

Gerd Küveler
Dietrich Schwoch

Arbeitsbuch Informatik

Eine praxisorientierte Einführung in die
Datenverarbeitung mit Projektaufgabe



Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Küveler, Gerd:

Arbeitsbuch Informatik: eine praxisorientierte Einführung
in die Datenverarbeitung mit Projektaufgabe / Gerd Küveler;
Dietrich Schwoch. – Braunschweig; Wiesbaden: Vieweg, 1996
(Vieweg Fachbücher der Technik)

ISBN 978-3-528-04952-2

ISBN 978-3-322-92907-5 (eBook)

DOI 10.1007/978-3-322-92907-5

NE: Schwoch, Dietrich:

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1996

Der Verlag Vieweg ist ein Unternehmen der Bertelsmann Fachinformation GmbH.



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Satz: saxxes, Essen

Gedruckt auf säurefreiem Papier

Vorwort

Die Teile I bis III dieses Buch entsprechen inhaltlich einer zweisemestrigen Einführung in die Datenverarbeitung für Ingenieurstudenten. Der Stundenumfang wird dabei mit insgesamt 4 Semesterwochenstunden Vorlesung und ebenso vielen Übungen angesetzt. Der Text, der auch ein Selbststudium erlauben soll, orientiert sich am PC und dessen Möglichkeiten. Die Einführung soll so praktisch wie möglich sein, d.h. der Leser erhält die Gelegenheit, das Gelernte weitgehend am eigenen – oder schuleigenen PC nachzuvollziehen. Aus diesem Grund nimmt der Übungsteil einen sehr breiten Raum ein. Die Diskette, die mit der beiliegenden Postkarte angefordert werden kann, befreit von unnötiger Tiparbeit, darüberhinaus enthält sie umfangreiche Demoprogramme und nützliche Werkzeuge.

Im Vordergrund stehen keine modernen Anwendungen und keine noch so komfortablen Benutzeroberflächen. Technisch orientierte Leser sollten auch heute noch „tief“ einsteigen: zunächst die strukturierte Programmierung am Beispiel einer universellen Programmiersprache kennenlernen, um später einen leichteren Zugang zu jeder anderen gerade benötigten Sprache, oder zu einem komplexen Anwendungsprogramm zu haben. Der funktionale Aufbau und die Arbeitsweise eines Mikrorechners ist der zweite Schwerpunkt, denn in zunehmendem Maße werden technische Projekte aller Art mit Hilfe solcher Systeme gelöst.

Dementsprechend stellen die Teile II und III

* Vermittlung einer Programmiersprache am Beispiel von PASCAL

* Funktionsweise von Mikrorechnern am Beispiel des PC und seines Prozessors 80(X)86

die Kernthemen dieses Buches dar.

Der Teil IV vermittelt einen ersten Einstieg in das wichtige Zukunftsthema Datennetze. Fast jeder Rechner wird heute vernetzt betrieben und selbst Heimcomputer können über Modems weltweit kommunizieren. Durch die Möglichkeit der lokalen und globalen Rechnernetzungen erleben wir z.Z. die Entstehung einer neuen Dimension in der Daten- und Informationsverarbeitung. Die Beschäftigung mit Rechnernetzen ist daher nicht länger eine Domäne von Informatikspezialisten. Dieses Thema muß zunehmend in die DV-Lehrpläne aller technisch orientierten Ausbildungsgänge einbezogen werden.

In den zahlreichen und z.T. recht ausführlichen Anhängen werden die im Hauptteil des Buches behandelten Themen ergänzt und erweitert. U.a. wird dort der Selbstbau einer Eingabe/Ausgabe-Erweiterungskarte in allen Einzelheiten erläutert. Diese Aufgabe eignet sich auch für ein Projekt.

Unser Dank gilt allen, die einen Beitrag zum Zustandekommen dieses Buches geleistet haben, besonders Björn Köhler, Hark Jones, Andreas Liebe, Andreas Papula, Dr. Erich Prochnio, Thomas Vogel und Dr. Serge Zakharian.

Eine extra Erwähnung verdient Holger Sahlmann, der bei zahlreichen technischen Details entscheidend mitgearbeitet hat.

