CSE 120
Operating Systems Principles

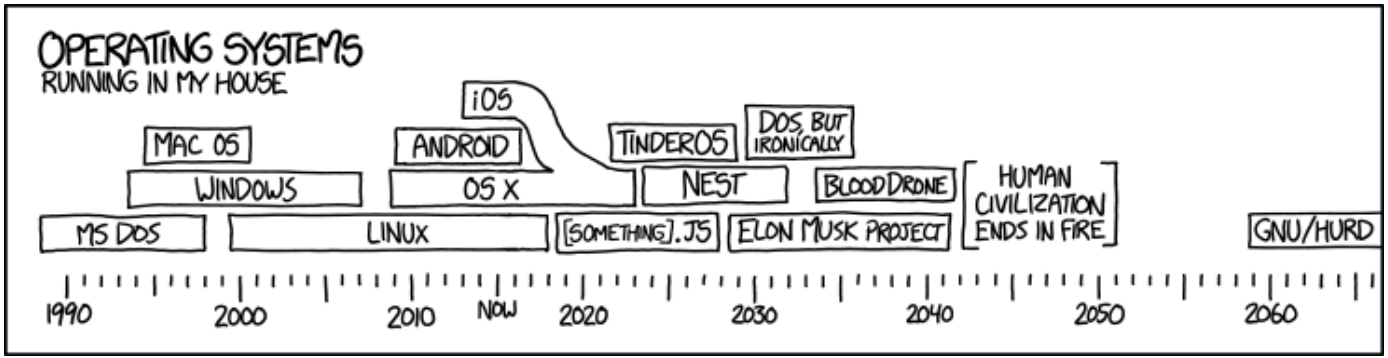
Spring 2025

Lecture 1: Course Introduction

Amy Ousterhout

This Course

- How do operating systems work?



https://xkcd.com/1508/

Lecture 1 Overview

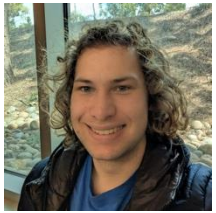
- Introductions
- Class overview
- Administrative info
- Introduction to operating systems

- Feel free to ask questions at any time

Course Staff

- Instructor
  - Amy Ousterhout
- TAs


Jonghyun (Ted) Park


Nick Petrone


Tianyi Shan

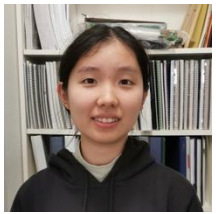

Mingyao Shen

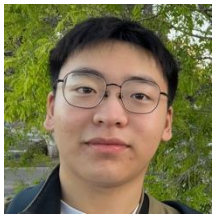
- Tutors


Raj Nawal


Arthur Utecht


Nicole Wong


Yuehua Xie


Andrew Yang

CSE 120 Class Overview

- Course material taught through class lectures, textbook readings, and assignments
- Course assignments are:
  - Homework questions
  - Three large programming projects in groups
  - In-class polls
  - Midterm and final exams
- Discussion sections
  - Projects and homework assignments
- Piazza
  - Ask questions via Piazza rather than email

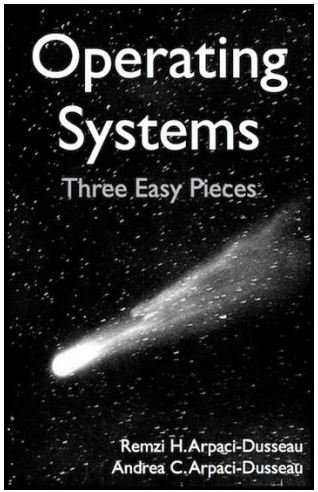
Class Website

- https://cseweb.ucsd.edu/classes/sp25/cse120-a/
- Serves many roles:
  - Course syllabus and schedule (updated over the quarter)
  - Lecture slides
  - Homework assignments
  - Project information
- Optional readings
  - Entirely for your interest only

Week	Date	Topic	Readings	Due
1	Tue 4/1	Course Intro	Ch. 1, 2	
	Thu 4/3	Interactions with Apps and Hardware	Ch. 6	
	Fri 4/4	Project 0		
2	Tue 4/8	Processes	Ch. 3, 4, 5	Project 0
	Thu 4/10	Threads	Ch. 26, 27	
	Fri 4/11	Project 1		Homework 1
3	Tue 4/15	Synchronization	Ch. 28, 29	
	Thu 4/17	Semaphores	Ch. 31	
	Fri 4/18	Project 1		

Textbook

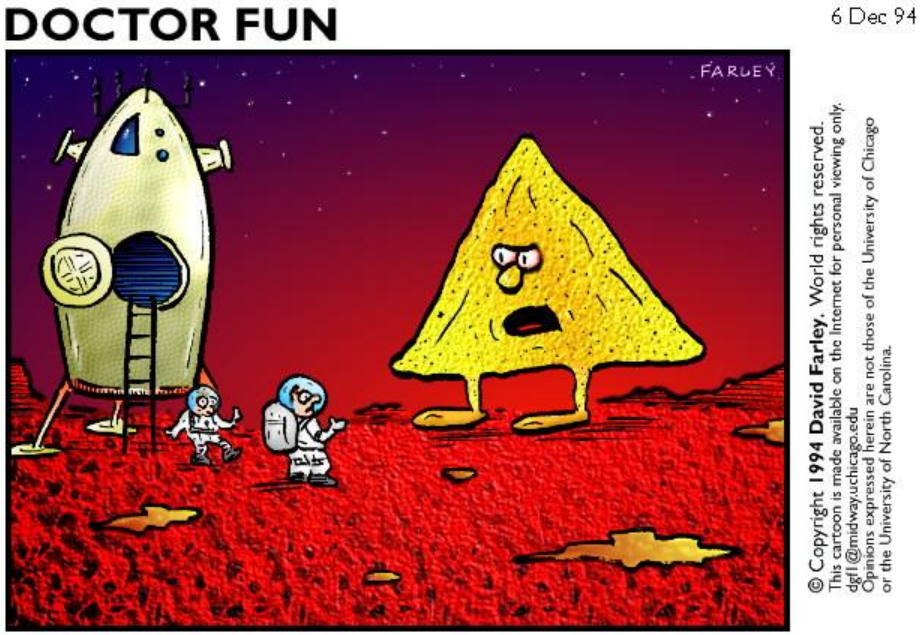
Remzi Arpaci-Dusseau and Andrea Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, Version 1.00



