

CS439: Principles of Computer Systems

Dr. Alison N. Norman
Department of Computer Science
The University of Texas at Austin
Fall 2026

Who am I?

- Education
 - Undergrad in CS from Georgia Tech
 - MS and Ph.D. in CS from UT Austin
- Teaching: CS439, CS109, CS398T [CS103F]
- Service
 - ABCS Faculty Advisor
 - Lab director for Summer Academy for Women
 - Associate Chair for Undergraduate Education
- Research: Supercomputing
- Addressing me:
 - Alison or Dr. Norman
 - she/her/hers

Today's Plan

- Consider what we learned in the videos for today
 - Interactive review (you participate!)
 - Recall and organize your thoughts
- Course organization and logistics

What We Learned

Operating Systems are:

- Referees:
 - Manage shared resources
 - Provide protection and communication for processes
- Illusionists:
 - Provide the illusion of infinite resources
- Glue:
 - Provide standard services which the hardware implements

Vocabulary

performance

referee

memory access model

overhead

virtualization

abstraction

response time

fairness

policy

security

resources

communication

portability

three interfaces

HAL

predictability

software

hardware

reliability

AMI

protection

throughput

availability

evaluation criteria

illusionist

isolation

efficiency

standard interfaces

mechanism

glue

API

legally executable instructions

Course Goals

Why are we here?

Two main technical goals:

- Learn the low-level software abstractions that make the computer work
 - Operating System
 - Network
 - Various aspects of memory management
- Use these topics as a case study to understand large-scale system design

What You'll Learn in this Course

1. How to approach problems

- Fundamental issues
- Design space
- Manage complexity
- Case studies

Goal: You will be able to devise good solutions to similar (and very different) problems.

What You'll Learn in this Course

2. Specific techniques you should be able to apply to other problems
 - Time-tested solutions to hard problems
 - Goal: be a good engineer
3. Details(ish) of modern operating systems
 - Lots of material, changes quickly
 - Not a priority of this class
 - Would rather you know the abstractions so that you can apply your knowledge to the next OS

What You'll Learn in this Course

4. “Soft” Skills

- Work Collaboratively
- Take responsibility for your own learning
- Consider new ideas and perspectives
- Develop confidence, resilience, and persistence

Class Information

Schedule Overview

- Introduction
- Concurrency and Synchronization
- Memory Management
- File Systems
- Networked Systems
- Parallel and Distributed Computing (briefly)
- Security (briefly)

Course Materials

- Website and Canvas: Go-to places for information
 - Website: syllabus, schedule, pre-class videos, slides, useful links, projects, problem sets, useful links
 - Canvas: grades, group formation, and assignment submission
 - Please ignore Canvas's calculation of your grades
- Ed Discussion: discussion board
 - Integrated into Canvas
- Textbooks:
 - Operating Systems: Three Easy Pieces by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (online, follow links on syllabus or schedule)
 - Operating Systems and Middleware: Supporting Controlled Interaction by Max Hailperin (online, follow links on syllabus or schedule)

What knowledge you need to begin this course

- Prerequisites: CS429(H) with a grade of at least a C-
- Solid basic understanding of hardware
- Solid programming skills (especially in C)

You must understand the components to understand the implications of how they interact!

Expectations

Each Class (General Rhythm)

- Pre-class videos on the day's topics along with some check-your-understanding assessments (1 hour or so)
- Synchronous discussion and learning about those ideas (2 hours)
- We'll be using small groups for discussion
 - Your talking is essential to your learning in the class
 - Try to engage and venture answers---don't let your classmates do all the work
 - If you find that you always answer, try to take a step back and wait for others to respond
 - Groups will be randomly selected to present during and/or at the conclusion of each activity

Weekly Discussion Sections

- Review any content confusions and the problem set
- Review design docs
- In groups!
- Work a problem
- Lots more talking. By you. :)

Expected Effort

- This is a hard course that requires a lot of effort
- Topics are new and detailed
- There are many design tradeoffs to understand
- Systems programming is hard
 - Debugging systems code is worse
- Projects can take 10-15 hours in the beginning and 30-40 hours later in the semester
 - *If* it goes well
 - Start early
 - Take breaks
 - Sleep

Professionalism

- Attend class
- Arrive on time
- Minimize disruptions
- Respect

Follow our Code of Conduct.

