CS 332.00 Operating Systems Winter 2024

General Course Info

Time:	4a (MW 12:30pm-1:40pm, F 1:10pm-2:10pm)
Location:	Anderson 323
Website:	https://www.cs.carleton.edu/faculty/tamert/courses/cs332-w24
Slack:	https://join.slack.com/t/carletoncs332-htf3648/signup

Instructor Info

Name:	Dr. Tanya Amert (tamert@carleton.edu)
Web:	https://www.cs.carleton.edu/faculty/tamert
Office:	Olin 301D
Student Hours:	M 1:50pm-3:00pm (5a), W 3:10pm-4:20pm (6a), Th 1:30pm-3:15pm, F 2:20pm-4:00pm (5a+part of 6a), or by appt. Check the course website for locations.

Textbooks and Resources

Required Textbook: We'll use *Operating Systems: Three Easy Pieces*, by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. This is a free online textbook: https://pages.cs.wisc.edu/~remzi/OSTEP/.

Additional readings will be posted to the Moodle site; you will also find the gradebook and classwork exercises on Moodle.

We will use https://www.cs.carleton.edu/faculty/tamert/courses/cs111-w24 as our main course website; the schedule will be posted there, along with links to readings. Assignments will be acquired from a git repo and submitted to Gradescope.

We are using Slack for questions and announcements. You can ask questions about any of the course material on Slack, but you must refrain from posting solutions to the assignments.

Course Description

If you're working in the lab, you might be editing a file while waiting for a program to compile. Meanwhile, the on-screen clock ticks, a program keeps watch for incoming e-mail, and other users can log onto your machine from elsewhere in the network. Not only that, but if you write a program that reads from a file on the hard drive, you are not expected to concern yourself with turning on the drive's motor or moving the read/write arms to the proper location over the disk's surface. Coordinating all this hardware and software is the job of the operating system. In this course we will study the fundamentals of operating system design, including the operating system kernel, scheduling and concurrency, memory management, and file systems.

Prerequisites

Computer Science 201 and Computer Science 208, or instructor permission.

Target Audience

This course surveys the nuts and bolts that are needed to build an operating system that can execute multiple processes seemingly concurrently. We will build upon the data structures introduced in CS 201 and draw heavily upon the data-organization concepts from CS 208. In addition, the lab assignments reinforce the concepts through further design and implementation.

Given that CS 208 is a prerequisite, students are expected to have a working knowledge of C.

Goals and Key Learning Objectives

By the end of this course, you should be comfortable:

- efficiently exploring a large, complex codebase written by other people;
- applying principles of isolation and privileged execution to design robust and secure software;
- identifying situations in which concurrent execution poses problems, such as race conditions and deadlock;
- appropriately using synchronization primitives, like locks and condition variables, to resolve these problems;
- explaining the role of virtualization in sharing resources and providing useful illusions of isolation; and
- describing in detail how threads provide parallelism and concurrency.

Key Dates

First day of class: Wednesday, January 3rd
Last day of class: Friday, March 8th

No class: Monday, February 5th (Midterm Break)

Exam #1: Week 3/4 (TBD)
Exam #2: Week 6/7/8 (TBD)
Final exam: Wednesday, March 13th, 8:30 a.m. – 11:00 a.m.

Technology Requirements

In this course, we will make use of the CS departmental server `mantis`. You will need to be able to connect to this server; it is easiest to do so from your own laptop. If you do not have your own laptop, or your laptop ceases to work, let me know.

Course Components and Grading

Course Components

Lessons: Class will usually be a mixture of lecture and small-group exercises. Students are expected to have completed a reading assignment prior to each lesson; in class, we may focus on some of the more complex topics from the readings.

Classwork: Many class periods will include graded in-class exercises ("classwork") designed to gauge pacing and general understanding of the material. This work will be done on paper, and usually begun on your own and completed with an optional partner. You need not prepare outside of class aside from doing the assigned readings. Classwork can be discussed in office hours to make up missed questions.

Lab Assignments: The lab assignments in this course are where you will work through many of the key concepts and really find yourself learning. There will be one short introductory lab and five more substantial labs. Early lab assignments will have designs given to you; for later labs, you will be evaluated both on your design and your feedback on others' designs, as well as on your implementation.

Final Project: For a final project, students will complete an open-ended extension to the lab operating system based on their own designs. Students will implement their designs and perform experiments to validate their work, possibly comparing to other approaches. In addition to submitting code, students will complete a project writeup and a presentation near the end of the term.

