

Operating Systems

Prepared by Dr. Alaa Eldeen Ahmed

IT263 / CYB510: Course Overview



Operating Systems Course Overview

Course Description

This course introduces the fundamental concepts of **Operating Systems** and their role in modern computing systems.

It covers the structure and functionality of operating systems, including **process management, CPU scheduling, memory management, file systems, synchronization, and virtualization.**

The course combines **theoretical concepts** with **practical examples** to help students understand how operating systems efficiently manage hardware resources and support application execution.

By the end of the course, students will have a solid foundation in operating system principles

Operating Systems Course Overview

Learning Outcomes:

Upon successful completion of this course, students should be able to:

- Evaluate computer architecture and organization in relation to operating systems design.
- Interpret and explain operating system processes in detail.
- Compare and contrast different techniques for memory management.
- Compare various approaches to process scheduling.
- Examine and assess issues involved in operating systems controlling the input/output function.
- Assess threats and mechanisms for providing operating system protection.
- Communicate the use of operating system virtualization

Assessment Strategy:

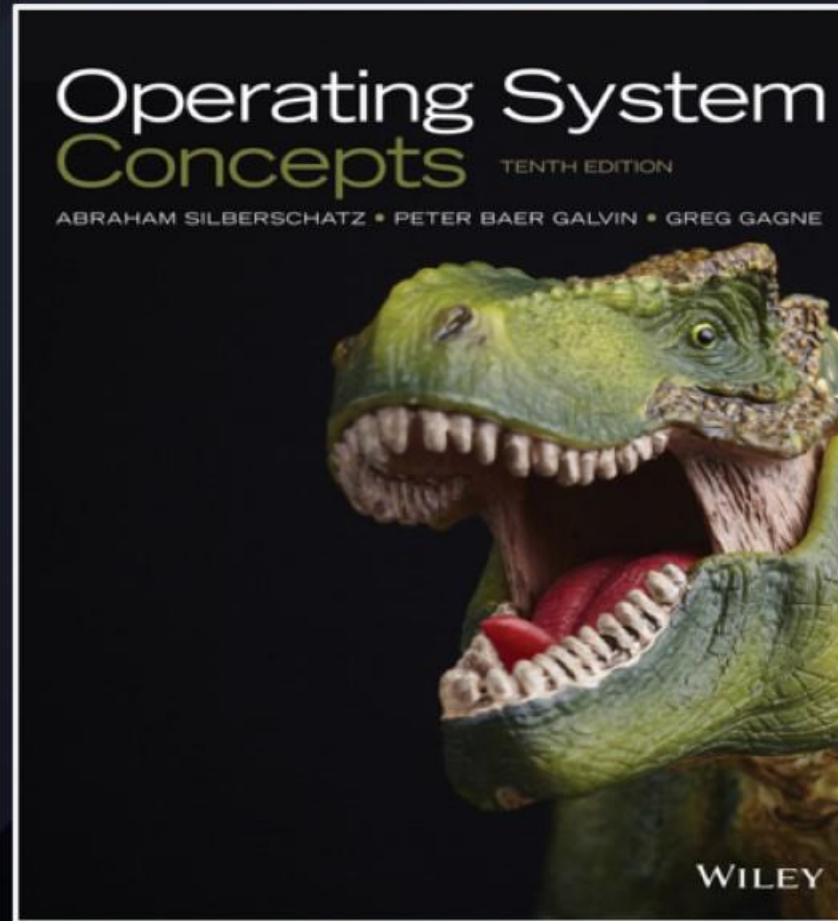
- The assessment strategy measures the student's understanding of Operating systems and their components.
- The assessment is mainly implemented using Examinations to indicate student's progress to master the course content, it includes:
 - Weekly quizzes to measure the students understanding to the topics covered in that week.
 - An end-of-semester final examination to measure the student's mastery in understanding the knowledge and design skills gained through the whole course.

Operating Systems Course Overview

- **Credit Hours:** 3 credit hours.
- The required textbook for this course is "Operating System Concepts, " by Silberschatz, Galvin, and Gagne. Most of the material in these course was originally taken from the 9th and 10th edition of that book.
- Lecture notes, power points slides and videos are additional resources that are prepared by me a supplementary learning resources.

Course Overview

Operating Systems Course Overview



- **Introduction to Operating Systems:** The course begins with a general introduction to operating systems.
- **Major Components of an OS:** Across 10 modules, each major component of an operating system is discussed and analyzed in detail, from processes to threads.

Operating Systems Course Overview

- Module 1: Operating System Introduction.
- Module 2: Processes
- Module 3: Thread
- Module 4: Scheduling Algorithms
- Module 5 Concurrency and synchronization tools.
- Module 6: Deadlock.
- Module 7,8 : Memory management.
- Module 9 : File Systems.
- Module 10 : Virtualization and protection.

- **Module 1 : Operating Systems Introduction**
 - **Basic Computer System Organization:** Provide an overview of the fundamental structure and components of a computer system.
 - **Major Operating System Components:** Offer a comprehensive tour of the key components and functions of an operating system.

- **Module 2 : Processes**

- **Process Concept:** Introduce the notion of a process as a program in execution, including its states and attributes.
- **Process Scheduling:** Explain how the operating system schedules processes to efficiently share CPU resources.

- **Module 2 : Processes**
- **Operations on Processes:** Describe the creation, execution, suspension, and termination of processes.
- **Inter-process Communication:** Discuss mechanisms that allow processes to exchange data and synchronize their actions.