Homeworks

- There will be 4 homeworks throughout the quarter
  - Reinforce the lecture material
  - Prepare you for the exams
- Grading
  - 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
- Feel free to collaborate

Nachos Project



"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 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
  - (You will do project 0 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 (your code runs here)
- You may also use your home machine
  - ssh to ieng6 (e.g., using VSCode)
- Testing and grading via gradescope
  - Make sure to test your projects there
  - You will be able to test before submitting
- Why work in the labs?
  - Classmates there to help (and have fun)
  - TAs and tutors there to help (they have posted hours in the lab)

In-Person Exams

- Midterm
  - Thursday May 1st (put in your calendar)
- Final
  - Monday June 9th (put in your calendar)
  - Covers entire quarter
- No makeup exams
  - Everyone must be able to attend these exam dates
    - » Unless absolute dire circumstances
- Notes 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)

In-Class Polls

- Several polls throughout the quarter
  - During lecture
  - Check if you understand the concepts
  - Conducted via Gradescope
  - Graded for participation
    - » You can miss 2 polls without impacting your grade

Grading

- Breakdown
  - In-class polls: 6%
  - Homeworks: 6%
  - Midterm: 22%
  - Final: 33%
  - Projects: 33%
- Grades will be curved
- Do the work → pass the class
  - Academic integrity is the main reason students fail the course

Late Policy

- Applies to homeworks and programming projects
  - Submit ≤ 24 hours after the deadline: grade reduced by 10%
  - Submit 24-48 hours after the deadline: grade reduced by 30%
- Exception: project 3 cannot be submitted late

Many Ways to Get Help

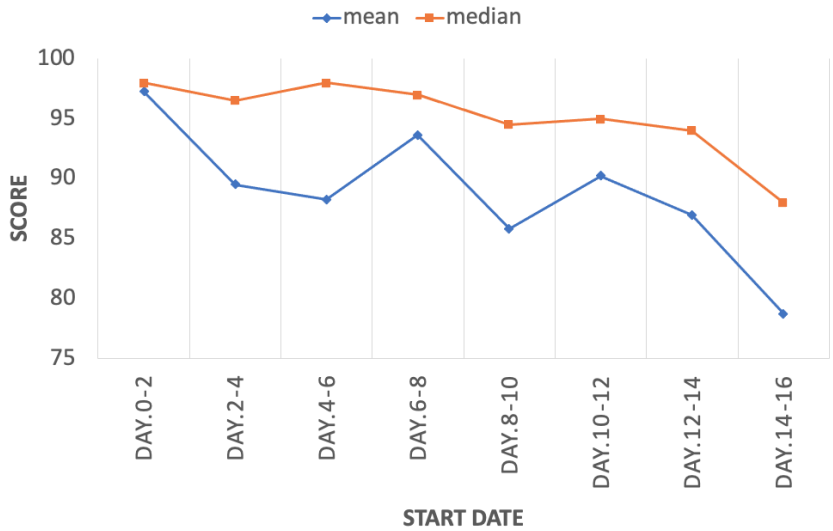
- Lecture
  - Tue and Thurs 3:30-4:50pm (York 2722)
  - Ask questions!
- Discussion Section
  - Fri 4-4:50pm (Peterson 110)
- Office hours
  - Wed 1:00-2:00 pm and Thu 2:00-3:00 pm (CSE 3130)
  - All topics (lecture, project, homework, random, …)
- Lab hours
  - TAs and tutors will have many lab hours
  - For projects, but also anything else in the course

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
  - Some homework problems are exercises to prepare you for the projects
- Ask questions
  - Asking questions in lecture/discussion is the best way to clarify material quickly
  - Piazza, lab + office hours will help with projects, homework

More Advice

- Start the projects early
  - They take longer than you might expect (really!)
  - Project 1 scores:



Academic Integrity

- 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
- Projects
  - Each team must write their own solution
    - » No sharing of code or written answers is allowed
  - Explicitly acknowledge any outside sources
  - We have tools to detect cheating
- Exams
- In-class polls
  - Only submit from lecture, only submit on behalf of yourself

