren Sie möglichst viele Varianten zur Auswertung im Kommando-Modus und in Form eines Reports!
6. Demonstrieren Sie am Beispiel die Vorteile integrierter Textverarbeitung und Datenverwaltung!

# Anhang A

## Metasprachliche Notation in erweiterter, nichtrekursiver. Chomsky-Backus-Naur-Form

### Elemente

Terminalsymbole	Terminalsymbole der Metasprache werden unverändert in den Zieltext übernommen. Es handelt sich um Wörter oder die Sonderzeichen Schrägstrich (/ oder \), Komma (,), Punkt (.), Semikolon (;), Vorzeichen (+, −). Die Darstellung der Wörter erfolgt durch Großschreibung und Fettdruck.
Nichtterminalsymbole	Nichtterminalsymbole der Metasprache werden im Zieltext durch an anderer Stelle definierte Konstruktionen ersetzt. Die Darstellung von Nichtterminalsymbolen erfolgt durch Bezeichnungen in spitzen Klammern (z. B. <Name>) oder im <i>Kursivdruck</i> .
Basissymbole	Basissymbole sind Buchstaben, Ziffern und das Unterstrichungszeichen ( <u>  </u> ).

### Bildungsregeln

Allgemein	Die Notation wird von links nach rechts unter Beachtung der Zeichen  , [, ], { und } gelesen. Es bedeuten:
	Exklusives Oder (eine Möglichkeit muß genutzt werden).
[ ]	Optional (der eingeschlossene Teil kann auch entfallen).
{ }	Wiederholung (der eingeschlossene Teil kann wiederholt verwendet werden, eventuell auch nullmal, d. h., er kann entfallen).

# Anhang B

## Kommandoübersicht DCP

**ASSIGN** [*Laufwerk\_neu=Laufwerk\_alt*]{*Laufwerk\_neu=Laufwerk\_alt*}  
Laufwerksumweisung und Rücksetzen (ohne Parameter).

**ATTRIB** [*Status*] [*Dateiänderungsattribut*] *Quellspezifikation*  
Veränderung von Dateiattributen.

**BACKUP** [*Quellspezifikation*] *Ziellaufwerk* [*/S*] [*/M*] [*/A*] [*/D: Datum*]  
[*/T: Zeit*]

Sicherungskopien anlegen. Schalter:

**/S** Sicherungskopien aller Dateien.

**/M** Sicherungskopien der seit der letzten Sicherung veränderten Dateien.

**/A** Hinzufügen von Sicherungsdateien zu Dateien, die sich bereits auf der Sicherungsdiskette/-platte befinden. Bei Nichtangabe werden die bereits vorhandenen Dateien gelöscht.

**/D** Sicherungskopien von den seit *Datum* veränderten Dateien.

**/T** Sicherungskopien von den seit *Zeit* veränderten Dateien.

*Basisname* ::= *Namenszeichen* {*Namenszeichen*}

*Bedingung* ::= *Fehlerniveau* | *Identität* | *Existenz*

*Betriebsmodus\_für\_Adapter* ::= *COM* [:] *Datenübertragungsrate*  
[,*Parität* [,*Datenbit*  
[,*Stoppbit* [,*P*]]]]

Schalter:

**P** Adapter für asynchrone Übertragung kann für einen Drucker mit serieller Schnittstelle benutzt werden.

*Betriebsmodus\_für\_Drucker* ::= *Paralleldrucker* [:] [*Zeile*] [,*Zoll*] [,*P*]

Schalter:

**P** Fortlaufende Wiederholungen bei Zeitlimitüberschreitungsfehlern.

*Bildschirmadapter* ::= **40** | **80** | **CO40** | **CO80** | **MONO**

*Bildschirmanzeige* ::= **ON** | **OFF**

Schalter:

**ON** Bildschirmanzeige aus.

**OFF** Bildschirmanzeige ein.

**BREAK** [*Unterbrechung*]  
*Kommandoabbruch.*



*Buchstabe* ::= A | B | C | D | E | F | G | H | I | J | K | L | M |  
 N | O | P | Q | R | S | T | U | V | W | X | Y | Z |  
 a | b | c | d | e | f | g | h | i | j | k | l | m |  
 n | o | p | q | r | s | t | u | v | w | x | y | z | \_

**CHDIR** [*Laufwerk*] [*Pfad*]

Aktuelles Dateiverzeichnis ändern.

(Statt **CHDIR** ist **CD** möglich)

**CHKDSK** [*Quellspezifikation*] [/F] [/V]

Diskettenanalyse und Ausgabe Statusbericht. Schalter:

/F Korrektur der Fehler im Dateiverzeichnis oder in der Dateizuordnungstabelle  
 und Korrekturen auf die Diskette/Platte schreiben.

/V Anzeige aller Dateien und deren Pfad.

**CLS**

Bildschirm löschen.

*Code* ::= \$ | t | d | p | v | n | g | l | b | q | h | e

*Com* ::= COM1 | COM2

**COMMAND** [/P] [/C*Zeichenkette*] [/E:*Umgebungsgröße*]

Start des sekundären Kommandointerpreters. Schalter:

/P Eintragung einer Kopie des Kommandointerpreters in den Speicher.

/C Übergabe einer Zeichenfolge.

/E: Angabe der Umgebungsgröße.

**COMP** [*Dateispezifikation\_1*] {*Dateispezifikation\_2*}

Vergleich zweier Dateien.

**COPY** *Kopieren\_gleiche\_Dateibezeichnung* |

*Kopieren\_unterschiedliche\_Dateibezeichnung* |

*Verketteten* | *Datenaustausch*

Kopieren von Dateien und Datenaustausch zwischen reservierten Einheiten.

**CTTY** *Einheiten*

Ändern Standardeingabe-/ausgabekonsole.

**DATE** [*Datum*]

Eingabe und Ausgabe des Datums.

*Dateiänderungsattribut* ::= +A | -A

Schalter:

+A Dateiänderungsattribut setzen.

-A Dateiänderungsattribut aufheben.

*Dateigröße* ::= Zahl

*Dateiname* ::= \Basisname [.Typname]

*Dateiname\_neu* ::= Quellspezifikation

*Dateispezifikation* ::= [Laufwerk] [Pfad] [Dateiname]

*Dateispezifikation\_1* ::= Quellspezifikation

*Dateispezifikation\_2* ::= 'Quellspezifikation

*Dateispezifikation\_alt* ::= Quellspezifikation

*Dateispezifikation\_Ziel* ::= *Dateispezifikation*

*Datenaustausch* ::= *Von Nach*

*Datenbit* ::= **7** | **8**

*Datenträgername* ::= *Zeichen* {*Zeichen* | *Spezialsymbol*}

*Datenübertragungsrate* ::= **110** | **150** | **300** | **600** | **1200** | **2400** | **4800** | **9600**

*Datum* ::= *Monat Trennzeichen Tag Trennzeichen Jahr*

**DEL** *Quellspezifikation*

Löschen von Dateien.

(Statt **DEL** ist **ERASE** möglich.)

**DIR** [*Quellspezifikation*] [**/P**] [**/W**]

Anzeige der Dateiverzeichniseinträge. Schalter:

**/P** Pause nach jeder Bildschirmseite.

**/W** Anzeige in fünf Spalten.

**DISCOMP** [*Laufwerk* [*Laufwerk*]] [**/1** [**/8**]]

Vergleich zweier Disketten. Schalter:

**/1** Vergleicht nur die erste Seite von doppelseitigen Disketten.

**/8** Vergleicht nur 8 Sektoren pro Spur.

**DISCOPY** [*Laufwerk* [*Laufwerk*]] [**/1**]

Kopieren von Disketten. Schalter:

**/1** Kopieren nur der ersten Seite der Diskette.

*Drucker* ::= **PRN** | *Paralleldrucker*

*Druckertyp* ::= **COLOR1** | **COLOR4** | **COLOR8** | **COMPACT**

**DMODE** [*Laufwerk=Spuren*] {*,Laufwerk=Spuren*}

Anzeige und Modifikation der Disk-Parameter.

**ECHO** [*Bildschirmanzeige* | *Kommentar*]

Bildschirmanzeige des aktuellen Kommandos aus einer Stapelverarbeitungsdatei.

*Einheiten* ::= **AUX** | *Com* | **CON**

**EXE2BIN** *Dateispezifikation* [*Dateispezifikation*]

Umwandlung von .EXE-Dateien in .COM- oder .BIN-Dateien.

*Existenz* ::= **EXIST** *Dateispezifikation*

**FDISK**

Erstellen DCP-Partition.

*Fehlerniveau* ::= **ERRORLEVEL** *Zahl*

**FIND** [**/V**] [**/C**] [**/N**] *Zeichenkette*

*Dateispezifikation* {*Dateispezifikation*}

Suchen einer Zeichenkette in den angegebenen Dateien. Schalter:

**/V** Anzeige aller Zeilen, die nicht die *Zeichenkette* enthalten.

**/C** Angabe der Anzahl der Zeilen, die die *Zeichenkette* enthalten.

**/N** Angabe der Zeilennummer und Ausgabe der Zeile, in der die *Zeichenkette* enthalten ist.

**FOR** *%% Variable* **IN** (*Satz*) **DO** Kommando

Wiederholte Ausführung von Kommandos in einer Stapelverarbeitungsdatei. Kommando ist durch ein gültiges DCP-Kommando, außer **FOR**, zu ersetzen.

**FORMAT** [*Laufwerk*] [**/S**] [**/1**] [**/8**] [**/V**] [**/B**]

Formatieren (Initialisieren) einer Diskette. Schalter:

**/S** Kopieren der Betriebssystemdateien von der Diskette im Standardlaufwerk auf die neu formatierte Platte oder Diskette.

**/1** Formatiert eine Seite der Diskette in einem double-sided Laufwerk.

**/8** Formatiert die Diskette/Platte mit 8 Sektoren pro Spur.

**/V** Bei Angabe eines Namens einer Diskette/Platte.

**/B** Erstellt eine Diskette mit 8 Sektoren pro Spur und ordnet den Systemdateien BIO.COM und DOS.COM Platz zu.

**GOTO** *Sprungziel*

Sprung an eine andere Stelle des Stapelverarbeitungsprogramms.

**GRAFTABL**

Laden der Zeichentabelle für Grafik.

**GRAPHICS** [*Druckertyp*] [**/R**]

Ausgabe Grafik-Bildschirm auf Drucker vorbereiten. Schalter:

**/R** schwarz als schwarz und weiß als weiß gedruckt, sonst umgekehrt.

*Hundertstel\_Sekunde* ::= **00** | **01** | .. | **99**

*Identität* ::= *Zeichenkette\_1* = *Zeichenkette\_2*

**IF** [**[NOT]**] *Bedingung* Kommando

Bedingte Ausführung von Kommandos in einer Stapelverarbeitungsdatei. Kommando ist durch ein gültiges DCP-Kommando zu ersetzen.

*Jahr* ::= [*Jahrhundert*] *Ziffer* *Ziffer*

*Jahrhundert* ::= **19** | **20**

**JOIN** [[*Verknüpfung*] | [*Zurücksetzen*]]

Verknüpfen von Verzeichnissen.

*Joker* ::= *Jokerzeichen* {*Jokerzeichen*}

*Jokerbasis* ::= \* | *Joker* | *Basisname*

*Jokername* ::= *Jokerbasis* [*Jokertyp*]

*Jokertyp* ::= \* | *Joker* | *Typname*

*Jokerzeichen* ::= *Namenszeichen* | ?

**KEYBGD**

Aktivierung eines Tastaturprogramms.

*Kommando\_Zeichenkette* ::= *Zeichenkette* = [Umgebung]

*Kommentar* ::= *Zeichenkette*

*Kopieren\_gleiche\_Dateibezeichnung* ::= [**/A**] [**/B**] *Quellspezifikation*

[**/A**] [**/B**] [**/V**] [*Laufwerk*] [*Pfad*]

Schalter:

**/A** Text (ASCII-Dateien) kopieren – das Datenendekennzeichen wird nach dem Kopieren hinzugefügt, bzw. bis dorthin wird kopiert.



**PATH** [*Suchpfadliste*] | ;

. Festlegen Suchpfad.

*Pfad* ::= \Verzeichnis { \Verzeichnis }

*Position* ::= Zahl

**PRINT** [/D:Drucker] [/B:Puffergröße] [/U:Pulsanzahl] [/M:Max.\_Pulsanzahl]  
 [/S:Zeitscheibe] [/Q:Warteschlangengröße]  
 [/C] [/T] [/P] [Quellspezifikation] {Quellspezifikation}

Ausdrucken von Dateien im Hintergrund. Schalter:

/D: Kennzeichnet den Drucker.

/B: Zur Angabe der Puffergröße.

/U: Zur Angabe der Anzahl der Taktgeberimpulse, die PRINT erwartet.

/M: Zur Angabe der Anzahl der Taktgeberimpulse, die PRINT zum Ausdrucken zur Verfügung stehen.

/S: Angabe des Zeitumfangs für Ausführung des PRINT-Befehls.

/Q: Angabe der Anzahl der Druckeinheiten, die sich in der Warteschlange befinden können.

/C Einstellung Löschmodus.

/T Beendigungsmodus für sämtliche in der Warteschlange stehenden Dateien.

/P Einstellung Druckmodus.

**PROMPT** [Zeichenkette] [Meta\_Zeichenfolge]

DCP-Systemabfrage verändern.

*Prüfung* ::= **ON** | **OFF**

Schalter:

**ON** Prüfung ein.

**OFF** Prüfung aus.

*Pulsanzahl* ::= Zahl

*Puffergröße* ::= Zahl

*Quelllaufwerk* ::= **A:** | **B:**

*Quelle* ::= Quellspezifikation | Laufwerk Pfad | Laufwerk\_Pfad

*Quellspezifikation* ::= Dateispezifikation | [Laufwerk] {Pfad} [Jokername]

**RECOVER** Dateispezifikation | Laufwerk

Wiederherstellen von Disketten oder Dateien.

