

2 Übersicht

2.1 Wiederholung: Computer Hardware

Einfacher PC (Hardware Schema):

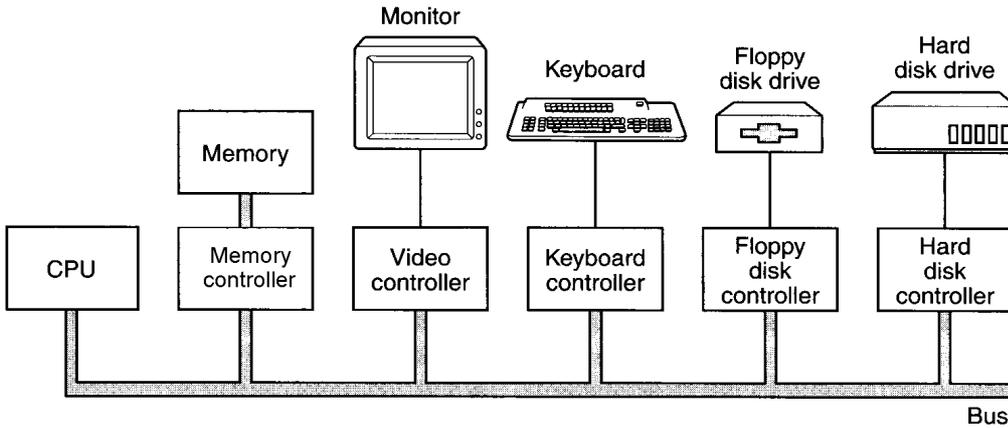


Abbildung 2: Übersicht Rechneraufbau

2.1.1 Funktionsweise einer "von Neumann" - Maschine

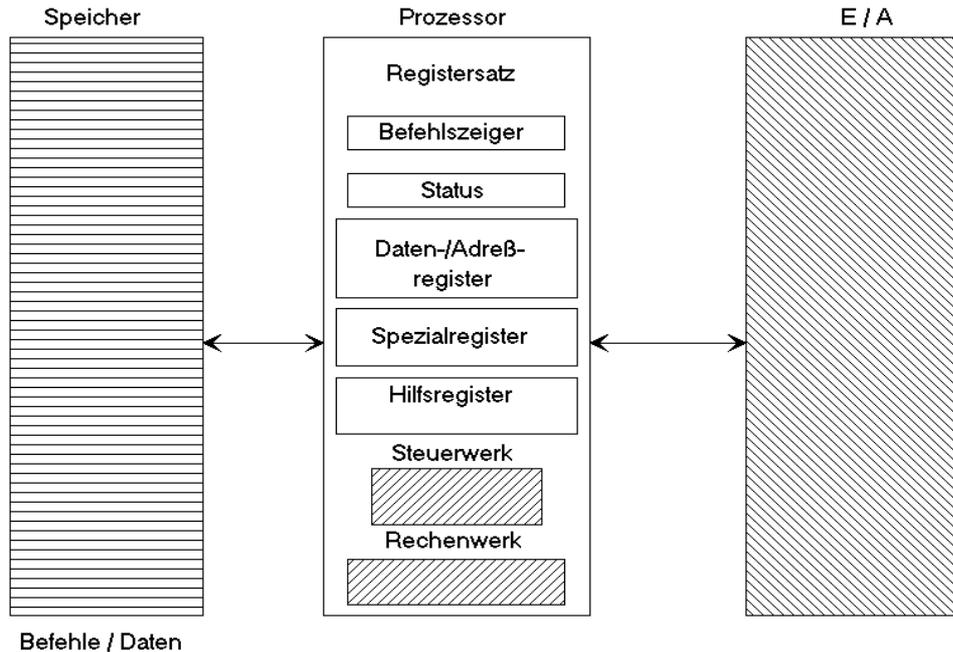


Abbildung 3: Übersicht "von Neumann" Maschine

Speicher: Befehle und Daten
 Prozessor: Grundzyklus zur Befehlsbearbeitung

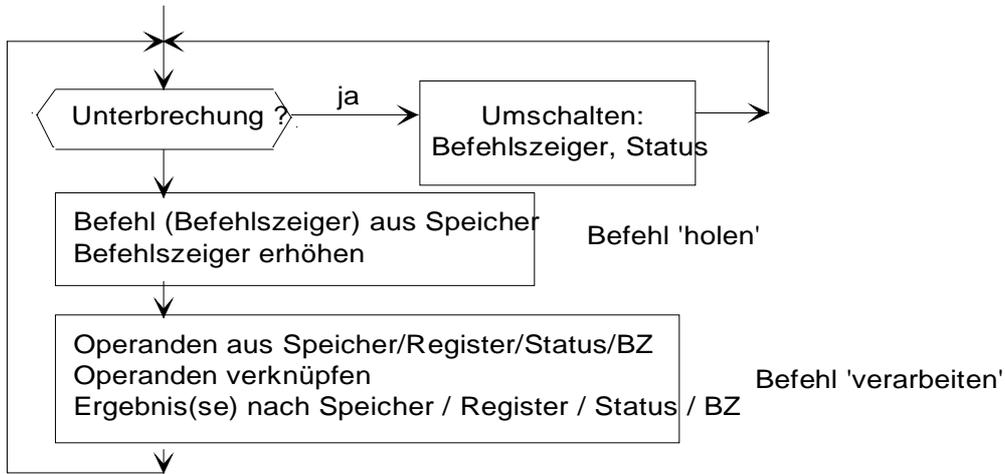


Abbildung 4: Programmausführung

Programmausführung:

- Prozeß aus Befehlsbearbeitungen
- Prozeß benötigt Betriebsmittel: Prozessor, Speicher, E/A-Geräte

