

Konzepte von Betriebssystem-Komponenten (KvBK)

Einführung

22. April 2020

Phillip Raffeck

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Organisatorisches

Phillip Raffeck



(Raum 0.036)

raffeck@cs.fau.de

- **Termin:** Mittwoch, 10:15-11:45, ~~Raum 0.035~~ Zoom/Jitsi
- **Webseite:** www4.cs.fau.de/Lehre/SS20/PS_KVBK/
- **Mailingliste** an alle Teilnehmer & Betreuer
kvbk@lists.informatik.uni-erlangen.de

Semesterplan

22.04	Organisation, Einführung und Themenvergabe
29.04	Moderner Wissenschaftsbetrieb
<hr/>	
06.05	Vortragstechniken
13.05	Arbeitstechnik und LaTeX
20.05	Praktische Übung: Lesen & Diskutieren von Papieren
27.05	Sauberes wissenschaftliches Arbeiten
03.06	entfällt
10.06	Praktische Übung: Papier- und Vortragsdiskussion
<hr/>	
17.06	entfällt
24.06	entfällt
01.07	entfällt
<hr/>	
08.07	Vorträge 1
15.07	Vorträge 2
22.07	Vorträge 3

Einarbeitung

- Eigenständiges Bearbeiten eines Themas
- Literaturrecherche: Vorgegebene Papiere als Ausgangsbasis
- Abgabe eines Exposés

Einarbeitung

- Eigenständiges Bearbeiten eines Themas
- Literaturrecherche: Vorgegebene Papiere als Ausgangsbasis
- Abgabe eines Exposés

Ausarbeitung

- Erstellen einer Ausarbeitung (4-6 Seiten)
- ACM Standard Proceedings Template (ACM SGI style)
- Berücksichtigen der Wissensbasis
- Sprache: Deutsch oder Englisch

- *keine* reine Nacherzählung/Übersetzung
- direkte Übernahme von Abbildungen vermeiden
- Aufgreifen und Vertiefen einzelner Aspekte
- Herausarbeiten eigener Fragestellung
- eigene Literaturrecherche
- Exposé: *roter Faden*

Vortrag

- Foliensatz zur Ausarbeitung
- *Optional: i4-Beamertemplate* (Beamer/neo)
- Berücksichtigen der *Hinweise zur Erstellung der Folien*
- Sprache: Deutsch oder Englisch
- Vortrag im Rahmen des Seminars
 - 20-minütiger Vortrag
 - 10-minütige Diskussion zum Vortrag

- Abgabe einer Exposé
 - Bis zum **27.05.2020**
 - Feedback von Betreuer
- Erste Version der Ausarbeitung
 - **Drei Wochen** vor der Präsentation
 - Feedback von Betreuer
- Erste Version der Folien
 - **Zwei Wochen** vor der Präsentation
 - Feedback von Betreuer
- Finale Version der Ausarbeitung
 - **Freitag** vor der Präsentation
 - Betreuer verteilt die Ausarbeitung über die Mailingliste

Semesterplan

22.04

29.04

06.05

13.05

20.05

27.05

Exposé

03.06.

10.06

17.06

Ausarbeitung

24.06

Folien

Ausarbeitung

01.07

Ausarbeitung

Folien

Ausarbeitung

08.07

Vorträge 1

Ausarbeitung

Folien

15.07

Vorträge 2

Ausarbeitung

22.07

Vorträge 3

Arbeitsmittel

- Verwendung von Git empfohlen
→ <https://gitlab.cs.fau.de/>
- Abgabe der Ausarbeitung/Folien per Git (oder E-Mail)

Arbeitsmittel

- Verwendung von Git empfohlen
→ <https://gitlab.cs.fau.de/>
- Abgabe der Ausarbeitung/Folien per Git (oder E-Mail)

Organisation

- Beim Seminar gilt **Anwesenheitspflicht**:
Bei Abwesenheit bitte (per E-Mail) Bescheid geben
- **Technikcheck** rechtzeitig vor der Präsentation
- Veröffentlichung der (finalen) Folien und Ausarbeitung auf der Seminarwebseite
(Falls nicht gewünscht, bitte Bescheid geben)

Themen

Douglas R. Hofstadter: Gödel, Escher, Bach – ein Endloses Geflochtenes Band

*Das Betriebssystem ist selbst ein Programm mit der Aufgabe, die Maschine als solche gegen **unbefugten Zugang** von Benutzern **abzuschirmen** (und damit das System zu schützen) und auch, den Programmierer gegen die vielen äußerst **komplizierten und verworrenen Probleme** [...] **abzuschirmen**.*

*Wenn verschiedene Benutzer **gleichzeitig zur gleichen CPU** „sprechen“, dann ist das Betriebssystem das Programm, das die Aufmerksamkeit in **geregelter Weise von einem zum andern** lenkt.*

