

Grundlagen der Betriebssysteme

Bearbeitet von
Volkmar Richter

1. Auflage 2004. Buch. 176 S. Hardcover

ISBN 978 3 446 22863 4

Format (B x L): 16,1 x 22,7 cm

Gewicht: 328 g

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](#) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

HANSER

Grundlagen der Betriebssysteme

Volkmar Richter

ISBN 3-446-22863-2

Leseprobe

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/3-446-22863-2> sowie im Buchhandel

1 Geschichte, Entwicklungslinien

Die Entwicklung der Betriebssysteme ist eng an die Entwicklung der Computertechnik geknüpft. Computer wurden anfangs in enger Zusammenarbeit von Technikern und Mathematikern entwickelt und dann auch oft in diesen Teams betrieben. Vor allem Universitäten mussten dann für die Ausbildung sehr schnell eine größere Zahl von Nutzern an diese Technik heranführen.

Der Übergang zur Serienfertigung von Computeranlagen brachte ebenfalls einen größeren Nutzerkreis, dem es nicht mehr zuzumuten war, sich mit der technischen Realisierung im Einzelnen auseinanderzusetzen. Es kam zu einer Arbeitsteilung. Die Herstellung und der Betrieb der Anlagen wurden voneinander getrennt, Rechenzentren bildeten sich, die im Unternehmen oder global Rechenleistungen als Serviceleistungen anboten. Gleichzeitig wurden damit die bis dahin existierenden Rechenbüros und Lochkartenstationen abgelöst. Die Rechenleistung war aber immer noch kostbar, immerhin handelte es sich bei den Rechnern um Investitionsobjekte von mehreren Millionen Dollar, die ganze Säle füllten. Die aus heutiger Sicht geringe Rechenleistung (einige tausend Gleitkommaoperationen je Sekunde) war entsprechend teuer. Für eine Stunde Rechenzeit würde man heute einen kompletten PC kaufen können. Stillstandszeiten mussten also möglichst vermieden werden.

Verantwortlich für den Betrieb waren in dieser Zeit die so genannten Operatoren, die vom Maschinenführer (Operator 1) bis zum Drucker und Magnetband-Bediener (Operator 3) gegliedert, eine ganze Mannschaft bildeten, die im Schichtbetrieb die Anlagen betrieben. Um Fehler zu vermeiden, wurden kleine Hilfsprogramme entwickelt, die zu Monitoren, einer Urform von Betriebssystemen, zusammengefasst wurden. Der englische Begriff **Operating System** setzte sich durch, als durch Einbindung der Mehrprozessarbeit (um die teure CPU besser auszulasten) aus einer ganzen Reihe von Forschungsaktivitäten und Einzelentwicklungen komplexe Systeme entstanden waren. Mit dem OS360 von IBM 1964-1968 kam das erste kommerzielle Multitasking-Betriebssystem auf den Markt und setzte Maßstäbe. Bis heute ist ein kompatibles System z/OS auf dem Markt.

Ausgehend von dem Multiuser-System MULTICS wurde parallel dazu die Familie der UNIX-Betriebssysteme entwickelt, die durch die freie Verfügbarkeit des Codes sehr schnell in Forschung und Lehre Eingang fand, aber auch zu einer enormen „Artenvielfalt“ gelangte. Minicomputer wie die PDP-Serie, die wesentlich billiger und anspruchsloser in Bezug auf Klima und Raumbedarf waren, sowie die Entwicklung von Workstation und PC wären ohne Betriebssysteme nicht denkbar, denn mit der massenhaften Verbreitung mussten auch die Personalaufwendungen reduziert werden. Während sich in den Zeiten der Großrechner die Beschäftigung von Systemprogrammierern lohnte, die das System pflegten und anpassten, wurden mit dem Masseneinsatz des Computers am Arbeitsplatz (Workstations, PC,...) die Anforderungen an die Stabilität und Wartungsfreiheit der Betriebssysteme immer höher.