2.1.2 Prozessor

Moderner Prozessor: a) Pipeline b) superscalar pipeline

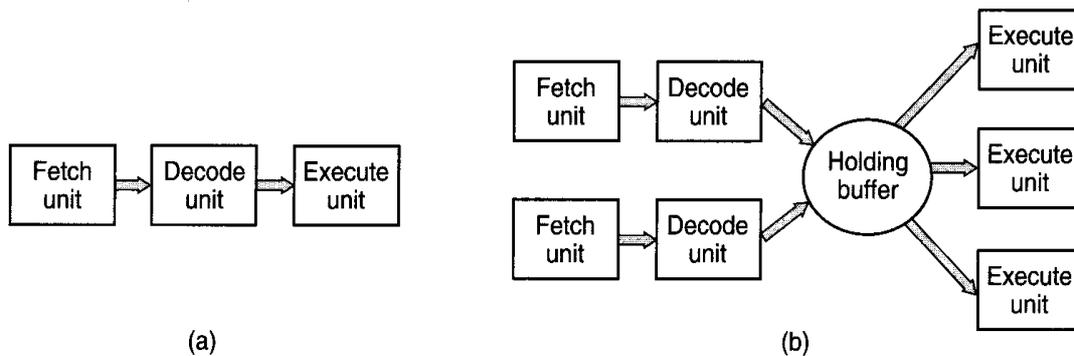


Abbildung 5: Pipeline, Superskalar Pipeline

2.1.3 Zentralspeicher (Memory)

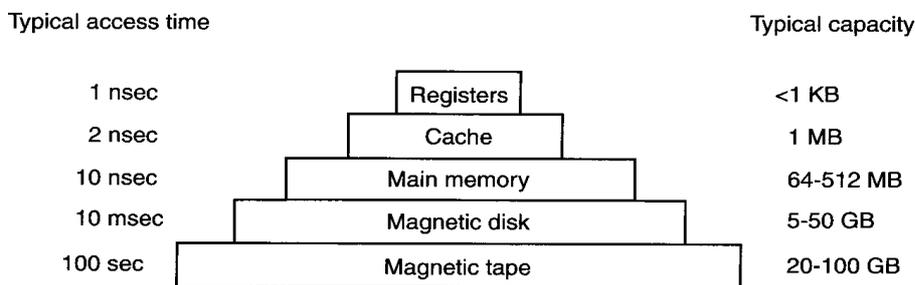


Abbildung 6: Zugriffspyramide

In embedded-Systemen: EEPROM als permanenter Speicher
 Neueste Entwicklung: MRAM magnetic RAM (persistent)

2.1.4 I/O Geräte

Steuerung von I/O Geräten. Eine Übersicht über typische I/O-Geräte gibt die folgende Tabelle:

Tabelle 1: Übersicht I/O Geräte

Gerät	Datenrate	Gerät	Datenrate
Tastatur	10 Bytes/s	Fast Ethernet	12,5 MB/s
Maus	100 B/s	ISA Bus	16,7 MB/s
56K Modem	7 kB/s	EIDE (ATA-2) Festplatte	16.7 MB/s
ISDN mit Kanalbündelung	16 kB/s	FireWire (IEEE 1394)	50 MB/s
Laserdrucker	100 kB/s	XGA Monitor	60 MB/s
Ethernet	1,25 MB/s	SCSI Ultra 2 Festplatte	80 MB/s
USB	1.5 MB/s	Ultrium Tape	320 MB/s
IDE Festplatte	5 MB/s	PCI Bus	528 MB/s
40x CDROM	6 MB/s	Sun Giagplane XP Backbone	20 GB/s

2.1.5 Datenbusse

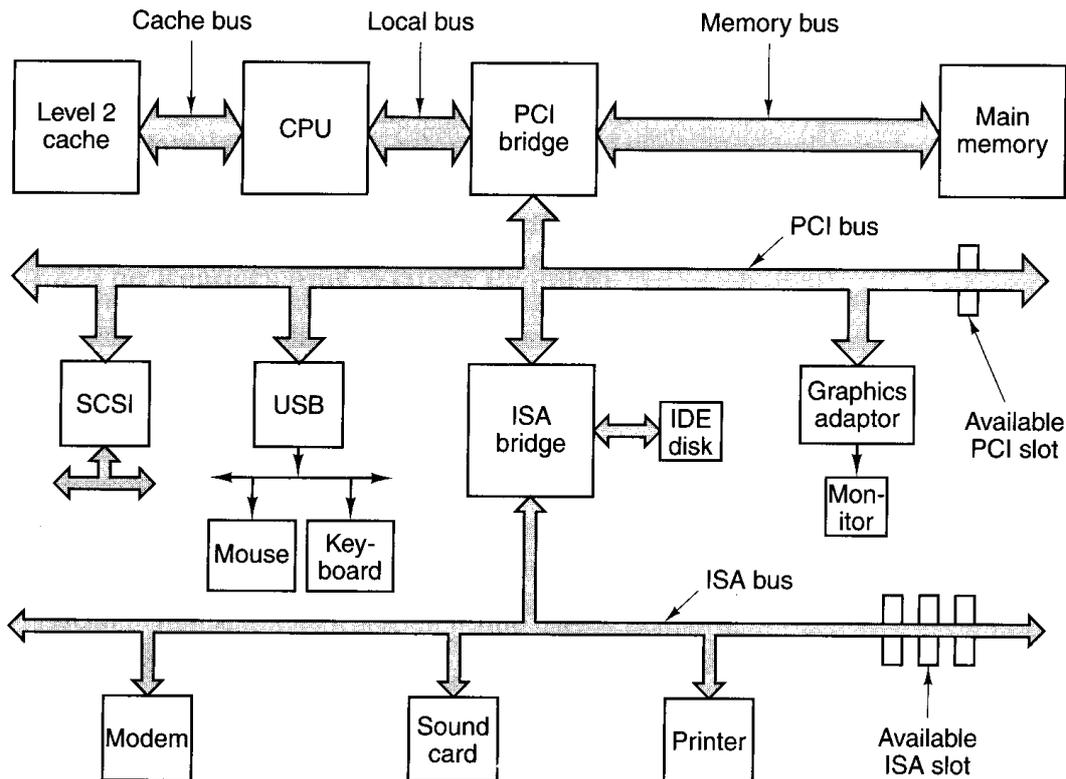


Abbildung 7: Pentium Busstruktur (aus [1])