Glashütten, im August 1995

Dieburg, im August 1995

*Gerd Küveler
Dietrich Schwach*

Inhaltsverzeichnis

I Grundlagen	1
1 Grundprinzipien der Datenverarbeitung	2
1.1 Historische Entwicklung der Datenverarbeitung	3
1.1.1 Von A wie Abakus bis Z wie Zuse	3
1.1.2 Die Rechnergenerationen	4
1.1.3 Computer und Science Fiction	6
1.2 Funktion und Aufbau eines Rechners	7
1.3 Übungen	12
2 Algorithmen und Struktogramme	13
2.1 Algorithmen	13
2.2 Struktogramme	14
2.3 Übungen	22
3 Programmiersprachen	23
3.1 Die Hochsprachen-Programmentwicklung	25
3.2 Ein Programmbeispiel in FORTRAN, PASCAL und C	28
3.3 Übungen	32
4 Das Betriebssystem DOS	33
4.1 Aufgaben eines einfachen Betriebssystems	34
4.2 Die Benutzerschnittstelle	35
4.3 Festplatten, Disketten, Dateien	37
4.4 Die wichtigsten DOS-Kommandos	39
4.5 Das hierarchische Dateiensystem	41
4.6 Wildcards, Filter und Pipes	45
4.7 Übungen	49
II Programmieren mit Turbo Pascal	53
5 Über Pascal und Turbo Pascal	54
6 Grundlagen	55
6.1 Ein einführendes Beispiel	55
6.2 Anweisungen, Wertzuweisungen und Datentypen	57

6.3	Der Aufbau eines Pascalprogramms	59
6.3.1	Die Bausteine der Sprache	59
6.3.2	Der Blockaufbau eines Programms.....	63
6.3.3	Separationszeichen.....	65
6.3.4	Kommentare.....	66
6.3.5	Die Freiheit der äußeren Form.....	66
6.4	Fehler	67
6.4.1	Syntaxfehler	67
6.4.2	Laufzeitfehler.....	68
6.4.3	Logische Fehler.....	69
6.5	Die integrierte Entwicklungsumgebung von Turbo Pascal.....	70
6.6	Aufgaben.....	73
7	Vordefinierte Standard-Datentypen und einfache Operationen	74
7.1	Der Umgang mit Zahlen.....	74
7.1.1	Ein wesentlicher Unterschied: Integer oder Real.....	74
7.1.2	Ganzzahlige Datentypen	76
7.1.3	Reelle Datentypen.....	81
7.1.4	Die Auswertung zusammengesetzter Ausdrücke.....	84
7.1.5	Standardfunktionen mit Zahlen.....	87
7.2	Verarbeitung von Einzelzeichen: der Datentyp Char.....	89
7.2.1	Der Umgang mit der ASCII-Tabelle.....	91
7.2.2	Standardfunktionen mit Char.....	92
7.3	Logische Variable und Ausdrücke: der Datentyp Boolean	94
7.4	Benutzerdefinierte Konstanten.....	99
7.5	Aufgaben.....	100
8	Interaktive Ein-/Ausgabe	102
8.1	Standard Ein-/Ausgabe.....	102
8.1.1	Eingabe: Die Anweisungen Read und ReadLn.....	103
8.1.2	Ausgabe: Die Anweisungen Write und WriteLn	108
8.2	Formatierte Ausgabe	113
8.3	Funktionen zur Bildschirmverwaltung und Tastaturabfrage.....	115
8.4	Aufgaben.....	118
9	Programmablaufstrukturen	120
9.1	Die Selektion.....	120
9.1.1	Die einseitige Verzweigung: IF .. THEN.....	120
9.1.2	Die Alternative: IF .. THEN .. ELSE	122
9.1.3	Die Mehrfach-Fallunterscheidung: CASE .. OF	125

9.2 Die Iteration.....	129
9.2.1 Die Zählschleife: FOR .. DO.....	129
9.2.2 Bedingungsschleifen	134
9.2.2.1 Die abweisende Bedingungsschleife: WHILE..DO	135
9.2.2.2 Die nicht-abweisende Bedingungsschleife: REPEAT..UNTIL... ..	140
9.3 Die Schachtelung von Kontrollstrukturen	146
9.4 Aufgaben	149
10 Modularisierung von Programmen: FUNCTION und PROCEDURE	157
10.1 Vereinbarungen von Funktionen und Prozeduren.....	160
10.2 Prozeduren ohne Parameterübergabe	162
10.3 Prozeduren mit Parameterübergabe.....	163
10.4 Funktionen.....	170
10.5 Der Geltungsbereich von Vereinbarungen	175
10.6 Seiteneffekte.....	179
10.7 Rekursionen.....	181
10.8 Aufgaben	185
11 Benutzerdefinierte Datentypen	188
11.1 Benutzerdefinierte Enumerationstypen	188
11.2 Teilbereiche	190
11.3 Felder: Der Datentyp ARRAY	191
11.3.1 Eindimensionale Felder	191
11.3.2 Mehrdimensionale Felder	204
11.4 Zeichenketten: String	207
11.5 Datenverbunde: RECORD	214
11.6 Aufgaben	219
12 Arbeiten mit Dateien	221
12.1 Textdateien: der Datentyp Text.....	223
12.2 Binärdateien: Der Datentyp FILE OF	233
12.3 Aufgaben	237
III Mikrocomputer	239
13 Interne Darstellung von Informationen	240
13.1 Darstellung positiver ganzer Zahlen.....	240
13.1.1 Binär- und Hexadezimalsystem	241
13.1.2 Umrechnungsverfahren.....	242
13.1.3 Rechnen im Dualsystem	249

