

---

# Betriebssysteme (BS)

## 01. Einführung

<https://sys.cs.tu-dortmund.de/DE/Teaching/SS2022/BS/>

---

06.04.2021

**Peter Ulbrich**

peter.ulbrich@tu-dortmund.de  
bs-problems@ls12.cs.uni-dortmund.de

Basierend auf *Betriebssysteme* von Olaf Spinczyk, Universität Osnabrück

# Lernziele

- Vorgänge innerhalb von **Rechensystemen** ganzheitlich verstehen



- Imperative **Systemprogrammierung (in C)** in Grundzügen kennenlernen
  - Im Kleinen für **Dienstprogramme** praktizieren
  - Im Großen durch **Betriebssysteme** erfahren
- Grundlagenwissen über **Betriebssysteme**, deren Struktur, Funktion, Algorithmen und Umsetzung

# Voraussetzungen

- Grundkenntnisse aus Vorlesungen
  - DAP1
  - Rechnerstrukturen
- Neugier
- Fähigkeit zum selbständigen Arbeiten
  - Teamfähigkeit
  - Beherrschung der Regeln guter wissenschaftlicher Praxis

# Inhalt

- Organisation
- Begriffe und Überblick
- Literatur
  
- C-Crashkurs

# Inhalt

- **Organisation**
- Begriffe und Überblick
- Literatur
- C-Crashkurs

# Organisation

- **Vorlesung (VL)** → Seminarraumgebäude 1 / H.001
  - 1,5 Std. wöchentlich, Mi. 16:00-18:00 (c.t.)
  - Vorstellung und detaillierte Behandlung des Lehrstoffs
- **Tafelübung (TÜ)** → Verschiedene Räume (siehe Webseite/Moodle)
  - 1,5 Std. Gruppen **alternierend** alle 2 Wochen
  - Einteilung in Gruppen **W1** (ungerade) und **W2** (gerade) Woche
  - Übungen starten in KW15 mit W1
  - Vertiefung der VL sowie Besprechung der Übungsaufgaben
- **HelpDesk (RÜ)** → Pool 4.031 (OH12), Matrix
  - Wöchentlich, Mo. 14:00-16:00, Di. 10:00-12:00 und Mi. 12:00-14:00
  - Hilfestellung beim Bearbeiten der Übungsaufgaben (C Problemen)
  - Fragen zur Übung
  - **Ein HiWi ist immer vor Ort! Nutzt dieses Angebot!**
- **Vor-/Nacharbeit**
  - N Std. wöchentlich,  $0 < N < 165$

**in der Summe: 3 SWS**

# Dozent und Übungsleiter



Peter Ulbrich



Andreas Lang



Alexander Lochmann



Jan Bessai



Davut Bozdag



Felix Homa



Lukas Kerren



Emilio Pielsticker

Helpdesk



Samanta Scharmacher



Marvin Weiler

# Tafelübungen

## ■ Anmeldung über **ASSESS**

<https://ess.cs.tu-dortmund.de/ASSESS/index.php?do=exerciselist&lectureid=426>

- mit Matrikelnummer und Uni-Mailadresse
- Auswahl aus 6 Übungszeiten
- Angaben von „Cliques“ möglich

## ■ Prioritäten-Verfahren bis Fr, **08.04. 12:00 Uhr.**

- danach: FCFS  
(Nachzügler, Gruppe wechseln, ...)

Wann?	Priorität	Beliebtheit (# Prio 1)
Di : 10:00 - 12:00	1 - höchste	<div><div></div></div> 29
Di : 14:00 - 16:00	1 - höchste	<div><div></div></div> 23
Di : 16:00 - 18:00	1 - höchste	<div><div></div></div> 11
Mi : 10:00 - 12:00	1 - höchste	<div><div></div></div> 29
Mi : 12:00 - 14:00	1 - höchste	<div><div></div></div> 19
Mi : 14:00 - 16:00	1 - höchste	<div><div></div></div> 29
Speichern		

## ■ Automatische Zuordnung zu passender Übungsgruppe

- Kalenderwoche und Übungsleiter legt der Optimierer fest
- Abgabegruppen sind nicht bindend (über das Semester)
- Getrennte Abgabegruppen: Früher Termin gilt!



