

Vorlesung

Systemprogrammierung I

Wintersemester 1998/99

(10320)

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

0-Title.fm 1998-11-02 11.07

0.1

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A Organisatorisches



Dozent

◆ Dr.-Ing. Franz J. Hauck, IMMD IV (Lehrstuhl für Betriebssysteme)

◆ hauck@informatik.uni-erlangen.de



Vorlesung und Übungen

◆ für Studierende der Fachrichtung Informatik im 3. Semester,
für Studierende der Fachrichtung Wirtschaftsinformatik im Hauptstudium
und einige weitere

◆ anrechenbare Semesterwochenstunden:
4 SWS Vorlesung, 4 SWS Übungen

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.1

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.1 Inhalt

- Vorlesung
 - ◆ Grundlagen der Betriebssysteme (eingeschränkt auf Monoprozessoren)
 - ◆ Konzepte moderner Betriebssysteme
 - ◆ Beispielhafte Betrachtung von UNIX, Windows, Windows NT und Mach
- Übungen
 - ◆ Umgang mit den in der Vorlesung vorgestellten Betriebssystemkonzepten
 - ◆ Betriebssystemschnittstelle des UNIX Betriebssystems
 - ◆ Umgang mit sogenannten „System calls“
 - ◆ Praktische Arbeiten: Ausprogrammieren von Übungsaufgaben

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.2

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.2 Vorlesung

- Termine: [Mi. und Do. von 16.15 bis 17.45 im H4](#)
- Skript
 - ◆ zwei Alternativen:
 - Folien der Vorlesung werden im WWW zur Verfügung gestellt und können selbst ausgedruckt werden
 - Folien werden kopiert und vor der Vorlesung ausgegeben; Gutscheinverkauf in den ersten Übungen; Kosten **10,00 DM**
 - ◆ weitergehende Informationen zum Nachlesen findet man am besten in der angegebenen Literatur
- URL zur Vorlesung
 - ◆ http://www4.informatik.uni-erlangen.de/Lehre/WS98/V_SP1/
 - ◆ hier findet man Termine, Folien zum Ausdrucken und Zusatzinformationen

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.3

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.2 Vorlesung (2)

■ Literatur

- ◆ A. Silberschatz; P. B. Galvin: Operating Systems Concepts. 4th Edition, Addison-Wesley, 1994.
- ◆ A. S. Tanenbaum: Modern Operating Systems, Prentice Hall, Englewood Cliffs, NJ, 1992.
- ◆ R. W. Stevens: Advanced Programming in the UNIX Environment. Addison-Wesley, 1992.

A.2 Vorlesung (3)

■ Live-Übertragung nach Nürnberg ist noch ungesichert

- ◆ Besteht Interesse?

■ Rückmeldungen und Fragen

- ◆ Gute Vorlesung kann nicht entstehen, wenn nur Informationen vom Dozent zu den Hörern fließen!
- ◆ Machen Sie mich auf Fehler aufmerksam!
- ◆ Stellen Sie Fragen!
- ◆ Nutzen Sie außerhalb der Vorlesung die Möglichkeit elektronische Post zu versenden: hauck@informatik.uni-erlangen.de !

A.3 Übungen

- Verantwortlich für die Übung
 - ◆ Dr.-Ing. Jürgen Kleinöder
 - ◆ Dipl.-Inf. Michael Golm
 - ◆ Dipl.-Inf. Martin Steckermeier
 - ◆ und studentische Hilfskräfte

- Aufteilung
 - ◆ Tafelübung – 2 SWS
 - ◆ Rechnerübung – 2 SWS

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.6

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.3 Übungen (2)

- Schein
 - ◆ wird auf die Lösung von Übungsaufgaben vergeben
 - ◆ Mindestpunktzahl zum Bestehen des Scheins nötig
 - ◆ Aufgaben sollten in Gruppen zu zwei Personen gelöst werden
 - ◆ Mitglieder einer Zweiergruppe müssen an der selben Tafelgruppe teilnehmen
- Anmeldung zur Übung und Einteilung in die Übungsgruppen
 - ◆ „login: span“ an allen CIP-Workstations der Informatik
 - ◆ Login ist freigeschaltet ab Donnerstag, 5. November 1998 08:00 Uhr
 - ◆ Benötigte Eingaben: persönliche Daten, Matrikelnummer, Termine der gewünschten Übungen

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.7

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.3 Übungen (3)

■ Tafelübungen

- ◆ Mo 10:00 - 12:00, Raum 00.151
- ◆ Mo 14:00 - 16:00, Raum 00.151
- ◆ Di 10:00 - 12:00, Raum K1
- ◆ Mi 08:00 - 10:00, Raum 0.031
- ◆ Do 12:30 - 14:00, Raum 2.038
- ◆ Do 14:15 - 15:45, Raum 2.038

- ◆ Besprechung von Übungsaufgaben
- ◆ Klärung offener Fragen
- ◆ Vermittlung ergänzender Informationen

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.8

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

A.3 Übungen (4)

■ Rechnerübungen

- ◆ Mo 16:00 - 18:00, Raum 01.155 u. 01.155-N
- ◆ Di 16:00 - 18:00, Raum 01.155 u. 01.155-N
- ◆ Mi 14:00 - 16:00, Raum 01.155 u. 01.155-N

