Syllabus for CS 111, Operating Systems Principles
Fall 2016
Peter Reiher

CS 111, Operating System Principles is meant to provide you with a solid grounding in the basic principles of operating system design. This grounding will give you a deeper understanding of how a vital piece of system software operates and introduce you to important concepts that are frequently used in large software systems you are likely to encounter in your future work.

CS 111 has prerequisites of CS 32, 33, and 35L. The class involves serious programming and would be extremely challenging for anyone without a programming background. It also assumes knowledge of many fundamental tools used in software development, particularly in Linux/Unix environments. If you have not taken these courses or otherwise acquired this background, you are likely to find CS 111 very difficult.

CS 111 is widely held to be one of the most difficult courses in the undergraduate Computer Science catalog, due to:

- the amount of reading
- the number of new and subtle concepts to be mastered
- the complexity of the principles that must be applied
- the amount of work involved in the projects

People who have had little difficulty with previous Computer Science courses are often surprised by the workload in this course. Keeping up in this course requires considerable work and discipline. Catching up after falling behind is extremely difficult.

Textbook and Readings

Most of the readings for this course will come from Remzi Arpaci-Dusseau's Operating Systems in Three Easy Pieces. This is an on-line textbook, so you will not need to purchase a book for this class. However, the course covers certain topics that are not discussed in this textbook. These topics will be covered from alternative readings (all available on-line). These readings will be posted on the class web site before the class in question. Tests will cover material from both lectures and all assigned readings.

Projects

There will be a total of five projects for this class. Project 0 is meant primarily to give you a self-evaluation to determine if you are sufficiently prepared to take the class. The other projects relate to important concepts and system components that you should understand after taking this class. The projects will require substantial programming and are intended to be challenging. All projects are to be performed individually.

The projects will be supervised primarily by the TAs. Questions about projects should be directed to them, rather than me, unless the TA has already failed to address your question. The TAs for the Fall 16 offering are:

Muhammad Mehdi - taqi@cs.ucla.edu

Diyu Zhou - zhoudiyupku@gmail.com

Jungbeom Lee - nanaya07@naver.com

Project 4 concerns embedded system design and touches on issues of integrating sensors into a system, distributed systems, and security. It will require you to obtain a particular piece of hardware, an Intel Edison kit. This kit is available from various sources and is likely to cost around $100. You will be provided with further information about exactly what you will need and how to obtain it before the class starts.

Tests and Quizzes

There will be a midterm and final exam for this class. The midterm will be held in class during the fifth week. The final exam will be held in the class' scheduled final exam period.

There will be a quiz for each class, except for the first class and the period of the midterm exam. It will be based on the assigned readings for that class session. This quiz must be taken on-line before the start of the session. If you fail to take a quiz before the class, you will not be able to take it later.

Academic Honesty

I expect all students to follow the UCLA Student Conduct Code . This code prohibits cheating, fabrication, multiple submissions, and facilitating academic dishonesty. You can find further information about this code at the Student Guide to Academic Integrity. The Office of the Dean of Students, which offers a workshop on academic integrity if you wish to understand UCLA's policies on this issue more thoroughly.

Group study is often useful and is encouraged, but projects are to be performed by each student individually, excepting project 4, which can be performed with one teammate. You are not permitted to use any other students' code or written material in your projects, and you should not give your code or written material to any other students. In cases where multiple projects show signs of plagiarism, all involved parties will be reported to the Dean, so even if it was your work that was copied, you could still face consequences. Brief quotations from other sources are sometimes acceptable in project submissions, but submissions should be predominantly your own original work. If you include any material in your project submissions that is obtained from an online source, book, or other place, you must specify the sources for these parts of your submitted work.

If you have questions about the academic honesty policy, please discuss them with me. Be warned that we take academic honesty very seriously. We look for plagiarism in various ways, and I report all suspected cases to the Dean's Office, as university policy requires. Once reported, the matter is in the hands of the Dean, and I cannot further influence it. I strongly advise that you take no chances on academic honesty issues. If an issue of this kind is unclear to you, talk to me or to the TAs.

Grading

Grading will be based on scores on the quizzes, tests, and labs, with the following breakdown:

Quizzes	5%
Midterm	20%
Final Exam	30%
Lab 0	5%
Labs 1-4	10% each

I do not have a formal curve for my classes, but I tend to adjust grades relative to the performance of other students in the class.

The TAs will be in charge of late policies and makeup policies for the labs. I will enforce the policies they choose. Makeups or alternate times for exams will only be possible with consent of the instructor prior to the exam date. No makeups for quizzes will be allowed.

Lecture Plans

The dates and lecture topics indicated here represent my current plans, and may alter based on circumstances.

Week 1

Thursday, September 22: Introduction

Tuesday, September 27: OS Services and Interfaces

Wednesday, September 28: Project 0 (Warmup) due

Week 2

Thursday, September 29: Processes

Tuesday, October 4: Scheduling

Wednesday, October 5: Project 1A (I/O and IPC) due

Week 3

Thursday, October 6: Basic Memory Management

Tuesday, October 11: Swapping and Paging

Wednesday, October 12: Project 1B (Networking) due

Week 4

Thursday, October 13: Threads and Interprocess Communications

Tuesday, October 18: Mutual Exclusion

Wednesday, October 19: Project 2A (Atomic Operations) due

Week 5

Thursday, October 20: Synchronization

Tuesday, October 25: Midterm exam

Wednesday, October 26: Project 2B (Complex Critical Sections) due

Week 6

Thursday, October 27: Deadlocks

Tuesday, November 1: Performance and Metrics

Wednesday, November 2: Project 2C (Lock Granularity and Performance) due

Week 7

Thursday, November 3: Disks and I/O

Tuesday, November 8: File Systems

Wednesday, November 9: Project 3A (File System Dump) due

Week 8

Thursday, November 10: File System Performance

Tuesday, November 15: Operating Systems Security

Wednesday, November 16: Project 3B (File System Analysis) due

Week 9

Thursday, November 17: Distributed Systems

Tuesday, November 22: Remote Data Architectures

Week 10:

No class on Thursday, November 24: Thanksgiving holiday

Tuesday, November 29: Distributed Computing

Thursday, December 1: To be announced

Friday, December 2: Embedded Systems Project due

Final exam: Monday, December 5, 11:30 AM – 2:30 PM