| | | |
|---|---|---|
| **Program** | ADP Data Science | |
| **Course Code** | CC-311 | |
| **Course Title** | Operating Systems | |
| **Credit Hours** | **Theory** | **Lab** |
| | 3 | 1 |
| **Lecture Duration** | 90 minutes (1.5 Hours), 2 lectures per week, 3 hours lab session per week | |
| **Semester** | 5 | |
| **Pre-requisites** | **Courses** | **Knowledge** |
| | Data Structures and Algorithms | |
| **Follow Up Courses** | System Programming | |

**Course Learning Outcomes (CLOs)**

| CLO No | Course Learning Outcome | Bloom Taxonomy |
|---|---|---|
| CLO-1 | Acquire the basic knowledge of computer organization computer architecture and assembly language. | C2 (Understand) |
| CLO-2 | Understand the concepts of basic computer organization, architecture, and assembly language techniques | C2 (Understand) |
| CLO-3 | Solve the problems related to computer organization and assembly language | C3 (Apply) |

| | |
|---|---|
| **Aims and Objectives** | 1. To understand the internals of operating system and practically access its services to have a clear understanding of the working of OS Kernel |
| **Learning Outcomes** | • Understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems (Understand)<br>• Analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions (Evaluate)<br>• Demonstrate the knowledge in applying system software and tools available in modern operating systems (Demonstrate) |

| | |
|---|---|
| **Syllabus** | Operating systems basics, system calls, process concept and scheduling, inter-process communication, multithreaded programming, multithreading models, threading issues, process scheduling algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks, memory management, swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files, file systems, file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management, system protection, virtual machines, operating system security |
| **Contents** | **Section 1:**<br>- Introduction<br>- Intro to Linux Environment<br>- Program v/s Process - Process Management<br>**Section 2:**<br>- I/O Redirection and IPC - Thread Management<br>**Section 3:**<br>- Introduction to Synchronization<br>- S/W-based and H/W-based CSP Solutions<br>- Synchronization using Semaphore<br>- Synchronization using Monitor -<br>Deadlocks<br>**Section 4:**<br>- Memory<br>- Paging |
| | - Virtual Memory<br>**Section 5:**<br>- Disk Geometry and Partitioning<br>- Disk Formatting and File System Monitoring<br>- File-System Architecture<br>- File Permissions |
| **Teaching-learning Strategies** | • Lectures<br>• Case Studies<br>• Project<br>• Assignments |
| **Assignments** | Types and Number with calendar |
| **Textbooks** | A. Operating System Concepts, by Galvin, Gagne, 10th Edition Silberschatz, Published in 2019, ISBN- 978-1-118-06333-0 |

| | |
|---|---|
| **Reference Material/Suggested Readings** | B. Modern Operating Systems, by Andrew S. Tanenbaum, 4th edition, Published in 2016, ISBN- 9789332575776<br>C. Operating Systems, Internals and Design Principles, by William Stallings, 9th edition,Published in 2017, ISBN-13: 978-0134670959<br>D. Dr. Muhammad Arif Butt, OS -Video Lectures: https://www.youtube.com/c/LearnWithArif/playlists |
| **Notes** | |

**Detailed Lecture wise plan**

| Week | Lecture | Topic | Source Book (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| 1 | 1 | **Introduction to course**, pre-requisite, policies, tools, and grading system. What is an Operating System (OS) and why it is needed to manage h/w? Operating System services, interrupts, traps and signals. Dual mode operations and protection mechanism. Types of operating systems and computing environments | Text A-Ch1 | |
| | 2 | **Introduction to virtualization and hypervisors**. Installing Linux (Ubuntu, Kali, CentOS) on Virtualbox. Introduction to Linux command line interface, Linux File Hierarchy Standard and basic shell commands. Linux system call interface. Compiling a C program on Linux command line interface | Text A-Ch1 | |
| 2 | 3 | **Editors used in Linux** (vim, peco, nano). Shell commands (wc, sort, uniq, grep, cut, paste, comm, comp, diff, whereis, which, locate, find, tar, gzip, gunzip). Program on disk and its components. Viewing contents of a program file. Process in memory and its components (stack, heap and PCB).<br>Command line arguments and environment variables. Viewing contents of a running program using readelf and objdump | Handouts | |