- ◆ Lösung der Übungsaufgaben
- ◆ Raum 01.155 u. 01.155-N ist reserviert
(Vorrang am Rechner für Übungsteilnehmer)
- ◆ Übungsleiter steht für Fragen zur Verfügung

■ Übungsbeginn ist Montag, 9. November 1998

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

A-Org.fm 1998-11-03 09.08

A.9

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

B Einführung

B.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].“

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.1

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

B.1 Was sind Betriebssysteme? (2)

■ Silberschatz/Galvin

- ◆ „... ein Programm, das als Vermittler zwischen Rechnernutzer und Rechner-Hardware fungiert. Der Sinn des Betriebssystems ist eine Umgebung bereitzustellen, in der Benutzer bequem und effizient Programme ausführen können.“

■ Brinch Hansen

- ◆ „... der Zweck eines Betriebssystems [liegt] in der Verteilung von Betriebsmitteln auf sich bewerbende Benutzer.“

★ Zusammenfassung

- ◆ Software zur Betriebsmittelverwaltung
- ◆ Bereitstellung von Grundkonzepten zur statischen und dynamischen Strukturierung von Programmsystemen

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

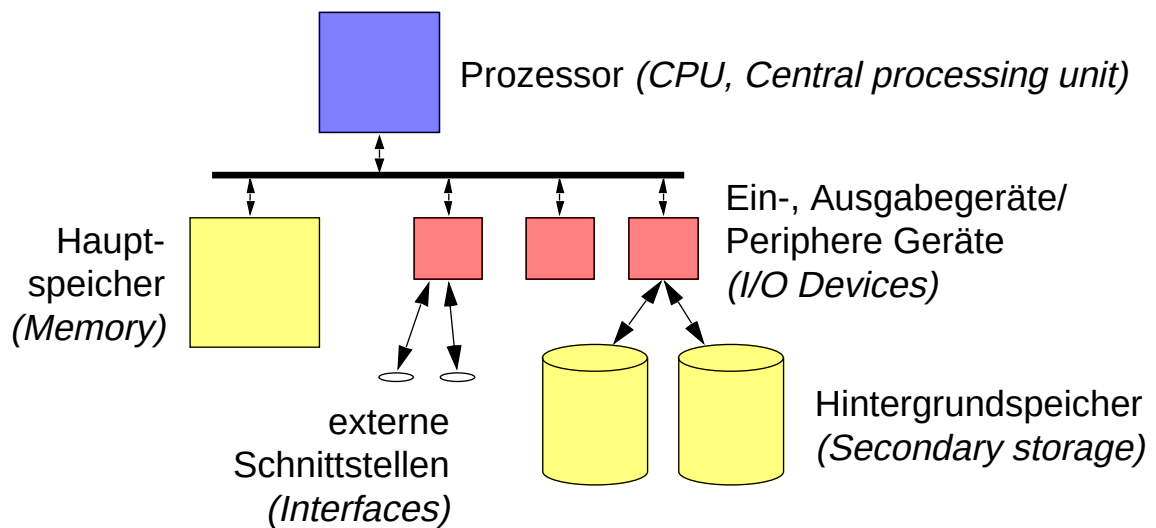
B-Intro.fm 1998-11-02 11.07

B.2

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

1 Verwaltung von Betriebsmitteln

■ Betriebsmittel



1 Verwaltung von Betriebsmitteln (2)

■ Resultierende Aufgaben

- ◆ Multiplexen von Betriebsmitteln für mehrere Benutzer bzw. Anwendungen
- ◆ Schaffung von Schutzumgebungen

■ Ermöglichung 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

2 Schnittstellen

- Betriebssystem soll Benutzervorstellungen auf die Maschinengegebenheiten abbilden

- ◆ Bereitstellung geeigneter Abstraktionen und Schnittstellen für

Benutzer: Dialogbetrieb, graphische Benutzeroberflächen

Anwendungsprogrammierer:
Programmiersprachen, Modularisierungshilfen,
Interaktionsmodelle (Programmiermodell)

