CS333: Storage Systems

Course Trajectory & Themes

Course Description

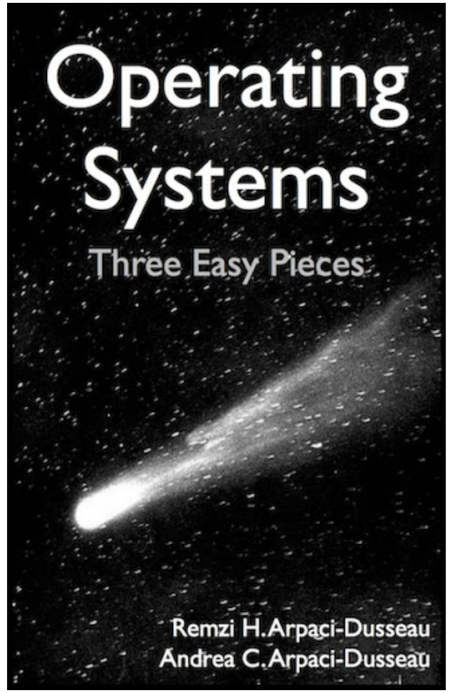
This course will examine topics in the **design**, **implementation**, and **evaluation** of storage systems. Topics include:

- the memory hierarchy;
- ways that data is organized (both logically and physically);
- storage hardware and its influence on storage software designs;
- data structures;
- performance models; and
- system measurement/evaluation.

Our readings and discussions will place an emphasis on identifying and evaluating **design trade-offs**.

Course Trajectory

- First half: Operating Systems: Three Easy Pieces (v1.10)
  - Build foundation from hardware (devices) up through applications
  - Available for free; most up-to-date PDFs linked on webpage



- Second half: Current/influential research papers
  - Papers posted on schedule page like rest of readings
    - Chosen with a trajectory in mind, but there is some flexibility if you are excited about a certain topic— let me know!
    - **Optional** readings often posted for diving deeper—often the main reading is chosen because it is clearly written or influential, not necessarily most recent

Course Structure

- Depending on the topic, lectures may be significantly shorter than 1h15m
  - Daily handout with notes and discussion questions
  - Some days we will use interactive demos to explore topic
  - Sync up at end to go over outstanding questions

- I will try to move class to a room that is better for small discussions, ideally with access to Unix computers (hopefully library's "fish bowl" room across the hall)

Course Components

- **Class meeting preparation and participation** (15%)

- **Warm-up question** during many (all?) class meetings
  - Build on/extend key ideas from readings
  - Short question/answer, sometimes without a single "best" answer
    - Goal is to *apply* concepts from the reading — not memorization
    - Goal is to "page in state" for our conversation on the day's topic
    - Graded on ✓ / ✓+ / ✓- basis
- Also consider **attendance** and **participation**

Course Components

- **Laboratory assignments** (35%)
  - System calls and C: Implement key Unix utilities
  - Environment Setup: Install and Customize Linux Environment
  - "Hello Fuse": Pseudo-file to explore FUSE infrastructure
  - File system design: FUSE Reference FS
    - Similar to textbook presentation: a simplified version of default Linux FS

Course Components

- **Midterm Exam** (25%)
  - One midterm exam on course's textbook-focused content
  - Will have both take-home and in-class components
    - With goal of making less stressful for you all
  - Date is posted on course schedule page

