CPS 510 – Operating Systems
Fall 2025

(Last Modified: June 30, 2025)

1 General

Course

Lecture	Tue/Thu 4:40-5:55pm
Location	Gross Hall 103

Instructors

Name	Matthew Lentz
Email	mlentz@cs.duke.edu
Website	https://www.cs.duke.edu/~mlentz
Office Hours	TBD + (Briefly) After Lecture

Graduate TAs

Chenyang Liu chenyang.liu@duke.edu

Undergraduate TAs

Lasal Mapitigama

Resources

Canvas	https://canvas.duke.edu/courses/57549
Gradescope	https://www.gradescope.com/courses/1063075
Ed	https://edstem.org/us/courses/80345/discussion
HotCRP	https://duke-cs510f25.hotcrp.com/

2 Overview

This course will focus on fundamental principles of operating systems. We will explore the various roles of the operating system which include managing and multiplexing available hardware resources, providing higher-level abstractions for applications to use in interacting with the hardware platform, and enforcing isolation and protection for software programs. We will cover topics such as: concurrency, file systems, synchronization, virtualization, inter-process communication, and scheduling. Additionally, we will explore how these concepts relate to modern systems beyond the operating system itself (e.g., distributed systems). The topics discussed in this course are similar to CPS 310; however, this course includes more advanced topics and projects, and incorporates readings and discussions of relevant papers (both recent and classical).

3 Expectations

3.1 Preconditions

The prerequisite for this course is either: 1) you are a graduate student in CS or ECE, or 2) you have completed CPS 210 (Introduction to Computer Systems) or ECE/CPS 250 (Computer Architecture). Therefore, I expect that you already understand the basics of computer architecture, and that you have experience in implementing programming projects in C.

If you feel that you could use a refresher on computer architecture, please refer to "Welcome to the Machine" by Jeff Chase https://users.cs.duke.edu/~chase/systems/arch-overview.pdf.

3.2 Postconditions

After completing this course, we expect you to be able to:

1. Understand the fundamentals of operating systems, including: concurrency, communication, protection, isolation, architecture, storage, and scheduling

2. Implement common primitives (e.g., threads) in a simple operating system (i.e., `xv6` from MIT) while writing programs that use these primitives on a real-world operating system (i.e., `Linux`)

3. Understand various factors that impact performance and resource efficiency as a result of hardware and software running underneath your programs (e.g., virtual memory)

4. Understand how operating systems concepts relate to broader systems (e.g., distributed systems), and how recent challenges/advances are shaping the design and implementation of modern operating systems (e.g., trusted hardware, high-speed I/O)

5. Formulate, implement, and evaluate a significant operating systems project

6. Read and understand operating systems research papers

4 Resources

Textbooks We will be using the OSTEP textbook for this course. For each lecture in the schedule, there will be corresponding readings in the OSTEP book. This is a free, open-source textbook that is available online at the link below:

1. **Operating Systems in Three Easy Pieces (OSTEP)**
   https://pages.cs.wisc.edu/~remzi/OSTEP/

Other resources will be provided to you throughout the semester. If you wish to read through additional perspectives on the course content, you may want to reference one (or both) of the following textbooks:

1. **Operating Systems: Principles and Practice**
   https://ospp.cs.washington.edu/index.html
   (Alternative to the above textbook)

2. **xv6: a simple, Unix-like teaching operating system**
   https://pdos.csail.mit.edu/6.S081/2021/xv6/book-riscv-rev2.pdf
   (May be useful to refer to for labs/projects)

Canvas We will be using Canvas as a general course platform, primarily used for reporting grades.

Gradescope We will be using Gradescope to manage the submission and automatic grading for labs. We will also use it for grading exams.

Ed We will be using Ed to serve as a discussion forum for the course and the primary place for making course announcements.

HotCRP For accessing the readings and posting responses, we will be using the HotCRP online conference tool. Some of you may already be familiar with this tool, since it is used to manage the submission and reviewing process for many of the academic conferences in computer science.

