



# Isolation und Schutz in Betriebssystemen

## 1. Einführung

Michael Schöttner

1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

## 1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

- Vorlesung: 1 SWS
  - Montag, 10:30 – 12:00Uhr, 25.12.01.51
  
- Übung: 3 SWS
  - Montag, 10:30 – 12:00Uhr, 25.12.01.51
  - Donnerstag, 12:30 – 14:00Uhr, 25.12.01.51
  
- Hauptziel: Jede Person implementiert Schutz- und Isolationsmechanismen in einem 64 Bit Betriebssystem für x86
  - Die Implementierung erfolgt in Rust & Assembler
  - Ausgangspunkt ist das eigene Betriebssystem aus BS-Entwicklung oder alternativ wird ein einfaches Betriebssystem bereitgestellt

# 1.1 Organisation

- Vorlesungs- und Übungsunterlagen in ILIAS
- Rocketchat via ILIAS
- Voraussetzungen:
  - Betriebssystem-Grundkenntnisse empfohlen
  - Spaß an hardwarenaher Programmierung
  - Durchhaltevermögen

- Relevante Inhalte für die Aufgaben

- Ziel: Schutz- und Isolationsmechanismen selbst entwickeln → verpflichtende Abgabe
- Das Betriebssystem muss in Qemu laufen!
  - Rechner zum Testen auf echter Hardware werden in der Übung bereitgestellt
- Die Aufgaben bauen aufeinander auf
- Die Programmierung erfolgt in Rust und punktuell mit NASM
  - Für das Debugging verwenden wir GDB und Qemu

- Typsichere, schnelle und moderne Programmiersprache
- Bietet eine Compile-Time Garbage Collection
  - Keine Null-Pointer
  - Keine kaputten Pointer (sofern nicht unsafe) genutzt wird
- Interessante Artikel:
  - <https://msrc.microsoft.com/blog/2019/07/why-rust-for-safe-systems-programming/>
  - <https://msrc.microsoft.com/blog/2019/07/we-need-a-safer-systems-programming-language/>
- Linux-Kernel erlaubt inzwischen auch Kernel-Module in Rust
  - <https://www.heise.de/news/Rust-Code-im-Linux-Kernel-Merge-steht-laut-Linus-Torvalds-ab-Linux-5-20-bevor-7154453.html>



- 5 ECTS für den Studiengang Master Informatik
- Mündliche Prüfung nach Ende der Vorlesungszeit
- Die Prüfung geht „nur“ über das eigene Betriebssystem
  - In der Prüfung muss eine Demo vorgeführt werden
  - Der Quelltext muss mindestens eine Woche vor der Prüfung abgegeben werden
  - Für die Note 1.0 müssen alle Funktionen aus allen Übungsblättern vorzeigbar und erklärbar sein.
  - Die Beantwortung der Fragen zu den Übungsaufgaben fließt auch in die Bewertung ein.

1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

## 1.2 Lernziele und Nutzen

- Vertiefung des Wissens von Betriebssystem-Techniken/-Konzepten
  - Funktionsweise
  - Struktur
  - Implementierung
- Schutz- und Isolationsfunktionen in Betriebssystemen verstehen und implementieren
- „Weniger auswendig lernen, stattdessen mehr selbst tun“

## 1.2 Lernziele und Nutzen

- Schutz- und Isolation auf Betriebssystemebene essentiell wichtig
- „If you want to travel around the world and be invited to speak at a lot of different places, just write a UNIX operating system“ - Linus Torvalds

1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

1. Einleitung
2. Segmentierung und Tasks
3. Systemaufrufe
4. Paging
5. Kernel Isolation

1. Privilegientrennung (User-Level Threads)
2. Systemaufrufe
3. Physikalische Speicherverwaltung
4. Paging (Isolation der User-Stacks)
5. Trennung Kernel- und User-Code
6. Prozesse (Isolation von Anwendungen untereinander)
7. VMAs, User-Level Heaps, dynamisch wachsender User-Stack

User  
mode

Kernel  
mode

Anwendungen als Kernel-Threads (nur ein Adressraum für alles)

