CMSI 3510: Operating Systems

Loyola Marymount University

Professor Julian Gonzalez

Fall 2025

Course Information

Prerequisites: CMSI 2210 and/or CMSI 284 or EECE 3140

Section 1: Pereira 127
Time: 1:45 PM – 3:25 PM
Course Site: www.jag.prof
Office: Foley Annex 139 **Email:** Julian.Gonzalez.lmu.edu

My office address, office number, and email are listed in the footer of the course website. Email is the preferred contact method. If my office door is open, feel free to stop by.

Course Description

This 4-unit course combines the modern development of Operating Systems with practical application in Rust. Topics include Concurrent and Parallel Programming, Investigations of computer hardware architecture, Scheduling, Memory Management, I/O management, and other concepts relevant to the view of Kernel Systems.

This course is offered in a synchronous, lecture-based format, meeting in person. Recordings of all lectures will be uploaded to the course site for asynchronous access, typically after class. You are encouraged to ask questions at any time during lectures.

Coursework consists of engaging with reading material, programming assignments, mandatory in-class activities, as well as optional creative projects and optional midterm and final assessments for additional credit. Collaboration is encouraged, but all submitted work must be your own original work. If you work with others or use outside sources, cite them appropriately so I can accurately assess each student's understanding.

This 4-unit course requires a minimum of 12 hours of work per week, in accordance with University policy. I encourage students to work consistently in manageable segments ("chiseling") to avoid last-minute stress. Additional details are provided in each assignment.

Required Materials

This course features TWO major required materials and some additional materials. You will need:

- A laptop/desktop with unrestricted terminal access
- Rust and an IDE (e.g., VSCode)
- Reliable internet access
- The Rust Programming Language Book
- Operating Systems and Middleware: Supporting Controlled Interaction

All reading materials, including textbooks, articles, videos, and other media, will be available on the course site.

Learning Outcomes

Modern computing often conceals its complexity behind layers of abstraction, allowing software engineers to treat systems as if they operate by "magic." Programs execute, devices respond, and applications scale—yet the underlying mechanisms remain hidden. In this course, we will strip away that sense of mystery by examining the principles and mechanisms that make modern computing possible.

This course is organized into three interconnected areas: Concurrent Programming, Computer Hardware, and Operating Systems. By moving through these domains, we will uncover the fundamental processes by which computing resources are allocated, coordinated, and abstracted into the tools and environments we use every day. While the breadth of operating systems cannot be fully covered in a single semester, this course will provide both a rigorous conceptual foundation and hands-on experience to prepare you for deeper exploration.

By the end of this course, students will be able to:

1. Model and implement concurrent systems in Rust, explain the representations of concurrency (threads, processes, and tasks), and use synchronization primitives (e.g., mutexes, semaphores, channels) to coordinate concurrent execution.

2. Analyze the hardware resources managed by an operating system, describe the functionality of CPUs, memory hierarchies, I/O devices, and supporting components on a motherboard, and explain how operating systems interact with and abstract these components.

3. Apply concurrency and hardware knowledge to core operating system mechanisms, understand and evaluate scheduling algorithms, file system structures, and memory management techniques, and demonstrate how these mechanisms work together to enable robust and efficient computing.

4. Develop systems-level software in Rust, write safe and efficient Rust programs that engage directly with concurrency and hardware abstractions, and describe how Rust's design supports systems programming and its applications in modern OS development, including Linux.

Through these objectives, students will gain a working knowledge of how operating systems function, the ability to critically reason about their design, and the foundation necessary to construct simplified operating system components of their own.

Grading & Standards-Based System

This course uses a **standards-based grading** system. Grades are earned additively rather than averaged. You can recover points later if needed, ensuring early struggles do not lock in a lower grade.

Course Structure

The course is divided into three standards (modules), each worth 80 points (total: 240). Assignments are organized as follows:

Standard	**Assignments**	**Points**
Syllabus	HW0 (3 pts), EX S0 (2 pts)	5 (bonus)
Concurrent Programming	HW1, HW2, AC1 (each 30pts)	80 needed
Computer Hardware	HW3, HW4, AC2 (each 30pts), OHW1 (20pts)	80 needed
Operating Systems	HW5, HW6, AC3 (each 30pts), OHW2 (20pts)	80 needed

Optional exams (Midterm, Final) provide additional opportunities to recover points in earlier standards.

Assignments & Policies

- **Homework:** Each standard has two primary homeworks (HW) and an optional homework (OHW) targeting the previous standard. Homeworks (HW) are broken down into two major areas: Reading and programming. The reading portion will consist of engaging with our two textbooks, additional papers, and watching videos. The programming portion will be distributed as a Rust project skeleton that must be completed to pass a series of unit tests that require applying our theory into practice. All homework, optional or otherwise, is due on Fridays.
- **Activities:** Each standard has a mandatory in-class activity. These activities will require full student engagement and are designed to cover lecture material in a hands-on manner. If attendance is impossible for an excused reason, there is a different assignment designed to replace the activity for the same credit.
- **Unexcused Extensions:** Automatic, unexcused extensions are available: turning in one portion of the Homework (written or programming) on Saturday results in a 5pt deduction. Turning in a portion on Sunday gives you the previous 5-point penalty and an additional 10-point deduction. Submitting a portion after Sunday without an excuse results in the loss of all points.
- **Excused Extensions:** Medical or personal emergencies can receive additional extensions with email communication requesting it.

Exams

The Midterm and Final are optional. They provide focused opportunities to boost scores in prior standards. If you have reached the point total required to pass a standard (80pts), then they are optional.

Submission

Assignments and exams are submitted through **Brightspace**. Grades and private materials will be posted there as well.

