CSC369H1F
Operating Systems

Karen Reid
csc369-2019-09@cs.toronto.edu

https://q.utoronto.ca/courses/111660

Why are you here?

Why did the department decide that an OS course is one of only 2 required 3rd year courses?

a) We like to torture students. (We had to suffer through these courses, so all CS grads must suffer likewise.)

b) Interviewers always ask OS questions, so you might as well know the answers.

c) You will probably have to write OS code in the future.

★ d) Understanding how the OS works is a fundamental concept in CS, and will help you become a better programmer/scientist.

No, really….

- How do programs run?
    - Fundamental concept of the course
- By understanding how the OS works you will have a better understanding of how the programs you write actually run.

Administrivia

- Instructor Contact:
  - Email: csc369-2019-09@cs.toronto.edu
  - Office: BA 4224
  - Office Hours: 3-4pm Mondays and Thursdays
- Webpage: quercus (feedback please!!)
- Syllabus: see web page

Prerequisites

- Make sure you have the prerequisites!

- If you don’t have the prerequisites, ask me for a waiver by email (no guarantees though!)

Course Overview

- Three assignments writing code in C (32%)
    - A1: File Systems
        - Part a) Proposal (2%)
        - Part b) Implementation (10%)
    - A2: Synchronization (10%)
    - A3: Virtual memory (10%)
- Weekly tutorial exercises for marks (5%)
- Weekly in class exercises for marks (5%)
- Midterm (13%)
- Final exam (45%)

Late policy:
10 grace tokens
1 token = 2 hours

Weekly exercises

- Strong evidence that people learn better or faster by doing rather than passively listening
- Exercises make it easier to connect lecture material (information delivery) to assignments and real world

Exercises

Exercises may have many forms:

- In class group exercises
- In class exercise, but option to hand in online
- Out of class exercise to be submitted online
- Every tutorial will have an exercise
- Tutorial exercises will be largely related to assignments

Marking Scheme:

- In class: Participation/ effort based
- Tutorial: Correctness and/ or effort

- Tutorial exercises will be best 9 out of 10
- In class exercises will be best n-2 out of n

Professionalism

- You are now part of the profession of software developers (even if you plan to go into research, or marketing, or product management)
- Work that you submit for this course should be thought of in the same way as work you submit to your boss.
- Specifications are not twisted legal documents.
    - They are an expression in English of what you are asked to do.
    - You are expected to make reasonable assumptions
        - “But it wasn’t explicitly stated” is not a sufficient argument
        - I am not out to trick you!