**REM** [Kommentar]

Anzeige von Kommentar auf dem Bildschirm bei der Abarbeitung einer Stapelverarbeitungsdatei.

**RENAME** Dateispezifikation\_alt Dateiname\_neu

Ändern der Dateibezeichnung.

(Für **RENAME** ist **REN** möglich.)

**REPLACE** Quellspezifikation [Laufwerk Pfad] [/A] [/P] [/R] [/S] [/W]

Ersetzen von Dateien.

Schalter:

- /A** Alle angegebenen Dateibezeichnungen werden angefügt.
- /S** Alle angegebenen Dateibezeichnungen werden ersetzt, wenn sie bereits vorhanden sind.
- /R** Es werden auch Dateien mit dem Attribut »Nur Lesen« ersetzt.
- /P** Dateien werden einzeln abgearbeitet.
- /W** Es werden zusätzliche Eingaben verlangt.

*Reservierte\_Einheiten* ::= **AUX** | **CON** | **LPT1** | **LPT2** | **LPT3**

**RESTORE** *Quellspezifikation* [**/S**] [**/P**]

Rückspeichern von Sicherungsdateien. Schalter:

- /S** Bei Rückspeichern werden alle Dateien angezeigt.
- /P** Auswahl, welche Dateien vor dem Rückspeichern angezeigt werden sollen.

**RMDIR** [*Laufwerk*] *Pfad*

Löschen von Unterverzeichnissen.

(Für **RMDIR** ist **RD** möglich.)

*Satz* ::= *Quellspezifikation* | *Satz\_1* | *Zeichenkette*

*Satz\_1* ::= *Dateispezifikation* {*Dateispezifikation*}

*Sekunde* ::= **00** | **01** | .. | **59**

**SELECT** [[*Quellaufwerk*] *Zielaufwerk* [*Pfad*]] **037 GD**

Installieren von Systemdisketten.

**SET** [*Kommando\_Zeichenkette*]

Einfügen einer Zeichenkette in die Kommandoprozessorumgebung.

**SHARE** [**/F**:*Dateigröße*] [**/L**:*Sperren*]

Unterstützung Dateizugriff/Diskettenaustausch. Schalter:

- /F** Zur Angabe der Dateigröße in Byte.
- /L** Zur Angabe der Anzahl gewünschter Sperren.

**SHIFT**

Änderung der Zuordnung Kommandozeilenparameter – Scheinparameter innerhalb einer Stapelverarbeitungsdatei.

**SORT** [**/R**] [**/+** *Position*]

Sortieren von Dateien. Schalter:

- /R** Sortierung in absteigender Folge.
- /+** Sortierung ab *Position*.

*Sperren* ::= *Zahl*

*Spezialsymbol* ::= **!** | **\$** | **\$** | **%** | **&** | **(** | **)** | **\_** | **-** | **{** | **}**

*Spezialzeichen* ::= **!** | **\$** | **#** | **\$** | **%** | **&** | **(** | **)** | **-** | **{** | **}** | **'** | **`**

*Sprungziel* ::= *Zeichenkette*

*Spuren* ::= **40** | **80**

*Status* ::= **+R** | **-R**

Schalter:

- +R** R/O-Status setzen.
- R** R/O-Status aufheben.

*Stunde* ::= 00 | 01 | .. | 23

*Stoppbit* ::= 1 | 2

**SUBST** [*Substitution*] | [*Zurücksetzen*]

Substituierung Laufwerk/Pfad.

*Substitution* ::= *Laufwerk* [*Laufwerk*] *Pfad*

*Suchpfadliste* ::= [*Laufwerk*] *Pfad* {;[*Laufwerk*] *Pfad*}

**SYS** *Laufwerk*

Übertragen der Betriebssystemdateien.

*Tag* ::= 00 | 01 | .. | 31

**TIME** [*Stunde*[.*Minute*[.*Sekunde*[,*Hundertstel\_Sekunde*]]]]

Eingabe und Anzeige der Systemzeit.

**TREE** [*Laufwerk*] [/F]

Angabe der Zugriffspfade. Schalter:

/F Anzeige der Dateinamen in den Unterverzeichnissen.

*Trennzeichen* ::= . | - | /

**TYPE** *Quellspezifikation*

Anzeige des Dateiinhalts.

*Typname* ::= *Namenszeichen*

*Umgebung* ::= *Pfad* | *Zeichenkette*

*Umgebungsgröße* ::= 160 | 161 | .. | 32768

*Umleitung* ::= *Paralldrucker*[:]=*Com*

*Unterbrechung* ::= ON | OFF

Schalter:

ON Programm kann jederzeit unterbrochen werden.

OFF Programm kann nur bei E/A-Operationen unterbrochen werden.

*Variable* ::= *Namenszeichen*

**VER**

Anzeige Versionsnummer.

*Verketteten* ::= *Dateispezifikation\_1* [/A] [/B]

{+*Dateispezifikation\_2* [/A] [/B]}

[*Dateispezifikation\_Ziel*] [/A] [/B] [/V]

Schalter:

1. Vor/hinter der ersten Dateispezifikation bzw. hinter einer nachfolgenden Quell-datei-angabe.

/A Kopiert Datei bis zum ersten \$1A.

/B Kopiert Datei laut Dateigröße im Dateiverzeichnis.

2. Hinter der zweiten Zieldatei-angabe.

/A Anfügen eines Dateiendekennzeichens (\$1A) an die Zieldatei.

/B Kein Anfügen eines Dateiendekennzeichens (\$1A) an die Zieldatei.

3. Hinter der ersten Dateispezifikation.

+ Verkettung von Dateien.

4. Hinter der letzten Dateispezifikation.

**/V** Überprüfung der auf die Zieldiskette geschriebenen Sektoren.

*Verknüpfung* ::= *Laufwerk Laufwerk Pfad*

**VERIFY** [*Prüfung*]

Kontrolllesen ein/aus.

*Verzeichnis* ::= *Zeichenkette*

**VOL** [*Laufwerk*]

Anzeige Disketten-/Plattenname.

*Von* ::= *Dateispezifikation | Reservierte\_Einheiten*

*Warteschlangengröße* ::= **1 | 2 | .. | 32**

**XCOPY** *Quelle Ziel*

**/A** **/D:Datum** **/E** **/M** **/P** **/S** **/V** **/W**

Kopieren von Gruppen von Dateien. Schalter:

**/A** Kopieren von Dateien mit Dateiänderungsattribut = 1.

**/D:** Kopieren von Dateien mit genau dem oder einem späteren Datum.

**/E** Auf die Zieleinheit werden auch leere Unterverzeichnisse kopiert.

**/M** Kopiert Dateien mit Dateiänderungsattribut = 1.

**/P** Vor Kopieren erfolgt eine Abfrage, ob die Datei kopiert werden soll.

**/S** Es werden alle Dateien aus dem Quelldateiverzeichnis kopiert.

**/V** Lesekontrolle aller geschriebenen Daten.

**/W** Zusätzliche Quittierung vor Kopieren.

*Zahl* ::= *Ziffer {Ziffer}*

*Zeichen* ::= *Buchstabe | Ziffer*

*Zeichenkette* ::= *Zeichen {Zeichen}*

*Zeichenkette\_1* ::= *Zeichenkette*

*Zeichenkette\_2* ::= *Zeichenkette*

*Zeile* ::= **80 | 132**

*Zeit* ::= *Stunde:Minute:Sekunde*

*Zeitscheibe* ::= **1 | 2 | .. | 225**

*Ziel* ::= [*Pfad*] *Dateiname*

*Zielfdateispezifikation* ::= *Quellspezifikation*

*Ziellaufwerk* ::= *Laufwerk*

*Ziffer* ::= **1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0**

*Zoll* ::= **6 | 8**

*Zurücksetzen* ::= *Laufwerk /D*

Schalter:

**/D** Auflösen der Verknüpfung/Substitution.



# Anhang C

## 1. Kommandos des TABCALC für das Betriebssystem SCP 1700

Anmerkung: Nichtterminalsymbole, die nur einmal auftreten, werden nicht in der alphabetischen Folge, sondern beim jeweiligen Kommando definiert. Das Auslösen der Tastaturfunktion CR wird nicht dargestellt. Die Kommunikation erfolgt auf der Grundlage der Menütechnik. Hierbei werden Kommandos und Optionen am Bildschirm angezeigt.

**ARRANGE** *Auswahl Bereich1 Zusatz* [[*Abschnitt Richtung Korrektur Zusatz*]  
[*Bereich3 Richtung*]]

Zeilen- und spaltenweises Sortieren nach Feldinhalten; Vor- und Rückwärtssortieren mit und ohne Korrektur der Bezugnahmen in Formeln.

*Abschnitt* ::= *Spaltenbereich* | *Zeilenbereich*

*Richtung* ::= **V** | **R**

**V** Vorwärts: Vorwärtssortieren.

**R** Rückwärts: Rückwärtssortieren.

*Korrektur* ::= **O** | **K**

**O** Ohne Korrektur: keine Formelanpassung.

**K** Korrektur: Formelanpassung erfolgt.

*Bereich3* ::= *2.Zeile* | *2.Spalte*

*2.Zeile*: Eingabe 2. Sortierzeile.

*2.Spalte*: Eingabe 2. Sortierspalte.

*Auswahl* ::= **Z** | **S**

**Z** Zeile

**S** Spalte

*Basisname* : Festlegung laut SCP 1700-Konvention.

*Bereich* ::= *Zeilenabschnitt* | *Spaltenabschnitt* | *Block* | *Feld* | *gesamte\_Tabelle* | *Bereich2*

*Bereich1* ::= *Zeile* | *Spalte*

*Bereich2* ::= *Zeilenbereich* | *Spaltenbereich* | *Bereich1*

**BLANK** *Bereich*

Löschen ungeschützter Bereichsinhalte.

*Block* ::= *Feld1:Feld2*

*Feld1* : linkes oberes *Feld* in einem Tabellenausschnitt.

*Feld2* : rechtes unteres *Feld* in einem Tabellenausschnitt.

**COPY** *Bereich Quelle Feld Ziel ,Menü\_Formelanpassung.*

Kopieren des Inhalts eines Bereichs in einen anderen.

*Dateiname ::= [Laufwerk:] Basisname [.Typname]*

**DELETE** *Auswahl Bereich2*

Löschen einer oder mehrerer Zeilen oder Spalten.

**EDIT** *Feld [Dateieingabe]*

Anzeigen und Belegen eines nicht geschützten, nicht verborgenen Feldes.

*Dateneingabe ::= Text | Wiederholtext | Zahl | Formel*

*Text:* Zeichenketten; Eingabe mit " (Anführungszeichen) einleiten; Textkonstante möglich.

*Wiederholtext:* Zeichenketten zur horizontalen Gliederung der Tabelle. Eingabe mit ' einleiten.

*Zahl:* Zahlen in Dezimal- und Exponentialschreibweise mit oder ohne Vorzeichen, Komma, Dezimalpunkt und Leerstellen.

*Formel:* Komponenten sind Vorzeichen, Zahlen, Feldadressen und Funktionen, die durch arithmetische Operatoren verbunden sind.

*Feld ::= A1 | .. | BK255*

**FORMAT** *Formatabschnitt Formatposition Formatart*

Einstellen der Anzeigeformate für einen Bereich.

*Formatabschnitt ::= G | S | Z | E | D*

**G** Global: Alle noch nicht spezifizierten Felder.

**S** Spalte: Spalte oder Spaltenbereich.

**Z** Zeile: Zeile oder Zeilenbereich.

**E** Einzel: Feld oder Gruppe von Feldern.

**D** Definieren: Anwenderformate 1..8

*Formatposition ::= Bereich | Menü*

*Menü ::= J | N | 0 | .. | 7*

**J | N** für Währungszeichen

Wert in Prozent

Tausendertrenner

Minuswert in ( )

Nullen als Leerraum

Währungszeichen: M

**0 | .. | 7** für Dezimalstellen

Skalenfaktor

*Formatart ::= F S N I E Z G A V R L TR TL B*

Anmerkung: Ein Teil der Optionen ist kombinierbar, andere schließen sich gegenseitig aus. Es existieren viele Kombinationen.

**F** Fertig: Eingegebene Formate werden gültig.

**S** Standard: Rücksetzen aller Formatierungen auf den Standard (N,R,TL,9).

**N** Normal: Zahlenwerte in günstigster Darstellungsart.

**I** Integer: Anzeige des ganzzahligen Zahlenwerts.

**E** Exponent: Anzeige in Exponentialschreibweise.

**Z** Zwei Dezimalstellen: Anzeige 2 Stellen nach dem Dezimalkomma.

**G** Grafik: Wertanzeige als Balkengrafik.

- A** Anwenderdefinition: Anzeige in besonders vereinbarten Formaten:  
1 | .. | 8
- V** Verbergen: Feldinhalt wird nicht angezeigt.
- R** Rechts: Formelergebnisse, Datumsangaben und Textkonstanten in Feldern rechtsbündig.
- L** Links: wie oben, linksbündig.
- TR** Text rechtsbündig.
- TL** Text linksbündig.
- B** Breite: Festlegung der Spaltenbreite.

*gesamte\_Tabelle* ::= **ALL**

#### **GLOBAL** *Regelart*

Wechsel globaler Regeln für Ausgabe, Berechnung und Cursorbewegung.

*Regelart* ::= **F | W | Z | S | M | A | B | U | W**

**F/W** Formel/Wert: Schalter für Formel- und Wertanzeige.

**Z** Zeilenweise: Zeilenweise Formelberechnung.

**S** Spaltenweise: Spaltenweise Formelberechnung.

**M** Manuell berechnen: Automatische Neuberechnung wird durch manuell ausgelöste ersetzt.

**A** Automatisch berechnen: Automatische Berechnung nach jeder Eingabe.

**B** Begrenzung: Ein- und Ausschalten der Zeilen- und Spaltenbezeichnungen.

**U** Überspringen: Überspringen leerer und geschützter Felder (ein/aus).

**W** Weiter: Ab- und Einschalten der automatischen Cursorbewegung nach Betätigung von CR.

#### **INSERT** *Auswahl Bereich2*

Einfügen leerer Zeile(n)/Spalte(n) vor der spezifizierten Zeile/Spalte.

