

# Syllabus

## CS 471: Operating Systems

### Old Dominion University

#### Spring 2006

**Meeting times:** 11:00-12:15, Tuesdays & Thursdays, Hughes 1117

**Instructor:** C. Michael Overstreet  
Office: E&CS 3206  
Phone: 683-4545  
E-mail: cmo@cs.odu.edu

**Office hours:** 2:00--3:30, Tuesdays & Thursdays. Other hours by appointment.

**Text:** Silberschatz, Galvin, Gagne *Operating Systems Concepts 7th ed.*, John-Wiley, 2005.  
Other material will be made available on-line.

**Prerequisites:** Intro to Computer Architecture (CS 270 or ECE 443), Data structures (CS 361). Unless you have a C or better of these classes, you cannot register for this class.

<b>Grading weights:</b>	Exams (2)	30%
	Final	25%
	Homework, small projects	15%
	Term project	30%

The final exam is comprehensive and will be used for the grade of any missed exams. Project grading is described below.

**Project:** The project will be done in two/three person teams and consists of several phases. Each phase has several deliverables: size estimates, time estimates, test plans, design, *fully documented* source code, test results, a programming report and a project report. For each component, team members will be responsible for different deliverables and tasks; these will change throughout the semester. Individual grades for each component will depend on both the component grade (a group grade) and the individual responsibilities of the team member. *Note:* code will not be graded unless commented as specified in assignment documents.

**Lateness policy:** To receive full credit, each project component must be submitted by midnight of the due date. Lateness penalties are computed as follows:

1 to 3 days	10% per day
4 to 7 days	50%
8 to 14 days	70%
no credit after 2 weeks	

No project component will be accepted after two weeks without a doctor's written excuse or *prior* permission of the instructor. Since many of the students work full-time, special arrangements for due dates can be made, but they *must be worked out in advance with the instructor*.

**Honor code:** All work turned in for credit, including exams and project components, is to be the work of the person whose name is on the exam or project. For all project components, the student can receive assistance from individuals other than the instructor only to ascertain the cause of errors. Thus you can get help if you need it to figure out why something doesn't work. You just can't get help from anyone, other than the instructor or a fellow team member, to figure out how to make something work. All solutions turned in for credit are to be your individual work and should demonstrate your problem solving skills, not someone else's. If a team member provided significant help, this should be noted. This text should appear on all assignments: "I pledge that I have neither given nor received help from anyone other than the instructor and team members for all project components included here." Each student should sign his or her name under the pledge. Any

deviation from this policy is an honor code violation and will be treated as such if detected. It should be included as a comment in your source code and part of your written reports."

**Special needs:** if you have special needs (visual, physical or whatever), please make these known to the instructor so that appropriate accommodations can be made.

### Tentative Schedule

(Subject to change)

Week	Topic	Reading	Homework	Project
Jan. 10	Class policies. OS objectives: efficiency, reliability, ease of use, architectures	Ch. 1	1.1, 1.5, 1.7, 1.10	
Jan. 17	OS structures, Virtual machines	Ch. 2	2.2, 2.3, 2.4, 2.5, 2.9	Group requests due Thurs. midnight
Jan. 24	OS Services, system calls, virtual machines, system design Project discussion, trace demo	Ch 3	3.5, 3.7, 3.8, 3.10, 3.13, 3.14	Ph. 1: Process attributes code validation Code review form
Jan. 31	Processes, IPC, RPC, Posix Project, ph. 1	Ch. 4, 5	4.4, 4.6, 5.1, 5.2, 5.6	
Fed. 7	CPU scheduling, synchronization, critical sections, semaphores, monitors	Ch. 6, 7	6.1, 6.3, 6.6, 6.7, 6.9 7.1, 7.2, 7.7, 7.8	
Feb. 14	Deadlocks, characterization, prevention, avoidance	Ch. 8	8.1, 8.2, 8.4, 8.6, 8.13	Ph. 2: Project design, design review, size and time estimates
Feb. 21	Exam 1			
Feb. 28	Memory, logical vs physical addresses, paging, segmentation	Ch. 9	9.2, 9.4, 9.5, 9.7, 9.9, 9.10, 9.11, 9.16, 9.17	Ph. 3 Minimal implementation, part 1, test reports
Mar. 7	Spring break			
Mar. 14	Virtual memory, page replacement, thrashing	Ch. 10	10.2, 10.3, 10.4, 10.5, 10.9, 10.10, 10.11	
Mar. 21	File system access, management **	Ch. 11, 12	11.1, 11.2, 11.7, 11.12, 12.1, 12.2, 12.4, 12.8	Ph. 4: Minimal implementation, part 2, test reports
Mar. 28	I/O Systems, secondary storage	Ch. 13, 14	13.1, 13.2, 13.10, 14.1, 14.2, 14.8, 14.10, 14.29	
Apr. 4	Distributed systems, file systems	Ch. 15, 16	15.5, 16.1, 16.5	Ph. 5: Complete implementation, test reports
Apr. 11	Exam 2	Ch. 17	17.3,	
Apr. 18	Protection, security	Ch. 18, 19	18.2, 18.12, 19.1	Ph. 6: Experimentation data complete
Apr. 25	Linux, Win XP overviews	Ch. 20, 21	20.1, 20.11, 21.2, 21.4	Ph. 7: Project report
May 2	Final exam, 12:30-3:30			