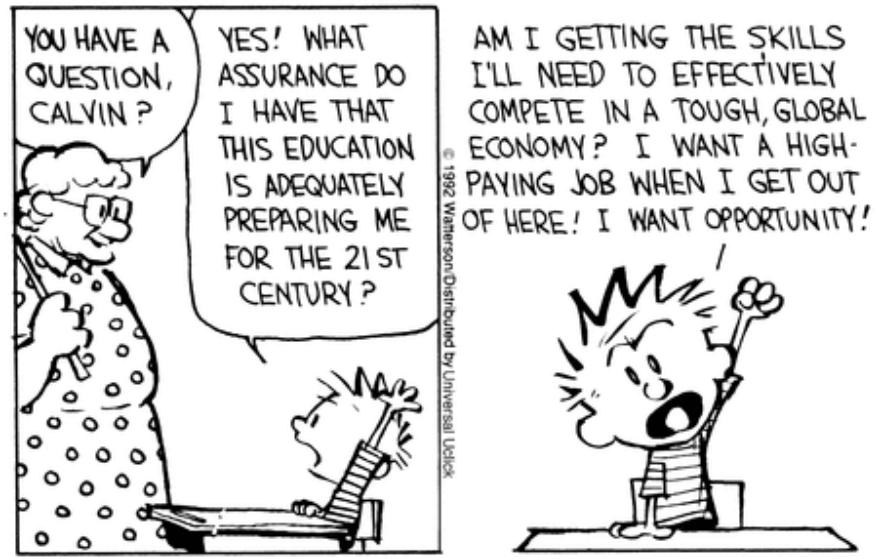
Podcasts

- Lectures and discussions will be recorded
  - Available via Canvas
- Rely upon them at your own risk
  - Recordings do fail occasionally
  - Think of the podcasts as supplements, not replacements

Questions

- Any questions about the class structure, contents, etc.?

Why?


YOU HAVE A QUESTION, CALVIN?
YES! WHAT ASSURANCE DO I HAVE THAT THIS EDUCATION IS ADEQUATELY PREPARING ME FOR THE 21ST CENTURY?
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!
© 1992 Watterson/Distributed by Universal Uclick

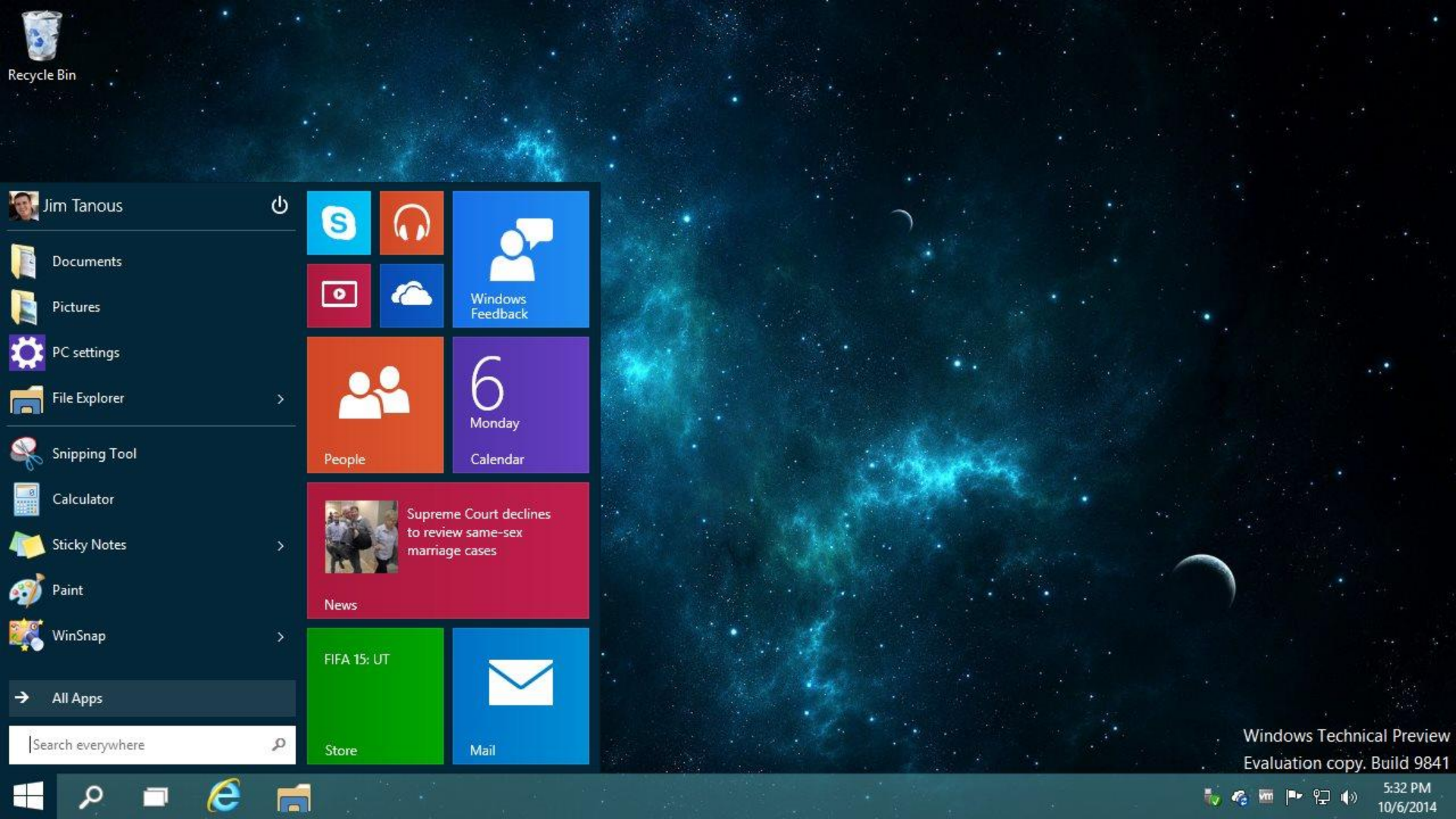
Why Operating Systems?

