GMR Institute of Technology
An Autonomous Institute Affiliated to JNTUK, Kakinada

GMRIT
Training Tomorrow's
Engineers Today

COURSE HANDOUT

II B.Tech - 4th Semester :: AY 2017-18

Course Title   : LINUX PROGRAMMING LAB

Course Code   : 16CS408   Dated: 04-11-2017

Course Structure   : 0-0-4-2

Course coordinator   : Mr. K LAKSHMANA RAO

Instructor(s)   : Mrs. S.Vani Kumari, Mr. V. Prasad, Mr.M. Sateesh

Course Description : This course covers extensively the LINUX operating system commands, shell programming, system programming such as Inter process communication (IPC) tools,signals and socket programming and basics on threads. It introduces students to explore on LINUX operating system concepts like: files, it's related command line tools to perform different operations on files, Shell scripting fundamentals, handling different types of signals, fundamentals on IPC, it's implementation, socket programming and implementation of threads .

Scope and Objective :

The main scope and objective of this course is to:

- Exposure on LINUX environment and usage of commands.

- Learn shell scripting fundamentals: its syntax, patterns, and Styles

- Write shell program to solve specified problems.

- Learn inter process communication tools and its implementation

- Learn handling signals and implementation of threads

- Learn Socket programming.

Course Outcome:

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

- use different LINUX commands.

- Compile and run a shell scripts.

- Implement inter process communication programs.

- Understand LINUX signals.

- Implementation of threads.

- Socket programming to implement client server communication.

Lab Manuals:

1. LINUX programming Lab Manual – Department of CSE GMRIT Rajam

Reference Books:

1. UNIX and Shell Programming by Behrouz A. Forouzan and Richard F. Gilberg
2. Advanced UNIX Programming by N.B.Venkateswarulu BS Publications
3. UNIX® Network Programming Volume 1, Third Edition: The Sockets Networking API By W. Richard Stevens , Bill Fenner , Andrew M. Rudoff.
4. UNIX Network programming Interprocess communications second edition by W.RECHARD STEVENS.

List of Experiments:

Experiment 1

Execute the following UNIX commands: cal -date -echo -bc -passwd -PATH-who -uname -pwd -cd -mkdir -rmdir -cat -cp -rm -mv -more -ls -wc.

Experiment 2

Execute the following UNIX commands: cmp -comm -diff -tar -df -du-mount-umount-chmod and vi editor commands.

Experiment 3

a) Write a shell script to accept three numbers and display the largest.

b) Write a shell script which will accept different numbers and find their sum.

c) Write a shell script to find the number of files in a directory

Experiment 4

a) Write a shell script to display first ten positive numbers using until loop.

b) Write a shell script to print the first 10 odd numbers using the while loop.

Experiment 5

a) Write a sed command that deletes the first character in each line in a file.

b) Write a sed comm and that deletes the character before the last character in each line in a file.

c) Write a sed command that swaps the first and second words in each line in a file.

Experiment 6

Implement the following forms of IPC. a) Pipes b) FIFO

Experiment 7

Implement file transfer using Message Queue form of IPC.

Experiment 8

Implementation of semaphore: Write a program that demonstrates how two processes can share a variable using semaphore

Experiment 9

To implement C Program to Catch Signals SIGINT -SIGKILL -SIGSTOP.

Experiment 10

Write a c program to implement Two threads displaying two strings "Hello" and "How are you?" independent of each other.

Experiment 11

Design TCP iterative Client and Server application to reverse the given input sentence.

Experiment 12

Design UDP Client Server to transfer a file.

Course Plan:

Lab Session No.	Learning Objectives:	Topics to be covered
1	Introduction about LINUX environment	**LINUX GUI,Terminal, file system and some simple commands.**
2	Introduces basic LINUX commands	**Experiment 1** Execute the following UNIX commands: cal -date -echo -bc -passwd -PATH-who -uname -pwd -cd -mkdir -rmdir -cat -cp -rm -mv - more -ls -wc.

3	File comparison related commands,storage media related commands, command to assign file security and vi editor commands.	**Experiment 2** Execute the following UNIX commands: cmp -comm -diff -tar -df -du-mount-umount-chmod and vi editor commands.
4	Basics for writing shell scripting	Fundamentals on shell scripting: Different types of shells, creation of shell script, variables in shell, statements in shell and command line arguments.
5	Writing basic shell scripts, scripting using conditional statement and iterative statements	**Experiment 3** a) Write a shell script to accept three numbers and display the largest. b) Write a shell script which will accept different numbers and find their sum. c) Write a shell script to find the number of files in a directory
6	Shell scripting using iterative statements.	**Experiment 4** a) Write a shell script to display first ten positive numbers using until loop. b) Write a shell script to print the first 10 odd numbers using the while loop.
7	Practice on sed command for manipulating files	**Experiment 5** a) Write a sed command that deletes the first character in each line in a file. b) Write a sed comm and that deletes the character before the last character in each line in a file. c) Write a sed command that swaps the first and second words in each line in a file.
8	Introduction on Inter process communication tools, fork system call, implementation of pipes	**Experiment 6**

	and FIFOs	Implement the following forms of IPC. a) Pipes b) FIFO
9	System calls on message queues.	**Experiment 7** Implement file transfer using Message Queue form of IPC.
10	System calls on semaphores	**Experiment 8** Implementation of semaphore: Write a program that demonstrates how two processes can share a variable using semaphore
11	Different types of signals and Kill command. System calls to catch signals.	**Experiment 9** To implement C Program to Catch Signals SIGINT -SIGKILL -SIGSTOP.
12	POSIX standard system calls to implement threads in c-language.	**Experiment 10** Write a c program to implement Two threads displaying two strings "Hello" and "How are you?" independent of each other.
13	System calls to implement connection oriented communication.	**Experiment 11** Design TCP iterative Client and Server application to reverse the given input sentence.
14	System calls to implement connection less communication.	**Experiment 12** Design UDP Client Server to transfer a file.

Evaluation scheme:

Component	Particular	Marks	Date & Time
Lab regularity	No of Experiments completed and recorded	10+5	04.12.17 to 07.04.18
Internal Examination	150 minutes	10	16.04.18 to 21.04.18
External Examination	180 minutes	50	16.04.18 to 21.04.18
	Total	75	

Signature of the Instructor   Signature of the course-coordinator