



# ***Betriebssysteme (BS)***

## **Zusammenfassung und Ausblick**

<http://ess.cs.tu-dortmund.de/DE/Teaching/SS2017/BS/>

---

**Olaf Spinczyk**

[olaf.spinczyk@tu-dortmund.de](mailto:olaf.spinczyk@tu-dortmund.de)

<http://ess.cs.tu-dortmund.de/~os>





# Inhalt

- Anwendungsbereiche für Betriebssysteme
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
- Evaluationsergebnisse
- Ausblick
  - Lehrveranstaltung über Betriebssysteme
  - Klausur



# Inhalt

- **Anwendungsbereiche für Betriebssysteme**
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
- Evaluationsergebnisse
- Ausblick
  - Lehrveranstaltung über Betriebssysteme
  - Klausur



# Vielfalt in der Systemsoftware

## High Performance Computing

- Minimale Kommunikationslatenzen



## Arbeitsplatzsysteme

- Intuitive Benutzeroberfläche



## Sichere Systeme

- Zugriffsschutz



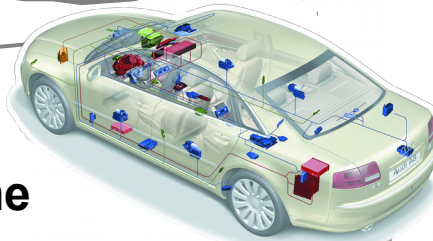
## Echtzeitsysteme

- Vorhersagbares Zeitverhalten



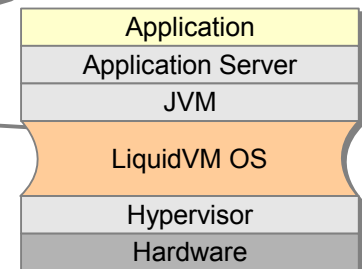
## Eingebettete und automotive Systeme

- Minimaler Speicherplatzbedarf



## Virtuelle Systeme

- Paravirtualisierung





# Die Eier legende Wollmilchsau

- Ein **Vielzweckbetriebssystem** ist für den wahrscheinlichsten Fall (den Normalfall) optimiert
- In allen Fällen, die von der künstlich definierten Norm abweichen, fallen Kosten an
- Auch ungenutzte Funktionen haben einen Preis
  - Laufzeitverbrauch durch unnötige Fallunterscheidungen
  - Speicherplatzbedarf
  - erhöhte Startzeiten
  - Verschlechterung der *cache-hit* Raten
- Besonders problematisch sind Eigenschaften, die sich auf viele Systemfunktionen auswirken
  - Linux-Kern: `grep EPERM` liefert mehr als 1200 Treffer!



# Alternative: Adaptive Systemsoftware

- **Ziel: feingranulare statische Konfigurierbarkeit**

- Anpassbarkeit an unterschiedlichste Anwendungen
- Ressourceneinsparung gegenüber Vielzwecksystemen
- Wiederverwendung und damit höhere Produktivität im Vergleich zu Individuallösungen

- **Herausforderungen:**

- Beherrschung der Variantenvielfalt
  - Analyse und Modellierung der Variabilität
- Minimierung der Modulabhängigkeiten, „*Plug&Play*“
  - Systementwurf
- Geeignete Sprachmittel für die Programmierung
  - Generizität und Wiederverwendung vs. Effizienz
- Werkzeugunterstützung
  - Techniken zur Konfigurierung



# Inhalt

- Anwendungsbereiche für Betriebssysteme
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
- **Evaluationsergebnisse**
- Ausblick
  - Lehrveranstaltung über Betriebssysteme
  - Klausur



# Evaluationsergebnisse

... findet man auf der Webseite zu BS

- Gesamtergebnis: **gut (2.01)**
  - Für eine Pflichtveranstaltung normal
  - Minimal *besser* als im Vorjahr
- Tendenzen im Vergleich zu den letzten Jahren
  - Bewertung der Vorlesung 1.95 → **1.82**
  - Bewertung der Übung 2.07 → **1.96**
- Auffälligkeiten (negativ)
  - Vorbereitung auf den Beruf → nur 2.5
  - Schwierigkeitsgrad/Aufwand der Übungen 5.68/5.91 → **6.63/6.64**
- Auffälligkeiten (positiv)
  - Übungsleiter kompetent u. gut vorbereitet → 1.65