# Übungsaufgaben

- Theoriefragen und praktische Programmieraufgaben
- Vorstellung der neuen Aufgaben in der TÜ
- Bearbeitung in **Dreiergruppen**  
(Gruppen müssen nicht in derselben TÜ angemeldet sein)
  - Kein Kopieren von anderen Gruppen!
  - **Wer ein Plagiat abgibt, erhält keine Studienleistung**
- Ausgabe der Übungsaufgabe erfolgt für alle Montags
  - Konkrete Daten siehe Webseite/Aufgabenblatt
- Abgabe der Übungsaufgabe in der Woche nach der TÜ
  - Erste Woche (W1) bis **Do. 12:00**
  - Zweite Woche (W2) bis **Mo. 12:00**
  - Näherungsweise mindestens eine Woche zwischen eigener TÜ und Abgabe
- Vorstellung der Lösung in der folgenden TÜ

# Leistungskontrolle/Anforderungen

## ■ Studienleistung

- 6 Übungsaufgaben (A0-A5), unterteilt in 2 Gruppen: A0-A2 und A3-A5
- Erreichen von **jeweils  $\geq 50\%$  der Punkte in beiden Aufgabengruppen**
  - Punkte(A0) + Punkte(A1) + Punkte(A2)  $\geq 15$
  - **UND** Punkte(A3) + Punkte(A4) + Punkte(A5)  $\geq 15$

## ■ Prüfung: Klausur nach Ende des Semesters (03.08. oder 08.09.)

- Studienleistung ist Voraussetzung für die Teilnahme (*ggf. Gültigkeit prüfen*)
- Relevant ist der Vorlesungs- **und** Übungsstoff!

## ■ Diese Anforderungen gelten für:

- Bachelor-Studium Informatik und Angewandte Informatik
- Lehramt (5 CP/Bachelor, 4 CP/Master)
- Schüler-Studierende
- *Alle anderen bitte melden!*

# Feedback

- Fragen zur laufenden Vorlesung / Übung?
  - Jederzeit! (asynchron)  
→ [Matrix #bs-helpdesk:fachschaften.org](https://matrix.fachschaften.org/#/room/#bs-helpdesk:fachschaften.org)
  - In den Tafelübungen  
→ [Persönlich](#)
  - Im Helpdesk der Übungen  
→ [Persönlich](#)
  
- Kommentare/Anregungen zu Organisation, Vorlesung, Übung?
  - Persönlich an den Dozenten
  - per eMail an [bs-problems@ls12.cs.uni-dortmund.de](mailto:bs-problems@ls12.cs.uni-dortmund.de)
  - Rauchzeichen  
→ Wir beissen nicht!

# Hybride Lehre im SS2022

## ■ Stoff der Vorlesung

- Folien sind in Moodle verfügbar
- Aufzeichnung der Vorlesung aus dem SS2021 bleibt verfügbar  
→ Ohne Gewähr auf Übereinstimmung

## ■ Stoff der Übungen

- Es gibt keine Aufzeichnung
- Wir bemühen uns **eine** Online-Übung anzubieten (siehe Moodle)  
→ Nach Verfügbarkeit, es besteht kein Anspruch

## ■ Rechnerarbeit / Helpdesk

- Nutzen Sie den Matrix-Raum (asynchron!)
- Wir bemühen uns Ihnen zu helfen  
→ Nach Verfügbarkeit, es besteht kein Anspruch

# Inhalt

- Organisation
- **Begriffe und Überblick**
- Literatur
- C-Crashkurs

**Los geht's ...**

Quizfrage:

**Was ist ein Betriebssystem?**

## Definitionen (1)

*„Ein Computer ist, wenn er genau betrachtet wird, nur eine Ansammlung von Plastik und Metall, das zur Leitung von Strom benötigt wird. Dieser „Industriemüll“ kann somit nicht ausschließlich das sein, was wir unter einem modernen Computer verstehen, etwas, das dem **Computer „Leben“ einhaucht** und ihn zu dem Werkzeug unseres Jahrhunderts macht.*

*Es ist das Betriebssystem, das die **Kontrolle** über das Plastik und Metall (Hardware) übernimmt und anderen Softwareprogrammen (Excel, Word, . . . ) eine **standardisierte Arbeitsplattform** (Windows, Unix, OS/2) schafft.“*

