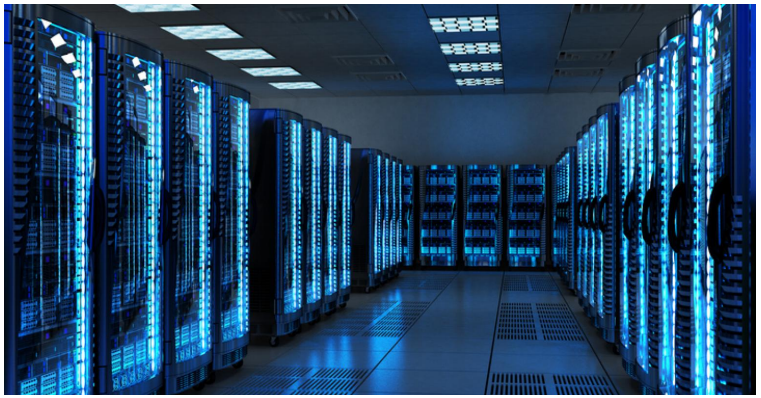
Why EE415?

Why Take This Course?

- Computers are everywhere



In datacenter

In your pocket

In your microwave

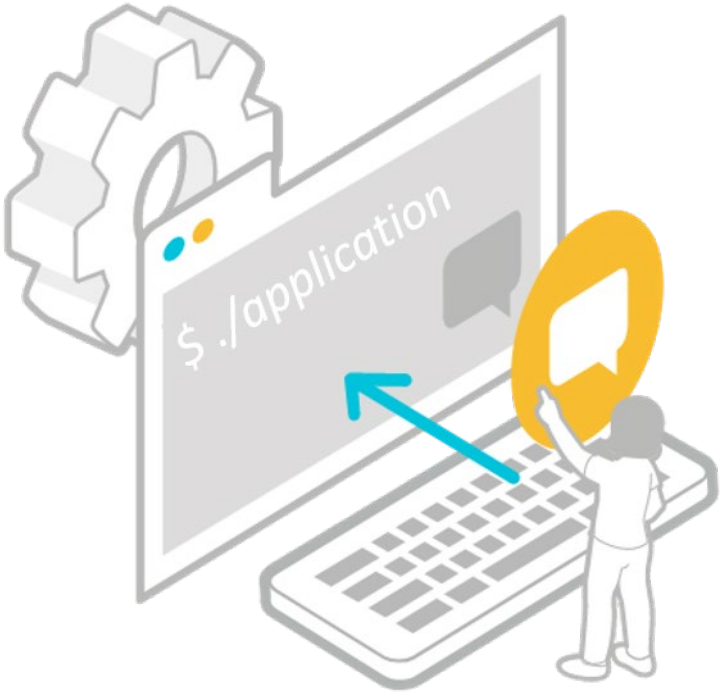
Up in space

Why Take This Course?

- Computers are everywhere
- We take hardware and OS features for granted

Why Take This Course?

- Computers are everywhere
- We take hardware and OS features for granted
- … but very few people truly understand how computers really work at a low-level



