CSCI 530 – OPERATING SYSTEMS

Semester: Spring 2016
Time: Online

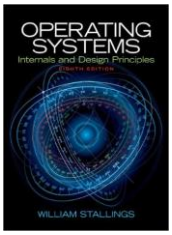
Instructor: Dr. Oner Ulvi Celepcikay
E-mail: Oner.Celepcikay@tamuc.edu
Tel: (713) 966-9354
Office Hours: will be posted on ECollege

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.

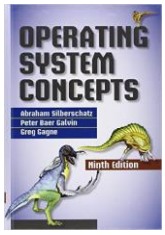
COURSE DESCRIPTION:
General theory and concept behind operating system design are discussed in this course. Topics include operating system structures, memory management, process scheduling, process synchronization and communication, deadlocks, and case studies of other commercially available operating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

TEXTBOOKS:



Required Book (OS): *Operating Systems Internals and Design Principles*

7th Edition by William Stallings Prentice-Hall Inc., 2011, ISBN-10:013230998X



Recommended Book (OS): *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 (Ch 1)
  - Operating system overview (Ch 2)
- (40%) Part Two: Processes (Chapters 3,4,5, and 6)
  - Process description and control (Ch 3)
  - Threads (Ch 4)
  - Concurrency I: Mutual exclusion (Ch 5)
  - Concurrency II: Deadlock and starvation (Ch 6)
- (25%) Part Three: Memory (Chapters 7 and 8)
  - Memory management (Ch 7)
  - Virtual memory management (Ch 8)
- (25%) Part Four: Scheduling (Chapters 9 and 10)
  - Uniprocessor scheduling (Ch 9)
  - Multiprocessor scheduling (Ch 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):
Course Grade: Course grades will be determined as follows:

Activity	Weight
Midterm Exam [Chapters 1, 2, 3, 4, 5]	25%
Final [All Covered Materials]	25%
Written Problem Sets	20%
Programming Assignments / Papers	20%
Quizzes	5%
Class Participation	5%

Final average Letter grade

90 – 100 → A
80 – 89 → B
70 – 79 → C
60 – 69 → D
Below 60 → F

Course Outline (preliminary):

LECTURE	CHAPTERS	SUBJECT
1	1	Computer Systems Overview
2	2	Operating Systems Overview
3	3	Process Description and Control
4	4	Threads, SMP and Microkernels
5	4	Threads, SMP and Microkernels
6	5	Concurrency I: Mutual exclusion& Synchronization
7	5	Concurrency I: Mutual exclusion & Synchronization
MIDTERM TEST		**Test 1**
8	6	Concurrency II: Deadlock & Starvation
9	7	Memory management
10	8	Virtual memory
11	8	Virtual memory
12	9	Uniprocessor Scheduling
13	9	Uniprocessor Scheduling
14	10	Multiprocessor and Real-Time scheduling
15	10	Multiprocessor and Real-Time scheduling

Final Exam

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 pocess 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 familarity.