

CSE 120

Principles of Operating Systems

Fall 2023

Lecture 1: Course Introduction

Geoffrey M. Voelker

Lecture 1 Overview

- Class overview
- Administrative info
- Introduction to operating systems
- Feel free to ask questions at any time

Personnel

- Instructor
 - ◆ Geoff Voelker
- TAs and Tutors
 - ◆ Yunxiang Chi (Tutor)
 - ◆ Edward Jin (Tutor)
 - ◆ William Lin (TA)
 - ◆ Jong Hyun (Ted) Park (Tutor)
 - ◆ Xiyan Shao (Tutor)
 - ◆ Charlotte Tang (TA)
 - ◆ Kaiyuan Wang (TA)
 - ◆ Timothy Wu (Tutor)
 - ◆ Hui Zhi (TA)

CSE 120 Class Overview

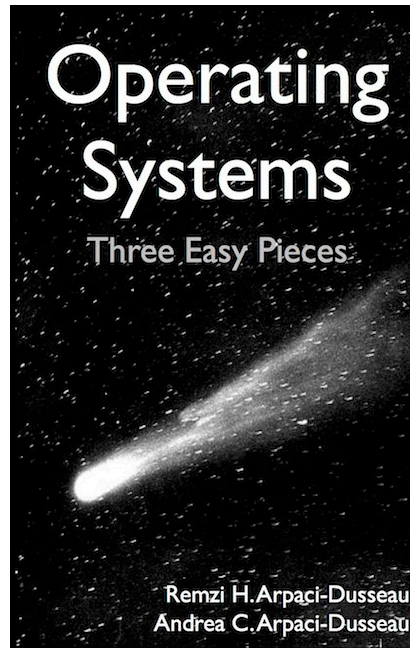
- Course material taught through class lectures, textbook readings, and assignments
 - ◆ Starting with Lecture 2, I will post slides the day before class
- Course assignments are
 - ◆ Homework questions
 - ◆ Three large programming projects in groups
 - ◆ Midterm and final exams
- Discussion sections
 - ◆ Lecture material, homework, projects
- Other forums
 - ◆ Piazza

Textbook

Remzi Arpaci-Dusseau and Andrea Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, Version 1.00



