Computer Systems **Course Syllabus**
CS5600 Fall 2010 **8 September 2010**

Instructor	Prof. Alan Mislove
Contact	`amislove@ccs.neu.edu` (please put "[CS5600]" in the subject line)
Office hours	Tuesdays, 3:00pm–5:00pm, 250 West Village H
Location	208 West Village H
Time	Wednesdays, 6:00pm–9:00pm
Web site	`http://www.ccs.neu.edu/~amislove/teaching/cs5600/fall10`
Forum	On Blackboard
Teaching assistant	Abutalib Aghayev
Office hours	Tuesdays, 5:00pm–7:00pm, 212 West Village H

DESCRIPTION

This course introduces students to the principles, design, and implementation of operating systems. The lectures focus primarily on the principles and design of operating systems; the projects expose students to the implementation aspects of operating systems and serves to solidify students' understanding of the course material.

LOGISTICS

The class will meet for 13 three-hour sessions, plus a two-hour final exam. Each three-hour session will typically be divided into three sessions, with short (five-minute) breaks between. The midterm exam will be administered during a regular class session.

PREREQUISITES

To succeed in this course, you to understand the basics of computer architecture, and to have experience implementing non-trivial systems-type projects. You should also be able to read UNIX manual pages, and be able to familiarize yourself with UNIX utilities.

This course will be project-centric, and *all students* will complete in projects in groups of two. Thus, to succeed in this course, you must be able to work in a group. I will allow you to form your own groups, but as you are free to choose your partner, I will not be sympathetic to complaints at the end of the semester about how your partner did not do any work.

Finally, to succeed in this course, you *must* have familiarity with C and UNIX compilation utilities (`make`, `gcc`, etc...). It is also highly recommended that you become familiar with using `gdb`, as this will greatly aid you in completing the projects. At a high level, you should be motivated, eager to learn, willing to work hard, and make up, on your own, any prerequisite deficiencies you may have.

GOALS

By the end of this course, I expect you to understand the following concepts

- Operating system basics - context switching and program loading
- Virtual memory at the hardware and operating system level
- Hardware virtualization
- File and block I/O, DMA
- Basic operating system security mechanisms

TEXTBOOK

The textbook for the course is

> Silberschatz, Galvin, and Gagne. *Operating Systems Concepts, 8th Ed.* Addison-Wesley, 2009. ISBN 0470128720.

and is available from the campus bookstore. Other useful texts and resources include

1. Goodheart and Cox. *The Magic Garden Explained, The Internals of* UNIX *System V Release 4*. Prentice-Hall, 1994. ISBN 0130981389.
2. McKusick, Bostic, Karels, Quarterman. *The Design and Implementation of the 4.4* BSD *Operating System*. Addison-Wesley, 1996. ISBN 0201549794.
3. Open Software Foundation. *Design of the* OSF/1 *Operating System*. Prentice-Hall, 1993. ISBN 0132028131.
4. Custer. *Inside Windows* NT, *Second Edition*. Microsoft Press, 1997. ISBN 1572316772.

PROGRAMMING PROJECTS

The overall goal of the projects in this course is to build an operating system for a simulated workstation. There will therefore be four programming projects, corresponding to the following phases of the project: thread management, multiprogramming, virtual memory, and file systems.

You will form groups of two people to do the programming projects.[1] To collaborate effectively, you should both be involved in all of the major design decisions. You should also determine a partitioning of responsibilities so that you can both work effectively in parallel. For example, one might be responsible for generating all the test code while the other is responsible for the main code. You may switch groups between programming projects.

Important: *You alone are responsible for finding a partner. The class forum (newsgroup) located on Blackboard is a particularly good resource for this. Breaks during lecture are also a good time to look for partners. Regardless, the course staff will be unsympathetic to requests for matching.*

While we be using either C to complete the programming projects. Therefore, familiarity with the C programming language is *absolutely essential* to successfully completing the projects.