13.2	Darstellung von vorzeichenbehafteten Ganzzahlen	250
13.3	Darstellung gebrochener Zahlen	257
13.4	Sonstige Zifferncodes.....	263
13.5	Darstellung von Zeichen	265
13.6	Das Prüfbitverfahren	266
13.7	Übungen	267
14	Architektur der 80(X)86-Prozessorfamilie	271
14.1	Aufbau eines Mikrocomputers	271
14.1.1	Mikroprozessor	272
14.1.2	Zentralspeicher	274
14.1.3	Ein/Ausgabe-Bausteine (I/O-Ports)	278
14.1.4	Busleitungen	278
14.2	Hardwaremodell der Intel 80(X)86-Prozessoren.....	280
14.2.1	Prozessor-Register	284
14.2.1.1	Arbeits- oder Datenregister.....	285
14.2.1.2	Indexregister	286
14.2.1.3	Stackpointerregister	286
14.2.1.4	Befehlszeigerregister.....	286
14.2.1.5	Flagregister	287
14.2.1.6	Segmentregister	290
14.2.2	Die Adressierung	291
14.2.3	Systemplatine.....	294
14.2.4	Übungen.....	296
15	Einführung in die Maschinensprache.....	298
15.1	Maschinenbefehle des 80(X)86.....	298
15.2	Das Hilfsprogramm DEBUG	301
15.3	Übungen	304
15.4	Befehlsarten.....	305
15.4.1	Transportbefehle	306
15.4.2	Arithmetische Befehle	307
15.4.3	Logische Befehle	310
15.4.4	Sprungbefehle.....	312
15.4.5	Befehle zur Prozessorsteuerung.....	314
15.4.6	Übungen.....	315
15.5	Adressierungsarten.....	318
15.5.1	Registeradressierung.....	319
15.5.2	Unmittelbare Adressierung.....	319
15.5.3	Direkte Adressierung	320

15.5.4	Indirekte Adressierung.....	320
15.5.5	Basisregister plus Displacement	324
15.5.6	Basisregister plus Indexregister plus Displacement.....	325
15.5.7	Detaillierter Aufbau eines Maschinencodes.....	325
15.5.8	Übungen.....	326
16	Schnittstellen zum Betriebssystem.....	328
16.1	BIOS und DOS.....	328
16.1.1	BIOS-Systemaufrufe.....	330
16.1.2	DOS-Systemaufrufe.....	332
16.2	Die Speichermodelle COM und EXE.....	335
16.3	Übungen	337
17	Unterprogramme und Programmunterbrechungen	339
17.1	Call-Unterprogramme	339
17.1.1	Die Befehle PUSH und POP	342
17.2	Interrupts	344
17.2.1	Die Interrupt-Vektor-Tabelle.....	345
17.2.2	Die Interruptarten.....	349
17.2.3	Der Interruptcontroller.....	350
17.3	Übungen	351
18	Controller-Bausteine und Ports	355
18.1	Die Befehle „IN“ und „OUT“	356
18.2	Beispiel: Programmierung des Interrupt-Controllers	358
18.3	Übung.....	365
19	Symbolische Assembler.....	368
19.1	Die Programmentwicklung.....	368
19.2	Die Syntax des MASM.....	369
19.3	Ausblick	377
19.4	Übung.....	378
IV	Rechnernetze.....	379
20	Grundlagen der seriellen Datenübertragung.....	381
20.1	Die asynchrone Datenübertragung	381
20.2	Die synchrone Datenübertragung.....	383
20.3	Fehlersicherung	384

21 Kommunikation über die RS232C/V.24-Schnittstelle.....	390
21.1 Die RS232C/V.24 - Schnittstelle.....	390
21.2 Terminalemulation und Kommunikationsprogramme	393
21.3 Kommunikation über Modems im Fernsprechnetz	396
21.4 Zugang zu öffentlichen Datennetzen (Datex-P).....	401
22 Das ISO/OSI - Schichtenmodell der Datenkommunikation	403
22.1 Probleme der Rechner-Rechner-Kommunikation	403
22.2 Das 7-Schichtenmodell	403
23 Lokale Netze	408
23.1 Funktionseinheiten	408
23.2 Basistopologien von Netzen.....	413
23.3 Klassifikation von LANs nach Netz-Zugriffsverfahren	415
23.4 Schichten-Protokolle	421
23.5 Koppellelemente.....	424
23.6 Das Netzwerk - Betriebssystem (NetWare).....	431
24 WANs praktisch: Anwendungen im Wissenschaftsnetz und INTERNET	447
24.1 Die Protokollwelten im WAN: OSI oder IP	447
24.2 Das Wissenschaftsnetz	449
24.3 Das INTERNET	454
Anhang	458
Lösungen der Übungsaufgaben.....	458
Anhang A Turbo Pascal: Standard Funktionen und Prozeduren.....	508
Anhang B DEBUG.....	532
B.1 Einsatzfelder eines Debuggers.....	532
B.2 Beschreibung des DEBUG	533
B.3 Beispiele für typische DEBUG-Sitzungen.....	539
Anhang C Der Assembler-Befehlssatz	545
Anhang D Das Lernprogramm EASY	571