Exams: There will be three 60-minute exams covering the material in this course. The first two cover 3-4 weeks each; the third is only available during the final exam. More details will be determined during the first week of the term. All exams will be pencil-and-paper and done in class. The availability of note sheets will vary between exams—more information will be announced closer to exam dates.

Grading Criteria

Assessment	Count	Percentage
Classwork	~20	5%
Lab Assignments: Implementation	5.5	30%
Lab Assignments: Designs	4	10%
Project	1	16%
Exams	3	39%

Lab assignments will be graded both for correctness and style. Good style means having well-documented (through comments and/or good variable/function names) and easily readable code. For designs, it means having clearly communicated ideas, free of typos and grammatical errors. For design feedback, it means giving constructive and actionable feedback.

Your grade for each of the first two exams is the maximum of your original exam grade (from the in-class exam) and the corresponding (optional) part of the final exam. The third exam occurs only during the final exam period, so your third exam score comes only from that part of the final exam.

Grade Derivation

Course grades will be computed as a weighted average following the above criteria. Course grades will be converted to letters using the common scale of A: ≥ 93, A-: ≥ 90, B+: ≥ 87, etc. These thresholds may be adjusted downward (i.e., to your benefit) for the entire class; this will typically be done based on difficulty/duration of exams.

Late Policy

Each assignment needs to be submitted to Gradescope by the time listed on the assignment webpage. If you do not use a token (explained below), or you do but submit after the updated deadline, late submissions will be penalized 60% within the first 24 hours.[1] Any work submitted more than 24 hours after its deadline will receive no credit.

Each student starts the term with four tokens. These can be used on any lab assignment *implementation* or *design document* for any reason, no explanation necessary. One token provides an automatic 24-hour deadline extension. If working with a partner, both partners must spend a token for an extended deadline.

At most one token can be used on an assignment. To use a token, you should email me by the assignment's deadline (no explanation, just to say that you're using one).

Due to the short turnaround between peer feedback of your design and the submission of your implementation, you will not be able to use tokens on *peer review* submissions.

There is no token use needed to make up missed classwork. (This also means that if you aren't feeling well, please stay home and get some rest!) You can make up classwork until 4pm on the last day of class.

Course Schedule (tentative, and subject to change)

Weeks 1-2	Introduction to operating systems
Weeks 3-5	Concurrency
Weeks 5-7	Address virtualization
Weeks 7-8	Scheduling
Weeks 8-9	File systems
Weeks 9-10	Advanced topics, project presentations

[1] Mathematically, the late penalty will be a cap on the possible score, e.g., an assignment submitted two hours late and graded to have 85% of the points earned will receive $\max(85, 100 - 60) = \max(85,40) = 40$ points.

Week 3/4 (TBD)	Exam 1
Week 6/7/8 (TBD)	Exam 2
Week 11	Exam 3 (part of final exam)

Note that some topics may be skipped or replaced depending on time constraints.

We will not meet for class on Monday February 5th due to Midterm Break.

Course Policies

Attendance Policy

Students are expected to attend all class periods. If you must miss class, you are responsible for checking the schedule to determine what you missed and studying the material on your own. If you know of an absence in advance, check with me for how you may best prepare.

Exams cannot be taken after the day on which they occur. If you know you must miss class, you may be able to take it in advance; discuss with me before your absence. If you're feeling unwell the day of the exam, we can arrange a separate room for you; if you're too sick to take an exam, you will need to rely on taking that portion of the final exam.

Student Hours ("Office Hours")

Student hours, also known as office hours, are a time that you can (and are encouraged to!) stop by to chat with me about anything. This includes questions about material covered in class or the textbook, questions regarding assignments, exam preparation, opinions on WoW expansions, the major in general, board games, or any other topic.

You are welcome to stop by my office during any scheduled office hours; if you have a schedule conflict, send an email and we'll find a different time.

Classroom Etiquette

In class, you are expected to maintain proper etiquette. This includes arriving on time, not having conversations during lecture, and most importantly not having your laptop/phone/etc. out during lecture unless we have lab worktime.

You are expected to take your own notes during class (this can be done on a tablet, but it must behave like a notebook—no web surfing or email checking!); you should not be taking pictures of the board.

Collaboration Policy

For lab assignments, you are allowed to work with up to one other student. You may look at each other's code and help with debugging, but *you are expected to type up all of your code yourself*. You may discuss general approaches with other students (ex: "I used a for loop to iterate through the list."), but *you should never be looking at the code of anyone but your partner for a given lab assignment*.

If you are working with a partner, you need only submit one design document for a given assignment. However, you are still required to make separate implementation submissions, and assumed to have typed up the code separately (even if it ends up very similar).

