SEMESTER-III

COURSE 6: OPERATING SYSTEMS

Theory Credits: 3 3 hrs/week

Course Objectives

1. To understand the services provided by and the design of an operating system.
2. To understand what a process is and how processes are synchronized and scheduled.
3. To understand different approaches to memory management.
4. To understand the structure and commands in unix
5. Students should be able to understand shell programming

Course Outcomes:

COURSE OUTCOME NO	UPON SUCCESSFUL COMPLETION OF THIS COU SHOULD HAVE THE KNOWLEDGE AND SKILLS	PROGRAM OUTCOME
CO1	Analyse the services and functions of operating systems	**PO5,PO7**
Co2	Analyse the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance.	**PO5,PO7**
Co3	Analyse memory management techniques, concepts of virtual memory	**PO5,PO7**
Co4	To understand Introduction to Unix:- Architecture of Unix, Features of Unix , Unix Commands	**PO5,PO7**
Co5	To understand **Shell programming and** Simple shell program examples	**PO5,PO7**

UNIT – I

Operating System:

Introduction, Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations. Evolution of Operating Systems ,types of operating system, Simple ,Batch, Multi programmed, time shared, Parallel, Distributed Systems, Real-Time Systems, Operating System services.

UNIT – II

Process and CPU Scheduling –

Process concepts The Process, Process State, Process Control Block, Process communication. Threads. Process Scheduling Scheduling Queues, Schedulers, Context Switch, Pre-emptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms,Process Synchronization, The Critical section Problem, Semaphores, Classic Problems of Synchronization,

UNIT – III

Memory Management and Virtual Memory –

Logical & physical Address Space, Swapping, Contiguous Allocation, Paging-Structure of Page Table Segmentation, Segmentation with Paging,Virtual Memory, Demand Paging, Performance of Demanding Paging Page Replacement Page Replacement Algorithms, Allocation of Frames.

UNIT – IV

Introduction to Unix:- Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip.

UNIT – V Shell programming:

Ordinary and environment variables. The profile. Read and read only commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples.

TEXT BOOK:

"Operating System Concepts"-Silberschatz, Galvin, Gagne—eight Edition-John Willey & Sons INC 1,2,3 units

Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill(4,5) units

REFERENCES BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.
2. Principles of Operating Systems by Naresh Chauhan, OXFORD University Press

Student Activity:

1. Load any new operating system into your computer.
2. Partition the memory in your system
3. Create a semaphore for process synchronization.

Recommended Co – Curricular Activities:

Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

1. Programming exercises,
2. Practical assignments and laboratory reports,

3. Observation of practical skills,
4. Individual and group project reports.
5. Efficient delivery using seminar presentations,
6. Viva voce interviews.
7. Computerized adaptive testing, literature surveys and evaluations,
8. Peers and self-assessment, outputs form individual and collaborative work.

@@@@

SEMESTER-III

COURSE 6: OPERATING SYSTEMS

Practical Credits: 1 2 hrs/week

Course Objective:

This course enables students to develop OS scheduling logics and also to gain hands-on experience of UNIX OS..

COURSE OUTCOME NO	UPON SUCCESSFUL COMPLETION OF THIS COURSE, STUDENTS SHOULD HAVE THE KNOWLEDGE AND SKILLS TO:	PROGRAM OUTCOME NO
CO1	To implement CPU scheduling algorithms in c programming language	Po5,po7
CO2	To implement file/directory handling commands in Unix.	Po5,po7
CO3	To display list of currently logged users in Unix shell script	Po5,po7
CO4	To implement binary search using shell script	Po5,po7
CO5	To implement Fibonacci series using shell script	Po5,po7

LAB LIST

1. Write the program to implement CPU scheduling algorithm for first come first serve
2. Scheduling
3. Write the program to implement CPU scheduling algorithm for first come first serve
4. Scheduling
5. Write a program to implement CPU scheduling algorithm for shortest job first scheduling.
6. write a program to implement CPU scheduling algorithm for shortest job first scheduling.
7. Write a ‘C’ program to perform priority scheduling.
8. Write a ‘C’ program to perform priority scheduling.
9. Write a program to implement CPU scheduling for Round Robin Scheduling.
10. Execute various file/directory handling commands in UNIX.
11. Execute various file/directory handling commands in UNIX.
12. Write a Simple shell script for basic arithmetic and logical calculations.
13. Write a shell script to display list of users currently logged in.
14. Write a shell script to delete all the temporary files.
15. Write a shell script to search an element from an array using binary searching.
16. Write a shell script to determine whether a given number is a prime number or not
17. Write a shell script to print the first *n* Fibonacci numbers.
18. Execute various system administrative commands

@@@@