Systems Programming in C++

Practical Course

Summer Term 2019

Course Goals

Learn to write good C++

- Basic syntax
- Common idioms and best practices

Learn to implement large systems with C++

- C++ standard library and Linux ecosystem
- Tools and techniques (building, debugging, etc.)

Learn to write high-performance code with C++

- Multithreading and synchronization
- Performance pitfalls

Formal Prerequisites

Knowledge equivalent to the lectures

- Introduction to Informatics 1 (IN0001)
- Fundamentals of Programming (IN0002)
- Fundamentals of Algorithms and Data Structures (IN0007)

Additional formal prerequisites (B.Sc. Informatics)

- Introduction to Computer Architecture (IN0004)
- Basic Principles: Operating Systems and System Software (IN0009)

Additional formal prerequisites (B.Sc. Games Engineering)

- Operating Systems and Hardware oriented Programming for Games (IN0034)

Practical Prerequisites

Practical prerequisites

- **No previous experience with C or C++ required**
- Familiarity with another general-purpose programming language

Operating System

- Working Linux operating system (e.g. Ubuntu)
- Basic experience with Linux (in particular with shell)
- You are free to use your favorite OS, **we only support Linux**

Lecture & Tutorial

- Lecture: **Tuesday, 14:00 – 16:00, MI 02.11.018**
- Tutorial: **Friday, 10:00 – 12:00, MI 02.11.018**
  - Discuss assignments and any questions
  - First **two** tutorials are additional lectures
- Everything will be in English
- **Attendance is mandatory**
- Announcements on the website

Assignments

- Brief non-coding quizzes in (random) lectures or tutorials
- Weekly programming assignments
  - No teams
  - Managed through our GitLab (more details in first tutorial)
- Final project at end of the semester
  - No teams
  - Managed through our GitLab (more details in first tutorial)
  - More extensive than assignments (several weeks of work)
  - Implementation from scratch (including infrastructure)
  - Lecture will prepare for the project

Topics

Very rough overview of topics

- C++ syntax and language features
- Common C++ programming techniques
- Proper usage of the C++ standard library
- Low-level (performance) considerations
- The C++ ecosystem (building, testing, debugging, profiling, ...)
- Keeping control of large projects

Literature

Primary

- Lippman, 2013. *C++ Primer (5th edition)*. Only covers C++11.
- Stroustrup, 2013. *The C++ Programming Language (4th edition)*. Only covers C++11.
- Meyers, 2015. *Effective Modern C++. 42 specific ways to improve your use of C++11 and C++14.*.

Supplementary

- Aho, Lam, Sethi & Ullman, 2007. *Compilers. Principles, Techniques & Tools (2nd edition)*.
- Tanenbaum, 2006. *Structured Computer Organization (5th edition)*.

Contact

Important links

- Website: http://db.in.tum.de/teaching/ss19/c++praktikum
- E-Mail: freitagm@in.tum.de, sichert@in.tum.de

Register for the course through the matching platform (https://matching.in.tum.de/)