Anhang E Selbstbau einer I/O-Erweiterungskarte	574
E.1 Der Systembus des PC	574
E.2 Die IO-Erweiterungskarte	577
E.3 Ein- Ausgabe.....	580
E.4 Praxis	582
E.5 Fertigung der I/O-Karte	584
E.6 Programmierung des Bausteins 8255.....	589
E.7 Die digitale I/O-Box	590
E.8 Testen der Karte.....	591
E.9 Beispielprogramme	592
Abbildungen zu Anhang E	595
Anhang F NOVELL NetWare (3.1x): Workstation Utilities	599
Anhang G ASCII-Tabelle	602
Sachwortverzeichnis	603

I

Grundlagen

In diesem Teil geht es darum, wichtige Grundlagen der Datenverarbeitung zu vermitteln, wobei wir zwar im wahrsten Sinne des Wortes bei (Adam und) EVA anfangen, uns aber dennoch einen Leser vorstellen, für den Computer selbstverständliche Gegenstände seiner Umgebung darstellen wie Autos und Fernsehgeräte. So werden Sie wissen, daß ein PC mindestens über eine Zentraleinheit, ein Bildschirmgerät und eine Tastatur verfügt. Wahrscheinlich haben Sie Computerspiele gespielt, Disketten kopiert und wissen deshalb, was eine Datei ist. Wenn nicht, wird es höchste Zeit, Computererfahrungen zu sammeln. Beginnen Sie eben mit diesem Teil und seinen Übungsaufgaben!

1 Grundprinzipien der Datenverarbeitung

Datenverarbeitung (DV) wird nicht erst seit Einführung des Computers betrieben. Das menschliche Gehirn kann offensichtlich auch ohne technische Hilfsmittel Daten verarbeiten.

■ Beispiel

Kopfrechenaufgabe: addiere 3 zu 2

Untergliederung in 3 Schritte

1. Aufnehmen der Daten mit unseren Sinnen
2. Verarbeiten der Daten im Gehirn
3. Mitteilen des Ergebnisses durch Wort oder Schrift



Hieraus leitet sich das Hauptprinzip der DV ab:

E	V	A
i	e	u
n	r	s
g	a	g
a	r	a
b	b	b
e	e	e
	i	
	t	
	u	
	n	
	g	

Dieses „EVA“-Prinzip liegt allen technischen DV-Anlagen zugrunde.

1.1 Historische Entwicklung der Datenverarbeitung

Der moderne Allzweckrechner entwickelte sich erst nach Ende des zweiten Weltkriegs. Trotzdem hat er eine lange Vorgeschichte. Bei den elektronischen Rechnern haben technologische Entwicklungssprünge zu den verschiedenen Computergenerationen geführt.

1.1.1 Von A wie Abakus bis Z wie Zuse

Die technische Geschichte der Rechenmaschinen weist folgende Meilensteine auf:

- ca. 1000 v. Chr. **Abakus (=Rechenbrett, Rechentafel)**
 - Speicherfunktion
 - Rechenfunktion: Addition und Subtraktion
 - noch heute meistbenutzte „Rechenmaschine“ der Welt

- 1623 **Schickardsche Rechenmaschine**
 - Zahnradgetriebe: 10 Räder mit je 10 Zähnen
 - Stellenwertübertrag wie beim Kilometerzähler
 - Addition und Subtraktion

- 1642 **Pascalsche Rechenmaschine**
 - Prinzip wie bei Schickard, jedoch mit verbesserter Mechanik

- 1672 **Rechenmaschine von Leibniz**
 - zusätzlich verschiebbare Staffelwalze zum schnellen Addieren und Multiplizieren

- 1833 **Differenzenmaschine nach Charles Babbage**
 - Babbage entwarf das Konzept eines programmierbaren Universalrechners mit Zahlenspeicher, Rechenwerk, Leitwerk sowie Ein- und Ausgabeeinheiten. Programmierung mit Lochkarten nach dem Vorbild der automatischen Webstühle nach Jacquard (um 1800).
Babbages „Analytical Engine“ konnte nicht gebaut werden, da bis zu 25 Zahnräder ineinander greifen mußten, was damals mechanisch noch nicht lösbar war. 1991 wurde sie jedoch erfolgreich nachgebaut.
 - sehr wichtig für die Theorie des Computers!

- 1886 **Papplochkarte nach Hermann Hollerith**
 - Einsatz zur Auswertung der 11. amerikanischen Volkszählung
 - Format einer 1-Dollar-Note
 - Hollerith gründete 1896 die Tabulating Machine Company. Fusionen führten 1924 zur IBM.

