

Seminare Wintersemester 2018/2019

Cyber-Physical Systems (Bachelor) Safety-Critical Systems (Master)

Rico Amslinger, Martin Frieb, Florian Haas, Christoph Kühbacher,
Christian Mellwig, Christian Piatka, Alexander Stegmeier,
Prof. Dr. Theo Ungerer



Lehrstuhl für Systemnahe Informatik
und Kommunikationssysteme
Institut für Informatik
Universität Augsburg

19. Oktober 2018

- ▶ Selbstständiges Arbeiten
 - ▶ Literaturrecherche
 - ▶ Einarbeiten in die Materie
- ▶ Ausarbeitung nach wissenschaftlichen Maßstäben
- ▶ Erarbeitung eines Vortrags zur Seminararbeit
- ▶ Vortrag und Diskussion der Themen im Blockseminar



- ▶ Bachelor
 - ▶ Ausarbeitung: 12–15 Seiten
 - ▶ Vortragsdauer: 25 Minuten, plus 5 Minuten Diskussion
- ▶ Master
 - ▶ Ausarbeitung: 15–20 Seiten
 - ▶ Vortragsdauer: 35 Minuten, plus 10 Minuten Diskussion
- ▶ \LaTeX -Vorlagen für Ausarbeitung und Vortragsfolien in Digicampus



I. Einarbeitung:

- ▶ Betreuer meldet sich per E-Mail für erstes Treffen
- ▶ Überblick über das Thema bekommen, Literaturrecherche



I. Einarbeitung:

- ▶ Betreuer meldet sich per E-Mail für erstes Treffen
- ▶ Überblick über das Thema bekommen, Literaturrecherche

II. Literaturrecherche:

- ▶ Literatur lesen, ordnen und zusammenfassen



I. Einarbeitung:

- ▶ Betreuer meldet sich per E-Mail für erstes Treffen
- ▶ Überblick über das Thema bekommen, Literaturrecherche

II. Literaturrecherche:

- ▶ Literatur lesen, ordnen und zusammenfassen

III. Einleitung & Gliederung:

- ▶ Einleitung und Motivation für Thema erstellen
- ▶ Struktur der Ausarbeitung festlegen
- ▶ Inhalte stichpunktartig definieren, mit Quellenangabe



IV. Ausarbeitung:

- ▶ Kapitel ausarbeiten
- ▶ Zusammenfassung und Fazit erstellen
- ▶ Vorgegebenen Umfang beachten
- ▶ Auf Sprache und Rechtschreibung achten



IV. Ausarbeitung:

- ▶ Kapitel ausarbeiten
- ▶ Zusammenfassung und Fazit erstellen
- ▶ Vorgegebenen Umfang beachten
- ▶ Auf Sprache und Rechtschreibung achten

V. Vortrag:

- ▶ Inhalt der Ausarbeitung verständlich und anschaulich zusammenfassen
- ▶ Vorgegebene Zeiten einhalten



- ▶ Zum jeweiligen Termin bis 08:00 Uhr
 - ▶ Aus \LaTeX erstellte PDFs
 - ▶ Per E-Mail an Betreuer und CC an `alexander.stegmeier@informatik.uni-augsburg.de`
 - ▶ Alle Termine sind einzuhalten!
-
- ▶ Zwischenstand bzw. Vorabversion von Ausarbeitung und Vortrag an Betreuer schicken (empfohlen)
 - ▶ Anmerkungen und Hinweise einarbeiten



- ▶ Themenvergabe: heute, 19.10.2018
- ▶ Termin mit Betreuer vereinbaren: Fr. 26.10.2018
- ▶ Literatur abgeben: Mo. 12.11.2018
- ▶ Gliederung & Einleitung abgeben: Mo. 26.11.2018
- ▶ Ausarbeitung abgeben: Mo. 14.01.2019
- ▶ Folien abgeben: Mi. 23.01.2019

- ▶ Blockseminar (Anwesenheitspflicht!):
 - Fr. 25.01.2019
 - Fr. 01.02.2019
 - Fr. 08.02.2019



Ursachen für Abbruch bzw. schlechte Ergebnisse sind meist:

- ▶ Probleme nicht rechtzeitig mit dem Betreuer besprochen
- ▶ Unterschätzung des Zeitaufwands:
 $4 \text{ LP} \Rightarrow 4 \cdot 30 \text{ h} = 120 \text{ Stunden Zeitaufwand}$
- ▶ Zu spät mit den einzelnen Abschnitten begonnen
- ▶ Zu wenig Rücksprache mit dem Betreuer

Lesen Sie die Vorlage für Seminararbeiten und verwenden Sie die bereitgestellten Vorlagen für Ihre Ausarbeitung und Präsentation!



- ▶ 9 Bachelorstudenten, 7 Bachelorthemen
- ▶ 11 Masterstudenten, 7 Masterthemen



- ▶ 9 Bachelorstudenten, 7 Bachelorthemen
- ▶ 11 Masterstudenten, 7 Masterthemen

Faire Verteilung der Themen:

- ▶ Jeder erstellt seine persönliche, priorisierte Themenliste
- ▶ Jede(r) Student(in) zieht eine Zahl
- ▶ Zuteilung nach absteigender Semesterzahl
- ▶ Student(inn)en wählen in aufsteigender Reihenfolge der gezogenen Zahl aus den noch freien Themen



1. Cyber-Physical Systems in der Medizin
2. Wear Core for Wearable Workloads
3. Timing-Analyse für parallele Echtzeit-Anwendungen mit P-SOCRATES
4. Visual SLAM / Visual Odometry
5. AI accelerators
6. Security Considerations for Cyber Physical Systems
7. Vergleich AUTOSAR und AUTOSAR Adaptive
8. Scheduling in Eingebetteten Systemen am Beispiel des Nvidia Jetson TX2
9. Verwendung von Nvidia CUDA in Echtzeit-Systemen



1. Systemweite Response-Time Analysis für Echtzeitsysteme
2. Edge-TM: Exploiting Transactional Memory for Error Tolerance and Energy Efficiency
3. Speicherkonsistenzmodelle für GPUs
4. Snapshot-Isolation Transactions
5. Echtzeitfähige Timer-Isolation im Linux-Kernel
6. Echtzeitfähige Cache-Kohärenz für Multicore-Prozessoren
7. Echtzeitfähige Hardware-Transaktionsspeicher

