EECS 482: Introduction to Operating Systems
Winter 2003

1 Basic Information

Prof. Brian Noble, 2245 EECS, **bnoble@umich.edu**

Graduate Student Instructors (GSI office hours will be held in the 3rd floor Media Union computer lab):
George Dunlap, Kim Kuether, Lukasz Opyrchal, and Matthew Smith

Web page: http://www.eecs.umich.edu/~bnoble/482
Newsgroup: news://news.engin.umich.edu/umich.eecs.class.482

Lectures:	001: TTh 11:30-1:00 1311 EECS	002: TTh 1:00-2:30 1109 FXB
Discussions:	011: M 11:30-12:30 1005 Dow	012: W 11:30-12:30 1005 Dow
	021: M 12:30-1:30 1018 Dow	022: W 12:30-1:30 1018 Dow

Note: Discussions and lectures are **not** linked; you may register for any discussion section.

2 Course Overview

EECS 482 is an introductory course in operating systems at the advanced undergraduate or beginning graduate level. The objective of the course is to familiarize you with the issues involved in the design and implementation of modern operating systems. The concepts in this course are not limited to any particular operating system or hardware platform. We will discuss examples that are drawn from historically significant and modern operating systems including MULTICS, UNIX, Mach, and Windows NT. We will cover topics such as processes and threads, concurrency and synchronization, CPU scheduling, virtual memory management, communication in distributed systems, secondary-storage management, file systems, and security.

To help you understand operating systems, you will implement several modules that form much of the core functionality in modern operating systems. These projects will give you practical exposure to topics such as threads, virtual memory management, client-server systems, and file systems. We will also provide practice questions that will be discussed in the discussion sections; you can use these to help gauge your understanding of the material as the course progresses.

3 Prerequisites

Students must have obtained a grade of C or better in EECS 370 and EECS 281/380. Students are expected to understand computer architecture and data structures, to have extensive C/C++ programming experience, and to be familiar with UNIX. Students with questions about whether they have sufficient preparation for this course should speak with the instructor as soon as possible.

4 Course Material

The required text for the course is *Modern Operating Systems (2nd edition)*, by Tanenbaum. There will also be supplementary readings posted. Students are required to read the home page and newsgroup frequently to stay current.

5 Course Projects

Three projects will be assigned during the term, each of which will require a substantial time commitment on your part. Each project will be completed by a group of 2-3 students. Projects will be auto-graded, and you will have three late days to use at your discretion throughout the semester, but no other extensions will ordinarily be given. All project code must be the work of your group this semester. While these projects are challenging, we can offer some tips to complete them without undue difficulty.

5.1 Group Policies

All projects in this course are group projects. You must form a group of 2-3 students for these projects. Groups may be drawn from the entire EECS 482 population; members of a group need not be in the same lecture or discussion

section. To declare a group's membership, send e-mail to eecs482staff@umich.edu with the group members names and uniqnames. The group declaration deadline is 8:00 PM, Friday, January 17th (shortly after Project 1 is assigned). After this date, we will randomly combine remaining students into groups of 2 or 3. It is very important to choose your partners carefully. You should discuss topics such as prior experience, course background, goals for this course, workload and schedule for this semester, and preferred project management and work style. Make sure you can find several blocks of time during the week to meet to discuss or carry out the project.

Students are expected to participate wholly in their group to the benefit of the entire group. All group members should be familiar with all aspects of the project, irrespective of their role on the project. We expect all group members to contribute their fair share, and we expect to assign the same project grade to all members of a group.

Each group member will evaluate the contributions of other group members after each project. Members who contribute less than their share may receive a lower grade on the project; non-contributing members will receive a zero. In case of disputes regarding contribution, a GSI or professor may interview group members.

Students may be "fired" from a group by the majority vote of the remaining members. The procedure for this is as follows: (1) documented "gentle warning" of risk of firing in e-mail, with cc to all group members and to eecs482staff@umich.edu, with cause and specific work required to remain in group; (2) allow at least 72 hours for compliance; (3) send documented statement of firing in e-mail, with cc to all group members and to eecs482staff@umich.edu. Fired group members will receive a reduced grade for the project from which they are fired, and must actively pursue and obtain membership in another group. The fired student must send e-mail to eecs482staff@umich.edu and the new group members when re-hired. Students that don't get re-hired or cannot find partners will need to work alone and be graded on the same curve as the other groups. It is in your interest not to get fired by your group.

