CS 45: Operating Systems
Course Introduction

Kevin Webb
Swarthmore College
January 23, 2024

Please sit towards the front, next to other students!

Instructor: Kevin Webb

- http://www.cs.swarthmore.edu/~kwebb/
- Please call me Kevin (or Professor Webb)
- Research: Control platforms for networks and distributed systems, CS Education

Instructor: Kevin Webb

- http://www.cs.swarthmore.edu/~kwebb/
- Please call me Kevin (or Professor Webb)
- Research: Control platforms for networks and distributed systems, CS Education
- Hobbies:
  Building stuff, cactus/fruit plants, PC games

Inst

- http://www.
- Please call m
- Research: Co
  distributed s
- Hobbies:
  Building stuf

Office Hours

- Monday 2:30 PM – 4:00 PM (Kevin)
- Tuesday 2:00 PM – 3:30 PM (Gautam)
- Wednesday Noon – 1:30 PM (Gautam)
- Wednesday 1:00 PM – 2:30 PM (Kevin)

- By appointment

Resources

- EdSTEM Q&A Forum, GitHub Enterprise
  - https://edstem.org/us/courses/52078/
  - https://github.swarthmore.edu/CS45-s24

- Slides & audio recordings on course website

- Lab sections:

Section	Time	Location	Instructor
Section A	Friday 8:50 AM - 10:20 AM	Clothier 16	Webb
Section B	Thursday 2:45 PM - 4:15 PM	Clothier 16	Webb
Section C	Friday 2:00 PM - 3:30 PM	SCI 256	Mohan

Contact Policy

- For lab/content questions, use please EdSTEM.
  - If no solutions, please post publicly!
    - You can post anonymous to class.
  - If you're posting non-trivial code, post privately!

- For non-content matters (e.g., accommodations), please send me an email: kwebb@cs.swarthmore.edu

How does this class work?

- This class is designed a bit differently from what you might normally be used to
  - Class will be centered around you
  - Requires your participation

Clickers!

- Lets you vote on questions in real time.
- Like pub trivia, but the subject is always OS.
- You NEED one of these for the course!

Locating your Clicker ID

Will only have numbers 0-9 and letters A – F.

I DO NOT WANT THE FCC ID!

Frequency AA!

Peer Instruction

- ~~Short quiz at the beginning of class~~
- During class: pose carefully designed questions
  - Solo vote: Think for yourself and select answer
  - Discuss: Analyze problem in teams of 3
    - Practice analyzing, talking about challenging concepts
    - Reach consensus
    - If you have questions, raise your hand and I'll come over
  - Group vote: Everyone in group votes
    - You must all vote the same to get your point
  - Class wide discussion:
    - Led by YOU (students) – tell us what you talked about in discussion that everyone should know!

Why Peer Instruction?

- You get a chance to think.
- I get feedback as to what you understand.
- It’s less boring!
- Research shows it promotes more learning than traditional lecture.

Giving out Candy

- To people willing to
  - Ask a question
  - Share an explanation
  - Summarize what their group talked about

- Your explanations are CRITICAL for fellow students' learning

Example Question

- Individual vote
- Group discussion / group vote
  - Room should be LOUD
- Class discussion

How many of the following…

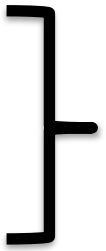
A: 0 B: 1 C: 2 D: 3 E: 4

Grading

- 0% Reading Quizzes
- 7.5% Class & Lab Participation
- 25% Midterm Exam
- 30% Final Exam
- 37.5% Programming Assignments

Grading

- 0% Reading Quizzes
- 7.5% Class & Lab Participation
- 25% Midterm Exam
- 30% Final Exam
- 37.5% Programming Assignments



- I will drop three no-shows. Course website has (CS department standard) extended absence policy.

Approximate Grading Scale

- A+:99+
- A: [93-99)
- A-: [90-93)
- B+: [87-90)
- B: [83-87)
- B-: [80-83)
- C+: [77-80)
- …

Course Policies

- Collaboration
  - You may discuss approaches, not solutions
  - You must submit your own work
  - Exams will include questions on programming
- Cheating
  - Zero tolerance for cheating, don't do it!
- Lab Lateness
  - 2 days of extra (at the granularity of days)
  - Let me know **after** you've submitted

CS Exam Policy

- Exam takers must place all non-essential items at the front of the room (or other designated area). Unless otherwise permitted, students may not have any electronic devices or course materials in their possession during the entirety of the exam. This includes cell phones, tablets, laptops, smart watches, course notes, articles and books, among others. These items should be placed at the front of the room near the proctor. If you need to leave the room during the exam, you must obtain permission from an instructor first. Any non-permitted discussion or aide in regards to exam material will result in immediate forfeiture of the exam and a report to the College Judiciary Committee. Please discuss any concerns or accommodations with your instructor prior to starting the exam.