- **Module 3 : Threads**
 - **Motivation for Using Threads:** Explain how threads improve application responsiveness, resource sharing, and overall system performance.
 - **Multithreading Models:** Describe different models for implementing threads and managing their execution within an operating system.
 - **Threading Issues:** Discuss challenges related to multithreading, such as synchronization, communication, and scheduling.

- **Module 4 : CPU Scheduling**
 - **Introduction to CPU Scheduling:** Explain CPU scheduling as a fundamental concept that enables multiprogramming in operating systems.
 - **CPU-Scheduling Algorithms:** Describe the various algorithms used to schedule CPU time among competing processes.
 - **Evaluation Criteria:** Discuss the criteria for evaluating and selecting an appropriate CPU-scheduling algorithm for a specific system.

Module 5 : Concurrency and synchronization tools.

- **Background:** Introduces the need for synchronization in concurrent systems to ensure correct and consistent access to shared resources.
- **The Critical-Section Problem:** Addresses the challenge of designing protocols that allow processes or threads to safely access shared data without interference.

Module 5: Concurrency and synchronization tools.

- **Software and Hardware Solutions:** Discusses mechanisms such as software-based algorithms and hardware-supported instructions to achieve mutual exclusion.
- **Semaphores:** Presents synchronization primitives used to control access to shared resources and coordinate process execution.

Module 5 : Concurrency and synchronization tools.

- **Classical Problems of Synchronization:** Examines well-known synchronization scenarios, such as the Producer–Consumer, Readers–Writers, and Dining Philosophers problems.

- **Module 6 : Deadlocks**
 - **System Model:** Describes the system resources, processes, and their interactions, forming the basis for analyzing deadlock conditions.
 - **Deadlock in Multithreaded Applications:** Examines how concurrent threads competing for shared resources can lead to deadlock situations.
 - **Deadlock Characterization:** Identifies the necessary conditions for deadlock, including mutual exclusion, hold and wait, no preemption, and circular wait

Module 6 : Deadlocks

- **Methods for Handling Deadlocks:** Presents the general strategies for managing deadlocks, such as prevention, avoidance, detection, and recovery.
- **Deadlock Prevention:** Eliminates one or more necessary conditions for deadlock to ensure that deadlocks cannot occur.

• **Module 6 : Deadlocks**

- **Deadlock Avoidance:** Dynamically allocates resources based on system state to ensure the system remains in a safe state.
- **Deadlock Detection:** Allows deadlocks to occur but periodically checks the system to identify deadlocked processes.
- **Recovery from Deadlock:** Describes techniques for restoring normal operation after a deadlock, such as process termination or resource preemption.

- **Module 7,8 : Memory Management**

- **Background:** Introduces fundamental memory management concepts and the need for efficient allocation of main memory in operating systems.
 - **Contiguous Memory Allocation:** Allocates a single, continuous block of physical memory to a process, offering simple management but potentially leading to fragmentation.
-

- **Module 7,8 : Memory Management**
 - **Background:** Introduces fundamental memory management concepts and the need for efficient allocation of main memory in operating systems.
 - **Contiguous Memory Allocation:** Allocates a single, continuous block of physical memory to a process, offering simple management but potentially leading to fragmentation.

- **Module 7, 8 : Memory Management**
- **Non-Contiguous Memory Allocation:** Allows a process to be allocated memory in separate physical locations, improving memory utilization and reducing fragmentation through techniques such as paging and segmentation.

- **Module 7, 8 : Memory Management**
 - **Non-Contiguous Allocation (Paging):** Divides physical memory into fixed-size frames and logical memory into pages, allowing processes to be stored non-contiguously while enabling efficient memory management.
 - **Translation Lookaside Buffer (TLB) in Paging:** A high-speed cache that stores recent virtual-to-physical address translations to reduce memory access time.

- **Module 7, 8 : Memory Management**
 - **Inverted Page Table:** A memory-efficient page table structure that maintains a single table indexed by physical frames rather than by virtual pages.
 - **Demand Paging:** Loads pages into memory only when they are needed, reducing memory usage and improving system performance.
 - **Page Replacement:** Determines which memory pages should be swapped out when a page fault occurs and available memory is limited.

• **Module 9 : File Systems**

- **File Organization and Access:** Define how files are structured, stored, and accessed efficiently to support reliable data retrieval and modification.
- **File Directories:** Provide mechanisms for naming, organizing, and locating files within the file system.
- **File Sharing:** Enable controlled access to files by multiple users or processes while preserving data integrity and security.

• **Module 10 : Operating System protection and Virtualization**

- **Goals of Protection:** Ensure that system resources are accessed only by authorized users or processes, preventing misuse, errors, and security breaches.
- **Protection Rings:** Define hierarchical levels of privilege, where higher-privilege rings control access to critical system operations and resources.
- **Domain of Protection:** Specifies the set of resources and the types of operations that a process or user is allowed to perform.

Module 10 : Operating System protection and Virtualization

- **Access Matrix:** Provides a structured model that represents the access rights of different domains over system objects.
- **Access Control Methods:** Enforce protection policies using mechanisms such as Access Control Lists (ACLs) and capability-based systems.
- **Objectives:** Maintain system security, integrity, reliability, and controlled sharing of resources

- **Module 10 : Operating System protection and Virtualization**
 - **Virtualization – Overview:** Introduces the concept of virtualization and its role in modern computing.
 - **Benefits and Features:** Highlights the advantages and key characteristics of virtualization.
 - **Types of Virtual Machines and Their Implementations:** Describes different types of virtual machines and how they are implemented in practice.

End of Lecture