Systemprogrammierer:
Werkzeuge zur Wartung und Pflege

Operateure: Werkzeuge zur Gerätebedienung und Anpassung von
Systemstrategien

Administratoren: Werkzeuge zur Benutzerverwaltung, langfristige
Systemsteuerung

Programme: „*Supervisor calls (SVC)*“

Hardware: Gerätetreiber

SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

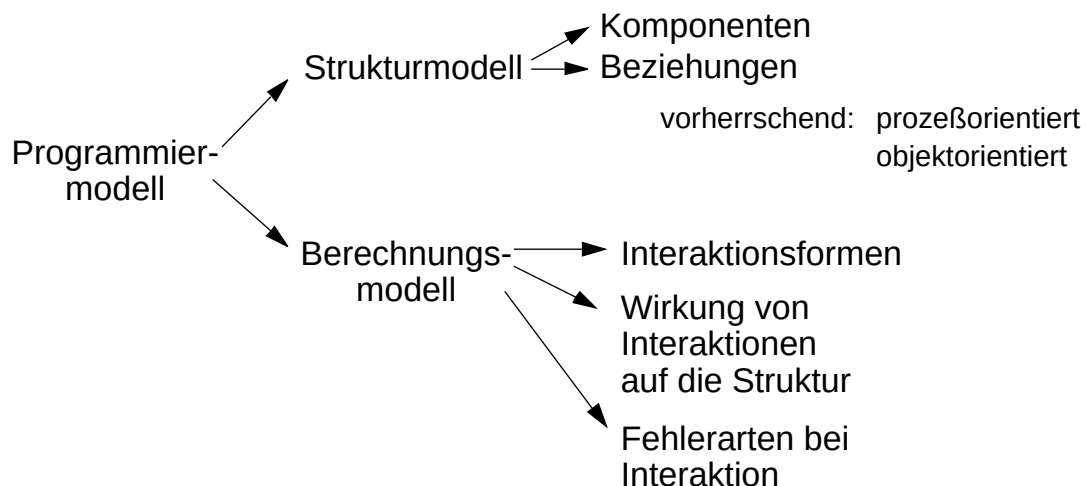
B.5

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Programmiermodelle

- Realisierungen von Programmiermodellen

- ◆ Keine Notwendigkeit genauer Kenntnisse über Hardwareeigenschaften und spezielle Systemsoftwarekomponenten
- ◆ Schaffung einer begrifflichen Basis zur Strukturierung von Programmsystemen und ihrer Ablaufsteuerung



SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.6

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Programmiermodelle (2)

■ Beispiele für Strukturkomponenten

- ◆ Dateien (Behälter zur langfristigen Speicherung von Daten)
- ◆ Prozesse (in Ausführung befindliche Programme)
- ◆ Klassen (Vorlagen zur Bildung von Instanzen)
- ◆ Instanzen
- ◆ Prozeduren
- ◆ Sockets (Kommunikationsendpunkte, „Kommunikationssteckdosen“)
- ◆ Pipes (Nachrichtenkanäle)

■ Beispiele für Beziehungen

- ◆ A kann B referenzieren, beauftragen, aufrufen, modifizieren
- ◆ Pipe P verbindet A und B

3 Programmiermodelle (3)

■ Beispiele für Interaktionsformen

- ◆ Prozedur-(Methoden-)Aufruf
- ◆ Nachrichtenaustausch
- ◆ Gemeinsame Speichernutzung

■ Wirkung von Interaktionen auf die Struktur

- ◆ Erzeugung und Tilgung von Prozessen
- ◆ Instantiierung

■ Fehlerarten bei Interaktion

- ◆ Verlust, Wiederholung oder Verspätung von Nachrichten
- ◆ Abbruch aufgerufener Methoden, Ausnahmebehandlung

4 Ablaufmodelle

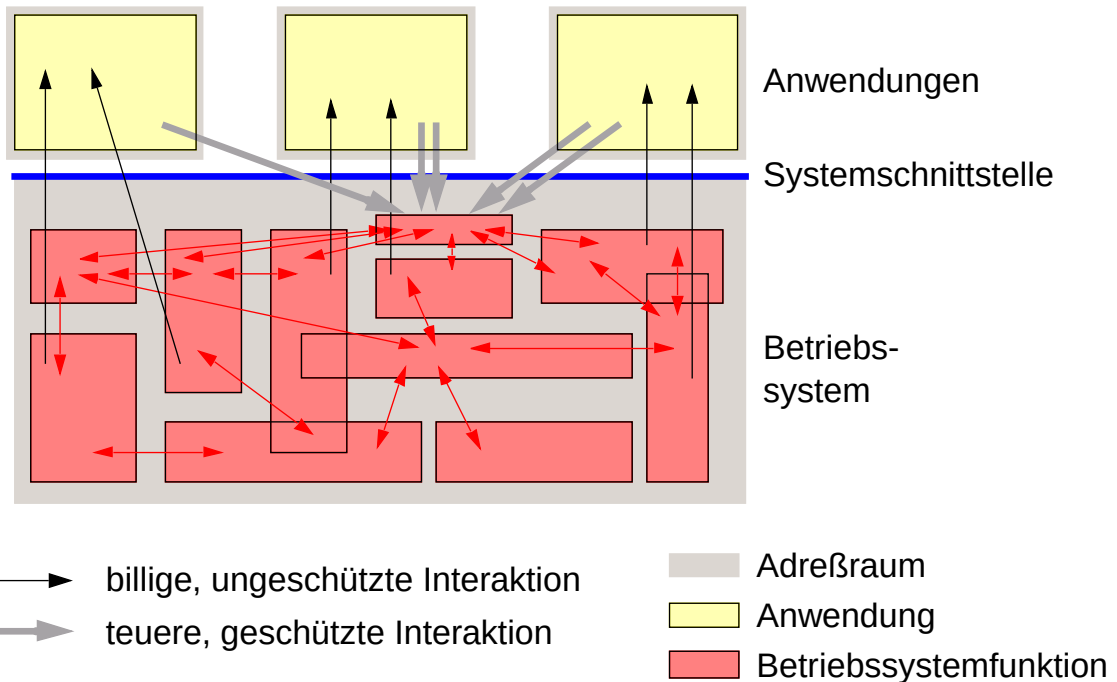
- Realisierungen von Ablaufumgebungen
- Bereitstellung von Hilfsmitteln zur Bearbeitung von Benutzerprogrammen und zur Steuerung ihrer Abläufe.
 - ◆ Laden und Starten von Programmen
 - ◆ Überwachung des Programmablaufs
 - ◆ Beenden und Eliminieren von Programmen
 - ◆ Abrechnung (*Accounting*)

