

CSci 530

OPERATING SYSTEMS

Fall, 2014

Tentative Course Syllabus

INSTRUCTOR:

Derek Harter, Ph.D. Assistant Professor, Department of Computer Science
Texas A&M University - Commerce
Office Hours: Tue, Wed, Thur (1:30pm to 5:00pm)
Office: JOUR 204
E-mail:derek.harter@tamuc.edu

CLASS MEETINGS:

01W

80476

Web Based Class

COURSE OBJECTIVES:

The course objectives are two-fold:

- To learn general theory, concept, and techniques related to the design of operating systems,
- To practice the design of an operating system by performing programming exercises of various OS components and principles.

COURSE DESCRIPTION:

General theory and concept behind operating system design are discussed in this course. Topics include operating system structures, memory management, process scheduling, process synchronization and communication, deadlocks, and case studies of other commercially available operating systems.

TEXTBOOKS:

Operating Systems Internals and Design Principles --- 7th Edition
by William Stallings Prentice-Hall Inc., 2011, ISBN-10:013230998X

RECOMMENDED READING:

Operating System Concepts --- 7th Edition by A. Silberschatz and P. Galvin
John Wiley & Sons, Inc., 2006, 0-471-69466-5 ISBN

TOPICS AND CHAPTERS TO BE COVERED:

- | | |
|-------|--|
| (10%) | Part One: Overview (Chapters 1 and 2) |
| | ▶ Computer system overview ₁ |
| | ▶ Operating system overview ₂ |
| (40%) | Part Two: Processes (Chapters 3,4,5, and 6) |
| | ⇒ Process description and control ₃ |
| | ⇒ Threads ₄ |
| | ⇒ Concurrency I: Mutual exclusion ₅ |
| | ⇒ Concurrency II: Deadlock and starvation ₆ |
| (25%) | Part Three: Memory (Chapters 7 and 8) |
| | ▶ Memory management ₇ |
| | ▶ Virtual memory management ₈ |
| (25%) | Part Four: Scheduling (Chapters 9 and 10) |
| | ⇒ Uniprocessor scheduling ₉ |
| | ⇒ Multiprocessor scheduling ₁₀ |

EXAMS: Exam 1 (Chapters 1, 2, 3, 4, 5)
Final Exam (comprehensive – Chapters 1-10)

COURSE REQUIREMENTS:

There will be regularly assigned homework problems. There will be programming assignments, which will require the students to spend time in the computer laboratory. To plan a minimum of three hours of outside preparation for each hour of class is a safe time allocation for successfully completing the course. Due dates for all assigned materials will be announced in class in advance. It is the student's responsibility to have all assignments ready on time. Any student who has to be absent on an assignment due date must arrange to have the assignment submitted early. Late assignment may not be accepted. Additional requirements of the course include a number of quizzes, homework assignments and tests.

STUDENTS WITH DISABILITIES:

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you have a disability requiring an accommodation, please contact: Office of Student Disability Resources and Services, Texas A&M University-Commerce, Gee Library, Room 132, Phone (903) 886-5150, StudentDisabilityServices@tamuc.edu

ACADEMIC ETHICS:

"All students enrolled at the University shall follow the tenets of common decency and acceptable behavior conducive to a positive learning environment." (See Student's Guide Handbook, Policies and Procedures, Conduct). Ethics also includes the issue of plagiarism, and copying code for programming assignments is just as serious as any other type of plagiarism. If you are caught sharing or using other people's work in this class, you will receive a 0 grade and a warning on the first instance. A subsequent instance will result in receiving an F grade for the course, and possible disciplinary proceedings.

ATTENDANCE POLICY:

Students are expected to follow all instructions and visit eCollege regularly many times weekly to complete the materials for this online course. If a student is unable to submit assignments by the due date for the assignment, they are expected to make alternative arrangements to assure that the assignment is turned in ON TIME, before the assignment is actually due. Any student wishing to withdraw from the course must do so officially as outlined in the class schedule. THE INSTRUCTOR CANNOT DROP OR WITHDRAW ANY STUDENT.

COURSE REQUIREMENT DEADLINES:

Credit will be given for ONLY those exam(s), program(s), and/or project(s) turned in no later than the deadline(s) as announced by the instructor of this class unless prior arrangement has been made with the instructor.

METHOD OF EVALUATION (TENTATIVE):

Two tests	(50% ⇒ 25% each)
Written Problem Sets	(25%)
Programming Assignments	(25%)

Final average	Letter grade
90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D

Course Outline (preliminary):

LEC		CHAPTERS	SUBJECT
1	8/25	1	Computer Systems Overview
2	9/1	2	Operating Systems Overview
3	9/8	3	Process Description and Control
4	9/15	4	Threads, SMP and Microkernels
5	9/22	4	Threads, SMP and Microkernels
6	9/29	5	Concurrency I: Mutual exclusion & Synchronization
7	10/6	5	Concurrency I: Mutual exclusion & Synchronization
		TEST	Test 1
8	10/13	6	Concurrency II: Deadlock & Starvation
9	10/2	0	7 Memory management
10	10/27	8	Virtual memory
11	11/3	8	Virtual memory
12	11/10	9	Uniprocessor Scheduling
13	11/17	9	Uniprocessor Scheduling
14	11/24	10	Multiprocessor and Real-Time scheduling
15	12/1	10	Multiprocessor and Real-Time scheduling
		12/8-12/13 Final Exam	Mandatory

Assignment Schedule (preliminary):

Wk	Date	Assg	Desc
2	9/5	Prob Set 1	Program Execution in Hypothetical Machine problem set
3	09/12	Prob Set 2	Operating System Achievements problem set
5	10/06	Prog 1	Process Simulator
8	10/18	Prob Set 3	Concurrency problem set
10	10/31	Prog 2	Page replacement simulator
11	11/08	Prob Set 4	Virtual Memory problem set

Student Learning Outcomes

1. List and understand basic functions and parts of an OS.
2. Understand modern memory management techniques, including virtual memory.
3. Know fundamental concepts of OS such as multiprogramming and multiuser systems.
4. Understand process management algorithms, structures and threading.
5. Understand issues with concurrent and parallel programming, including deadlocks.
6. Learn specific mechanisms for modern OS such as Linux and Windows Vista.

Learning outcomes will be measured through mapping assignment and test questions to specific outcome items, as well as through exit surveys of student experiences with the outcome familiarity.