Andrew S. Tanenbaum: Modern Operating Systems

*[O]perating systems perform two essentially unrelated functions: **providing** application programmers [...] a clean **abstract set of resources** instead of the messy hardware ones and **managing** these **hardware resources**.*

*The job of the operating system is to **create good abstractions** and then implement and **manage** the abstract objects thus created.*

*[The operating system's] primary task is to keep **track** of which programs are using which resource, to **grant** resource requests, to **account** for usage, and to **mediate conflicting requests** from different programs and users.*

Betriebsmittel

- Umgang mit Nebenläufigkeitsproblemen
- Korrektheit, Effizienz, Vorhersagbarkeit von Synchronisationsmechanismen

Speicher

- Ausnutzen spezieller Speicher(arten)
- Berücksichtigung von Speicher/Daten im Scheduling

Zeit

- Einhalten zeitlicher Garantien, Termintreue
- Schutz durch zeitliche Isolation

Energie

- Energiegewahrheit: Bedarf und Versorgung
- Systemadaption, (Ausfall-)Schutz

Abstractions in General (*reserviert*)



Pekka Enberg, Ashwin Rao und Sasu Tarkoma. “I/O Is Faster Than the CPU: Let’s Partition Resources and Eliminate (Most) OS Abstractions”. In: *Proceedings of the Workshop on Hot Topics in Operating Systems*. 2019, S. 81–87

Concurrency Modelling



Catherine E Nemitz u.a. “Concurrency Groups: A New Way to Look at Real-Time Multiprocessor Lock Nesting”. In: *Proceedings of the 27th International Conference on Real-Time Networks and Systems*. 2019, S. 187–197

Deterministic Resource Access



Alexander Zuepke und Robert Kaiser. “Deterministic Futexes: Addressing WCET and Bounded Interference Concerns”. In: *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2019, S. 65–76

Scratchpad Awareness



Rohan Tabish u. a. “A Real-Time Scratchpad-centric OS for Multi-Core Embedded Systems”. In: *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2016, S. 1–11



Giovani Gracioli u. a. “Designing Mixed Criticality Applications on Modern Heterogeneous MPSoC Platforms”. In: *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019

Memory-Centric Scheduling



Rodolfo Pellizzoni u. a. “A Predictable Execution Model for COTS-Based Embedded Systems”. In: *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE. 2011, S. 269–279



Juan M Rivas u. a. “Implementation of Memory Centric Scheduling for COTS Multi-Core Real-Time Systems”. In: *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019

Data Awareness (*reserviert*)



Antonio Barbalace, Javier Picorel und Pramod Bhatotia.
“ExtOS: Data-centric Extensible OS”. In: *Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on Systems*.
2019, S. 31–39

Cache Awareness (*reserviert*)



Hendrik Borghorst und Olaf Spinczyk. “Increasing the Predictability of Modern COTS Hardware through Cache-Aware OS-Design”. In: *OSPERT 2015* (2015), S. 41



Hendrik Borghorst und Olaf Spinczyk. “CyPhOS—A Component-Based Cache-Aware Multi-Core Operating System”. In: *International Conference on Architecture of Computing Systems*. Springer. 2019, S. 171–182

Schedule as a Resource



Anna Lyons u. a. “Scheduling-Context Capabilities: A Principled, Light-Weight Operating-System Mechanism for Managing Time”. In: *Proceedings of the Thirteenth EuroSys Conference*. 2018, S. 1–16

Representing Timers



Pratyush Patel, Manohar Vanga und Björn B Brandenburg. “TimerShield: Protecting High-Priority Tasks from Low-Priority Timer Interference”. In: *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2017, S. 3–12

Scheduling Security (*reserviert*)





Chien-Ying Chen u. a. “A Novel Side-Channel in Real-Time Schedulers”. In: *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2019, S. 90–102



Man-Ki Yoon u. a. “TaskShuffler: A Schedule Randomization Protocol for Obfuscation against Timing Inference Attacks in Real-Time Systems”. In: *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2016, S. 1–12

Intermittent Execution

 Brandon Lucia und Benjamin Ransford. “A Simpler, Safer Programming and Execution Model for Intermittent Systems”. In: *ACM SIGPLAN Notices* 50.6 (2015), S. 575–585

 Andrea Maioli u. a. “On Intermittence Bugs in the Battery-Less Internet Of Things”. In: *Proceedings of the 20th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*. 2019, S. 203–207

Representing Power Consumption (*reserviert*)



Liwei Guo u. a. “Power Sandbox: Power Awareness Redefined”. In: *Proceedings of the Thirteenth EuroSys Conference*. 2018, S. 1–15

- Themen werden nach Windhundverfahren vergeben
- Eigene Themenvorschläge willkommen
- Vortragstermin prinzipiell selbst gewählt
 - eventuell gruppiert nach Themen
 - drei Vortragstermine am Semesterende
 - zwei bis drei Präsentationen pro Termin