5 Implementierung

- Umfang zehntausende bis mehrere Millionen Befehlszeilen
- Verschiedene Strukturkonzepte
 - ◆ monolithische Systeme
 - ◆ geschichtete Systeme
 - ◆ Minimalkerne
 - ◆ offene objektorientierte Systeme

5 Implementierung (2)

■ Monolithische Systeme



SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

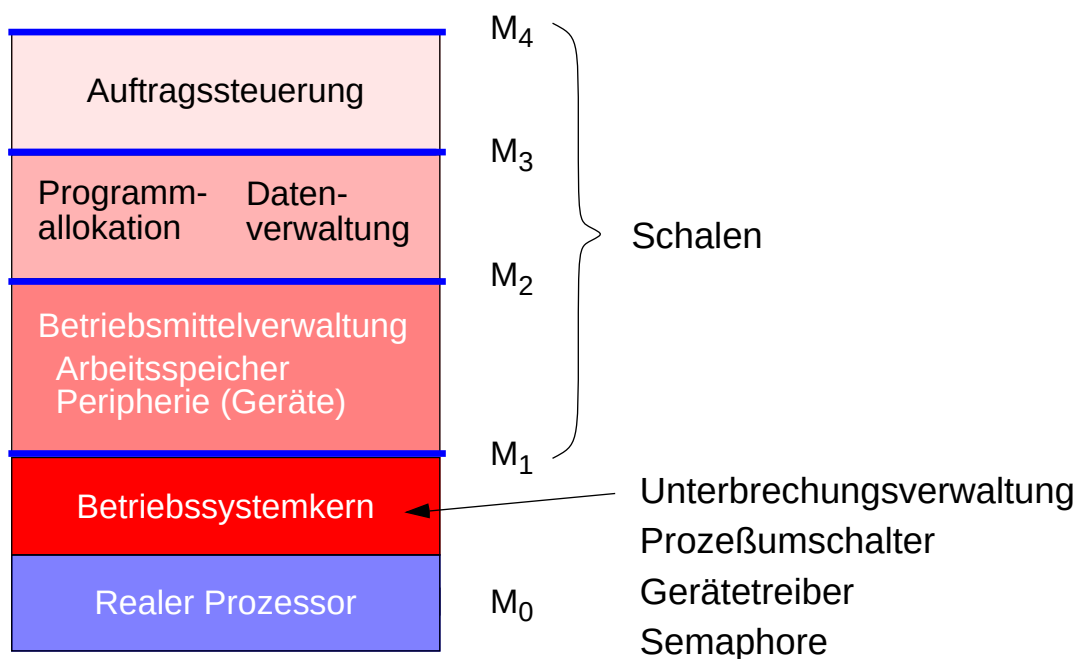
B-Intro.fm 1998-11-02 11.07

B.11

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

5 Implementierung (3)

■ Geschichtete Systeme



SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

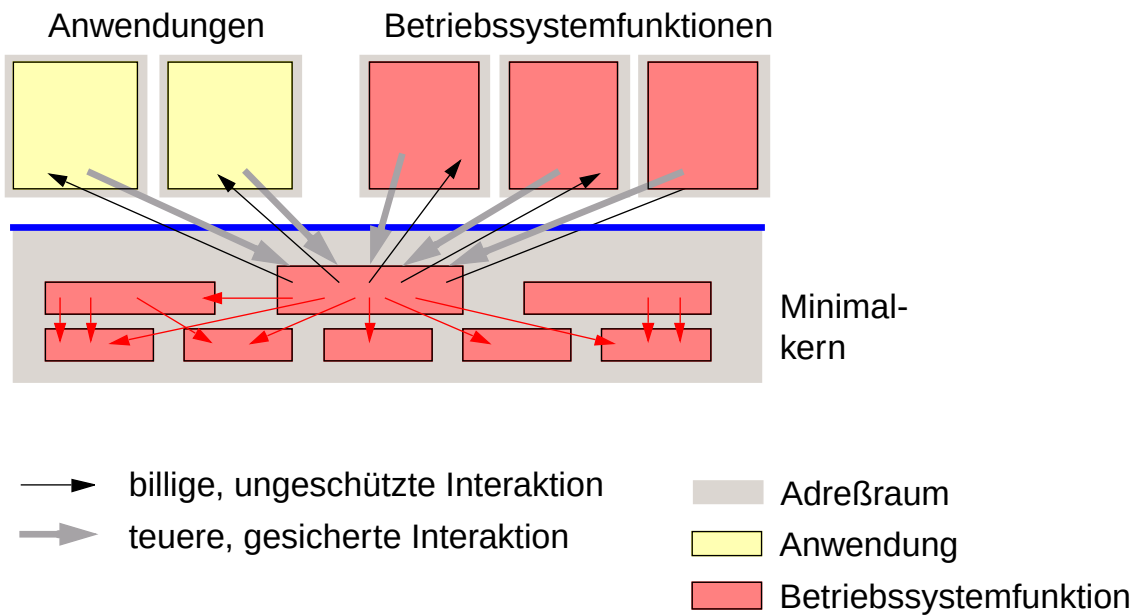
B-Intro.fm 1998-11-02 11.07

B.12

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

5 Implementierung (4)

■ Minimalkerne



SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

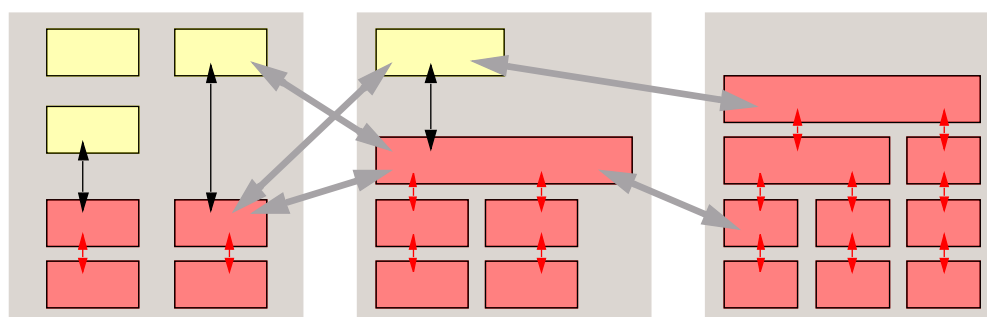
B-Intro.fm 1998-11-02 11.07

B.13

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

5 Implementierung (5)

■ Objektbasierte, offene Systeme



SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.14

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

B.2 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

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.15

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

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 (*Simultaneous peripheral operation on-line*)
- ◆ 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

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.16

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 1965–1970

1965

