

CMSC 412 (Spring 1998)

Professor:	TA:
Dr. Jeff Hollingsworth	Simon Hawkin
4161 AV Williams	1151 A V Williams
(40) 5-2708	
hollings@cs.umd.edu	cema@cs.umd.edu
Office Hours:	Office Hours:
Tu 10:45-12:00	TBA
W 11:00-12:00	

Class URL: <http://www.cs.umd.edu/~hollings/cs412/s98/>

You are expected to check the class web page on a regular basis (at *least* weekly).

Catalog Description:

A hands-on introduction to operating systems, including topics in: multiprogramming, communication and synchronization, memory management, IO subsystems, and resource scheduling policies. The laboratory component consists of constructing a small kernel, including functions for device IO, multi-tasking, and memory management.

Objective:

An in-depth understanding of how an operating system manages resources in a computer and provides programmers with a machine and device independent interface. The emphasis of this class will be on operating system concepts. Running examples will be drawn from contemporary OS's including UNIX and Windows (NT and 95).

Prerequisites: CMSC 311, CMSC 330

Topics Covered (in approximately the order we will cover them):

- Introduction to Operating Systems (1 week)
- Concurrent Processes (2 weeks)
- Kernel implementation techniques (1 week)
- CPU scheduling (1 week)
- Memory Management (2 weeks)
- File and I/O Systems (2 weeks)
- Security and Protection (1 week)
- Networking and Distributed Systems (2 weeks)
- Objects and Naming (1 week)
- Window and Display Services (1 week)

Required Course Text:

Operating System Concepts 4th Edition, Siberschatz and Galvin, Addison-Wesley 1994.

Programming Projects:

Understanding operating system concepts is a hands-on activity. This class will include several substantial programming projects that will require students to read and understand provided code, write new modules, and debug the resulting system. **The programming assignments will be time consuming and students taking this class should plan their class schedules accordingly.**

The instructor reserves the right to **fail**, regardless of overall numeric score, students who do not submit a good faith attempt to complete all programming assignments.

Grading:

Final Exam	30%
Midterms (2 each worth 15%)	30%
Programming Assignments	40%

Exams:

Midterm #1 - March 10 in class
Midterm #2 - April 21 in class
Final - ???

Re-grade policy. All requests to change grading of homework, programming projects, or exams must be submitted in writing (typed) within one week of when the assignment was made available for pickup. Requests must be specific and explain why you feel your answer deserves additional credit. A request to re-grade an assignment can result in the entire assignment being re-evaluated and as a result the score of *any part* of the assignment may be increased **or lowered** as appropriate.

Academic Integrity:

All work that you submit in this course must be your own; unauthorized group efforts will be considered academic dishonesty. See the Undergraduate Catalog for definitions and sanctions. Academic dishonesty is a serious offense that may result in **suspension or expulsion** from the University. In addition to any other action taken, the grade “XF” denoting “failure due to academic dishonesty” will normally be recorded on the transcripts of students found responsible for acts of academic dishonesty. Sharing of code on programming assignments is a form of academic dishonesty.

Late Policy:

No late homework or programming assignments will be accepted. If you are unable to complete a programming assignment due to illness or family emergency, please see Dr. Hollingsworth as soon as possible to make special arrangements.

Office Hours and Email:

I am happy to answer questions during office hours, and by email. However, office hours and email are not intended as a replacement for lecture. As a result, I will only see people during office hours or respond to your email if you regularly attend class.