- 1941 **Z3 nach Konrad Zuse**
 - erster programmierbarer Computer der Welt
 - Schaltelemente waren Relais.
 - 1944 das amerikanische Gegenstück Mark I nach Howard Aikens

1.1.2 Die Rechnergenerationen

Die im zweiten Weltkrieg entwickelten Rechner werden heute der Vorgeneration zugeordnet (bis 1946).

In Deutschland „Z3“ (Konrad Zuse, 1941)

Anzahl der Additionen pro Sekunde: 20

Anzahl der Multiplikationen pro Sekunde: 15

In Amerika „Mark1“ (Howard Aikens, 1944)

Anzahl der Additionen pro Sekunde: 3

Anzahl der Multiplikationen pro Sekunde: 0.16

Übersicht über die Rechnergenerationen

Generation	Hardware-beispiele	Technologien	Speicherkapazität/ Verarbeitungsgeschwindigkeit	Software: Sprache/ Betriebssystem
1. 1946 bis 1958	ENIAC Z22 (Zuse) UNIVAC IBM 650 SIEMENS 704	Elektronen- röhren Trommel- speicher	0.02 MIPS/ 1 - 2 KByte	Maschinen- sprache/ _____
2. 1959 bis 1964	IBM 1400 AEG TR CDC 6600 SIEMENS- 2002	Transistoren Kernspeicher	0.1 MIPS/ 32 KByte	Assembler, FORTRAN/ Stapelverarb. Mehrpro- grammbetrieb
3. 1965 bis 1980	IBM /370 DEC PDP-11 SIEMENS- 7000 Cray 1	ICs Halbleiter- speicher	5 MIPS/ 1 - 2 MByte	div. Hoch- sprachen: C, Pascal, ... Multiuserdialog
4. 1981 bis heute	PC Cray XMP Sperry 1100 IBM 309x DEC VAX	Mikro- prozessoren Optische Speicher	50 MIPS/ 8 - 32 MByte	Sprachen der 4. Generation Code- Parallelisierung
Gegenwart	Workstations Hochleistungs- PC	i486 Pentium Power PC Computer-Netze	100 MIPS GByte	Netzsoftware Objektorien- tierte Sprachen: C++, ..
5. Entwicklung seit ca. 1990		supraleitende Keramiken ?	1000 MIPS/ viele GByte	?

Erläuterungen MIPS = Megainstructions per Second (Millionen Befehle pro Sekunde)

KByte = Kilobyte (1024 Byte, siehe Teil III)

MByte = Megabyte

GByte = Gigabyte

1.1.3 Computer und Science Fiction

Der erste Großrechner der Welt hieß ENIAC (1946). Dieses Monstrum mit 18000 Röhren wog nicht weniger als 30 Tonnen. Er konkurrierte noch mit Relaisrechnern wie der Z5, die zwar langsamer, aber zuverlässiger waren.

Die Autoren von Zukunftsromanen glaubten damals mit Recht, daß es bald immer bessere Computer geben würde. In einem wichtigen Punkt allerdings irrten sie: Je leistungsfähiger ein Rechner, umso größer muß er wohl sein, so dachten sie. Clifford D. Simak schrieb 1949 den Roman „An der Grenze“. Die Hauptrolle spielte ein Computer, der so groß ist wie die ganze Erde. Noch 1966 hieß ein anderer Science Fiction Computer „Colossus“, auch ein Riese, wie schon sein Name verrät.

Die Wirklichkeit kam ganz anders: mit rasantem Tempo wurden die Computer immer besser und schneller. Dabei nahm ihre Größe nicht zu, sondern ab. Der Grund: immer kleinere und schneller Bauteile wurden erfunden. Die Erfindung des Transistors und später des Chips (IC) revolutionierte die Elektronik. Kein Science Fiction Autor hat diese Entwicklung vorausgeahnt.

Mehr Glück hatten sie mit den Robotern. Science Fiction Autoren haben sogar interessante Beiträge zur Robotergeschichte geleistet. Im Jahre 1922 erschien in Prag das Theaterstück „WUR“. Karel Capek, der Autor, ist der Urheber der Bezeichnung „Roboter“, denn WUR bedeutet „Werstands Universal Robots“ und handelt von einer Fabrik, die massenweise Roboter herstellt.

Einen interessanten Beitrag zur Roboterforschung leistete auch der amerikanische Schriftsteller Isaac Asimov, der 1940 in seiner Zukunftserzählung „Ich, der Robot“ seine drei Robotergesetze aufstellte:

1. Ein Roboter darf einen Menschen nicht verletzen oder durch Untätigkeit zulassen, daß ein Mensch zu Schaden kommt.
2. Ein Roboter muß den Befehlen gehorchen, die er von Menschen erhalten hat, mit Ausnahme solcher Befehle, die zu dem ersten Gesetz in Widerspruch stehen.
3. Ein Roboter muß seine eigene Existenz schützen, solange dieses nicht zu dem ersten oder zweiten Gesetz im Widerspruch steht.