FREE



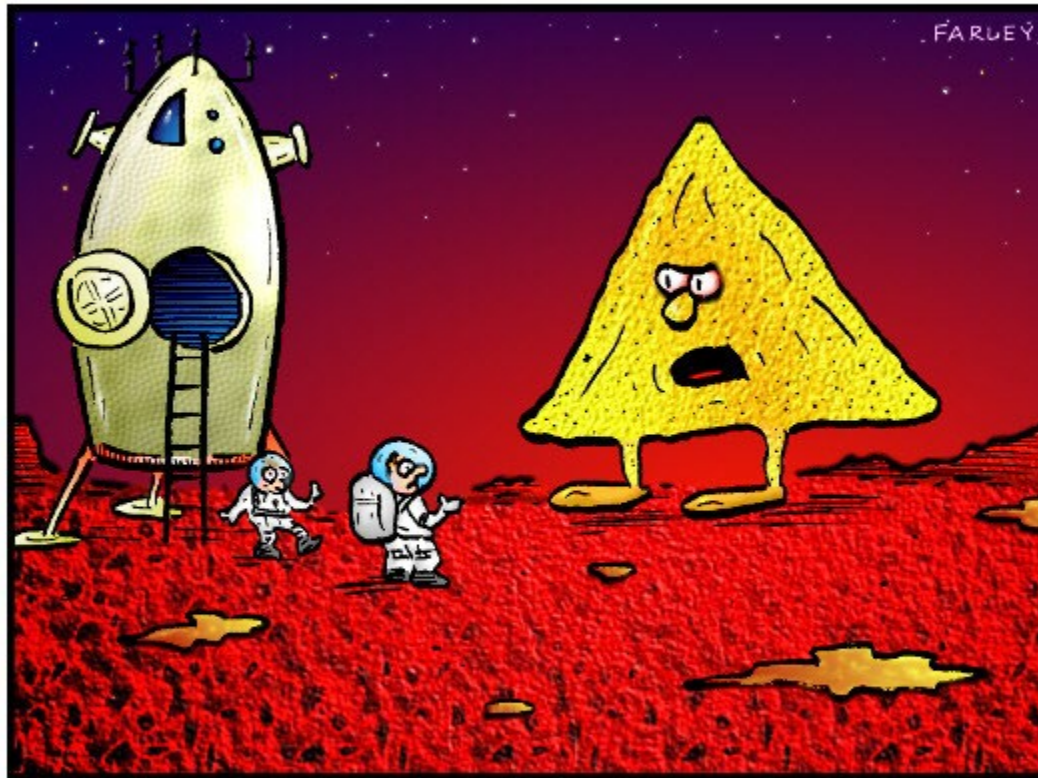
Homeworks

- There will be 4 homeworks throughout the quarter
 - ◆ Reinforce lecture material
- Homeworks provide practice learning the material
 - ◆ Unfortunately, wasted a lot of time and energy dealing with homework cheating in the past
 - ◆ So: You get full credit for a technical answer related to the homework question
 - ◆ Amount learned from doing homework is proportional to effort
 - ◆ Your choice on how much effort

Nachos Projects

DOCTOR FUN

6 Dec 94



© Copyright 1994 David Farley. World rights reserved.
This cartoon is made available on the Internet for personal viewing only.
dgfl@midway.uchicago.edu
Opinions expressed herein are not those of the University of Chicago
or the University of North Carolina.

"This is the planet where nachos rule."

Nachos

- Nachos is an instructional operating system
 - ◆ It is a user-level operating system and a machine simulator
 - » Not unlike the Java runtime environment
 - ◆ Programming environment will be Java on Unix (Linux)
 - ◆ **The projects will require serious time commitments**
 - » **Waiting until the last minute is not a good strategy**
- You will do three+ projects using Nachos
 - ◆ Concurrency and synchronization
 - ◆ System calls, processes, multiprogramming
 - ◆ Virtual memory
- You will work in **groups of 1-3** on the projects
 - ◆ Start thinking about partners
 - ◆ (Project 0 you'll do individually in the meantime)

Labs

- We will use the labs in the CSE basement
 - ♦ Linux running on x86 machines
 - ♦ ieng6.ucsd.edu as the server
- You may also use your home machine
 - ♦ The same project source will work on Windows (mostly)
 - ♦ Note: We will test and grade via gradescope
 - ♦ **Be sure to test your projects there as well**
 - » You will be able to test before the deadline
- Why work in the labs?
 - ♦ Classmates there to help (and have fun)
 - ♦ TAs there to help (will have posted hours in the lab)
 - ♦ I will visit the labs to help

In-Person Exams

- Midterm
 - ♦ Tuesday **October 31st** (put in your calendar)
- Final
 - ♦ Tuesday **December 12th** (put in your calendar)
 - ♦ Covers second half of class + selected material from first part
 - » I will be explicit about the material covered
- **No makeup exams**
 - ♦ Everyone must be able to attend these exam dates
 - » Unless absolute dire circumstances
- Crib sheet
 - ♦ You can bring **one double-sided 8.5x11” flat page** of notes to each exam to assist you in answering the questions
 - ♦ (Not a substitute for understanding the concepts, of course)

Grading

- Breakdown
 - ◆ Homeworks: 6%
 - ◆ Midterm: 28%
 - ◆ Final: 33%
 - ◆ Projects: 33%
- Course grades will be on a “curve”
- Do the work → Pass the class
 - ◆ Academic integrity violations are the main reason students fail the course

Many Ways to Interact

- Lecture
 - ◆ Ask questions!
- Discussion
 - ◆ Fri 9-9:50am (but not this Friday)
- Office hours
 - ◆ Mon 3-4pm & Wed 4-5pm
 - ◆ Any topic (lecture, project, homework, random, ...)
- Lab hours
 - ◆ TAs and tutors will have many lab hours
 - ◆ For projects, but also anything else in the course
- I'll help in the labs the week projects are due