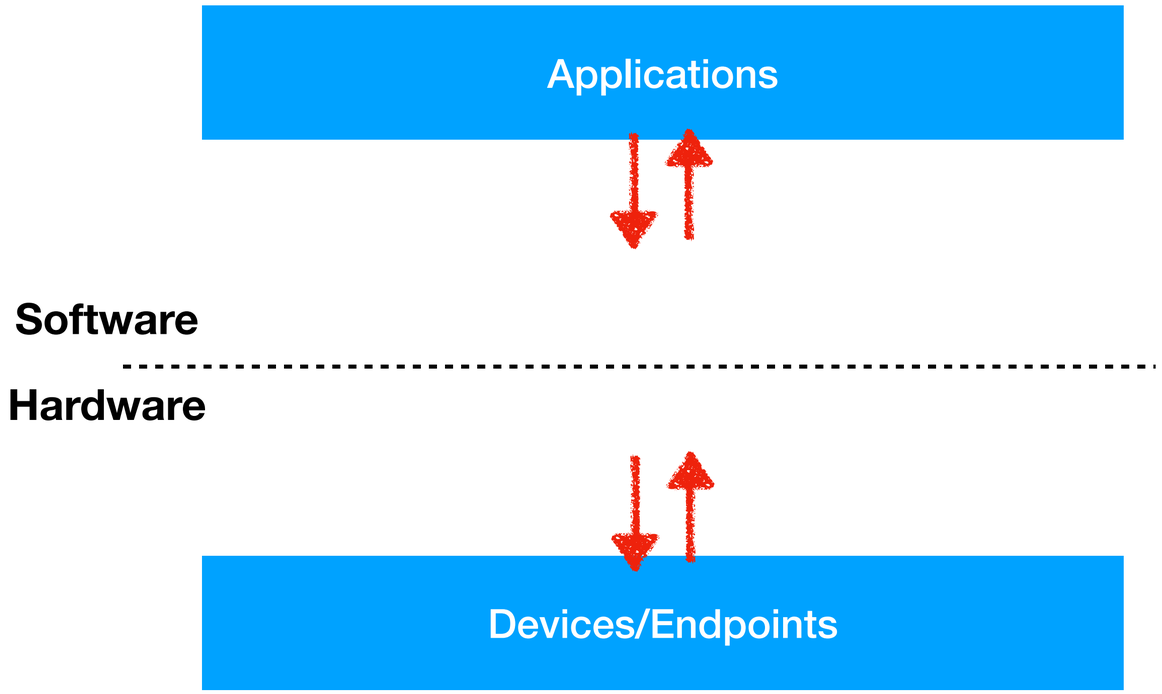
Course Components

- **Final Project** (25%)
  - Open-ended
    - You will propose a project;
    - We will narrow its scope and define achievable deliverables/evaluation criteria
    - Teams will present their findings during the last day of class

