

---

# Betriebssysteme (BS)

## 12. Zusammenfassung

<https://sys.cs.tu-dortmund.de/de/lehre/ss25/bs>

---

07.07.2025

**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 14.07. im Rahmen der Vorlesung (SRG1, H.001)
- **Bitte seien Sie bereits um ab 12:00 Uhr anwesend!**
- 45min Klausur, 45min Besprechung
- Anmeldung in AsSESS bis zum **10.07.2025 (23:59)**

[https://sys-sideshow.cs.tu-dortmund.de/ASSESS/index.php?do=exerciseslist&lectureid=489&filter\\_s=SoSe&filter\\_y=2025](https://sys-sideshow.cs.tu-dortmund.de/ASSESS/index.php?do=exerciseslist&lectureid=489&filter_s=SoSe&filter_y=2025)



# Inhalt

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

# Inhalt

## ■ Anwendungsbereiche für Betriebssysteme

- Vielfalt der Anforderungen
- Adaptive Systemsoftware

## ■ Klausur

## ■ Evaluation

# Vielfalt in der Systemsoftware

## High Performance Computing

- Minimale Kommunikationslatenzen



## Arbeitsplatzsysteme

- Intuitive Benutzeroberfläche



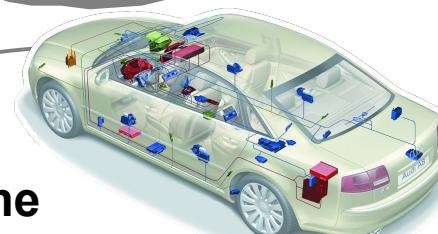
## Sichere Systeme

- Zugriffsschutz



## Eingebettete und automotive Systeme

- Minimaler Speicherplatzbedarf



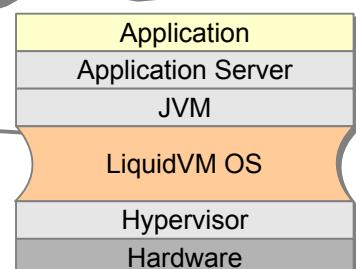
## Echtzeitsysteme

- Vorhersagbares Zeitverhalten



## 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 Betriebssystemsicht

# Inhalt

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

# Inhalt

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

# Leistungsnachweise

## ■ Bachelor Informatik / Bachelor Angewandte Informatik

- Klausur am **22.07.2025**, Wiederholungstermin **29.09.2025**
  - Studienleistungen werden in den kommenden Tagen eingetragen.
  - Achtung: Gültigkeit alter Studienleistungen sehen wir nicht → PV fragen!
  - **Anmeldezeitraum** geht bis zum **15.07. (23:59 Uhr)** → nicht vergessen!
  - **Abmeldezeitraum** endet für Informatik ebenfalls am **15.07. (23:59 Uhr)**
  - Details gibt es rechtzeitig vorher in Moodle und/oder per E-Mail an alle angemeldeten Teilnehmerinnen.
  - Nachteilausgleiche müssen vorab eingereicht werden

## ■ Andere Studiengänge oder organisatorische Fragen zur Prüfung?

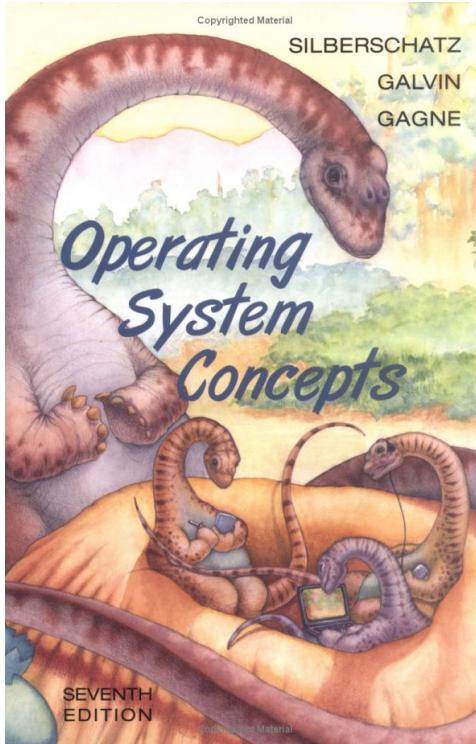
- Ausnahmen oft bei: Ausserfachliche Qualifikation, Zulassungsauflagen, o.ä.
- Im Zweifel lieber fragen!
- Mail an [bs-problems@ls12.cs.tu-dortmund.de](mailto:bs-problems@ls12.cs.tu-dortmund.de)
- Angabe von Name, Matrikelnummer und Studiengang nicht vergessen

