



Dundigal, Hyderabad – 500043

List of Laboratory Experiments

LINUX PROGRAMMING LABORATORY								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSC16	Core	L	T	P	C	CIA	SEE	Total
		1	0	2	2	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 36				Total Classes:36		
Branch: CSE	Semester: III	Academic Year: 2021-22				Regulation: UG20		
Course overview: This course provides open-source operating system concepts in Linux environment. It focuses on practice on shell commands, shell programming and demonstration of process concepts such as creation and establishing communication using Linux system calls. The Linux environment and demonstration of operating systems concepts using Linux system calls in C programs.								
Course objectives: The students will try to learn: <ol style="list-style-type: none"> 1. The Linux shell commands in command-line environment. 2. The shell programming constructs. 3. The simulation of file operations and Process management with inter-process communications techniques 								
Course outcomes: After successful completion of the course, students will be able to: <ul style="list-style-type: none"> CO1 Demonstrate text processing utilities, file handling utilities, security by file permissions, process utilities, disk utilities and networking commands with different options available for solving problems. CO2 Make use of bourne shell constructs, decision structures and loops in designing programs for complex problems. CO3 Interpret to write, compile, debug and run C language program in linux shell environment for implementing kernel level concepts. CO4 Identify basic methods and techniques used in solving simple programming tasks in the area of execution environment, processes signal and threads. CO5 Experiment with IPC mechanisms such as pipes, named pipes, shared memory, message queues, semaphores and sockets for inter process communication. CO6 Choose the appropriate protocol such as TCP or UDP for effective communication in client-server applications. 								
WEEK NO	EXPERIMENT NAME						CO	
WEEK – I	BASIC COMMANDS I						CO1	
	Study and Practice on various commands like man, passwd, tty, script, clear, date, cal, cp, mv, ln, rm, unlink, mkdir, rmdir, du, df, mount, umount, find, unmask, ulimit, ps, who, w.							
WEEK – II	BASIC COMMANDS II						CO1	
	Study and Practice on various commands like cat, tail, head , sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, tar, cpio.							
WEEK – III	SHELL PROGRAMMING I						CO2	
	a. Write a Shell Program to print all .txt files and .c files. b. Write a Shell program to move a set of files to a specified directory. c. Write a Shell program to display all the users who are currently logged in after a specified time. d. Write a Shell Program to wish the user based on the login time.							

WEEK – IV	SHELL PROGRAMMING II	CO2
	a. Write a Shell program to pass a message to a group of members, individual member and all. b. Write a Shell program to count the number of words in a file. c. Write a Shell program to calculate the factorial of a given number. d. Write a Shell program to generate Fibonacci series.	
WEEK – V	SIMULATING COMMANDS I	CO3
	a. Simulate cat command b. Simulate cp command	
WEEK – VI	SIMULATING COMMANDS II	CO3
	a. Simulate tail command b. Simulate head command	
WEEK – VII	SIMULATING COMMANDS III	CO3
	a. Simulate mv command b. Simulate nl command	
WEEK –VIII	SIGNAL HANDLING	CO4
	Write a program to handle the signals like SIGINT, SIGDFL, SIGIGN	
WEEK - IX	INTERPROCESS COMMUNICATIONS	CO5
	Implement the following IPC forms a. FIFO b. PIPE	
WEEK - X	MESSAGE QUEUES	CO5
	a. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers. b. Write a C program (receiver.c) that receives the messages (from the above message queue as specified and displays them.	
WEEK – XI	SHARED MEMORY	CO5
	Implement shared memory form of IPC.	
WEEK – XII	SOCKET PROGRAMMING	CO6
	a. Write client and server programs (using c) for interaction between server and client processes using TCP Elementary functions. b. Write client and server programs (using c) for interaction between server and client processes using UDP Elementary functions.	