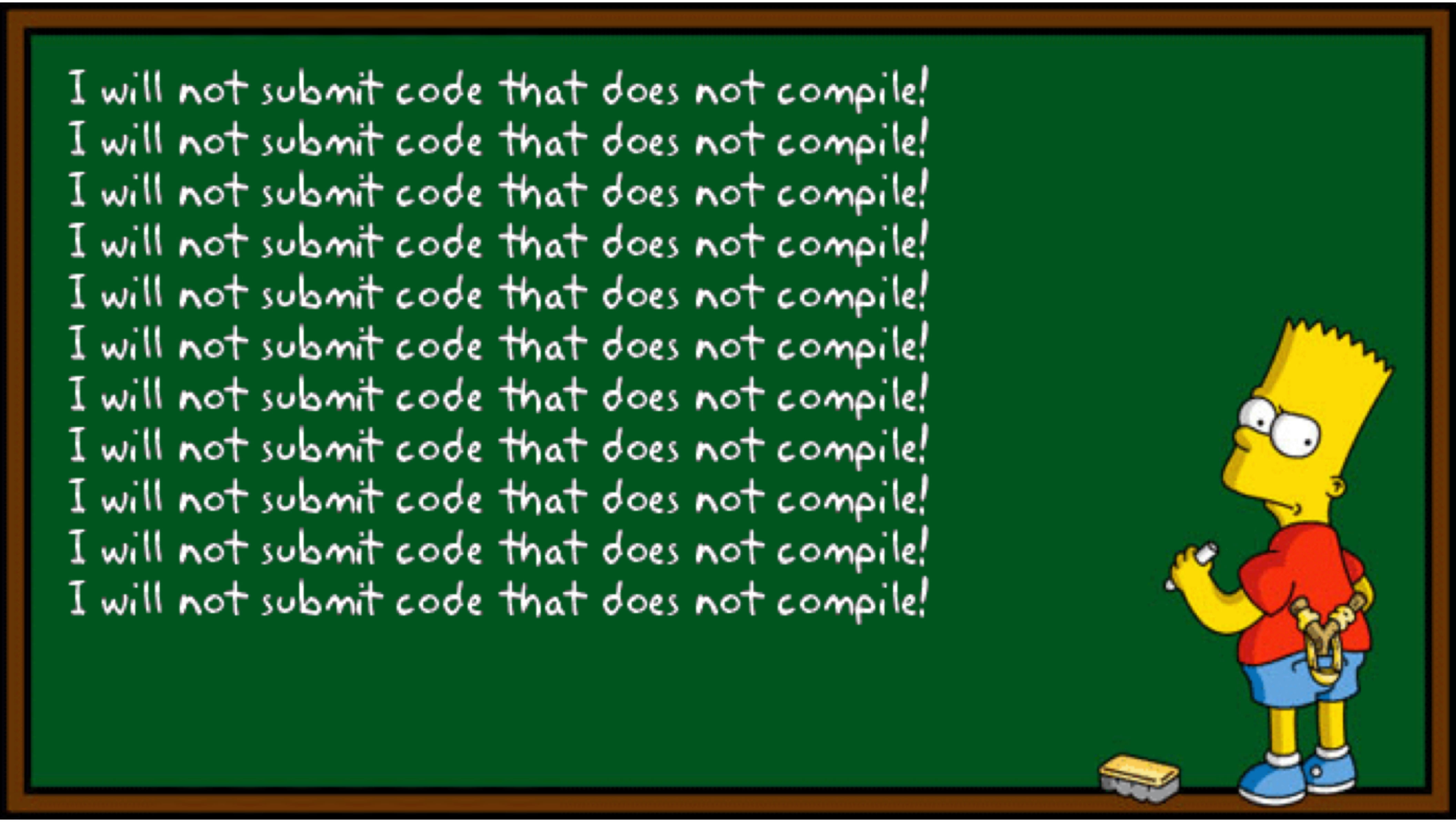
Assignments

- Write good, professional code
- Comment it properly
- Debug it properly, find corner cases (it is your responsibility to write robust code, not everything will be explicitly spelled out for you in the handout!)
- Solve problems as they come, find workarounds if needed
- When in doubt / when you're stuck, ask! Read others' questions too... may be useful for your learning experience
- Very important experience before getting a programming job

Assignments

- Due at 10:00 pm on the due date
- Code must work on the teaching labs
- Commit all source files
    - clone a new copy after you submit to be sure!
- Code style matters
- Test as you go

Code that does not compile gets zero!



I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!
I will not submit code that does not compile!

Assignments

Start early on the assignments!

