CMPS 111: Introduction to Operating Systems

Computer Science Department
University of California, Santa Cruz
Winter 2014

Syllabus

Time:	MWF 3:30-4:40pm
Location:	Engineering 2 - 194
Section:	Baskin Engineering - 109 (Tue 8-10am, Thur 8-10am)
Instructor:	Prof. Darrell Long (darrell«at»cs.ucsc.edu)
Office & hours:	Darrell (Engineering 2 - 371): TBD
TA:	Daniel Lipovetsky (danl«at»cs.ucsc.edu)
TA office & hours:	Engineering 2 - 480 (Mon 2-3pm, Wed 1:30-2:30pm)
Prerequisites:	CMPS 101 and CMPE 110
Required text:	*Modern Operating Systems, 3rd Edition*, Tanenbaum
Optional texts:	*Operating Systems: Design & Implementation (3rd edition)*, Tanenbaum & Woodhull
	The Design and Implementation of the FreeBSD Operating System, McKusick, *et al.*
	E. G. Coffman and P. J. Denning, Operating Systems Theory (out of print).
Home page:	http://piazza.com/ucsc/winter2014/cmps111

Course Objectives

The goal for students in this course is to learn the fundamental principles of operating systems. To help you accomplish this, we will discuss the most of the important aspects of operating systems in general, examine specific examples from current operating systems, and do operating system kernel programming assignments using a Unix-like teaching-oriented operating system (MINIX 3). The specific topics we will cover include:

- Basic operating system concepts
- Processes & scheduling
- Deadlocks
- Memory management
- Operating system management of input/output devices
- File systems
- Protection & security
- Introduction to distributed operating systems (if time permits)

Where possible and appropriate, we will use examples from Linux, FreeBSD, and other modern operating systems to illustrate concepts covered in class.

Prerequisites & Texts

The formal prerequisites for this class are CMPS 101 and CMPS 110. Students should have taken these classes or equivalent relatively recently; experience has shown that students who took these classes more than two years ago tend to have more difficulty with the material in CMPS 111. In addition to the class prerequisites, you must have your own ITS account. Because assignments will be turned in on the ITS system, ***you must have your own ITS account by the end of the first week of classes***. You may not borrow another person's account. Doing so is a violation of the ITS user policy, and will be reported. You should also be familiar with C programming using Unix (including the UCSC Unix systems) using such tools as `gcc` and `makefile`s. You're welcome to ask for help with these tools, but such questions will have lower priority than those about material covered in the course.

The required text (*Modern Operating Systems, 3rd Edition*) is available at the UCSC bookstore as well as at on-line booksellers such as Amazon.com and BarnesAndNoble.com. It really is a required text—homework problems may be taken from the book, and lectures and notes will complement the material as presented there. ***Make sure you get the third edition of the text***. The optional texts are just that—optional. One covers a popular operating system—FreeBSD—in depth and may be of interest to those who want more details about operating systems in the real world. The other provides some additional detail on the structure of the MINIX 3 operating system. This material will be very helpful in doing your programming assignments. However, it's not required because some people may not need the extra documentation that the book provides. The book is supposed to be available on-line (for a fee) via SafariX, and the book is available from Amazon.com as well.

Assignments & Grades

Assignments will be posted on the web, and will be accessible from anywhere on the Internet. It is likely that some assignments will be posted before they are officially assigned; however, you should not assume that an assignment on the web is in final form until the date it is assigned. In other words, *assignments are subject to change before the date that they are officially assigned*.

Homework

There will be a homework assigned about every week and a half, due about one week later. The homework will give you a chance to see how well you understand the concepts we've covered in class. They will be graded out of a total of 5 points.

Programming projects

Programming projects are an important component of this course. You'll be making changes to the MINIX 3 operating system, which runs on x86 processors or simulations of them. x86 simulators capable of running MINIX are available for Mac OS X, Windows XP, Linux, and installed on `unix.ic`. The programming tools you'll need (C compiler, debugger, etc.) are part of the MINIX install CD, which, along with more details on the projects, is available from the project page.

Projects 2–4 may be done in teams of two students, both of whom must be enrolled in the class—your friend who works at Microsoft or Google isn't eligible, unless he or she is in the class this quarter. Project partners receive the same grade for the project (modulo grace days, as described below).

Examinations

There will be two examinations in the class: a midterm around week 6, and a final exam during examination week. Both examinations are mandatory. If you miss an examination, you should have "barfed up a lung" or something equally dramatic.

Grades

Assignments are due on the date and time specified in the assignment. Late homeworks will not be accepted. To accommodate potential scheduling issues during the class, you have *3 grace days* you may use to extend the due date of a programming assignment (for you only, not the whole class). You may use as many grace days as you wish for any assignment, assuming you haven't already used them up already. Programming assignments turned in after the due date lose 25 points per day, including weekends and holidays—assignments more than three days late will receive a maximum grade of 5 points. Please make a note in a README file that you're using grace days when you submit the assignment; you cannot retroactively apply them to assignments you've already submitted. If you're part of a project team that wants to use grace days, **each person** on the team must use his/her own grace days to extend the due date. For example, if Alice and Bob want a one day extension for the due date for their project, Alice must use one of her days and Bob must use one of his; it's not acceptable to use two of Alice's days. If only one partner uses grace days, grades for the two partners will be adjusted accordingly. You must turn in a reasonable attempt at each project to pass the class.