2.1.6 Multi-Prozessor-Architekturen

Mehr-Prozessor-Rechner:

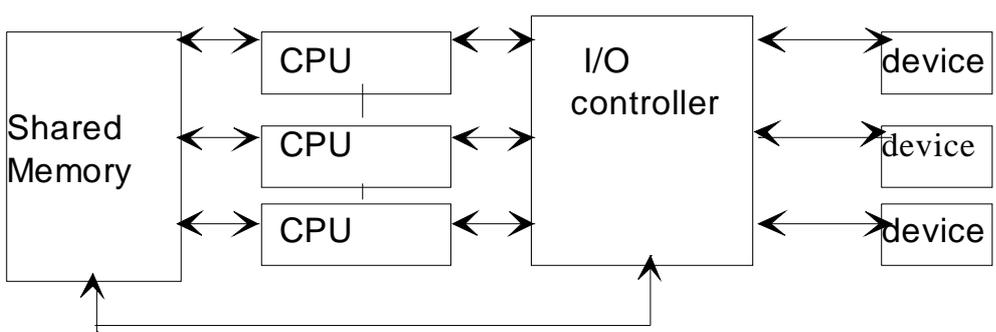


Abbildung 8: Schema Rechner

Verschiedene Arten von Mehrprozessorsystemen:

a) Shared Memory: Alle Prozessoren greifen auf denselben Speicher zu (als Pentium Hyperthreading auch im PC Standard); max. 16-32 CPUs

- Windows Server 2003:
 - Enterprise Edition: 8 CPUs
 - Datacenter: 128 CPUs
- Solaris 9:
 - *One million simultaneous processes on a single system*
 - *Up to 128 CPUs in a single system and 848 CPUs in a clustered environment*
 - *Support for up to 576 GB of memory*
 - *More than four billion network connections*
 - *Up to 252 TB file systems with Sun StorEdge QFS*
 - *IPv6, enabling a 128-bit IP address space*
 - *A 64-bit Java virtual machine*

b) Message-Passing: Jeder Prozessor hat eigenen Speicher, Austausch über eine Kommunikationsmatrix (Crossbar)

c) Wide-Area Distributed System: Stand-Alone Systeme, die über ein WAN verbunden sind

- Grid-Computing

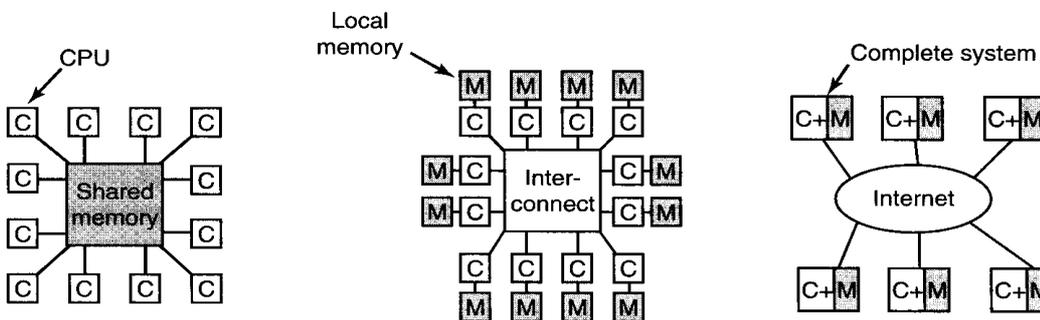
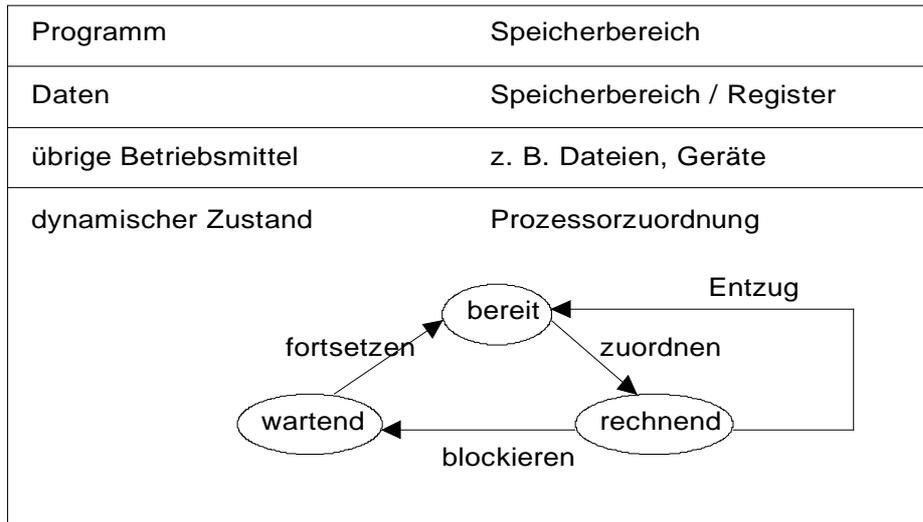


Abbildung 9: Mehr-Prozessor-Systeme: Shared Memory, Interconnected, Distributed

2.2 Aufgaben, Dienste und Komponenten von Betriebssystemen

2.2.1 Prozeß- und Prozessorverwaltung

Prozeß



Prozessorzuteilung

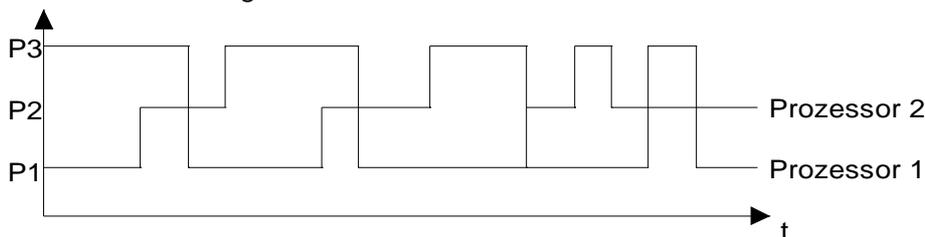


Abbildung 10: Prozess-Zustände, Prozessorzuteilung

- logische Prozesse (Erzeugung, Löschung)
- Prozessorzuordnung
- gegenseitige Beeinflussung von Prozessen, Synchronisation
- Ablaufsteuerung, betriebsmittelabhängige Planung und Kontrolle des Gesamtablaufs
- Interprozess-Kommunikation:
 - a) Message Passing
 - b) Shared Memory