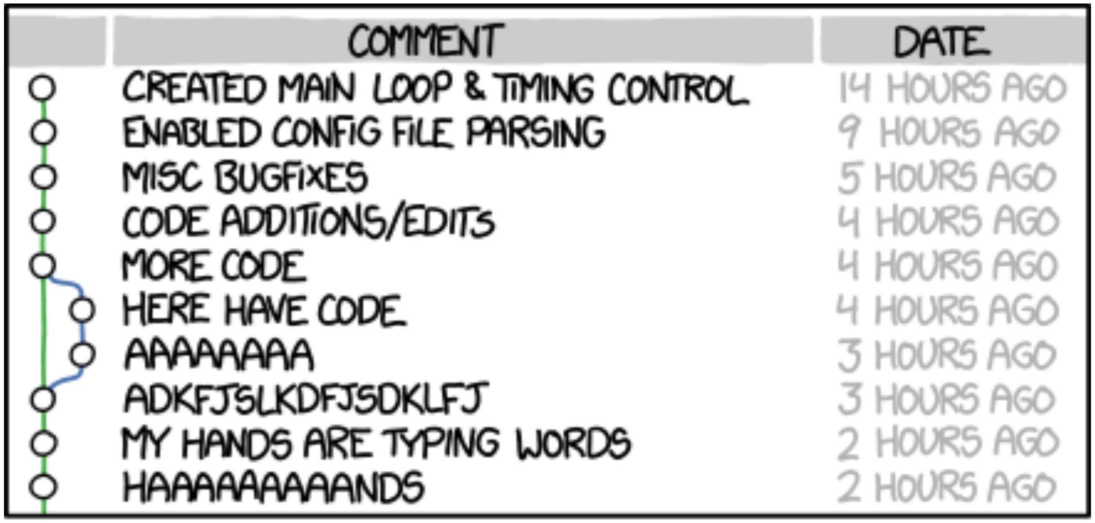
- Make sure you can commit in your repository; commit often!
- Do not wait until the very last minute to submit your assignment!
- Read the submission instructions carefully ⇒ penalties for incorrect submissions!
  - 20% penalty for remarks due to incorrect submission

Must know your code for the entire assignment!

- For partners: Work together, even if you split the work!
- May conduct interviews for some or all assignments!
  - Not having a good understanding of all code ⇒ 0 marks!

Version control

- git – use it wisely!
- Make sure to revise, if necessary
- Commit often! Don't forget to push!
- Write reasonable comments



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

- See also: https://chris.beams.io/posts/git-commit/

Help!

- Where to get help?
  - Office hours! (I don’t bite.)
  - TA lab hours
  - Tutorial
  - Piazza
  - Email me

Academic Integrity

- Plagiarism and cheating
- Very serious academic offences
- There is a clear distinction between collaboration and cheating
    - Of course you can help your friend track down a bug
    - Make sure you are doing you own work
- All potential cases will be investigated fully

Readings

Strongly Recommended!

Operating Systems: Three Easy Pieces

by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

If you want more:

Modern Operating Systems

by Andrew Tannenbaum

Do the readings!

What is the OS?

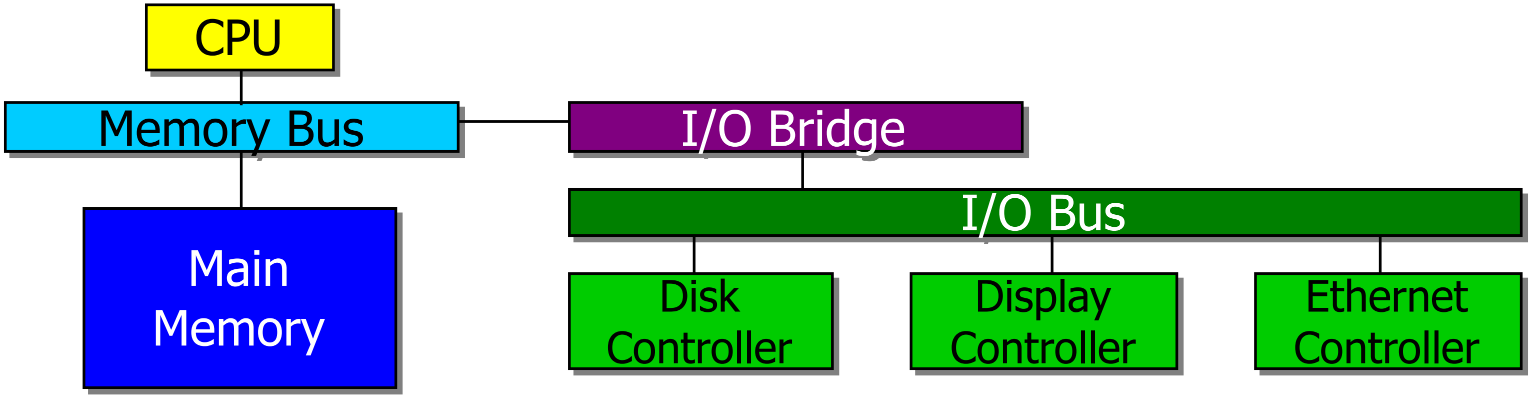
```
#include <stdio.h>
int main() {
        printf(“Hello world\n”);
}
```

User App

User App

User App

Hardware



What is an OS and why do I want one?

