Syllabus for CS 111, Operating Systems Principles
Summer 2017
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. Exam questions WILL assume you have read and understood all assigned readings, so if you don't read them, you are likely to do poorly on the exams.

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, except for project 3, which can be performed by teams of two students.

The projects will be supervised primarily by the TA. Questions about projects should be directed to him, rather than me, unless the TA has already failed to address your question. The TA for the Summer 17 offering will be Diyu Zhou (email address: zhoudiyupku@gmail.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, augmented with a Grove sensor kit. These kits are available from various sources and are 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. While project 4 is not finally due until the end of the class, elements of the project are due as early as the end of the 4th week of class, and require you to have obtained the hardware to complete. So I recommend you buy it immediately.

Tests

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 final class of the 10th week of the quarter.

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 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, except for project 3. You are not permitted to use any other students' code or written material in your projects (other than your partner's for project 3), and you should not give your code or written material to any other students. You are not permitted to post your code to Github or to other web sites. 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. If you are able to find project solutions on line, we are likely to find them, too. 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 TA.

Grading

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

Class Evaluation	1%
Midterm	24%
Final Exam	30%
Lab 0	5%
Labs 1-4	10% each

You will receive 1% credit for this class merely for filling out the class evaluation, which will be made available online in the final week of classes. How you evaluate the class will have no bearing on the credit; merely submitting a complete evaluation will earn the 1%.

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 TA will be in charge of late policies and makeup policies for the labs. I will enforce the policies he chooses.   Makeups or alternate times for exams will only be possible with consent of the instructor prior to the exam date.

Lecture Plans

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

Week 1

Tuesday, June 27: Introduction

Thursday, June 29: OS Services and Interfaces

Friday, June 30: Project 0 (Warmup) due

Week 2

Tuesday, July 4: No class: Independence Day

Thursday, July 6: Processes

Friday, June 7: Project 1A (I/O and IPC) due

Week 3

Tuesday, July11: Scheduling

Thursday, July 13: Basic Memory Management

Friday, July 14: Project 1B (Networking) due

Week 4

Tuesday, July 18: Swapping, Paging, and Virtual Memory

Thursday, July 20: Threads and Interprocess Communications

Friday, July 21: Project 4A (Edison Configuration) due

Week 5

Tuesday, July 25: Asynchronous Completion and Mutual Exclusion

Thursday, July 27 : Semaphores and Locking

Friday, July 28: Project 2A (Atomic Operations) due

Week 6

Tuesday, August 1:   Midterm exam

Thursday, August 3: Deadlocks

Friday, August 4: Project 2B (Complex Critical Sections) due

Week 7

Tuesday, August 8: Performance and Metrics

Thursday, August 10: Disks and I/O

Friday, August 11: Project 4B (Edison sensors) due Project 3A due

Week 8

Tuesday, August 15: File Systems

Thursday, August 17: File System Performance

Friday, August 18: Project 3A (File System Dump) due

Week 9

Tuesday, August 22: Operating Systems Security

Thursday, August 24: Distributed Systems

Friday, August 25: Project 3B (File System Analysis) due

Week 10:

Tuesday, August 29: Remote Data Architectures

Thursday, August 31: Final exam

Friday, September 1: Project 4 (Internet of Things Security) due