

---

# Betriebssysteme (BS)

## 13. Zusammenfassung

<https://sys.cs.tu-dortmund.de/de/lehre/ss24/bs>

---

28.06.2023

**Peter Ulbrich**

[peter.ulbrich@tu-dortmund.de](mailto:peter.ulbrich@tu-dortmund.de)

[bs-problems@ls12.cs.tu-dortmund.de](mailto:bs-problems@ls12.cs.tu-dortmund.de)

<https://sys.cs.tu-dortmund.de/de/lehre/kummerkasten>

In Teilen basierend auf *Betriebssysteme* von Olaf Spinczyk, Universität Osnabrück

# Probeklausur

- Am 15.07. im Rahmen der Vorlesung (SRG1, H.001)
- 45min Klausur, 45min Besprechung
- Anmeldung in AsSESS bis zum **10.07.2024 (23:59)**

[https://sys-sideshow.cs.tu-dortmund.de/ASSESS/?do=exerciselist&lectureid=470&filter\\_s=SoSe&filter\\_y=2024](https://sys-sideshow.cs.tu-dortmund.de/ASSESS/?do=exerciselist&lectureid=470&filter_s=SoSe&filter_y=2024)



# Inhalt

- **Anwendungsbereiche für Betriebssysteme**
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
  
- Klausur

# Vielfalt in der Systemsoftware

## High Performance Computing

- Minimale Kommunikationslatenzen



## Arbeitsplatzsysteme

- Intuitive Benutzeroberfläche



## Sichere Systeme

- Zugriffsschutz



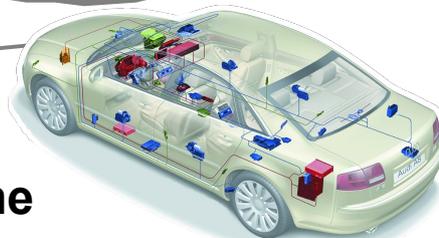
## Echtzeitsysteme

- Vorhersagbares Zeitverhalten



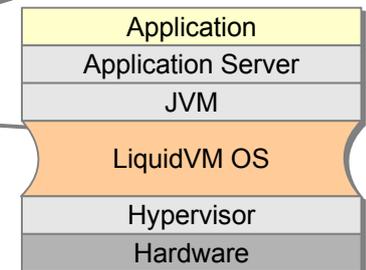
## Eingebettete und automotive Systeme

- 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 3600 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

# LVs der Arbeitsgruppe Systemsoftware

- **Bachelor-Arbeit** (immer!)
  - **empirische Arbeiten** z.B. aus den Betriebssystem-Unterbereichen  
Zuverlässigkeit oder Synchronisation
- **Master-Vertiefung Verlässliche Systemsoftware**
  - Wie werden **verlässliche Systeme** entwickelt?
  - Zuverlässige (robuste) Software entwickeln (externe Fehler!)
  - Software zuverlässig entwickeln (fehlerfreie Implementierung!)
- **Master-Vertiefung Betriebssystembau**
  - Wie werden **Betriebssysteme** tatsächlich gebaut?
  - Betriebssystemkomponenten und deren Entwurf
  - PC-Technologie aus Betriebssystemersicht

# Inhalt

- Anwendungsbereiche für Betriebssysteme
  - Vielfalt der Anforderungen
  - Adaptive Systemsoftware
- Klausur

# Leistungsnachweise

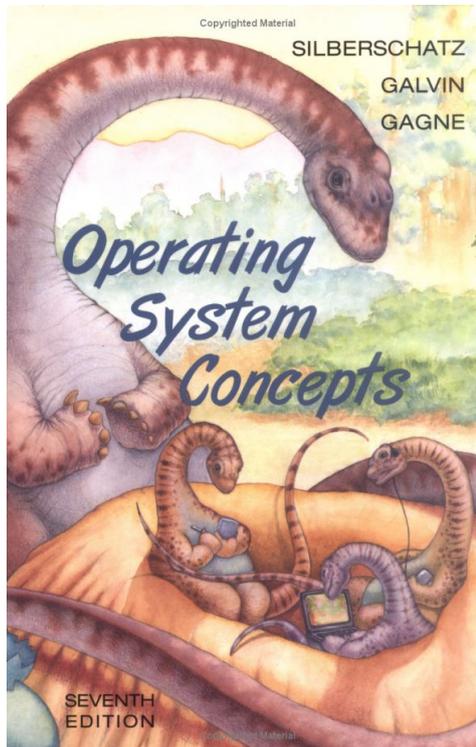
- **Bachelor Informatik / Bachelor Angewandte Informatik**
  - Klausur am **07.08.2024**, Wiederholungstermin **19.09.2024**
    - Studienleistungen werden in den kommenden Tagen eingetragen.
    - Achtung: Gültigkeit alter Studienleistungen sehen wir nicht → PV fragen!
    - **Anmeldezeitraum** geht bis zum **23.07.** → nicht vergessen!
    - Details gibt es rechtzeitig vorher in Moodle und/oder per E-Mail an alle angemeldeten Teilnehmerinnen.
    - Falls Sie (auch kurzfristig) doch nicht mitschreiben möchten:  
**Bitte unbedingt abmelden!**
  