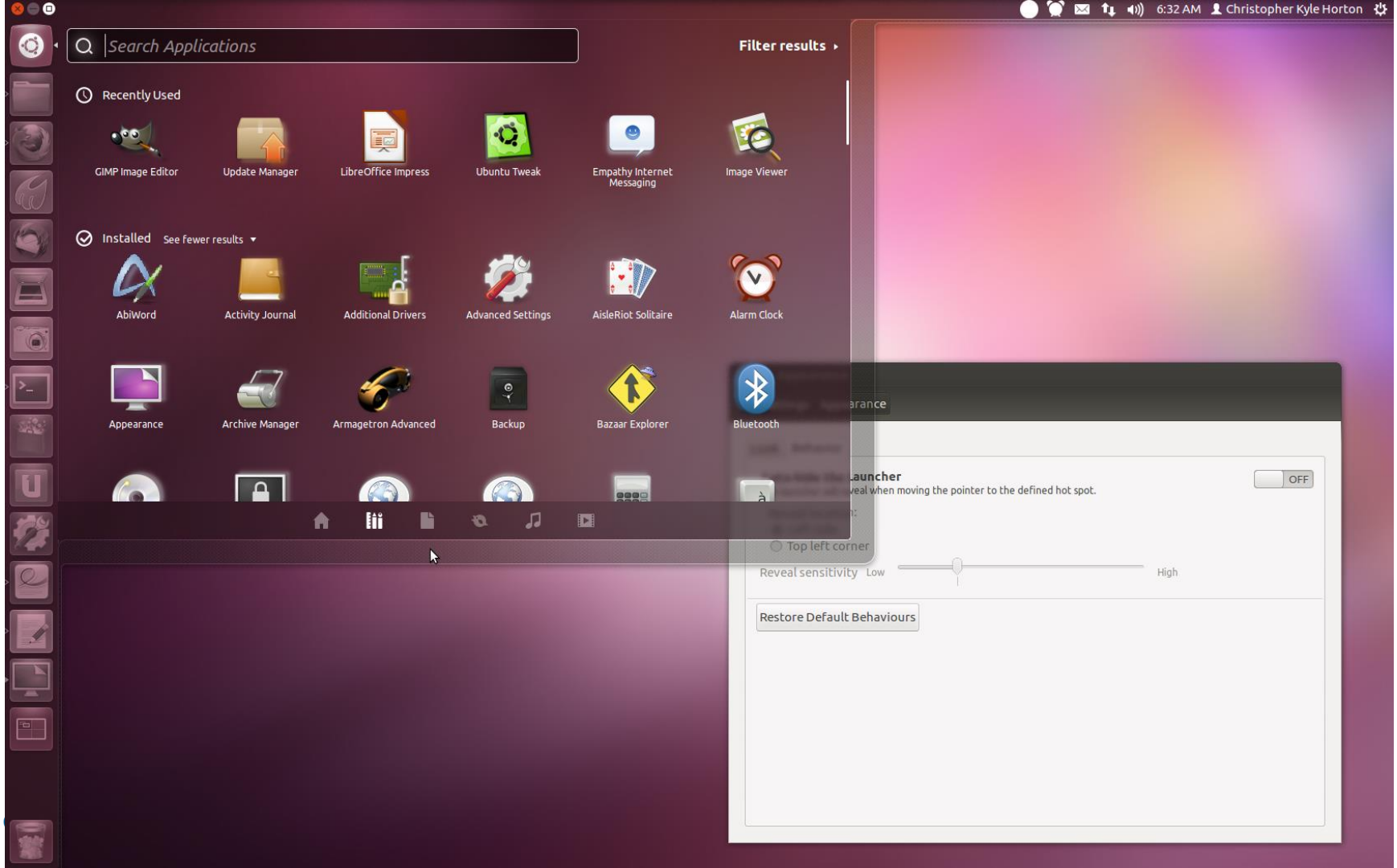
- Why take a course in operating systems?
  - Not everyone will become OS developers, after all
- Understand what you use
  - Understanding how an OS works helps you develop apps
  - System functionality, performance, efficiency, etc.
- Pervasive abstractions
  - Elegant ways of decomposing complex problems
  - 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

