Shanghai Jiao Tong University

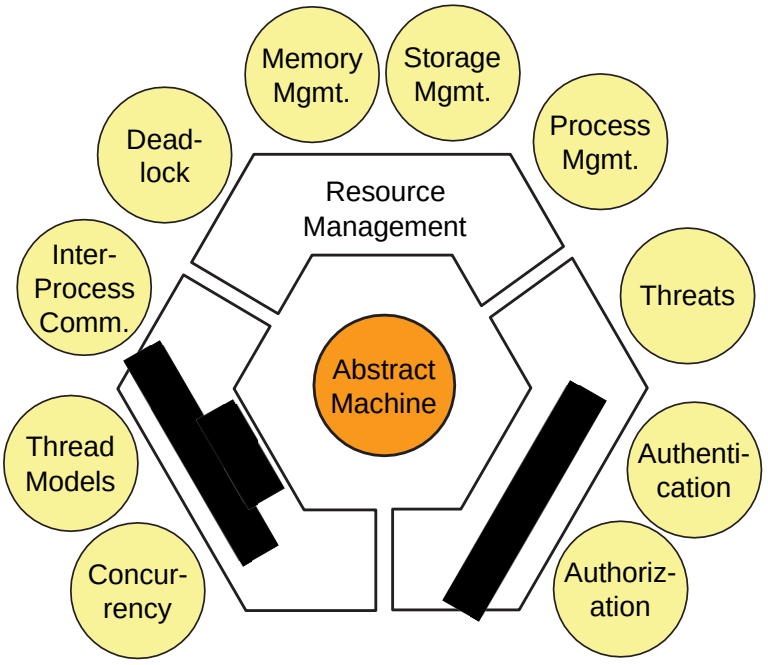
VE482 Operating System

Instructor Information	TBD		
Term	June 28 2021- July 29 2021	**Credits**	4 units
Class Hours	Monday through Thursday, 120 mins per teaching day		
Discussion Sessions	2 hours each week, conducted by teaching assistant(s)		
Total Contact Hours	66 contact hours (1 contact hour = 45 mins, 3000 mins in total)		
Required Texts (with ISBN)	NOT Required textbook: Silberschatz and Galvin, Operating System Concpets, Essentials, John Wiley & Sons, 2011, Hoboken, New Jersey.		
Prerequisite	Data Structure Programming Language (C/C++/Java/Python) Computer Organization		
The course might be moved to online delivery due to COVID-19 pandemic. Students will be notified once the decision is made.			

Course Overview

This is an introductory course in computer operating systems. In this course we will study the theoretical and practical concepts behind modern operating systems. In particular, we will study the basic structure of an operating system, its components, design strategies, algorithms and schemes used to design and implement different components of an operating system. Major components to be studied include: processes, inter-process communication, scheduling, memory management, virtual memory, storage management, network management, and security.

Concept Map



Grading Policy

Activity	
Programming Assignment 1:	5%
Programming Assignment 2:	5%
Programming Assignment 3:	5%
Programming Assignment 4:	5%
Midterm	20%
Written Assignments	10%
Final (comprehensive)	50%

Research Project: This is available for advanced students (who have prior exposure to OS and C programming). We need to agree on the topic in the first week of the classes. You can do this project instead of the normal programming assignments.

Double Grading Policy: This course has a significant portion of the grade allocated for the programming component. You are expected to submit ***only*** your work in these assignments. You can receive advise or tips from others (instructor, teaching assistants, or peers), but the final submission should be yours. You are expected to know all the design decisions in the program and explain all aspects of the program handed in as part of the assignment. To test this condition, we will randomly select some students and ask them to explain their programming assignments. The eventual marks for an assignment will be the minimum of the two marks. For example, if 85 is the marks obtained in the first (normal) evaluation of the programming assignment and 50 is the marks obtained in the second evaluation, then effectively you have 50.

Late Assignment Policy: There will be two deadlines for each assignment: proper deadline and cut-off date. After the proper deadline, there will be a penalty of 10% for each day the assignment is late until the cut-off date. After the cut-off date, the assignment cannot be handed in. No individual requests for extensions will be granted unless they are for medical reasons.

Regrading Policy: If you find your assignments or exams are not marked according to the marking scheme, you are encouraged to consult me or the TAs. When you resubmit your assignment or exam for regarding, we reserve the right to regrade the full exam or assignment without restricting the attention to the disputed portion.

Teaching Method

The course will consist of three hours of instructor led classes together with two hours of tutorials per week taken by the TAs. The class time will be devoted to the presentation and development of new concepts and the application of these concepts to examples and problems, while the tutorials will discuss solutions to the programming projects and written assignments. The primary focus of the tutorials is to provide sufficient “how-to” knowledge through the discussion of the assignments to help in the development of the programming project series.

Grading Scale

Number grade	Letter grade	GPA
90-100	A	4.0
85-89	A-	3.7
80-84	B+	3.3
75-79	B	3.0
70-74	B-	2.7
67-69	C+	2.3
65-66	C	2.0
62-64	C-	1.7
60-61	D	1.0
≤59	F (Failure)	0

Class Schedule

Date	Lecture
Day 1	Introduction to OS: definition and organization of OS
Day 2	OS Structure: OS services, system calls, design of an OS
Day 3	Process: definition and status of process, context switch
Day 4	Thread: why multi-threads? Multithread programming
Day 5	CPU Scheduling: why CPU scheduling? FCFS, SJF, Priority methods
Day 6	CPU Scheduling: Round-Robin method, implementation of scheduling methods in real OS & On class project I
Day 7	Process Synchronization: critical section, Peterson's solution
Day 8	Process Synchronization: Semaphores, classic process synchronization problems
Day 9	Main Management: physical and logical address, allocation
Day 10	Midterm
Day 11	Main Management: page tables, TLBs, segment
Day 12	& On class project II
Day 13	Virtual Memory: role of virtual memory, dynamic loading
Day 14	Virtual Memory: page replacement algorithms
Day 15	File systems Interface: definition of files and directories, naming, mounting & On class project III
Day 16	File systems Implementation: file control block, allocation methods of a file
Day 17	Mass Storage Systems: Magnetic disk performance, disk scheduling, NAS/SAN, disk arrays
Day 18	System Protection and Security: protection policies and cryptography
Day 19	& On class project IV
Day 20	Distributed Systems: distributed network, distributed file systems