#### **JOB** *Dateiname | Leer Menü\_Dateiverzeichnis*

Laden und Ausführen einer Kommandodatei vom Typ .XQT.

#### **KILL** *Killoperation | Leer Menü\_Dateiverzeichnis [Killoperation]*

Löschen einer Datei auf Diskette.

*Killoperation* ::= *Dateiname* Löschen

*Löschen* ::= **N | J**

**N** Nein: Kein Löschen, Rückkehr zu TABCALC.

**J** Ja: Löschen.

*Laufwerk* ::= **A | B | C | D**

*Leer*: Keine Eingabe, nach Anschlag CR erscheint *Menü\_Dateiverzeichnis*.

#### **LOAD** *Loadoperation | Leer Menü\_Dateiverzeichnis [Loadoperation]*

Vollständiges oder teilweises Laden einer TCT-Datei von der Diskette auf den Bildschirm.

*Loadoperation* ::= *Dateiname* *Ladeparameter* [*Bereich Quelle Feld Ziel Zusatz*]

[*Menü\_Formelanpassung*]

*Ladeparameter* ::= **A | T | W | S**

- A** Alles: Laden der gesamten Tabelle mit Formateinstellungen.
- T** Teilweise: Laden eines Tabellenausschnitts an beliebige Positionen der aktuellen Tabelle.
- W** Werte: Laden nur numerischer Werte sowie Texte.
- S** Summieren: Numerische Werte der geladenen Felder werden zu denen der aktuellen Tabelle addiert.

*Menü\_Dateiname* ::= **N** | **B** | **U**

Menü für den Fall, daß der eingegebene Dateiname bereits existiert.

- N** Name ändern: Eingabe eines neuen *Dateinamens*.
- B** Back-up-Kopie: Bereits vorhandene Datei erhält den Typ BAK.
- U** Überschreiben: Löschen der Vorgängerdatei.

*Menü\_Dateiverzeichnis* ::= **A** | **N** | **L** | **D**

Menü zur Ausgabe von Dateiverzeichnissen verschiedener Laufwerke.

- A** Alle Dateien: Anzeige aller Dateinamen des Dateiverzeichnisses des eingestellten Laufwerkes.
- N** Nur TCT-Dateien: Anzeige der TCT-Dateinamen mit Angabe der TABCALC-Version und des Tabellentitels.
- L** Laufwerk ändern: Eingabe *Laufwerk*
- D** Dateinamen eingeben: Rückkehr zur Eingabe *Dateinamen*

*Menü\_Formelanpassung* ::= **K** | **O** | **E** | **W** | + | - | \* | /

Menü zur Beeinflussung von Kopierfunktionen.

- K** Korrektur: Automatische Formelanpassung.
- O** Ohne Korrektur: Formeln unverändert.
- E** Einzelne Korrektur: Individuelle Korrektur jeder Formelkomponente (J/N).
- W** Werte: Übernahme nur der Werte.
- +** Addition Wert des Quellfeldes zum Wert des Zielfeldes. Summe steht im Zielfeld.
- Subtraktion Wert des Quellfeldes von dem des Zielfeldes. Differenz steht im Zielfeld.
- \*** Multiplikation Wert Quellfeld mit Wert Zielfeld. Produkt steht im Zielfeld.
- /** Division Wert Zielfeld durch Wert Quellfeld. Quotient steht im Zielfeld.

**MOVE** *Auswahl Bereich2 Quelle Bereich1 Ziel Menü\_Formelanpassung*

Verschieben von Zeilen und Spalten vor oder hinter die angegebene Zeile oder Spalte. Anmerkung: Arithmetische Operationen im *Menü\_Formelanpassung* sind nicht möglich.

**OUTPUT** *Ausgabeinhalt Bereich Ausgabemedium* [[*Dateiname*] |  
*Leer Menü\_Dateiverzeichnis*]

Ausgabe der Tabelle auf Drucker, Platte oder Bildschirm.

*Ausgabeinhalt* ::= **T** | **F**

- T** Tabellenform: Kopie der Tabelle vom Bildschirm.
- F** Feldinhalte: Auflisten der Feldinhalte.

*Ausgabemedium* ::= **D** | **B** | **P** | **A**

- D** Drucker: Ausgabe auf Drucker entsprechend Standard-Druckparametern.
- B** Bildschirm: Ausgabe auf Bildschirm.

- P** Platte (Diskette): Ausgabe als PRN-Datei; nicht durch TABCALC lesbar; Eingabe Dateiname erforderlich. Abruf Dateiverzeichnis möglich.
- A** Ausgabeformat: Änderung Standard-Druckparameter.
  - Seitenlänge
  - Zeilenlänge
  - Automatischer Papiervorschub
  - Doppelte Breite
  - Linker Rand

**PROTECT** *Bereich*

Schützen vor unbeabsichtigten Veränderungen.

*Quelle* ::= Positionsangaben der zu kopierenden Daten des Ursprungsbereiches.

**QUIT** *Ende* [*Speichermenü*] [*Dateiname*] | [*Leer Menü*\_Dateiverzeichnis]

Verlassen von TABCALC; Rückkehr in das rufende Programm.

*Ende* ::= **J** | **N**

**J** Ja: Endebestätigung wird verlangt.

*Speichermenü* erscheint anschließend.

**N** Nein: Rückkehr zu TABCALC.

*Speichermenü* ::= **A** | **E** | **K**

**A** Alle: Speichert alle Tabellen neu auf Diskette: Eingabe *Dateiname* oder Abforderung *Menü*\_Dateiverzeichnis.

**E** Einzelne: Abfrage bei jeder Tabelle, ob gespeichert werden soll: Eingabe *Dateiname* oder Abforderung *Menü*\_Dateiverzeichnis.

**K** Keine: TABCALC wird ohne Speicherung einer Tabelle verlassen.

**REPLICATE** *Bereich3* *Quelle* *Bereich4* *Ziel* *Menü*\_Formelanpassung

Wiederholtes Kopieren von Einzelfeldern, Zeilen- oder Spaltenabschnitten – keine Blöcke!

*Bereich3* ::= *Feld* | *Zeilenabschnitt* | *Spaltenabschnitt*

*Bereich4* ::= *Spaltenabschnitt* | *Zeilenabschnitt*

**SAVE** *Saveoperation* | *Leer Menü*\_Dateiverzeichnis [*Saveoperation*]

Vollständiges oder teilweises Speichern einer Tabelle oder eines Bereiches auf Diskette.

*Saveoperation* ::= *Dateiname* [*Menü*\_Dateiname] *Speichermenü1* [*Bereich* *Speichermenü2*]

*Speichermenü1* ::= **A** | **W** | **T**

**A** Alles: Werte, Formeln, Texte und Formate der gesamten Tabelle werden gespeichert.

**W** Werte: Gesamte Tabelle wird ohne Formeln gespeichert.

**T** Teilweise: **A** oder **W** werden für einen zu definierenden Bereich gespeichert.

*Speichermenü2* ::= **A** | **W**

**A** Alles: Siehe Speichermenü1.

**W** Werte: Siehe Speichermenü1.

*Spalte* ::= **A** | **B** | .. | **Z** | **AA** | .. | **AZ** | **BA** | .. | **BK**

*Spaltenabschnitt* ::= *Spalte\_Zeile:Spalte\_Zeile*

Bedingung: Spaltenbuchstaben müssen übereinstimmen.

*Spaltenbereich* ::= *Spalte:Spalte*

*Spalte\_Zeile* ::= *Feld*

#### **TITLE** *Fixierbereich*

Ständiges Anzeigen spezifizierter Zeilen und/oder Spalten auf dem Bildschirm.

*Fixierbereich* ::= **V** | **H** | **B** | **L**

**V** Vertikal: Fixiert die aktuelle Spalte und alle Spalten links davon.

**H** Horizontal: Fixiert die aktuelle Zeile und alle Zeilen darüber.

**B** Beides: Fixiert die horizontale und vertikale Anzeige.

**L** Löschen: Rückgängigmachen der Festlegungen.

*Typname*: Festlegung laut SCP 1700-Konvention

Standard sind: **TCT** | **PRN** | **XQT**

**TCT** TABCALC-Tabellendatei.

**PRN** Ausgabeprotokolldatei.

**XQT** Befehlsdatei.

#### **UNPROTECT** *Bereich*

Aufheben des Änderungsschutzes.

#### **WINDOW** *Fenstertyp*

Horizontales und vertikales Teilen des Bildschirmes in zwei Fenster, die synchron oder ungekoppelt rollen können.

*Fenstertyp* ::= **H** | **V** | **L** | **S** | **N**

**H** Horizontal: Bildschirmteilung entlang der aktuellen Zeile in 2 übereinanderliegende Fenster.

**V** Vertikal: Bildschirmteilung entlang der aktuellen Spalte in 2 nebeneinanderliegende Fenster.

**L** Löschen: Aufheben der Bildschirmteilung.

**S** Synchron: Synchrones Rollen beider Fenster.

**N** Nicht synchron: Keine Kopplung der Bildschirmfenster.

#### **ZAP** *Löschabfrage* [[*Speicherabfrage*] [*Dateiname*] ]

[*Leer Menü\_Dateiverzeichnis*]]

Löschen der aktuellen Tabelle. Es erscheint ein leeres Arbeitsblatt.

TABCALC bleibt aktiv.

*Löschabfrage* ::= **J** | **N**

**J** Ja: Aktuelle Tabelle wird zum Löschen vorbereitet.

**N** Nein: ZAP-Kommando wird verlassen.

#### *Speicherabfrage* ::= **J** | **N**

**J** Ja: Aktuelle Tabelle speichern; Eingabe *Dateiname* oder Ausgabe *Menü\_Dateiverzeichnis*.

**N** Nein: Tabelle wird ohne zu speichern gelöscht.

*Zeile* ::= **1** | .. | **255**

*Zeilenabschnitt* ::= *Spalte\_Zeile:Spalte\_Zeile*

Bedingung: Zeilenzahlen müssen übereinstimmen.

*Zeilenbereich* ::= *Zeile:Zeile*

*Ziel* : Positionsangabe des Empfangsbereiches der zu kopierenden Daten.

*Zusatz* ::= **J** | **N**

Menü zur Inanspruchnahme weiterer Optionen eines Kommandos.

**J** Ja: Fragestellung kommandoabhängig.

**N** Nein: Fragestellung kommandoabhängig.

## 2. Funktionen (ohne Finanzfunktionen)

<b>ABS</b> ( <i>Wert</i> )	Absolutbetrag
<b>ACOS</b> ( <i>Wert</i> )	Arkuskosinus-Funktion
<b>AND</b> ( <i>Wert1</i> , <i>Wert2</i> )	logisches »UND«
<b>ASIN</b> ( <i>Wert</i> )	Arkussinus-Funktion
<b>ATAN</b> ( <i>Wert</i> )	Arkustangens-Funktion
<b>AVERAGE</b> ( <i>Liste</i> )   <b>AV</b> ( <i>Liste</i> )	Arithmetisches Mittel
<b>COS</b> ( <i>Wert</i> )	Kosinusfunktion
<b>COUNT</b> ( <i>Liste</i> )	Anzahl der Listenelemente (<) 0
<b>DATE</b> ( <i>TT</i> , <i>MM</i> , [ <i>JJ</i> ] <i>JJ</i> )	Datumseingabe
<b>DAT</b> ( <i>TT</i> , <i>MM</i> , [ <i>JJ</i> ] <i>JJ</i> )	
<b>DAY</b> ( <i>Datumswert</i> )	Tag aus einem Datum
<b>DVAL</b> ( <i>Wert</i> )	Umwandlung in Datum (TT.MM.[ <i>JJ</i> ] <i>JJ</i> )
<b>ERROR</b>   <b>ERR</b>	Anzeige/Berechnung unmöglich
<b>EXP</b> ( <i>Wert</i> )	Exponentialfunktion
<b>FALSE</b>	logische Konstante, numerisch 0
<b>IF</b> ( <i>Ausdruck</i> , <i>Wert1</i> , <i>Wert2</i> )	bedingte Zuweisung
<b>INT</b> ( <i>Wert</i> )	ganzzahliger Teil des Werts
<b>ISDATE</b> ( <i>Wert</i> )	Test auf Datentyp »Datum«
<b>ISERROR</b> ( <i>Wert</i> )   <b>ISERR</b> ( <i>Wert</i> )	Test auf Datentyp »ERROR«
<b>ISNA</b> ( <i>Wert</i> )	Test auf Datentyp »N/A«
<b>ISNUM</b> ( <i>Wert</i> )	Test auf Datentyp »Numerisch«
<b>ISTEXT</b> ( <i>Wert</i> )	Test auf Datentyp »Text«
<b>JDATE</b> ( <i>Datumswert</i> )	modifiziertes, Julianisches Datum
<b>LN</b> ( <i>Wert</i> )	natürlicher Logarithmus
<b>LOG10</b> ( <i>Wert</i> )	dekadischer Logarithmus
<b>LOOKUP</b> ( <i>Wert</i> , <i>Spaltenabschnitt</i> )	Suchfunktion
<b>LOOKUP</b> ( <i>Wert</i> , <i>Zeilenabschnitt</i> )	
<b>MAX</b> ( <i>Liste</i> )	größtes Listenelement
<b>MIN</b> ( <i>Liste</i> )	kleinstes Listenelement
<b>MOD</b> ( <i>Wert1</i> , <i>Wert2</i> )	Rest der Division
<b>MONTH</b> ( <i>Datumswert</i> )	Monat aus einem Datum
<b>NA</b>	Wert nicht verfügbar
<b>NOT</b> ( <i>Wert</i> )	logische Negation
<b>OR</b> ( <i>Wert1</i> , <i>Wert2</i> )	logisches »ODER«
<b>PI</b>	Kreiskonstante 3,1415926..
<b>ROUND</b> ( <i>Wert1</i> , <i>Wert2</i> )	Rundung des Werts

<b>SIN</b> ( <i>Wert</i> )	Sinusfunktion
<b>SQRT</b> ( <i>Wert</i> )	Quadratwurzel
<b>SUM</b> ( <i>Liste</i> )	Summe der Listenelemente
<b>TAN</b> ( <i>Wert</i> )	Tangensfunktion
<b>TODAY</b>	heutiges Datum
<b>TRUE</b>	logische Konstante, numerisch 1
<b>WDAY</b> ( <i>Datumswert</i> )	Wochentag aus einem Datum
<b>YEAR</b> ( <i>Datumswert</i> )	Jahreszahl aus einem Datum
Es bedeuten:	
( <i>Wert</i> )   ( <i>Wert1, Wert2</i> [, ..., <i>Wertn</i> ])	eine Konstante, eine Feldadresse, eine Funktion, eine Rechen- oder Vergleichsoperation.
( <i>Liste</i> )	Aufzählung mehrerer Parameter vom Typ Wert, Feldadresse und Bereich.
( <i>Datumswert</i> )	Ergebnisse der DATE-, DVAL- oder TODAY-Funktion.