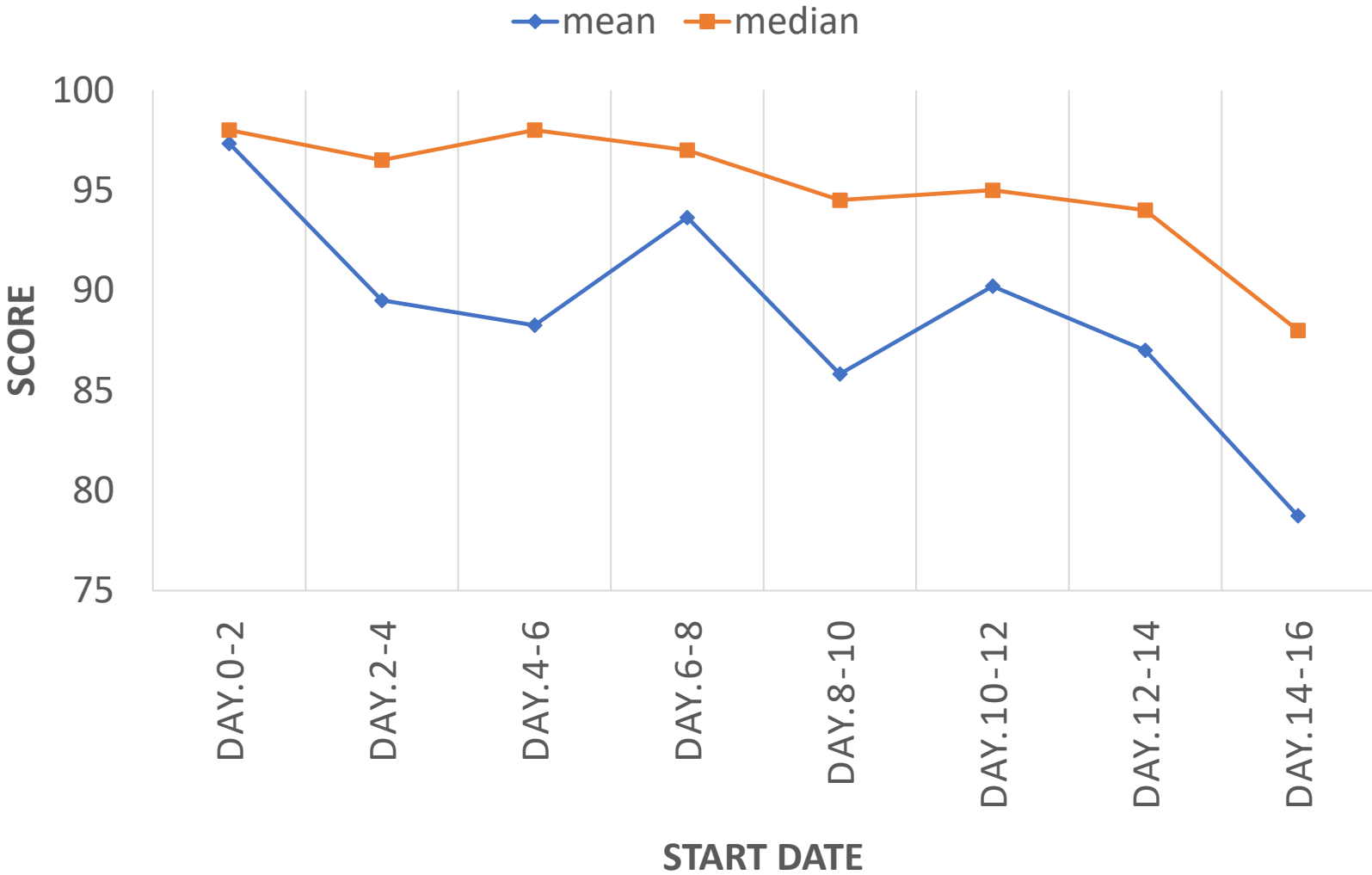
Advice

- **Attend the lectures**
 - ◆ Lecture material is the basis for exams and directly relates to the projects
- **Do the homework**
 - ◆ Concepts seem straightforward...until you apply them
 - ◆ Excellent practice for the exams, and some homework problems are exercises for helping with the project
- **Ask questions**
 - ◆ Asking questions is the best way to clarify lecture material at the time it is being presented
 - ◆ Piazza, lab + office hours will help with projects, homework

More Advice

- **Do not violate academic integrity**
 - ◆ It is much, much better to get a 0 for an assignment than to fail the course for academic integrity violations
 - ◆ If you are starting to panic – for any reason – contact me so that we can figure out a path forward
- **Start the projects early**
 - ◆ They take longer than you might expect (really!)