# Evaluationsergebnisse (2)

- Einzelmeinungen:
  - “C-Programmierung wird in der Vorlesung gar nicht erklärt! Aber danach kommen riesige Programmieraufgaben, die nur dazu gut sind, Studierende zu quälen. Sie müssen vier mal so leicht sein, wie jetzt!!!”
  - “Bücher nicht mehr als PDF vorhanden/erhältlich!”
  - “Man könnte für einige Aspekte wichtige Anwendungen in der Realität darstellen.”
  - “Man könnte mehr darauf achten, motivierte Tutoren einzustellen. Durch die Lösung scrollen, egal wie einfach sie ist, ist nicht das beste zum lernen” / “Hendrik bester Mann!”
  - “[Bildchen von einem Hasen] ...Tippt das ab!”
- ESS-Kummerkasten (→ BS Webseite)
  - Für alle, die uns noch mehr sagen wollen



# Inhalt

- Anwendungsbereiche für Betriebssysteme
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
- Evaluationsergebnisse
- **Ausblick**
  - Lehrveranstaltung über Betriebssysteme
  - Klausur



# LVs der Arbeitsgruppe ESS

- Bachelor Fachprojekt
  - FP-SWA – „Software im Automobil“ (WS)
    - Praktische Durchführung einer SW-Entwicklung für Autos
- Bachelor-Arbeit (immer!)
- Master-Basis
  - SUS – „Software ubiquitärer Systeme“ (SS)
    - Basisveranstaltung für „Eingebettete und Verteilte Systeme“
    - Ein vertikaler Streifzug durch die Systemsoftware ubiquitärer Systeme
- Master-Vertiefung
  - BSB – „Betriebssystembau“ (WS)
    - Vertiefung im Bereich der Betriebssysteme
    - Bau eines eigenen PC Betriebssystems im Rahmen der Übung
  - ESS-Seminar (WS – aber nicht jedes)
    - Zuletzt „Fehlertoleranz und Echtzeit“



# Leistungsnachweise

- **Informatik-Bachelor**

- **Klausur** am 14.8.2017, Nebentermin 26.9.2017
  - Studienleistungen für BS werden bis morgen weitergeleitet.
  - Anmeldefrist für 14.8. wird um ein paar Tage verschoben: **4.8.2017**
  - Details gibt es rechtzeitig vorher auf der BS-Webseite.

- **Lehramt (4 Credit Points)**

- Klausur zum selben Zeitpunkt, aber anderer Inhalt
  - keine Fragen zu “Sicherheit” und “Multiprozessorsysteme”
  - Details gibt es rechtzeitig vorher auf der BS-Webseite.

- **Andere Studiengänge oder spezielle Fragen zur Prüfung?**

- Mail an Hendrik Borghorst (hendrik.borghorst@tu-dortmund.de)
- Angabe von Name, Matrikelnummer und Studiengang nicht vergessen



