

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY FACULTY OF TECHNOLOGY & ENGINEERING
Devang Patel Institute of Advance Technology and Research
Department of Computer Science & Engineering

Subject Name: Operating System
Subject Code: CS350

Semester: - 5th
Academic year: 2022-23

Course Outcome (COs):

At the end of the course, the students will be able to

- Visualize and understand Operating system functionality and working of OS. Understanding of functionality, services of operating system and differentiate between different types of OS.
- Define thread and process. Visualize how processes and threads are managed by the operating systems. Simulate and analyze various process scheduling algorithms. Explain and analyze different Inter process communication techniques.
- Describe deadlock and classify detection, recovery, prevention and avoidance algorithms. Test scenarios to report deadlock.
- Compare and evaluate various memory management schemes. Simulate and analyze Memory management algorithms. Identify and describe the role of I/O devices.
- Understand the file systems. Understand the secondary storage and simulate disk scheduling algorithm.
- Understand the basic commands of Linux file systems.

	Aim of the Practical	Hours	CO’S																						
1.	Study Practical: A. LINUX Architecture B. Types of OS- Linux, Flavors of LINUX UNIX, MAC, Window etc. C. Difference Between Lollipop and Marshmallow Operating System Version	2	1																						
2.	<p>Study of Unix Architecture and the following Unix commands with option:</p> <table><tr><td>User Access:</td><td>login, logout, passwd, exit</td></tr><tr><td>Help:</td><td>man, help</td></tr><tr><td>Directory:</td><td>mkdir, rmdir, cd, pwd, ls, mv</td></tr><tr><td>Editor:</td><td>vi, gedit, ed, sed</td></tr><tr><td>File Handling / Text Processing:</td><td>cp, mv, rm, sort, cat, pg, lp, pr, file, find, more, cmp, diff, comm, head, tail, cut, grep, touch, tr, uniq</td></tr><tr><td>Security and Protection:</td><td>chmod, chown, chgrp, newgrp</td></tr><tr><td>Information:</td><td>learn, man, who, date, cal, tty, calendar, time, bc, whoami, which, hostname, history, wc</td></tr><tr><td>System Administrator:</td><td>su or root, date, fsck, init 2, wall, shut down, mkfs, mount, unmount, dump, restor, tar, adduser, rmuser</td></tr><tr><td>Terminal:</td><td>echo, printf, clear</td></tr><tr><td>Process:</td><td>ps, kill, exec</td></tr><tr><td colspan="2">I/O Redirection (<, >, >>), Pipe (), *, gcc</td></tr></table>	User Access:	login, logout, passwd, exit	Help:	man, help	Directory:	mkdir, rmdir, cd, pwd, ls, mv	Editor:	vi, gedit, ed, sed	File Handling / Text Processing:	cp, mv, rm, sort, cat, pg, lp, pr, file, find, more, cmp, diff, comm, head, tail, cut, grep, touch, tr, uniq	Security and Protection:	chmod, chown, chgrp, newgrp	Information:	learn, man, who, date, cal, tty, calendar, time, bc, whoami, which, hostname, history, wc	System Administrator:	su or root, date, fsck, init 2, wall, shut down, mkfs, mount, unmount, dump, restor, tar, adduser, rmuser	Terminal:	echo, printf, clear	Process:	ps, kill, exec	I/O Redirection (<, >, >>), Pipe (), *, gcc		2	1, 2
User Access:	login, logout, passwd, exit																								
Help:	man, help																								
Directory:	mkdir, rmdir, cd, pwd, ls, mv																								
Editor:	vi, gedit, ed, sed																								
File Handling / Text Processing:	cp, mv, rm, sort, cat, pg, lp, pr, file, find, more, cmp, diff, comm, head, tail, cut, grep, touch, tr, uniq																								
Security and Protection:	chmod, chown, chgrp, newgrp																								
Information:	learn, man, who, date, cal, tty, calendar, time, bc, whoami, which, hostname, history, wc																								
System Administrator:	su or root, date, fsck, init 2, wall, shut down, mkfs, mount, unmount, dump, restor, tar, adduser, rmuser																								
Terminal:	echo, printf, clear																								
Process:	ps, kill, exec																								
I/O Redirection (<, >, >>), Pipe (), *, gcc																									
3.	<p>1. Write a script called hello which outputs the following:</p> <ul style="list-style-type: none">• your username• the time and date• who is logged on• Also output a line of asterisks (*****) after each section. <p>2. Write a shell script which calculates nth Fibonacci number where n will be provided as input when prompted.</p> <p>3. Write a shell script which takes one number from user and finds factorial of a Given number.</p>	2	1,2,3																						
4.	<p>Program maintenance using make utility</p> <p>A. Write a program that is spread over two files.</p> <p>B. Use following Makefile for program maintenance. To use make utility, use make Command.</p>	2	1,2																						
5.	<p>Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, stat, readdir, opendir.</p> <p>A. Write a program to execute fork () and find out the process id by getpid() system call.</p> <p>B. Write a program to execute following system call fork (), execl(), getpid(), exit(), wait() for a process.</p> <p>C. Write a program to find out status of named file (program of working stat () system cal</p>	2	1,5,6																						

6.	Write a C program in LINUX to implement Process scheduling algorithms and compare. A. First Come First Serve (FCFS) Scheduling B. Shortest-Job-First (SJF) Scheduling C. Priority Scheduling (Non-preemption) after completion extend on Preemption. D. Round Robin (RR) Scheduling	2	2
7.	Process control system calls: A. The demonstration of fork () B. execve() and wait() system calls along with zombie and orphan states.	2	6
8.	Thread management using pthread library. Write a simple program to understand it.	2	2, 3
9.	Write a C program in LINUX to implement inter process communication (IPC) Using Semaphore.	2	6,3
10.	Simulate Following Page Replacement Algorithms. A. First In First Out Algorithm B. Least Recently Used Algorithm C. Optimal Algorithm	2	5,2
11.	Thread synchronization using counting semaphores and mutual exclusion using mutex.	2	5,3
12.	Write a C program in LINUX to implement Bankers algorithm for Deadlock Avoidance.	2	6
13.	Write a C program in LINUX to perform Memory allocation algorithms and Calculate Internal and External Fragmentation. (First Fit, Best Fit, Worst Fit).	2	6

- Additional Practical(s):
- To implement of Dinning Philoshopr problem
 - Dinning Phiolosphor
 - Reader-Writer
 - To implement Disk-Scheduling Algorithm(s).
 - H2O Building Problem
 - Dining Savages Problem
 - Sleeping Barber Proble