Project 1 Scores



Class Web Page

<https://cseweb.ucsd.edu/classes/fa23/cse120-a/>

- Serves many roles...
 - ◆ Course syllabus and schedule (updated over quarter)
 - ◆ Lecture slides
 - ◆ Homework handouts
 - ◆ Project handouts
- Optional material
 - ◆ Entirely for your interest only
- Supplemental readings
 - ◆ Seminal research papers, e.g., describing early Unix
 - ◆ Concepts in papers might seem obvious and familiar, but they were new at one time

Podcasts

- Lectures and discussions will be recorded
 - ◆ They will be available via canvas

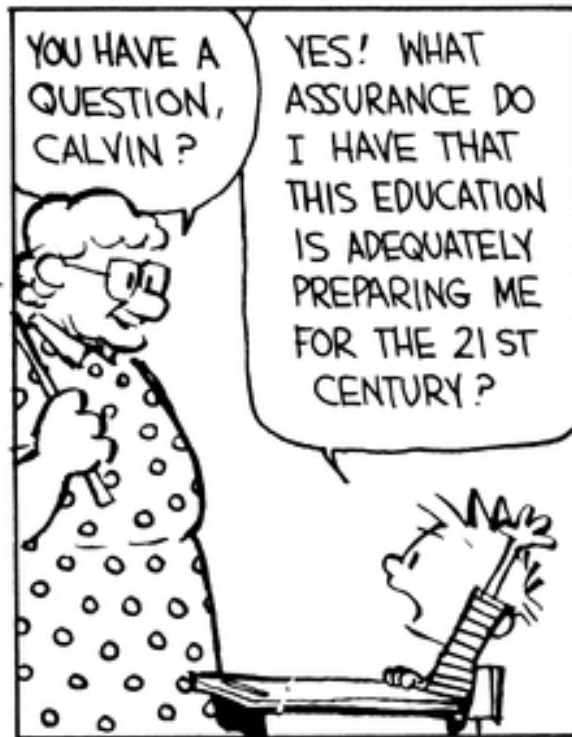
Podcasts

- Lectures and discussions will be recorded
 - ◆ They will be available via canvas
- NB: Rely upon them at your own risk
 - ◆ Occasionally recordings have failed
 - ◆ Best to think of the podcasts as supplements, not replacements

Questions

- Before we continue, any questions about the class structure, contents, etc.?

Why?



© 1992 Watterson/Distributed by Universal Uclick

AM I GETTING THE SKILLS I'LL NEED TO EFFECTIVELY COMPETE IN A TOUGH, GLOBAL ECONOMY? I WANT A HIGH-PAYING JOB WHEN I GET OUT OF HERE! I WANT OPPORTUNITY!



Why Operating Systems?

- Why take a course in operating systems?
 - ♦ It's not like everyone will become OS developers, after all
- Understand what you use
 - ♦ Understanding how an OS works helps you develop apps
 - ♦ System functionality, performance, efficiency, etc.
- Pervasive abstractions
 - ♦ Concurrency: Threads and synchronization are common modern programming abstractions (Java, C#, C++, Rust, etc.)
- Complex software systems
 - ♦ Many of you will go on to work on large software projects
 - ♦ Oses serve as examples of complex systems

CSE 120 Course Material

- This course addresses classic OS concepts
 - ◆ Services provided by the OS
 - ◆ OS implementation on modern hardware
 - ◆ Interaction of hardware and software
 - ◆ Techniques for implementing software systems that are
 - » Large and complex
 - » Long-lived and evolving
 - » Concurrent
 - » Performance-critical
- **System software tends to be mysterious**
 - ◆ Virtual memory? Wazzat?
- **Our goal is to explain those mysteries**

