

The Course

An overview of IN2140

Tuesday, January 21, 2025

People

- Lecturers
  - Carsten Griwodz
    griff @ ifi
    
  - Pål Halvorsen
    paalh @ ifi
    
  - Özgü Alay
    ozgua@ifi
    
  - Kristoffer Robin Stokke
    krisrst@ifi
    
  - Jie Bian
    jiebi @ ifi
    
- Group teachers
  - Philip Høeg Eriksen, philipe
  - Jasuhan Yoganathan, jasuhany
  - Sondre Sundby Rosslund, sondrsro
  - Ka Thas, kavint
  - Lars Nøvik, larsnov
  - Birk Bergerud-Nygård, birkb

Administrative questions by email

Learning Outcome



- Course content
  - an introduction to **operating systems** (OS),
    viewed from both the user's and machine's point of view
    - have basic knowledge of the most important components of an operating system
    - be able to develop small applications that call OS services
    - know algorithms, don't program the operating system kernel (that is IN3000)
  - insight into how **data communication** is taking place
    - be able to choose between algorithms that enable today's data communication in the Internet
    - be able to develop programs with processes that communicate – within the computer and over a network
  - understanding theory through practice by **programming**
    - experience implementing algorithms for operating systems and data communication without the protective cover provided by runtimes and interpreters
    - experience programming in C

Learning Outcome: OS



- have basic knowledge of the most important components of an operating system
- be able to develop small applications that call OS services
- know algorithms

- Why?
  - Know that operating systems hide hardware complexity for you
  - Know about kernel space and user space, and how they interact
  - Learn typical algorithms for the most important operations performed by the operating system for you
  - Understand that different algorithms are good for different workloads
  - Understand when software is wasting performance unnecessarily
  - Select the right hardware, services and algorithms when you build complex systems
  - Need-to-know for advanced courses in operating systems, compilers, distributed systems, parallel programming, HPC, ...

Learning Outcome: Datacom



- be able to choose between algorithms that enable today's data communication in the Internet
- be able to develop programs with processes that communicate – within the computer and over a network

- Why?
  - Know how data and information are moved over a network
  - Understand how networks are structured
  - Know those parts of the network that you can change yourself

  - Write simple networked and distributed applications
  - Make the right choices when programming networked and distributed applications
  - Understand and change basic network configurations

  - Understand the cost of distributing applications
  - Understand the performance of complex distributed applications
  - Need-to-know for advanced courses in networks, distributed systems, parallel programming, HPC, security, ...