# Anhang D

## Befehlsübersicht REDABAS-3

### 1. Befehle

**ACCEPT** [*Text*] **TO** *Speichervariable*

Anzeige einer Eingabeaufforderung an den Nutzer. Die folgende Eingabe wird als Zeichenkette auf dem spezifischen Speicherplatz gespeichert.

*Alphanumerischer\_Ausdruck* ::= Ausdruck

**APPEND** [**BLANK**]

Anfügen eines Datensatzes. Die Option BLANK erlaubt das Anfügen eines leeren Datensatzes.

**APPEND FROM** *Basisname\_der\_Datenbankdatei* [**FOR** *Bedingung* | **WHILE** *Bedingung*]  
[**SDF** | **DELIMITED** [**WITH BLANK** | **WITH** *Begrenzung*]]

Entnahme von Datensätzen aus der spezifizierten Datei und Anhängen an die eröffnete Datei. Über die FOR | WHILE-Klausel kann die Entnahme selektiv erfolgen. Die Option SDF (System-Daten-Format) unterstellt, daß die Datensätze das ASCII-Format haben. DELIMITED wird benutzt, wenn die zu kopierenden Datensätze Begrenzungszeichen zwischen den Datenfeldern haben sollen. Das Begrenzungszeichen kann durch den Nutzer festgelegt werden.

*Arbeitsbereich* ::= Hauptspeicherbereich zur Aufnahme von Datenbankdateien.

Arbeitsbereiche sind **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10**

**ASSIST**

Umschaltung in ein menügesteuertes Datenbanksystem. Im oberen Teil des Bildschirms werden Kommandogruppen über Aufbau, Modus, Position, Extrakt, Planung und Dienste zur menügesteuerten Nutzung angeboten.

*Ausdruck* ::= Verknüpfung von Feldvariablen, Speichervariablen, Konstanten, Funktionen und Operatoren. Ausdrücke können vom Typ Numerisch, Alphanumerisch, Logisch und Datum sein.

*Ausdrucksliste* ::= *Ausdruck* {, *Ausdruck*}

**AVERAGE** [*Ausdrucksliste*][*Bereich*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]  
[**TO** *Speichervariablenliste*]

Berechnung des arithmetischen Mittels von numerischen Feldern über ausgewählte Datensätze. Das Ergebnis kann über die TO-Option auf temporären Speicherplätzen abgespeichert werden.

*Basisname* ::= Name einer Dateiart ohne Typdeklaration

*Bedingung* ::= Auswahlkriterium für Datenverwaltung, Dateizugriff und Programmierung.

*Befehl* ::= REDABAS-Befehl.

*Befehlsdatei* ::= Datei aus REDABAS-Befehlen.

*Begrenzung* ::= Zeichen zur Abgrenzung von Feldelementen bei der Datenübernahme.

*Bereich* ::= Wirkungsbereich von Befehlen. Bereichsangaben sind

**ALL** | **NEXT** *Zahl* | **RECORD** *Zahl*

**BROWSE** [**FIELDS** *Feldliste*]

Gleichzeitiges Editieren von max. 17 Datensätzen. Feldauswahlmöglichkeiten über die Option **FIELDS**.

**CANCEL**

Abbruch der Ausführung einer Kommandodatei und Rückkehr auf die Steuerebene des Datenbankbetriebssystems.

**CHANGE** [*Bereich*] [**FIELDS** *Feldliste*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]

Aufruf zum menügesteuerten Ändern eines oder mehrerer Datenfelder.

**CLEAR**

Löschen des Bildschirms und Positionieren des Cursors in die linke obere Ecke des Bildschirms.

**CLEAR ALL**

Schließen aller Datenbankdateien, Löschen aller temporären Speicherplätze und Anwählen des Arbeitsbereiches 1 (SELECT 1).

**CLEAR GETS**

Schließen der mit GET zuvor aufgerufenen Feldvariablen. Der folgende READ-Befehl ignoriert die editierten Werte.

**CLEAR MEMORY**

Löschen aller Speichervariablen (vgl. **RELEASE ALL**).

**CLOSE** *Dateibezeichnung*

Explizites Schließen der unter Dateibezeichnung spezifizierten Dateiart.

**CONTINUE**

Aufforderung zum Suchen eines weiteren Datensatzes nach einem vorausgegangenen **LOCATE**-Kommando.

**COPY TO** *Basisname\_der\_Datenbankdatei* [*Bereich*] [**FIELDS** *Feldliste*]

[**FOR** *Bedingung* | **WHILE** *Bedingung*] [**SDF** | **DELIMITED**

[**WITH BLANK** | **WITH** *Begrenzung*]]

Kopieren einer Datei. Eine bereits unter diesem Namen existierende Datei wird überschrieben. Mit der Option **FIELDS** kann eine Feldauswahl für das Kopieren spe-

zifiziert werden. Werden SDF oder DELIMITED benutzt, so entsteht eine Datei vom Typ .TXT (ASCII-Format). Bei DELIMITED werden die Felder durch Anführungszeichen getrennt. Als Begrenzung kann über WITH ein anderes Zeichen festgelegt werden. Merk-Felder können nicht mit der Option SDF oder DELIMITED kopiert werden.

**COPY FILE** *Dateiname* **TO** *Dateiname*

Kopierkommando für jede beliebige Datei. Der Dateityp ist anzugeben.

**COPY STRUCTURE TO** *Basisname\_der\_Datenbankdatei* [**FIELDS** *Feldliste*]

Kopieren der Struktur der aktiven Datei. Über die Option FIELDS ist eine Feldauswahl möglich. Eine bereits existierende Datei wird überschrieben.

**COPY TO** *Basisname\_der\_Datenbankdatei* **STRUCTURE EXTENDED**

Kopieren der Strukturbeschreibung der aktiven Datei als Datensätze in die Zieldatei. Pro Datenfeld der Quelldatei wird ein Datensatz angelegt, der den Feldnamen, den Feldtyp, die Feldlänge und die Anzahl der Dezimalstellen enthält.

**COUNT:** [*Bereich*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]  
[**TO** *Speichervariable*]

Zählen von Datensätzen und wahlweises Abspeichern des Ergebnisses.

**CREATE** *Basisname\_der\_Datenbankdatei*

Anlegen einer Datenbankdatei.

**CREATE** *Basisname\_der\_Datenbankdatei* **FROM** *Basisname\_der\_Datenbankdatei*

Erzeugen einer Datei, deren Struktur in einer Strukturdatei beschrieben ist. Der Inhalt der Strukturdatei ist mit Hilfe des Befehls COPY STRUCTURE EXTENDED erstellt worden.

**CREATE LABEL** *Basisname\_der\_Etikettendatei*

Erstellen eines Etikettenformats für die eröffnete Datei. In zwei Editiermasken werden Inhalt und Format des Etiketts festgelegt. Das Kommando ist gleichwertig zu MODIFY LABEL.

**CREATE REPORT** *Basisname\_der\_Reportdatei*

Aufrufen des integrierten Reportgenerators. Das Kommando ist gleichwertig zu MODIFY REPORT.

*Dateibezeichnung* ::= Schlüsselwörter zur Spezifizierung von ausgewählten Dateiarten.  
Schlüsselwörter sind **ALTERNATE** | **DATABASES** | **FORMAT** |  
**INDEX** | **PROCEDURE**

*Dateikurzname* ::= Kurzbezeichnung für Datenbankdateien (ALIAS-Name).

*Dateiname* ::= *Basisname*. *Dateityp*

Vollständige Bezeichnung einer Dateiart einschließlich des Dateityps.

*Dateiname\_alt* ::= *Dateiname*

*Dateiname\_neu* ::= *Dateiname*

*Dateityp* ::= Dateikennzeichnung durch die Erweiterung des Dateinamens. Dateitypen sind: **BAK** | **DBD** | **DBM** | **DEF** | **ETI** | **IDX** | **MSK** | **PRG** | **TXT** | **VAR**

*Datumsvariable* ::= Variable vom Datentyp DATUM.

**DELETE** [*Bereich*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]

Markieren von Datensätzen zum Löschen. Löschmarkierte Datensätze werden mit einem \* vor dem ersten Datenfeld des Datensatzes versehen. Die Voreinstellung für Bereich ist RECORD.

**DELETE FILE** *Dateiname*

Löschen einer nicht geöffneten Datei aus dem Dateiverzeichnis.

*Dezimalstellen* ::= *Zahl*

**DIR** [*Laufwerk*] [*Pfad\*] [*Maske*]

Anzeigen des Diskettenverzeichnisses im spezifizierten oder aktuellen Laufwerk.

**DISPLAY** [**OFF**] [*Bereich*] [**FIELDS** *Feldliste*] [**FOR** *Bedingung* | **WHILE** *Bedingung*] [**TO PRINT**]

Anzeigen von Datensätzen der aktiven Datenbankdatei. Mit der Option OFF kann die Anzeige des Datensatzzählers ausgeblendet werden. Die Bereichsvoreinstellung ist RECORD. Die FOR | WHILE-Option erlaubt selektives Anzeigen. Die Ausgabe wird bei 20 Datensätzen automatisch unterbrochen und kann durch einen beliebigen Tastendruck fortgeführt werden.

**DISPLAY MEMORY** [**TO PRINT**]

Anzeigen oder Drucken von Inhalt, Größe, Typ der verwendeten temporären Speichervariablen, Anzahl der genutzten und freien Speicherplätze (Maximum ist 256 Speicherplätze und 6000 Byte insgesamt).

**DISPLAY STATUS** [**TO PRINT**]

Anzeigen oder Drucken eines Überblicks über den augenblicklichen Systemstatus (Arbeitsbereiche eröffneter Dateien, Status aller SET-Kommandos, Belegung der Funktionstasten).

**DISPLAY STRUCTURE** [**TO PRINT**]

Anzeigen oder Drucken der Struktur der augenblicklich aktiven Datenbankdatei.

**DO** *Basisname\_der\_Befehlsdatei* [**WITH** *Parameterliste*]

Aufrufen eines gespeicherten Programms zur Verarbeitung. Über die WITH-Option können Parameter angegeben werden, die von anderen Programmen bzw. von der Tastatur übernommen werden.

**DO CASE**

**CASE** *Bedingung* *Befehl* {*Befehl*} {**CASE** *Bedingung* *Befehl* {*Befehl*}} [**OTHERWISE** *Befehl* {*Befehl*}]

**ENDCASE**

Fallunterscheidung.

**DO WHILE** *Bedingung* *Befehl* {*Befehl*} **ENDDO**

Anweisung zur Realisierung von Zyklen in Programmen.

**EDIT** [[**RECORD**] *.Zahl*]

Menügesteuertes Editieren von Datensätzen.

**EJECT**

Seitenvorschub bei eingeschaltetem Drucker.

**ERASE** *Dateiname*

Löschen einer geschlossenen Datei aus dem Dateiverzeichnis.

**EXIT**

Abbrechen der Ausführung der Befehle in einer DO WHILE-Schleife und Fortsetzung der Programmabarbeitung mit dem Befehl nach dem zugehörigen ENDDO.

*Externer\_Befehl* ::= Befehle bzw. Befehlsfolgen, die nicht zu REDABAS gehören.

*Feldname* ::= Name eines Datenfeldes in einer Datenbankdatei.

*Feldliste* ::= *Feldname* {, *Feldname*}

**FIND** *Zeichenkette*

Suchen des spezifizierten Datensatzes in einer Indexdatei. Gesucht wird der erste Datensatz, in dessen Indexfeld der angegebene Suchbegriff als Ganzes oder als Teil auftritt.

**GET**

Aufrufen von Feld- und Speichervariablen auf dem Bildschirm, die dort editiert und mit READ wieder eingelesen werden (vgl. @ und READ).

**GO** [RECORD] *Zahl***GO TOP****GO BOTTOM****GO** *Speichervariable*

Positionierbefehle für den Datensatzzähler.

**GOTO** wie **GO****HELP** [*Systemparameter*]

Aufrufen eines Menüs, um den Benutzer über die Systemleistungen und -parameter zu informieren.

**IF** *Bedingung* *Befehl* {*Befehl*} [**ELSE** *Befehl* {*Befehl*}] **ENDIF**

Verzweigungsbefehl.

**INDEX ON** *Ausdruck* **TO** *Basisnahme\_der\_Indexdatei*

Anlegen einer Indexdatei für die eröffnete Datei (Stammdatei). Indiziert wird mit dem angegebenen Ausdruck. Er besteht aus maximal 100 Zeichen.

**INPUT** [*'Text'*] **TO** *Speichervariable*

Eingabe über die Tastatur auf einen temporären Speicherplatz. Im Gegensatz zu ACCEPT können alle Datentypen eingegeben werden. Alphanumerische Daten müssen in Hochkommata eingeschlossen werden.