🔍 Type to search

🏠 Home 🔍 ☰ ☰ ☰ ☰

- 🕒 Recent
- ★ Starred
- 🏠 Home
- 📁 Documents
- 📁 Downloads
- 🎵 Music
- 🖼️ Pictures
- 📺 Videos

📁 Desktop 📁 Documents 📁 Downloads 🎵 Music

🖼️ Pictures 🌐 Public 📄 Templates 📺 Videos

☰
☰
☰

🔍 Settings ☰

- 🌐 Network
- 📶 Bluetooth
- 🎨 Appearance
- 🔔 Notifications
- 🔍 Search
- 📱 Multitasking
- 🗄️ Applications >
- 🔒 Privacy >
- 👤 Online Accounts
- 👤 Sharing
- 🔊 Sound
- 🔋 Power
- 🔄 Updates

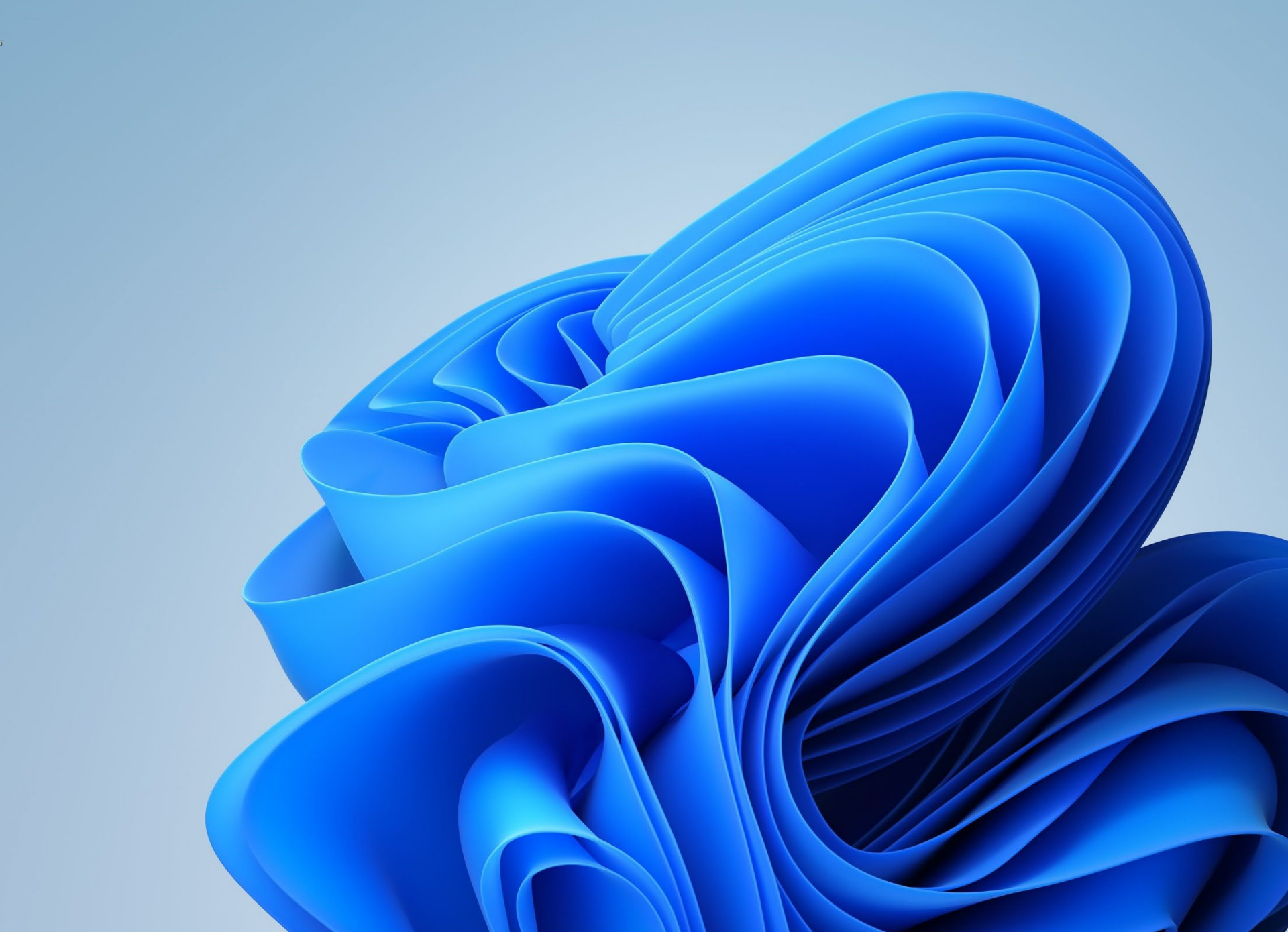
Appearance

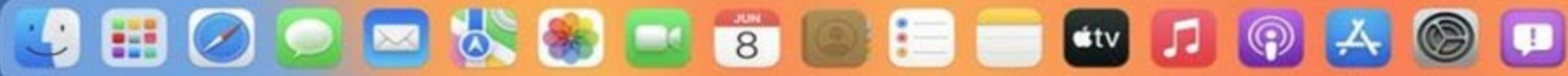
📁 Light 📁 Dark

Background + Add Picture...

🔧

🌐 📅 🔊 📄 📁 ⚙️ ☰







```

top - 10:05:04 up 373 days, 1:29, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 206 total, 1 running, 122 sleeping, 1 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 98967544 total, 72343520 free, 1141584 used, 25482440 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 96887280 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	PU	%MEM	TIME+	COMMAND
27210	voelker	20	0	33536	3692	316	3	0.0	0:00.05	top
27211	root	20	0	66208	5360	466	3	0.0	0:00.01	sshd
27877	root	20	0	0	0	0	3	0.0	0:05.72	kworker/0:2
1	root	20	0	225572	9432	6	0.0	0.0	19:46.34	systemd
2	root	20	0	0	0	0	0.0	0.0	0:07.77	kthreadd
4	root	0	-20	0	0	0	0.0	0.0	0:00.00	kworker/0:+
6	root	20	0	0	0	0	0.0	0.0	0:57.38	kworker/u1+