# Klausurvorbereitung

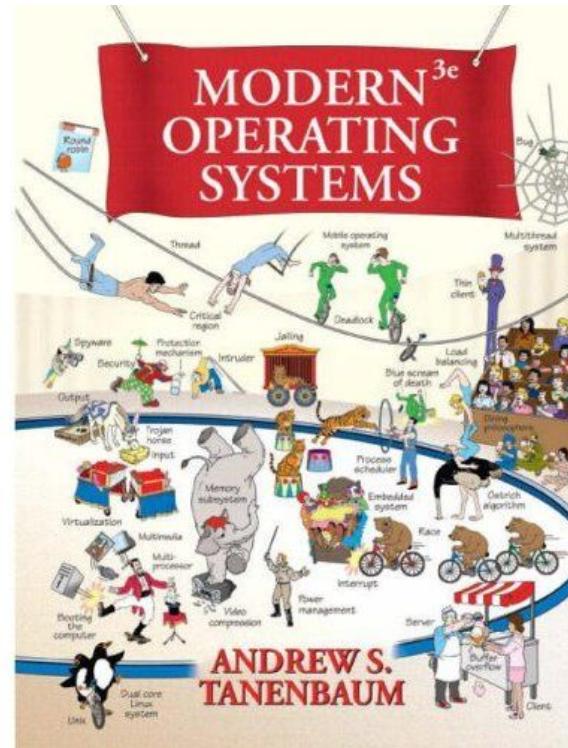
Mix aus Fragen zum Vorlesungsstoff und Übungsthemen

- 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

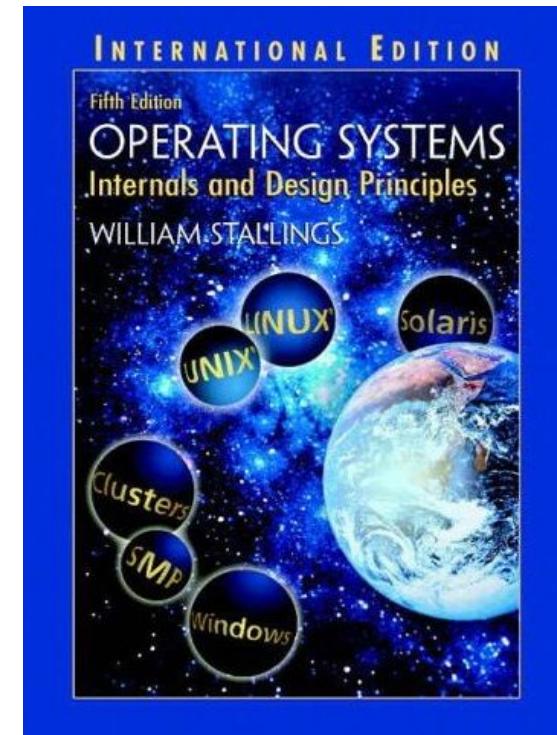
# Literatur: Standardwerke



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



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



**Operating Systems.: Internals and D**  
esign Principles. von William  
Stallings

# Inhalt

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

# Inhalt

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

# Evaluation: Übersicht

## Globalindikator

Bewertung der Organisation der Veranstaltung  
(Skalenbreite: 5)

Bewertung der Lernumgebung (Skalenbreite: 5)

Bewertung des Stoffes der Lehrveranstaltung  
(Skalenbreite: 5)

Bewertung des Live-Angebots der Veranstaltung (z.  
B. Vorlesungen oder Webinare) (Skalenbreite: 5)

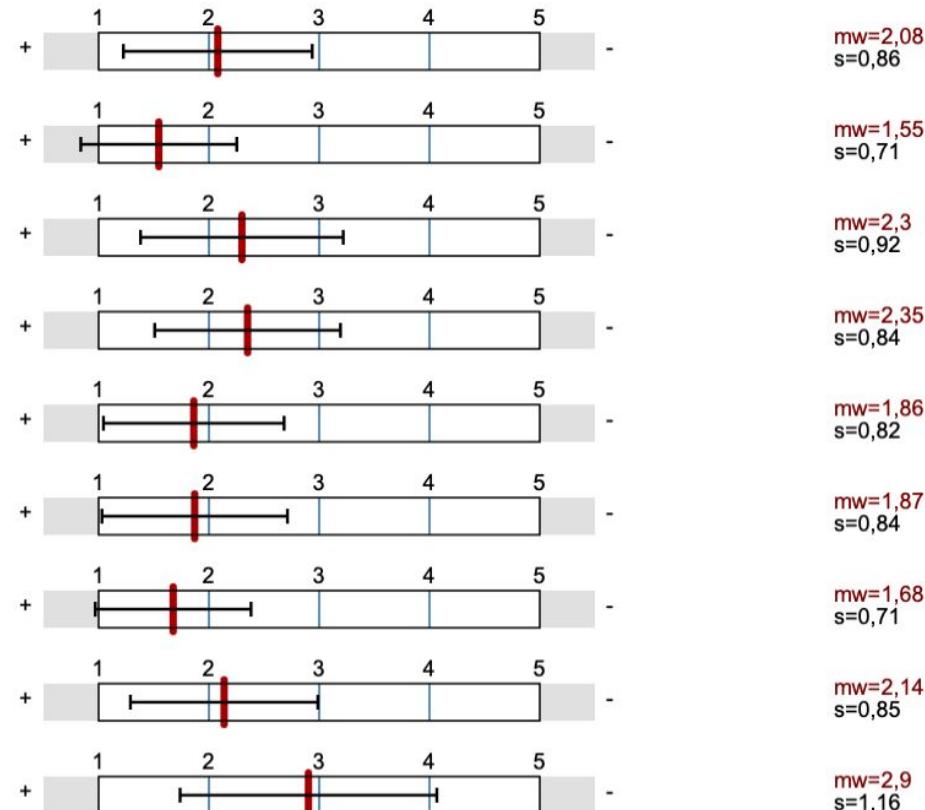
Bewertung der Vorlesungsvideos (Skalenbreite: 5)