The TAs have been instructed to grade in part on design and implementation style and to be increasingly strict about this as the semester proceeds. In other words, it is not enough to get

[1] If necessary, one group of three will be allowed.

a working solution; you must implement the solution in a clean way that would simplify making further enhancements. It will benefit you in the long run to work on your software engineering skills.

Programming projects are due at 11:59pm on the specified date.

WEB SITE AND FORUM

This class will make extensive use of Blackboard for posting materials and submitting assignments. Please make sure that you regularly check the email address used by Blackboard (either neu.edu or husky.neu.edu) or ensure that it is forwarded correctly.

We will provide a Web forum via Blackboard that can be used by students to ask questions and exchange wisdom while completing the projects in this course. Please use the forum to post questions and answers that may be useful to others. Specifically, questions of the form “How do I link foo?”, “Does such-and-such option in the compiler work for you?”, or “What is the precise interpretation of project 2 question III, part b?” should be posted on the forum first. If you mail me (or other course staff) these questions, we might not be able to answer them on time.

LECTURES AND SCRIBING

I will try to record and post lectures in this course to the course web page. There is no guarantee that I will be able to do this, so I would not recommend relying on being to watch a video of the lectures.

In order to help the class to record information conveyed during lecture, each student will scribe one hour of lecture at some point throughout the term. Scribing consists of taking lecture notes, cleaning them up, expanding on examples, and, essentially, making so that they serve as a useful point of reference for the concepts described. Once completed, scribing students should email their scribe notes to the instructor, who will collate them and post them on the course web site.

Scribe write-ups are due before the following class.

EXAMS

There will be one midterm and one final. All exams will be open book and open notes, but computers are not allowed nor is any access to the Internet via any device. The exams will cover material from lectures, readings, and the projects. They will cover the material discussed during the first and second halves of the class, respectively (i.e., they are not cumulative).

GRADING

The breakdown of the grades in this course is

60%	Projects (15% each)
35%	Exams (15% midterm and 20% final)
5%	Participation and scribing

Each project will include a breakdown and description of how it will be graded.

Any requests for grade changes or regrading must be made within seven days of when the work was returned. To ask for a regrade, attach to your work a page that specifies (a) the problem or problems you want to be regraded, and (b) for each of these problems, why do you think the problem was misgraded.

LATE POLICY

For programming projects, we will use flexible slip dates. Each student is given an automatic extension of 4 calendar days for the semester. You can use the extension on any project during the semester in increments of a day. For instance, you can hand in one project 4 days late, or one project 2 days late and two projects 1 day late. The slip time will be deducted from each group member's remaining slip time.[2] This should let you schedule due dates around the due dates for other courses. After you have used up your slip time, any project handed in late will receive no credit and will not be accepted. Extensions will not be granted.

ON CHEATING

It's OK to ask someone about the concepts, algorithms, or approaches needed to do the assignments. We encourage you to do so; both giving and taking advice will help you to learn. However, what you turn in must be your own, or for projects, your group's own work; copying other people's code, solution sets, or from any other sources is strictly prohibited. The project assignments must be the work of the students turning them in.

I will have no sympathy for students who are caught cheating. All students are subject to the Northeastern University Academic Integrity Policy. All cases of suspected plagiarism or other academic dishonesty will be referred to the Office of Student Conduct and Conflict Resolution (OSCCR).

ADVICE

As the course is project-centric, students are recommended to start early on projects. The projects require substantial design, implementation, and testing effort, especially for students who are unfamiliar with network programming. Students are encouraged to drop by the instructor's or teaching assistant's office hours (or set up a separate appointment via email) if they run into problems while completing the assignments.

Finally, computer use during class is allowed but expected to be for class purposes (e.g. note-taking, reference) only. Please avoid non-class-related computer use so that I do not have to revisit this policy.

[2] Note on slip days: Slip days can only be used if all group members have at least one remaining slip day.