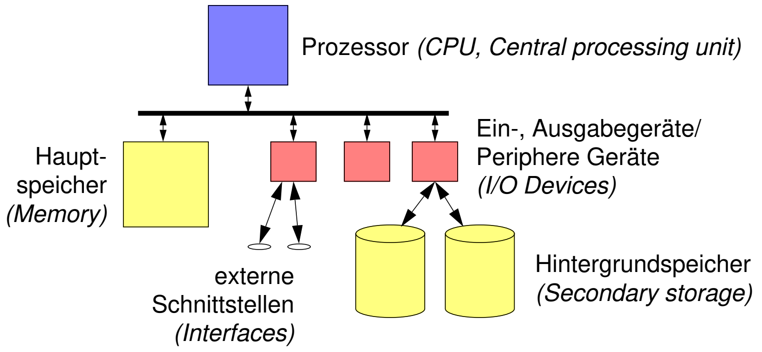
D Einführung Betriebssysteme

1 Was sind Betriebssysteme?

- DIN 44300
  - „...die Programme eines digitalen Rechensystems, 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.**“
- Tanenbaum
  - „...eine Software-Schicht ..., die alle Teile des Systems verwaltet und dem Benutzer eine Schnittstelle oder eine *virtuelle Maschine* anbietet, die einfacher zu verstehen und zu programmieren ist [als die nackte Hardware].“
- ★ Zusammenfassung:
  - Software zur Verwaltung und Virtualisierung der Hardwarekomponenten (Betriebsmittel)
  - Programm zur Steuerung und Überwachung anderer Programme

2 Verwaltung von Betriebsmitteln



2 Verwaltung von Betriebsmittel (2)

- Resultierende Aufgaben
  - Multiplexen von Betriebsmitteln für mehrere Benutzer bzw. Anwendungen
  - Schaffung von Schutzumgebungen
  - Bereitstellen von Abstraktionen zur besseren Handhabbarkeit der Betriebsmittel
- Ermöglichen einer koordinierten gemeinsamen Nutzung von Betriebsmitteln, klassifizierbar in
  - aktive, zeitlich aufteilbare (Prozessor)
  - passive, nur exklusiv nutzbare (periphere Geräte, z.B. Drucker u. Ä.)
  - passive, räumlich aufteilbare (Speicher, Plattenspeicher u. Ä.)
- Unterstützung bei der Fehlererholung

3 Schnittstellen

- Das Betriebssystem soll Benutzervorstellungen auf die Maschinengegebenheiten abbilden und geeignete Schnittstellen bereitstellen für

Benutzer:
Dialogbetrieb, graphische Benutzeroberflächen

Programmierer:
Programmiersprachen, Modularisierungshilfen, Interaktionsmodelle (Programmiermodell)

Operateure:
Werkzeuge zur Gerätebedienung und Anpassung von Systemstrategien

Administratoren:
Werkzeuge zur Benutzerverwaltung, langfristige Systemsteuerung

Programme:
„*Supervisor calls* (SVC)“,
„*Application Programmer Interface* (API)“

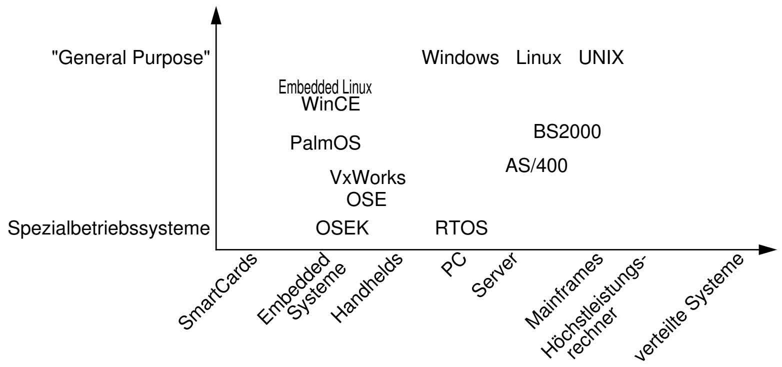
Hardware:
Gerätetreiber

4 Ablaufmodelle

- Betriebssystem realisiert eine Ablaufumgebung
- Bereitstellung von Hilfsmitteln zur Bearbeitung von Benutzerprogrammen und zur Steuerung ihrer Abläufe.
  - ◆ Laden und Starten von Programmen
  - ◆ Überwachung des Programmablaufs
  - ◆ Beenden und Eliminieren von Programmen
  - ◆ Abrechnung *(Accounting)*