Overview and Themes

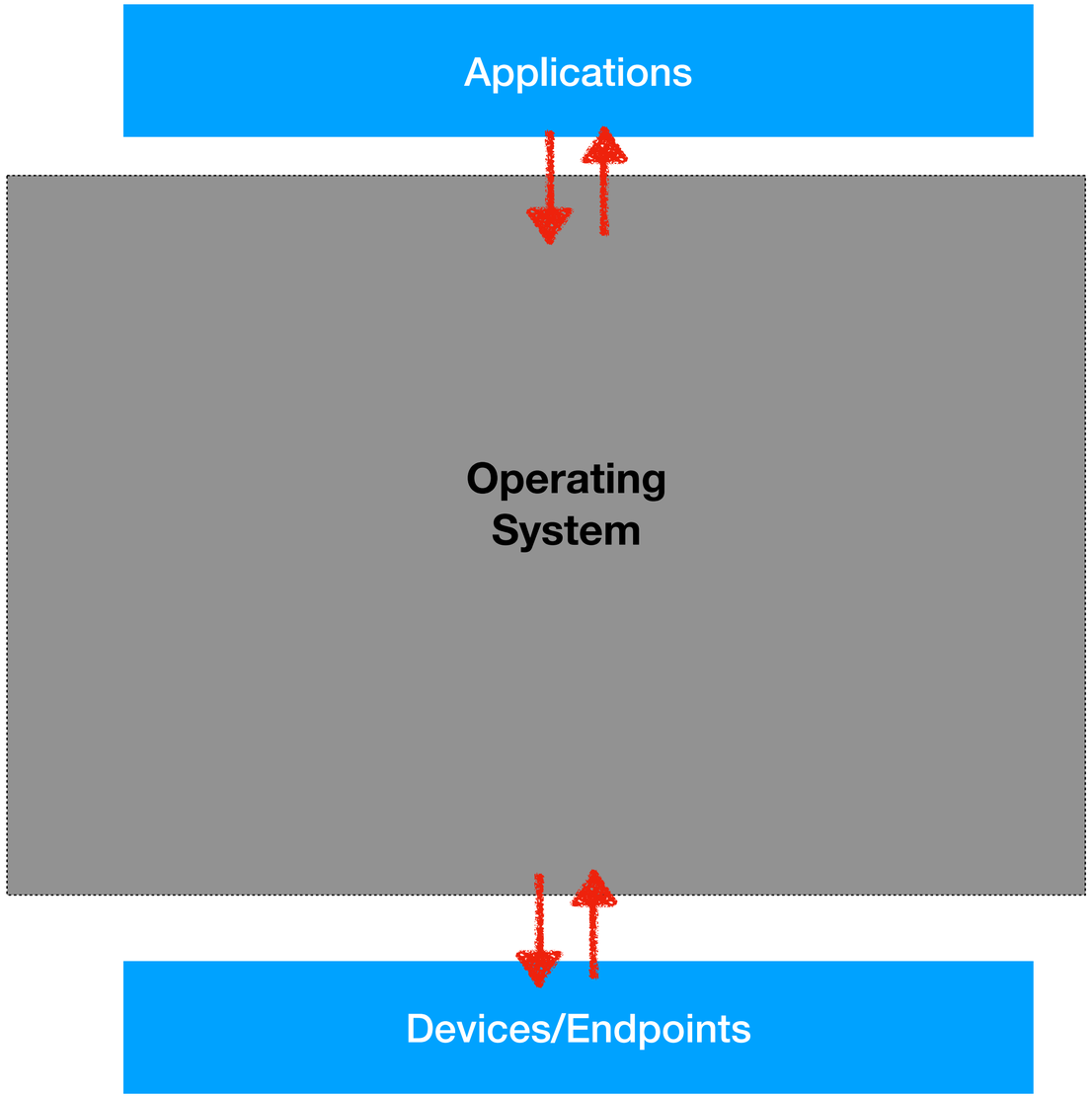
Schedule and Course Topics

Simplified “Storage Stack”



- What are some common/important applications that store/access *persistent* data?
- What interfaces do applications use to read/write their data?
- What is missing from this diagram?
- What interfaces do devices provide?
- What are some devices/endpoints that we might use to store data *persistently?*