Tentative Schedule

- Midterm – March 7, in class
- Final - TBD
- Labs
  - Released on Thursdays (lab section)
  - Due on Wednesdays
  - Multi-week labs. Don't wait until the 2nd week to start it…

Labs

- This will be a lab-heavy course with lots of C programming! When I say “start early”, I mean work a little and then let it stew…

- ***“Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.” -Kernighan’s Law***

- Course content will not always match labs. Sorry! This is to give you more time on the labs…

Lab 0

- Lab 1 will build upon CS 31 shell.
- You should (re)familiarize yourself with your old shell lab.
- See: https://www.cs.swarthmore.edu/~kwebb/cs45/s24/labs/lab0.html

Lab on March 8

- Yes, we're going to meet for lab on March 8.
- We need all the lab meetings we can get!

Lab Checkpoints

- I want to hear from ***you.***
- One perspective: “having lab checkpoints (~one-week into a two-week lab) gives me a concrete deadline to work toward and ensures that I make progress”
- Another perspective: “it’s my time, and I’ll manage it best if I have the flexibility to determine my own priorities”

Lab Checkpoints

- Here's what I've tried:
  A. No checkpoints, the lab is due at the final due date.
  B. Enforced checkpoints, for credit.
  C. Checkpoints are recommended, but not required. If you meet them, you can get credit back towards points you miss in grading.
  D. Something else, ideas?

Think about this a bit more, and I'll post a poll on EdSTEM soon.

Lab Attendance

- If you can't attend a lab, please:
  - Let me know ASAP
  - Attend one of the other lab sections if possible (let me know in advance)

- If you're missing more than 2 lab meetings (without a good reason, e.g., concussion, family emergency, etc.), we have a problem.

Administrative Questions?

- All of this info (should be) on class website
- Feel free to ask on EdSTEM discussion board

Your TODO List

- Register your clicker!
  - Link available on EdSTEM
- Please let me know ASAP about:
  - Your preferred name/pronouns, if different than roster information
  - Academic accommodations – I need AT LEAST two weeks notice before use (SDS deadline)
- Fill out lab 1 partnership form, if you know who you want to work with.

What *is* an Operating System?

- Textbook: "A program that … provides a basis for application programs and acts as an intermediary between the computer user and computer hardware."

- Wikipedia: "System software that manages computer hardware and software resources and provides common services for computer programs."

What *is* an Operating System?

- Textbook: "A program that … provides a basis for application programs and acts as an intermediary between the computer user and computer hardware."

- Wikipedia: "System software that manages computer hardware and software resources and provides common services for computer programs."

What *is* an Operating System?

- Textbook: "A program that … provides a basis for application programs and acts as an intermediary between the computer user and computer hardware."

- Wikipedia: "System software that manages computer hardware and software resources and provides common services for computer programs."

What *is* an Operating System?

For this course, “Operating System” refers to *just the OS Kernel.*

That is, just the low-level control software, not all userspace utilities!

What this Course is NOT

- Deep dive into details of one particular OS
  - We will use / modify Linux in the lab though!
- System administration / configuration

Major Course Themes

1. Abstractions (Hardware and Software)

Hiding complex details of reality behind a simpler illusion or interface.