OS/360

THE

MULTICS

UNIX
1970

- ◆ 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*)
- ◆ Problem: Kapselung von Prozessen und Dateien → geschützte Adreßräume, Zugriffsschutz auf Dateien
- ◆ Dijkstra: Programmsysteme als Menge kooperierender Prozesse (heute *Client-Server*)
- ◆ Problem: Prozeßinteraktion bei gekapselten Prozessen → Nachrichtensysteme zur Kommunikation, gemeinsamer Speicher zur Kooperation

SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.17

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

4 1970–1975

1970

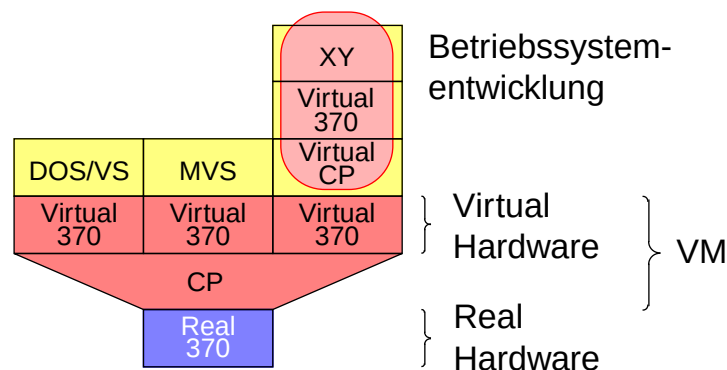
VM

HYDRA

MVS

1975

- ◆ Modularisierung:
Datenkapselung, Manipulation durch Funktionen (nach Parnas)
- ◆ Virtuelle Maschinen: Koexistenz verschiedener Betriebssysteme im gleichen Rechner



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

SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.18

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

5 1975–1985

1975

- ◆ Vernetzung
- ◆ Protokolle (z.B. TCP/IP)
- ◆ Verteilte Systeme
- ◆ Newcastle Connection
- ◆ Fernaufruf (*Remote procedure call, RPC*)

LOCUS

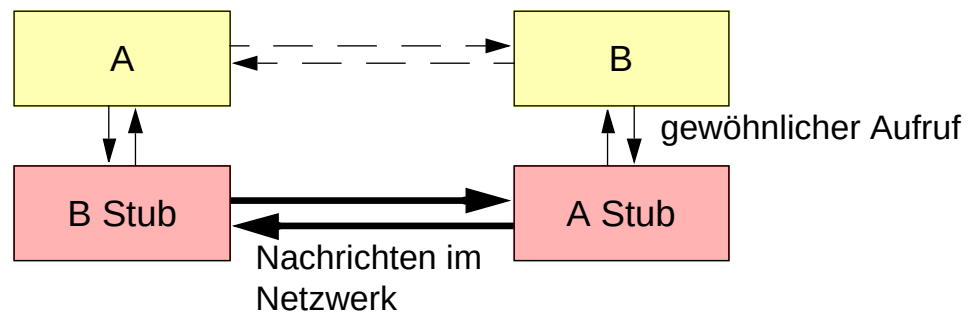
1980

MS-DOS

NC

EDEN

1985



SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.19

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

6 1985–1997

1985

OS/2

MACH 3.0

1990

WINDOWS

SPRING

WIN NT

1997

- ◆ Kryptographie
- ◆ Authentifizierung und Authentisierung
- ◆ Objektorientierte Systeme
- ◆ Parallele Systeme
- ◆ Mikrokerne
- ◆ Objektorientierte Mikrokerne
- ◆ Internet, Multimedia