D.1 Klassifikation von Betriebssystemen

- Unterschiedliche Klassfikationskriterien
  - ➤ Zielplattform
  - ➤ Einsatzzweck, Funktionalität



D.1 Klassifikation von Betriebssystemen (2)

- Wenigen "General Purpose"- und Mainframe/Höchstleistungsrechner-Betriebssystemen steht eine Vielzahl kleiner und kleinster Spezialbetriebssysteme gegenüber:

  C51, C166, C251, CMX RTOS, C-Smart/Raven, eCos, eRTOS, Embos, Ercos, Euros Plus, Hi Ross, Hynet-OS, LynxOS, MicroX/OS-II, Nucleus, OS-9, OSE, OSEK Flex, OSEK Turbo, OSEK Plus, OSEKtime, Precise/MQX, Precise/RTCS, proOSEK, pSOS, PXROS, QNX, Realos, RTMOSxx, Real Time Architect, ThreadX, RTA, RTX51, RTX251, RTX166, RTXC, Softune, SSXS RTOS, VRTX, VxWorks, …

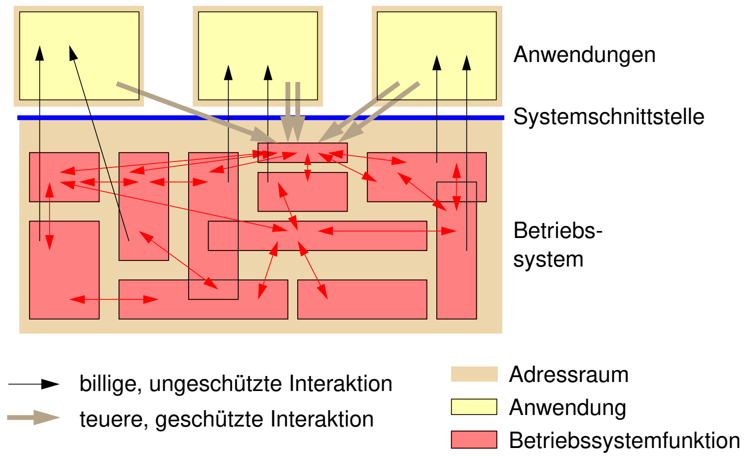
➥ Einsatzbereich: Eingebettete Systeme, häufig Echtzeit-Betriebssysteme, über 50% proprietäre (in-house) Lösungen

- Alternative Klassifikation: nach Architektur

D.2 Betriebssystemarchitekturen

- Umfang zehntausende bis mehrere Millionen Befehlszeilen
  - ◆ Strukturierung hilfreich
- Verschiedene Strukturkonzepte
  - ◆ monolithische Systeme
  - ◆ geschichtete Systeme
  - ◆ Minimalkerne
  - ◆ Laufzeitbibliotheken (minimal, vor allem im Embedded-Bereich)
- Unterschiedliche Schutzkonzepte
  - ➤ kein Schutz
  - ➤ Schutz des Betriebssystems
  - ➤ Schutz von Betriebssystem und Anwendungen untereinander
  - ➤ feingranularer Schutz auch innerhalb von Anwendungen