Bibliotheken

Speicher-  
Verwaltung

Threads & Scheduler

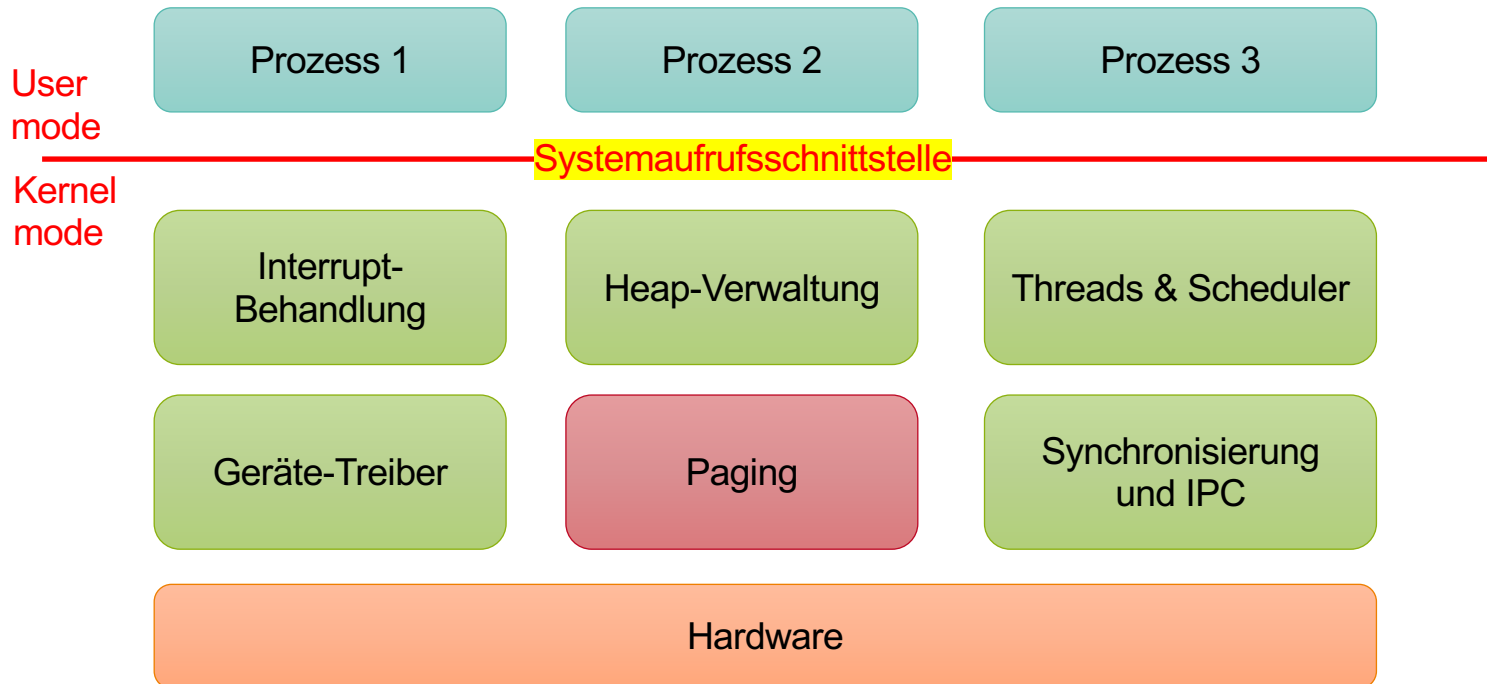
Geräte-Treiber

Interrupt-  
Behandlung

Synchronisierung

Hardware





1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

- <https://OSDev.org>
- Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A, 3B, 3C, & 3D): System Programming Guide  
<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html>

1.1 Organisation

1.2 Lernziele und Nutzen

1.3 Inhalte

1.4 Literatur

1.5 Einordnung der Veranstaltung

# 1.5 Einordnung der Veranstaltung

## Bachelor

Betriebssysteme und Systemprogrammierung, 4V+2Ü, 10 LP

Bachelor Seminar „Moderne Hardware“, 4S, 5 LP

## Master

Betriebssystem-Entwicklung,  
1V+3Ü, 5 CP

Aktuelle Trends in der Systemsoftware,  
2S, 5 CP

Isolation und Schutz in Betriebssystemen,  
1V+3Ü, 5 CP