Managing group dynamics and effectively using each group member's time and talents can be as difficult as solving the project. We are happy to offer advice on how to handle these issues. Be open and candid with your group about any potential problems early on so that your group can plan around such problems and not fall behind. A sure way to make your group upset at you is not finishing your part of the work at an agreed-upon deadline *and* not informing them about the problems early enough for them to help. We encourage everyone to read the note *Coping with Hitchhikers and Couch Potatoes on Teams*, posted on the course web site.

5.2 Turning in Projects

You will be submitting your projects electronically by running a program called submit482. Projects are due at 6:00 pm on the due date. To account for short-term unexpected events like computer crashes, submission problems, and clock skew, we will allow 6 hours of slack and accept projects until exactly 11:59 pm. Sometimes unexpected events make it difficult to get a project in on time. For this reason, each student will have a total of 3 late days to be used for projects throughout the semester. **These late days should only be used to deal with unexpected problems such as illness**. They should not be used simply to start later on a project or because you are having difficulty completing the project. Projects received after the due date (assuming that you have no late days left) will receive a zero, even if it is just one second late. **DON'T CUT IT TOO CLOSE—one zero can have a drastic effect on your grade!** Try to save one or two late days for the last project. Weekend days are counted in the same way as weekdays (e.g. if the project deadline is Friday and you turn it in Sunday, that's two days late). Each submission by any group member counts as a submission by all members of the group (i.e. you can't use one member's 3 late days on one project and another member's 3 late days on another project; all late submissions get charged to all group members). Groups formed by merger will be credited with the fewest number of late days remaining to any group member.

Extension requests (other than the use of free late days) should be made before the original due date. Extensions will only be granted for medical or personal **emergencies**. All extension requests must be accompanied by written verification, for example a written note from your doctor. Extensions are not granted for reasons such as: the printer went down, you erased all your files, you lost your listing, the terminal room was crowded and you couldn't get a terminal, you had other course work or job commitments which interfered, etc.. You can avoid all these problems by starting the projects early and keeping backup files. If you are having trouble understanding the material or designing a program, please come to office hours for help right away. In most cases, your project group members will be expected to make up the deficit without needing an extension.

5.3 Doing your Own Project

All projects in this course are to be done by your own group. Violation will result in initiation of the formal procedures of the Engineering/LSA Honor Councils, as appropriate. We use an automated program and manual checks to correlate projects with each other and with prior solutions.

At the same time, we encourage students to help each other learn the course material. As in most courses, there is a boundary separating these two situations. You may give or receive help on any of the concepts covered in lecture or discussion and on the specifics of C++ syntax. You are allowed to consult with other students in the current class to help you understand the project specification (i.e. the problem definition). However, you may not collaborate in any way when constructing your solution—the solution to the project must be generated by your group working alone. You are not allowed to work out the programming details of the problems with anyone outside your group or to collaborate to the extent that your programs are identifiably similar. You are not allowed to look at or in any way derive advantage from the existence of project specifications or solutions prepared in prior years (e.g. programs written by former students, solutions provided by instructors, project handouts). If you worked on the projects in the past (e.g. if you are repeating EECS 482), you are not allowed to re-use code that you or your group wrote from the prior semester. If you have any questions as to what constitutes unacceptable collaboration, please talk to the instructor right away.

You are expected to exercise reasonable precautions in protecting your own work. Don't let other students borrow your account or computer, don't leave your program in a publicly accessible directory, and take care when discarding printouts.

5.4 Tips for Success in the Projects

The most common reason for not doing well on the projects is not starting them early enough. You will be given sufficient time to complete each project. However, if you wait until the last minute to start, you may not be able to finish. Plan to do some work on a project every day. Also plan to have it finished a few days ahead of the due date; many unexpected problems arise during programming, especially in the testing phase. The computing sites can become quite crowded as deadlines approach, making it difficult to get a computer. Plan for these things to happen. Your lack of starting early is not an excuse for turning in your project late, even if some unfortunate situations arise such as having your computer crash.

Late days should only be used to deal with unexpected problems such as computer crashes, illness, submission problems, or deadline conflicts. They should not be used simply to start later on a project or because you are having difficulty completing the project. Try to save one or two late days for the last project.