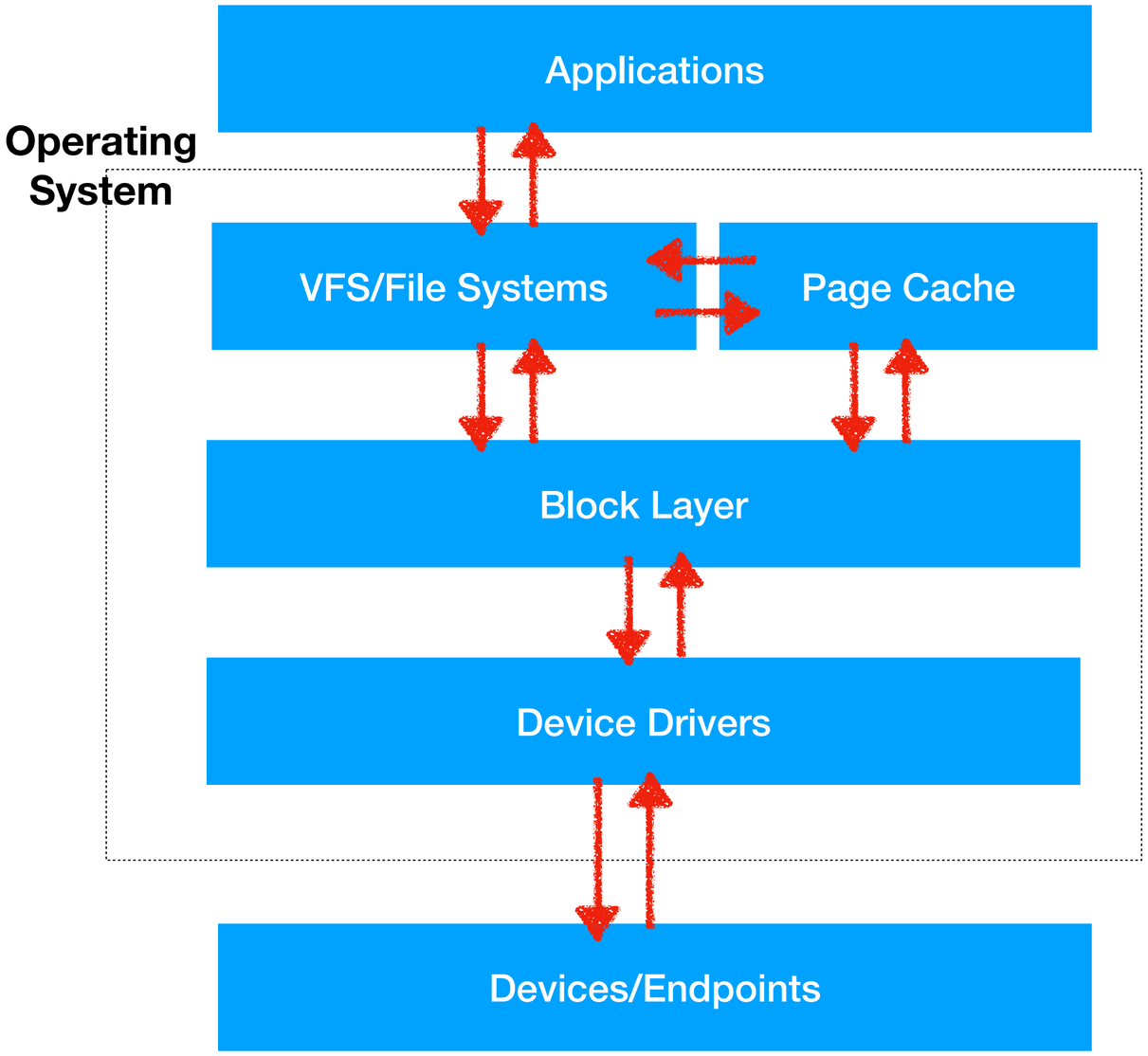
Simplified “Storage Stack”



The Operating System wears a lot of hats, but one of its jobs is to give applications a way to index their data in a uniform way.