In einem aber haben die Schriftsteller sich dennoch geirrt: die echten Roboter sehen nicht menschenähnlich aus. Solche Roboter wurden bisher nur für Ausstellungen gebaut. Roboter, wie sie heute vieltausendfach in der Industrie eingesetzt werden, gleichen eher riesenhaften Armen. Gebaut werden sie für spezielle Aufgaben, z. B. schweißen sie Karosserieteile in Autofabriken zusammen.

Herz und Hirn eines realen Roboters ist ein kleiner Mikrocomputer mit einem Steuerungsprogramm. Spezielle „Sensoren“ erlauben es dem Roboter, zu „fühlen“ oder gar zu „sehen“, je nach seiner Aufgabe. Das Programm wertet alles aus. Es erteilt dann die notwendigen Befehle an die „Aktoren“. Solche Aktoren bewegen z.B. eine „Eisenhand“ mit einem Lackiergerät für Autos.

Fazit Nur selten gelingt es, Entwicklungen einigermaßen zutreffend vorherzusehen. Weder Wissenschaftler, noch Schriftsteller oder Politiker sind auf diesem Gebiet erfolgreich. Forschung läßt sich nur in begrenztem Maße sinnvoll planen: was heute sonnenklar erscheint, kann morgen Schnee von gestern sein.

Beseelte Computer und menschenähnliche Roboter: kaum jemand glaubt heute noch, daß sie jemals Wirklichkeit werden. Dennoch – die Vorstellung hat nichts von ihrer Faszination eingebüßt. Und so entstehen immer noch Bücher und v.a. äußerst erfolgreiche Filme zu diesen Themen:

Menschlich föhlende Computer:

- 2001 – Odyssee im Weltall
- Nummer Fünf lebt

Robotermenschen (Androiden):

- Blade Runner
- Terminator
- Terminator 2
- Robocop
- Robocop 2
- Robocop 3

Einer hat es geschafft, sich selbst in einen Androiden zu verwandeln, der Soul-Sänger Michael Jackson. Menschliche Roboter scheinen ungeheuer anziehend zu sein, Jackson beweist es.

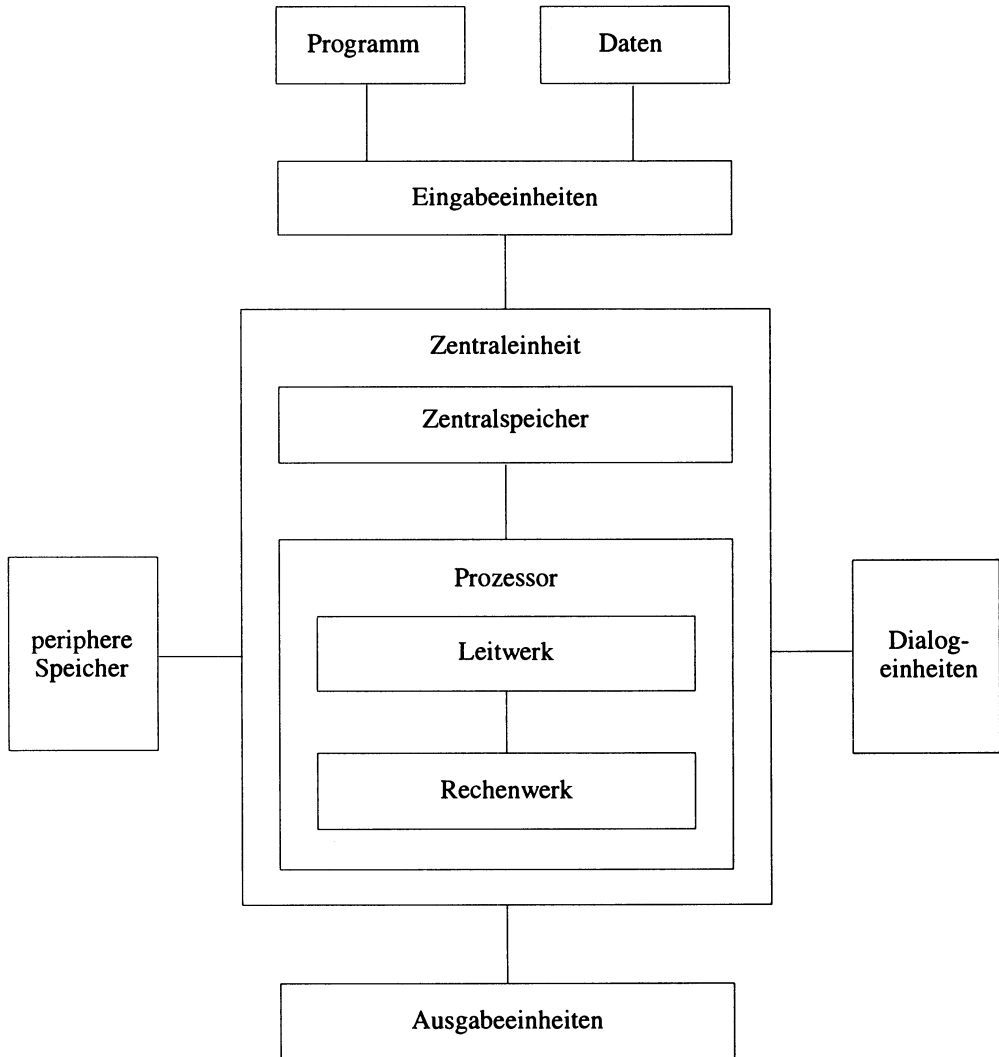
1.2 Funktion und Aufbau eines Rechners

Ab Mitte der 40er Jahre entwickelte John von Neumann ein theoretisches Rechnermodell. Seine Vorschläge (Entwicklung und Ausnutzung neuerer mathematischer Maschinen, 1954) lauteten:

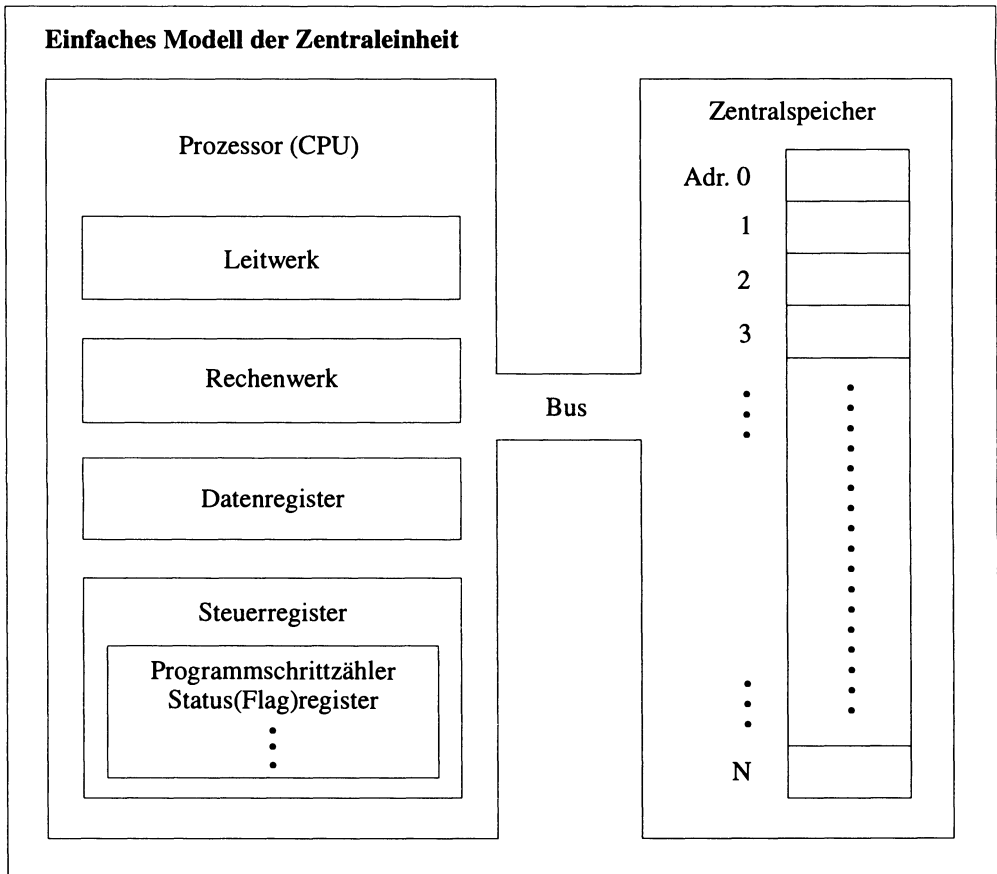
- Rechner sollen nicht mehr von „außen“ mit Lochstreifen gesteuert werden, sondern die Programmbefehle sind wie Daten in den Zentralspeicher zu laden (Geschwindigkeitsvorteile),
- Einführung bedingter Befehle (wenn $A > 0$ dann ...), damit höhere Flexibilität

Von Neumann stellte folgendes, bis heute gültiges Modell eines Rechners auf, das ebenfalls das EVA-Prinzip widerspiegelt. In aktualisierter Form läßt es sich so schematisieren:

Von-Neumannsches-Rechnermodell



Dieses Modell gilt für alle sequentiell arbeitenden Computer. Das sind Rechner, die ein Programm der Reihe nach, Schritt für Schritt, abarbeiten. Dazu gehören, mit geringen Modifikationen, auch PC. Am Beispiel eines modernen PC kann man sich veranschaulichen, welche Komponenten des PC den Funktionsblöcken des von-Neumann-Modells entsprechen (s. Übungsaufgaben 1.3). Lediglich sogenannte Parallelrechner, die mehrere Prozessoren besitzen, entsprechen nicht mehr dem von-Neumann-Prinzip. Auf diese speziellen Rechner soll hier jedoch nicht näher eingegangen werden. Wir wollen nun die Zentraleinheit eines Rechners gewissermaßen unter der Lupe betrachten.