1.1 TABELLE

Zeittafel ausgewählter Betriebssysteme

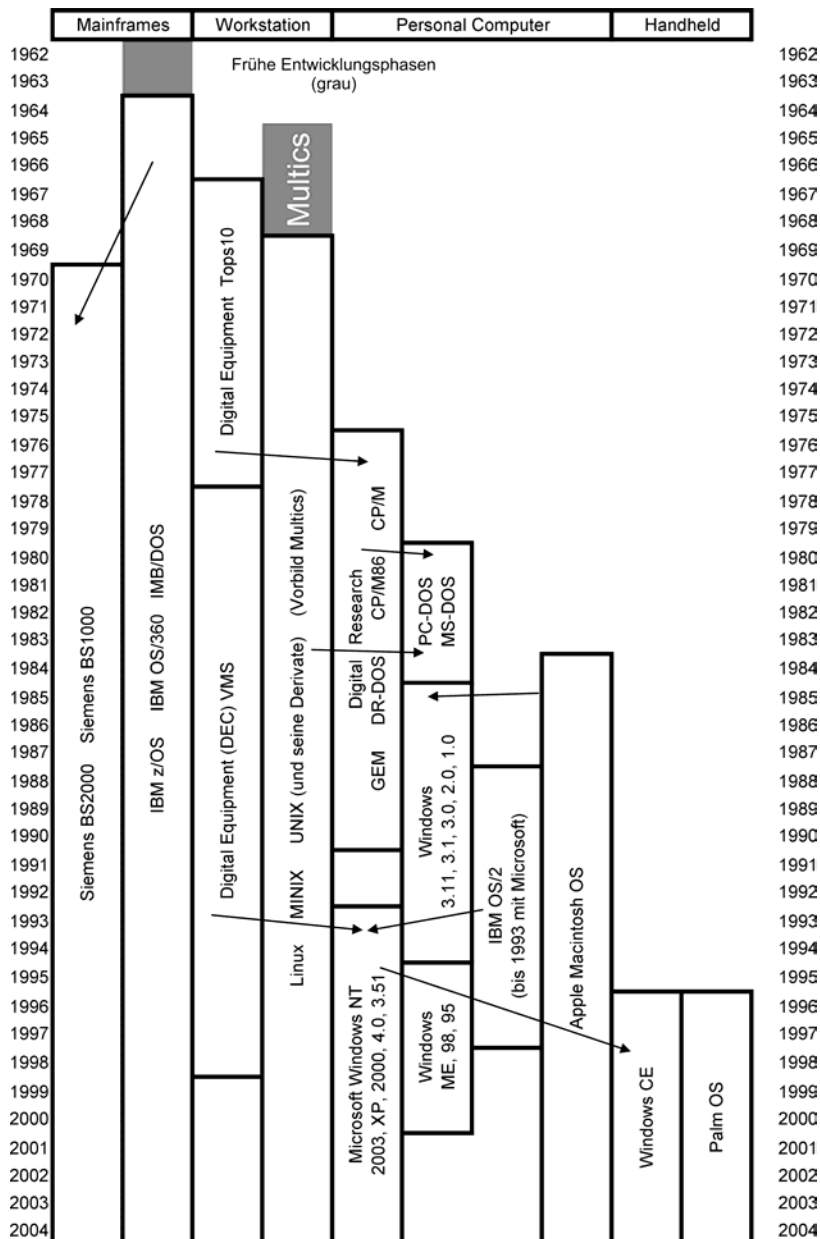


Tabelle 1.1 zeigt eine Zeittafel der Entwicklungsstufen, wobei weder Auswahl noch die eingezeichneten Verwandtschaften den Anspruch auf Vollständigkeit haben. Die Pfeile in der Darstellung sind Verwandtschaften, die durch augenfällige Ähnlichkeiten, aber auch durch Wechsel von Entwicklern zwischen den Firmen naheliegend erscheinen, aber weder zwingend bewiesen sind, noch moralisch oder juristisch bewertet werden sollten. Urheberschutz und Patentierbarkeit von Programmen standen bis Mitte der 70er-Jahre nicht auf der Tagesordnung.

Daniel Ingalls prägte 1981 den Ausspruch: „Ein Betriebssystem ist eine Sammlung von Dingen, die nicht in eine Programmiersprache passen. Deswegen sollte es keines geben.“ Programmiersprachen gehen von einer standardisierten Umgebung aus, in der die zu erzeugenden Programme laufen sollen. Solange es also nicht gelingt, eine universelle Maschine zu entwickeln, muss es gerade deshalb **Betriebssysteme** geben, die diese standardisierte Schnittstelle für die Programme erzeugen.