Whole-Person Care

Whole-Person Care

- Personal pronouns
- Mental health
- Religious holy days
- Students with disabilities
- Children in class/on Zoom
- Outside responsibilities
- Emergency situations
- Harassment

Evaluation

Exams

- Two exams and a final
 - Exams dates are determined and on the schedule (mark your calendars!)
 - Final is as scheduled by the registrar
 - Will NOT be at time currently listed
- Two types of questions:
 - Core topic questions, which you must master at some level to pass the course
 - Systems exam questions, which you must show enough mastery across all of them to pass the course

Participation

- Essential to learning the material deeply
- To earn full credit for a class:
 - Get the asynchronous credit: Watch the pre-class videos *before* class day and complete the formative assessments
 - Get the synchronous: Be there, sign in, display your name tent, participate in class, and turn in any in-class assignments. Stay off electronics.
- To earn full credit for section:
 - Arrive on time, stay until the end
 - Arrive with the requested materials already completed (problem sets, design docs, etc)
 - Participate in discussion
 - Be ready to present an answer whenever selected by the TA

Projects

- There will be 5 projects in this course
 - Each includes implementation and a design document
 - Pair or group programming is required (read guidelines!)
 - They are not equally weighted
- They will be difficult, as you are implementing while learning
 - New concepts are always difficult! The better you know the concepts, the easier the implementation will be.
 - Systems programming is difficult---there will be previously un-seen types of bugs, so you will be learning new debugging techniques as well as sharpening the skills you have.

Pair or Group Programming

- Program together 90% of the time
 - That 10% allows for cases when the bugs won't let you sleep---each person is limited to no more than 5%
- Switch drivers every 20-30 minutes
 - Everyone drives an equal amount
 - Only driver adds code when they are driving
- Use version control
 - Must use departmental GitLab instance
 - Driver must commit all changes each time they finish driving
- Be respectful
 - Arrive prepared and on time
 - Pay attention when someone is talking
 - Pay attention when someone is coding
 - Maintain personal space
- We will require weekly reflections on what is going well in your group and what is not
- If a member of your group is unresponsive, please let us know ASAP

WHY Pair or Group Programming?

- More effective and time efficient way to learn the material
- Research shows students perform better in a class with pair programming than without
 - And also in subsequent classes with or without it!
- Also, these projects are hard, and the trouble is in the design. The next trouble is in translating the design to code. Everyone should be together for that process---no one should have to do it alone!

Alternative Grading System

- Our goal is that you learn the material.
- Points are not fungible---you must do well in every category.
- For Core Topics, you will have three possible opportunities to demonstrate mastery. Some level of mastery is required for you to pass the class.
- For Systems Exam Questions, you will have one or two opportunities to demonstrate knowledge, but mastery of individual questions is not required.

	Participation and Assignments				Core Topics		
Final Grade	Participation	Project Test Cases	Project Design	Systems Exam Questions	Synchronization	Virtual Memory	File Systems
A	140 or more	33 or more	33 or more	72 or more	4s or 5s in all subtopics	4s or 5s in all subtopics	4s or 5s in all subtopics
B	125-139	25-32	25-32	63-71	3 in a single subtopic and 4s or higher in the others	3 in a single subtopic and 4s or higher in the others	3 in a single subtopic and 4s or higher in the others
C	95-124	18-24	18-24	54-62	3s or higher in all subtopics	3s or higher in all subtopics	3s or higher in all subtopics
D	60-94	12-17	12-17	45-53	2 in a single subtopic and 3s or higher in the others	2 in a single subtopic and 3s or higher in the others	2 in a single subtopic and 3s or higher in the others
Revise/Retake?	No	Project 2 code	No	If core topics are mastered by the exam, systems questions may be attempted during retakes.	Yes. One chance on the exam, retake up to two subtopic questions after each exam. Exact questions eligible for retakes to be determined by course staff.		
How are the numbers in the column determined?	Total of In-class, Module, and Section Participation	Total of Project Test Cases Scores, including weights	Total of Project Design Scores, including weights and modifications due to reflections	Total of Systems Exam Question scores across all three exams and ISP scores	Most recent score on that core topic.		
How many points will be available? / understand there are no guarantees and numbers are subject to change.	approximately 155	approximately 37	approximately 40	approximately 80	Weird question for this category, but let's say 5.		

Collaboration and Cheating

Collaborate!

- Work with your fellow students
- Discuss the problem sets, the projects, and course material
- Discuss early and discuss often

Collaboration and Cheating

- Collaboration
 - Discussing problem sets and programming assignments
 - Discussing possible interpretation of questions, technical details
- Cheating
 - Copying solutions code or programs from someone else, previous semesters' solutions, or public domain
 - Use of AI-powered software, such as ChatGPT or GitHub Copilot, to generate solutions, parts of solutions, explain specifications, etc
 - Participating in discussion group where one person writes solution and everyone else copies it
 - Uploading material to a third-party website---whether and LLM or a more traditional one like CourseHero
 - Penalty for cheating is an academic sanction up to and including an F in the course and a referral to the Dean of Students office

How to Succeed

How to Succeed in This Course

- Keep up
- Attend class
- Do the reading
- Do the projects (and start them early!)
- Ask questions
- Get to know the people in the class
 - How many people you know is the number one indicator of success
 - study groups, problem discussion, etc.
- Sleep, eat
- Work hard, play hard

Julia Evans
@b0rk

Following



some learning skills

some learning skills JULIA EVANS @b0rk

Know what you don't understand hmm how does IPv6 work?	Knowing what you do and don't understand is a skill! it takes a long time to learn!	when you're confused, identify <u>why</u> => (this is hard!)
build a lot of connections ooh, these 2 different things I know are actually related! neat!	fill in the gaps you want to fill in today I will learn IPv6!	remember learning is something you can get better at! when I'm confused, now I can figure out why!