The most common reason for spending too much time on a project is hacking before thinking. Resist the urge to start banging out code as your first step; you are likely to code yourself into a corner. Sit down together with your project partners and plan out the architecture of your solution. Expect to revise this architecture several times before settling on a plan. Pay particular attention to the project descriptions, and generate a list of behaviors the specification requires of your solution. Design your project with independently and incrementally testable subsystems rather than save all testing for the end. Assign one project member as the Testing Czar; that member's job is to see how they can break the team's solution. The second most common reason for spending too much time on a project is hunt-and-peck debugging; trying things at random, just to see if they work. If you find yourself in this position, step away from the keyboard and think about what is happening, or come see us in office hours.

There are many sources of help from which you can draw. Most questions can be submitted to the GSIs, professors, and your fellow classmates via the course newsgroup (umich.eecs.class.482). These will typically be answered within the day, often more quickly during working hours. We ask that you **do not** pose questions via email, as you will not receive a response as quickly, and the rest of the students in the course do not benefit from the answer. If you have a question that is inappropriate for the newsgroup, (for example, you have a specific question about your own code, and cannot post the question publicly), please see a member of the course staff in person during office hours rather than send email.

Many computing sites have consultants available. They are fine sources of help with questions regarding the computers and installed software (such as Unix, news, and the C++ compiler). However, they are not likely to be able to help you with questions about computer programming, the C++ language, or specific errors in your program.

6 Exams

There will be two exams during the semester: an in-class midterm on March 6th and a cumulative final exam. **The final exam is not during the regularly scheduled time for either lecture.** It is tentatively scheduled to be held jointly, on April 21st from 7:30—9:30 PM. As soon as the Registrar approves the alternate time, I will announce as much to the class.

You are expected to take both exams at the scheduled times. Unless a documented medical or personal emergency causes you to miss an exam, you will receive a zero for that exam. If you anticipate conflicts with the exam time, talk to the instructor at least **1 month** before the exam date. The exam dates are given at the beginning of the term so you can avoid scheduling job interviews or other commitments on exam days. Outside commitments are not considered a valid reason for missing an exam.

7 Attendance and Discussion Sections

You are expected to attend lecture regularly, and to be at your discussion section weekly. Discussion section meetings will typically involve active participation, whether by discussion, group exercises, or Q&A sessions.

8 Grading Policy

Final grades will be based on the total points earned on the projects and exams. Grades will be assigned on a curve, consistent with past incarnations of 482. Factors such as class participation and the quality/completeness of your practice questions may be used to adjust your final grade if it falls near a borderline. The tentative point breakdown is as follows:

- Projects 45%
- Midterm 25%
- Final 30%

Incompletes will generally **not** be given. In accordance with university policy, doing poorly in a course is not a valid reason for an incomplete. If you are having problems in the course, your best bet is to come talk to the instructor as soon as you are able.

9 Computing Environment

You may use any Sun workstation that runs Solaris and g++. If you are an engineering ore declared LSA CS student, we recommend that you use the CAEN workstations, either directly or by logging into one from another computer. If you are not, you can pay to get access to CAEN resources, or you may use the CAEN/ITD work workstations (e.g. in the Media Union) or login to an ITD Sun workstation (login.itd.umich.edu).

10 Course Schedule

week	Tuesday	Thursday	Readings	Projects
6-Jan	course intro	thread intro	Ch 1	
13-Jan	too much milk	monitors	Ch 2-2.2.2, 2.2.7-2.4	P1 out, groups
20-Jan	semaphores	thread implementation	Birrell	
27-Jan	monitor implementation	deadlock	Ch 3	
3-Feb	scheduling	address spaces	Ch. 2.5, 7.4	
10-Feb	address translation	paging	Ch 4	
17-Feb	page replacement	kernel/user space		P1 due, P2 out
24-Feb	Spring Break	Spring Break		
3-Mar	review	midterm		
10-Mar	networking intro	TCP	Ch 8.3-8.3.3	
17-Mar	RPC	disks	Ch 8.2.4, Ch. 5.3, Ch. 5.4.3	P2 due, P3 out
24-Mar	file system layout	file system caching, DFS	Ch 6	
31-Mar	encryption	security		
7-Apr	security	special topics	Ch 9	
14-Apr	review	study days		P3 due
21-Apr	finals week	finals week		