Practical course

Advanced Systems Programming in C/Rust
(SoSe 2022)
Preliminary meeting

Chair of Decentralized Systems Engineering

https://dse.in.tum.de/
## About us

### Chair of Decentralized Systems Engineering

[https://dse.in.tum.de/](https://dse.in.tum.de/)

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redha Gouicem</td>
<td><a href="mailto:redha.gouicem@tum.de">redha.gouicem@tum.de</a></td>
</tr>
<tr>
<td>Atsushi Koshiba</td>
<td><a href="mailto:atsushi.koshiba@tum.de">atsushi.koshiba@tum.de</a></td>
</tr>
<tr>
<td>Peter Okelmann</td>
<td><a href="mailto:okelmann@in.tum.de">okelmann@in.tum.de</a></td>
</tr>
<tr>
<td>Myoung Jin Nam</td>
<td><a href="mailto:myoungjin.nam@tum.de">myoungjin.nam@tum.de</a></td>
</tr>
<tr>
<td>Sebastian Reimers</td>
<td><a href="mailto:sebastian.reimers@tum.de">sebastian.reimers@tum.de</a></td>
</tr>
</tbody>
</table>
Systems programming applications

Low level systems programming is an essential building block for high level applications
Software core properties

Performance

Reliability

Security

Efficient low level systems programming is critical to ensure these properties
System stack

System stack diagrams illustrate the relationship between different levels of software and hardware. Systems programming spans across multiple system levels and application domains. It involves the design, development, and maintenance of software that directly interacts with computer system hardware.

- **User** interacts with applications.
- **Application** runs on the operating system.
- **Operating system** manages hardware resources.
- **Hardware** includes components like processors, memory, and storage.

Cloud services represent the integration of software and hardware in a cloud computing environment, offering scalable resources on demand. Time to get hands-on experience!
Course topics

• This course covers some of the most important aspects of systems programming:

Kernel and system calls

File I/O

Concurrency and synchronization

Memory management
Course topics

• This course covers some of the most important aspects of systems programming:

  - Processes
  - Networking
  - Performance profiling
  - Virtualisation (containers/KVM)
Lab format

• Lab assignments
  • 8 practical programming exercises
  • Deadline of 2 - 3 weeks depending on the difficulty/workload
  • Online submission

• Weekly meeting
  • Video with theoretical background coverage
  • Question and answer session to explain and discuss each assignment
  • Slack channel for questions

• Assessment:
  • 8 programming assignments (100%) with public & private unit tests
  • No further exam / quiz / projects
Grading system

- Github classroom (https://classroom.github.com/)
  - Template repository for each task with detailed instructions & test cases

- Automated tests
  - Points are distributed among the exercises based on the estimated workload (30-60 points)
  - Specially designed test cases with gradually increasing difficulty
  - Hidden tests to detect & prevent gaming the grading system

- Grading scheme:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Grade</th>
<th>From</th>
<th>To</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>119</td>
<td>5.0</td>
<td>210</td>
<td>224</td>
<td>2.7</td>
</tr>
<tr>
<td>120</td>
<td>134</td>
<td>4.7</td>
<td>225</td>
<td>239</td>
<td>2.3</td>
</tr>
<tr>
<td>135</td>
<td>149</td>
<td>4.3</td>
<td>240</td>
<td>254</td>
<td>2.0</td>
</tr>
<tr>
<td>150</td>
<td>164</td>
<td>4.0</td>
<td>255</td>
<td>269</td>
<td>1.7</td>
</tr>
<tr>
<td>165</td>
<td>179</td>
<td>3.7</td>
<td>270</td>
<td>284</td>
<td>1.3</td>
</tr>
<tr>
<td>180</td>
<td>194</td>
<td>3.3</td>
<td>285</td>
<td>300</td>
<td>1.0</td>
</tr>
<tr>
<td>195</td>
<td>209</td>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Languages / OS

• Languages
  • Choice between C, C++ and Rust
  • Can be switched for each task
  • Limited choice of allowed libraries (different per language)

• OS Environment information
  • All executables must run on Linux, x86_64
  • Use virtual machines if you run a different OS (i.e. Hyper-V on Windows)
Assignment format (examples)

• Given a code skeleton:
  • Implement functionalities of a filesystem
  • Conversion of a program to a multithreaded version using locks appropriately
  • Write your own memory allocator
  • Implement your own client/server applications

• Given a complete implementation:
  • Do performance profiling
  • Identify bottlenecks & implement performance optimizations
Learning goals

• Acquire fundamental knowledge to build robust systems
• Familiarize yourself with end-to-end system design
• Learn techniques for profiling, debugging and optimization of low-level code
• Get a good understanding of memory- and resource management
• Improve hands-on experience through a variety of programming tasks

• Importantly, have fun!
Prerequisites

- Knowledge equivalent to the lectures
  - Fundamentals of Programming (IN0002)
  - Introduction to Computer Architecture (IN0004)
  - Basic Principles: Operating Systems and System Software (IN0009)

- Programming knowledge
  - Foundations of programming (C, C++ or Rust)
  - Work in a Linux environment

If the prerequisites are unclear/strict -- please check with the instructor!
Code of conduct

• University plagiarism policy

• Decorum
  • Promote freedom of thoughts and open exchange of ideas
  • Cultivate dignity, understanding and mutual respect, and embrace diversity
  • Racism and bullying will not be tolerated
Interested?

Matching platform

Welcome to the Matching platform matching.in.tum.de!

Dear students,

we changed the name of the course "Seminar: Recent advances in Computer Systems", for consistency reasons. The new name are "Seminar: Hot Topics in Computer Systems", now.

Login with your TUM identifier. Login for exchange students (without TUM identifier) Any questions? Visit the FAQs!

Sign up on the TUM matching platform

Fill our survey form (Link available in TUM online)
Contacts

- Redha Gouicem
  - redha.gouicem@tum.de

Workspace: http://ls1-courses-tum.slack.com/
Website: https://dse.in.tum.de/
Channel: #ss-22-sys-prog

Join us with TUM email address (@tum.de)