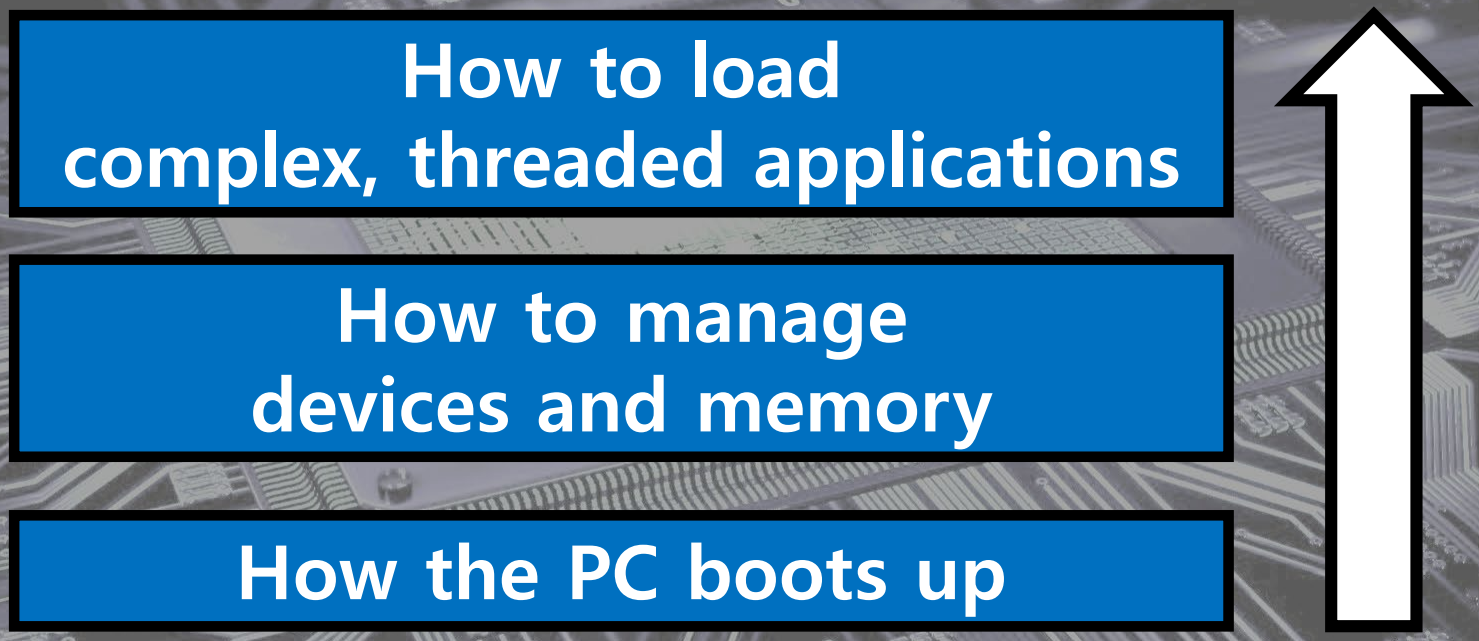
Why Take This Course?

- Computers are everywhere
- We take hardware and OS features for granted
- … but very few people truly understand how computers really work at a low-level

So, the first goal of this course is…

Goal1: Why Hardware is The Matter?

Fundamental understanding about the connection of computer hardware and OS



Goal 2: Why Software is The Matter?

Deeper Understanding about Operating System
In the aspect of "Software"

- Not hardware
- No theory

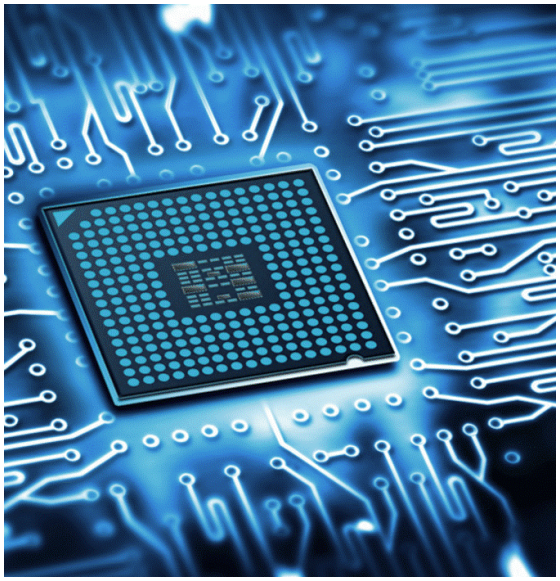
Goal 3: Learn From Experience

Apply What You Learned from Lectures to Projects

- You will build a simple OS in this class
- This will be a **huge** amount of work
- However, you will also learn a **huge** amount (don't worry, not that serious though)