8:41 PM - 26 Jul 2018

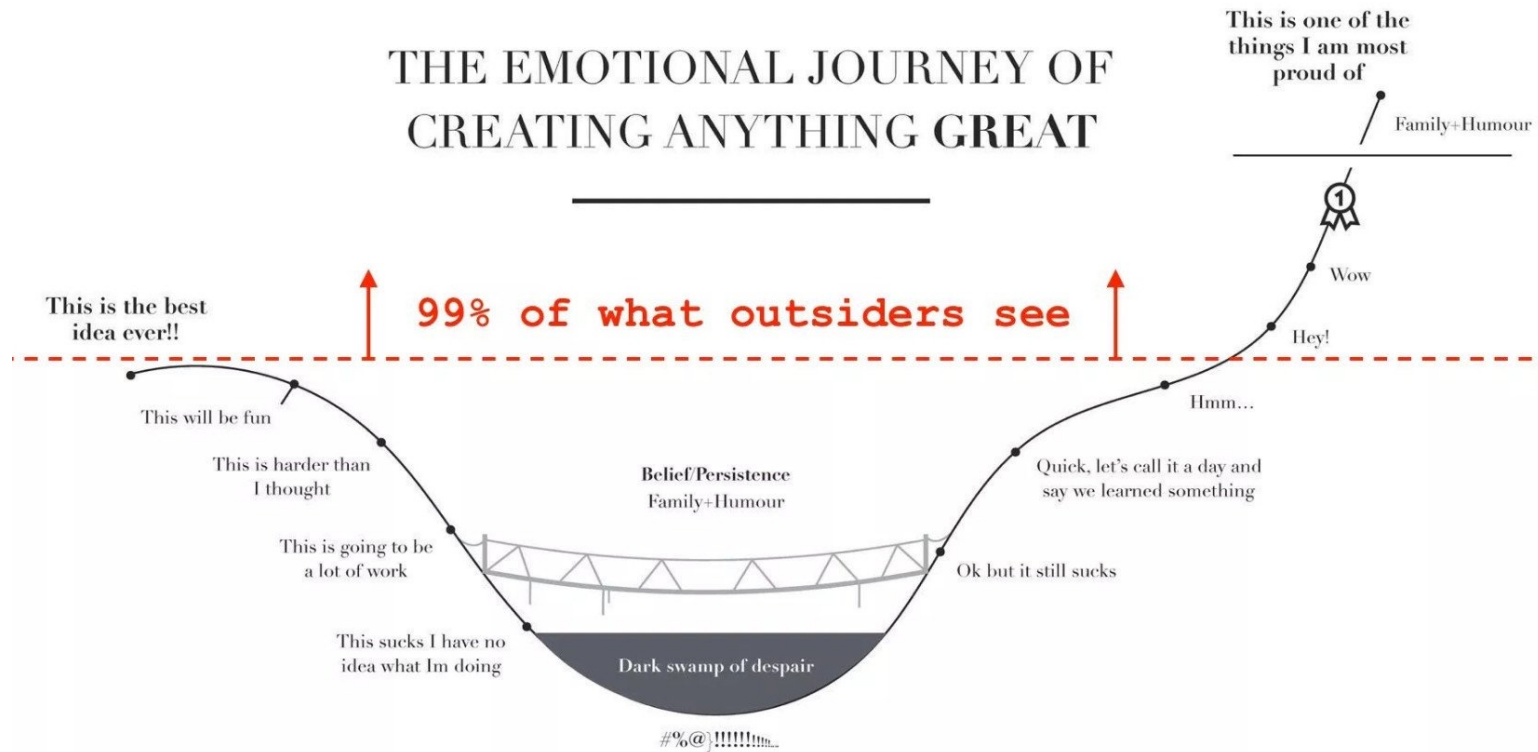
How to Get Help

- Ask questions!
 - In class
 - In office hours
 - On Office Hours webpage (soon)
 - Check that page for days, times, and locations (and starting date!)
 - Discussion board
 - Use Anonymous feature if necessary
- Many resources/tutorials on website and in Canvas modules

Teaching Staff

- Teaching Assistants
 - There are so many! They are fantastic, and they are here to help you.
 - Check the website to see pictures and find office hour locations
- And me!

THE EMOTIONAL JOURNEY OF CREATING ANYTHING GREAT



THE EMOTIONAL JOURNEY IS INEVITABLE AND PERHAPS NECESSARY

<https://john.do/emotional-journey-creating/>

“It’s not that I’m so smart, it’s just that I stay with problems longer.”

-Albert Einstein