- **Andere Studiengänge oder organisatorische Fragen zur Prüfung?**
  - Ausnahmen oft bei: Ausserfachliche Qualifikation, Zulassungsaufgaben, o.ä.
  - Im Zweifel lieber fragen!
  - Mail an [bs-problems@ls12.cs.tu-dortmund.de](mailto:bs-problems@ls12.cs.tu-dortmund.de)
  - Angabe von Name, Matrikelnummer und Studiengang nicht vergessen

# Klausurvorbereitung

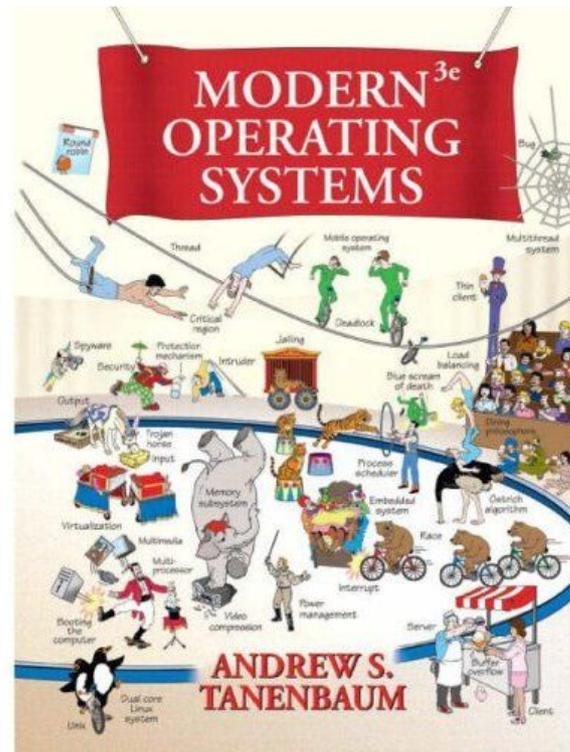
## Mix aus Fragen zum Vorlesungsstoff und Übungsthemen

- Inhalt der Folien lernen
  - Klassifizieren: Was muss ich **lernen**? Was muss ich **begreifen**?
- Übungsaufgaben verstehen, C und UNIX „können“
  - Korrekturkommentare zur Lösung in AsSESS lesen
  - Ggf. die Aufgaben noch einmal lösen, optionale Zusatzaufgaben bearbeiten
- Beispielaufgaben lösen (→ BS-Webseite)
  - Probeklausur probieren
- **HelpDesk und Fragestunde-Termine vor den beiden Klausuren nutzen (Termine siehe Moodle)**
- ergänzend Literatur zur Lehrveranstaltung konsultieren

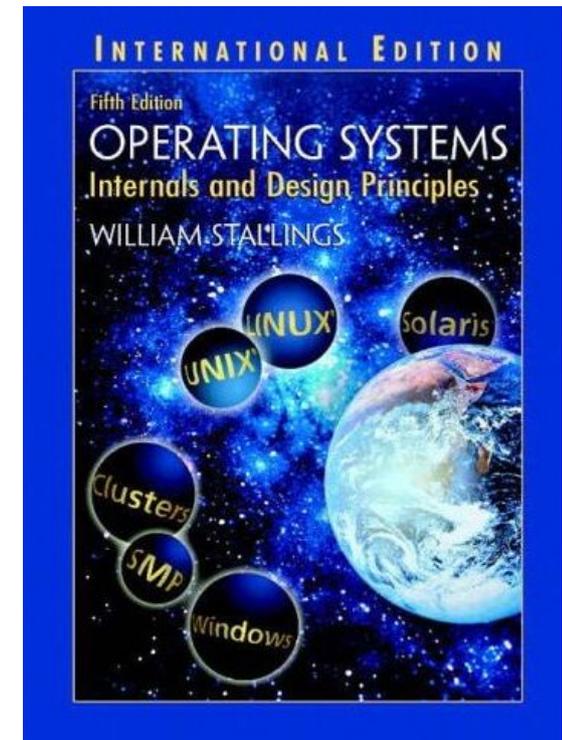
# Literatur: Standardwerke



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