Learning Outcome: C



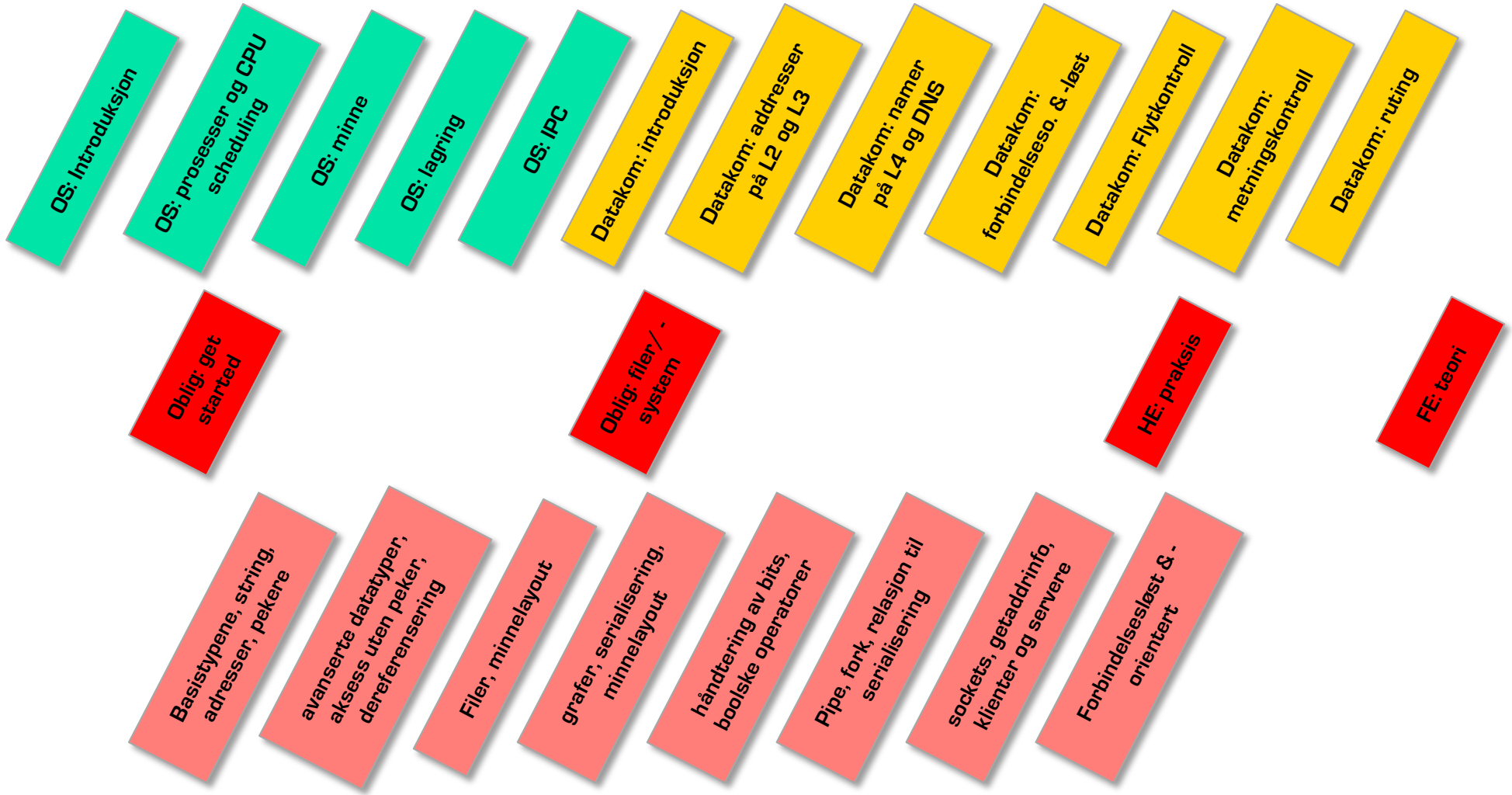
- experience implementing algorithms for operating systems and data communication without the protective cover provided by runtimes and interpreters
- experience programming in C

- Why?
  - Understand memory and memory addresses in much more detail than from theory
  - Understand how operating systems are related to computer architecture
  - Understand that networks do not know about data structures and program structures