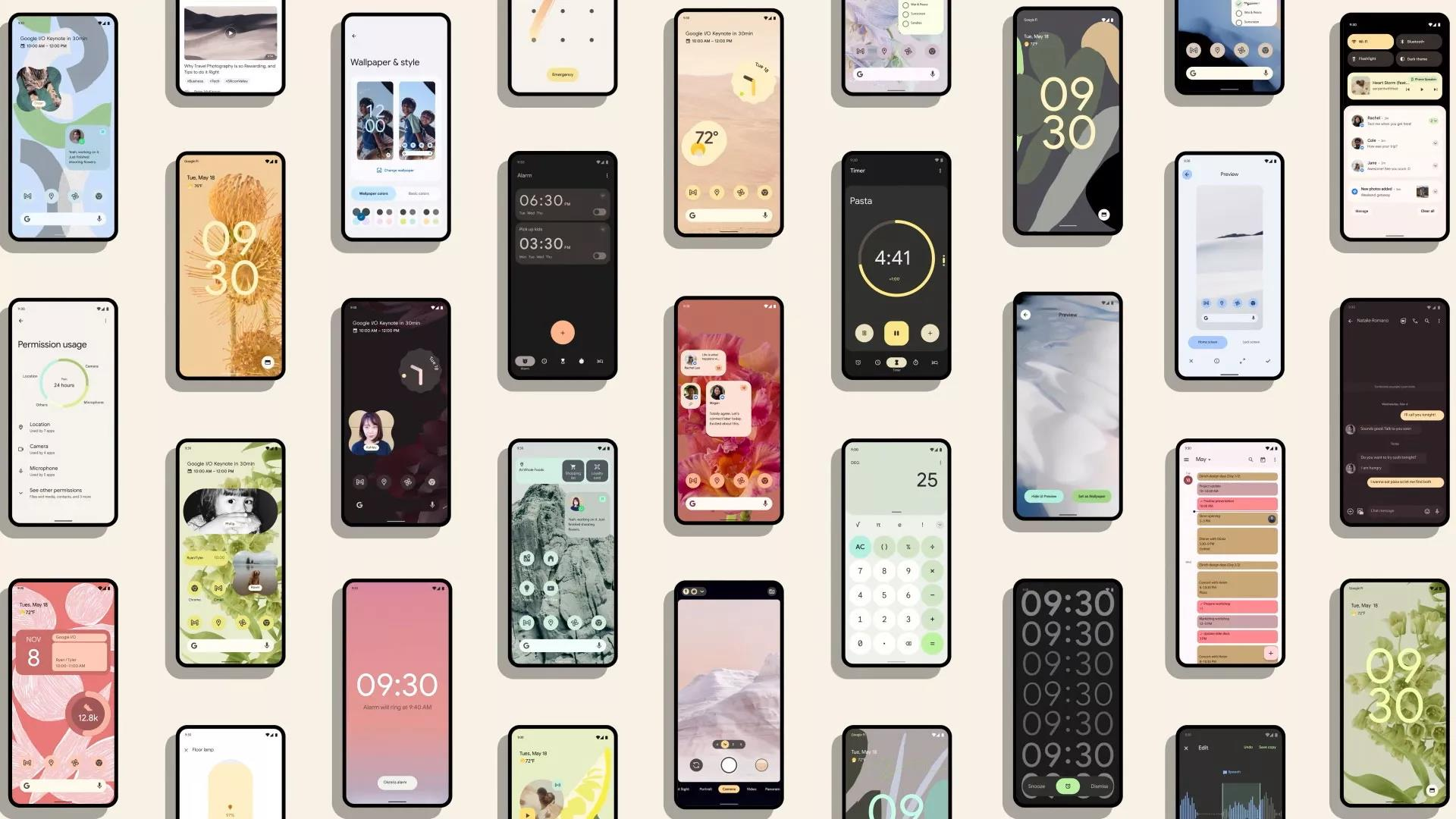
- This course addresses classic OS concepts
  - Services provided by the OS
  - OS implementation on modern hardware
  - Interactions between hardware and software
  - Techniques for implementing software systems that are
    - » Large and complex
    - » Long-lived and evolving
    - » Concurrent
 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


Today
Notifications
Thursday,
October 16th
Weather
9:20 AM
Tomkins Cove
68°
9:20 AM
New York
68°
9:20 AM
Albany
70°
9:20 AM
Poland
64°
Show More...
World Clock
New York
San Francisco
Social
Calculator
0
C
+/–
%
÷
7
8
9
×
4
5
6
–
Edit


Recycle Bin
Jim Tanous
Documents
Pictures
PC settings
File Explorer
Snipping Tool
Calculator
Sticky Notes
Paint
WinSnap
All Apps
Search everywhere
Windows Feedback
People
6
Monday
Calendar
Supreme Court declines to review same-sex marriage cases
News
FIFA 15: UT
Store
Mail
Windows Technical Preview
Evaluation copy. Build 9841
5:32 PM
10/6/2014