Bewertung des Vorlesungsmaterials (Skalenbreite:  
5)

Bewertung der Übung (Skalenbreite: 5)

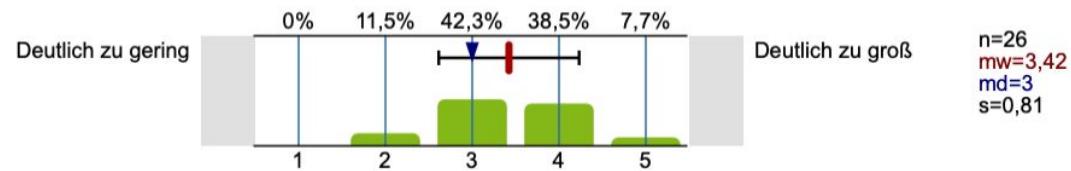
## Vermittelte Kompetenzen

Die Antworten innerhalb dieses Abschnitts fließen  
nur in die Gesamtübersicht der Lehre der Fakultät  
ein, aber sind nicht für die vergleichende  
Bewertung der Veranstaltung relevant.  
(Skalenbreite: 5)

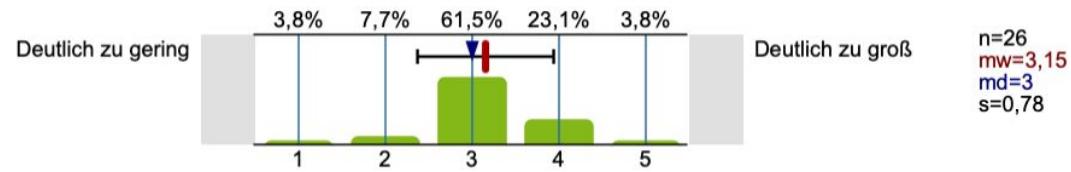


# Evaluation: Aufwand

8.13) Der Schwierigkeitsgrad der Übungsaufgaben ist



8.14) Der Aufwand zum Lösen aller Übungsaufgaben ist



- Das zweite Semester ist sehr überfüllt, wenn man versucht Regelstudienzeit zu schaffen opfert man sein ganzen Hobbys und Freizeit. Ich muss aber zugeben, es ist sehr angenehm, dass die Abgaben jede zweite Woche sind. Sollte für jedes Modul bitte sein. Insgesamt: Interessante Vorlesung, aber die Implementations Aufgaben werfen einen ins kalte Wasser falls man kein C kann, zu wenig Zeit für die Nachbearbeitung.

## 2. Bewertung der Organisation der Veranstaltung

2.6) Anmerkungen zur Organisation bzw. Onlinepräsenz (Webseite, Moodle-Raum) der Veranstaltung:

- Der Moodle Raum ist sehr gut organisiert und es gibt sehr umfangreiche Informationen
- Moodle war sehr überfrachtet. Durch die Kategorisierung ist es inzwischen OK, aber immer noch sehr viel Text.
- Nichts
- Videoaufzeichnungen der Vorlesung und Übung sind online vorhanden.

## 4. Bewertung des Stoffes der Lehrveranstaltung

4.4) Anmerkungen zum Stoff der Veranstaltung:

- Ein guter Überblick über Betriebssysteme und die Funktionsweise eines Rechners.
- Es gibt sehr viel Bonusmaterial auf Moodle. Aber ich kenne oft den Kontext nicht ganz.
- Keine

## 5. Bewertung des Live-Angebots der Veranstaltung (z. B. Vorlesungen oder Webinare)

5.17) Kommentare zu den Vorlesungen oder Webinaren:

- Die Folien wären angenehmer und besser durchsuchbar wenn sie in Decker erstellt werden würden und nicht einfach die PDF Folien in Decker hochgeladen. Danke
- Folien sind manchmal voll. Aber gut strukturiert.
- Keine (2 Nennungen)
- Man hätte noch mehr Beispiele für die Programmierung machen können weil ohne diese Einleitung für die C-Programmierung vor allem diese Befehle für das Betriebssystem die in C vorhanden sind war schwer