

CS 481/ECE 437 Computer Operating Systems, Spring 2012

Instructor: Jed Crandall, jedcrandall@gmail.com

Never hesitate to email me directly about anything. If you're emailing me something about a group assignment, always cc the members of your group unless there's some reason for privacy.

Office and office hours: FEC 335, Tuesdays, Wednesdays, and Thursdays from 2:00pm to 3:00pm.

Prerequisites: CS 341 (Computer Organization and Design) If you haven't taken and passed 341, come talk to me early in the semester. In general, I'll assume you already know about virtual memory and page tables, system calls, assembly language, *etc.*, though we will review some of this material very briefly.

TA: The TA is Xu Zhang, he will attend class regularly. He will mostly help out with system administration-type duties so won't be holding office hours.

Mailing lists: There will be two mailing lists, one required and one optional. See the course website for details.

Course website: <http://www.cs.unm.edu/~crandall/481spring2012/>

I'll post lots of important stuff here, like the lab assignments, links to the mailing lists and Google calendar, grades, *etc.*

Required texts: *Operating System Concepts, 8th Edition*, by Silberchatz *et al.*, and *Understanding the Linux Kernel, 3rd Edition*, by Bovet and Cesati. I recommend buying hard-copies of the books, since I can't accommodate students who use PDFs of the book during open-book tests (unless there are special arrangements, *e.g.*, with the UNM testing center for that specific student).

Class meeting time and place: Tue/Thur 12:30pm to 9:15am, in Mechanical Engineering room 210 (ME 210).

Grading: The final grade will be calculated as 50% labs, 30% attendance, 10% midterm, and 10% final. The points for each will be added up and divided into the total possible before weighting, so a 100-point lab does not necessarily contribute the same amount to your grade as a 100 point final. I reserve the right to curve the overall grades at the end of the semester (up, never down) if I don't feel that they reflect the amount of effort students put into the class. The overall grade will be out of 100, weighted as described above. For letter grade purposes, below 60 is an F, 60 and up is a D, 65 and up is a C-, 70 and up is a C, 75 and up is a C+, 80 and up is a B-, 82 and up is a B, 85 and up is a B+, 87 and up is an A-, and 90 and up is an A. I only give A+'s in extreme circumstances.

Labs: There will be three labs, each worth 100 points. In general, you'll be using C for the labs, but sometimes a scripting language such as perl or Python will be acceptable for some parts of the lab. I'll make it clear on the writeup for each lab assignment what languages can be used. Normally, I'm agnostic about languages, but CS 481/ECE 437 is a core course and is part of the systems sequence so it's required that I make sure you are proficient in the C language before passing CS 481.

Late lab assignments will be accepted only in special circumstances (medical, *etc.*).

Attendance and reading assignments: I will be taking attendance every day of the semester. Your

grade for attendance will be the fraction of regularly scheduled lecture periods for which you are present. I may mark you as not present, without immediately notifying you, for any of the following reasons:

- If you don't show up to class that day.
- If you're more than five minutes late.
- If you're using any computer (even a personal computer) for things not related to this class (e-mail, Facebook, etc.).

I'll drop up to two “not present” days at the end of the semester. Things like medical emergencies, attending conferences, *etc.*, may be considered excused absences (*i.e.*, not count against your grade) if you contact me about them in a timely manner.

Midterm: The midterm will be on Tuesday, 8 March in class at the regular time. It may be curved.

Final: The final will be on Thursday, 10 May from 10:00am to 12:00pm, during finals week. I assume it will be in the same room where the class meets. It may be curved.

UNM statement of compliance with ADA: “Qualified students with disabilities needing appropriate academic adjustments should contact the professor as soon as possible to ensure your needs are met in a timely manner. Students must inform the professor of the disability early in the class so appropriate accommodations can be met. Handouts are available in alternative accessible formats upon request.”

Cheating and collaboration, personal statements:

Unless stated otherwise in an explicit way in the lab assignment writeup, you're expected do your own lab implementation and setup, write your own code, collect your own data, and write your own lab writeups. Sharing of source code, configuration files, *etc.*, will be considered cheating unless you have it in writing from me that it's okay. This is not a programming class, but C programming is a significant part of the class so you should treat this as a programming class in terms of always writing your own source code and not sharing it with your classmates. Everything you write in the English language, and all ideas that you present as your own in the lab writeups, needs to be original material by you. *If you copy and paste any material (English text, figures, etc.) from any source you must clearly delineate it and attribute it properly to its source. Representing the work and materials of others as your own will not be tolerated in this class. Plagiarism will be treated the same as cheating, meaning you'll get a 0 on the assignment and I may pursue further action through University policies.*

Each lab assignment will have specific instructions about what is acceptable in terms of cheating and collaboration. Be sure to read it, and if you don't understand it ask me questions.