Abhängig vom Einsatzgebiet differiert die Ausprägung einzelner Eigenschaften der Systeme. Neben der Schnittstelle zur Hardware ist eine Schnittstelle für den Anwendungsprogrammierer erforderlich, wird eine Unterstützung für den Bediener oder Operator gefordert und müssen Informationen zur Abrechnung der erbrachten Leistungen gesammelt werden.

Die weltweite Vernetzung der Systeme und die Dislozierung von Rechen- und Speicherleistungen machte es notwendig, Sicherheitstechnologien in die Systeme zu integrieren, die im Zusammenwirken von Hard- und Software die Ausführung von destruktivem Code (Viren, Würmer, Trojaner,...), aber auch das Ausspähen und Verfälschen von Daten zuverlässig verhindern.

Der enorme Aufwand, der für die Erstellung von Betriebssystemen notwendig ist, wird oft unterschätzt. An der Weiterentwicklung von Linux, arbeiten weltweit tausende engagierte Amateure und zusätzlich noch Mitarbeiter der Firmen, die Linux einsetzen. Das OS360 wurde berühmt, als teuerstes Projekt der Computergeschichte. Frederik P. Brooks beschreibt das Problem des Managements dieser riesigen Anstrengung anschaulich in „Vom Mythos des Mann-Monats“, einem Klassiker der Softwaretechnik. Und die Größe und Bedeutung von Microsoft für die PC-Welt ist allgemein bekannt.

1.1 Architektur von Betriebssystemen

Betriebssysteme bilden die Hardware von Computersystemen in eine dem Nutzer zugängliche Form ab. Diese Funktion beinhaltet es, eine Schnittstelle zu der oftmals komplexen Hardware und eine Schnittstelle zum Nutzer, der sich des Systems bedienen will, zu schaffen. Damit wird jede Computernutzung auch zur Nutzung eines Betriebssystems. Das was uns beispielsweise als PC gegenübertritt ist in Wirklichkeit das Betriebssystem, hinter dem die PC-Hardware verborgen ist.

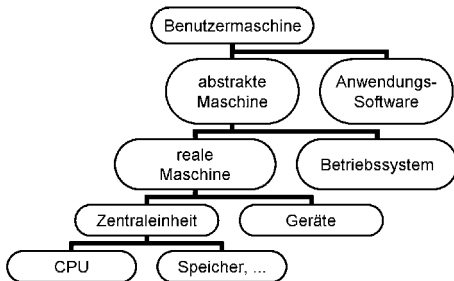
Die Architektur von Betriebssystemen wird von diesen beiden Seiten beeinflusst. Befassen wir uns zunächst mit den grundlegenden Aufgaben eines Betriebssystems:

Verbergen der Komplexität der Maschine vor dem Anwender (Abstraktion)

Bei Coy finden sich dazu die Abstraktionsebenen reale Maschine, abstrakte Maschine und Benutzermaschine, letztere sind in diesem Sinne **virtuelle Maschinen**.

1.1 BILD

Abstraktionsebenen nach Coy



- Bereitstellen einer Benutzerschnittstelle

Nahmen anfangs die Betriebssysteme die Anweisungen des Nutzers in einem Dialog entgegen, der aus einer Eingabeaufforderung (Prompter) und einem Bedienerkommando bestand, das von einem Kommandointerpreter in den Aufruf von Prozeduren umgesetzt wurde, entwickelten sich schnell minimalisierte Programmiersprachen (Shellprogrammierung), die eine teilweise Automatisierung der Bedienung erlaubten. Heute werden grafische Oberflächen eingesetzt, die entweder mit Drag and Drop (Ziehen und Absetzen mit der Maus) oder in Form grafischer Programmierung (Flussdiagramme) die Steuerung erlauben.

- Bereitstellen einer normierten Programmierschnittstelle

Das Application Programmer Interface (API) besteht aus Prozeduraufrufen, die in neueren Systemen auch objektorientiert realisiert sind, speziellen Systemaufrufen, die beim Aufruf gleichzeitig den Zugang zu privilegierten Operationen erlauben und – wenn vorhanden – auch Compiler, Linker, Editor und andere Dienst- und Hilfsprogramme.