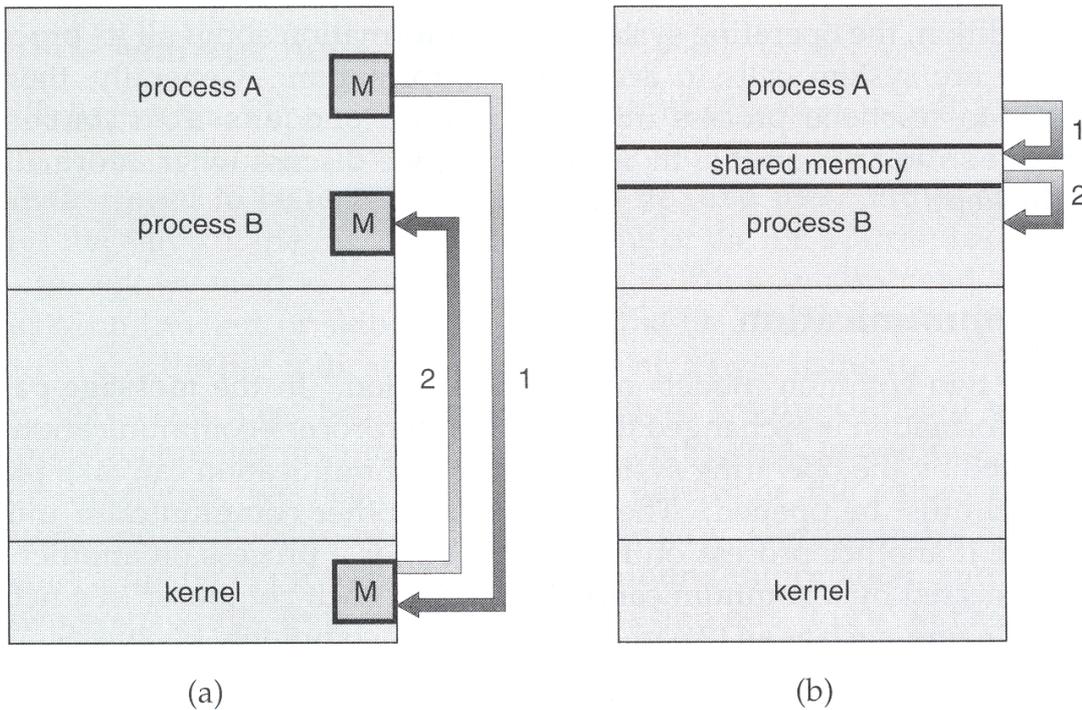


Abbildung 11: Interprozess-Kommunikation (a) Message Passing, (b) Shared Memory

2.2.2 Programmverwaltung

(übersprungen)

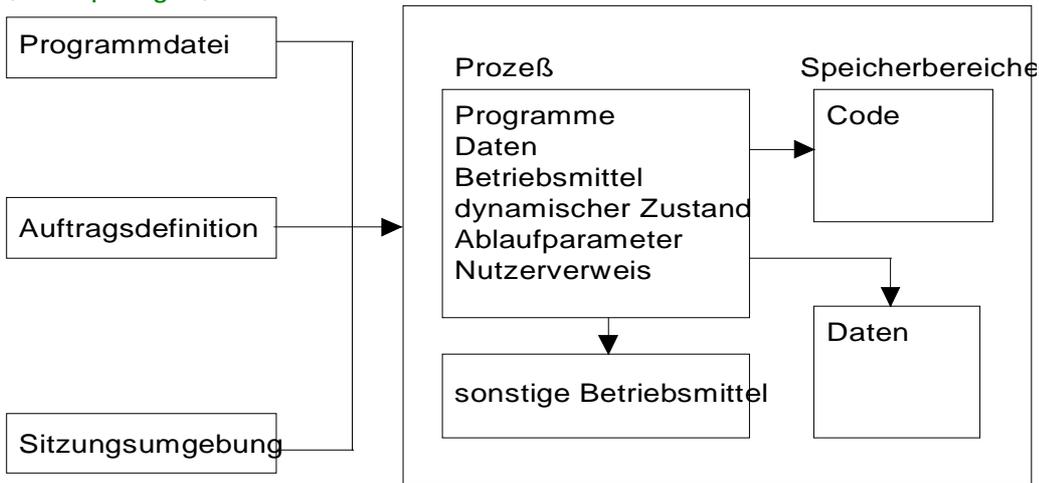


Abbildung 12: Übersicht Programmverwaltung

- Definition von Programmen
- Zuordnung von Programmen zu Prozessen
- Abbildung des Programmablaufs auf Prozesse
- Besonderheiten
 - mehrfach-verwendbarer Code
 - dynamisches Binden

- z.B. shared Libraries (.dll unter Windows)
- Komponenten-Middleware COM/DCOM, CORBA

2.2.3 Zentralspeicherverwaltung

Virtuelle Speicherverwaltung (Paging)

- Jeder Prozess hat eigenen, virtuellen Speicher
- Memory Management Unit setzt jeden Adress-Zugriff von der virtuellen Adresse auf die physikalische um -> eigene MMU-Hardware nötig
- Ist der physikalische Speicher ausgeschöpft, können Seiten auf die Festplatte ausgelagert werden. Greift ein Prozess auf diese Seiten zu, so wird ein Seitenfehler ausgelöst -> die Seite muss von Festplatte nachgeladen werden