**INSERT** [BLANK] [BEFORE]

Einfügung eines Datensatzes nach der aktuellen Stelle des Datensatzzählers. Soll vor dieser Stelle eingefügt werden, dann ist die Option BEFORE zu geben. Mit der Option BLANK wird ein leerer Datensatz eingefügt.

**JOIN WITH** *Dateikurzname* **TO** *Basisname\_der\_Datenbankdatei*  
**FOR** *Bedingung* | **WHILE** *Bedingung* [**FIELDS** *Feldliste*]

Die aktive Datei wird mit der mit WITH angesprochenen selektierten Datei zu einer neuen Datei zusammengefügt. Die FOR | WHILE-Klausel enthält die Bedingung für die Vereinigung der Sätze, die FIELDS-Option enthält die Feldauswahl für die neue Datei. Allen Feldern, die aus der zweiten (WITH)-Datei stammen, ist der ALIAS-Name mit dem Zeichen → voranzustellen.

**LABEL FORM** *Basisname\_der\_Formatdatei* [*Bereich*] [**SAMPLE**] [**TO PRINT**]  
 [**FOR** *Bedingung* | **WHILE** *Bedingung*]  
 [**TO FILE** *Basisname\_der\_Textdatei*]

Aufrufen einer mit MODIFY LABEL erzeugten Formatdatei zum Erstellen von Etiketten. Mit den Optionen TO PRINT bzw. TO FILE kann die Ausgabe auf einen Drucker oder in eine Textdatei geleitet werden. Die Bereichsvoreinstellung ist ALL. Die Variante LABEL FORM *Basisname\_der\_Formatdatei* SAMPLE wird benutzt, um einzelne Musteretiketten zu drucken und dabei den Drucker und die Papierführung einzustellen.

*Laufwerk* ::= Zulässige Laufwerksbezeichnung des Betriebssystems.

*Länge* ::= *Zahl*

*Liste* ::= Liste von Feldvariablen, Speichervariablen oder Ausdrücken

**LIST** [**OFF**] [*Bereich*] [*Feldliste*]  
 [**FOR** *Bedingung* | **WHILE** *Bedingung*] [**TO PRINT**]

Zeilenweises Auflisten der Datensätze einer Datei. Der Befehl ist gleichwertig zu DISPLAY, außer der Bereichsvoreinstellung (RECORD bei DISPLAY, ALL bei LIST) und der gruppenweisen Anzeige (20 Sätze bei DISPLAY, alle Sätze bei LIST).

**LIST MEMORY** [**TO PRINT**]

Der Befehl ist identisch zu DISPLAY MEMORY, außer der Anzeige in 20er Gruppen.

**LIST STATUS** [**TO PRINT**]

Der Befehl ist identisch zu DISPLAY STATUS, außer der Pause zwischen den 20er Gruppen.

**LIST STRUCTURE** [**TO PRINT**]

Der Befehl ist identisch zu DISPLAY STRUCTURE, außer der Pause zwischen den 20er Gruppen.

**LOCATE** [*Bereich*] **FOR** *Bedingung* | **WHILE** *Bedingung*

Sequentielles Durchsuchen der aktiveri Datenbankdatei nach jenen Datensätzen, die der gestellten Bedingung entsprechen. NEXT *Zahl* bewirkt, daß das Suchen beim aktuellen Datensatz begonnen wird. Mit LOCATE wird stets nur der erste Satz gefunden. Das weitere Suchen nach Datensätzen erfolgt mit CONTINUE.

**LOOP**

Rückkehr zum Schleifenanfang innerhalb einer DO WHILE-Schleife.

*Maske* ::= Zusammenstellung allgemeiner Ersetzungszeichen für den Datei- und Feldnamen sowie die Form von Datenein- und -ausgaben.

**MODIFY COMMAND** *Basisname\_der\_Befehlsdatei*

Aufruf des Texteditors

**MODIFY LABEL** *Basisname\_der\_Etikettendatei*

Der Befehl ist identisch mit CREATE LABEL. Er kann auch benutzt werden, um Etikettendateien zu editieren.

**MODIFY REPORT** *Basisname\_der\_Reportdatei*

Der Befehl ist identisch mit CREATE REPORT. Er ermöglicht auch das Editieren von Reportdateien.

**MODIFY STRUCTURE** *Basisname\_der\_Datenbankdatei*

Änderung der Struktur einer existierenden Datenbankdatei.

**NOTE** *Text* | \* *Text*

Kommentar in einer Programm- oder Prozedurdatei.

*Numerischer\_Ausdruck* ::= Verknüpfung von Feldvariablen, Speichervariablen, Konstanten, Funktionen und Operatoren vom Typ Numerisch.

**PACK**

Physisches Löschen von Datensätzen, die mit den Befehlen DELETE bzw. CTRL-U im Editorstatus markiert worden sind.

*Parameter* ::= Variable, Konstante, Ausdruck zur Spezifizierung der Befehlsausführung.

Parameterliste ::= *Parameter* {, *Parameter*}

**PARAMETERS** *Parameterliste*

Parameter aus einem übergeordneten Programm werden übernommen und zur Verarbeitung bereitgestellt. Die Daten werden beim Aufruf des Unterprogramms mit DO *Basisname\_der\_Befehlsdatei* WITH *Parameterliste* übergeben.

*Pfad* ::= Festlegung des Weges zum Dateiverzeichnis.

*Pfadliste* ::= *Pfad* {, *Pfad*}

**PRIVATE** [ALL [LIKE *Maske* | EXCEPT *Maske*]] | [*Speichervariablenliste*]

Einschränkung des Zugriffs zu Speichervariablen. Der Zugriff zu privaten Speichervariablen wird nur dem aktuellen Programm erlaubt. Sie werden nach Abschluß der Programmroutine gelöscht.

**PROCEDURE** *Prozedurname*

Beginn eines Programms in einer Prozedurdatei. Das Programmende wird mit RETURN angegeben.

*Prozedurname* ::= Zusammenfassung und Bezeichnung von Unterprogrammen unter einem Namen.

**PUBLIC** *Speichervariablenliste*

Vereinbarung globaler Speichervariablen. Der Befehl PUBLIC muß vor der Initialisierung der Variablen gegeben werden.

**QUIT**

Beenden der Arbeit, Schließen der offenen Dateien und Rückkehr in das Betriebssystem.

**READ**

Wiedereinlesen von Variableninhalten in Programmen. Die Variableninhalte können zuvor über den GET-Befehl angezeigt und editiert werden.

**RECALL** [*Bereich*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]

Reaktivieren von Datensätzen, die mit den Befehlen DELETE bzw. CTRL-U im Editorstatus markiert worden sind. Die Voreinstellung für *Bereich* ist der aktuelle Datensatz.

**REINDEX**

Aktualisieren aktivierter Indexdateien.

**RELEASE** [*Speichervariablenliste*] | [**ALL** [**LIKE** *Maske* | **EXCEPT** *Maske*]]

Löschen von Speichervariablen.

**RENAME** *Dateiname\_alt* **TO** *Dateiname\_neu*

Umbenennen von geschlossenen Dateien. Beim Umbenennen einer Datenbankdatei mit Feldern vom Typ Merk muß die Datenbank-Merkdatei gesondert umbenannt werden.

**REPLACE** [*Bereich*] *Feldname* **WITH** *Ausdruck* [, *Feldname* **WITH** *Ausdruck*] [**FOR** *Bedingung* | **WHILE** *Bedingung*]

Ersetzen des Inhalts von Datenfeldern durch den nach WITH stehenden Ausdruck. Voreingestellt ist der aktuelle Datensatz.

**REPORT FORM** *Basisname\_der\_Reportdatei* [*Bereich*] [**FOR** *Bedingung* | **WHILE** *Bedingung*] [**TO FILE** *Basisname\_der\_Textdatei*] [**TO PRINT**] [**NOJECT**] [**HEADING** '*Text*'] [**PLAIN**]

Abruf von Information aus der aktiven Datenbankdatei, deren Form zuvor durch den Reportgenerator (vgl. CREATE REPORT) zusammengestellt worden ist, auf den Bildschirm, den Drucker oder auf Diskette. Die Bereichsangabe (Voreinstellung = ALL) und die FOR | WHILE-Klausel erlauben selektives Arbeiten. Durch die Option PLAIN werden die Seitennumerierung und die Datumsausgabe unterdrückt. Die Option HEADING erlaubt das Einfügen von zusätzlichen Überschriftenzeilen. Die Option NOJECT unterdrückt einen Seitenvorschub zu Beginn des Ausdrucks.

**RESTORE FROM** *Basisname\_der\_Speichervariablendatei* [**ADDITIVE**]

Einlesen von Speichervariablen, die mit SAVE TO auf einer Speichervariablendatei gesichert worden sind. Bei der Option ADDITIVE ergänzen die eingelesenen Speichervariablen die bereits im Arbeitsspeicher vorhandenen.

**RETURN** [**TO MASTER**]

Beenden eines Programms und Rückgabe der Steuerung an die übergeordnete Ebene. Die Option TO MASTER bewirkt, daß die Steuerung an die höchste Programmebene (z. B. bei tiefverschachtelten Programmenüs) zurückgegeben wird.

**RUN** *Externer\_Befehl* | ! *Externer\_Befehl*

Ausführen eines Kommandos auf Betriebssystemebene. Nach Beendigung des Kommandos ist wieder der REDABAS-Systemstatus vorhanden.



**SAVE TO** *Basisname\_der\_Speichervariablendatei* [**ALL LIKE** *Maske* |  
**EXCEPT** *Maske*]

Abspeichern von Speichervariablen nach dem Vollständigkeits- bzw. Auswahlprinzip.

*Schalter* ::= Ein- und Ausschalten von Systemparametern während des Dialogs. Schalterstellungen sind:

**ON** – Einschalten,

**OFF** – Ausschalten

*Schlüssel* ::= Feldname, der als Ordnungsbegriff fungiert

**SEEK** *Ausdruck*

Aufsuchen von Datensätzen in Indexdateien. Gesucht wird stets der erste Satz, in dessen Schlüsselfeld der Suchausdruck gänzlich oder teilweise enthalten ist.

**SELECT** *Arbeitsbereich*

Auswahl eines von zehn möglichen Arbeitsbereichen für die Datenbankdatei. Beim Systemstart ist stets der erste Bereich angewählt.

**SET**

Aufruf eines menügesteuerten Editors, in dem verschiedene Betriebsparameter eingestellt werden können.

**SET ALTERNATE** *Schalter*

An- und Abschalten der Ausgabe in die mit SET ALTERNATE TO zu erzeugende Textdatei.

**SET ALTERNATE TO** *Basisname\_der\_Textdatei*

Anlegen einer Textdatei, die alle Tastatureingaben und Bildschirmausgaben enthält. Vor diesem Befehl muß die Datei durch SET ALTERNATE TO *Basisname\_der\_Textdatei* eröffnet werden. Die Datei muß nach Abschluß der Verarbeitung mit CLOSE ALTERNATE wieder geschlossen werden. Voreinstellung ist OFF.

**SET BELL** *Schalter*

Ein- bzw. Ausschalten eines Warnsignals zur Kenntlichmachung fehlerhafter Eingaben. Voreinstellung ist ON.

**SET CARRY** *Schalter*

Übernahme von Daten aus dem vorhergehenden Datensatz in den mit APPEND BLANK angehängten bzw. mit INSERT BLANK eingefügten Datensatz. Voreinstellung ist OFF.

**SET COLOR TO**

Menüunterstütztes Auswählen der Bildschirmdarstellung.

**SET CONFIRM** *Schalter*

Automatischer Cursorsprung nach dem Eingeben von Daten in eine Editiermaske. Voreinstellung ist OFF.

**SET CONSOLE** *Schalter*

Ausgabe oder Unterdrückung der Ausgabe auf dem Bildschirm. Voreinstellung ist ON.

**SET DEBUG** *Schalter*

Unterstützung der Fehlersuche in Programmen. Bei ON werden die Ergebnisse der ECHO- und STEP-Funktion auf den Drucker ausgegeben. Voreinstellung ist OFF.

**SET DECIMALS TO** *Numerischer\_Ausdruck*

Angabe der Anzahl anzuzeigender Dezimalstellen.

**SET DEFAULT TO** *Laufwerk*

Zuweisung eines Laufwerks als Standardlaufwerk.

**SET DELETED** *Schalter*

Selektion löschmarkierter Datensätze für die Bearbeitung. Das gilt nicht, wenn die Bereichsangabe mit RECORD bzw. NEXT spezifiziert ist. Voreinstellung ist OFF.

**SET DELIMITER** *Schalter***SET DELIMITER TO** [*'Zeichen'*] [DEFAULT]

Festlegung vom Standard abweichender Begrenzer in Editiermasken. Voreinstellung ist OFF.

**SET DEVICE TO PRINT | SET DEVICE TO SCREEN**

Festlegung des Ausgabegerätes für @-Befehle. Die Voreinstellung ist SCREEN.

**SET ECHO** *Schalter*

Ausgeben der Befehle eines Programms auf den Bildschirm oder auf den Drucker während der Ausführung des Programms. Die Voreinstellung ist OFF.

**SET ESCAPE** *Schalter*

Bestimmen, ob eine Programmausführung durch das Drücken der ESCAPE-Taste unterbrochen bzw. nicht unterbrochen werden kann. Die Voreinstellung ist ON.

**SET EXACT** *Schalter*

Vollständiger Vergleich von Zeichenketten. Bei ON muß ein Suchbegriff vollständig sein, bei OFF ist ein verkürzter Suchbegriff möglich. Die Voreinstellung ist OFF.

**SET FILTER TO** [*Bedingung*]

Heraussuchen jener Datensätze aus einer Datei, welche der gestellten Bedingung entsprechen. Mit SET FILTER TO wird die Wirkung des Befehls wieder aufgehoben.

**SET FIXED** *Schalter*

Festkommadarstellung bei numerischen Ausgaben. Die Voreinstellung ist OFF.