- Verwaltung der Ressourcen der Maschine

Die Prozessoren, der Hauptspeicher, aber auch Hintergrundspeicher (Plattenspeicher, Magnetbänder,...) und andere Geräte (Terminal, Drucker, Plotter,...) müssen auf die einzelnen Prozesse, die auf der Maschine laufen, aufgeteilt werden. Bei Hauptprozessor(en) nimmt die Rechenzeit eine spezielle Stellung ein, da nur die Prozessoren Prozesse tatsächlich vorantreiben.

- Verfolgung von Schutzstrategien bei der Ressourcenbereitstellung

Bei der Zuordnung von Ressourcen muss trotz der Konkurrenz der Prozesse um die meist knappen Ressourcen sichergestellt werden, dass keine fremden Zugriffe erfolgen können.

- Koordination von Prozessen

Die Prozesse auf einer Anlage sollen in endlicher Zeit beendet werden können, das setzt eine gewisse Kooperation zwischen den Prozessen voraus. Die Organisation dieser Kooperation wird je nach Anwendungsgebiet mit verschiedenen Methoden realisiert, da beispielsweise Echtzeitanwendungen in der Steuerung von technischen Anlagen andere Anforderungen haben als die Dialogarbeit an einem PC.

- Strukturierung des Systems

Sehr bald bemerkten die Entwickler von Betriebssystemen, dass allein die Ansammlung wesentlicher benötigter Funktionen einen derartigen Umfang annimmt, dass eine Strukturierung unumgänglich ist. Waren die ersten Systeme noch „aus dem Ganzen gefeilt“, so setzte sich sehr schnell eine modulare und damit leicht erweiterbare und anpassbare Struktur durch. Das OS360 wurde beispielsweise, ausgehend von einer umfangreichen Parameterdatei, in stundenlangen Kopier-, Assemblier- und Linkschritten an die vorliegende Maschine angepasst. In jedem Betriebssystem findet sich eine zentrale Kern-Komponente, der **Nucleus** oder **Kernel**, um den sich die restlichen Bestandteile gruppieren. Der Kernel von UNIX wird beim Wechsel der Hardware auch heute noch neu kompiliert. Bei Windows NT gibt es eine modulare Systemumgebung, die dynamisch an die Anforderungen anpassbar ist (z.B. Plug and Play, ab Windows 2000).

1.1 DEFINITION

Betriebssystem - Begriff nach der Norm (DIN 44300)

Das Betriebssystem wird von den **Programmen** eines digitalen Rechensystems gebildet, die zusammen mit den Eigenschaften der **Rechenanlage** die Basis der möglichen **Betriebsarten** des digitalen Rechensystems bilden und die insbesondere die **Abwicklung von Programmen** steuern und überwachen.



Die Funktionen werden hierarchisch von den Hardwarefähigkeiten ausgehend übereinander gestapelt, man erhält eine **Schichtenarchitektur**. Während in der Vergangenheit der Kernel alle wesentlichen Funktionen enthielt, bevorzugt man inzwischen einen Microkernel, der minimalistisch ausgestattet, eine größere Flexibilität des Betriebssystems ohne erneute Kompilierung oder Assemblierung ermöglicht. Die Entwicklung ist noch lange nicht abgeschlossen, neue Maschinenkonzepte, neue Vernetzungsmethoden und steigende Rechen- und Speicherleistungen, aber auch neue Angriffsszenarien erzeugen ständig neue Anforderungen an zukünftige Betriebssysteme.

Aktuelle Herausforderungen für die Systementwickler sind:

- Betriebssysteme müssen effektiv an die Bedürfnisse von Anwendungen anpassbar sein und die Erweiterung und Wartung von Betriebssystemen muss einfacher werden.
- Effizienz, Schutzmechanismen, Zugriffs- oder Cache-Strategien sind zu verbessern.
- Ein höheres Abstraktionsniveau bei der Programmierung muss erreicht werden.
- Betriebssysteme für verteilte Systeme, Mehrprozessorsysteme und Parallelrechner sind effektiver zu gestalten.

- Die Anwendungen dürfen nicht gezwungen sein, Betriebssystemmechanismen zu umgehen.