- How does it relate to the other parts of a computer system?
    - Convenient abstraction of H/W
    - Protection, security, authentication
    - Communication
- Make sure to review some computer organization (258), systems concepts (209) and C concepts (209)!

Goals of the OS

Primary: convenience for the user

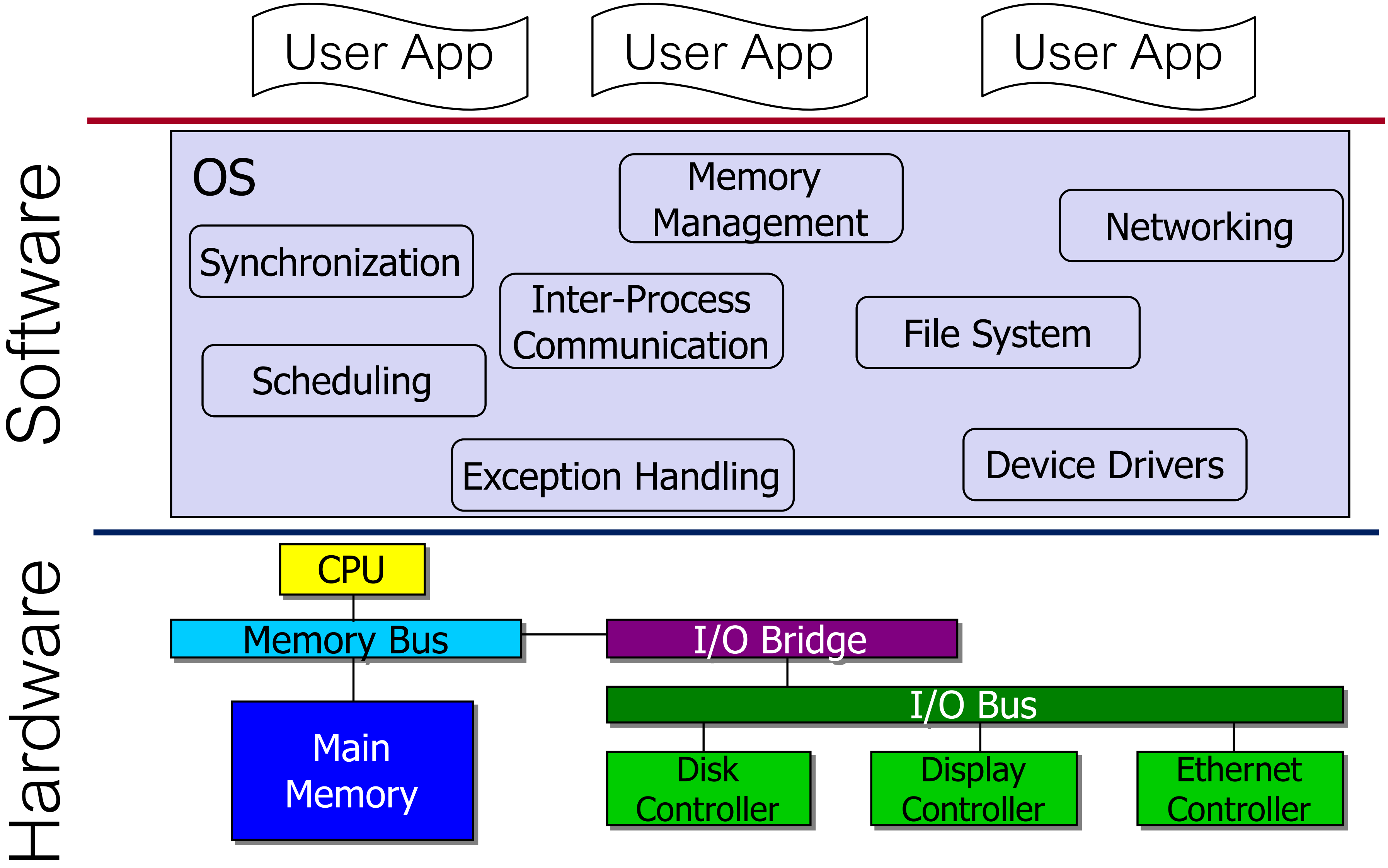
- It must be easier to compute with the OS than without it