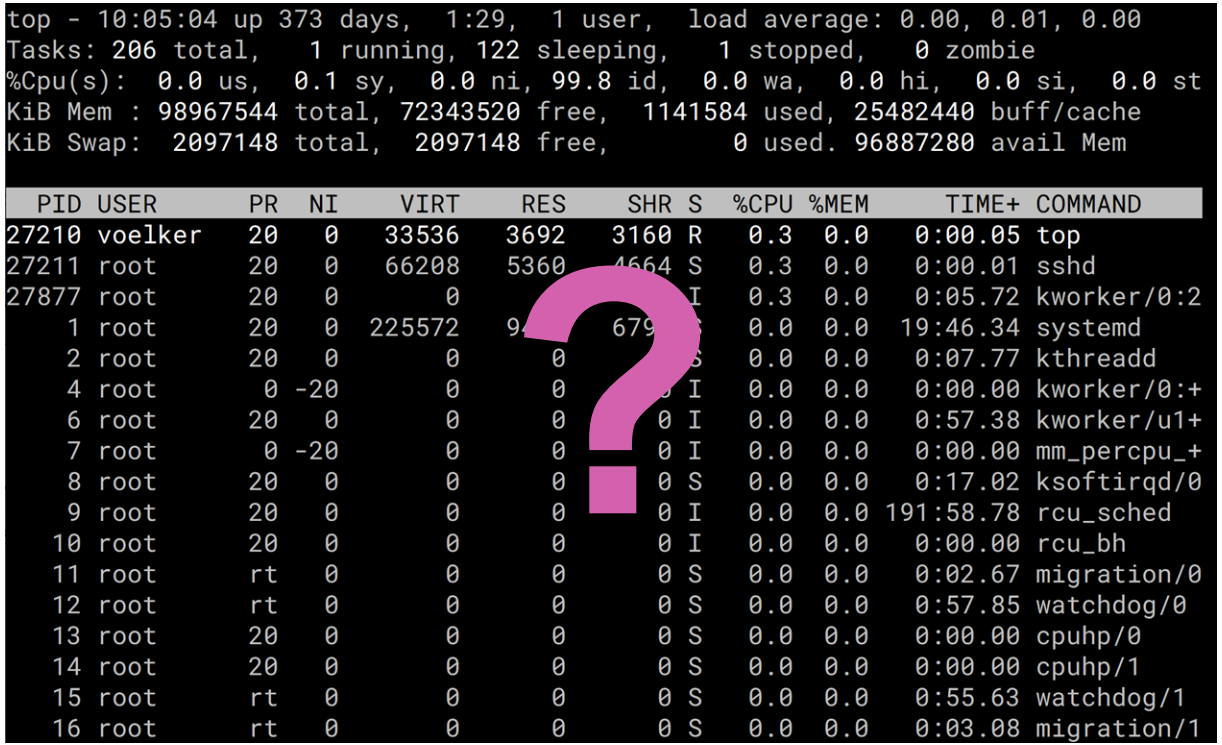

6:32 AM
Christopher Kyle Horton
Search Applications
Filter results
Recently Used
GIMP Image Editor
Update Manager
LibreOffice Impress
Ubuntu Tweak
Empathy Internet Messaging
Image Viewer
Installed
See fewer results
AbiWord
Activity Journal
Additional Drivers
Advanced Settings
AisleRiot Solitaire
Alarm Clock
Appearance
Archive Manager
Armagetron Advanced
Backup
Bazaar Explorer
Bluetooth
OFF
Top left corner
Reveal sensitivity
Low
High
Restore Default Behaviours

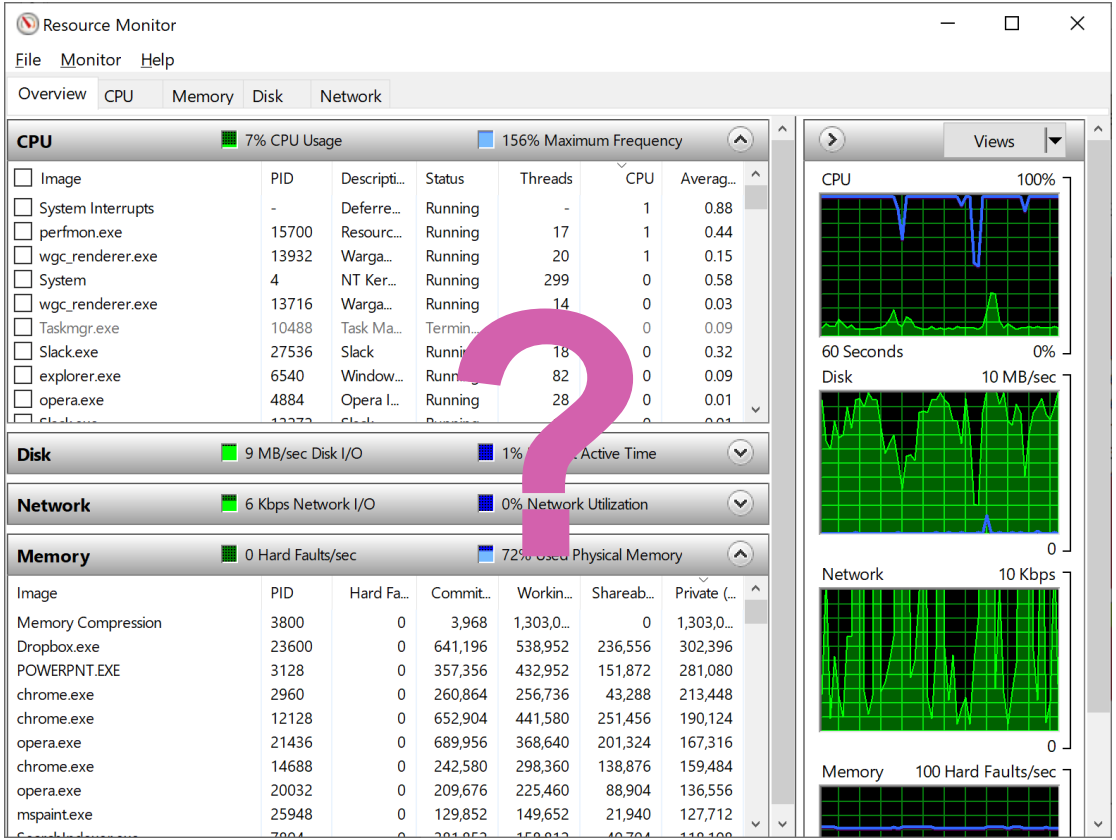

5:34
LTE
Twitter
Relay
Crossword
Messenger
Spotify


Google I/O Keynote in 30min
10:00 AM – 12:00 PM
Wallpaper & style
Change wallpaper
Wallpaper colors
Basic colors
Tue, May 18
09
30
Alarm
06:30 PM
Tue Wed Thu
Pick up kids
03:30 PM
Mon Tue Wed Thu
Google I/O Keynote in 30min
10:00 AM – 12:00 PM
Timer
Pasta
4:41
+1:00
Preview
Permission usage
24 hours
Location
Used by 7 apps
Camera
Used by 4 apps
Microphone
Used by 5 apps
See other permissions
Files and media, contacts, and 3 more
Google I/O Keynote in 30min
10:00 AM – 12:00 PM
Natalie Romano
Tues, May 18
72°F
NOV
8
12.8k
09:30
Alarm will ring at 9:40 AM
Dismiss alarm
Floor lamp
Tues, May 18
72°F
DEG
25
AC
Tue, May 18
09:30
09:30
09:30
09:30
09:30
Snooze
Dismiss
May
Edit
Undo
Save copy
Tue, May 18
72°F
09
30