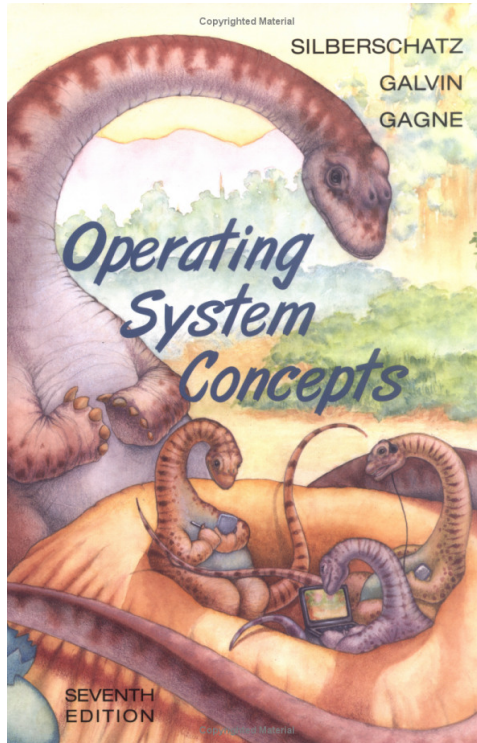
# Klausurvorbereitung

Mix aus Fragen zum Vorlesungsstoff und Übungsthemen

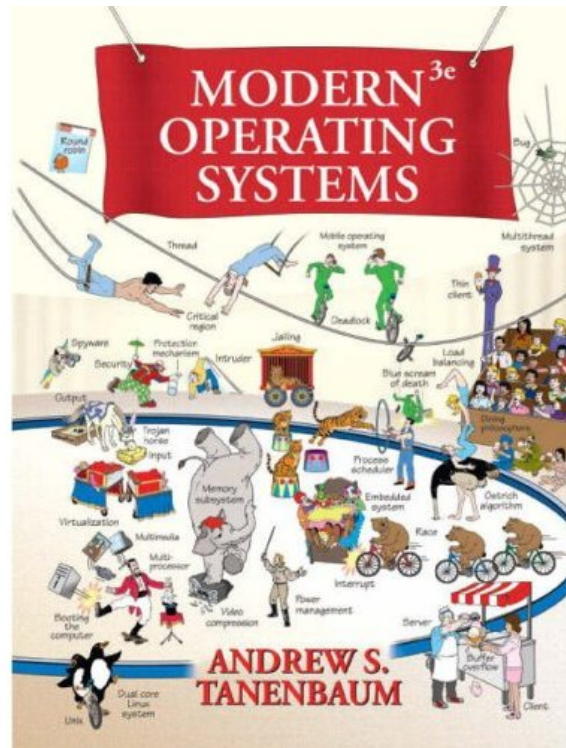
- Probeklausur
  - In den Tafelübungen dieser Woche!
- Inhalt der Folien lernen
  - Klassifizieren: Was muss ich lernen? Was muss ich begreifen?
- Übungsaufgaben verstehen, C und UNIX „können“
  - ASSESS System bleibt mindestens bis zur Klausur offen
    - Bei Fragen zur Korrektur melden
  - Am besten die Aufgaben noch einmal lösen
  - Optionale Zusatzaufgaben bearbeiten
- Beispielaufgaben lösen (→ BS Homepage)
  - „*Last Chance Test*“ und Musterlösung
  - Probeklausuren mit Besprechungsfolien
- Literatur zur Lehrveranstaltung durchlesen



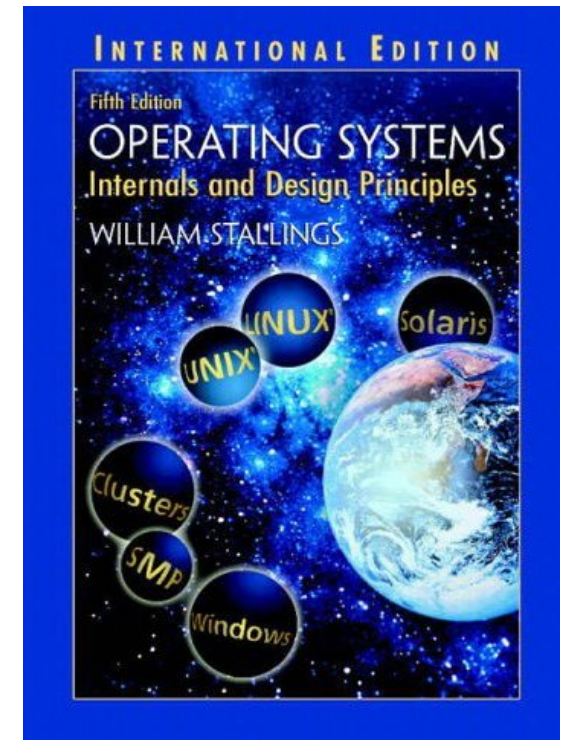
# Literatur: Standardwerke



Operating System Concepts.  
von Abraham Silberschatz,  
Peter Galvin, und Greg Gagne



Modern Operating Systems 2/e. von  
Andrew S. Tanenbaum



Operating Systems.:  
Internals and Design Principles. von  
William Stallings