

INF-2201 Operating System Fundamentals

Introduction

John Markus Bjørndalen, 2024

Topics

- Course structure / plan
- Plagiarism and other cheating
 - How to learn / what does approaches teach you?
- Introduction to booting and running code on a computer.

What you will be doing

- Build an operating system kernel
 - (partly) from scratch
 - We provide you with pre-code for each project, so you should not try to solve all problems (you will not have the time to get them done correctly)
- 6 projects
 - P1: Boot mechanism
 - P2: Non-preemptive Scheduling
 - P3: Preemptive Scheduling
 - P4: Interprocess Communication (IPC) and Process Management
 - P5: Virtual Memory
 - P6: File System



Skypellling:
"A busy beaver writing an operating system on an old computer"
Made with Image Creator from Designer. Powered by DALL-E 3.

What you will be doing

There will be a lot of work, but you gain something:

- Better insight into how an operating system works than people that just read about it
- Building abstractions:
 - How to design an abstraction
 - How to implement the abstraction
- You will be better prepared for other courses at Ifi

Major changes this year

2024:

- No home exams
- Oral exam at the end. The plan is:
 - Groups of 2 students
 - Up to 1 hour per group
 - Mainly focused on the projects!
- Projects
 - P1 is an optional tutorial (1 week)
 - P2-P5 are mandatory (a pass is required to take the exam)
 - Groups of 2 (limit due to exam)
 - Approx 3 weeks for each
 - Design reviews approx 1 week into the project
 - Last mandatory hand-in end of April
 - P6 is optional
- Updated projects
 - Work in progress
 - Moving away from older architectures that are not easily available
 - Simplification and more consistency

2023:

- P1, P3, P5 mandatory projects
- P2, P4, P6 home exams
- Last hand-in end of May

Major changes in the future

- 10 credit course
 - Maybe followed by a 10 credit advanced operating systems and concurrency course
- P1 (boot mechanism) will be drastically changed
- Probably ARM (Raspberry Pi) or UEFI boot on PC
- Project 2-6 restructured (and some simplifications)

Course plan

Two ways to view it (both links are in Canvas)

- <https://github.com/uit-inf-2201-s24/uit-inf-2201-s24.github.io/blob/master/index.md>
- <https://uit-inf-2201-s24.github.io/>
-

We need feedback

- Both what you like and what could be improved
- The good parts help us see what we need to keep

Cheating



Skypellling:
"A criminal raccoon running away with a stolen operating systems book"
Made with Image Creator from Designer. Powered by DALL-E 3.

Approaches to solving a problem

Approach	Risk Especially plagiarism	What did you learn?
Ask friend for solution	Copying content Not learning problem solving	Copying and adapting solutions
Ask Google	Copying content Not learning problem solving	Copying and adapting solutions
Ask ChatGPT / AI tool	Copying content (without knowing) Not learning problem solving	Copy problem text and adapt prompts Copying and adapting solutions
Try a simple (brute force?) solution first, then above tools	Fewer if you clearly reference what you copied.	Problem solving + above

Plagiarism – generic advice (WIP)

- Always check your references
 - Can you find the paper the way you cited/referenced it?
 - Does it say what you claim in your text?
 - Did you copy any of the text without making it clear it's a quote? (The same goes for figures, tables, source code...)
 - There are a lot of incorrect references out there, and lazy people copy other people's references without checking them.
- The "acceptable" limit may not be the same everywhere!
 - Problem: what used to be acceptable or not focused on may be an issue in the future.

Plagiarism (source code)

- If you borrow a library, function etc from somewhere
 - At least add a comment in the code stating where you copied it from
 - The comment could also be "this is based on X"
 -

Resources

- Canvas
- <https://github.com/uit-inf-2201-s24>
 - Need GitHub user names to add to org.
- Discord – invite is in Canvas

Tomorrow (Wednesday)

- P1 as a tutorial
 - Get you started with the source code and the development environment (compiler, emulator, ...)
 - Familiarize yourself with how the operating system boots
- Bring laptops if you can!
 - Getting things up and running now will save you time later