Course Calendar

Office Hours: Regularly held on Monday and Wednesdays, 2–5 PM in Foley Annex 139.

The following calendar lists all lectures, homework due dates, activities, exams, and holidays. Dates are subject to change; updates will be communicated via email and Brightspace.

Date	Type	Details
Aug 26 (Tue)	Lecture	(LN0) Syllabus Day
Aug 28 (Thu)	Lecture	(LN1) Introduction to The Rust Programming Language
Sep 1 (Mon)	Holiday	Labor Day – No Class
Sep 2 (Tue)	Lecture	(LN2) Variables and Data Types
Sep 4 (Thu)	Lecture	(LN3) Functions and The Owner/Borrow System
Sep 5 (Fri)	Homework	HW0 Due
Sep 9 (Tue)	Lecture	(LN4) Higher Order Functions and Memory Management
Sep 11 (Thu)	Lecture	(LN5) OOP, Structs, Traits, and Enums
Sep 16 (Tue)	Lecture	(LN6) Threads and The History of Operating Systems
Sep 18 (Thu)	Lecture	(LN7) Introduction to Parallel Programming Theory
Sep 19 (Fri)	Homework	HW1 Due
Sep 23 (Tue)	Activity	(AC1)
Sep 25 (Thu)	Lecture	(LN8) Introduction to CPU Architecture
Sep 30 (Tue)	Lecture	(LN9) The Process
Oct 2 (Thu)	Lecture	(LN10) The Scheduling Problem
Oct 3 (Fri)	Homework	HW2 Due
Oct 7 (Tue)	Lecture	(LN11) The Scheduling Problem Continued
Oct 9 (Thu)	Lecture	(LN12) Deadlocks and Resource Contention
Oct 10 (Fri)	Holiday	Autumn Day! No Class
Oct 12 (Sun)	Exam	Midterm Release Day
Oct 14 (Tue)	Lecture	(LN13) Introduction to Memory Architecture
Oct 16 (Thu)	Lecture	(LN14) Memory Allocation and Fragmentation
Oct 17 (Fri)	Homework	HW3 Due
Oct 18 (Sat)	Exam	Midterm Due
Oct 21 (Tue)	Activity	(AC2)
Oct 23 (Thu)	Lecture	(LN15) Logical vs Physical Addressing and Virtual Memory
Oct 28 (Tue)	Lecture	(LN16) Paging and Segmentation
Oct 30 (Thu)	Lecture	(LN17) Introduction to I/O Architecture
Oct 31 (Fri)	Homework	HW4 Due
Nov 4 (Tue)	Lecture	(LN18) Direct Memory Access and Drivers
Nov 6 (Thu)	Lecture	(LN19) Sockets and Networking
Nov 11 (Tue)	Lecture	(LN20) Introduction to Storage Architecture
Nov 13 (Thu)	Lecture	(LN21) The File System
Nov 14 (Fri)	Homework	HW5 Due
Nov 18 (Tue)	Lecture	(LN22) The Terminal
Nov 20 (Thu)	Activity	(AC3)
Nov 25 (Tue)	No Class	Thanksgiving Break

Date	Type	Details
Nov 26–28 (Wed–Fri)	Holiday	Thanksgiving
Dec 2 (Tue)	Lecture	(LN26) Applications in LINUX
Dec 4 (Thu)	Lecture	(LN27) Applications in LINUX Continued
Dec 5 (Fri)	Homework	HW6 Due
Dec 8 (Mon)	Exam	Final Release Day
Dec 12 (Fri)	Exam	Final Due

Student Responsibilities

This page summarizes the University rules that relate to this course and our in-person classroom environment.

Academic Honesty Policy

> "Loyola Marymount University is a community dedicated to academic excellence. Academic honesty in scholarship and creative work stands at the center of LMU's academic life, and is essential for true learning and creation of knowledge to take place. As a university in the Jesuit and Marymount traditions, this community expects its members to act in accordance with the highest standards of honesty and ethics at all times. Violations of academic honesty undermine the fundamental educational mission of the University and cannot be tolerated. Students are responsible for understanding the standards of academic honesty and determining how they apply to their academic work and behavior."

Academic Honesty Policy Website

Disability Support Services

> "The mission of the Disability Support Services (DSS) Office at LMU is to provide equal access and opportunities for students with established disabilities. We are committed to promoting and celebrating the diversity of our students, staff, and faculty and work to eliminate systemic barriers, address individual bias, and maintain a respectful and equitable working environment."

Disability Support Services Website

LMU Expectation for Classroom Behavior

> "Disruptive behavior which is persistent or significantly interferes with classroom activities may be subject to disciplinary action. A student may be referred to the Office of Student Conduct and Community Responsibility if their behavior constitutes a violation of the conduct code."

OSCCR Website

Reporting of Sexual or Interpersonal Misconduct

> "Loyola Marymount University ("LMU") recognizes the significant, unacceptable and nationwide existence of Sexual and Interpersonal Misconduct on college campuses. LMU is dedicated to the prevention of such misconduct and to providing a caring, supportive and effective response when such misconduct occurs."

Misconduct Reporting Information

Communication Expectations

> Official communication must occur through LMU-approved channels (student email, Department Teams/Slack, or letters on official LMU letterhead). Students are responsible for regularly checking their LMU email. Other official channels may be audited in cases of misconduct.

Emergency Preparedness

> "Building a more resilient LMU is a shared responsibility among all students, faculty, and staff. Emergency Management serves as an all-encompassing tool for campus emergency information, disaster readiness events at LMU, and tips on how to prepare for and respond to various emergencies."

Emergency Preparedness Website

This syllabus is tentative and subject to change. Updates will be communicated promptly via email. A more detailed and expansive version of this Syllabus is provided at www.jag.prof/cmsi-3510/syllabus