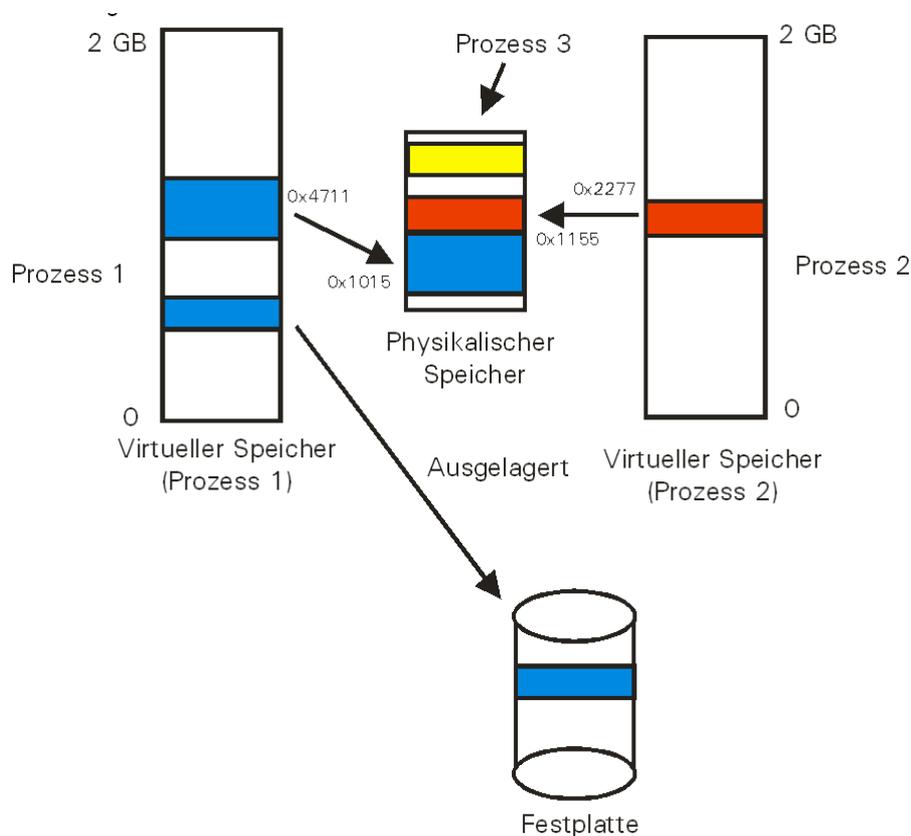


Abbildung 13: Übersicht virtuelle Speicherverwaltung

Aufgaben:

- Belegung, Freigabe und Zuordnung von verschiedenen Speicherbereichen an Prozesse
- Verwaltung des Zentralspeichers
- gegenseitiger Zugriffsschutz
- virtuelle Adreßräume

2.2.4 Ein-/Ausgabe-Steuerung

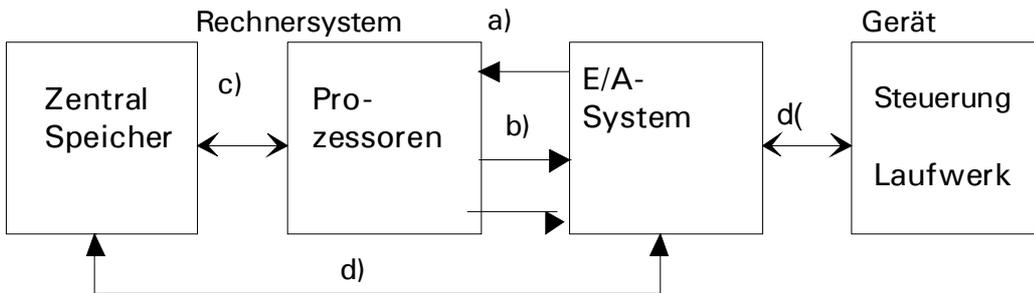


Abbildung 14: Schema E/A Steuerung

Datenfluss:

- a) Ein/Ausgabeunterbrechungen
- b) Ein/Ausgabebefehle
- c) Interprozessorsignale
- d) Daten
- e) Busprotokoll:
 - Bus Befehle, Gerätedaten (Adresse, Status), Transferdaten

Aufgaben:

- Geräte-/Konfigurationsverwaltung
 - * Verkabelung
 - * Parameter
- physikalische Ein-/Ausgabe
 - * Gerätesteuerung
 - * Datentransfer
- logische Geräte
 - * z.B. Dateien
 - * Abbildungs- und Zugriffsparameter
- logische Ein-/Ausgabe
 - * Dateiorganisation
 - * Dateizugriffe
- Betriebsmittelverwaltung
 - * Zuordnung zu Prozessen

2.2.5 Externe Datenverwaltung

(übersprungen)

2.2.5.1 *Physikalische Speichermedien*

- * Magnetplatte (Schreib-/Lesekopf, Sektor, Spur)
- * Magnetband (Block, Spur, Sprosse)

Dateien

- *logischer Adressraum der Datei
 - Satz: logische Zugriffseinheit für Ein/Ausgabe
 - Logische Zugriffsstruktur: Satz-Schlüsselfeld

Feld: vom Programm auswertbarer Teil eines Satzes
*Physikalischer Adressraum der Datei

Block: physikalische Zugriffseinheit für Ein/Ausgabe

Dateiorganisation

* Darstellung: sequentiell, indexsequentiell, direkt ...

* Zugriffsmethoden: sequentiell, indexsequentiell, physikalisch ..

* Verwaltung:

Statisch: Dateikataloge, Speicherplatz, ...

Dynamisch: Sperrenverwaltung, ...

2.2.6 Betriebsmittelverwaltung (Resource Management)

- Betriebsmittel
 - physikalisch (z.B. Festplatte) – logisch (z.B. Datei)
 - aktiv (z.B. Prozessor, Festplattencontroller) - passiv (z.B. Drucker)
 - exklusiv (z.B. CD Brenner) – teilbar (z.B. Netzwerk)
 - unterbrechbar (z.B. Speicherbelegung: kann gewappt werden) – ununterbrechbar (z.B. CD-ROM Brenner)
- Anforderung, Belegung, Freigabe
- Fremdzuweisung, Entzug
- Wettbewerb um Betriebsmittel: Achtung Verklemmung (Deadlocks)!
 - Beispiel Deadlock: ein Programm fordert zunächst den Drucker, dann eine Datei „message.log“ an, um eine Meldung auszugeben. Ein anderes Programm fordert zunächst die Datei „message.log“ an, dann den Drucker -> Verklemmung möglich!