Zum letzten Punkt ist noch zu ergänzen, dass oft der gut gemeinte Schutz (vor was auch immer) den Anwender vor Probleme stellt, die er nur mit sanfter Gewalt lösen kann, denn sowohl der Wechsel der Hardware als auch des Betriebssystems ist selten machbar. Erst vor kurzem kauften wir eine Netzwerklizenz eines Foto-Bearbeitungs-Programms, also einer typischen Anwendersoftware, die davon ausging, dass der Nutzer Administratorrechte hätte. Wer hat Schuld, der Systementwickler, der sein System unangreifbar machen wollte, oder der Entwickler, der sich nicht mit den Schutzmechanismen auseinandersetzen wollte oder mangels Unterlagen nicht konnte? Systementwickler, Systembetreuer, Administratoren müssen sich stets als Dienstleister verstehen. Den Anwender abzuqualifizieren und zu gängeln ist nicht sinnvoll. Für jede notwendige Einschränkung ist eine verständliche Begründung erforderlich und ein Werkzeug, um mit den daraus entstehenden Einschränkungen ohne Effektivitätsverlust umgehen zu können.

1.2 Hardwaresysteme, Softwaresysteme

Für die Betrachtung komplexer Anlagen hat sich der Systembegriff bewährt. Als **System** wird ein nach Aufgabe oder Wirkung abgrenzbares Gebilde betrachtet. Wichtig ist, ein System immer als Funktionseinheit zu betrachten. Das gesamte System wird in **Subsysteme** unterteilt. Jedes Subsystem hat Teilfunktionen zu erbringen. Subsysteme können weitgehend unabhängig voneinander weiterentwickelt werden. Dabei ist wichtig, dass sich die Subsysteme später untereinander verstehen.

Die Einteilung in Systeme und Subsysteme ist willkürlich, man strebt dabei aber möglichst einfach zu beschreibende Einheiten an. Subsysteme, die aus dem gewählten Betrachtungsaspekt nicht mehr sinnvoll teilbar sind, werden als Atome bezeichnet. Jedes System hat auch eine Umgebung, diese Umgebung kann als Hüllsystem betrachtet werden, in der das betrachtete System ein Subsystem ist. In der Informationstechnologie ist die Hülle entweder ein technisches System oder ein soziales System.

1.2.1 Hardwaresysteme

Von Anbeginn hatten sich zwei klassische Strukturen herausgebildet, die heute als **Harvard Architektur** und als **Von-Neumann-Architektur** bezeichnet werden. Grundlegender Unterschied ist, dass die Harvard Architektur den Speicher für Programme von dem Speicher für Daten trennt. Das hat technische Vorteile, benötigt aber eine zusätzliche Verbindung für die Bearbeitung von Programmdaten. Aus der Sicht eines Compilers oder Assemblers sind die erzeugten Programme nur Daten, die erst in dem Moment zu Programmtexten werden, in dem ein Prozessor sie zu seiner Steuerung benutzt. Man spricht hier von einer Dualität zwischen Programmen und Daten.

Für Spezialanwendungen wurden weitere Strukturen entwickelt, die aber aufgrund verschiedener problematischer Aspekte keinen breiten Einsatz finden.

1.2 DEFINITION**Harvard-Architektur**

Rechnerarchitektur, bei der das Rechenwerk auf einen Datenspeicher und das Steuerwerk auf einen Programmspeicher zugreift. Durch den gleichzeitigen Zugriff ist eine schnelle Arbeitsweise möglich.



Mikroprozessoren bevorzugen diese Architektur teilweise, haben dann aber Probleme mit der Vielzahl von Buspins. Die Super-Harvard-Architektur von Analog Devices benutzt in ihren Signalprozessoren Dual-Port-Speicherbausteine, die an beide Busse angeschlossen sind und kombiniert so die Vorteile dieser Architektur mit der Von-Neumann-Architektur.

1.3 DEFINITION**Von-Neumann-Architektur**

bezeichnet eine Rechnerarchitektur, die für Daten und Instruktionen den gleichen physischen Speicher benutzt. Sie wurde von dem Mathematiker John von Neumann 1944 vorgeschlagen. Der Entwurf war für die Maschine EDVACgedacht, die von 1951 bis 1961 im Einsatz war.