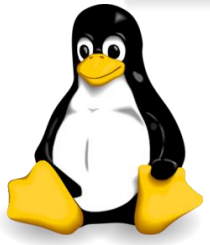
Abstraction

User / Programmer
Wants low complexity

Applications
Specific functionality

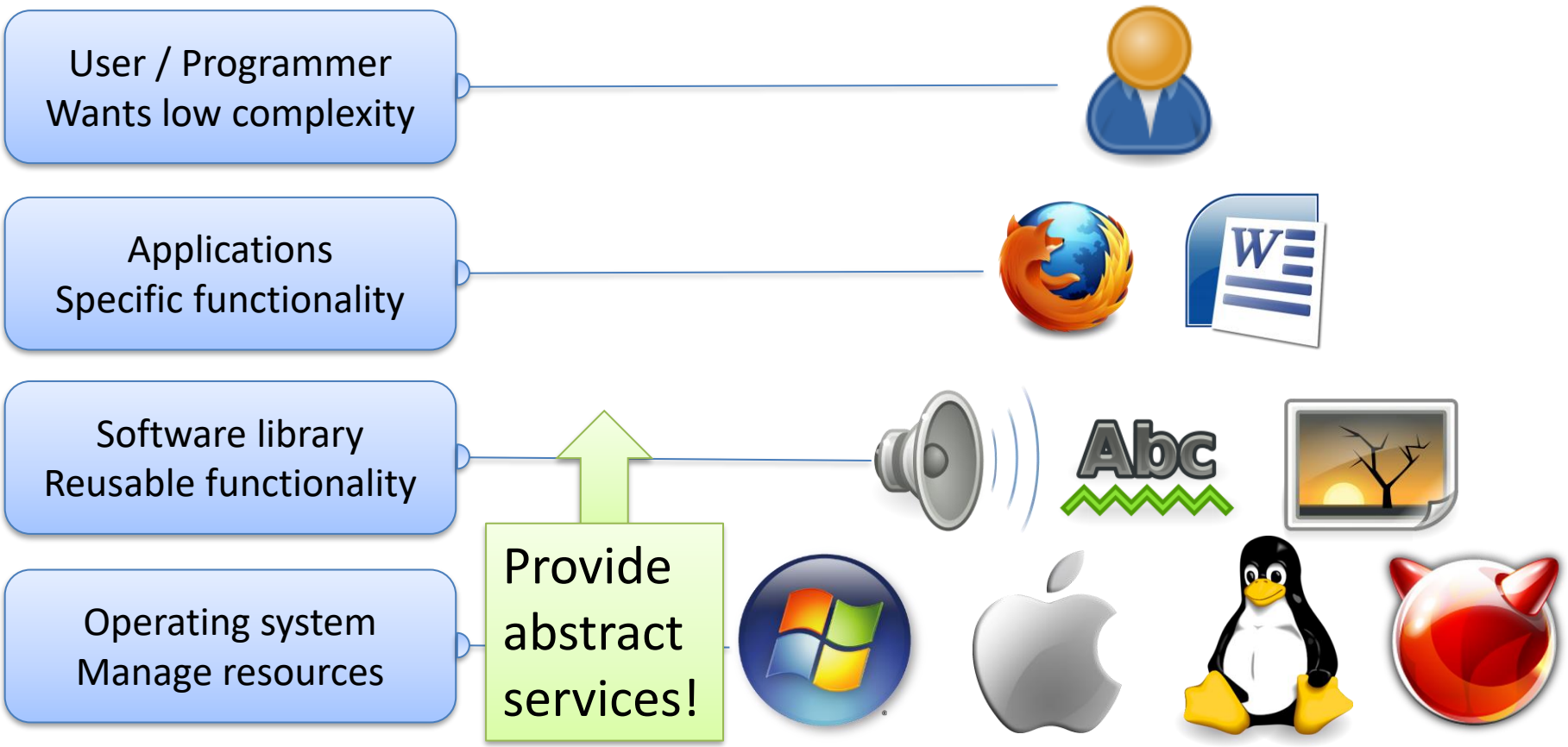
Software library
Reusable functionality

Operating system
Manage resources



Complex devices
Compute & I/O

Abstraction



Hide the details of complex hardware from the software above.

OS Abstractions

- Primary: Process (program in execution)
- Processes need resources…
  - Threads of execution
  - Virtual memory address space
  - Files (and file-like things: sockets, pipes, devices)
- Interfaces for user control
  - e.g., open/read/write/close for files

