Resource Person:

Module:

Course Title: Operating Systems

Course Code:

Course Type: Core Course

Pre-Requisite: Introduction to Computer Programming

Counseling Hours: Weekdays 4:00-6:00 pm

Program: MCS/MIT

Program Head: Imran Saleem

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Signature** | **Date** |
| **Prepared By**  (Resource Person) |  |  |  |
| **Checked By**  (Program Head) |  |  |  |
| **Approved By**  (Director SPA) |  |  |  |

**Course Description:**

This course presents to make participants have an understanding of the design issues of different aspects of operating systems. To make participants have an in depth understanding of the various OS services for threads, inter-process communication, process synchronization, process and memory management, and file systems, offered as system or library calls in UNIX and Linux operating systems.

**Course Instructional Objectives**

* The main instructional goal is to inculcate in depth understanding of the various OS services for threads, inter-process communication, process synchronization, process and memory management, and file system offered as system or library calls in UNIX and Linux OS.
* This will be achieved through lectures, quizzes, and assignments

**Course Student Objectives**

* To describe the basic organization of computer systems
* To provide a grand tour of the major components of operating systems
* Understand the basics of threads.
* Inter-Process Communication.
* Process Synchronization.
* Memory Management.
* Compare different file systems being used in different operating system.

**Course Contents**

Following is the session-wise breakup of the course:

**Session 1: Introduction**

* One-to-one introduction
* Course Introduction, Teaching & Assessment Methodology
* Distribution of Course outlines & Discussion on Course Outlines
* Computer System Overview
  + Microprocessor evolution
  + Instruction execution
  + Interrupts and signals
  + Memory hierarchy
  + Cache memory
  + Direct memory access
  + Multiprocessor and multicore organization

**Learning Outcomes**

The main learning objective of the reading is to provide a basic and conceptual understanding of the course domain.

**Activities**

* Group formation.

**Session 2: Operating Systems Overview**

* Components, Services and Structure of Operating System
* Objectives and functions
* Evolution of operating systems
* OS design considerations
* Overview of different operating systems

**Learning Outcomes**

The main learning objective of the reading is to provide a basic and conceptual understanding of Operating System Structures.

**Activities:**

* Assignment 1

**Session 3: Process Description and Control**

* Definition
* States
* Description
* Process control
* OS execution
* Introduction to UNIX/Linux Interface
* UNIX process management
* UNIX/Linux Inter process communication tools

**Learning Outcomes**

The main learning objective of the reading is to provide an understanding of how processes are handled by an operating system.

**Activities**

* Assignment 2
* Group Activity

**Session 4: Threads**

* Processes and threads
* Types of threads
* Multicore and multithreaded systems
* Thread Management case study

**Learning Outcomes**

In this session students will learn about threads and multithreaded environments and how an operating system manages these threads.

**Activities**

* Assignment 3
* Group Activity

**Session 5: Concurrency: Mutual Exclusion and Synchronization**

* Principles of concurrency
* Hardware support
* Semaphores
* Monitors
* Message passing
* Readers/writers problem

**Learning Outcomes**

This session will focus on concurrency and synchronization between different processes. The students will learn how an operating system makes sure data and hardware are available to every process.

**Activities**

* Group Activity

**Session 6: Concurrency: Deadlocks and Starvation**

* Principles of deadlocks
* Deadlock prevention
* Deadlock avoidance
* Deadlock detection
* An integrated deadlock strategy
* Dining philosophers problem
* Critical Section Problems and Solutions

**Learning Outcomes**

This session will build on the concepts from the previous lecture and provide a deep understanding of what deadlocks are and how they pose problems for the normal execution of operating systems as well as how to deal with such situations.

**Activities**

* Group Activity
* Assignment 4

**Session 7: CPU Scheduling: Uniprocessor Scheduling**

* Basic Concepts
* Types of processor scheduling
* Process Scheduling Algorithms
* Traditional UNIX scheduling

**Learning Outcomes**

The main learning objective of the reading is to provide an understanding of CPU scheduling algorithms.

**Activities**

* Assignment 5
* Group Activity

**Session 8: CPU Scheduling: Multiprocessor Scheduling**

* Multiprocessor and multicore scheduling
* Real time scheduling
* Scheduling in Linux, UNIX and Windows based systems

**Learning Outcomes**

Building on concepts from the previous lecture, this session will focus on multiprocessor machines and how they handle scheduling in real time.

**Activities**

* Discussion

**Session 9: Midterm**

**Session 10: Memory Management**

* Introduction
* Memory partitioning
* Paging
* Segmentation
* Demand Paging
* Page Replacement Algorithm
* Loading and linking programs

**Learning Outcomes**

This session focuses on basic concepts of memory management and memory organization.

**Activities**

* Discussion
* Group Activity

**Session 11: Virtual Memory**

* Hardware control and structures
* Operating system software
* Memory Management in
  + UNIX
  + Linux
  + Windows

**Learning Outcomes**

This session will be focused on topics related to real-time memory management and virtualization. The students will learn the memory management schemes of different operating systems as well.

**Activities**

* Assignment 6

**Session 12: I/O Management and Disk Scheduling**

* I/O devices
* Organization of I/O functions
* OS Design Issues
* I/O buffering
* Disk scheduling
* RAID
* Disk cache

**Learning Outcomes**

This session will focus on issues related to I/O and how the operating system manages it all.