- **What pieces of the OS are responsible for managing storage?**

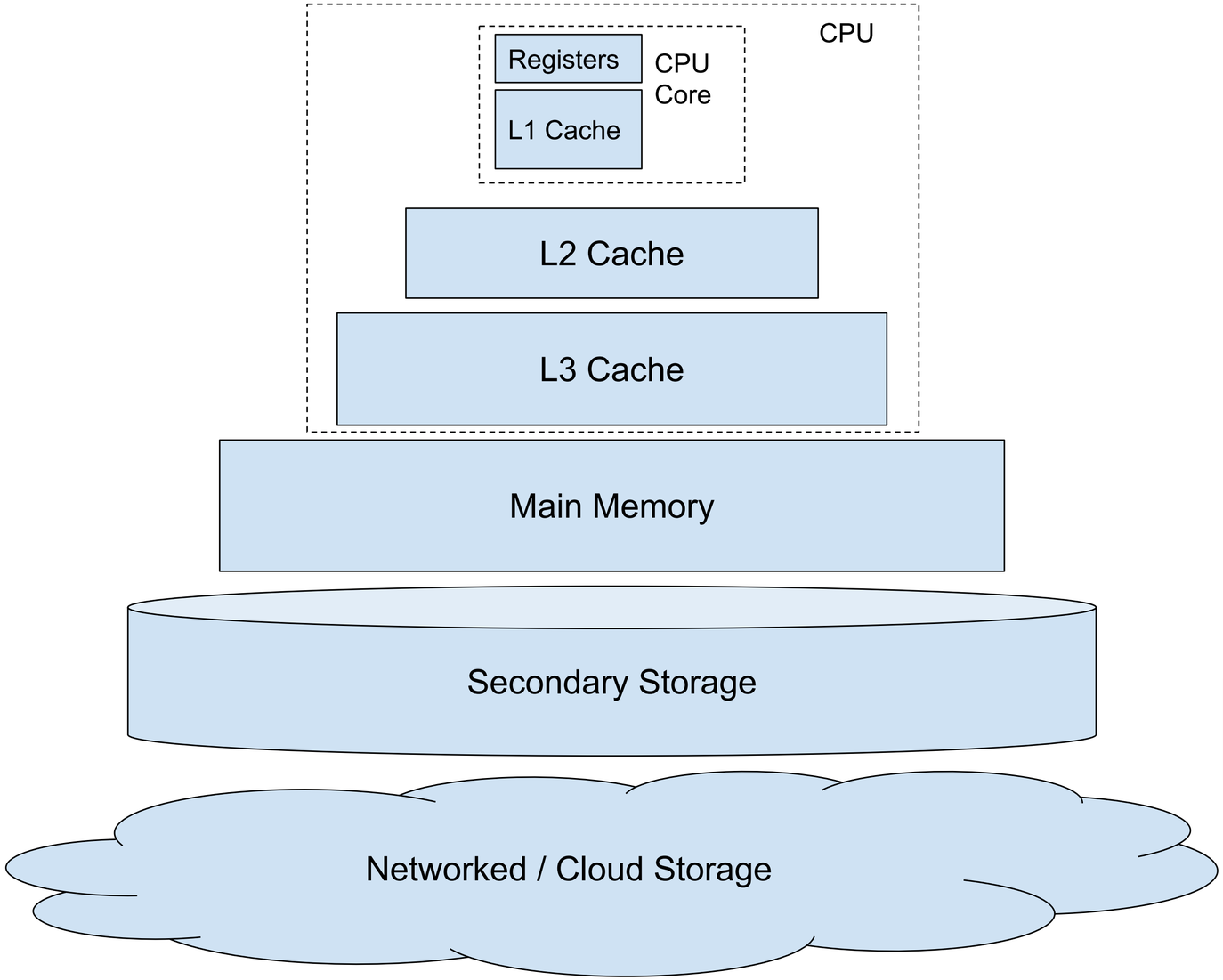
Simplified “Storage Stack”



The Operating System consists of a series of interacting layers.

- **Different interfaces at each layer's boundaries**
- **Different requirements/needs at each layer**
- **Different costs/challenges to changing the details at each layer**
  - **Ability to add APIs?**
  - **Ability to remove APIs?**
  - **Ability to modify APIs?**

The Memory Hierarchy



- **Different sizes at each level**
- **Different speeds at each level**
  - **Latency**
  - **Bandwidth**
- **Different units of transfer between each level, as influenced by points above**

The memory hierarchy is central to the design of the software storage stack from the previous slide.

Course Trajectory

- We will identify different design decisions and trade-offs made all throughout the storage stack
  - How do these decisions affect the way we design applications?
  - How do these decisions affect performance?
  - How do these decisions affect correctness?
- We'll apply performance models in order to better understand these design tradeoffs
- We'll explore different devices and how those devices affect the ways we design software
- We'll explore new data structures, and how those data structures can be used to improve system designs
- We'll have fun!

Course Structure

- Any questions about logistics?

- Course website:
  - http://cs.williams.edu/~jannen/teaching/f26/cs333/index.html
  - also accessible from https://cs.williams.edu

Folders…

- Each day, folders will be used to distribute and collect work
- When you arrive, grab your folder (sorted order)
- When you leave, return your folder (append-only)
- You are free to leave things in the folder as storage, and they will be there next class
  - but if you need something, it will be locked between classes…
- Folders streamline many things
  - attendance
  - distribution