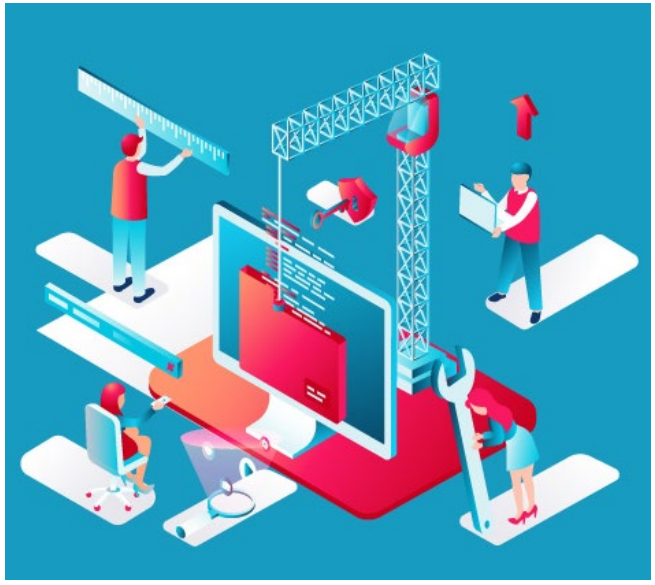
At The End of This Course…

- You will be able to understand



Low-level details of computer hardware and modern CPUs

Key functions of OSes

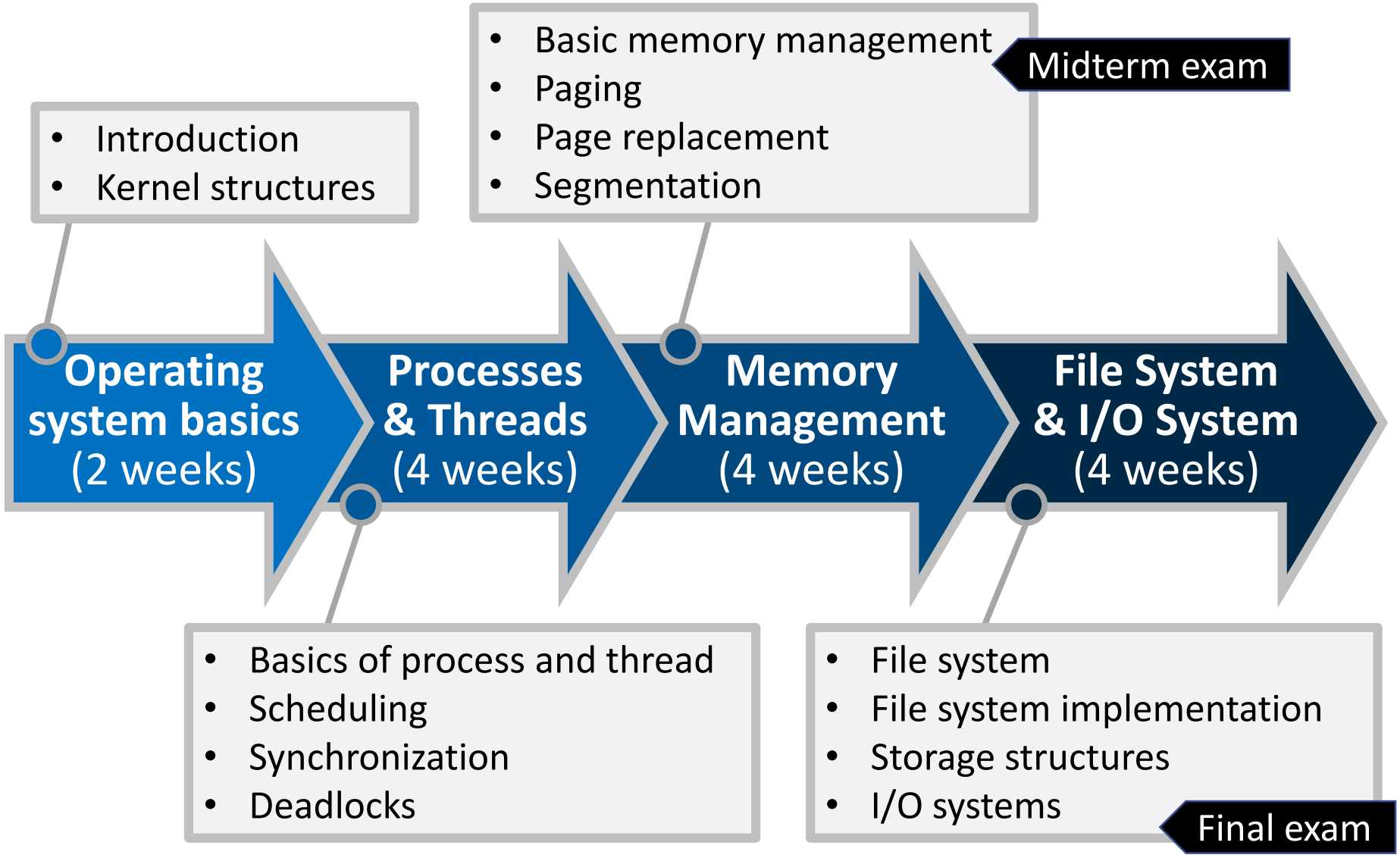


The fact that designing systems is an art, not a science

Contents

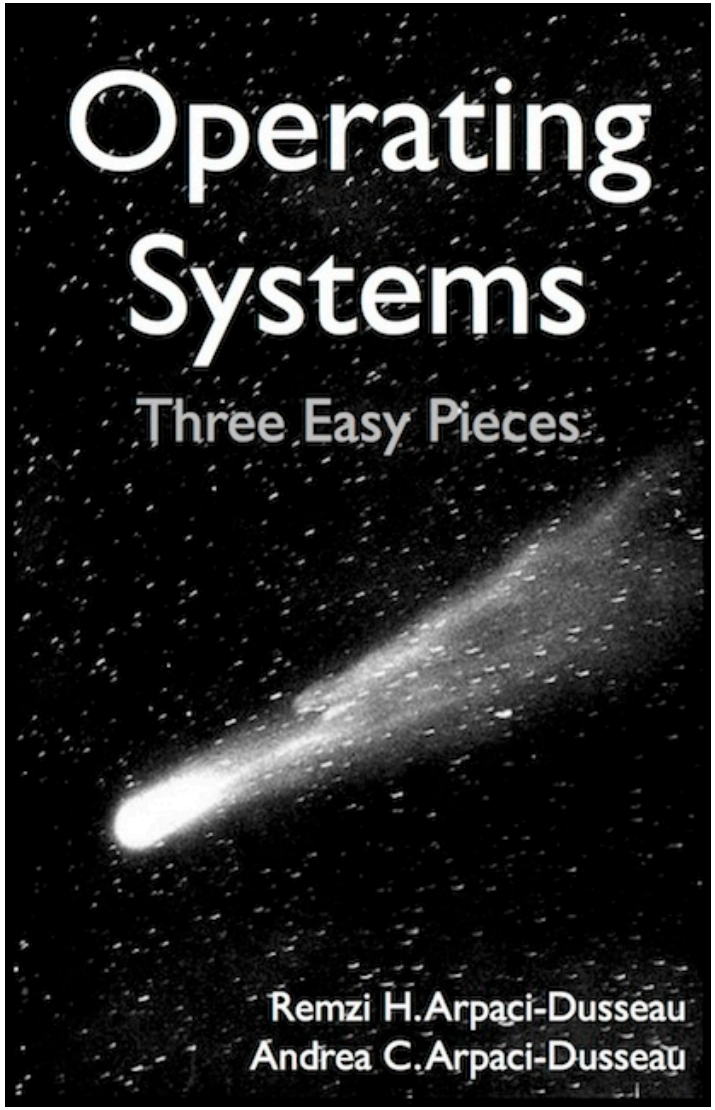
What will we learn to achieve the goals previously mentioned?

Topics in This Course

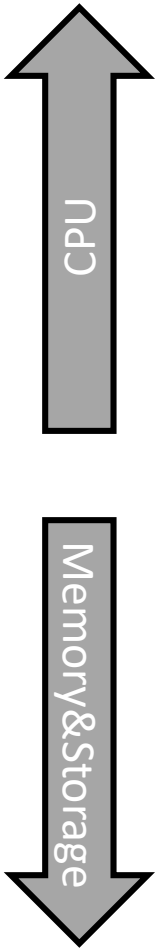


Textbook

- Operating Systems: Three Easy Pieces
  - Remzi and Andrea Arpaci-Dusseau
  - Very easy to go over (well-written)
- Free, PDFs available online at http://pages.cs.wisc.edu/~remzi/OSTEP/