**SET FORMAT TO** *Basisname\_der\_Formatdatei*

Aufrufen einer vom Nutzer erstellten Formatdatei für das Editieren und Erfassen von Datensätzen.

**SET FUNCTION** *Tastenummer* **TO** *'Zeichenkette'*

Belegung von Funktionstasten mit REDABAS-Befehlen.  
Standardbelegungen sind:

F 1 : HELP	F 6 : DISPLAY STATUS
F 2 : ASSIST	F 7 : DISPLAY MEMORY
F 3 : LIST	F 8 : DISPLAY
F 4 : DIR	, F 9 : APPEND
F 5 : DISPLAY STRUCTURE	F 10 : EDIT

**SET HEADING** *Schalter*

Benutzung der Überschriftenzeile bei den Standardanzeigen über LIST, DISPLAY, \*SUM und AVERAGE. Die Voreinstellung ist ON.

**SET HELP** *Schalter*

Hilfestellung bei Syntaxfehlern. Die Voreinstellung ist ON.

**SET INDEX TO** [*Basisname\_der\_Indexdatei*] {, *Basisname\_der\_Indexdatei*}

Eröffnen von Indexdateien (maximal 7). Die Sortierfolge der Stammdatei wird durch die 1. Indexdatei bestimmt. Die anderen Indexdateien werden jedoch bei Veränderungen der Stammdatei mit aktualisiert.

**SET INTENSITY** *Schalter*

Umschalten in eine inverse Darstellung. Die Voreinstellung ist ON.

**SET MARGIN TO** *Zahl*

Randeinstellung bei Druckausgaben.

**SET MENUS** *Schalter*

Ein- oder Ausschalten der Menüs zur Cursorsteuerung. Die Voreinstellung ist ON.

**SET PATH TO** [*Pfadliste*]

Festlegen alternativer Zugriffspfade.

**SET PRINT** *Schalter*

Entscheidung über die Druckausgabe von @... SAY-Befehlen. Die Voreinstellung ist OFF.

**SET PROCEDURE TO** [*Prozedurname*]

Eröffnen einer Prozedurdatei. Es kann immer nur eine Prozedurdatei geöffnet sein.

**SET RELATION TO** [*Schlüssel*] **INTO** *Dateikurzname*

Synchronisation der aktiven Datei mit einer anderen eröffneten Datei, die nach INTO mit ihrem Kurznamen (ALIAS-Namen) angegeben wird. Die Rücknahme der Synchronisation erfolgt über SET RELATION TO ohne Parameter.

**SET SAFETY** *Schalter*

Warnung vor dem Überschreiben und Zerstören von Dateien beim Kopierbefehl. Die Voreinstellung ist ON.

**SET STEP** *Schalter*

Unterbrechung der Programmausführung nach jedem Befehlsschritt. Die Programmfortsetzung erfolgt durch Drücken einer beliebigen Taste. Die Voreinstellung ist OFF.

**SET TALK** *Schalter*

Anzeige von Systemreaktionen auf Nutzeraktivitäten. Die Voreinstellung ist ON.

**SET UNIQUE** *Schalter*

Prüfen auf Duplikate in indizierten Dateien. Nicht als Unikate erkannte Datensätze werden eliminiert. Die Voreinstellung ist OFF.

**SKIP** *[[+ | -] Ausdruck]*

Erhöhen des Datensatzzählers der aktiven Datei um eine Position. Durch die Option kann der Datensatzzähler um den Wert des Ausdrucks vorwärts (+) bzw. rückwärts (-) bewegt werden.

**SORT TO** *Basisname\_der\_Datenbankdatei* **ON** *Feldname* *[/A] [/D]*

*{, Feldname [/A] [/D]}*

*[Bereich] [FOR Bedingung | WHILE Bedingung]*

Sortieren einer eröffneten Datenbankdatei entsprechend den Sortiermerkmalen. Das wichtigste Sortiermerkmal ist zuerst anzugeben. Für jedes Feld können aufsteigende (A) oder absteigende (D) Sortierfolgen angegeben werden. Die Voreinstellung ist A.

*Spalte ::= Zahl*

*Speichervariable ::= Name einer temporären Variablen*

*Speichervariablenliste ::= Speichervariable {, Speichervariable}*

*Start ::= Positionsangabe eines Zeichens als Zahl.*

**STORE** *Ausdruck* **TO** *Speichervariable* *{, Speichervariable} | Speichervariable = Ausdruck*

Initialisierung und Wertebelegung von Speichervariablen.

**SUM** *[Bereich] [Ausdrucksliste] [TO Speichervariablenliste] [FOR Bedingung | WHILE Bedingung]*

Summieren von Inhalten numerischer Felder einer eröffneten Datei. Die Summe kann auf Speichervariablen abgelegt werden. Ausdrucksliste und Speichervariablenliste korrespondieren durch ihre Reihenfolge.

*Systemparameter ::= Information über Leistungen von REDABAS als Ganzes bzw. einzelner Elemente.*

*Tastenummer ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10*

*Text ::= Beliebige Folge darstellbarer Zeichen.*

**TEXT** *Text* **ENDTEXT**

Markieren von Textblöcken. Die Schlüsselwörter TEXT und ENDTEXT müssen gesondert auf einer Zeile stehen.

**TOTAL ON** *Schlüssel* **TO** *Basisname\_der\_Datenbankdatei*

*[Bereich] [FIELDS Feldliste] [FOR Bedingung | WHILE Bedingung]*

Verdichten von Dateien durch das Addieren numerischer Felder entsprechend dem aufgeführten Kriterium. Das Ergebnis wird auf einer neuen Datei gespeichert.

**TYPE** *Dateiname* [**TO PRINT**]

Anzeigen oder Drucken (TO PRINT) von Textdateien.

**UPDATE ON** *Schlüssel* **FROM** *Dateikurzname* **REPLACE** *Feldname* **WITH** *Ausdruck* {, *Feldname* **WITH** *Ausdruck*} [**RANDOM**]

Aktualisieren der eröffneten Datei mit Daten aus einer Datei (*Dateikurzname*). Die Korrespondenz zwischen den Datensätzen wird über den Ordnungsbegriff hergestellt. Mit REPLACE wird der Inhalt der eröffneten Datei durch einen Ausdruck aus der FROM-Datei ersetzt. Die Option RANDOM bewirkt, daß eine unsortierte FROM-Datei sequentiell abgearbeitet wird.

**USE** [*Basisname\_der\_Datenbankdatei*] [**INDEX** *Basisname\_der\_Indexdatei* {, *Basisname\_der\_Indexdatei*}] [**ALIAS** *Dateikurzname*]

Eröffnen einer Datenbankdatei. Über die Option INDEX werden die angegebenen Indexdateien eröffnet. USE ohne Option schließt eine Datenbankdatei im Arbeitsbereich.

*Variable* ::= Oberbegriff für Feld- und Speichervariablen vom Typ Numerisch | Alpha-numerisch | Logisch | Datum.

**WAIT** [*'Zeichenkette'*] [**TO** *Speichervariable*]

Unterbrechung der Programmausführung und Ausgabe einer Standardaufforderung. Die Fortsetzung erfolgt durch das Drücken einer beliebigen Taste, deren Inhalt auf einer Speichervariablen abgelegt werden kann. Die Standardaufforderung kann durch eine andere Zeichenfolge ersetzt werden.

*Zahl* ::= Ganze positive Zahl

**ZAP**

Löschen aller Datensätze der aktiven Datei. Die Datensätze werden im Gegensatz zu DELETE vorher nicht löschmarkiert.

*Zeichen* ::= Zulässiges ASCII-Zeichen.

*Zeichenkette* ::= Beliebige Folge von Zeichen. Begrenzer für Zeichenketten sind ' ' | " " | [ ].

*Zeichenkette\_1* ::= *Zeichenkette*

*Zeichenkette\_2* ::= *Zeichenkette*

*Zeile* ::= *Zahl*

**@** *Zeile, Spalte* [**SAY** *Ausdruck* [**PICTURE** *Maske*]] [[**GET** *Variable* [**PICTURE** *Maske*]] [**RANGE** *Ausdruck* [, *Ausdruck*]]

Positionsgebundene Druckausgabe bzw. Bildschirmein- und -ausgabe unter Verwendung von Masken. Die RANGE-Option wird im Zusammenhang mit Variablen vom Typ NUMERICAL und DATE benutzt. Der Ausdruck spezifiziert das Maximum und das Minimum des Eingabebereichs im Zusammenhang mit GET. Mit der Option PICTURE kann die Aus- und Eingabe der Daten formatiert werden. Für jede Position in der PICTURE-Maske kann der zulässige Zeichentyp festgelegt werden.

**@** *Zeile, Spalte* [**CLEAR**]

Cursorpositionierung auf dem Bildschirm. Die Option CLEAR ermöglicht partielles Löschen des Bildschirms

? | ? ? [Ausdrucksliste]

Abfragen von Speicherinhalten und Berechnen von Ausdrücken. Bei 2 Fragezeichen wird der Zeilenvorschub unterdrückt.

\* Vergleiche NOTE

= Vergleiche STORE

## 2. Funktionen

**ASC**('Zeichenkette')

Ordinalwert (ASCII) des ersten Zeichens einer *Zeichenkette*.

Beispiel:

? ASC('Nichts')                      Ergebnis: 78

**AT**('Zeichenkette\_1', 'Zeichenkette\_2')

Position der *Zeichenkette\_1* in *Zeichenkette\_2*. Ist sie nicht enthalten, ist die Position 0.

Beispiel:

? AT('sch', 'Hochschule')              Ergebnis: 5

**BOF**()

Prüfen einer Datenbankdatei auf Dateianfang (BOF = BEGIN OF FILE).

Beispiel:

GO TOP

? BOF()                      Ergebnis: .F.

SKIP-3

? BOF()                      Ergebnis: .T.

**CDOW**(*Datumsvariable*)

Bereitstellen der Wochentagsangabe als Zeichenkette (CDOW = CHARACTER DAY OF WEEK).

Beispiel:

DATE() = 19. 11. 87

? CDOW(DATE())                      Ergebnis: Donnerstag

**CHR**(*Zahl*)

ASCII-Darstellung der *Zahl*.

Beispiel:

? CHR(65)                      Ergebnis: A

**CMONTH**(*Datumsvariable*)

Bereitstellen der Monatsangabe der *Datumsvariablen* als Zeichenkette.

Beispiel:

DATE() = 19. 11. 87

? CMONTH(DATE())                      Ergebnis: November

**COL**()

Aktuelle Spaltenposition des Cursors.

**CTOD('Zeichenkette')**

Umwandlung einer Zeichenkette in eine Variable vom Typ Datum  
(CTOD = CHARACTER TO DATE).

Beispiel:

STORE '19. 11. 87' TO Tag

Heute = CTOD(Tag)

Morgen = Heute + 1

? Morgen Ergebnis: 20. 11. 87

**DATE()**

Speicherplatz für das Tagesdatum.

Beispiel:

? DATE() Ergebnis: Tagesdatum

**DAY(Datumsvariable)**

Numerischer Wert der Tagesangabe im Datum.

Beispiel:

Systemdatum = 19. 11. 87

? DAY(DATE()) Ergebnis: 19

Stag = DAY(DATE())

? TYPE('Stag') Ergebnis: N

**DELETED()**

Prüfen von Datensätzen auf Löschmarkierung.

Beispiel:

GO 27

? DELETED() Ergebnis: z. B. .T.

**DOW(Datumsvariable)**

Numerischer Wert des Wochentags im Datum (DOW = DAY OF WEEK). Die Nummerierung beginnt bei Sonntag = 1.

Beispiel:

Systemdatum = 20. 11. 87

? DOW(DATE()) Ergebnis: 6 = Freitag

? DOW(DATE() + 2) Ergebnis: 1 = Sonntag

**DTOC(Datumsvariable)**

Umwandlung einer *Datumsvariablen* in eine Zeichenkette  
(DTOC = DATE TO CHARACTER).

Beispiel:

STORE DTOC(DATE()) TO Test

? TYPE('Test') Ergebnis: C

**EOF()**

Prüfen einer Datenbankdatei auf Dateiende (EOF = END OF FILE).

Beispiel:

GO BOTTOM

SKIP

? EOF() Ergebnis: .T.

**EXP(Numerischer Ausdruck)**Exponentialwert des numerischen Ausdrucks (Funktion :  $e^x$ ).

Beispiel:

? EXP(1) Ergebnis: 2.718

**FILE('Laufwerk' Dateiname')**

Prüfen des Vorhandenseins einer Datei. Das Ergebnis ist entweder .T. oder .F..

Beispiel:

? FILE('b:probe.dbf') Ergebnis: z. B. .T.

**INT(Numerischer\_Ausdruck)**

Ganzzahliger Teil eines numerischen Ausdrucks.

Beispiel:

? INT(12.345) Ergebnis: 12

**LEN('Zeichenkette')**

Länge einer Zeichenkette.

Beispiel:

? LEN('Das ist eine Zeichenkette') Ergebnis: 25

**LOG(Numerischer\_Ausdruck)**

Natürlicher Logarithmus des numerischen Ausdrucks.

Beispiel:

? LOG(2.71828) Ergebnis: 1.00000

**LOWER('Zeichenkette')**

Kleinbuchstaben von Zeichenkette.

Beispiel:

? LOWER('LEIPZIG') Ergebnis: leipzig

**MONTH(Datumsvariable)**

Numerischer Ausdruck der Monatsangabe aus dem Datum.

Beispiel:

Systemdatum : 19. 11. 87

? MONTH(DATE()) Ergebnis: 11

**PCOL()**

Spaltenposition des Druckkopfes.

**PROW()**

Zeilenposition des Druckkopfes.

**RECNO()**

Aktuelle Position des Datensatzzählers.

Beispiel:

? RECNO() Ergebnis: z. B. 13

**ROUND(Numerischer\_Ausdruck, Zahl)**

Aufgerundeter Dezimalbruch mit Zahl Stellen.

