INF113: Recap and Q&A

Kirill Simonov

12.11.2025

Final exam

- Takes place November 21, 15:00, Solheimsgaten 18
- Exam score is 70% of the final grade
- No prerequisites for taking the exam
- Consists of multiple-choice and text answer questions
  – Will also include light-weight programming tasks with auto-tests
- Questions will be in English, Bokmål and Nynorsk
- No support materials on the exam
- All topics discussed on the lectures may appear in the exam

Exam preparation

- Past exams published in the "Exam preparation" module on Mitt
- Revisit lecture slides or lecture recordings, published on Mitt
- Revisit book chapters, see the list on the course homepage

Tools

- C programming language
  - Basic syntax
  - Using common standard functions such as `printf`
- Assembly
  - Concept of basic operations and registers, such as `mov`, `syscall`
  - How system calls are invoked from assembly
  - Not expected to memorize instruction sets, operation codes, argument combinations...
- Bash and command-line tools
  - Standard utilities such as `cd`, `mv`, `cat`
  - Diagnostics tools like `valgrind`, `strace`

Processes

- Machine instructions and CPU cycle
- Compiling vs linking vs running
- System calls and context switches
- Time-sharing and multiple processes
- fork/exec/wait
- Scheduling
  - Shortest-to-completion-first and turnaround time metric
  - Round-robin and response time metric
  - Multi-level feedback queue (MLFQ)

Memory

- Static memory, stack and heap memory as used in C programs
  - Memory bugs, `valgrind`
- Concept of virtual memory/virtual address space of an individual process, address translation
  - Base and bounds and hardware registers
  - Segmentation
- Free space management
- Paging and page tables
  - Address translation cache a.k.a. TLB
  - Multi-level page tables

Concurrency

- Concept of a thread and issues arising from sharing the same memory
- Mutexes that protect critical sections, spinning lock designs
- Conditional variables a.k.a. putting threads to sleep in a queue
- Semaphores, their uses, implementing with locks and conditional variables
- Concurrent applications: parent-child join, producer/consumer
- Concurrent issues: race conditions, deadlocks

Storage

- Hardware view: devices, drivers, HDDs, SSDs
- Files and directories, system calls that access/modify them
- Very simple file system: inodes, allocation bitmaps, superblock; FFS: block groups
- Crash consistency and `fsck`
- Journaling, data vs metadata journaling

Q&A

- Any questions about course topics and what may appear on the exam
- Questions about Assignment 3