Assistance We want to help you and we encourage you to visit office hours and/or ask questions on Ed as needed. Please be respectful of staff time: start early, seek help early, and do not expect extra time beyond the posted schedule. We expect you to make a good faith effort to solve problems yourself before seeking help; additionally, when asking for help, please explain your thought process and what you have tried so far so that we can better help you. Keep up with the Ed board and check to see if your question is already answered. If a TA refuses to help you, please respect their decision and raise the issue with the instructors if you feel you are treated unfairly.

We list office hours here, but for up-to-date information (e.g., cancellations, rescheduling) please refer to the Ed post that contains the link to a Google Calendar as well as the Zoom link for remote office hours.

5 Grading

Your final grade in the course will be determined by the following percentage allocations:

Type	**%**	**Description**
Exams	40	Midterm (15) and final (25) exams
Project	25	Large-scale project and writeup
Labs	25	User- and kernel-space programming assignments
Responses	5	Writing thoughtful responses to the readings
Participation	5	Actively engaging with in-class and online discussions

Note that attendance is not mandatory; however, we strongly encourage it because there will be a number of interactive elements throughout the course. You are responsible for all material covered and assignments given out during any class that you miss.

5.1 Exams

There will be one midterm exam and one final exam. The midterm exam will be given in-class on TBD The final exam is scheduled for TBD from TBD in TBD. Each exam is cumulative, covering all content from the start of the course up to the exam; however, for the midterm exam, content covered by the class immediately prior to the exam is out of scope.

For the midterm exam, you may create and make use of a single, two-sided sheet of notes using standard letter paper. This sheet of notes may be hand-written or typed. For the final exam, you may have two sheets of notes (same rules apply), as this allows you to re-use your sheet from the midterm and create a new one for the later content of the course. With the exception of the note sheets above, the exam is closed book and closed notes.

5.2 Project

One important aspect of this course is to give you experience in the design, implementation, and evaluation of a large-scale operating systems project. It is your responsibility to pick a problem to work on; however, I will provide some concrete options as well as general directions that you can consider. By the end of the course, you will hand in your implementation, a writeup in a form similar to a workshop paper, and provide a short demonstration to the TAs. You may form your own groups of 3-4 students, and you can use the forum as a way to find others to work with; note that you do *not* have the option to work alone.

The project will be broken down into several stages:

1. **Project Group [Due TBD]:** Email me the list of 3-4 group members for your project. Please feel free to use the class forum to help form groups.
2. **Project Proposal [Due TBD]:** A one page PDF document for proposing your project. It should include the following elements:
    - Problem Definition
    - (Some) Discussion of Proposed Approach

- Evaluation Plan

3. **Project Writeup [Due TBD]:** A 4-6 page PDF document in the form of a workshop paper submission. It should include the following elements, although you have some freedom with respect to the exact organization:

    - Introduction: Motivate the problem, outline your approach, and make your overall contributions clear
    - Design: Describe the problem in more detail, the high-level overview and architecture of your solution, and details about the components of your solution
    - Implementation: Describe the *necessary* details to understand the evaluation (next) given your design (previous)
    - Evaluation: Present and explain results that demonstrate that your solution works

Unless otherwise stated, you should submit all of your deliverables via email. I encourage you, but do not require you, to typeset your documents in LaTeX.

5.3 Labs

There will be six programming lab assignments that are meant to be completed individually. You can find more information on the labs here: https://courses.cs.duke.edu/fall25/compsci510/assignments.html.

5.4 Reading Responses

Each student should individually submit responses to the readings before each class session. *Responses for papers are due by 11:59pm ET the day before the class in which they will be discussed*; for instance, if we will be discussing a paper on Tuesday, please submit the response by 11:59pm ET on Monday. This gives me a chance to read through all of your responses and determine how to focus some of the discussions during class. You will submit you responses via HotCRP, as mentioned in the "Resources" section.

Responses should be roughly two paragraphs for each paper. While there is no strict format for these responses, you can think about how you might describe each paper to a colleague. For instance, you might consider talking about: 1) the problem they are trying to solve, 2) the key insight(s) to address the problem, 3) assumptions and design choices, 4) how well the idea was executed and evaluated, and 5) remaining questions that you have (and your initial thoughts on what the answers might be).