SPI

Systemprogrammierung I

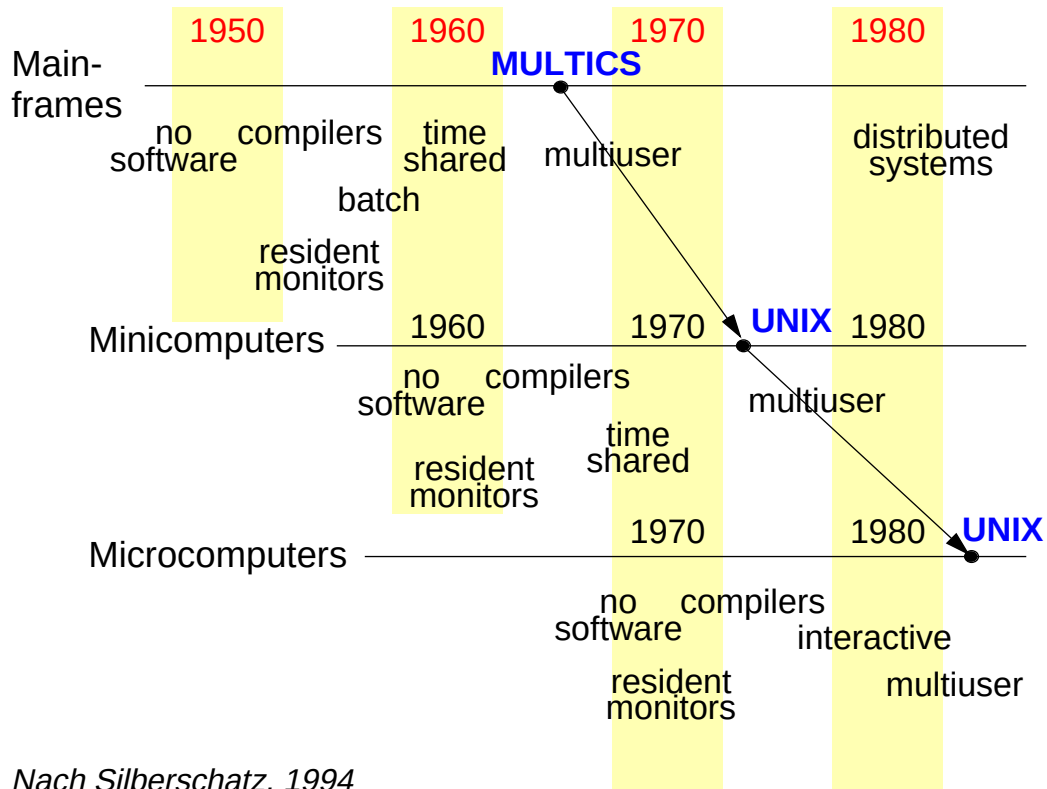
© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.20

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

7 Migration von Konzepten



Nach Silberschatz, 1994

SPI

Systemprogrammierung I

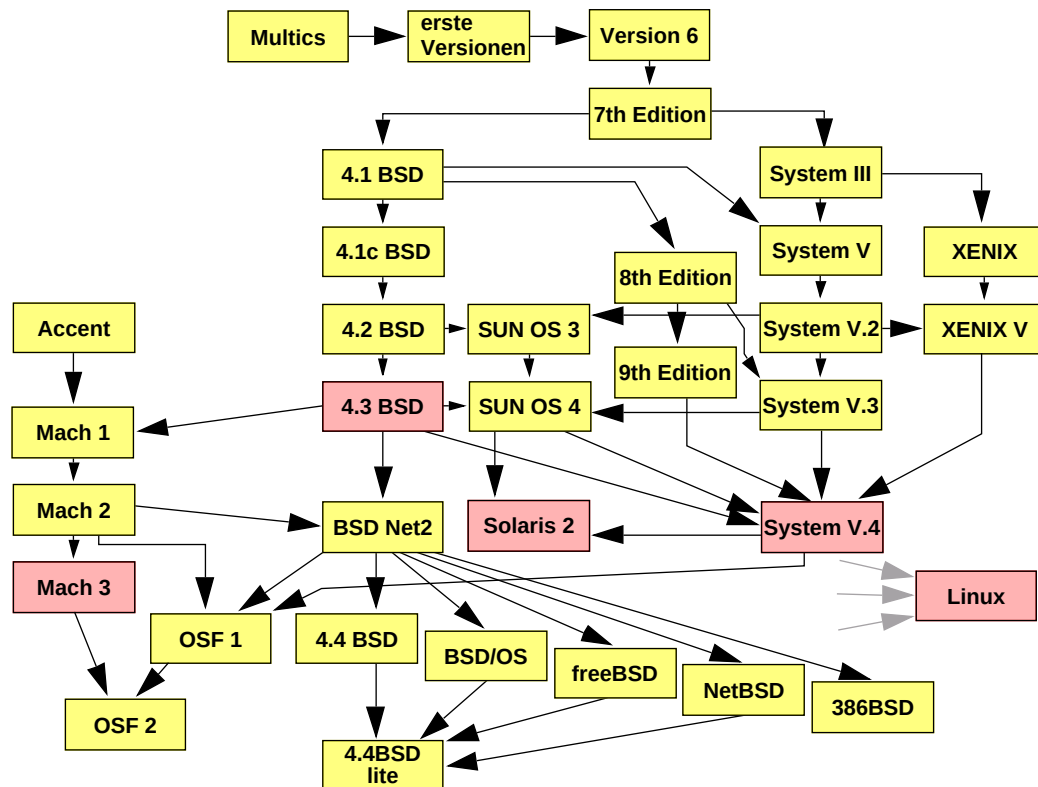
© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.21

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

8 UNIX Entwicklung



SPI

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

B-Intro.fm 1998-11-02 11.07

B.22

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

B.3 Warum Systemprogrammierung I?