**Activities**

* Discussion
* Group Activity

**Session 13: File Management**

* File organization and access
* B-trees
* File directories
* File sharing
* File protection and allocation
* Record blocking
* Secondary storage management

**Learning Outcomes**

This session will focus file management in an operating system along with some practical examples of how it is done.

**Activities:**

* Group activity

**Session 14: Embedded Operating Systems**

* Embedded systems
* Characteristics of an embedded operating systems
* Embedded Linux
* Tiny OS

**Learning Outcomes**

In this session students will be introduced to another type of operating systems: the embedded operating systems. The students will learn what the similarities and differences a general purpose OS and an embedded OS.

**Activities:**

* Discussion
* Group Activity

**Session 15: Virtual Machines**

* Approaches to virtualization
* Processor issues
* Memory management
* I/O management
* VM examples

**Learning Outcomes**

This session will introduce students to virtual machines and different issues that need to be addressed while virtualizing an operating system.

**Activities:**

* Discussion

**Session 16: Final Term Exam**

**Recommended Book (s) & Text:**

* *Textbook:* Operating System Concepts (Silberschatz, Galvin, Gagne)
* Modern operating system (Andrew S. Tanenbaum 3rd edition)
* Operating Systems(William Stallings)

**E-Resources:**

* Class Slides and reference material handouts will be available on LMS.

**ASSESSMENT METHODOLOGY**

|  |  |
| --- | --- |
| Class Participation | 10 |
| Group Activities\quizzes | 20 |
| Assignments | 20 |
| Mid Term Exam | 20 |
| Final Term Exam | 30 |
| Total | 100 |

* The chapter slides will be uploaded on LMS a week before the class, it is the responsibility of the students to read the chapter at least once before coming to the class to get a better understanding of the concepts being discussed.
* Use of unfair means in any of the assessment activities is strictly forbidden. Anyone who violates this rule will be dealt with strictly per the UMT policy.
* Group activities will be done in groups of 3 people. Students that are absent during the activities will be marked zero.
* Class participation marks will be given based on healthy and constructive discussion done in class. This includes asking and answering questions logically keeping the course objective in mind.
* Any disruptive behavior which may result in distractions for the rest of the class will result in marks deductions from class participation.

**CALENDAR OF ACTIVITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Session** | **Sub-Topic** | **Readings** | **Activities** |
| 1 | **Introduction**   * One-to-one introduction * Course Introduction, Teaching & Assessment Methodology * Distribution of Course outlines & Discussion on Course Outlines * Computer System Overview   + Microprocessor evolution   + Instruction execution   + Interrupts and signals   + Memory hierarchy   + Cache memory   + Direct memory access   + Multiprocessor and multicore organization | Chapter 1 | Group Formation |
| 2 | **Operating Systems Overview**   * Components, Services and Structure of Operating System * Objectives and functions * Evolution of operating systems * OS design considerations * Overview of different operating systems | Chapter 2 | Assignment 1 |
| 3 | **Process Description and Control**   * Definition * States * Description * Process control * OS execution * Introduction to UNIX/Linux Interface * UNIX process management * UNIX/Linux Inter process communication tools | Chapter 3 | Assignment 2  Group Activity 1 |
| 4 | **Threads**   * Processes and threads * Types of threads * Multicore and multithreaded systems * Thread Management case study | Chapter 4 | Assignment 3, Group Activity 2 |
| 5 | **Concurrency: Mutual Exclusion and Synchronization**   * Principles of concurrency * Hardware support * Semaphores * Monitors * Message passing * Readers/writers problem | Chapter 5 | Group Activity 3 |
| 6 | **Concurrency: Deadlocks and Starvation**   * Principles of deadlocks * Deadlock prevention * Deadlock avoidance * Deadlock detection * An integrated deadlock strategy * Dining philosophers problem * Critical Section Problems and Solutions | Chapter 7 | Assignment 4  Group Activity 4 |
| 7 | **CPU Scheduling: Uniprocessor Scheduling**   * Basic Concepts * Types of processor scheduling * Process Scheduling Algorithms * Traditional UNIX scheduling | Chapter 6 | Assignment 5  Group Activity 5 |
| 8 | **CPU Scheduling: Multiprocessor Scheduling**   * Multiprocessor and multicore scheduling * Real time scheduling * Scheduling in Linux, UNIX and Windows based systems | Chapter 6 |  |
| 9 | **MIDTERM** | | |
| 10 | **Memory Management**   * Introduction * Memory partitioning * Paging * Segmentation * Demand Paging * Page Replacement Algorithm * Loading and linking programs | Chapter 8 | Group Activity 6 |
| 11 | **Virtual Memory**   * Hardware control and structures * Operating system software * Memory Management in   + UNIX   + Linux   + Windows | Chapter 9 | Assignment 6 |
| 12 | **I/O Management and Disk Scheduling**   * I/O devices * Organization of I/O functions * OS Design Issues * I/O buffering * Disk scheduling * RAID * Disk cache | Chapter 13 | Group Activity 7 |
| 13 | **File Management**   * File organization and access * B-trees * File directories * File sharing * File protection and allocation * Record blocking * Secondary storage management | Chapter 11 | Group Activity 8 |
| 14 | **Embedded Operating Systems**   * Embedded systems * Characteristics of an embedded operating systems * Embedded Linux * Tiny OS | Chapter 17 | Group Activity 9 |
| 15 | **Virtual Machines**   * Approaches to virtualization * Processor issues * Memory management * I/O management * VM examples | Chapter 16 |  |
| 16 | **FINALTERM** | | |