CSE 120
Principles of Operating Systems

Spring 2023

Lecture 1: Course Introduction

Yiying Zhang
Lecture 1 Overview

- Class overview
- Administrative info
Personnel

- Instructor
  - Yiying Zhang

- TAs and Tutors (shared with section B)
  - Kyle Wang (TA)
  - Fengyuan Wu (TA)
  - Steven Wu (TA)
  - Charlotte Tang (TA)
  - Yunxiang Chi (Tutor)
  - Yuke Liu (Tutor)
  - Xiyan Shao (Tutor)

- Lectures: M/W/F 2-2:50pm (PCYNH 106)
- Discussion session: W 5-5:50pm (PCYNH 106) or F 4–4:50pm (Center Hall 214)
About Me
(https://cseweb.ucsd.edu/~yuying/)

Vincent C. Rideout (MS 1940, Caltech)
-- Gerald Estrin (PhD 1951, University of Wisconsin)
---- David Martin (PhD 1966, University of California at Los Angeles)
----- David Patterson (PhD 1976, University of California at Los Angeles)
------ Remzi and Andrea Arpaci-Dusseau (PhD 1999, University of California Berkeley)
------- Yijing Zhang (PhD 2013, University of Wisconsin)

• Research interests:
  ◆ Operating systems
  ◆ Distributed systems
  ◆ Computer architecture
  ◆ Systems+networking, systems+security, systems+PL, systems+ML
  ◆ Undergraduate research opportunities at my lab (WukLab)!
CSE 120 Class Overview

- Course material taught through class lectures, textbook readings, and handouts
- Lecture slides
  - I will post slides one day before class date
- Course assignments are
  - Homework questions
  - Three large programming projects in groups
  - Midterm and final exams
CSE 120 Class Overview

• Discussion sections
  ♦ TAs will go over projects and answer questions related to projects and homework (mainly projects)

• Discussion forum
  ♦ Piazza

• My office hour hours: Mon noon-1pm (CSE 3124)
  ♦ Main forum for asking questions related to lecture materials

• TA and tutor lab hours: TBD (send emails before then)
  ♦ Main forum for asking questions related to projects
Class Web Page

http://cseweb.ucsd.edu/classes/sp23/cse120-a/

• Serves many roles…
  ♦ Course syllabus and schedule (updated over quarter)
  ♦ Lecture slides
  ♦ Homework handouts
  ♦ Project handouts

• Optional material and supplemental readings
  ♦ Entirely for your interest only
Textbook


**FREE**
Homeworks

- There will be 4 homeworks throughout the quarter
  - Reinforce lecture material
- Homeworks provide practice learning the material
  - 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
"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
    - Will become more clear very soon
  - Programming environment will be Java on Unix (Linux)
  - The projects will require serious time commitments
    - Waiting until the last minute is not a viable option!

- You will do three big projects using Nachos
  - Concurrency and synchronization
  - System calls, processes, multiprogramming
  - Virtual memory

- You will work in groups of 1-4 on the projects
  - Start thinking about partners
Labs

• You will need to connect to ieng6
  ♦ You can also try to set up local environments and run locally (we will only provide limited support/help for that)
  ♦ Note: We will test and grade on ieng6 machines
  ♦ Be sure to test your projects there
    » You will be able to test before the deadline

• Lab hours: main forum to ask questions about projects
  ♦ In person in basement labs
  ♦ Find schedule on course website
  ♦ TAs and tutors will use autograder to order questions (FCFS)
  ♦ They will not debug for you!
Exams

• Midterm
  ♦ **Wed May 3rd** 7pm - 8:50pm (LEDDN AUD 216)
  ♦ Covers first half of class

• Final
  ♦ Thursday **June 14th** 3pm - 5pm (PCYNH 106)
  ♦ Covers second half of class

• Both exams will be open book
  ♦ What’s exactly allowed TBD, e.g., only two pages of sheets, or only print outs, no electronics

• **No makeup exams**
  ♦ Everyone must be able to attend these exam dates
    » Unless absolute dire circumstances
Grading

- Homeworks: 6%
- Midterm: 22%
- Final: 28%
- Projects: 44%
  - Introduction (4%)
  - Threads (32%)
  - Processes and Multiprogramming (32%)
  - Virtual Memory (32%)
How *Not* To Pass CSE 120