I will drop the lowest reading response grade.

5.5 Late Policy

We expect you to turn in your work by the day and time it is due. Note that if a time is not listed, you should assume the deadline is 11:59pm ET on the day listed. However, for labs ONLY, we allow you to turn in assignments up to 72 hours late with no penalty. This prevents the need to file any STINFs for assignments.

Exceptions to this late policy are allowed only for Dean's Excuses, which you can find more information about here: https://trinity.duke.edu/undergraduate/academic-policies/class-attendance-and-missed-work.

5.6 Regrading Policy

All regrading requests (labs, projects, quizzes, exams) must be submitted within one week of the graded item being returned/available. Please use the regrade request functionality available on Gradescope to issue your request. Requests made later than one week will be denied.

6 Academic Integrity

We expect everyone to uphold the Duke Community Standard, which you can find here: https://studentaffairs.duke.edu/conduct/about-us/duke-community-standard. In particular, this standard is comprised of:

- I will not lie, cheat, or steal in my academic endeavors
- I will conduct myself honorably in all my endeavors
- I will act if the Standard is compromised

Please ask me if you are unsure which actions may (or may not) violate the community standard as part of this course. However, you can find specific collaboration guidelines for different types of coursework below.

6.1 Collaboration Guidelines

Labs You may, and are very much encouraged to, discuss the labs with your fellow classmates. This includes discussing the specification, proposed approaches to solving the lab, working through high-level designs on a whiteboard, or asking questions on the forums. However, each of you is responsible for writing up *entirely your own* implementation for the lab. Do *not* share code, either snippets or solutions as a whole, with one another; this also extends to the forums, unless there is explicit approval from the instructor or one of the TAs. Do not incorporate code you find on the Internet (or from paper archives), from students who took the class previous semesters, or other sources. Note that this includes the use of AI-based tools (e.g., ChatGPT, Github Copilot), which you should avoid using for the purposes of assisting with the labs in this course. The line should be quite clear here, but please reach out if you have any questions regarding this policy.

Project You should only collaborate within your project group.

Reading Responses You are more than welcome to discuss papers with other students; however, I expect you to each author your own responses to the papers. As with labs, you should not use AI-based tools to assist you in writing these responses.

7 Students with Disabilities

Duke University is committed to providing equal access to students with documented disabilities. Students with disabilities may contact the Student Disability Access Office (SDAO) to ensure your access to this course and to the program. There you can engage in a confidential conversation about the process for requesting reasonable accommodations both in the classroom and in clinical settings. Students are encouraged to register with the SDAO as soon as they begin the program. Please note that accommodations are not provided retroactively. More information can be found online at access.duke.edu or by contacting SDAO at 919-668-1267, SDAO@duke.edu.

8 Environment

Interactive discussions are one of the key components that make this type of course useful, especially when we dive into more advanced topics. I want everyone to make sure that they do their best to foster an inclusive environment, since that will enable us to have the richest discussions. In general, please treat teaching staff and other students with kindness and respect both in class and outside of class (e.g., Ed forums). We will disable anonymous posting if we see any threatening or distruptive posts. If you feel uncomfortable for any reason, please let me know.

Let us know if you have concerns we can address regarding your safety or health. Please know that we are concerned for you. We understand that you may be facing negative reactions to stress and pressure, other personal challenges, or just the burdens of managing your life and future. Be mindful of your needs for sleep, exercise, proper food, recreation, social connection, and constructive engagement with your problems. We encourage you to take advantage of Duke resources for wellness and mental health. Ask for help when you need it.

9 Course Evaluations

Please take a moment of your time at the end of the semester to submit a course evaluation. These evaluations are incredibly useful to both us personally as well as to the department as a whole. You can provide your feedback at the following link: http://duke.evaluationkit.com/. Note that if you have suggestions for how we improve the course, feel free to reach out at any time.

10 Modifications

We tried to make this syllabus both correct and complete; however, we reserve the right to modify the contents of the syllabus while the course is underway. We will make sure that any modifications are clearly communicated to you with sufficient advance notice.