19IT219 OPERATING SYSTEMS

source:
https://www.proprofs.com/

Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

CS	WA/RA	SSH	SA	S	BS
5	5	30	20	5	5

PREREQUISITE COURSE: Data Structures

COURSE DESCRIPTION AND OBJECTIVES:

This course aims at imparting concepts and principles of Operating Systems. The course impart 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.

COURSE OUTCOMES:

Upon completion of the course, student will able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand, classify the basic concepts of operating system and Real time operating system.	-
2	Apply the concepts of process scheduling algorithms and process synchronization techniques to derive the efficiency of resource utilization .	1
3	Analyze the requirements for attempting operating systems principles.	2
4	Design the various memory management schemes for a given scenario.	3
5	Simulate the operating systems principles using simulation tools and programming in Unix.	5

SKILLS:

- ✓ *Install/ remove an operating system in a computer.*
- ✓ *Manage open source operating systems like Ubuntu, Fedora etc.*
- ✓ *Understand the concepts of Processes scheduling and File Systems.*
- ✓ *Analyze the various algorithms used for Memory management.*
- ✓ Identification of different disk scheduling methodologies.

UNIT– I **L- 9**

INTRODUCTION: What Operating System do; Operating System structure; Process concept - overview, process scheduling, operations on process; Threads; Inter process communication; Process scheduling - scheduling criteria, scheduling algorithms; Multiple-Processor scheduling; Case study- process scheduling in Linux.

UNIT – II **L- 9**

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

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

UNIT – III **L- 9**

MEMORY MANAGEMENT: Continuous memory allocation; Paging; Structure of the page table; Segmentation; Demand paging; Page replacement algorithms.

UNIT – IV **L- 9**

SECONDARY STORAGE STRUCTURE: Overview of mass-storage structure, disk structure, disk scheduling; File Systems - file concept, access methods, directory structure, file system mounting, file sharing protection; File-system structure, file system implementation, directory implementation, allocation methods, free space management.

UNIT - V **L- 9**

FUNDAMENTALS OF REAL-TIME SYSTEMS: Concepts and misconceptions; Multidisciplinary design challenges; Birth and evolution of real-time systems.

REAL TIME OPERATING SYSTEMS: From pseudo kernels to operating systems; Theoretical foundations in scheduling; System services for application programs.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Execution of various file/directory handling commands.
2. Simple shell script for b asic arithmetic and logical calculations.
3. Shell scripts to check various attributes of files and directories.
4. Write a Shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
5. Write a Shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
6. Write a Shell script that displays list of all the files in the current directory to which the user has read, Write and execute permissions.
7. Write a Shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
8. Write a Shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
9. Write a Shell script to list all of the directory files in a directory.
10. Write a Shell script to find factorial of a given integer.
11. Write a Shell script to count the number of lines in a file that do not contain vowels.
12. Write an awk script to find the number of characters, words and lines in a file.
13. Write a C Program that makes a copy of a file using standard I/O and system calls.
14. Write in C the following Unix commands using system calls

    a.cat b.mv

12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls-l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C Program that demonstrates redirection of standard output to a file .EX: ls>f1.

TEXT BOOK:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9th edition, John Wiley & Sons Inc, 2013.

REFERENCE BOOKS:

1. Richard. Stevens and Stephen A Rago, “Advanced Programming in the Unix Environment”, 3rd edition, Addison-Wesley, 2013.
2. William Stallings, “Operating Systems: Internals and Design Principles”, 6th edition, Prentice Hall, 2005.
3. Andrew S Tanenbaum ,”Modern Operating Systems”, 3rd edition, Prentice India, 2007
4. N.Matthew and R.Stones,”Beginning Linux Programming”, 4th edition, (Wrox) Wiley Publishing Inc., 2008
5. N.B.Venkateswarlu, “Advanced Unix Programming”, 1st edition, BS Publications, 2008.
6 . M.G.Venkatesh Murthy,”Introduction to Unix & Shell Programming”, 1st edition, Pearson Education, 2005.