CPS 510 – Operating Systems
Fall 2022

(Last Modified: August 3, 2022)

1 General

Course

Time	Tue/Thur 12:00pm - 1:15pm
Location	Gross 103

Instructors

Name	Matthew Lentz
Email	mlentz@cs.duke.edu
Website	https://www.cs.duke.edu/~mlentz
Office	LSRC D314
Office Hours	TBD (or by appointment)

Teaching Assistants (TAs)

Yongji Wu yongji.wu769@duke.edu → TBD at TBD

Resources

Website	https://courses.cs.duke.edu/fall22/compsci510/
Sakai	https://sakai.duke.edu/portal/site/f98ca783-cb13-44ce-bbc9-bb47be871d0d
HotCRP	https://duke-510-f22.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, I expect you to be able to:

1. Understand the fundamentals of operating systems, including concurrency, communication, protection, isolation, architecture, storage, and scheduling
2. Understand how operating systems concepts relate to broader systems (e.g., distributed systems), and how recent challenges/advances shape modern operating systems (e.g., trusted hardware, high-speed I/O)
3. 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`)
4. Formulate, implement, and evaluate a significant operating systems project
5. Read and understand operating systems research papers

4 Resources

Textbooks There are no required textbooks for this course. Papers and other various resources will be provided to you throughout the semester.

That being said, you may want to make use of the following textbooks listed below. I will make note of relevant chapters in the OSTEP book that correspond to topics discussed in lectures on the schedule.

1. **Operating Systems in Three Easy Pieces (OSTEP)**
   https://pages.cs.wisc.edu/~remzi/OSTEP/
2. **Operating Systems: Principles and Practice**
   https://ospp.cs.washington.edu/index.html
   (Alternative to the above textbook)

Sakai We will be using Sakai for course announcements, project submissions, and grades. Project submissions will be handled through Gradescope, which is linked with Sakai.

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.

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 weekly readings
Participation	5	Actively engaging with in-class and online discussions

Note that attendance is not mandatory; however, I strongly encourage it since one of the primary aspects of this class revolves around the discussion component. 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 Saturday December 17th from 2:00-5:00pm in Gross Hall 103. The final exam will be cumulative, covering all content from the start of the course.

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 a project implementation along with a writeup (similar to that of a workshop paper). 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 Team [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 through the Duke Box links available through Sakai. 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/fall22/compsci510/labs.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).

5.5 Late Policy

I expect you to turn in your work by the day and time it is due. Note that if a time is not listed, you can assume the deadline is 11:59pm ET on the day listed. The only exceptions to this are based on the Class Attendance and Missed Work Policy, which you can find here: https://trinity.duke.edu/undergraduate/academic-policies/class-attendance-and-missed-work.

6 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. If you feel uncomfortable for any reason, please let me know.

7 Academic Integrity

I 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.

7.1 Collaboration Guidelines

Labs You are more than welcome 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 your own implementation for the project. 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.

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.

8 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.

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 me personally as well as to the department as a whole. Note that if you have suggestions for how I can improve the course, feel free to reach out at any time.

10 Modifications

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