- Rasche Einarbeitung in spezielle Systeme
 - ◆ MVS, BS2000, VM, Solaris, Unix, Windows NT, Windows 95, MS/DOS
- Strukturierung komplexer Programmsysteme
 - ◆ Unterteilung in interagierende Komponenten
- Konzeption und Implementierung spezialisierter Systeme
 - ◆ Embedded systems
 - ◆ Automatisierungssysteme
- Erstellung fehlertoleranter Systeme
- Verständnis für Abläufe im Betriebssystem
 - ◆ Ökonomische Nutzung der Hardware
 - ◆ Laufzeitoptimierung anspruchsvoller Anwendungen

1 Phänomene der Speicherverwaltung

- Beispiel: Initialisierung von großen Matrizen
 - ◆ Variante 1:

```
#define DIM 6000

int main()
{
    register long i, j;
    static long matrix[DIM][DIM];

    for( i= 0; i < DIM; i++ )
        for( j= 0; j < DIM; j++ )
            matrix[i][j]= 1;

    exit(0);
}
```

1 Phänomene der Speicherverwaltung (2)

■ Beispiel: Initialisierung von großen Matrizen

◆ Variante 2:

```
#define DIM 6000

int main()
{
    register long i, j;
    static long matrix[DIM][DIM];

    for( j= 0; j< DIM; j++ )
        for( i= 0; i< DIM; i++ )
            matrix[i][j]= 1;

    exit(0);
}
```

◆ Schleifen sind vertauscht

1 Phänomene der Speicherverwaltung (3)

■ Meßergebnisse

◆ Variante 1:

User time= 3,69 sec; System time= 1,43 sec; Gesamtzeit= 22,03 sec

◆ Variante 2:

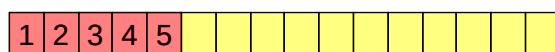
User time= 21,86 sec; System time= 2,33 sec; Gesamtzeit= 86,39 sec

■ Ursachen

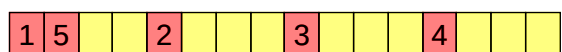
◆ Variante 1 geht sequentiell durch den Speicher

◆ Variante 2 greift versetzt ständig auf den gesamten Speicherbereich zu

Beispiel: `matrix[4][4]` und die ersten fünf Zugriffe



Variante 1



Variante 2

1 Phänomene der Speicherverwaltung (4)

■ Ursachen

◆ Logischer Adressraum

- Benutzte Adressen sind nicht die physikalischen Adressen
- Abbildung wird durch Hardware auf Seitenbasis vorgenommen (Seitenadressierung)
- Variante 2 hat weniger Lokalität, d.h. benötigt häufig wechselnde Abbildungen

◆ Virtueller Speicher

- Möglicher Adreßraum ist größer als physikalischer Speicher
- Auf Seitenbasis werden Teile des benötigten Speichers ein- und ausgelagert
- bei Variante 2 muß viel mehr Speicher ein- und ausgelagert werden

2 Phänomene des Dateisystems

■ Beispiel: Sequentielles Schreiben mit unterschiedlicher Pufferlänge

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#define BUFLen 8191

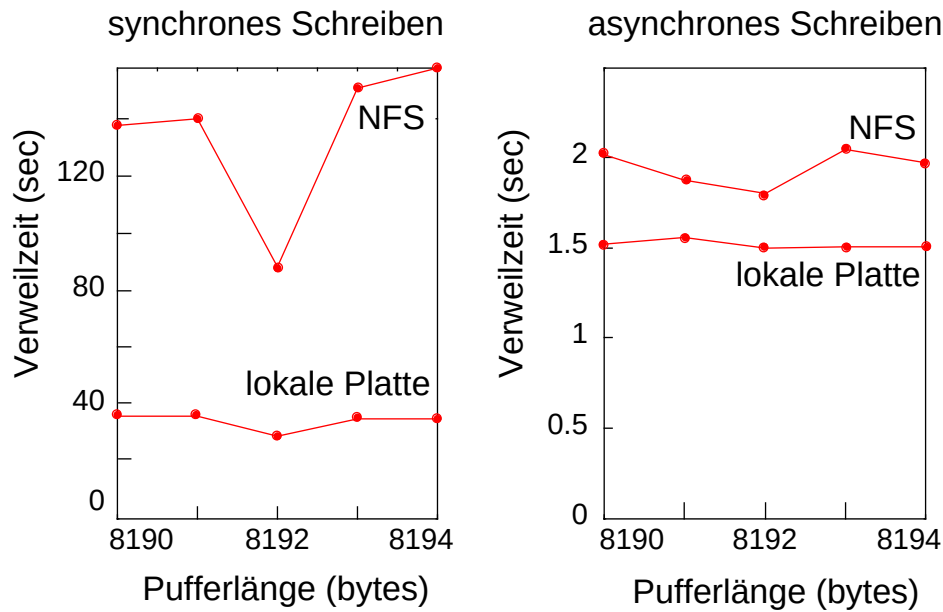
int main()
{
    static char buffer[BUFLen];
    int i, fd= open( "filename",
                    O_CREAT|O_TRUNC|O_WRONLY|O_SYNC,
                    S_IRUSR|S_IWUSR );

