Course code	Course Name	L-T-P-Credits	Year of Introduction
CS232	Free and Open Source Software Lab	0-0-3-1	2016

Pre-requisite: CS204 Operating systems

Course Objectives: To expose students to FOSS environment and introduce them to use open source packages in open source platform.

List of Exercises/Experiments: (Minimum 12 exercises/experiments are mandatory)

1. Getting started with Linux basic commands and directory structure, execute file, directory operations.
2. Linux commands for redirection, pipes, filters, job control, file ownership, file permissions, links and file system hierarchy.
3. Shell Programming : Write shell script to show various system configuration like
   - Currently logged user and his logname
   - Your current shell
   - Your home directory
   - Your operating system type
   - Your current path setting
   - Your current working directory
   - Show Currently logged number of users
4. Write shell script to show various system configuration like
   - About your OS and version, release number, kernel version
   - Show all available shells
   - Show mouse settings
   - Show computer CPU information like processor type, speed etc
   - Show memory information
   - Show hard disk information like size of hard-disk, cache memory, model etc
   - File system (Mounted)
5. Shell script program for scientific calculator.
6. Write a script called addnames that is to be called as follows, where *classlist* is the name of the classlist file, and *username* is a particular student's username.
   ./addnames*classlistusername*
   The script should
   - check that the correct number of arguments was received and print an usage message if not,
   - check whether the classlist file exists and print an error message if not,
   - check whether the username is already in the file, and then either
   - print a message stating that the name already existed, or
   - add the name to the end of the list.
7. Version Control System setup and usage using GIT.
   - Creating a repository
   - Checking out a repository
   - Adding content to the repository
   - Committing the data to a repository

- Updating the local copy
- Comparing different revisions
- Revert
- Conflicts and Solving a conflict

8. Text processing and regular expression with Perl, Awk: simple programs, connecting with database e.g., MariaDB
9. Shell script to implement a script which kills every process which uses more than a specified value of memory or CPU and is run upon system start.
10. GUI programming : Create scientific calculator – using Gambas or try using GTK or QT
11. Running PHP : simple applications like login forms after setting up a LAMP stack
12. Advanced linux commands curl, wget, ftp, ssh and grep
13. Application deployment on a cloud-based LAMP stack/server with PHP eg: Openshift, Linode etc.
14. Kernel configuration, compilation and installation : Download / access the latest kernel source code from *kernel.org*, compile the kernel and install it in the local system. Try to view the source code of the kernel
15. Virtualisation environment (e.g., xen, kqemu, virtualbox or lguest) to test an applications, new kernels and isolate applications. It could also be used to expose students to other alternate OSs like *BSD
16. Compiling from source : learn about the various build systems used like the auto* family, cmake, ant etc. instead of just running the commands. This could involve the full process like fetching from a cvs and also include autoconf, automake etc.,
17. Introduction to packet management system : Given a set of RPM or DEB, how to build and maintain, serve packages over http or ftp. and also how do you configure client systems to access the package repository.
18. Installing various software packages. Either the package is yet to be installed or an older version is existing. The student can practice installing the latest version. Of course, this might need Internet access.
    - Install samba and share files to windows
    - Install Common Unix Printing System(CUPS)

Expected outcome:

Students will be able to:

1. Identify and apply various Linux commands
2. Develop shell scripts and GUI for specific needs
3. Use tools like GIT, .
4. Perform basic level application deployment, kernel configuration and installation, packet management and installation etc.