7	root	0	-20	0	0	0	0.0	0.0	0:00.00	mm_percpu_+
8	root	20	0	0	0	0	0.0	0.0	0:17.02	ksoftirqd/0
9	root	20	0	0	0	0	0.0	0.0	191:58.78	rcu_sched
10	root	20	0	0	0	0	0.0	0.0	0:00.00	rcu_bh
11	root	rt	0	0	0	0	0.0	0.0	0:02.67	migration/0
12	root	rt	0	0	0	0	0.0	0.0	0:57.85	watchdog/0
13	root	20	0	0	0	0	0.0	0.0	0:00.00	cpuhp/0
14	root	20	0	0	0	0	0.0	0.0	0:00.00	cpuhp/1
15	root	rt	0	0	0	0	0.0	0.0	0:55.63	watchdog/1
16	root	rt	0	0	0	0	0.0	0.0	0:03.08	migration/1

Resource Monitor

File Monitor Help

Overview CPU Memory Disk Network

CPU 7% CPU Usage 156% Maximum Frequency

Image	PID	Descripti...	Status	CPU	Averag...
System Interrupts	-	Deferre...			0.88
perfmon.exe	15700	Resourc...			0.44
wgc_renderer.exe	13932	Warga...		26	0.15
System	4	NT Ker...	Running	299	0.58
wgc_renderer.exe	13716	Warga...	Running	14	0.03
Taskmgr.exe	10488	Task Ma...	Termin...	32	0.09
Slack.exe	27536	Slack	Running	18	0.32
explorer.exe	6540	Window...	Running		0.09
opera.exe	4884	Opera l...	Running	0	0.01
chrome.exe	13372	Chrom...	Running	0	0.01

Disk 9 MB/sec Disk I/O Active Time

Network 6 Kbps Network I/O 0% Network Utilization

Memory 0 Hard Faults/sec Physical Memory

Image	PID	Hard Fa...	Commi...	Shareab...	Private (...)
Image					
Memory Compression	3800	0	3,968	1,303,0...	0
Dropbox.exe	23600	0	641,196	538,952	236,556
