King Saud University
College of Computer and Information Sciences
Department of Computer Science

CSC 227 – Operating Systems (3-0-1) - Required Course
Spring 2026 (II)

Catalog description

This is an introductory course in Operating Systems. As such, it is intended to cover many of the concepts related to most of the actual Operating Systems. Although the study of a particular Operating System is out of the scope of this course, nevertheless, we will cover most of the concepts found in any existing Operating System. We will review computer system and operating system structures, processes and threads (concepts of, communication, synchronization and deadlocks), CPU Scheduling, memory management and virtual memory.

Prerequisite

Data Structures (CSC 212)

Prerequisite to

CSC 329 Computer Networks, CSC328 Systems Programming, CSC453 Parallel Processing.

Textbook

Operating System Concepts, Abraham Silerschatz, Greg Gagne, and Peter Baer Galvin, **Tenth Edition**, John Wiley & Sons Inc., ISBN 978-1-119-32091-3

Course Objectives

The objective of the course is to provide the concepts of operating systems design and implementation. It identifies and describes the major and common components of an operating system with stating their functions and purposes especially process management (process scheduling, and synchronization), and memory management (segmentation, paging, and swapping).

	Course learning Outcomes (CLOs)	PLO Code
1	**Knowledge and Understanding**	
1.1	Describe and explain the role and fundamental components of a modern operating system.	K1
1.2	Describe how operating systems manage computing resources and relate the use of system calls, APIs, interrupts, and protection rings to providing services to users and application software.	
1.3	Explain the concepts of processes and threads, and compare their effects on performance.	
1.4	Compare and contrast between process scheduling algorithms, identify problems, apply possible solutions, and examine performance implications.	K2
1.5	Identify critical sections and apply constructs of process synchronization like locks and semaphores.	

1.6	Illustrate different memory management schemes, including contiguous allocation, paging and swapping.	
2	**Skills**	
2.1	Design and implement some components of modern operating systems.	S1
2.2	Function effectively as a member or leader of a team engaged in the design, development and implementation of operating system SW components.	S3

Exam dates

Midterm examination: As per college schedule
Final examination: As per college schedule

Evaluations (Tentative)

Activity	%	Note
Midterm Exam	30%	*
Project/Programming Assignments	15%	
Quizzes	15%	5 Quizzes No make up for missed quiz
Final Exam	40%	

Course policies

- Student will be denied from entering the final exam if he exceeds 25% absence rate.
- No late submission is accepted.
- Your answers or programs will NOT be judged only by the final results/conclusions.
- Plagiarism/Cheating and Incompletes: Please, see the King Saud University policies on plagiarism and incompletes.

Instructor

Yazied Hasan
(yhasan@ksu.edu.sa)
https://faculty.ksu.edu.sa/ar/yhasan\

Topic	Sections	No of Weeks	Contact Hours
Chapter 1: Introduction	1.1 What Operating Systems Do 1.2 Computer-System Organization 1.3 Computer-System Architecture 1.4 Operating-System Operations 1.5 Resource Management 1.6 Security and Protection 1.7 Virtualization 1.8 Distributed Systems 1.9 Kernel Data Structures 1.10 Computing Environments 1.11 Free and Open-Source Operating Systems	2	6L + 2T
Chapter 2: System Structures	2.1 Operating-System Services 2.2 User and Operating-System Interface 2.3 System Calls 2.4 System Services 2.5 Linkers and Loaders 2.6 Why Applications are Operating System Specific 2.7 Operating-System Design and Implementation 2.8 Operating-System Structure 2.9 Building and Booting an Operating-System 2.10 Operating System Debugging	2	6L + 2T
Chapter 3: Processes Concept	3.1 Process Concept 3.2 Process Scheduling 3.3 Operations on Processes 3.4 Inter-process Communication (only 3.4.1)	1.5	5L + 2T
Chapter 4:Multi-threading Programming	4.1 Overview 4.2 Multicore Programming 4.3 Multithreading Models 4.4 Thread Libraries 4.6 Threading Issues	1.5	3L+T
Chapter 5: Process Scheduling	5.1 Basic Concepts 5.2 Scheduling Criteria 5.3 Scheduling Algorithms 5.5.1 Multi-Processor Scheduling	2	6L + 2T
Chapter 6: Synchronization	6.1 Background 6.2 The Critical-Section Problem	1.5	5L+2T

Tools	6.3 Peterson's Solution 6.4 Hardware Support for Synchronization 6.5 Mutex Locks 6.6 Semaphores 6.6 Liveness		
Chapter 9: Memory-Management Strategies	9.1 Background 9.2 Contiguous Memory Allocation 9.3 Paging 9.4 Structure of the Page Table 9.5 Swapping 9.6 Example: Intel 32- and 64 bit Architectures 9.7 Example: ARM Architecture	1.5	5L+2T
Chapter 9: Virtual Memory	10.1 Background 10.2 Demand Paging, 10.4 Page Replacement	1	2L+T