19CS211 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

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.

COURSE OUTCOMES:

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

COs	Course Outcomes	POs
1	Understand, classify the basic concepts of operating system and Real Time Operating System (RTOS).	1
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.	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. Use vi editor to create a file with some text and save the file.
2. Add and Delete content to the file created above.
3. Write programs that use the following processing utilities.

   a. wc, od, cmp, comm, diff, head, tail, cut, paste, sort, grep, uniq

   b. Disk backup utilities c. Du, df, tar, cpio, ps, who
4. Write a shell script to generate a multiplication table.
5. Write a shell script that copies multiple files to a directory.
6. Write a shell script which counts the number of lines and words present in a given file.
7. Write a shell script, which displays the list of all files in the given directory.
8. Write a shell script (of small calculator) that adds, subtracts, multiplies and divides the given two integers.
9. Write a shell script to reverse the rows and columns of a matrix.
10. Write a C program that counts the number of blanks in a text file.

    i. Using standard I/O ii. Using system calls.
11. Write a C program that illustrates how to execute two commands concurrently with a command pipe.
12. Write a C program that illustrates file locking using semaphores.
13. Write a C program that implements a producer-consumer system with two processes.(using semaphores)
14. Write a C program that illustrates inter process communication using shared memory system calls.
15. Write a C program that illustrates the following:

    i. Creating a message queue.

    ii. Writing to a message queue.

    iii. Reading from a message queue.

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9th edition, John Wiley & Sons Inc, 2013.
2. Phillip A Laplante, Seppo J Ovaska, “Real Time Systems Design and Analysis”, 4th Edition, Wiley and IEEE Press, 2012.

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.