    for( i= 0; i < 1000; i++ )
        write( fd, buffer, BUFLen );

    exit(0);
}
```

2 Phänomene des Dateisystems (2)

■ Meßergebnisse



2 Phänomene des Dateisystems (3)

■ Ursachen

- ◆ Synchrones Schreiben erfordert sofortiges Rausschreiben der Daten auf Platte (nötig beispielsweise, wenn hohe Fehlertoleranz gefordert wird – Platte ist immer auf dem neuesten Stand)
- ◆ 8192 ist ein Vielfaches der Blockgröße der Plattenblocks
- ◆ kleine Abweichungen von der Blockgröße erfordern zusätzliche Blocktransfers

B.4 Überblick über die Vorlesung

- ★ Inhaltsübersicht
 - A. Organisation
 - B. Einführung
 - C. Dateisysteme
 - D. Prozesse und Nebenläufigkeit
 - E. Speicherverwaltung
 - F. Implementierung von Dateien
 - G. Ein-, Ausgabe
 - H. Verklemmungen
 - I. Datensicherheit und Zugriffsschutz
 - J. Fallbeispiel: Mach

SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

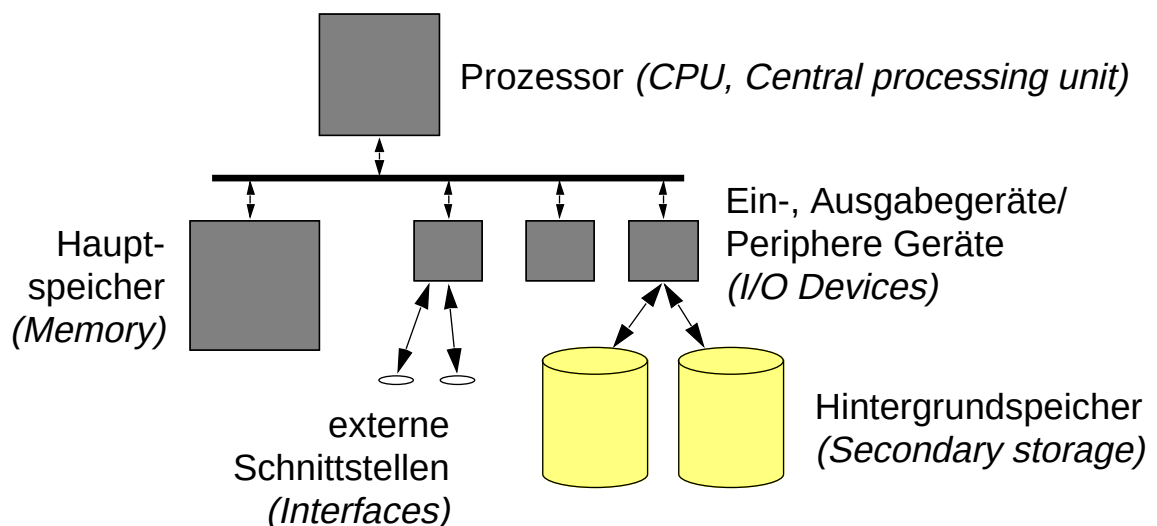
B-Intro.fm 1998-11-02 11.07

B.31

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

C Dateisysteme

■ Einordnung



SP I

Systemprogrammierung I

© Franz J. Hauck, Univ. Erlangen-Nürnberg, IMMD IV, 1998/1999

C-File.fm 1998-11-02 11.45

C.1

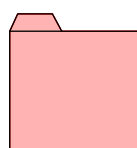
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

C Dateisysteme (2)

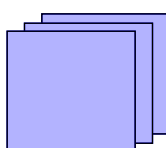
- Dateisysteme speichern Daten und Programme persistent in Dateien
 - ◆ Betriebssystemabstraktion zur Nutzung von Hintergrundspeichern (z.B. Platten, CD-ROM, Floppy Disk, Bandlaufwerke)
 - Benutzer muß sich nicht um die Ansteuerungen verschiedener Speichermedien kümmern
 - einheitliche Sicht auf den Sekundärspeicher
- Dateisysteme bestehen aus
 - ◆ Dateien (*Files*)
 - ◆ Katalogen / Verzeichnissen (*Directories*)
 - ◆ Partitionen (*Partitions*)

C Dateisysteme (3)

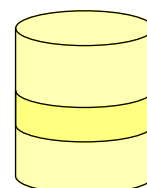
- Datei
 - ◆ speichert Daten oder Programme
- Katalog
 - ◆ erlaubt Benennung der Dateien
 - ◆ enthält Zusatzinformationen zu Dateien
- Partitionen
 - ◆ eine Menge von Katalogen und deren Dateien
 - ◆ Sie dienen zum physischen oder logischen Trennen von Dateimengen.



Katalog



Dateien



Partition