**1.4 DEFINITION****Datenflussrechner**

benutzen hintereinander geschaltete arithmetische oder logische Verknüpfungseinheiten, die von den Daten sukzessiv durchlaufen werden, und verbinden eine extrem hohe Leistung mit einer starren Programmierung. Digitale Signalprozessoren orientieren sich teilweise an dieser Arbeitsweise.

**1.5 DEFINITION****Parallelrechner**

besitzen eine Vielzahl von arithmetisch logischen Einheiten (ALU), die in der Lage sind, Berechnungen von Vektoren und Matrizen parallel für ganze Gruppen von Werten durchzuführen.

**1.6 DEFINITION****Massiv parallele Systeme**

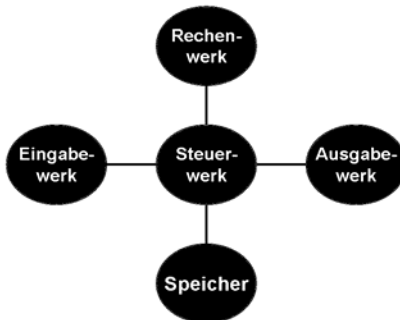
bestehen aus einer Vielzahl von selbstständigen CPU (mit Speicher), die über schnelle Kommunikationskanäle verbunden sind. Die Idee eines Transputers (z.B. T9000), auf einem Chip CPU, Speicher und 4 I/O-Kanäle unterzubringen, die zu beliebig großen Matrizen zusammengeschaltet werden können, entstand Mitte der 80er Jahre, wurde aber nicht weiter verfolgt, da die Wärmeabfuhr bereits bei geringer CPU Zahl problematisch war (geplant waren bis 65536 Prozessoren), und die Leistung von Singlechip-Mikroprozessoren ständig stieg.



Ein Computer mit Von-Neumann-Architektur besteht aus fünf Funktionseinheiten, dem Steuerwerk, dem Rechenwerk, dem Speicher, dem Eingabe- und Ausgabewerk sowie einem Verbindungssystem zwischen diesen Einheiten. Die Struktur des Von-Neumann-Rechners ist unabhängig von den zu bearbeitenden Problemen. Zur Lösung eines Problems muss von außen eine Bearbeitungsvorschrift, das Programm, eingegeben und im Speicher abgelegt werden. Ohne dieses Programm ist die Maschine nicht arbeitsfähig. Programme, Daten, Zwischen- und Endergebnisse werden in demselben Speicher abgelegt.

1.2 BILD

Struktur der Von-Neumann-Maschine



Der Speicher ist in gleichgroße Zellen unterteilt, die fortlaufend durchnummeriert sind. Über die Nummer (Adresse) einer Speicherzelle kann deren Inhalt abgerufen oder verändert werden. Aufeinander folgende Befehle eines Programms werden in aufeinander folgenden Speicherzellen abgelegt. Das Ansprechen des nächsten Befehls geschieht vom Steuerwerk aus, durch Erhöhen der Befehlsadresse um eins. Sprungbefehle erlauben, von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abzuweichen.

Es gibt verschiedene Gruppen von Befehlen, meist sind vorhanden:

- arithmetische Befehle wie Addieren, Multiplizieren, Konstanten laden,...
- logische Befehle wie Vergleiche, logisches NICHT, UND, ODER,...
- Transportbefehle, z.B. vom Speicher zum Rechenwerk und für die Ein-/Ausgabe
- bedingte Sprünge
- sonstige Befehle wie Schieben, Unterbrechen, Warten,...

Alle Befehle können in verschiedenen Adressierungsarten ausgeführt werden, die Daten (Befehle, Adressen usw.) werden binär codiert, abgesehen von frühen Konstruktionen mit Dezimalrechenwerken. Logische Schaltwerke im Steuerwerk und den anderen Komponenten sorgen für die Entschlüsselung (Decodierung).

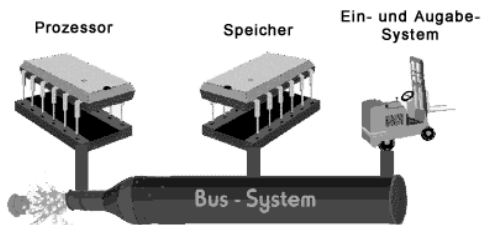
Eine echte Von-Neumann-Architektur wird selten verwendet. Meist wird ein zusätzlicher Schritt zur Reaktion auf Ereignisse in der Umgebung eingebracht, die asynchron zum Programmablauf eintreten. Diese Interruptbearbeitung ermöglicht es dem Computer, die

Wartezeit auf Ereignisse sinnvoll zu nutzen, und ist für effektive Betriebssysteme essentiell.

