INF-2203 Operating System Fundamentals

Introduction

John Markus Bjørndalen, 2026

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 don't have to solve all the problems a kernel needs to solve
- 4 projects
  - P1: Booting OS and starting programs
  - P2: Protecting the kernel
  - P3: Multithreading and concurrency
  - P4: Multiple processes and IPC

Skype/Bing:
"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
  - You will be the guys that actually understand how an OS works
- 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

Main changes
- The course is now **10 credits** (from 20 credits in inf-2201)
- Updated projects
  - Completely rewritten projects
  - Work in progress
  - Moving away from older architectures that are not easily available
  - Simplification and more consistency

We keep
- The same textbook
- Oral exam at the end, mainly focused on the projects!
  - Spending time learning will pay off
  - Spending as little time as possible to get a "pass" will not

---

Course plan

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

- https://github.com/uit-inf-2203/uit-inf-2203.github.io/blob/master/index.md
- https://uit-inf-2203.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

Skype/Bing:
"A criminal racoon 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	Copying and adapting solutions Not learning problem solving Did you understand anything?
Ask Google	Copying content	Same as above
Ask ChatGPT / AI tool	Copying content (without knowing)	Same as above + Copy problem text and adapt prompts
Try a simple (brute force?) solution first, then above approaches. Compare solutions	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-2203
  - Need GitHub user names to add to org.
- Discord – invite is in Canvas
-