There may be a time where you submit a project but it doesn't work for the TA (the TA will not fix your programs for you). If this happens, your project will be returned to you and you will be given the opportunity to fix and resubmit the project. After the project is returned to you, *every 24 hour period used to fix the project will use up a grace day*. If you no longer have any grace days left, 10% will be docked off of the

project for every day late. If you feel that the project should have worked, and you are not at fault, please contact the TA immediately in order to schedule an appointment to resolve the issue.

Grades will generally follow a 90/80/70/60 scale, but we reserve the right to adjust the curve upward or downward as appropriate. Your grades will be determined as follows:

- Programming assignments: 10% *each* (total of 40%)
- Homework: 10% total (all homeworks weighted equally)
- Midterm: 18%
- Final: 30%
- Class participation: 2%

***Note*:** You must average above 50% on the programming assignments and on the examinations in order to pass the class. A score below 50% on *either* part of the class will result in a failing grade regardless of your overall score. This is a necessary, but not sufficient, condition for passing the class—averaging above 50% on both parts does *not* guarantee a passing grade.

Attendance

Class attendance is mandatory; we typically won't take attendance, but class participation *is* 2% of your grade. So how do we know? Don't worry, *we know*.

Homeworks, assignments, and important dates will be posted to Piazza, but this is provided as a courtesy and is not always complete. *It's your responsibility to find out what you missed if you don't attend class.* Laboratory section attendance is also required, though attendance won't be taken. However, you'll miss important material on the programming assignments if you don't attend one laboratory section per week. This is where the programming assignments will be discussed in detail. Office hours are optional. They're your chance to ask the professor and TA questions about the material being covered, programming assignments, or anything else about operating systems (or other general computer science issues) you want to discuss.

Getting Answers to Your Questions

Operating systems is a tough subject, so there are several ways to get help with concepts covered in class, homework, and programming projects, listed in approximately the order you should try them for help.

- Attend classes and laboratory sections
- Check the class web page frequently (every day or two)
- Read and post to the class discussion forum (web-based)
- Meet with the professor and TA during office hours
- Send electronic mail to the professor and TA (cmps111-staff «at» ucsc.edu)

Please meet with the professor or TA in person if you have an in-depth question, such as detailed help with the programming assignment design or debugging. Please don't just drop by outside of office hours; if you can't attend office hours, arrange a meeting *in advance* by contacting the person you want to meet with.

Academic Honesty: Collaboration versus Cheating

In an ideal world, academic honesty wouldn't be an issue, but, sadly, experience has made the following discussion necessary.

You're encouraged to discuss the course material and concepts with other students in the class. If you do so, you may not take notes during the discussion. You should also follow the *Simpsons rule* (historically, this was called the *Gilligan's Island rule*, but all of you are too young to have seen Gilligan's Island unless watch certain cable channels late at night): you should watch an episode of *The Simpsons* between discussing the material with other students and working on your own assignment; other 30 minute activities, such as watching *Animaniacs*, *Spongebob Squarepants*, or *Gilligan's Island* are also acceptable.

If you get significant help on an assignment from a source other than the textbook, slides, your own notes, or the course staff, you *must* acknowledge it in whatever you turn in. It's not cheating to read other textbooks or look for help on the Internet; it *is* cheating if you copy solutions from one of these sources. Under no circumstances may you look at anyone else's code or show anyone else your code (except for your project partner, of course). While you may discuss the concepts used in the programming assignments, you may not discuss implementation details of the assignments themselves. If you are caught copying or otherwise turning in work that is not solely your own, ***you will fail the course and a letter will be sent to your Department, the School of Engineering, and to your Provost and academic preceptor***.

The bottom line is that you're expected to conduct yourself as a person of integrity—you're expected to adhere to the highest standards of academic integrity. This means that plagiarism in any form is completely unacceptable. You are a (soon-to-be) computing professional; we encourage you to consult the code of ethics appropriate to your discipline (the ACM, IEEE, and IEEE Computer Society).

Plagiarism will be assumed until disproved on work that is essentially the same as that of other students. This includes identically incorrect, off-the-wall, and highly unusual duplicate answers where the probability of a sheer coincidence is extremely unlikely. All parties to this unacceptable collaboration will receive the same treatment. In the case of programs, reordering routines, renaming files, and simply renaming variables does *not* make two programs different.

Additional Resources: “On-Campus Recovery Meetings”

Are you or someone close suffering from compulsive use of food, alcohol, other drugs, gaming, gambling, love, sex and/or co-dependency? Is your life feeling unmanageable? Students like you have found relief from mental obsessions of all kinds. The solution is simple: We help one another and practice 12 steps of recovery. Join us on Mondays, 8:00 PM, at Stevenson Senior Commons. Coffee and treats provided! See website for more information: http://healthcenter.ucsc.edu/shop/aod-program/aa.html