Hardware

https://www.servethehome.com/supermicro-sys-222h-tn-review-2u-intel-xeon-6-server-daputstor/


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 S %CPU %MEM TIME+ COMMAND
27210 voelker 20 0 33536 3692 3160 R 0.3 0.0 0:00.05 top
27211 root 20 0 66208 5360 4664 S 0.3 0.0 0:00.01 sshd
27877 root 20 0 0 0 0 I 0.3 0.0 0:05.72 kworker/0:2
1 root 20 0 225572 9 679 S 0.0 0.0 19:46.34 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:07.77 kthreadd
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:+
6 root 20 0 0 0 0 I 0.0 0.0 0:57.38 kworker/u1+
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_+
8 root 20 0 0 0 0 S 0.0 0.0 0:17.02 ksoftirqd/0
9 root 20 0 0 0 0 I 0.0 0.0 191:58.78 rcu_sched
10 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_bh
11 root rt 0 0 0 0 S 0.0 0.0 0:02.67 migration/0
12 root rt 0 0 0 0 S 0.0 0.0 0:57.85 watchdog/0
13 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
15 root rt 0 0 0 0 S 0.0 0.0 0:55.63 watchdog/1
16 root rt 0 0 0 0 S 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 Threads CPU Averag...
System Interrupts - Deferre... Running - 1 0.88
perfmon.exe 15700 Resourc... Running 17 1 0.44
wgc_renderer.exe 13932 Warga... Running 20 1 0.15
System 4 NT Ker... Running 299 0 0.58
wgc_renderer.exe 13716 Warga... Running 14 0 0.03
Taskmgr.exe 10488 Task Ma... Termin... 0 0.09
Slack.exe 27536 Slack Runni 18 0 0.32
explorer.exe 6540 Window... Runn 82 0 0.09
opera.exe 4884 Opera I... Running 28 0 0.01
Disk 9 MB/sec Disk I/O 1% Active Time
Network 6 Kbps Network I/O 0% Network Utilization
Memory 0 Hard Faults/sec 72% Used Physical Memory
Image PID Hard Fa... Commit... Workin... Shareab... Private (...
Memory Compression 3800 0 3,968 1,303,0... 0 1,303,0...
Dropbox.exe 23600 0 641,196 538,952 236,556 302,396
POWERPNT.EXE 3128 0 357,356 432,952 151,872 281,080
chrome.exe 2960 0 260,864 256,736 43,288 213,448
chrome.exe 12128 0 652,904 441,580 251,456 190,124
opera.exe 21436 0 689,956 368,640 201,324 167,316
chrome.exe 14688 0 242,580 298,360 138,876 159,484
opera.exe 20032 0 209,676 225,460 88,904 136,556
mspaint.exe 25948 0 129,852 149,652 21,940 127,712
Views
CPU 100%
60 Seconds 0%
Disk 10 MB/sec
0
Network 10 Kbps
0
Memory 100 Hard Faults/sec

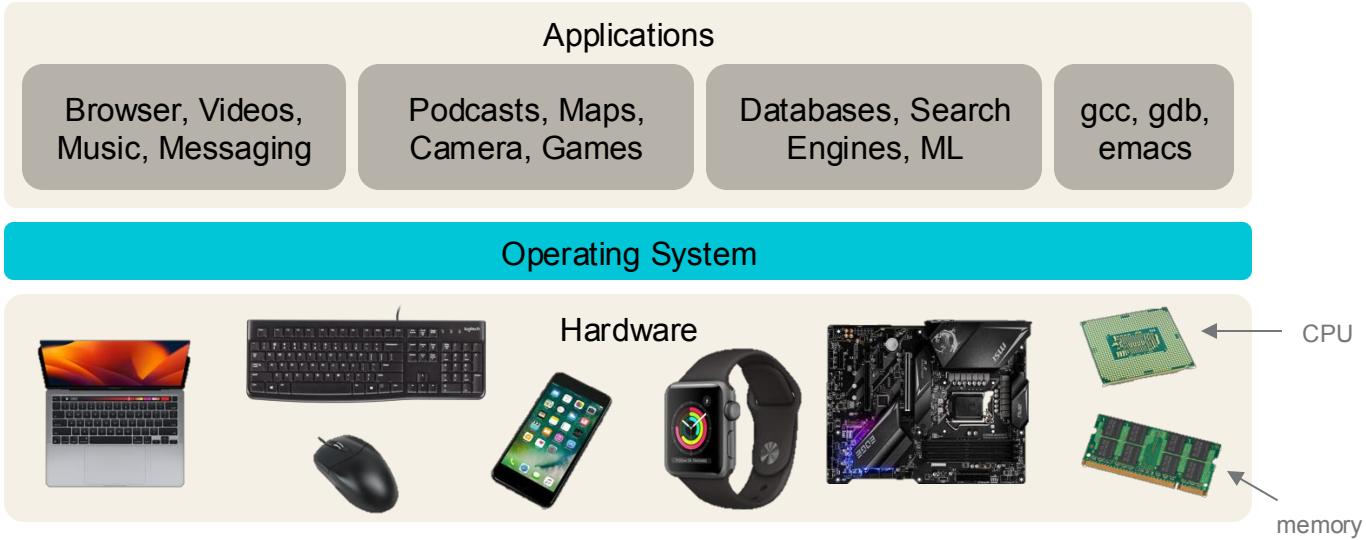
Fundamental OS Concepts

- 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…)

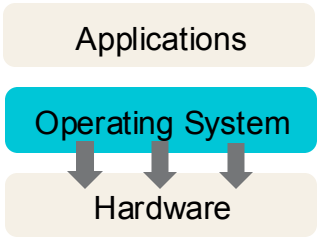
What is an Operating System?



