CS 3204 Operating Systems Spring 2002

Instructor:	Dr. Ben Keller 625 McBryde Hall 231-9367 keller@cs.vt.edu
Office Hours:	10:00-11:00 MWF 9:30-10:30 TTh
CRNs:	11370, 8:00-9:15 TTh, 126 McBryde 11371, 11:00-12:15 TTh, 126 McBryde
GTAs:	Vikas Aggarwal, Feng Zhang, Padmapriya Kandahai

GTA office hours will be held in 124 McBryde (the Unix lab). The schedule will be posted on the course website.

Course Description: This course covers both theoretical and practical issues underlying operating system design and implementation. Lectures focus primarily on theoretical and conceptual aspects of operating systems. Programming projects focus on the application of concepts and implementation details.

Prerequisites: CS 2604 and ECE 2504 Introduction to Computer Engineering. Computer Science majors and minors must have completed CS 2604 with a final grade of C or better. ECE majors must have passed 2604 with a final grade of C- or better.

Text: Nutt, Gary, *Operating Systems: A Modern Perspective*, Addison Wesley Longman, 2000.

Course Web Page: The course web page will be the primary source of information regarding assigned work in class. You should check the web page frequently. The web page is `http://courses.cs.vt.edu/~cs3204/spr02/keller/`.

Grading and Evaluation: Your grade in this course will be based on evaluation of homework, projects, a midterm test and a final exam. The final grade for the course will be determined using the following distribution.

Homework	15%
Projects	45%
Midterm test	15%
Final exam	25%

- Homework — There will be a number of homework assignments that may consist of questions requiring written answers or programming solutions. All homework solutions must be submitted electronically to the curator system as either a Word or PDF document unless otherwise required by the assignment. In some cases, homework may involve programming and may be autograded by the curator system. In most cases, however, the GTA will grade the homework solutions.

- Projects — There will be five projects in this course. All projects will be submitted electronically to the Curator system. Many of your programs will be evaluated by demonstration to the GTA. (The programming standards used for evaluation are given on the web page.)
- Examinations — The midterm test will cover roughly half of the material in the course and is tentatively scheduled for 26 February. The final exam for 11370 (8 a.m.) will be given on 8 May from 10:05 to 12:05 p.m.; and for 11371 (11 a.m.) on 6 May from 4:25 to 6:25 p.m. The final exam will be comprehensive.

Grading Policies: The follow policies are the rules by which we will operate this semester.

- Missed work — No make up work will be allowed for the course. If you have a formal excuse from a dean (either the Dean of Student's office, or the dean of your college), your final exam grade will be substituted for your midterm grade.
- Late work — Only projects may be submitted late, and then for a penalty of 10% per weekday up to a maximum of three days late. Saturday or Sunday submissions will be counted as a submission on the following Monday, and so will have a one day penalty. The time that the Curator records for the submission will be the official submission time, and while we will be liberal in interpreting the exact due date, you should not rely on this. No other work will be accepted late.
- Hardware/software failure, power failure, etc. — No extensions will be given for failure of your personal hardware or software. Extensions will also not be given for problems with CS lab machines. Extensions may be given for widespread outages of power or networks, but are not likely.
- Demonstration grading — some projects may be evaluated by demonstration to a GTA. Demonstrations must be given within two weeks of the due date, if a project is not graded within that time period, it will not be evaluated. A project that is not demonstrated will not receive a grade.
- Compiler restrictions — All program grading will be done on Mandrake Linux using Gnu G++ 3.0.X (the lab machines will have 3.0.3).
- Programming Standards — Simply having a program that produces the correct results is not sufficient in this course; it must also correctly implement the data structure, adhere to the programming standards of the course, and be well designed. The programming standards are given on the course web page. These standards apply to all projects, unless specifically stated in the assignment.

  Note that the standards that are used here are different than those used in previous courses. When developing systems software our emphasis will be on efficient resource usage rather than reusability.
- Appealing grades — Questions or complaints about grading should be addressed to the person who graded the work; however, the instructor is the ultimate authority on grading policy for the course.

- Appeal Deadline — Any questions or complaints regarding the grading of an assignment or exam must be raised within one week of the return of the graded work by the instructor or GTA. This is not necessarily the same as when you pick it up.

- Final grades – Letter grades will be assigned according to the usual 10-point scale: with 90% guaranteeing an A-, 80% guaranteeing a B-, and so forth. A curve may be applied to determine the final letter grades at the instructor's discretion.

Honor Code: *All graded assignments must be your own work.* The instructor and GTAs are available for assistance on completing solutions to homework and programming assignments. Assistance from anyone else regarding the specifics of your solution is a violation of the honor code.

Special Accommodations: Any student requiring special accommodations is asked to meet with the instructor in the first week of classes.