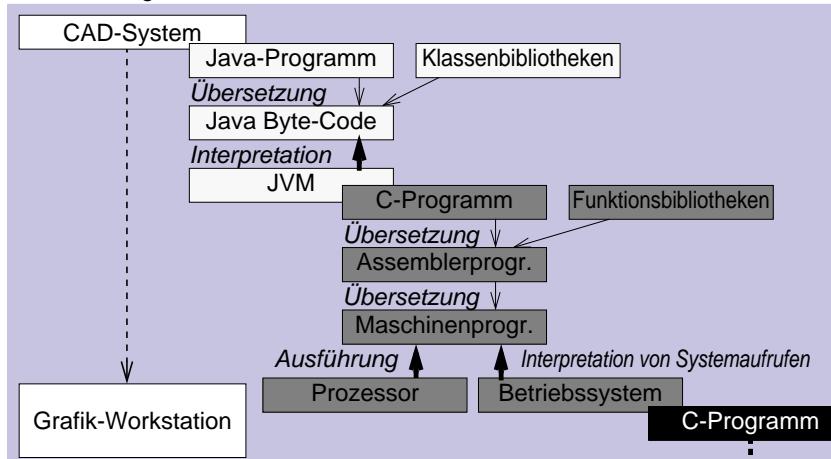


## C Systemarchitekturen

- Große Diskrepanz zwischen Anwendungsproblem und dem Ablauf der Lösung auf einer Hardware



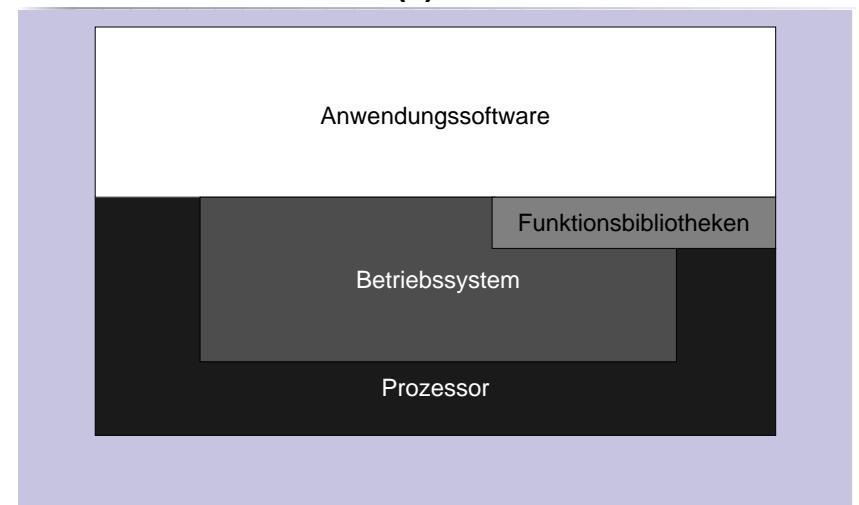
## C.1 Softwareschichten

- Anwendungs-/Problemorientierte Darstellungen
  - ◆ Modelle
    - Matlab/Simulink
    - UML
  - ◆ Programmiersprachen / höhere Abstraktionsebenen
    - Software-Komponenten
    - Java, C#, C++, Tcl/Tk
- Softwarewerkzeuge konvertieren / generieren
  - Matlab/Simulink → C
- Ausführungsumgebungen unterstützen / konvertieren / interpretieren
  - Enterprise Java Beans
  - JVM oder .NET
- ➔ Ziel: durch Prozessor ausführbarer Maschinencode

## C.1 Softwareschichten (2)

- verschiedene Ausführungsmodelle für Maschinencode
  - ◆ vollständig durch den Prozessor ausführbar
    - alle Funktionen müssen vollständig durch die Werkzeuge in direkt ausführbaren Maschinencode umgewandelt worden sein
    - keinerlei weitere Unterstützung zur Laufzeit erforderlich
    - kann so in ROM oder EPROM gespeichert werden
    - z. B. Steuerung einer Waschmaschine
  - ◆ zusätzliche Unterstützung zur Ausführungszeit erforderlich
    - "darunter liegende" Softwareschicht realisiert Dienste: Betriebssystem
    - z. B. Daten in Datei speichern, Daten über Internet übertragen
    - Realisierung: partielle Interpretation bestimmte Maschinencodes werden nicht direkt vom Prozessor ausgeführt sondern stoßen die Abarbeitung von Betriebssystemfunktionen an

## C.1 Softwareschichten (3)



## C.2 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.**“

### Andy Tanenbaum

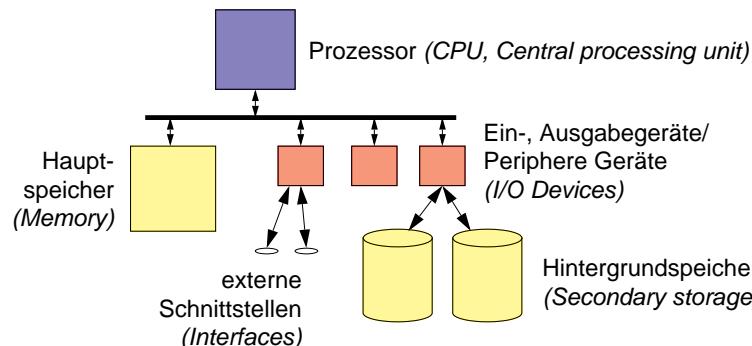
- „...eine Software-Schicht ..., die alle Teile des Systems verwaltet und dem Benutzer eine Schnittstelle oder eine *virtuelle Maschine* anbietet, die einfacher zu verstehen und zu programmieren ist [als die nackte Hardware].“

### Zusammenfassung:

- Software zur Verwaltung und Virtualisierung der Hardwarekomponenten (Betriebsmittel)
- Programm zur Steuerung und Überwachung anderer Programme

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

## 1 Verwaltung von Betriebsmitteln



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

## 1 Verwaltung von Betriebsmittel (2)

### Resultierende Aufgaben

- Multiplexen von Betriebsmitteln für mehrere Benutzer bzw. Anwendungen
- Schaffung von Schutzzonen
- 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 (peripherie Geräte, z.B. Drucker u. Ä.)
- passive, räumlich aufteilbare (Speicher, Plattenspeicher u. Ä.)

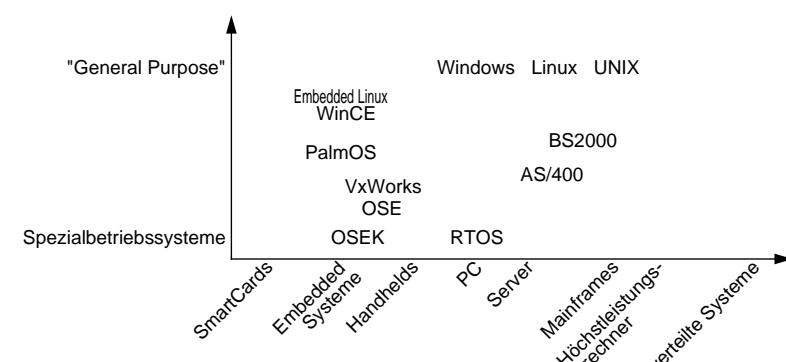
### Unterstützung bei der Fehlererholung

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

## 2 Klassifikation von Betriebssystemen

### Unterschiedliche Klassifikationskriterien

- Zielplattform
- Einsatzzweck, Funktionalität



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

## 2 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

## 3 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

## 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)