Anderes Beispiel:

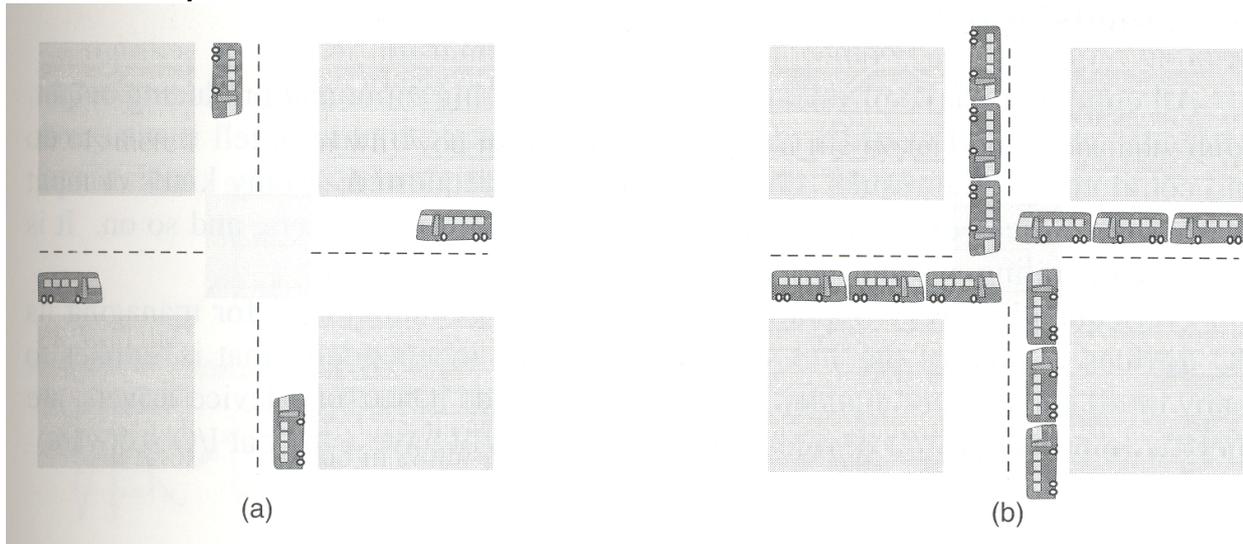


Abbildung 15: Deadlock an der Kreuzung

- Auswahl- und Zuteilungsstrategien
- Verklemmungen (Verhinderung, Entdeckung, Auflösung)

2.2.7 Zugriffsschutz (Security)

Gegenseitiger Schutz der Prozesse :

- Speicher auslesen, überschreiben
- Fremde Dateien lesen, manipulieren
- Tastatureingaben, Bildschirmausgaben

-> Rechtesystem „erlaubt“ oder „verbietet“ Zugriff auf Objekte des BT, z.B. Dateien, Programme,

2.2.8 Kommando-Interpreter (Shell)

Direkter Zugang des Benutzers zum BT, Dateisystem

è Teil des BT, z.B. DOS, Windows

è Fremdprodukt, z.B. bei Unix „bash“

2.3 Betriebssystem Strukturen

2.3.1 Einfache Strukturen

2.3.1.1 Beispiel MS-DOS

- Sehr einfache Struktur
- User-Programme greifen direkt auf BIOS (hardware) zu, keine Kontrolle, kein Schutz
- Nur 1 Programm gleichzeitig möglich
 - Jedoch „speicher-resistente Programme“ TSR (terminate and stay resident) möglich: diese TSR installieren sich in einer Interrupt-Kette, z.B. bei Tastatureingaben, lesen alle Tastatureingaben & werden bei Bedarf aktiv.

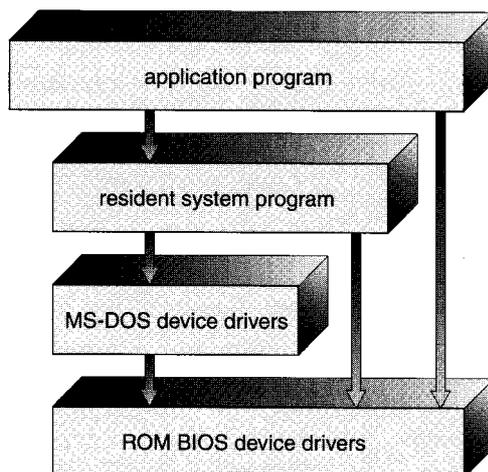


Abbildung 16: DOS Struktur

2.3.1.2 Beispiel: Unix

- Großer, monolithischer Kernel der das gesamte BT enthält: ein großes ausführbares Programm
- 2 Schichten bzw. Modi: User und Kernel; Definierter Übergang vom User-Space in den Kernel Space via System Trap:

- Typischerweise Software-Interrupt mit Parameterübergabe (Register, Stack, gemeinsamer Speicher)
- Parameter werden genau geprüft
- Im User-Mode nur beschränkter Zugriff auf Hardware-Ressourcen:
 - *Prozessor-Interrupts* können nicht abgeschaltet werden -> Prozess schaltet Scheduler ab!
 - Kein Zugriff auf *Interrupt-Vektortabelle* -> Prozess verbietet Scheduler-Timer-Interrupt auf sich selbst -> Scheduler geht nicht mehr
 - *Kein direkter Speicherzugriff* -> User-Prozess zerstört Kernel-Datenstrukturen, löscht z.B. alle anderen Prozesse
 - *Kein Zugriff auf I/O* möglich -> User-Prozess stört andere I/O Vorgänge
 - *Hardware-Unterstützung nötig*: wenn der Prozessor den User-Mode nicht unterstützt, kann man keine saubere Trennung implementieren
- Typischerweise 5 Bereiche:
 - Prozess-Kontrolle
 - Datei-Verwaltung
 - Geräte-Verwaltung
 - Informationsaustausch, z.B. date / time Funktionen
 - Interprozess-Kommunikation