- Do not attend lectures
  - Lectures are redundant, the slides are online, and the material is in the book anyway
  - Lecture material is the basis for exams and directly relates to the projects. Exams will cover materials only in lectures (not in slides or textbooks)!
How *Not* To Pass CSE 120

- Wait until the last couple of days to start a project
  - We’ll have to do the crunch anyways, why do it early?
  - The projects cannot be done in the last few days
  - Repeat: The projects cannot be done in the last few days

- Do not do the homework
  - It’s only 6% of the grade, get full credit for turning anything in
  - Concepts seem straightforward...until you apply them
  - Excellent practice for the exams, and some homework problems are exercises for helping with the project
Project 1 Scores

- Mean
- Median

SCORE

START DATE

DAY.0-2
DAY.2-4
DAY.4-6
DAY.6-8
DAY.8-10
DAY.10-12
DAY.12-14
DAY.14-16
How *Not* To Pass Even More

- Do not ask questions in lecture, office/lab hours, or online
  - It’s scary, I don’t want to embarrass myself
  - Asking questions is the best way to clarify lecture material at the time it is being presented
  - Office hours and email will help with homework, projects

- Violate academic integrity
  - It is much better to get a 0 for an assignment than to fail the course for academic integrity violations
Academic Integrity

- Exams
  - Work them on your own!

- Projects
  - Each team must write their own solution
  - No discussion of or sharing of specific code or written answers is allowed
  - Any sources used outside of textbook/handouts/lectures must be explicitly acknowledged
  - Your responsibility to protect your files from
    » e-copying using UNIX file protection
    » public access, including disposal

- We take cheating very seriously, with a zero tolerance policy
  - We will run tools to catch that (at the end of the quarter), do not even attempt!
Diversity and Inclusion

We are committed to fostering a learning environment for this course that supports a diversity of thoughts, perspectives and experiences, and respects your identities (including race, ethnicity, heritage, gender, sex, class, sexuality, religion, ability, age, educational background, etc.). Our goal is to create a diverse and inclusive learning environment where all students feel comfortable and can thrive.

Our instructional staff will make a concerted effort to be welcoming and inclusive to the wide diversity of students in this course. If there is a way we can make you feel more included please let one of the course staff know, either in person, via email/discussion board, or even in a note under the door. Our learning about diverse perspectives and identities is an ongoing process, and we welcome your perspectives and input.

We also expect that you, as a student in this course, will honor and respect your classmates, abiding by the UCSD Principles of Community (https://ucsd.edu/about/principles.html).

Please understand that others’ backgrounds, perspectives and experiences may be different than your own, and help us to build an environment where everyone is respected and feels comfortable. If you experience any sort of harassment or discrimination, please contact the instructor as soon as possible. If you prefer to speak with someone outside of the course, please contact the Office of Prevention of Harassment and Discrimination: https://ophd.ucsd.edu/.
Students with Disabilities

We aim to create an environment in which all students can succeed in this course. If you have a disability, please contact the Office for Students with Disability (OSD), which is located in University Center 202 behind Center Hall, to discuss appropriate accommodations right away. We will work to provide you with the accommodations you need, but you must first provide a current Authorization for Accommodation (AFA) letter issued by the OSD. You are required to present their AFA letters to Faculty (please make arrangements to contact me privately) and to the OSD Liaison in the department in advance so that accommodations may be arranged.
Basic Needs Resources

- Are you eating properly? Do you have adequate access to nutritious food? Do you have stable housing? Are you homeless or couch surfing?
- If you or someone you know has food and/or housing insecurity, please note: http://basicneeds.ucsd.edu
  - The Triton Food Pantry (in the old Student Center), https://www.facebook.com/tritonfoodpantry/, is free and anonymous, and includes produce.
  - Financial aid resources, the possibility of emergency grant funding, and off-campus housing referral resources are available.
  - CAPS and college deans can connect students to the above resources, as well as other community resources and support.
Questions

• Before we start the material, 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!
Why Operating Systems?

• Why are we making you sit here today, having to suffer through a core 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
  ♦ OS is the foundation of virtualization, what cloud runs on
  ♦ System functionality, performance, efficiency, etc.

• Pervasive abstractions
  ♦ Concurrency: Threads and synchronization are common modern programming abstractions (Java, C#, C++, Go, etc.)