POWERPNT.EXE	3128	0	357,356	432,952	151,872
chrome.exe	2960	0	260,864	256,736	43,288
chrome.exe	12128	0	652,904	441,580	251,456
opera.exe	21436	0	689,956	368,640	201,324
chrome.exe	14688	0	242,580	298,360	138,876
opera.exe	20032	0	209,676	225,460	88,904
mspaint.exe	25948	0	129,852	149,652	21,940
SearchIndexer.exe	7884	0	381,852	158,812	40,784

CPU 100%

Disk 10 MB/sec

Network 10 Kbps

Memory 100 Hard Faults/sec

Fundamental OS Concepts

- Processes and threads
 - ♦ What they are, why we have them, how to implement them
- Correct concurrent programs
 - ♦ Concurrency, synchronization
- Virtual memory
 - ♦ What it is, why we have it, how to make it work
- File systems
 - ♦ Making persistent storage friendly to users and applications
- Some advanced topics at the end
- Goal: By the end of the quarter you will look at your computer in a completely different way

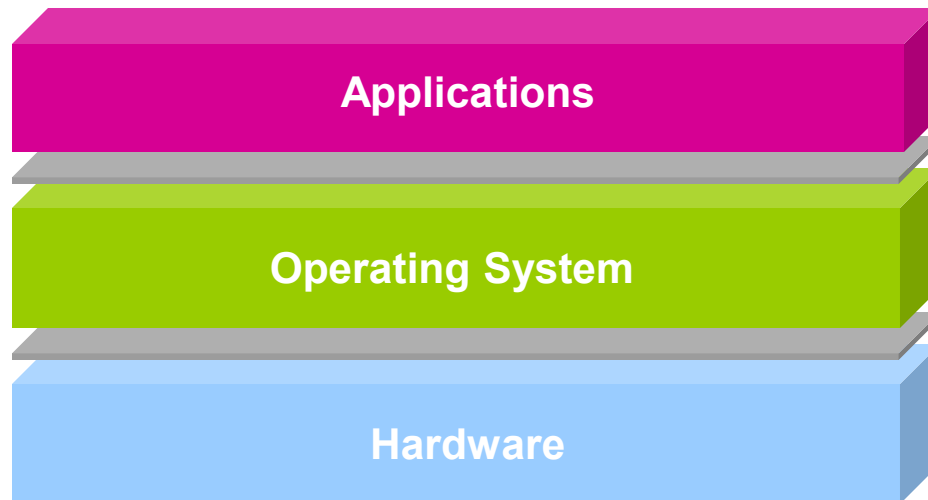


What is an Operating System?

- How would you answer?
 - ◆ (Yes, I know that's why you're taking the course...)
 - ◆ (Note: There are many answers...)

What is an Operating System?

- The operating system is the software layer between user applications and the hardware



- The OS is “all the code that you didn’t have to write” to implement your application

What Programmers See