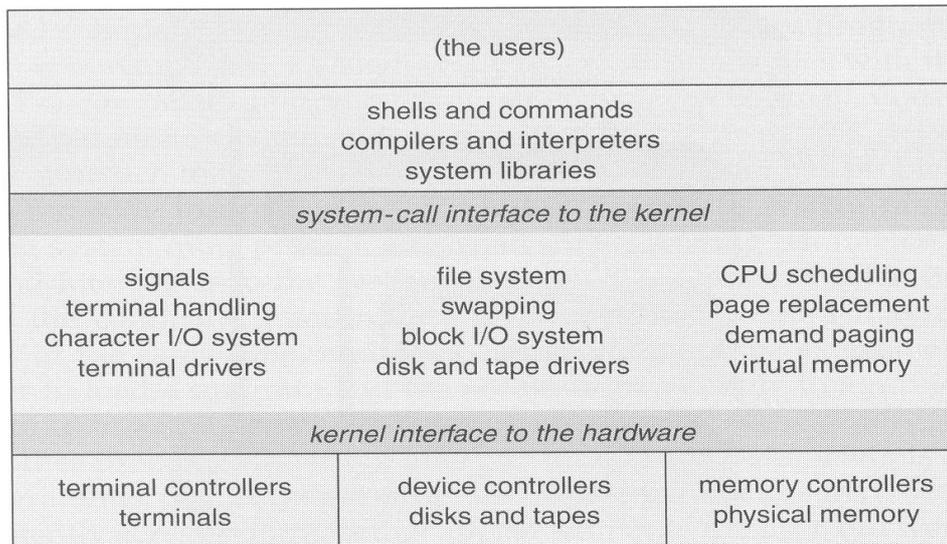


Abbildung 17: Unix Struktur (siehe Folie)

2.3.2 Microkernel

- è Carnegie Mellon University entwickelt das BT „Mach“ mit Mikrokernel Ansatz:
 - Alle nicht unbedingt benötigten Komponenten werden aus dem Kernel entfernt und als User- oder Systemkomponenten implementiert

- Kernel stellt Kommunikationsdienste zwischen den Komponenten zur Verfügung, z.B. durch Message Passing
- Dadurch sehr kleiner Kernel möglich
- è Vielfach in anderen Systemen verwendet, z.B. Tru64 UNIX, MacOS X Server; hybrider Ansatz in WinNT
- è Teilweise in XP verwirklicht

2.3.2.1 Beispiel Windows XP

- Hardware Abstraction Layer
- Kernel: Non-Preemptive -> muss deshalb klein und schnell sein
- Executive
- Win32 Subsystem: Programmierer-Schnittstelle zum OS; einige tausend Funktionen
- User mode / kernel mode
- Window Manager im Kernel -> bei Unix typischerweise im User-Mode (X11 Server)

