

Operating Systems

CS 33211

Fall 2004, MSB 115, TR 9:15-10:30am

Instructor **Dr. Mikhail Nesterenko**

e-mail: mikhail@cs.kent.edu
office: MSB 356
phone: 672-9109
web: <http://www.cs.kent.edu/~mikhail/>
office hours (tentative): TR 1:30-3:00pm

Course Prerequisites

CS 33001(Data Structures) and CS 35101 (Computer Architecture) or their equivalents. Familiarity with the basics of C++ is helpful since you will be modifying C++ code during your projects. Refer to class' webpage for the pointers to C++ tutorials.

Course Overview

The goal of this course is to provide an introduction to operating systems. We will study process management, threads, deadlocks, memory management, file system organization. If time permits we will also examine the networking and distributed aspects of OS design. You3 will use Nachos instructional operating system for several programming projects.

Textbook

Operating Systems Concepts, 6th edition, Silberschatz, Galvin and Gagne, John Wiley and Sons, 2001

Class Web Page, Mailing List, Contacting the Instructor

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

- course 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. The instructions how to subscribe to the mailing list as well as the list archive are linked to the course's mailing list. **You have to subscribe to the mailing list within the first two weeks of classes.** You can subscribe to the list from the account of your choice (it does not have to be Kent-State's university or departmental account.) You have to check your mail at least once a day while the school is in session.

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.

Class Assignments

Homeworks. There will be approximately 3 homework assignments during the semester. The homework assignments will be pencil-and-paper based and will involve solving problems from the textbook.

Quizzes. There will be approximately 3 quizzes. The date of the quiz is announced about a week in advance (there will be no surprise quizzes.) A quiz is held during the first 10 minutes of the class. Late students will not be given extra time to complete the quiz. A quiz usually contains 10 multiple-choice questions.

Programming projects. There will be approximately 3 programming projects. The projects will be based on the Nachos instructional operating system, and will involve reading and writing C++ code. 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 course's webpage for links to online tutorials.

The projects will be submitted and graded electronically. The details on project submission will be given to you together with project assignment.

Exams. There will be two exams (held during class) and a final exam (held during finals week.) You will be tested on the material I covered in class. The textbook alone may not be sufficient for adequate preparation for the exams.

Late Policies

- quizzes no late quizzes accepted, no make-up quizzes;
- exams no late exams, no make-up exams;
- homeworks no late homeworks;
- projects late projects accepted. 10% of the grade is subtracted for each day the project is late. For penalty calculation Saturday and Sunday are counted as one day.

Late work will be accepted as stated above. I may waive the late policy conditions only in case of a *documented* illness or some extraordinary circumstance. In either case you have to contact me immediately. With respect to projects and homeworks my decision to grant you a waiver is partially influenced by the degree of completion of the work assigned. For example, if the project is assigned for 2 weeks, by 10th day I expect you to complete 65-70% of the work.

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 for debugging which may take significantly longer than the initial code writing. Waiting to start coding until the night before the project is due is a bad idea.

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.

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 calculated as follows:

- | | |
|--|--|
| • quizzes (approximately 3) | 10 points each (the lowest quiz score will be dropped) |
| • programming projects (approximately 3) | 50 points each |
| • homeworks (approximately 3) | 20 points each |
| • exams(2) | 100 points |
| • final exam | 100 points |

The sum of possible scores on all assignments is considered 100% and your final course grade will be determined as follows – 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. Note, that this means that **your score will not be rounded up: if you get 69.99% you will get a D not a C**. Thus you should always be able to determine how well you are doing in the course.

You will provide me with a pseudonym. Your grades will be posted on the course's webpage under your pseudonym.

Students with Disabilities

University policy 3342-3-18 requires that students with disabilities be provided reasonable accommodations to ensure their equal access to course content. If you have a documented disability and require accommodations, please contact the instructor at the beginning of the semester to make arrangements for necessary classroom adjustments. Please note, you must first verify your eligibility for these through Student Disability Services (contact 330-672-3391 or visit www.kent.edu/sds for more information on registration procedures).