Major Course Themes

1. Abstractions (Hardware and Software)
2. Hardware Gatekeeping and Protection

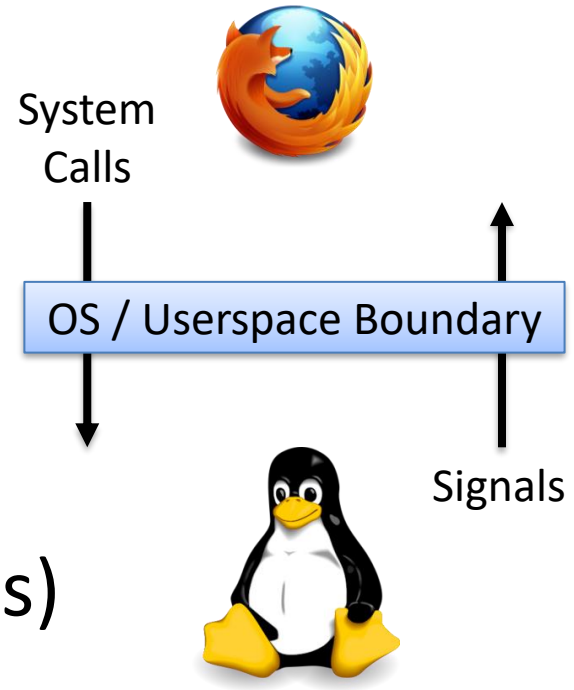
OS enforces control over which processes/users can access resources.

Gatekeeping and Protection

- Clearly defined user process < -- > OS interface

- On any system call,
  verify permission and
  validate parameters

- Potentially limit access
  to resources (e.g., memory, files)



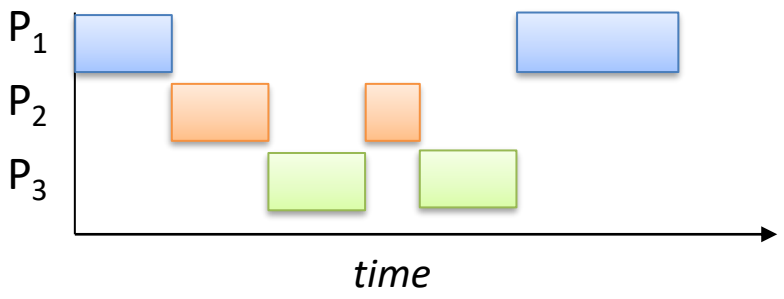
Major Course Themes

1. Abstractions (Hardware and Software)
2. Hardware Gatekeeping and Protection
3. Resource Sharing and Multiplexing

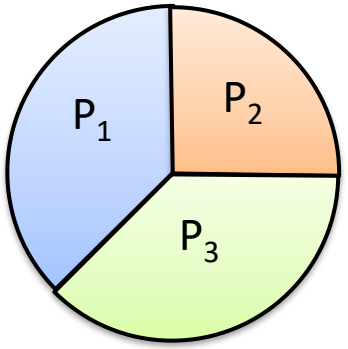
Multiple processes share the machine.
How much CPU time should they get?
Memory? Can/should one be prioritized?

Resource Sharing

Processing: Time



Memory: Space



Reality

- Multiple processes
- Small number of CPUs
- Finite memory

Abstraction

- Process is all alone
- Process is always running
- Process has all the memory

Major Course Themes

1. Abstractions (Hardware and Software)
2. Hardware Gatekeeping and Protection
3. Resource Sharing and Multiplexing
4. Design Decisions and Tradeoffs

The “best” solution typically depends on the scenario and expected use case.

Design Tradeoffs

- Design decisions for super computer are very different from mobile phones…
- This course: mainly focus on general-purpose computing (desktops, laptops), especially in lab assignments.
- During discussions: consider alternative platforms (embedded devices, real-time devices, super computers, etc.).

Major Course Themes

1. Abstractions (Hardware and Software)
2. Hardware Gatekeeping and Protection
3. Resource Sharing and Multiplexing
4. Design Decisions and Tradeoffs
5. Mechanism vs. Policy

What we can do (and how) as opposed to what we should do (governance).

Policy vs. Mechanism (by example)

- Mechanism: processes can request memory, OS can validate and choose to satisfy request or not.

- Policy: admins get as much memory as they want, regular users are limited to X GB

Why this material is important…

- Systems is core to the business model of many HUGE employers… Google, MS, Amazon, etc.

- To be the best programmer you can be, you MUST understand what's happening at the next layer down!

Reminder: Your TODO List

- Register your clicker!
- Please let me know ASAP about:
  - Your preferred name/pronouns, if different than roster information
  - Academic accommodations – I need AT LEAST two weeks notice before use (SDS deadline)
- Fill out lab 1 partnership form