Advanced Operating Systems
CS 63201 / 73201
Fall 2000, MSB 228, TU 1:45-3:00pm

Instructor Dr. Mikhail Nesterenko

email: mikhail@mcs.kent.edu
office: MSB 356
phone: 672-9109
web: http://www.mcs.kent.edu/~mikhail/
office hours: MWF 3:45-4:30pm or by appointment (email to make)

Course Prerequisites

CS 4/53201 Operating Systems or equivalent OS course. Familiarity with the basics of C++ is helpful since you will be modifying C++ code during your projects. Yet it is not mandatory since only a subset of C++ is used. Refer to class' webpage for the pointers to C++ tutorials.

Course Overview

The goal of this course is to provide an introduction to distributed operating systems. We will study the key elements of distributed operating systems: communication, remote procedure calls, distributed shared memory, synchronization, resource and process management, etc. To review the concepts students will be assigned several homeworks and three programming projects. There may be several quizzes. There will be three exams.

Textbook

required: *Distributed Operating Systems*, Sinha, IEEE Computer Society Press, 1997.
recommended: *Distributed Operating Systems*, Tanenbaum, Prentice Hall, 1995.

Class Web Page, Mailing List, Contacting the Instructor

The web page for the class is http://www.mcs.kent.edu/~mikhail/classes/aos.f00/ (I have a link to this page from my homepage). The web page will contain links to the following course materials:

- class syllabus;
- lecture notes;
- homework assignments, homework and exam's solutions;
- programming projects, information on Nachos operating system, C++ tutorials.

I have set up a mailing list to for the students taking this course. I am going to send announcements and other class related information to this list. It is very important to be on this list to get the latest news and updates about the class. **You have to subscribe to the mailing list yourselves.** How to subscribe to the mailing list is explained on the course's webpage.

The simplest way to contact me is via e-mail. If you need to talk to me in person - see me during my office hours or make an appointment via e-mail.

Lectures

Students are expected to attend each lecture. I will not take roll, yet attendance and active participation during a lecture will help you learn the material and succeed in class.

A substantial part of the notes for this course is drawn from material provided by professor Walker intend to complement his notes with additional material, which may not be included in the textbook.

I will put the notes of the lecture each morning before the class, you are welcome to print them out. I am also going to hand them out in class.

Homework Assignments and Programming Projects

There will be approximately 4 homework assignments and 2 programming projects during the semester. The homework assignments will be pencil-and-paper based, while the projects will be based on the Nachos instructional operating system, and will involve reading and writing C++ code – the programming language Nachos is written on.

Nachos Programming Projects

The Nachos instructional operating system is written in C++ (actually, a subset of C++ that uses classes and methods, but avoids troublesome C++ constructs like inheritance and overloading). If you need quick refresher on C++, check the class' webpage for references.

Late Policies

Late work will be accepted as stated below unless you make prior arrangements with me or have a documented illness (in which case I expect you to contact me as soon as possible).

In general, you will have adequate time to complete each assignment. However, you should begin working on each assignment early so that you will have plenty of time to become familiar with it and with the Nachos code that you must read and/or modify, and so that you have time to "sleep on" the difficult parts. Waiting to start coding until the night before the project is due is a bad idea.

No late homeworks will be accepted.

Late projects will be accepted. 10% of the grade is subtracted for each day the project is late.

Exams

There will be two exams (held during class) and a final exam (held during finals week). All exams are closed book and closed notes, and must be individual work. It is expected that you take each exam at the scheduled time, unless you make *prior* arrangements with me, or have a *documented* illness (in which case I expect you to contact me as soon as possible). You will be tested on the material I covered in class. The textbook alone may not be sufficient for adequate preparation for the exams.

Academic Integrity

Student-teacher relationships are built on trust. Students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments which students turn in are their own. Acts that violate this trust undermine the educational process. Academic dishonesty in any form will be penalized up to assigning grade F for the course.

Cooperation on Homework Assignments and Programming Projects

For both homework assignments and programming projects, I strongly believe that discussion with your peers is an excellent way to learn. If you don't understand something, discussing it with someone who does can be far more productive than beating your head against the wall.

Having advocated discussion, then, I must be about clear what is allowed, and what is not. In general, students are allowed to cooperate as follows: you are allowed to discuss with other students the assignment, and general methods for solving the assignment. However, you are not allowed to work with someone else to actually *solve* the assignment, or to *write code* (even pseudocode) for a program, and you are certainly not allowed to *copy* anyone else's solution; doing any of these things will be considered cheating, and will constitute grounds for failing the course.

Note that there is a fine line between discussion and cheating. If you are unsure what is allowed and what isn't, feel free to discuss the distinction with me, but if something feels uncomfortable, it's probably not allowed.

Finally, you should be careful not to give others access to your code. This means that you shouldn't keep your program in a publicly accessible directory, you shouldn't leave your terminal unattended, and you shouldn't forget to pick up your printouts.

Grades

Your final course grade will be broken down as follows:

- Homeworks (approximately 3) 50 points each
- Programming projects (approximately 2) 100 points each
- Exams (2) 100 points each
- Final exam 100 points

The sum of possible scores on all assignments is considered 100% and your final course grade will be determined as A = 90–100%, B = 80–99.99%, etc. There will be no curve at the end of the course, although individual exams, homeworks, etc. may occasionally (although rarely) be curved. Thus you should always be able to determine how well you are doing in the course.

Students with Disabilities

In accordance with University policy, if you have a documented disability and require accommodations to obtain equal access in this course, please contact me at the beginning of the semester or when given an assignment for which an accommodation is required. Students with disabilities must verify their eligibility through the Office of Student Disability. Services (SDS) in the Michael Schwartz Student Services Center (672-3391).