• Complex software systems
  ♦ Many of you will go on to work on large software projects
  ♦ OSes serve as examples of complex systems
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
- Can your program allocate more memory than what your machine has physically? Why?

Our goal is to explain those mysteries
What this course is not about

• How to use an OS
• Graphic user interfaces of OS’s
• A particular OS (although we use UNIX/Linux to explain many concepts)
• Different OS kernel architectures or distributed OS’s or virtualization or other more advanced topics (take grad OS for that)
top - 20:48:08 up 275 days,  1 user,  load average: 0.06, 0.07, 0.05
Tasks: 171 total,  1 running,  169 sleeping,  0 stopped,  0 zombie
Cpu(s): 0.1%us, 0.1%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   16467276k total, 1415968k used,  230748k free, 171168k buffers
Swap:  0k total,  0k used,  884340k cached

PID USER   PR NI  VIRT  RES  SHR S  %CPU %MEM    TIME+ COMMAND
14677 voelker   20  0 55548 3232 2364  R   0:00.07 top
24637 voelker   20  0  86300 6364 1024  S   32:06.70 mosh-server
  1 root   20  0 57812 1636  584   S  1:26.73 init
  2 root   20  0  0   0   0   S  0:03.13 kthreadd
  3 root   20  0  0   0   0   S  0:04.38 migration/0
  4 root   20  0  0   0   0   S  0:04.94 ksoftirqd/0
  5 root   20  0  0   0   0   S  0:00.01 watchdog/0
  6 root   20  0  0   0   0   S  0:04.39 migration/1
  7 root   20  0  0   0   0   S  0:11.22 ksoftirqd/1
  8 root   20  0  0   0   0   S  0:00.01 watchdog/1
  9 root   20  0  0   0   0   S  0:18.05 migration/2
 10 root   20  0  0   0   0   S  0:44.37 ksoftirqd/2
 11 root   20  0  0   0   0   S  0:00.01 watchdog/2
 12 root   20  0  0   0   0   S  0:18.06 migration/3
 13 root   20  0  0   0   0   S  0:01.67 ksoftirqd/3
 14 root   20  0  0   0   0   S  0:00.01 watchdog/3
 15 root   20  0  0   0   0   S  2:30.94 events/0
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…)
A Typical Computer from a Hardware Point of View

- CPU
- Chipset
- Memory
- I/O bus
- Network

CSE 120 – Lecture 1 – Course Intro
Computer System Components

- user 1
- user 2
- user 3
- user n

- compiler
- assembler
- text editor
- . . .
- database system

system and application programs

- computer hardware

?
Computer System Components

user 1

user 2

user 3

... user n

compiler

assembler

text editor

... database system

system and application programs

operating system

computer hardware
What is an Operating System?

“Code” that *sits between*:

- programs & hardware
- different programs
- different users
About this course...

Principles of OS design
- Some theory
- Some rational
- Lots of practice

Goals
- Understand OS design decisions
- Last piece of the “puzzle”
- Basis for future learning

To achieve the goals:
- Learn concepts in class
- Get hands “dirty” in labs
Topics we’ll cover

- Process management
- Memory management
- I/O management (file system)
- A touch of advanced topics if we have time
Fundamental OS Issues

• The fundamental issues/questions in this course are:
  ♦ **Structure:** how is an operating system organized?
  ♦ **Sharing:** how are resources shared among users?
  ♦ **Naming:** how are resources named (by users and programs)?
  ♦ **Protection:** how are users/programs protected from each other?
  ♦ **Reliability and fault tolerance:** how to mask failures?
  ♦ **Concurrency:** how to control parallel activities?
  ♦ **Performance:** how to make efficient use of resources, reduce OS overhead?
  ♦ **Scale and growth:** how to handle increased demand?
  ♦ **Compatibility:** can we ever do anything new?
  ♦ **Distribution:** how to coordinate remote operations?

• And the **principles** in this course are the design **methods**, **approaches**, and **solutions** to these issues
Expect (some) pain

Somewhat fast pace

Lots of programming and debugging

Some difficult (abstract) concepts
For next class...

- Browse the course web
  
  https://cseweb.ucsd.edu/classes/sp23/cse120-a/

- Sign up on Piazza!
- Finish quiz #FinAid-Course-Survey on Canvas!
- Read Chapters 1, 2, and 6
- Start finding partners for project groups

- Let the fun begin!