Da alle Transporte von Daten und Befehlen über denselben Speicherbus gehen, bezeichnet man diesen als Flaschenhals des Von-Neumann-Rechners und versucht durch erhöhte Taktfrequenzen, Busbreiten und schnellere Cachespeicher, die hohe Last auf diesem Kanal zu bewältigen.

1.3 BILD

„Flaschenhals“ der Von-Neumann-Maschine



Das Steuerwerk arbeitet mit einer Schleife von (mindestens) 5 Schritten. Aus Zeitgründen wird dabei oft versucht, eine Überlappung der Schritte zu erreichen, indem man beispielsweise den nächsten Befehl schon während der Ausführung der Operation liest. Diese vorausschauende Arbeitsweise kann zwar hin und wieder zu Ergebnissen führen, die der Prozessor wieder verwerfen muss, steigert aber insgesamt die Leistung.

1.2 TABELLE

Zentrale Steuerschleife eines von Neumann Rechners

1. Fetch	Der Befehl wird aus dem Speicher gelesen und in das Befehlsregister eingeschrieben.
2. Decode	Der Befehl im Befehlsregister wird entschlüsselt.
3. Load	Die Operanden werden aus dem Speicher gelesen und in die Register des Rechenwerks eingetragen.
4. Execute	Die Operation wird im Rechenwerk ausgeführt.
5. Store	Das Ergebnis wird aus dem Ergebnisregister des Rechenwerks in den Speicher geschrieben.
0. Cycle	Nach der eventuellen Prüfung auf Interrupts beginnt die Schleife von vorn.



1.2.2 Softwaresysteme

Auch die zu einem System gehörende Software kann als Subsystem betrachtet werden, dabei unterscheidet man zwischen Anwendungssoftware und Systemsoftware. Auf die

Unterscheidung in Firmware (in Hardware integriert), Middleware (Verbindung verteilter Systeme – Vermittlungssoftware also) und „echter“ Software wird in diesem Buch verzichtet.

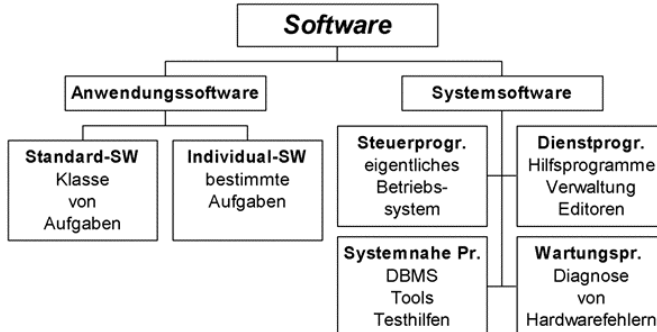
1.7 DEFINITION**Systemsoftware**

Zur Systemsoftware zählen alle Systemprogramme, die nicht anwendungsbezogen, sondern für den Betrieb grundsätzlich erforderlich sind (Betriebssystem).

Dabei wird das Betriebssystem im eigentlichen Sinn, das Steuerprogramm des Computers also, durch eine Reihe von Ergänzungen unterstützt. (Bild 1.4)

1.8 DEFINITION**Anwendungssoftware**

Alle anwendungsbezogenen Programme (Textverarbeitung, Datenbanken, Bildbearbeitungsprogramme,...) fasst man unter dem Oberbegriff Anwendungssoftware zusammen.

1.4 BILD**Softwareklassen**

1.3 Modelle von Betriebssystemen

1.3.1 Schalenmodelle

Der hierarchische Aufbau von Betriebssystemen legt es nahe, als Modell ein Schichten- oder Schalenmodell zu benutzen, bei dem die inneren Schichten von den äußeren verdeckt (gekapselt) werden. Jede dieser Schichten (im Englischen „**shell**“) besteht dann aus Segmenten, die einzelne Aufgaben in dieser Schicht realisieren. Im Inneren befindet