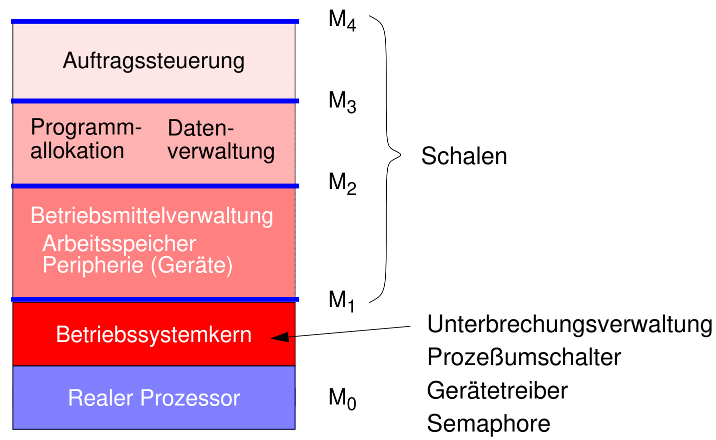
1 Monolithische Systeme



1 Monolithische Systeme (2)

- Prozedursammlung
  - uneingeschränkte Aufruffolgen
  - alles in einem Adreßraum (Alle Adressen des Betriebssystems sind von allen Modulen erreichbar)
- ★ Vorteile
  - Effiziente Kommunikation, effizienter Datenzugriff innerhalb des Kerns
  - Privilegierte Befehle jederzeit ausführbar
- ▲ Nachteile
  - Keine interne Strukturierung (änderungsunfreundlich, fehleranfällig)
  - Kein Schutz zwischen Kernkomponenten (Problem: zugekaufte Treiber)

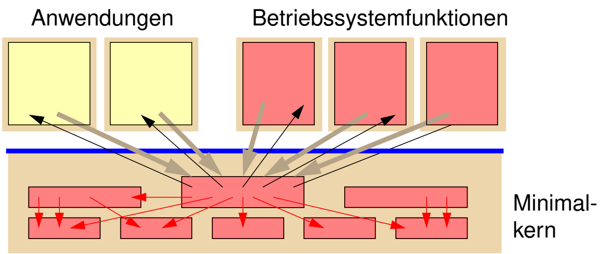
2 Geschichtete Systeme



2 Geschichtete Systeme (2)

- Module werden in Schichten angeordnet
  - eindeutige und zyklenfreie Benutztrelation
  - die Korrektheit von Schicht n hängt von Schicht n-1 ab
  - tiefere Schichten implementieren bevorzugt Mechanismen
  - höhere Schichten implementieren bevorzugt Strategien
- ★ Vorteile
  - Schutz zwischen verschiedenen BS-Teilen
  - Interne Strukturierung
- ▲ Nachteile
  - Mehrfacher Schutzraumwechsel ist teuer
  - Unflexibler und nur einseitiger Schutz (von unten nach oben)

3 Minimalkerne



- billige, ungeschützte Interaktion
- teuere, gesicherte Interaktion
- Adressraum
- Anwendung
- Betriebssystemfunktion

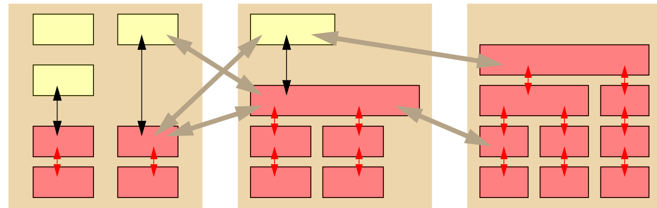
3 Minimalkerne (2)

- Die Kernfunktion umfaßt typischerweise die Mechanismen für den Zugriff auf die Hardwarekomponenten.
- Die Strategien und hardwareunabhängigen Betriebssystemteile werden als Anwendungen behandelt.

★ Vorteile
- ◆ Gute Modularisierung
- ◆ Schutz der Komponenten voreinander

▲ Nachteil
- ◆ Kommunikation zwischen Modulen ist teuer

4 Objektbasierte, offene Systeme



- billige, durch Objektkapselung geschützte Interaktion
- teuere, durch Adressraumgrenze geschützte Interaktion
- Adressraum
- Anwendungsobjekte
- Betriebssystemobjekte

4 Objektbasierte, offene Systeme (2)

- Sicherung der Modulgrenzen durch Programmiermodell und Software
  - ◆ Byte-Code-Verifier einer virtuellen Maschine (z.B. in Java)überprüft den Objektzugriff.
  - ◆ Vertauenswürdiger Compiler (z.B. in Modula 3) sichert die Objektkapselung.

