LINUX PROGRAMMING LABORATORY

VI Semester: CSE

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
ACS109	Core	-	-	3	2	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 39				Total Classes: 39		

OBJECTIVES:

The course should enable the students to:

I. Analyze the Linux utilities and Linux environment.
II. Learn the fundamentals of shellscripting/programming.
III. Understand the basic Linuxadministration.
IV. Implement inter process communication and managementconcepts.

COURSE OUTCOMES(COs):

CO1 Identify and use Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security.
CO2 Work confidently in Linux environment.
CO3 Work with shell script to automate different tasks as Linux.
CO4 Illustrate file processing operations such as standard I/O and formatted I/O.
CO5 Design various client server applications using TCP or UDP protocols.

COURSE LEARNING OUTCOMES (CLOs):

The students should enable to:

1. Learn the importance of Linux architecture along with features.
2. Identify and use linux utilities to create and manage simple file processing operations
3. Apply the security features on file access permissions by restricting the ownership using advance linux commands.
4. Implement the SED, GREP and AWK commands for pattern matching and mathematical functions.
5. Understand the shell responsibilities of different types of shells.
6. Develop shell scripts to perform more complex tasks in shell programming environment.
7. Illustrate file processing operations such as standard I/O and formatted I/O.
8. Understand process structure, scheduling and management through system calls.
9. Generalize signal functions to handle interrupts by using system calls.
10. Illustrate memory management of file handling through file/region lock
11. Design and implement inter process communication (IPC) in client server environment by using pipe, message queues, named Pipes,
12. Illustrate client server authenticated communication in IPC through semaphores and shared memory.
13. Demonstrate various client server applications on network using TCP or UDP protocols.
    Design custom-based network applications using the sockets interface in heterogeneous platforms.

LIST OF EXPERIMENTS

Week-1	GENERAL PURPOSE UTILITIES COMMANDS

Learning installation and upgradation of the Linux operating system.
Basic Linux commands: User and session management commands: useradd, groupadd, userdel, groupdel, passwd; General purpose utilities: echo, printf, bc,who, whoami, tty, uname, clear, ls.

Week-2	FILE SYSTEM, TEXT PROCESSING COMMANDS AND VI EDITOR

Linux commands: cat-create a file, append a file and open a file. file, wc, cp, rm, mv, more, head,tail, gzip, gunzip. vi editor- commands, navigation commands and creating a vi editor file.

Week-3	SED, GREP,EGREP,FGREP

1. Finding a file containing a particular textstring
2. Regular expressions in grepcommand.
3. Search multiple words / string pattern using grep command on bashshell
4. Illustratebywritingscriptthatwillprint,message–HelloWorld,inBoldandBlinkeffect,andin different colors like red, brown etcusing echocommands.
5. Write a program that will output the desiredpatterns
   1
   2 2
   3 3 3
   4 4 4 4
   5 5 5 5 5

Week-4	BASIC SHELL SCRIPTING

1. Write a shell script that accepts a file name, starting and ending line numbers as argumentsand displays all the lines between the given linenumbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments toit.
3. Writeashellscriptthatdisplays alistofallthefiles inthecurrentdirectorytowhichtheuserhas read, write and executepermissions.

Week-5	SHELL SCRIPTING

1. Write a program to generate Fibonacciseries
2. Write a program to check whether given string is palindrome ornot
3. Write a shell script to find factorial of a given integer.

Week-6	INPUT OUTPUT REDIRECTIONS AND COMMAND SUBSTITUTIONS

1. Writeashellscriptthatreceivesanynumberoffilenamesasargumentschecksifevery argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is alsoreported.
2. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
3. Write a shell script to list all of the directory files in adirectory.

Week-7	AWK SCRIPT

1. Write an awk script to count the number of lines in a file that do not containvowels.
2. Write an awk script to find the number of characters, words and lines in a file.
3. Write an awk script to calculate average marks of eachstudent.
4. Write an awk script to replace a string in afile.

Week-8	PATTERN SCANNING AND PROCESSING SCRIPTS

1. Write a C program that makes a copy of a file using standard I/O and systemcalls.
2. Illustrate to redirect the standard input (stdin) and the standard output (stdout) of a process, so that scanf().
3. Write an reads from the pipe and printf () writes into thepipe.

Week-9	PATTERN SCANNING AND PROCESSING SCRIPTS

1. Write a program that takes one or more file/directory names as command line input and reports the following information on the file. A. File type. B. Number of links. C. Time of last access. D. Read, write and executepermissions.
2. Write a C program to emulate the Unix ls –lcommand.
3. Write a C program to list for every file in a directory, its inode number and filename.

Week-10	PROCESS ATTRIBUTES AND USAGE OF FORK( )

1. WriteaCprogramtocreateachildprocessandallowtheparenttodisplay–parentlandthechild todisplay–childlonthescreen.
2. Write a C program to create a zombieprocess.
3. Write a C program that illustrates how an orphan iscreated.

Week-11	USAGE OF PIPES AND NAMED PIPES

1. Write a C program that illustrates how to execute two commands concurrently with acommand pipe. Ex:- ls –l |sort
2. Write C programs that illustrate communication between two unrelated processes using namedpipe.
3. Write a C program to create a message queue with read and write permissions to write3 messages toit

Week-12	SYNCHRONIZATION AND LOCKING TECHNIQUES

1. Write a C program to allow cooperating processes to lock a resource for exclusive use, usinga) Semaphores b) flock or lockf system calls.
2. Write a C program that illustrates suspending and resuming processes usingsignals.
3. Write a C program that implements a producer-consumer system with two processes.(using Semaphores).

Week-13	CLIENT SEVER IMPLEMENTATION USING SOCKETS AND SHARED MEMORY

1. Write client and server programs (using c) for interaction between server and client processesusing Unix domainsockets.
2. Write client and server programs (using c) for interaction between server and client processesusing Internet domainsockets.
3. Write a C program that illustrates two processes communicating using sharedmemory.
4.

Reference Books

1. W.Richard,Stevens,–AdvancedProgramminginthe UNIXEnvironment|, PearsonEducation,1st Edition,2005.
2. SumitabhaDas,–UnixConcepts andApplications|,Tata McGrawHill,4th Edition,2006.
3. NeilMathew,RichardStones,‖BeginningLinuxProgramming‖,Wrox,WileyIndia,4thEdition, 2011.