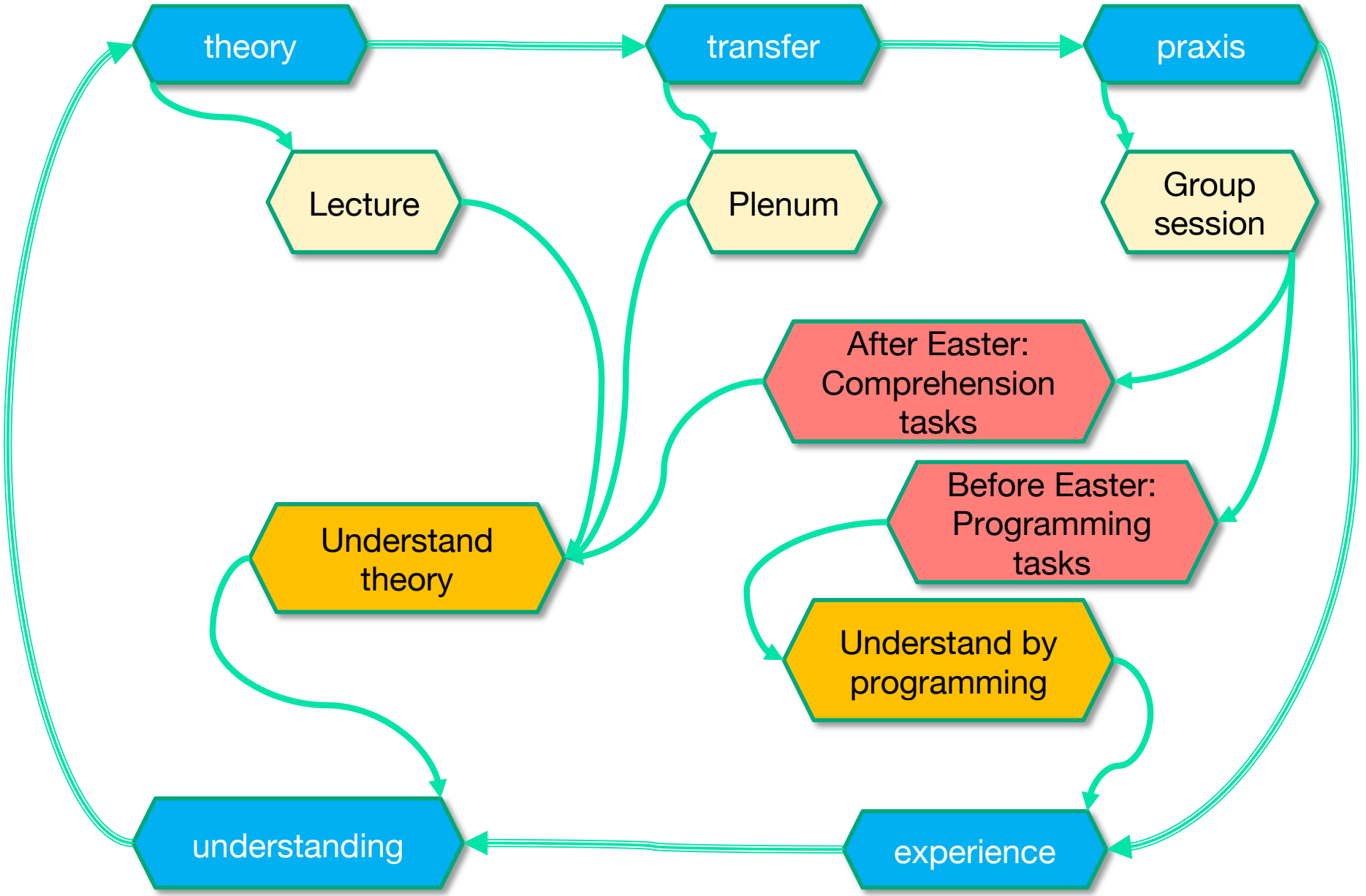
  - C brings you very close to the computer's architecture and OS
  - Most importantly
    - C allows you to easily **interpret every** (accessible and readable) **byte** in the computer **as you like**
    - Most programming languages make that very hard or impossible
    - You cannot write an OS without re-interpreting memory

  - C is the (grand-) parent of C++, Objective C, Java, C#, Scala, Python, PHP, Perl, Ruby, ...
  - C is not the final word in OS writing, but so far, all others failed. Will **Rust** succeed?

Connecting the dots



Teaching structure



Teaching structure

Theory

- Input
  - Weekly lectures
  - Weekly slides
  - Tanenbaum's books



From theory to practice

- Input
  - **Live** weekly, unrecorded sessions
  - From theory to code

Practice

- Input
  - C programming education
  - Assignment help
  - Weekly exercises



Teaching structure

Theory

Voluntary

Attend «lecture» session

Ask theory questions

Examination

4-hour exam,
mostly multiple choice

Practice

Voluntary

Attend «group» session

Solve programming tasks

Ask for help

Mandatory

2x semester mandatory programming exercise

deliver in groups

Examination

3-week home exam,
solving a programming task

deliver in groups

From theory to practice

Voluntary

Attend «plenum» session

Get a practical look at theory

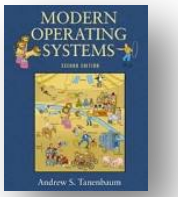
Ask theory & programming questions

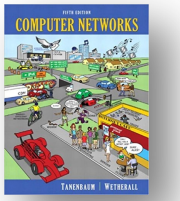
Pensum



- Topics taught in the videos, slides and live sessions together with the given slides and exercises constitute the “pensum”

- Supporting books
  - Modern Operating Systems:
    Andrew S. Tanenbaum
  - Computer Networks:
    Andrew S. Tanenbaum
  - C Notes for Professionals book:
    editor: web@petercv.com
  - Other useful C books:
    - The C programming Language:
      Brian Kernighan and Dennis Ritchie
    - C: A Reference Manual:
      Samuel P. Harbison and Guy L. Steele

