

22CS207 OPERATING SYSTEMS

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Knowledge of computers fundamentals, Computer organization & Digital logic and its design.

COURSE DESCRIPTION AND OBJECTIVES:

This course aims at concepts and principles of Operating Systems, its overall responsibility in acting as an interface between the system's hardware components and the user. Further, it also helps students to understand the different scheduling policies, process synchronization mechanisms, deadlock handling mechanisms and memory management techniques.

MODULE-1

UNIT-1

10L+0T+10P=20 Hours

LINUX FILE SYSTEM & PROCESS SCHEDULING

Introduction to LINUX File System: The LINUX file System, File System Hierarchy, File system Commands, File Attributes, File Permissions.

Filters: cmp, comm, diff, head, tail, find, cut, paste, sort, uniq.

Regular Expressions: grep, egrep, fgrep, Sed- line addressing, context addressing, text editing, substitution.

Introduction to Operating System: What Operating System do; Operating System Structure; Process concept-overview, Process Scheduling, Operations on Process; Inter Process Communication; Threads; Process (CPU) Scheduling-Scheduling Criteria, Scheduling Algorithms; Multiple-Processor scheduling;

UNIT-2

6L+0T+6P=12 Hours

PROCESS SYNCHRONIZATION AND DEADLOCKS

Process Synchronization: The critical-section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

Deadlocks: Deadlock characterization; Methods of handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery.

PRACTICES:

- Use the cat command to create a file containing the following data. Call it mytable.txt use tab to separate the fields.
1425 ravi 15.65
4320 ramu 26.27
6830 sita 36.15
1450 raju 21.86
- a. Use the cat command to display the file, mytable.txt.
 - b. Use the vim command to correct any errors in the file, mytable.txt.
 - c. Use the sort command to sort the file mytable.txt according to the first field.
 - d. Call the sorted file mytable.txt (same name)
 - e. Print the file mytable.txt.
 - f. Use the cut & paste commands to swap fields 2 and 3 mytable. Call it mytable.txt (same name)

- g. Print the new file, mytable.txt.
- Write a shell script that takes a command-line argument and reports on whether it is directory, a file, or something else.
 - Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.
 - Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
 - Write a shell script that computes the total and average marks of a student according to the following;
 - If average marks ≥ 69 then result is -Distinction.
 - If average marks ≥ 59 and ≤ 70 then result is -First Class.
 - If average marks ≥ 49 and ≤ 60 then result is -Second Class. If average marks ≤ 50 then result is -Pass.
 - Note that any subject marks ≤ 40 then result is -Fail.
 - Accept student name and six subject marks through the keyboard.
 - Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
 - Write a shell script, which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.
 - Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
 - Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
 - Implementation of new process creation and its communications.
 - Implement of thread creation and deletion.
 - Implementation of FCFS scheduling.
 - Implementation of SJF and RR Scheduling.
 - Implementation of producer consumer problem.
 - Implementation of Banker's algorithm for Dead lock avoidance.

MODULE-2

UNIT-1

8L+0T+8P=16 Hours

MEMORY MANAGEMENT

Memory Management: Basic concept of memory management, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

Virtual Memory Management: Demand Paging, Page Replacement: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU), Least Recently used (LRU), Allocation of Frames.

UNIT-2

8L+0T+8P=16 Hours

SECONDARY STORAGE STRUCTURE

Secondary Storage Structure: Over view of mass-storage structure, disk structure, disk scheduling;

File System Interface - File concept, Access Methods, Directory & Disk Structure, File-System Mounting, File Sharing, Protection; File-system structure.

File System Implementation- Directory implementation, Allocation Methods, Free Space Management.

PRACTICES:

- Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p, and n distinct page numbers occur in it.
 - a) What is a lower bound on the number of page faults?
 - b) What is an upper bound on the number of page faults?
- Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from "bad" to "perfect" according to their page-fault rate. Separate those algorithms that

SKILLS:

- ✓ Manage open-source operating systems like Ubuntu, Fedora etc.
- ✓ Know the concepts of Processes scheduling and File Systems.
- ✓ Identification of different disk scheduling methodologies.

suffer from Belady's anomaly from those that do not.

- a) LRU replacement. b) FIFO replacement.
- c) Optimal replacement. d) Second-chance replacement.
- Consider the page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.
- How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.
- LRU replacement.
- FIFO replacement.
- Optimal replacement.
- How many page faults occur for your algorithm for the following reference string with four page frames? 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.
- What is the minimum number of page faults for an optimal page replacement strategy for the reference string above with four page frames?
- Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. Three alternative results are shown below. For each case, what is happening?
 - a) Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?
 - b) CPU utilization 13 percent; disk utilization 97 percent.
 - c) CPU utilization 87 percent; disk utilization 3 percent.
 - d) CPU utilization 13 percent; disk utilization 3 percent.
- Implementation of Disk scheduling algorithm—FCFS.
- Implementation of Disk scheduling algorithm—SSTF and SCAN.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Classify the basic concepts of operating system and explore Linux ecosystem.	Analyze	1	1
2	Apply the concepts of process scheduling algorithms and process synchronization techniques to derive the efficiency of resource utilization.	Apply	1	1, 2, 3, 5, 12
3	Analyze the requirements for attempting Operating systems principles.	Analyze	1,2	1,2,12
4	Design the various memory management schemes For a given scenario.	Create	2	3,5
5	Apply the concepts of file system interface and implementation.	Apply	1,2	2,5

TEXT BOOKS:

- Sumitabha Das, Unix concepts and applications, TMH Publications, 4th Edition, July 2017.
- Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 9th Edition, John Wiley & Sons Inc, 2013.

REFERENCE BOOKS:

- Richard. Stevens and Stephen A Rago, "Advanced Programming in the Unix Environment", 3rd Edition, Addison-Wesley, 2013.
- William Stallings, "Operating Systems-Internals and Design principles" PHI, 7th Edition, 2012.
- Gary J. Nutt. Addison-Wesley, "Operating Systems: A Modern Perspective", 2nd Edition, 2001.
- B.A. Forouzan & R.F. Giberg, - Unix and shell Programming, Thomson, 1st Edition, New Delhi, 2003.