Beispiel:

? ROUND(12345.6789, 2) Ergebnis: 12345.6800



**ROW()**

Zeilenposition des Cursors.

Beispiel:

? ROW() Ergebnis: z. B. 12

**SPACE(Zahl)**

Folge von *Zahl* Leerzeichen.

Beispiel:

STORE SPACE(60) TO leer

**SQRT(Numerischer\_Ausdruck)**

Quadratwurzel aus einem positiven Zahlenausdruck.

Beispiel:

? SQRT(64) Ergebnis: 8

**STR(Numerischer\_Ausdruck [, Länge [, Dezimalstellen]])**

Umwandlung eines numerischen Ausdrucks in eine Zeichenkette. Die Länge der Zeichenkette (Gesamtlänge) und die Anzahl der zu berücksichtigenden Dezimalstellen kann angegeben werden.

Beispiel:

? STR(1234.567, 8, 1) Ergebnis: 123456.7

**SUBSTR('Zeichenkette', Start [, Länge])**

Teilzeichenkette von *Zeichenkette* entsprechend der Start- und Längenangabe.

Beispiel:

? SUBSTR('Informatik', 8, 3) Ergebnis: tik

**TIME()**

Speicherplatz für die Uhrzeit.

Beispiel:

? TIME() Ergebnis: z. B. 13:10:37

**TRIM(Alphanumerischer Ausdruck)**

Entfernen nachfolgender Leerzeichen aus alphanumerischen Datenfeldern.

Beispiel:

Datenfeld fname, 12, c

Datenfeld vorname, 12, c,

? fname, vorname

Ergebnis: MARX

KARL

TRIM(fname) + ' ' + vorname Ergebnis: MARX KARL

**TYPE('Ausdruck')**

Datentyp von *Ausdruck* (C : Alphanumerisch, N : Numerisch, L : Logisch, D : Datum).

Beispiel:

? TYPE('Date()') Ergebnis: D

**UPPER('Zeichenkette')**

Großbuchstaben der *Zeichenkette*.

Beispiel:

Name = 'Leipzig'

? UPPER(Name) Ergebnis: LEIPZIG

**VAL**(*'Alphanumerischer Ausdruck'*)

Numerischer Wert einer Zeichenkette. Nichtnumerische Zeichen erhalten den Wert 0.

Beispiel:

? VAL('123XYZ')                      Ergebnis: 123.00

**YEAR**(*Datumsvariable*)

Numerischer Wert der Jahreszahl eines Datums in der vollständigen Form.

Beispiel:

Systemdatum: 19. 11. 1987

? YEAR(DATE())                      Ergebnis: 1987

Jahr = YEAR(DATE())

? TYPE('Jahr')                      Ergebnis: N

*'Ausdruck'\$Feldname*

Vergleich des Inhalts eines Datenfeldes mit einem Ausdruck. Der Vergleich ist .T., wenn der Ausdruck innerhalb des Datenfeldes auftritt.

Beispiel:

LOCATE FOR '77'\$wohnort              Ergebnis: z. B. .T.

**&Speichervariable**

Indirekte Adressierung von Variablen und Dateien, deren Namen auf *Speichervariablen* abgelegt sind (Makro-Funktion &).

Beispiel:

USE Sdat INDEX>Nameind

STORE 'Schuster' to Sname

FIND &Sname

# Literaturverzeichnis

---

## 1. Bücher und Broschüren

- 1.1. Apelt, S.; Huhn, G.;  
Meuche, H.-F. u. a. Grundlagen der elektronischen Datenverarbeitung für Ökonomen. Berlin: Verlag Die Wirtschaft 1986 (2. Auflage).
- 1.2. Bachmann, P. Programmsysteme. Berlin: Akademie-Verlag 1983.
- 1.3. Bernstein, K. (Hrsg.) Tabellenkalkulation mit Personalcomputern. Berlin: Verlag Die Wirtschaft 1987.
- 1.4. Eggerichs, W. dBASE II. Band 1: Einführung; Band 2: Programmierung. Berlin: VEB Verlag Technik.
- 1.5. Engelen, M.; Stahn, H. Software-Engineering ARS-Technologie. Berlin: Akademie-Verlag 1984.
- 1.6. Entreß, G.; Entreß, L. Einführung in die Informationsverarbeitung. Berlin: VEB Verlag Technik 1986.
- 1.7. Goldammer, G. PASCAL für die Anwendung in der Wirtschaft. Berlin: Verlag Die Wirtschaft 1987.
- 1.8. Grafik, W. Aufbau und Arbeitsweise von 16-Bit-Mikroprozessoren. Berlin: VEB Verlag Technik 1987 (Reihe Automatisierungstechnik, Band 226).
- 1.9. Grund, F. Prinzipien des Betriebssystems OS/EC. Berlin: VEB Deutscher Verlag der Wissenschaften 1987 (2., berichtigte Auflage).
- 1.10. Hansen, R. Wirtschaftsinformatik. Band 1 bis 3. Stuttgart – New York: Gustav-Fischer-Verlag 1986 (5., neubearbeitete und stark erweiterte Auflage).
- 1.11. Helmerich, R.;  
Schwindt, P. CAD-Grundlagen. Leipzig: VEB Fachbuchverlag 1986.
- 1.12. Hempel, U.; Loley, H. Datenbanken mit Personalcomputern. Berlin: Verlag Die Wirtschaft 1987.
- 1.13. Higman, B. Programmiersprachen. Leipzig: BSB B. G. Teubner Verlagsgesellschaft 1971.
- 1.14. Jaglom, A. M.;  
Jaglom, I. M. Wahrscheinlichkeit und Information. Berlin: VEB Deutscher Verlag der Wissenschaften 1967.
- 1.15. Kalfa, W. Betriebssysteme. Berlin: Akademie-Verlag 1988.

- 1.16. Kämmerer, W. Einführung in mathematische Methoden der Kybernetik. Berlin: Akademie-Verlag 1971.
- 1.17. Kochan, D. u. a. CAD/CAM Schlüsseltechnologie als Intensivierungsfaktor. Berlin: Verlag Die Wirtschaft 1987.
- 1.18. Krauss, M.; Kutschbach, E.; Woschni, E.-G. Handbuch Datenerfassung. Berlin: VEB Verlag Technik 1984.
- 1.19. Lindemann, B. Lokale Computernetze. Berlin: Verlag Die Wirtschaft 1988.
- 1.20. Löffler, H. Rechnernetzwerke. Berlin: VEB Verlag Technik 1982 (Reihe Automatisierungstechnik, Band 199).
- 1.21. Löffler, H. Lokale Netze. Berlin: Akademie-Verlag 1987.
- 1.22. Müller, K.-H. (Hrsg.) Projektierung von EDV-System-Zugängen. Berlin: Verlag Die Wirtschaft 1982 (Schriftenreihe Informationsverarbeitung).
- 1.23. Paulin, G. Grundlagen der Programmieretechnik. Berlin: VEB Verlag Technik 1974.
- 1.24. Paulin, G. Kleines Lexikon der Mikrorechenetechnik. Berlin: VEB Verlag Technik 1986 (Reihe Automatisierungstechnik, Band 206).
- 1.25. Petschke, D.; Müntefering, P. Textverarbeitung mit Personalcomputern. Berlin: Verlag Die Wirtschaft 1988.
- 1.26. Rothhardt, G. Praxis der Softwareentwicklung. Berlin: VEB Verlag Technik 1987.
- 1.27. Schneider, H.; Köhli, S.; Zvacek, W. Büroautomatisierung. Berlin: VEB Verlag Technik 1987 (Reihe Automatisierungstechnik, Band 221).
- 1.28. Schumann, J.; Gerisch, M. Software-Entwurf. Berlin: VEB Verlag Technik 1984.
- 1.29. Stamer, E.; Ziese, G. Arbeitsplatzcomputer. Berlin: Verlag Die Wirtschaft 1988.
- 1.30. Völz, H. Kleines Lexikon der Speichertechnik. Berlin: VEB Verlag Technik 1987 (Reihe Automatisierungstechnik, Band 224).
- 1.31. Zeth, R. u. a. Der Personalcomputer 1715. Berlin: Verlag Die Wirtschaft 1987.

## 2. Zeitschriftenartikel

- 2.1. – Komplexprogramm des wissenschaftlich-technischen Fortschritts der Mitgliedsländer des RGW bis zum Jahre 2000. In: Einheit, Berlin 41 (1986) 2.
- 2.2. Bischoff, C. Einheitliche Textverarbeitung auf unterschiedlicher Hardware mit dem kommandoorientierten Textprogramm TEPROS. In: Neue Technik im Büro, Berlin 31 (1987) 1.

- 2.3. David, J.; Willenberg, M. Texthandbücher – die Voraussetzung für eine rationelle Textverarbeitung. In: Neue Technik im Büro, Berlin 27 (1983) 5.
- 2.4. Drücker, W.; Urban, D. Die elektronischen Schreibmaschinen robotron S 6100. In: Neue Technik im Büro, Berlin 31 (1987) 1.
- 2.5. Fischer, J. Dokumentationsherstellung mit der Textverarbeitungskomponente NROFF des Betriebssystems MÜTOS 1630. In: Neue Technik im Büro, Berlin 31 (1987) 1.
- 2.6. Geißer, M.; Picht, J.; Teichmann, L. Gestaltung einer Software-Produktions-Umgebung für 16-Bit-PC. In: edv aspekte, Berlin 7 (1988) 1.
- 2.7. Händler, B. Der Klarschriftleser robotron K 6710 und seine Anwendung zur Rationalisierung von Arbeitsprozessen. In: Neue Technik im Büro, Berlin 29 (1985) 2.
- 2.8. Hoffmann, J.; Schindler, R. Textbausteinverarbeitung – Ermittlung und Gestaltung von Textbausteinen. In: Neue Technik im Büro, Berlin 27 (1983) 5.
- 2.9. Koziolk, H. Unsere ökonomische Strategie mit dem Blick auf das Jahr 2000 – hoher Anspruch an die Wirtschaftswissenschaften. In: Einheit, Berlin 41 (1986) 8.
- 2.10. Manski, H.; Marschner, M.; Schenke, N. robotron-Computersystem für das Berliner Grand-Hotel. In: Neue Technik im Büro, Berlin 31 (1987) 4.
- 2.11. Moser, D. Automatisierte Textverarbeitung mit dem elektronischen Schreibsystem robotron A 5310. In: Neue Technik im Büro, Berlin 27 (1983) 5.
- 2.12. Naumburger, V.; Püschel, K. SEA 1600 – ein Gerät zur phonetischen Ein- und Ausgabe. In: Neue Technik im Büro, Berlin 26 (1982) 6.
- 2.13. Queiser, B. Text 30 – ein universelles, menügesteuertes Textverarbeitungssystem für den Bürocomputer robotron A 5120/A 5130. In: Neue Technik im Büro, Berlin 29 (1985) 2.
- 2.14. Richter, J.; Terpe, B. ROLANET 1 – ein lokales Netz des VEB Kombinat Robotron. In: Neue Technik im Büro, Berlin 30 (1986) 5.
- 2.15. Schmiedchen, H.; Teucher, B. Schreib/Lese-Einheit für Plastkarten mit Magnetstreifen. In: Neue Technik im Büro, Berlin 27 (1983) 4.
- 2.16. Weber, R. TEXT 40/M16 – Programmsystem zur Textverarbeitung auf 16-Bit-Bürocomputern. In: edv aspekte, Berlin 6 (1987) 1.

### 3. Gesetzliche Bestimmungen und Normative

- 3.1. Anordnung vom 28. 5. 1975 über die Nomenklatur der Arbeitsstufen und Leistungen von Aufgaben des Planes Wissenschaft und Technik.  
GBl. I/1975 Nr. 23 S. 426.
- 3.2. Anordnung vom 5. 2. 1982 über die Rahmenrichtlinie für die Ermittlung, Planung, Kontrolle und Abrechnung der Effektivität der Maßnahmen des wissenschaftlich-technischen Fortschritts.  
In: GBl. I/1982 Nr. 8 S. 165.
- 3.3. Anordnung vom 27. 1. 1989 über die Planung, Bilanzierung und Abrechnung von Software.  
In: GBl. I/1989 Nr. 6 S. 100.
- 3.4. Anordnung vom 28. 2. 1986 über Informations- und Beratungsleistungen zur Entwicklung, Produktion und Mehrfachnutzung von Software in der DDR.  
In: GBl. I/1986 Nr. 9 S. 94.
- 3.5. Verordnung vom 11. 9. 1986 über den Erneuerungspaß und das Pflichtenheft.  
In: GBl. I/1986 Nr. 30 S. 409.
- 3.6. Anordnung Nr. 2 vom 18. 12. 1986 über die Nomenklatur der Arbeitsstufen und Leistungen von Aufgaben des Planes Wissenschaft und Technik.  
In: GBl. I/1987 Nr. 1 S. 7.
- 3.7. Anordnung vom 19. 12. 86 über Grundsätze für das einheitliche Herangehen an die Ermittlung, Planung und Nachweisführung des Nutzens und der Effektivität der Maßnahmen des wissenschaftlich-technischen Fortschritts – Nutzensanordnung.  
In: GBl. I/1987 Nr. 1 S. 1.
- 3.8. Fachbereichsstandard TGL 44 530/28: Informationsverarbeitung/Termini und Definitionen für Software-Qualitätssicherung.
- 3.9. Fachbereichsstandard TGL 44 545: Informationsverarbeitung/Phasenmodell des Softwarelebenszyklus.
- 3.10. Fachbereichsstandard TGL 44 535: Informationsverarbeitung/Dokumentation von Programmen, Programmsystemen und Softwareanwendungssystemen.
- 3.11. Fachbereichsstandard TGL 44 690: Bildschirmarbeitsplatz.