| Week | Lecture | Topic | Source Book (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| | 4 | **CPU and I/O bound processes**. Process state models. Five, six and seven state process models. Process scheduling queues. Long term, medium term and short term schedulers. Concept of process/context switch. Process creation and termination. Shell commands related to process management. Fork, wait and exit system calls. Interrupt, trap and system calls. Process resource limits. | Text A-Ch2 | |
| 3 | 5 | **Discussion on working of a Linux shell** and the concept of how an internal and external command executes. Running programs in the background and foreground. Switching programs to different states. Basic commands related to process states like ps, fg, bg and top. | Text A-Ch3 | |
| | 6 | **The open, read, write and close** paradigm in Linux. The concept of PPFDT. The connection of an opened file from process PPFDT to System Wide File Table, to I-node table and finally to disk blocks. Cooperating Processes. Taxonomy of Inter-process Communication | Handouts | **Lab:** |
| 4 | 7 | **I/O Redirection, UNIX IPC tools.** Using pipes, FIFOs and signals in Linux | Text A-Ch2 | |
| | 8 | **Concurrent and parallel programming**. Introduction to threads. Multi-threading. Merits and demerits of threads. User level vs Kernel level threads. Threading models. Programming using Pthread library. | Text A-Ch4 | **Lab:** |
| 5 | 9 | **Process scheduler and Dispatcher.** Preemptive vs non-preemptive scheduling. CPU and IO bursts. CPU scheduling and scheduling criteria, FCFS, SJF, SRTF, and Priority scheduling. | Text A Ch6 | |
| | 10 | **Round Robin,** Virtual Round Robin, Multi level Queue Scheduling and Multi level Feed- back queue scheduling, Rotating Stair-case Dead line scheduler, | Text A-Ch6 | |

| Week | Lecture | Topic | SourceBook (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| 6 | 11 | **Rotating Stair-case dead line scheduler,** UNIX SVR3 scheduling algorithm. Changing process priorities using nice, renice commands. Changing nice value of running processes and executing a program with a nice value other than the default. The concept of hard and soft CPU affinity in Linux. Displaying and changing the scheduling parameters of Linux processes using schedtool | Handouts | |
| | 12 | **Introduction to synchronization,** Concurrency Control, Race Condition, Critical Section Problem. Concept of atomic operation. General format of a CS problem solution. Characteristics of a good CSP solution. | Text A-Ch5 | |
| 7 | 13 | **Software Based Solutions to CSP:** Dekker solution, Peterson solution, and Leslie Lamport's Bakery algorithm. Concept of busy waiting. | Text A-Ch5 | |
| | 14 | **H/W based solutions to CSP:** Disabling of interrupts, TSL and swap instructions. Thread synchronization using pthread_mutex_t variable and pthread_mutex_lock() and pthread_mutex_unlock() library calls | Text A-Ch5 | |
| 8 | 15 | **Introduction to semaphores.** Binary and counting semaphores. Achieving mutual exclusion using semaphores. Achieving serialization using semaphores. Solution to Standard Synchronization problems using semaphores, Producer Consumer, Dining Philosopher, Reader writer, Sleeping Barber, Smokers problem | Text A-Ch5 | |
| | 16 | **Limitations of semaphores,** Introduction to Monitors, Condition variables, Hoare and Mesa monitors. Solution to standard synchronization problems using monitors. | Text A-Ch5 | |
| | 17 | **Introduction to Dead locks,** Four necessary and sufficient conditions for Dead Locks, Resource allocation graph, Dead lock handling methods, Dead lock prevention | Text A-Ch7 | |