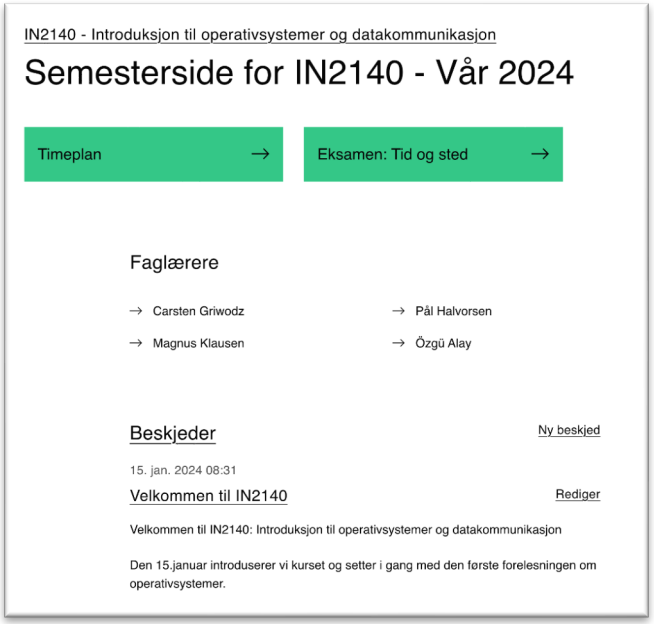






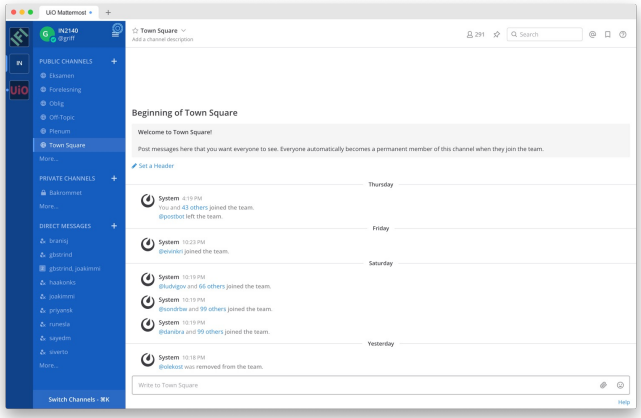
Information

- Course homepage in Vortex
  - your starting point
  - messages & news
  - schedule
  - videos in Panopto
  - Mattermost for discussions



- Devilry
  - for submitting your responses to mandatory assignment

- Inspera
  - for both exams



Exams and Mandatory Exercises

- Mandatory exercises
  - two C programming exercises
    - getting started with C
    - preparation for home exam
  - tentative dates: delivery 11.02. + 11.03.
- One home exam
  - focus on programming and documenting one concept related to OS and DataCom
  - tentative dates: delivery 29.04.
- Final exam
  - focus on the concepts, not programming
  - date 30.05.
  - 4 hours, 9:00-13:00



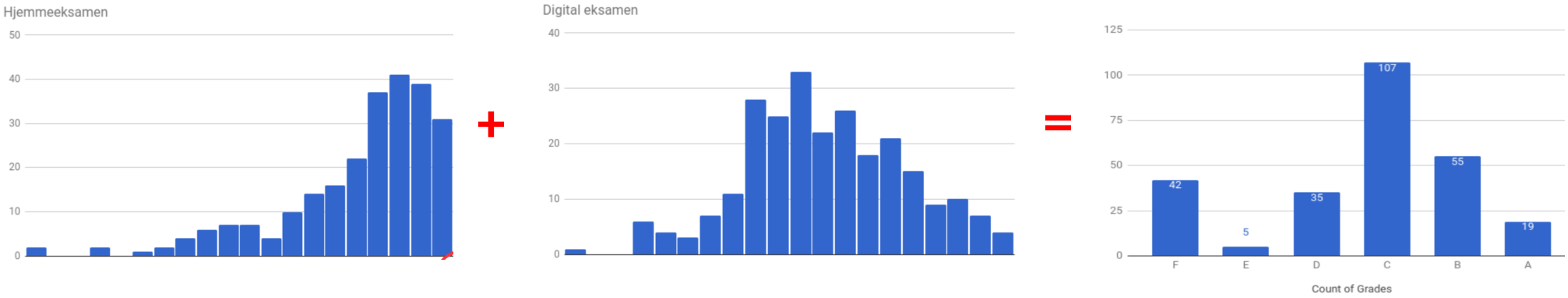
Note:
spring 2025 exams are similar to spring 2024 exams

Evaluation and grading

- Passing the mandatory assignments is a prerequisite for admission to the **theory** exam
  - Passing mandatory assignments in early years qualifies for taking the exams.
  - Both exams must be taken in the same semester.

- Grading:
  - 40% homeexam
  - 60% final exam

- Both exams count



programming is fun, but counts only for 40% of the final grade

Questions?