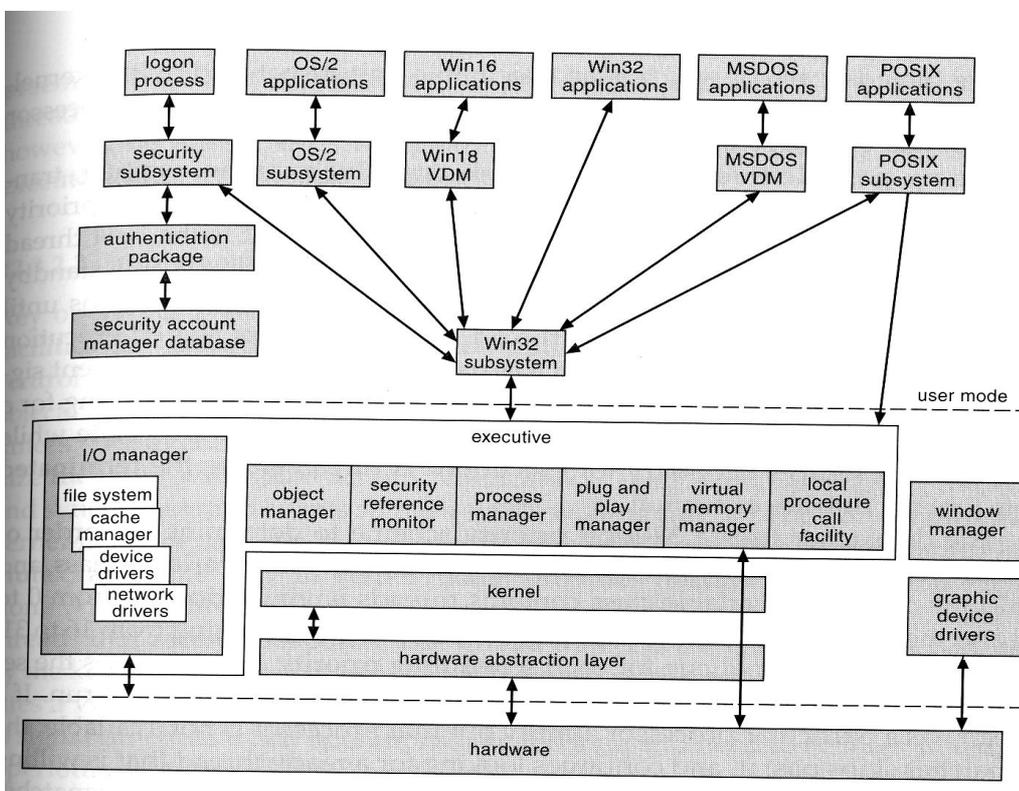


Abbildung 18: Windows Übersicht

2.3.3 Konzept virtueller Maschinen

2.3.3.1 Übersicht VM

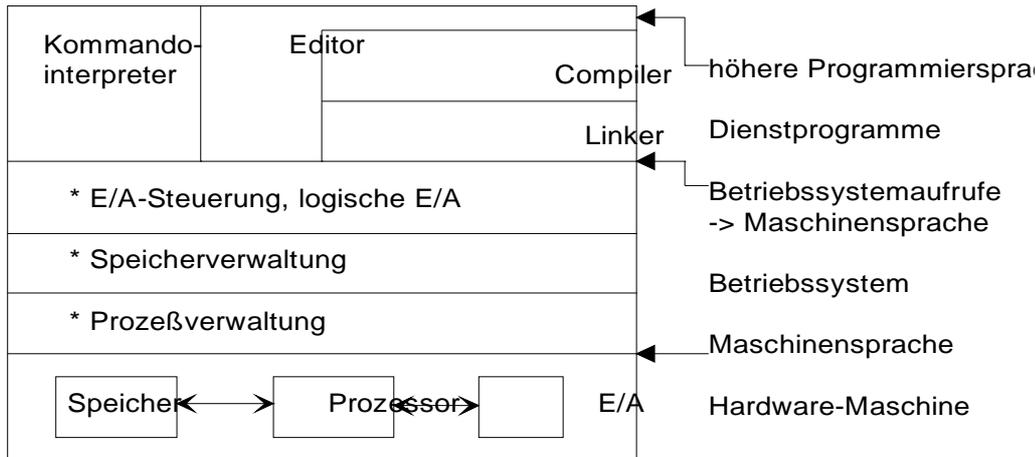


Abbildung 19: Aufbau BT

Maschine:

Mechanismus zur Interpretation von in Maschinensprache formulierten Programmen.

Virtuelle Maschine:

Durch interpretierende oder abbildende Software auf Basis einer gegebenen Maschine definiert.

Hierarchie virtueller Maschinen:

Hardware und Software-Schichten definieren Maschinen mit steigendem Sprachniveau.

è Strenges Schichtenmodell

- Jeder Layer greift nur auf den nächst-niedrigeren Layer zu
- der untere Layer stellt eine vollständige „Maschine“ zur Verfügung
- unterschiedliche Definitionen in der Literatur, z.B. Schichtenmodell nur bei VM

è Erstmals in IBM OS/370 realisiert:

2.3.3.2 OS/370

è Vorteile:

- Vollständiger Schutz durch Kapselung -> Sicherheit
- System-Entwicklung ohne Unterbrechung möglich

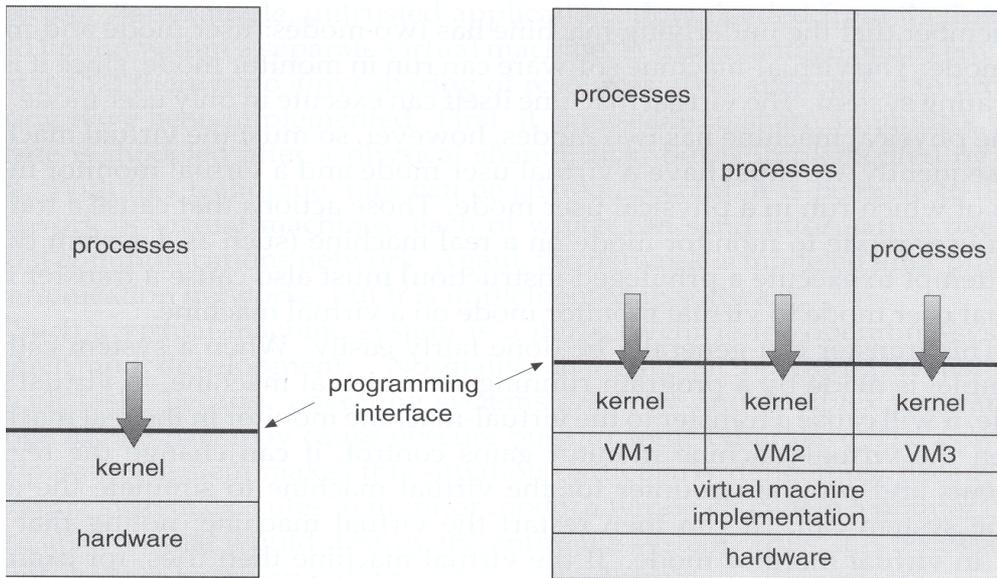


Abbildung 20: Virtual Machine (VM) OS/370

2.3.3.3 Java Virtual Machine (JVM)

Virtuelle Maschine auf bytecode-Basis:

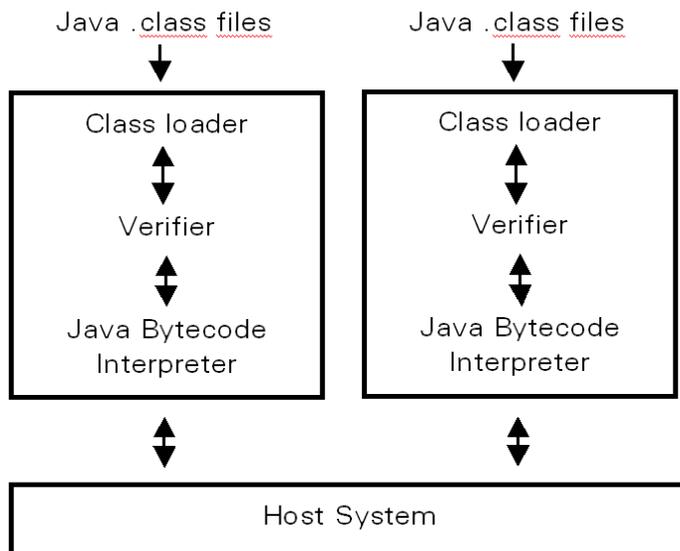


Abbildung 21: Java VM

2.3.3.4 Weitere Virtuelle Maschinen

- Emulation von PC-Rechnern:
 - VMWare emuliert PC-Hardware
 - Windows auf Linux-Systemen ist möglich
- Microsoft Virtual Server: neue Server-Familie
- Win32-Emulation: WINE (**W**ine **I**s **N**ot an **E**mulator)