★ Vorteile
- ◆ Schutz auf mehreren Ebenen (Sprache, Code-Prüfung, Adressraum)
- ◆ Modularisierung und Effizienz möglich

▲ Nachteile
- ◆ Komplexes Sicherheitsmodell

D.3 Geschichtliche Entwicklung

1 1950–1960

1950

◆ Einströmige Stapelsysteme
(Single-stream batch processing systems)
Aufträge zusammen mit allen Daten werden übergeben und sequentiell bearbeitet

◆ Steuerung durch Auftragsabwickler
(Resident monitor, Job monitor)
Hilfsmittel: Assembler, Compiler, Binder und Lader, Programmbibliotheken

1960

2 1960–1965

1960

◆ Autonome periphere Geräte → Überlappung von Programmbearbeitung und Datentransport zw. Arbeitsspeicher und peripheren Geräten möglich
- Wechselpufferbetrieb (abwechselndes Nutzen zweier Puffer)
- Mehrprogrammbetrieb *(Multiprogramming)*
- Spooling *(**S**imultaneous **p**eripheral **o**peration **o**n-**l**ine)*

◆ Mehrere Programme müssen gleichzeitig im Speicher sein → Auslagern von Programmen auf Sekundärspeicher

◆ Programme müssen während des Ablaufs verlagerbar sein *(Relocation problem)*

◆ Echtzeitdatenverarbeitung *(Real-time processing)*, d.h. enge Bindung von Ein- und Ausgaben an die physikalische Zeit

1965

3 1965–1970

1965
OS/360

◆ Umsetzung von Programmadressen in Speicherorte zur Laufzeit: Segmentierung, Seitenadressierung *(Paging)*

◆ Virtueller Adressraum: Seitentausch *(Paging)*
Seiten werden je nach Zugriff ein- und ausgelagert

◆ Interaktiver Betrieb *(Interactive processing, Dialog mode)*

◆ Mehrbenutzerbetrieb, Teilnehmersysteme *(Time sharing)*

THE

◆ Problem: Kapselung von Prozessen und Dateien → geschützte Adressraum, Zugriffsschutz auf Dateien

MULTICS

◆ Dijkstra: Programmsysteme als Menge kooperierender Prozesse (heute *Client-Server*)

◆ Problem: Prozessinteraktion bei gekapselten Prozessen → Nachrichtensysteme zur Kommunikation, gemeinsamer Speicher zur Kooperation

UNIX
1970

4 1970–1975

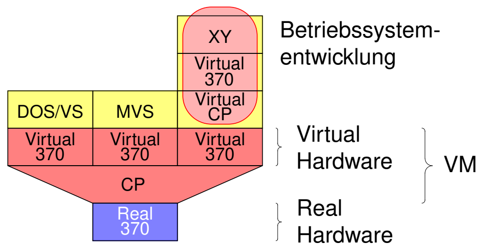
1970
VM

◆ Modularisierung:
Datenkapselung, Manipulation durch Funktionen (nach Parnas)

◆ Virtuelle Maschinen: Koexistenz verschiedener Betriebssysteme im gleichen Rechner

Hydra

MVS



◆ Symmetrische Multiprozessoren: HYDRA
- Zugangskontrolle zu Instanzen durch Capabilities
- Trennung von Strategie und Mechanismus

◆ Komplexe Dateisysteme

1975

5 1975–1985

1975

- Vernetzung
- Protokolle (z.B. TCP/IP)
- Verteilte Systeme
-
- Newcastle Connection