Each test will state at the top what materials you're allowed to use (book, notes, *etc.*). Not noticing, for example, that the top of the test says that it's not open notes is not an excuse. Anything not specified as open is closed. In other words if the test instructions don't say “open-iPod” you should assume that the test is closed-iPod, and if the test instructions don't say “open-cheat-sheet-on-the-inside-of-your-water-bottle-label”, assume that the test is closed-cheat-sheet-on-the-inside-of-your-water-bottle-label.

All university policies regarding these matters will be strictly enforced. Typically I'll give the cheating parties a 0 on the assignment, but I may pursue further action in some cases.

My expectations of you as students

- **Be studious:** I'm fairly old-fashioned, I expect students to come to class, to come on time, to stay on task, to take the time to make sure they understand things well, etc.
- **Take responsibility for your own learning:** you're registered for a 400-level class at a major

research institution. If you find that coming to the regularly scheduled class time is a waste of time, then you're not taking responsibility for your own learning. Don't expect me to spoon-feed you information that is already well-known, you don't want to pay about \$750 in tuition for me to tell you what's in the two books that total about \$150 that you could read yourself if you wanted to. A good philosophical approach for you to take in this class is to “teach the teacher.”

- **Do only excellent work:** anything worth doing is worth doing well.
- **Show leadership and be a mentor:** don't think that this class is only about computer operating systems. I don't anticipate any group work this semester, but I would encourage you to form study groups, share recent OS news or helpful resources with your classmates, *etc.*
- **RTFM:** read the manual. I don't mind at all when students ask me questions that they could get the answer to from the man pages, but it's a bad habit since a good systems programmer always checks the man pages first. So, don't feel like I'm rebuking you if I say, “RTFM,” it's just that I want to instill good habits in you and me acting as a man page is a bad habit for both of us.

Material to be covered:

We'll try to stick to the very basics of operating systems this semester. This includes processes, threads, thread and process management, scheduling, synchronization, virtual memory, memory management, process communication and signals, I/O, UNIX security mechanisms, and file systems. Concurrency will be a major theme of the semester, and most of our exploration will be based on Ubuntu Linux 10.04 and the Linux Kernel 2.6 source code. There are other things that I'm interested in that I'm sure will come up this semester, especially experimental design and security, but since this is a core course my main concern this semester will be making sure you have a solid foundation of understanding for the basic elements of an operating system.

First assignment:

Your first assignment will be to get a virtual machine up and running with Ubuntu Server 10.04 32-bit for the x86. It needs to be the server edition, so as to avoid the complications that a GUI adds to the OS. It should be Ubuntu 10.04 so that I can help you in an environment that I'm familiar with. 32-bit x86 is also required, so that if I ask you to dump some assembly language it will be assembly language that I can read. Also, for some architectures, 32-bit is virtualized more efficiently by the hardware. So, don't deviate, make sure you have a fresh install of Ubuntu Server 10.04 32-bit on an x86 virtual machine. Also don't use paravirtualization, such as UML, since we don't want there to be anything but trivial differences between the kernel image on your virtual system and the kernel image that we would expect to see on real hardware.

I recommend VirtualBox, but you can use any virtual machine software you wish. You'll want to avoid installing on bare metal, and it's probably best to choose a virtual machine that supports checkpoints, since you'll be installing modules into the Linux kernel and maybe even changing its source code. Also VirtualBox has been known to have bugs that corrupt virtual machines from time to time. So checkpointing and some backups/version management of your code is a must. If you use VirtualBox, always make sure to check the "Use host I/O cache" checkbox for the SATA controller for your virtualized hard drive. This will help to avoid bugs, especially on the NFS mount for the department.

You'll need to be a sudoer (Ubuntu doesn't have a root user). Ubuntu adds the user you added during installation to the admin group so that you're automatically a sudoer, so the only reason I'm telling you that you'll need sudoer access is so that you'll know why you need to use a virtual machine. Also, you're going to be doing unspeakable things to the machine you'll be doing your labs on, so using any Ubuntu machine that anybody else cares about is not a good idea, you should use a fresh install.

You'll probably want to add a NAT network interface before doing the OS install, so that your VM can be on the Internet.

Your virtual machine should be on a computer that you can use regularly for doing your lab assignments, so it should be your own laptop or the CS dept. lab machines. If you need help finding a place to host your virtual machine or with setting it up, let me know. We have a server where we can run headless virtual machines that you can access remotely. It's also possible to run headless virtual machines on the CS lab machines and access them remotely. We won't be using graphics or GUIs for anything this semester, really. It would be a good idea to run your VM on your own laptop and then bring your laptop to class to follow along during lectures, but it's not required. In general, I'll be projecting a Linux shell session during lectures, and you can either watch my screen and take notes or type the same commands on your own screen.

This is not an assignment that you'll turn in for a grade, but you should make sure you have a working Ubuntu Server 10.04 32-bit virtual machine before class on Tuesday, 24 January, because we'll be diving right into the first lab on that day.