OPERATING SYSTEMS

COURSEOUTCOMES

At the end of the course, the student will develop ability to

1. Exemplify the basic principles used in the design of operating systems, and objective functions of operating systems.
2. Analyze Critical-section problem, assess how computing resources (such as CPU and memory) are managed by the operating system.
3. Compare and contrast the common algorithms used for both pre-emptive and non-pre- emptive scheduling of tasks in operating systems, such a priority, performance comparison, and fair-share schemes.
4. Explain memory hierarchy and cost-performance trade-offs
5. Summarize the full range of considerations in the design of file systems
6. Compare, contrast, and evaluate the key trade-offs between multiple approaches to operating system design, and identify and report appropriate design choices when solving real-world problems.

UNIT I

Computer system overview-basic elements, Instruction execution, operating system overview-objectives and functions, Evolution of OS. Process description and control - process states, process description, process control; Process and threads, Types of Threads.

UNIT II

Principles of concurrency - critical section, mutual exclusion, semaphores, monitors message passing, Readers/Writers problem. Scheduling: Types of schedulers–CPU scheduling algorithms.

UNIT III

Memory management requirements, partitioning, paging, and segmentation, Address translation, paging levels, Virtual memory, Page replacement algorithms.

UNIT IV

Deadlocks – prevention- avoidance – detection

I/O management and disk scheduling–I/O devices, organization of I/O functions, I/O buffering, disk scheduling, RAID.Memory mapped I/O,DMA, Interrupt handlers, device drivers, power management

UNIT V

File management–file and file system, file architecture, file organization and access, directories, file sharing, record blocking, Secondary storage management.

UNIT VI

Computer Security Concepts - Threats, Attacks, and Assets, Intruders, Malicious Software Overview–Trojan Horses, Viruses, Worms, Spyware, Rootkits and Bots, Access control, security maintenance-Firewalls, Antivirus, Code Signing, Intrusion Detection, Explore Java Security

TEXT BOOKS

1. William Stallings, “Operating Systems – internals and design principle”, Prentice Hall India, 8th Edition, 2014.
2. Silberschatz, Peter Galvin, “Operating System Concepts”, 9th Edition, 2013,

REFERENCE BOOKS

1. Andrew S. Tannenbaum& Albert S. Woodhull, “Operating System Design and Implementation”, Prentice Hall India, 3rd Edition, 2009.
2. Gary Nutt, “Operating System - A Modern Perspective”, Pearson Education Asia, 3rd Edition 2003.
3. Harvey .M. Deitel, “Operating Systems”, 3rd Edition, 2003.
4. Ida M.Flynn, Ann MclverMcHoes, “Understanding Operating Systems”, 7th Edition , Thomson Learning ,2014

WEB LINKS

1. http://nptel.ac.in/courses/106108101/
2. http://williamstallings.com/OperatingSystems/OS7e-Student/
3. http://williamstallings.com/OS/OS6e.html

https://www.sanfoundry.com/operating-system-questions-answers-basics/

OPERATING SYSTEMS LAB

COURSEOUTCOMES

At the end of the course, the student will develop ability to

1. Develop a solution to Producer-Consumer problem.
2. Execute various system calls in operating system.
3. Apply and implement the design techniques for disk scheduling.
4. Apply, analyze, design and implement the file contiguous technique.
5. Design the memory management scheme.
6. Develop programs for page replacement and dead locks conditions.

Week 1

Implementation of

a) Basic Unix Commands: man, list, date, calendar, echo, banner, who, tty, binary calculator, clear, manipulation (tput) .

b) Directory Related Commands: pwd, mkdir, cd, rmdir.

Week 2

Implementation of

a) File Related Commands: cat ,sort ,cp ,my , rm ,wc ,lp ,pg ,df ,free, filters and pipe.

b) Communication Through UNIX Commands: mesg, write, wall, mail, reply.

Week 3

a) Write a program to implement System Calls--(Fork, Exec, Sleep….)

b) Create a process in UNIX environment.

c) Write a program to illustrate exec.

d) Create child with sleep.

e) Write a program to demonstrate signal handling in UNIX (Kill).

Week 4

Implement CPU Scheduling algorithms

a) First Come First Serve.

b) Shortest Job First.

Week 5

Implement CPU Scheduling algorithms

a) Round Robin.

b) Priority.

Week 6

Producer Consumer Problem Using Semaphore

Implement the solution for Bounded Buffer (Producer-Consumer) Problem Using Inter Process Communication Technique – Semaphores.

Week 7

Memory Management Scheme

a) Write a Program to implement Memory Management scheme like Paging.

b) Write a Program to implement Memory Management scheme like Segmentation.

Week 8

Implementation of Contiguous allocation techniques:

a) Worst-Fit

b) Best-Fit

c) First-Fit

Week 9

Simulate all Page Re-Placement Algorithms.

Week 10

Simulate Banker’s algorithm for Deadlock Avoidance.

Week 11

Simulate all Disk scheduling algorithms

Week 12

Simulate file storage allocation techniques:

a) Contiguous (Using Array)

b) Linked –List (Using Linked List)

c) Indirect Allocation (Indexing)