Ewert et al., Literatur zu „Freehand 10“

## Definitionen (2)

*„**Be'triebs-sys-tem** Programmbündel, das die **Bedienung eines Computers** ermöglicht.“*

Universalwörterbuch Rechtschreibung

*„Summe derjenigen Programme, die als **residentialer Teil** einer EDV-Anlage für den Betrieb der Anlage und für die Ausführung der Anwenderprogramme erforderlich ist.“*

Lexikon der Informatik

*„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die **Abwicklung von Programmen** steuern und überwachen.“*

DIN 44300



## Definitionen (3)

*„Ein Programm, das als **Vermittler** zwischen Rechnerbenutzer und Rechnerhardware fungiert. Der Sinn des Betriebssystems ist eine Umgebung bereitzustellen, in der Benutzer bequem und effizient Programme ausführen können.“*

Silberschatz [1]

*„Eine **Softwareschicht**, die alle Teile des Systems verwaltet und dem Benutzer eine Schnittstelle oder **virtuelle Maschine** anbietet, die leichter zu verstehen und zu programmieren ist [als die darunterliegende reale Maschine].“*

Tanenbaum [2]

# Vielfalt der Anforderungen

## High Performance Computing

Minimale Kommunikations-  
latenzen



## Arbeitsplatz- systeme

Intuitive Benutzer-  
oberfläche



## Sichere Systeme

Zugriffsschutz



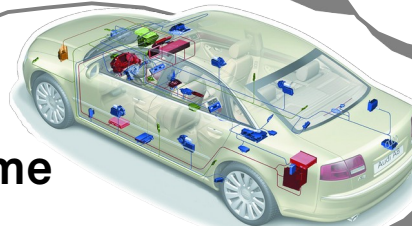
## Echtzeitsysteme

Vorhersagbares  
Zeitverhalten



## Eingebettete und automotive Systeme

Minimaler  
Speicherplatzbedarf



## Zwischenfazit (Fortsetzung nächste Woche)

- Es gibt viele Auslegungen des Begriffs **Betriebssystem**
- Festhalten kann man:
  - Das Betriebssystem dient den Anwendern bzw. deren Anwendungsprogramm(en) → **Sinnvolle Schnittstellen und Konzepte**
  - Es muss die Hardware genau kennen und den Umgang damit vereinfachen → **Geeignete Abstraktionen**
  - Das Betriebssystem verwaltet und überwacht das Gesamtsystem → **Ressourcen- und Rechteverwaltung**
- Hardware und Anwendungsanforderungen bestimmen die Dienste des Betriebssystems
  - Struktur und Funktionsweise ergeben sich entsprechend
  - Um zu verstehen, welche Abstraktionen Betriebssysteme heute anbieten, werden wir deren Entwicklungsgeschichte im Zusammenhang mit der Technologieentwicklung betrachten

# Ausblick auf den Stoff der Veranstaltung

- Kontrollflussabstraktionen
  - Fäden, Prozesse
- Prozessorzuteilung
- Kooperation und Konkurrenz von Kontrollflüssen
  - Synchronisation, Verklemmungen
- Speicherverwaltung
- Ein- und Ausgabe
- Dateisysteme
- Sicherheit
- Multiprozessorsysteme

# Inhalt

- Organisation
- Begriffe und Überblick
- **Literatur**
- C-Crashkurs

# Empfohlene Literatur

- [1] A. Tanenbaum, Herbert Bos  
**Moderne Betriebssysteme** (4. Ausgabe).  
Prentice Hall, 2009. ISBN 978-3868942705
- [2] A. Silberschatz et al.  
**Operating System Concepts** (9. Ausgabe).  
Wiley, 2014. ISBN 978-1118093757
- [3] Dausmann, Manfred et al.  
**C als erste Programmiersprache** (8. Ausgabe).  
Springer, 2014. ISBN 978-3834822710
- [4] B. W. Kernighan, D. M. Ritchie.  
**The C Programming Language**.  
Prentice-Hall, 1988. ISBN 978-0131103627
- [5] R. Stevens, S. Rago.  
**Advanced Programming in the UNIX Environment**.  
(3. A.) Addison-Wesley, 2013. ISBN 978-0321637734