| Week | Lecture | Topic | SourceBook (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| 9 | 18 | **Dead Lock Avoidance.** Bankers and Safety Algorithm.<br>Dead Lock Detection and Recovery Algorithms | Text A-Ch7 | |
| 10 | 19 | **Memory management,** address binding and linking, Logical vs Physical addresses, Dynamic loading, Dynamic linking and shared libraries. Overlays, swapping. Introduction to contagious memory allocation | Text A-Ch8 | |
| | 20 | **MFT and MVT,** Placement algorithms, Internal and External fragmentation, Buddy partitioning scheme. | Text A-Ch8 | |
| 11 | 21 | **Introduction to paging,** Page Table, Address translation in paging, Paging parameters for Intel and PDP11 | Text A-Ch8 | |
| | 22 | **Implementing page table in cache,** memory and CPU registers, Structure of Page Tables (Hierarchical, Inverted and Hashed Page tables). Introduction to Segmentation, address translation in segmentation. | Text A-Ch9 | |
| 12 | 23 | **Introduction to paged segmentation,** address translation in a paged segmentation. Address translation in Intel 80386 (Real and protected mode) | Text A-Ch9 | |
| | 24 | **Virtual Memory,** Background, Demand Paging, Performance of Demand Paging,<br>Page Replacement algorithms (FIFO, Optimal, LRU, LFU, MFU, Buffering) | Text A-Ch9 | |
| 13 | 25 | **Copy on Write protocol and vfork() system call,** Allocation Of Frames, Thrashing, Resident Set Management, Working Set Model, Page fault frequency, memory mapped files. | Text A-Ch9 | |
| | 26 | **Hard Disk Geometry:** Spinning and Solid state disk. Working of spinning disk and its interfaces (IDE, ATA, SATA, SCSI, SAS). The concept of Logical Block Addressing and its mapping on CHS address.<br>**Hard Disk Partitions:** Partitioning a hard disk. Different types of partition tables, MBR and GPT. Linux tools used for partitioning a hard disk like fdisk, gdisk, parted, gparted, cfdisk, sfdisk | Text A-Ch10 | |

| Week | Lecture | Topic | SourceBook (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| 14 | 27 | **Disk Formatting:** Concept of a file system and the basic functionalities that every file system should offer. Comparison of different file systems like ext2/3/4, reiserfs, hpfs, minix, ntfs, vfat, xfs and zfs. Use of Linux tools like mkfs, mke2fs, mkntfs, mkfs.fat, mkfs.minix to put a file system on a partition<br>**File System Mounting:** Introduction to the concept of file system mounting. Linux configuration files related to file system mounting. Linux commands like mount, umount, lsblk, blkid. Maintaining integrity of file system using Linux commands like<br>fsck, e2fsck, fsck.fat, fsck.nfs | Text A-Ch11 | |
| | 28 | **File System Architecture:** Schematic view of a standard UNIX file system. Describe the contents of boot block, super block, inode block, and data blocks. Discuss Inmemory and on-disk structures used by a file system. Describes what actually happens behind the curtain when a user creates, accesses and deletes a file and how Linux keep track of opened files by a process. Use of Linux commands like df, du, lsof, fuser, and tune2fs to perform these tasks<br>**Hard and Soft Links:** Discuss the use of hard and soft links on all UNIX based systems. Differences between hard and soft links. Use of Linux command ln to create hard and soft links | Text A-Ch12 | |
| 15 | 29 | **File Permissions:** Discuss the use of standard file permissions. How to change the existing file permissions on a file using symbolic and octal way. Use of chmod and chown commands. Setting the default file permissions on a newly created file using the umask command<br>**Special File Permissions:** Concept and use of Saved SUID bit on files. Concept and use of Saved SGID bit on files and directories. Concept and use of Sticky bit on files and directories. | Text A-C14 | |

| Week | Lecture | Topic | SourceBook (Ch#) | Recommendation for Learning Activities |
|---|---|---|---|---|
| | 30 | **Access Control Lists:** Discuss the security on files using Access Control List. Concept of Discretionary Access control and Mendatory Access control. How to set ACLs on files. A discussion on default ACLs or ACLs on directories<br>**Device Files:** Seven File Types in Linux and the concept of device files. Describes the contents of /dev/ directory. Describes Major and minor numbers and shows how you can create your own device files. Important /dev/ directory files like zero, null, full, random, urandom and ttys | Handouts | |
| 16 | 31 | **Terminal Attributes:** Overview of Terminal Devices and a comparison between disk and terminal files. Examine current attributes of terminal driver on a Linux machine and changing them using stty command. Overview of Canonical and Noncanonical mode of terminal drivers.<br>**Time Management in Linux operating system**<br>**Managing services using systemd:** Introduction to Linux system daemon. Overview of to systemd unit files, specially Target Unit Files and Service Unit Files. Shell commands to manage services using systemctl. Writing/running a basic service of your own | Handouts | |
| | 32 | **Log Files:** Logging mechanism in Linux<br>Booting process of a Linux system: A discussion on five phases of Linux Operating system: BIOS / UEFI Initialization, Master Boot Record, Boot Loader, Kernel Initialization, init or systemd Process | Text A-Ch18 | |