```
#include <stdio.h>

int
main (int argc, char *argv[]) {
    printf ("hello world\n");
}
```


The OS and Hardware

- The OS **abstracts/controls/mediates** access to hardware resources
 - ◆ Computation (CPUs)
 - ◆ Volatile storage (memory) and persistent storage (disk, etc.)
 - ◆ Communication (network, modem, etc.)
 - ◆ Devices (keyboard, display, printer, camera, etc.)
- The OS defines a set of logical resources (**objects**) and a set of well-defined operations on those objects (**interfaces**)
 - ◆ Physical resources (CPU and memory)
 - ◆ Logical resources (files, programs, names)
 - ◆ Sounds like OO...

The OS and Hardware (2)

- Benefits to applications
 - ◆ Simpler (no tweaking device registers)
 - ◆ Device independent (all network cards look the same)
 - ◆ Portable (across Win95/98/ME/NT/2000/XP/Vista/7/8/10/11/...)
 - ◆ Transportable (same program across different OSes (Javascript))

The OS and Applications

- The OS defines a **logical, well-defined environment**...
 - ◆ Virtual machine (each program thinks it owns the computer)
- ...for users and programs to **safely coexist, cooperate, share resources**
 - ◆ Concurrent execution of multiple programs (timeslicing)
 - ◆ Communication among multiple programs (pipes, cut & paste)
 - ◆ Shared implementations of common facilities
 - » No need to implement the file system more than once
 - ◆ Mechanisms and policies to manage/share/protect resources
 - » File permissions (mechanism) and groups (policies)

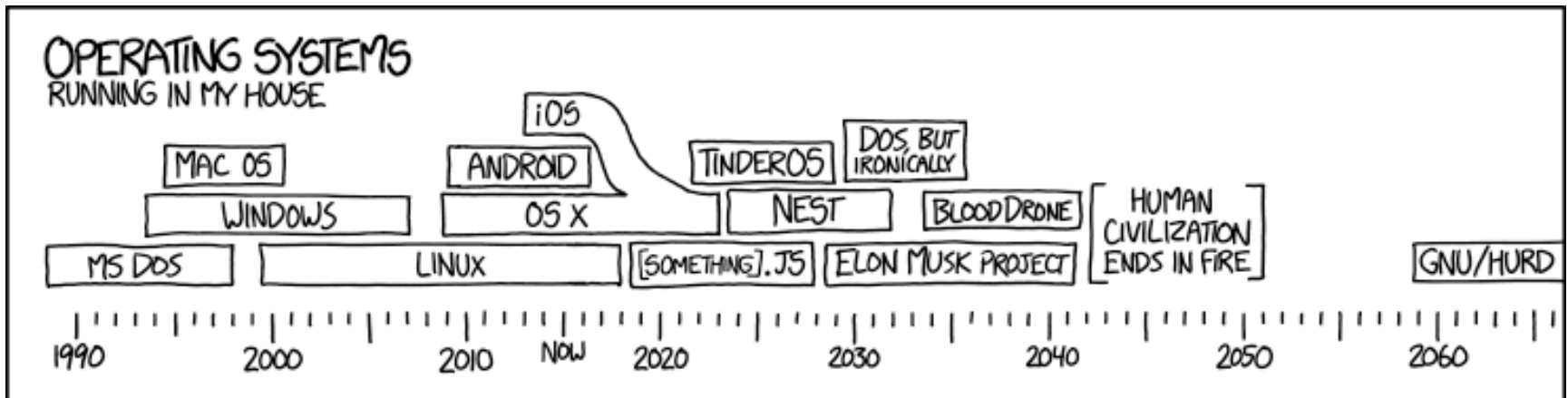
More Questions to Ponder

- What is part of an OS? What is not?
 - ◆ Is the windowing system part of an OS?
 - ◆ Is the Web browser part of an OS?



More Questions to Ponder

- What is part of an OS? What is not?
 - ◆ Is the windowing system part of an OS?
 - ◆ Is the Web browser part of an OS?
- Popular OSes today are Windows, Linux, and OS X
 - ◆ How different/similar do you think these OSes are?
 - ◆ How would you go about answering that question?



More Questions to Ponder

- What is part of an OS? What is not?
 - ◆ Is the windowing system part of an OS?
 - ◆ Is the Web browser part of an OS?
- Popular OSes today are Windows, Linux, and OS X
 - ◆ How different/similar do you think these OSes are?
 - ◆ How would you go about answering that question?
- OSes change all of the time
 - ◆ Consider the series of releases of Windows, Linux, OS X...
 - ◆ What are the drivers of OS change?
 - ◆ What are the most compelling issues facing OSes today?

Pondering Cont'd

- How many lines of code in an OS?
 - ♦ Win10 (2015): 50M
 - ♦ OS X (2006): 86M
 - ♦ Linux (2020): 28M
 - ♦ What is largest kernel component?
- What does this mean (for you)?
 - ♦ OSes are useful for learning about software complexity
 - ♦ OS is just one example of many complex software systems
 - » Chrome (2015): 17M
 - » Hadoop (2018): 3.9M
 - » JDK (2015): 6M
 - » Unreal Engine 4: 2.3M
 - ♦ As a software developer, you will face complexity

For next class...

- Browse the course web
<https://cseweb.ucsd.edu/classes/fa23/cse120-a/>
- Catch up on Chapters 1 & 2
- Read Chapter 6 for Tue lecture
- Start thinking about partners for project groups
- I will stay after to answer any additional questions