Secondary: efficient operation of the computer system

- The two goals are ~~sometimes~~ often contradictory
- Which goal takes precedence depends on the purpose of the computer system

Roles of the OS

- An OS is a virtual machine
  - Extends and simplifies interface to physical machine
  - Provides a library of functions accessible through an API
- An OS is a resource allocator
  - allows the proper use of resources (hardware, software, data) in the operation of the computer system
  - provides an environment within which other programs can do useful work
- An OS is a control program
  - controls the execution of user programs to prevent errors and improper use of the computer
  - especially concerned with the operation and control of I/O devices



User App

User App

User App

Software

OS

Memory Management

Networking

Synchronization

Inter-Process Communication

File System

Scheduling

Exception Handling

Device Drivers

Hardware

CPU

Memory Bus

I/O Bridge

I/O Bus

Main Memory

Disk Controller

Display Controller

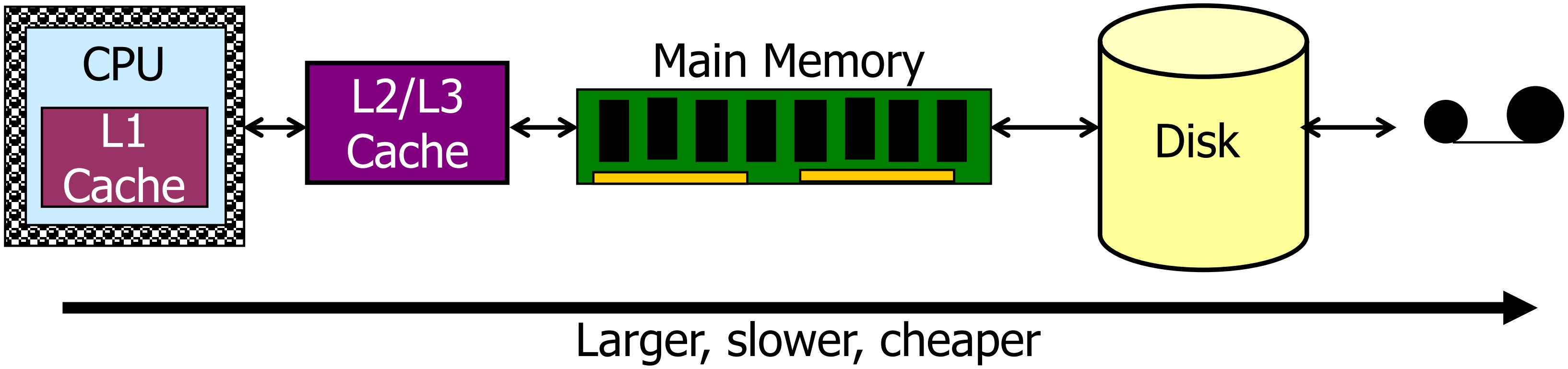
Ethernet Controller

Storage Hierarchy

- processor registers, main memory, and auxiliary memory form a rudimentary memory hierarchy

- the hierarchy can be classified according to memory speed, cost, and volatility

- caches can be installed to hide performance differences when there is a large access-time gap between two levels



Major OS Themes

Virtualization

- Present physical resource as a more general, powerful, or easy-to-use form of itself
- Present illusion of multiple (or unlimited) resources where only one (or a few) really exist
- Examples: CPU, Memory (demo)

Concurrency

- Coordinate multiple activities to ensure correctness

Persistence

- Some data needs to survive crashes and power failures

Need abstractions, mechanisms, policies for all

Next Up

- Hardware support for OS
- Bootstrapping
- Processes
  - What is a process?
  - Process lifecycle