

Course Code	Course Name	Course Type	Cd	L	T	P	Marks		
							Sessional	Final Exam	Total
COM-312	Operating Systems Lab	PCC	2	0	0	4	50	0	50

Course Outcomes

At the end of the course the student will be able to

CO1	Develop competence in shell scripting on Linux operating system.
CO2	Use system commands to manipulate files, determine and control resource usage and perform administrative tasks.
CO3	Understand the architecture of the Linux operating system and its different subsystems.
CO4	Develop programs making use of linux system calls to solve given problems.
CO5	Program complex inter-process interactions involving communication and synchronization.

List of Activities for Operating Systems Lab (C)

S. No.	Activity Title
1	Practice basic bash, IO redirection, pipe, file and directory operations, searching files, process management and vi/nano editor commands.
2	- Use the date and who commands, in one line, such that the output of date is displayed on the screen and the output of who is redirected to a file. - Write a sed command that swaps the first and second words in each line in a file.
3	- Write a shell script that takes a command line argument and reports on whether it is a directory or a file. - Write a shell script that takes file names as arguments and convert all of them to uppercase.
4	- Write a shell script that determines the period for which a specified user is working on the system. - Write a shell script that displays all the lines between start and end line numbers passed as argument. - Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
5	Write a shell script and C program to perform the following string operations: - To extract a substring from a given string. - To find the length of a given string. Compare the running time of above shell script and C program using the time command.
6	Write a C program that takes, as a command line argument, the number of megabytes of memory it will use and during execution it should consume that much memory. Observe memory usage during program execution using free command.
7	Write a CPU bound C program and a I/O bound C program and observe the effect of their CPU share using the top command and its variants.
8	Observe and understand the process map of the previous two lab programs using pmap command.
9	Write a program in C that creates a child process, waits for the termination of the child and lists its PID.
10	Write a C program to print the address of a variable and enter a long loop (say using while(1)). Start three to four processes of the same program and observe the printed address values.
11	- Write C Program to demonstrate wait and signal operations in semaphores. - Implement consumer-producer problem using semaphores.
12	Implement pipe processing using IPC.
13	Develop applications in IPC using shared memory concept and message queues.