# Bildnachweis

---

Abbildungen 1.15., 2.10., 2.24., 2.26., 3.5.:	Viola Boden, Leipzig
Abbildungen 2.12., 2.18.	VEB Kombinat Robotron
Abbildung 2.3.	Rechentchnik/Datenverarbeitung, Berlin 24 (1987) 3.
Abbildung 2.8.:	Vgl. Autorenkollektiv: Grundlagen der elektronischen Datenverarbeitung für Ökonomen. Berlin: Verlag Die Wirtschaft 1986, S. 38.
Abbildung 2.9.:	Vgl. Autorenkollektiv: Grundlagen der elektronischen Datenverarbeitung für Ökonomen. Berlin: Verlag Die Wirtschaft 1986, S. 36.
Abbildung 2.11.:	Vgl. Helmerich, R.; Schwindt, P.: CAD-Grundlagen. Leipzig: VEB Fachbuchverlag 1986, S. 31.
Abbildung 2.13.:	Vgl. Krauss, M.; Kutschbach, E.; Woschni, E.-G.: Handbuch Datenerfassung. Berlin: VEB Verlag Technik 1984, S. 150.
Abbildung 2.14.:	Vgl. Entreß, G.; Entreß, L.: Einführung in die Informationsverarbeitung. Berlin: VEB Verlag Technik 1986, S. 149.
Abbildung 2.16.:	Krauss, M.; Kutschbach, E.; Woschni, E.-G.: Handbuch Datenerfassung. Berlin: VEB Verlag Technik 1984, S. 66.
Abbildung 2.18.:	Neue Technik im Büro, Berlin 27 (1983) 4.
Abbildung 2.28.:	Krauss, M.; Kutschbach, E.; Woschni, E.-G.: Handbuch Datenerfassung. Berlin: VEB Verlag Technik 1984, S. 134.
Abbildung 2.29.:	Vgl. Autorenkollektiv: Grundlagen der elektronischen Datenverarbeitung für Ökonomen. Berlin: Verlag Die Wirtschaft 1986, S. 55.
Abbildung 4.10.:	Neue Technik im Büro, Berlin 31 (1987) 4.
Alle übrigen Abbildungen entstanden nach Vorlagen der Autoren.	

# Sachwortverzeichnis

---

## A

ADA 136  
 Adreßbreite 38  
 AIV-System  
     s. Informationsverarbeitungssystem,  
     automatisiertes  
 Akustikkoppler 64  
 ALGOL 135  
 Algorithmus 133  
 Alternative 124  
 Analogrechner 33  
 Analysieren 112  
 A-Nomenklatur 117  
 Anschlußbild 38, 42  
 Anwenderdokumentation 116  
 Anwendersoftware 25, 108  
     – Klassifikation 109  
     – Qualität 109  
 Arbeitsplatzcomputer  
     s. Personalcomputer  
 Arbeitsplatz, rechnergestützter 27  
 Arbeitsstation 74  
 ARIADNE 162  
 Arithmetikprozessor 40  
 Assembler 131  
 Asynchronübertragung 68

## B

Backtracking 138  
 Backus-Naur-Form 135  
 Balkencode 11  
 Bandbreite 66  
 BASIC 135  
 Basissoftware 25  
 Basissystem 27  
 Baumnetz 72  
 BCD-Darstellung 21  
 Befehlsdatei 157

Befehlsverkettung 105  
 Betreiben 117  
 Betriebssystem 25, 70, 82, 87  
 Bibliotheksmodul 80  
 Bildschirmtechnik 46  
 Bildschirmtext 77  
 Bildzeichen 11  
 Binärzahl 14  
     – Normalisierung 22  
 Bit 10  
 Bitgruppe 12  
 Bitmuster 11, 15  
 Blockdiagramm 121  
 BOTTOM-UP-Methode 120  
 Box 80  
 Bürocomputer  
     s. Personalcomputer  
 Bürofernschreiben 76  
 Bus 37  
 Busnetz 72  
 Byte 12, 40

## C

CAD 29  
 CAM 29  
 CAP 29  
 Checkliste 111  
 Chip 38  
 CIM 29  
 COBOL 135  
 Code 12  
     – genormter 17  
     – 7-Bit-Code 18  
     – 8-Bit-Code 18  
 Codetabelle 17  
 Codierung 11, 14  
     – nichtnumerische Information 16  
     – numerische Information 19



- Compiler 132  
Computer 24  
CPU  
    s. Zentralprozessor  
Crossreferenzliste 141
- D  
Dämpfung 66  
Datei 26  
Dateiart 157  
Dateiattribut 97, 104  
Dateispezifikation 89  
Dateiverwaltung 82  
Daten 10, 25  
Datenbank 26  
Datenbankbetriebssystem 81  
Datenbankdatei 157  
Datenbanksoftware 81  
Datenbanksystem 81  
Datensatz 25  
Datenstation  
    s. Kommunikationsstation  
Datentyp 25  
Datenübertragung 77  
Datumumleitung 104  
Datenverwaltung 154  
DCP 89, 96  
Dialog  
    – kommandogesteuert 95  
    – menügesteuert 95  
Dialogverarbeitung 13, 94  
Dienstprogramm 80  
Digitalrechner 33  
Digitizer 48  
Diskette 58  
Diskettenbehandlung 61  
Disketteneinteilung 59  
Diskettenformat 60  
Diskettenkapazität 59  
Doppelwort 12  
DOS/EC 91  
Drucktechnik 48  
Dummy 115  
Duplexbetrieb 67  
Duplizieren 103  
Durchsatz 34
- E  
EAN 25  
EAROM 42  
EC-Chiffre 34  
Echtzeitverarbeitung 96  
Editieren 115  
EEPROM 42  
Einchipprozessor 38  
Einerkomplement 20  
Einführen 117  
Ein- und Ausgabesystem 82  
Einzelplatz-Textverarbeitungssystem 148  
E-Nomenklatur 117  
Entscheidungstabelle 124  
Entwerfen 114  
Entwicklungssoftware 80  
Entwurfssystem 126  
EPROM 42  
EVA-Tabelle 122  
Expertensystem 138
- F  
Fallunterscheidung 124  
Feldformat 150  
Feldtyp 150  
Fernkopieren 76  
Fernnetz 72  
Fertigstellen 116  
Festplattenspeicher 61  
Filter 105  
Firmware 83  
Folienspeichertechnik 58  
Formularverarbeitung 147  
FORTRAN 134
- G  
Gerätedateiname 98  
Gerätetreiber 69, 82  
Gleitkommadarstellung 22  
Grafik-Modus 47  
Großrechenanlage 34  
Gruppenname 98
- H  
Halbbyte 12  
Halbduplexbetrieb 67  
Halbleiterplatte 62  
Hardware 23  
Hardwareergonomie 27  
Hauptspeicher 40  
Hauptspeicherkapazität 40  
Hexadezimalsystem 15  
Hybridrechner 33

## I

Implementieren 115  
 Indexdatei 157  
 Individualsoftware 108  
 Informatik 9  
 Information 9  
 – nichtnumerische 17  
 – numerische 19  
 Informationsgehalt 10  
 Informationsverarbeitungssystem,  
 automatisiertes 23, 26  
 – Aufwand 31  
 – Effektivität 31  
 – Nutzen 30  
 – Ziele 29  
 Installation 82  
 Instruktion 132  
 Intelligenz, künstliche 138  
 Interface  
 s. Anschlußbild  
 Interpreter 133  
 ISO-Basis-Referenzmodell 70  
 ISO-Standard 139  
 Iteration 123

## J

Job 93

## K

Klausel 138  
 Kleinrechenanlage 34  
 Kommando 83  
 – residentes 85, 100  
 – transientes 85, 103  
 Kommandointerpreter 90  
 Kommando-Modus 154  
 Kommandosteuerung 145  
 Kommandoverwaltung 83  
 Kommandowort 83  
 Kommunikation 12  
 zwischen Programmen 14  
 Kommunikationskette 12  
 Kommunikationsprotokoll 69  
 Kommunikationssoftware 81  
 Kommunikationsstation 63  
 Komplementation 21  
 Konfiguration 35  
 Konvertierung 12  
 Konzentrator 65

## L

Leitungsvermittlung 68  
 Lesegerät 52  
 Lese-Schreib-Speicher  
 s. RAM  
 Lesetechnik, optische 51  
 Lichtstift 48  
 Linker 132  
 LISP 138  
 Lochkartentechnik 45  
 Lochstreifentechnik 44

## M

Magnetbandtechnik 56  
 Magnetkartentechnik 52  
 Magnetspeicher 56  
 Maschinenbefehl 129  
 Maschinennetz 72  
 Maschinenprogramm 129  
 Maschinensprache 129  
 Maskendatei 157  
 Master-Slave-Übertragung 68  
 Maus 48  
 Mehrnutzerbetrieb 86  
 Mehrplatz-Textverarbeitungssystem 149  
 Mehrprogrammbetrieb 86  
 Mehrpunktverbindung 65  
 Mensch-Maschine-Kommunikation 13, 33  
 Menüsteuerung 144  
 Mikroprogramm 39  
 Mikroprogrammspeicher 39  
 Mikroprozessor 40  
 Mikrorechnersystem  
 s. Personalcomputer  
 Modem 64  
 Modul 121  
 MULTICOMP 162  
 Multiplexer 65  
 MUTOS 90

## N

Nachricht 9  
 Netz, lokales 14, 73  
 Netztopologie 71  
 NROFF 149  
 Nur-Lese-Speicher  
 s. ROM  
 Nutzkoeffizient 31  
 Nutzungstechnologie 93

- O  
Objektprogramm  
    s. Maschinenprogramm  
OCR-Schrift 51  
Offline-Kommunikation 13  
Online-Kommunikation 13  
Operand 129  
Operationscode 129  
Orgware 28  
OS/EC 91
- P  
Paketvermittlung 68  
Paragraph 12  
PASCAL 135  
Peripherie 33  
Personalcomputer 33  
Pfad 89  
Pflichtenheft 117  
Pins 38  
Pipe-Operator 105  
PL/1 135  
Plattentechnik 61  
Plotter 50  
Programm 130  
Programmbibliothek 80  
Programmentwicklungssystem 80  
Programmiersprache  
    – deklarative 137  
    – deskriptive 136  
    – höhere 132  
    – maschinenorientierte 131  
    – prozedurale 133  
Programmiersprache C 136  
Programmierungsumgebung 139  
Programm-Modus 155  
PROLOG 138  
PROM 42  
Prototyping 128  
Prozeßdatenverarbeitung 96  
Prozeßrechner 96
- Q  
Qualitätssicherung 111  
Quellprogramm 130  
Quelltextformatierung 141
- R  
RAM  
    – dynamischer 41  
    – statischer 41  
Rechenwerk 39  
Rechner 24  
Rechnerfamilie 36  
Rechnergeneration 35  
Rechnernetz 14, 70  
Rechnersystem 24  
    – Grundstruktur 33  
    – Klassen 33  
REDABAS 154  
Redundanz 10  
Regel 138  
Register 33, 39  
Repeater 64  
Reportdatei 157  
Ringnetz 72  
ROM 41  
Rückflußdauer 31
- S  
Schalter 84  
Schlüssel 25  
Schlüsselsystematik 25  
Schnittstelle  
    s. Anschlußbild  
Schriftzeichen 11  
SCP 88  
Seitendrucker 50  
Sektor 60  
Selektion 123  
Sequenz 123  
Serienbriefverarbeitung 147  
Serverstation 74  
Shell  
    s. Kommandointerpreter  
Sicherungsdatei 157  
Signal 9  
Simplexbetrieb 67  
Software 24  
Softwareentwicklung 119  
Softwareentwicklungssystem 126  
Softwareergonomie 27  
Softwarelebenszyklus 112  
Softwarepaket, integriertes 162  
Softwarephase 112  
Softwarerahmen 81  
Softwaretechnologie 119  
Softwarewartung 117  
Speicheradresse 40  
Speicherdirektzugriff 43  
Speicher, externer 56  
Speicherzelle 40

- Sprachausgabe 11, 55
- Spracheingabe 11, 54
- Sprachgeneration 138
- Standardinterface 43
- Standardsoftware 108
- Standardwerkzeug 81, 143
- Stapelverarbeitung 13, 93
- Statement
  - s. Instruktion
- Sternnetz 71
- Steuerprogramm 79, 83
- Steuerwerk 39
- Steuerzeichen 17
- Steuerzeichenkommando 84
- Struktogramm 123
- Strukturdiagramm 121
- Stup 115
- Suchpfad 102
- Synchronübertragung 68
- Systemkern 139
- System, rechnergestütztes 27
- Systemservice 139
  
- T
- Tabellenkalkulation 149
- Taktzyklus 38
- Tastatur 46
- Telekommunikation 75
- TEPROS 149
- Terminalstation 64
- Testen 115
- TEXT 30 148
- TEXT 40 148
- Textbausteinverarbeitung 146
- Textdatei 157
- Texteingabe 146
- Textgestaltung 145
- Textkorrektur 146
- Text-Modus 47
- Textverarbeitung, automatisierte 143
- Textverarbeitungssystem 147f.
- TOP-DOWN-Methode 119
  
- Treiber
  - s. Gerätetreiber
- Typname 96
- Typsoftware 108
  
- U
- Übertragungsbreite 67
- Übertragungsgeschwindigkeit 66
- Übertragungsmedium 66
- Übertragungsnetzwerk 65
- Umgebungsergonomie 27
- Unterbrechungssystem 43
  
- V
- Variablendatei 157
- Verarbeitung, interaktive 94
- Verarbeitungsbreite 38
- Verbinden 115
- Vermittlungsstation 64
  
- W
- Wechselplattenspeicher 62
- Wirtschaftsinformatik 28
- Wirtsrechnersystem 115
- Wort 12
  
- Z
- Zahlenformel, allgemeine 14
- Zahlensystem 14
- Zeichen 9
- Zeichendrucker 49
- Zeichenfolge 9
- Zeichengenerator 19
- Zellendrucker 50
- Zentraleinheit 33
- Zentralprozessor 38
- Zielrechnersystem 115
- Zugriff 26
- Zugriffszeit 42
- Zweierkomplement 20
- Zweipunktverbindung 64