If you worked with someone else or discussed any part of the assignment, you must each list each other's names in comments at the top of the relevant code files. See the assignment instructions for more details.

Exams and classwork are closed-book, closed-note, and will be taken on paper. Exams are expected to be individual efforts. The availability of note sheets will vary – more information will be provided closer to each exam date.

You are expected to refrain from using any online source to generate the solutions to your assignments, but you can use them for C mechanics.

- Acceptable: Searching for the structure of a for loop in C.
- Not acceptable: Searching for how to sort the lines of a file in C.

Unacceptable sources for help when completing the labs include using search engines (e.g., Google, Bing), web forums, or generative AI (e.g., ChatGPT). You should refrain from using coding tools such as GitHub Copilot to help with your code; you won't have such tools on exams, and this class is your opportunity to learn before you may find yourself in a professional software development environment. I want you to think through the implementation yourself, so that in the future you can more easily discern if the output of such tools is correct.

If you find yourself puzzled by an error message, before considering searching online, ask yourself what the error message means. As always, consider whether a classmate may have encountered this error before; asking on Slack may help others, not just yourself!

Academic Integrity

It is a violation of the academic integrity to use any tool (e.g., ChatGPT or GitHub Copilot) to generate code or text for your assignments, or to submit assignment code or text written by others. Additionally, you should never be in possession of anyone else's assignment code/writeup before the due date, and never look at anyone else's code but your partner's.

If any student is suspected to have violated the academic integrity policy, a report will immediately be made to the Academic Standing Committee, as described at https://apps.carleton.edu/campus/doc/integrity. Ask the instructor if you are unsure about what constitutes acceptable collaboration.

Illness/COVID-19 Policy

Some of the policies stated in this syllabus may need to be modified at times due to COVID-19 or other illnesses. For example, if I have COVID, we may need to have some class periods over Zoom. If you test positive, you should talk to a friend to get notes, and let me know so that we can work around your absences to enable you to still achieve mastery over the course content. We will work together to make this term a success, and you will be informed about any necessary changes as soon as possible.

Inclusion

I strive to create an inclusive and respectful classroom that values diversity. Our individual differences enrich and enhance our understanding of one another and of the world around us. This class welcomes the perspectives of all ethnicities, cultures, gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities.

College Policies

Accommodations for Students with Disabilities

Carleton College is committed to providing equitable access to learning opportunities for all students. The Office of Accessibility Resources (Henry House, 107 Union Street) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, vision, hearing, mobility, or speech impairments), please contact OAR@carleton.edu to arrange a confidential discussion regarding equitable access and reasonable accommodations.

Reporting Sexual Misconduct

Carleton is committed to fostering an environment free of sexual misconduct. Please be aware all Carleton faculty and staff members, with the exception of Chaplains and SHAC staff, are "responsible employees." Responsible employees are required to share any information they have regarding incidents of sexual misconduct with the Title IX Coordinator. Carleton's goal is to ensure campus community members are aware of all the options available and have access to the resources they need. If you have questions, please contact Laura Riehle-Merrill, Carleton's Title IX Coordinator, or visit the Sexual Misconduct Prevention and Response website.

Math Tutoring

The Math Skills Center supports all Carleton students in any mathematics course they are taking in which they are experiencing difficulty, either with the mathematical concepts or with the mathematical tools needed to succeed in the course. Their mission is to "level the playing field" by giving students who enter Carleton without strong mathematics backgrounds the tools they need to succeed here at Carleton.

Quantitative Resource Center

The Quantitative Resource Center (QRC) offers free peer support for students working with numbers in their non-Math/Stats classes. Chat, drop in, or make an appointment with a trained peer tutor for help with graphs, charts, and writing with numbers; Excel, R, and statistical analysis; and math up through Pre-Calculus.

Writing Center

The Writing Center provides a space staffed with peer writing consultants who can work with you during any stage of the writing process (brainstorming to final proofreading). Hours and more information can be found on the writing center website. You can reserve specific times for conferences by using their online appointment system.

Support for Students who use English in Addition to Other Languages

If English is not your primary or home language and you believe you might benefit from working regularly with a writing consultant this term, email Melanie Cashin, Multilingual Writing Coordinator, at mcashin@carleton.edu. She can arrange once- or twice-a-week meetings between you and a specific writing consultant throughout the term.

Disclaimer

The instructor reserves to right to make changes to the syllabus, including homework and project due dates, exam dates, and class cancellations and modality changes, or example due to weather or illness. These changes will be announced as early as possible.