Lecture Plan



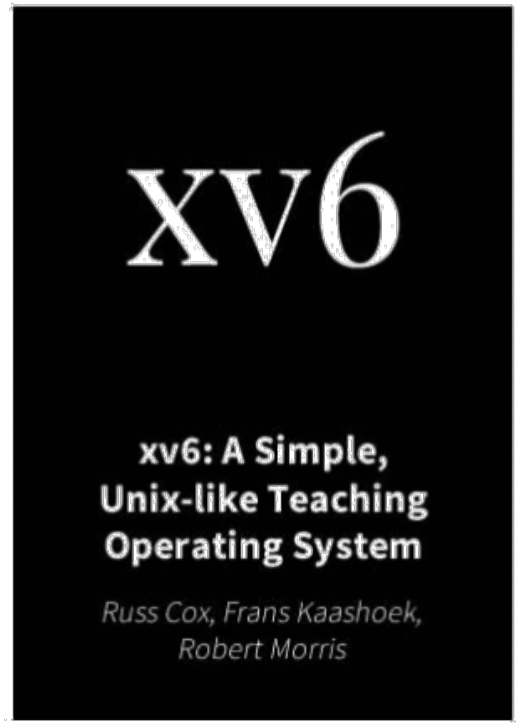
Period	Topics
Week 1	Introduction
Week 2	Kernel structures
Week 3	Processes and Threads
Week 4	Scheduling
Week 5	Synchronization
Week 6	Deadlocks
Week 7	Basic Memory Management
Week 8	**Mid-term exam**
Week 9	Paging
Week 10	Page Replacement
Week 11	Segmentation
Week 12	File System
Week 13	File System Implementation
Week 14	Storage Structures
Week 15	I/O Systems
Week 16	**Final exam**

Projects

- This course is **PROJECT-centric**
  - You will build an operating system. If you are not familiar with programming, what we recommend is to
    - start early!
    - seriously, start early!
- Projects
  - Due at 11:59:59pm on specified date
  - Use **KLMS** to submit your projects
  - If you do not have an appropriate programming environment, use **EE415 server (Hae-dong lounge)** for your projects

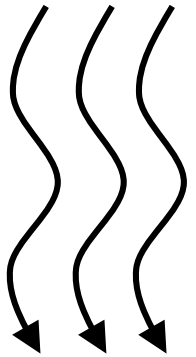
xv6 Based Projects

- **xv6**
  - Built over ANSI C and designed towards Intel x86 Sixth Edition Unix (V6)
  - A teaching OS for MIT OS course.
  - https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev10.pdf
- **Project 0: Introduction to xv6**
  - You will learn how to use xv6
  - You need adding simple system calls to xv6

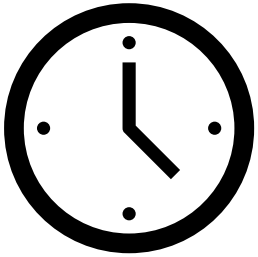


xv6 Based Projects

- **Project 1: kernel threading**
  - Create kernel thread
    - `clone()` and `join()`
  - Add user-level thread library
    - `thread_create()`
    - `lock_acquire()` and `lock_release()`

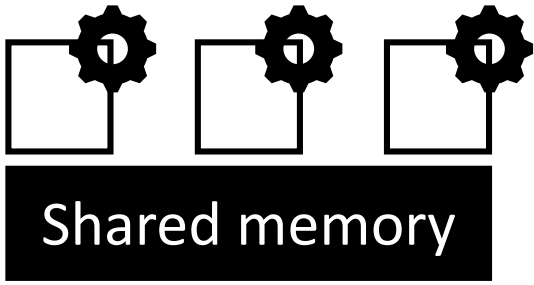


- **Project 2: process scheduler**
  - Create priority-based scheduler



xv6 Based Projects

- **Project 3: nullptr dereferences and shared memory page**
  - Add nullptr dereferences
    - Kill process when dereferences null pointer.
  - Add shared memory
    - System call for assign shared memory
    - System call for how many processes are sharing memory



- **Project 4: filesystem optimization**
  - Optimize filesystem for small-sized files