LOCUS

- Fernaufruf *(Remote procedure call, RPC)*

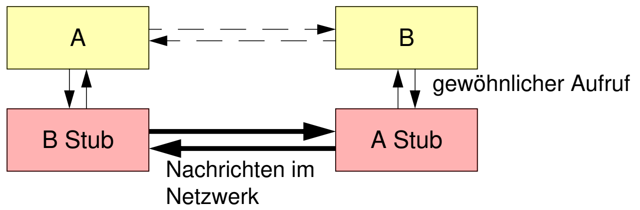
1980

MS-DOS

NC

EDEN

1985



6 1985–1999

1985

- Kryptographie

OS/2

- Authentifizierung und Authentisierung
- Objektorientierte Systeme

Mach 3.0

- Parallele Systeme
- Mikrokerne

1990

Windows

- Objektorientierte Mikrokerne

Spring

Win NT

- Internet, Multimedia

1995

7 1995–1999

1995

SPIN

Exokernel

L4

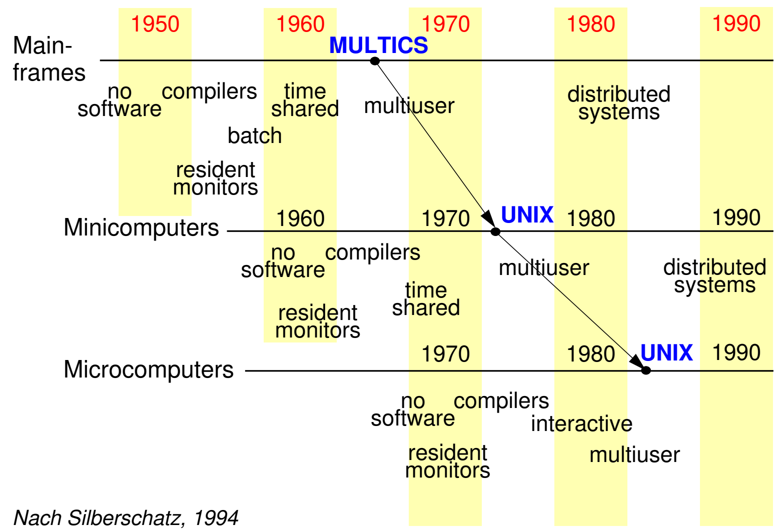
Linux

- Echtzeitscheduling in Standardbetriebssystemen
- 64bit-Adressierung

JX

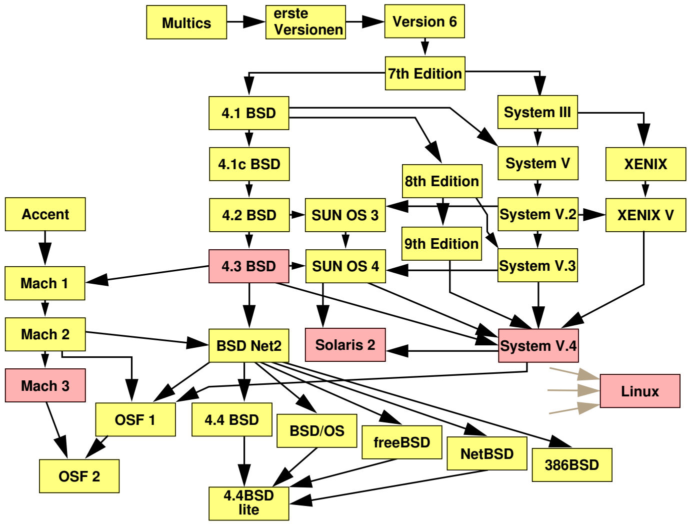
2000

8 Migration von Konzepten



Nach Silberschatz, 1994

9 UNIX Entwicklung



D.4 Betriebssystemkomponenten

- Speicherverwaltung
  - Wann darf welche Information wohin im Speicher ablegen?
- Prozessverwaltung
  - Wann darf welche Aufgabe bearbeitet werden?
- Dateisystem
  - Speicherung und Schutz von Langzeitdaten
- Ein/Ausgabe
  - Kommunikation mit der "Außenwelt" (Benutzer/Rechner)