- Code that sits between applications and hardware
- Provides abstractions to layers above
- Implements abstractions for and manages resources below

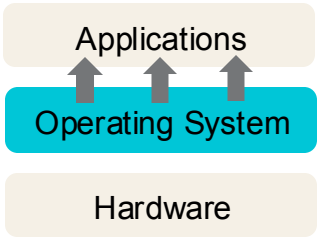
The OS and Hardware

- The OS controls and mediates access to hardware resources
  - Computation (CPUs)
  - Volatile storage (memory) and persistent storage (disk, etc.)
  - Communication (network, etc.)
  - Devices (keyboard, camera, monitor, etc.)
- OS operations
  - Resource allocation
  - Resource reclamation
  - Protection – between and from applications



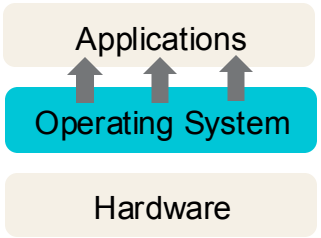
The OS and Applications

- The OS provides abstractions to applications
- The OS defines a set of logical resources (objects) and a set of well-defined operations on those objects (interfaces)
  - Files – create, read, write
  - Threads – create, yield, exit
- Many benefits over dealing with hardware directly
  - Hides the complexity of hardware
  - Allows hardware to evolve independently
  - Provides the illusion of “infinite memory”
    or “sole application running”



The OS and Applications

- Users and programs can safely coexist, cooperate, and share resources
  - Concurrent execution of multiple applications (time sharing)
  - Communication among multiple applications (pipes)
  - Share common services
    - » No need to implement your own file system

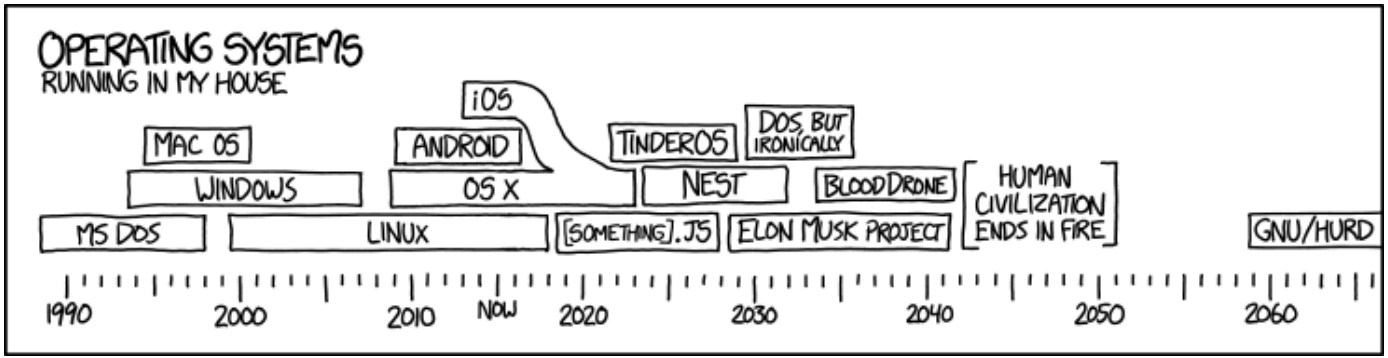


What is part of an OS? What is not?

- Windowing system?
- Web browser?
- Network stack?
- Memory allocation?
  - E.g., `malloc()` and `free()`

How similar are different OSes?

- Popular OSes today: Windows, Linux, and OS X
  - How different/similar do you think these OSes are?
  - What parts are different/similar?



https://xkcd.com/1508/

Do OSes change over time?

- Core operating systems concepts date back to the 1970s - have these changed?
- Hardware is evolving
  - 60s-70s – mainframes
  - 70s-80s – minicomputers
  - 80s-90s – PCs
  - 90s-00s – laptops
  - Today – smartphones, watches, etc.
- New applications
  - Web-based applications
  - Virtual reality
  - Machine learning
  - …

minicomputers

PCs

smart watches

What properties should an OS provide?

- Efficiency or performance?
- Fairness?
- Portability?
- Security?
- Robustness?
- Goals can conflict
  - Fairness vs. efficiency
  - Security vs. performance

Implications for OS Design

- OSes face constraints and tradeoffs
  - Different objectives for different applications
- OSes must adapt over time due to evolution of:
  - Hardware
  - Applications
- Design principles can guide us
  - Abstraction
  - Modularity
  - Simplicity
  - Caching

Separation of Policy and Mechanism

- Fundamental design principle in computer science
- Mechanism: tool that achieves some effect
- Policy: decision about what effect should be achieved
- CPU scheduling example:
  - Treat all users equally
  - Treat all applications equally
  - Prioritize some applications over others
- Separation leads to flexibility!

About this course…

- Principles of OS design
  - Some theory
  - Some rational
  - Lots of practice
- Goals
  - Understand OS abstractions and design decisions
  - Last piece of the puzzle
- To achieve these goals:
  - Learn concepts in class
  - Hands-on experience in labs

Upcoming Tasks

- Project 0
  - Due Tuesday April 8th at 11:59pm, done individually

- Browse the course website:
  - `https://cseweb.ucsd.edu/classes/sp25/cse120-a/`
- Read Chapters 1, 2, and 6
- Start thinking about partners for project groups

- I will stay after to answer any additional questions