



Shanghai Jiao Tong University

VE482 Operating System

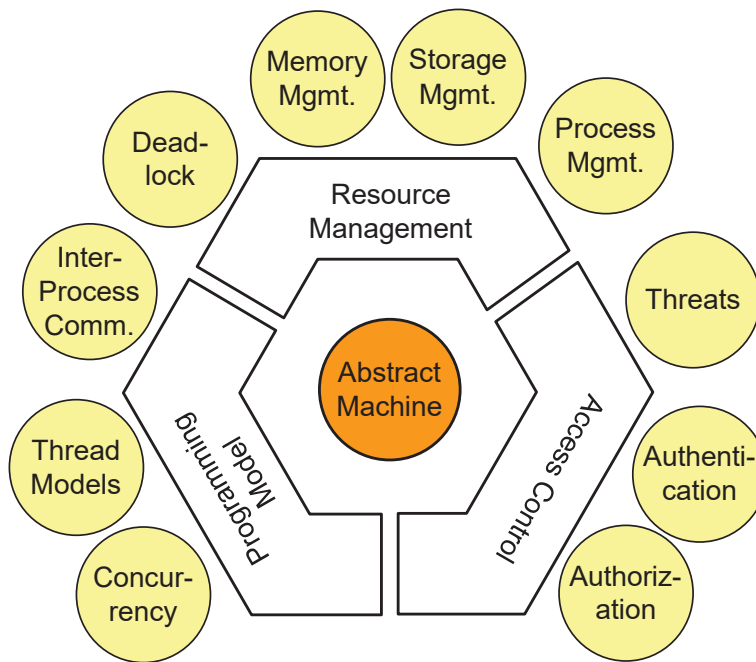
Instructor:	Chentao Wu	Email:	wuct@cs.sjtu.edu.cn
Home Institution:	Shanghai Jiao Tong University	Office:	TBD
Office Hours:	TBD		
Term:	2 July – 2 August, 2018	Credits:	4 units
Classroom:	TBD	Teaching Assistant(s):	TBD
Class Hours :	Monday through Thursday, 5.30-7.30 pm		
Discussion Session:	2 hours each week, conducted by teaching assistant(s)		
Total Contact Hours:	66 contact hours (1 contact hour = 45 mins, 3000 mins in total)		
Required Texts (w/ ISBN):	NOT Required textbook: Silberschatz and Galvin, <i>Operating System Concepts</i> , Essentials, John Wiley & Sons, 2011, Hoboken, New Jersey.		
Prerequisite:	N/A		



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%
Total	100%



NOTE: The official programming language of this course is C.

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.



Course Schedule

The table below shows a tentative course schedule (I will make a “best effort” to stick to the proposed schedule).

Date	Topics
2-Jul	Introduction to OS: definition and organization of OS
3-Jul	OS Structure: OS services, system calls, design of an OS
4-Jul	Process: definition and status of process, context switch
5-Jul	Thread: why multi-threads? Multithread programming
9-Jul	CPU Scheduling: why CPU scheduling? FCFS, SJF, Priority methods
10-Jul	CPU Scheduling: Round-Robin method, implementation of scheduling methods in real OS
11-Jul	Process Synchronization: critical section, Peterson’s solution
12-Jul	Process Synchronization: Semaphores, classic process synchronization problems
16-Jul	Main Management: physical and logical address, allocation
17-Jul	Midterm
18-Jul	Main Management: page tables, TLBs, segment
19-Jul	Virtual Memory: role of virtual memory, dynamic loading
23-Jul	Virtual Memory: page replacement algorithms
24-Jul	File systems Interface: definition of files and directories, naming, mounting
25-Jul	File systems Implementation: file control block, allocation methods of a file
26-Jul	Mass Storage Systems: Magnetic disk performance, disk scheduling, NAS/SAN, disk arrays
30-Jul	System Protection and Security: protection policies and cryptography
31-Jul	Distributed Systems: distributed network, distributed file systems
1-Aug	Case studies and Summary: windows & Linux
2-Aug	Final Exam