Ein Prozessor – heute meist als Mikroprozessor (s. Teil III) realisiert – besitzt neben Leit- und Rechenwerk noch einige Register, sozusagen als „Privatspeicher“. Gemäß dem von-Neumann-Modell werden sowohl das Programm als auch die zu verarbeitenden Daten in den Zentralspeicher geladen. Der Prozessor hat über ein System von Leitungen, den Bus, Zugriff auf diesen Speicher. Weil der Computer ein digital arbeitender Automat ist, sind alle Programmbefehle (Maschinenbefehle) binär codiert, d.h. sie bestehen aus Folgen von Nullen und Einsen (s. Teil III).

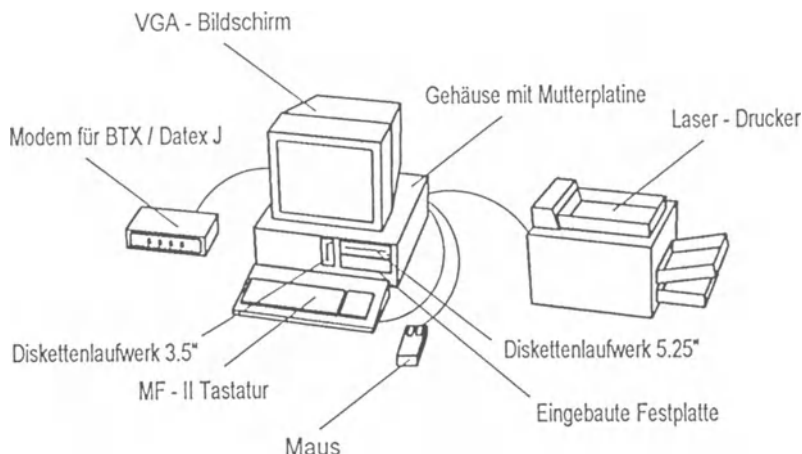
Das Programm wird sequentiell, Maschinenbefehl für Maschinenbefehl, vom Leitwerk interpretiert und ausgeführt. Bei arithmetischen und logischen Operationen zieht es das Rechenwerk zu Hilfe. Dieses wiederum ist beispielsweise in der Lage, zwei Daten zu addieren. Im allgemeinen müssen diese zuvor vom Zentralspeicher über den Datenbus in die Datenregister transportiert werden. Das Ergebnis kann später auf dem gleichen Weg zurück in den Speicher gebracht werden.

Den Zentralspeicher kann man sich als ein „überhohes“ Schubladenelement vorstellen, dessen einzelne Schubladen Daten oder Programmbefehle enthalten können. Jede „Schublade“ (jeder Speicherplatz) wird über seine numerische Adresse angesprochen. Das geschieht per Adressbus.

Das Programm wird mit Hilfe des Programmschrittzählers verwaltet. Dieses spezielle Steuerregister enthält stets die Speicheradresse, unter welcher der Maschinenbefehl zu finden ist, der als nächster vom Leitwerk ausgeführt werden muß.

Zur Speicherung des augenblicklichen Prozessorzustands dient das Zustands- oder Flag-register (*Flag* = Flagge). Die einzelnen Flags zeigen an, ob beispielsweise das Ergebnis der letzten arithmetischen Operation Null ergab (*Zero Flag*), oder ob ein arithmetischer Übertrag notwendig ist, weil das „zu große“ Ergebnis nicht vollständig in ein Datenregister paßt (*Carry Flag*). Einige Prozessoren besitzen weitere Steuerregister.

Ein PC, wie er am Arbeitsplatz oder zu Hause steht, ist typischerweise so konfiguriert:



Alle externen Einheiten sind über Schnittstellen mit der Zentraleinheit verbunden. Der Drucker ist meist über die parallele (Centronics-) Schnittstelle angebunden, die Maus häufig über die serielle V.24-Schnittstelle. Für Bildschirm, Tastatur und externe Speicher existieren spezielle Schnittstellen.

Wer heute einen Computer erwirbt, wird sich in der Regel nicht mit der bloßen Hardware zufrieden geben. Wenn der Computer ein IBM-kompatibler PC ist, wird der Käufer den Händler nach der neuesten Version von MS-DOS fragen. MS-DOS ist ein Betriebssystem, eine Software, die man als Betriebssystem bezeichnet. Erst ein solches Betriebssystem macht für den Normalkunden einen Allzweckrechner zu einem brauchbaren Hilfsmittel. Das Betriebssystem gestaltet den Umgang mit der Computerhardware komfortabel. Ohne Betriebssystem wäre das Einlesen einer Tastatureingabe eine schwierige Aufgabe für einen Spezialisten. Das Betriebssystem „vermittelt“ zwischen der Hardware und der Anwendungssoftware (Spiele, Datenbankprogramme, selbstgeschriebene Programme). Wir können somit für ein Computersystem folgendes einfache Dreischichtenmodell aufstellen: