

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:**

- Getting started with Linux basic commands for directory operations, displaying directory structure in tree format etc.
- Linux commands for operations such as redirection, pipes, filters, job control, changing ownership/permissions of files/links/directory.
- Advanced linux commands curl, wget, ftp, ssh and grep
- Shell Programming : Write shell script to show various system configuration like
  - Currently logged user and his login name
  - Your current shell
  - Your home directory
  - Your operating system type
  - Your current path setting
  - Your current working directory
  - Number of users currently logged in
- Write shell script to show various system configurations like
  - your OS and version, release number, kernel version
  - all available shells
  - computer CPU information like processor type, speed etc
  - memory information
  - hard disk information like size of hard-disk, cache memory, model etc
  - File system (Mounted)
- Write a shell script to implement a menu driven calculator with following functions
  - Addition
  - Subtraction
  - Multiplication
  - Division
  - Modulus
- Write a script called addnames that is to be called as follows  
`./addnames ulist username`  
 Here *ulist* is the name of the file that contains list of user names and *username* is a particular student's username. The script should
  - check that the correct number of arguments was received and print a message, in case the number of arguments is incorrect
  - check whether the ulist file exists and print an error message if it does not
  - check whether the username already exists in the file. If the username exists, print a message stating that the name already exists. Otherwise, add the username to the end of the list.

8. Version Control System setup and usage using GIT. Try the following features.
- 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 a conflict Resolution
9. Shell script which starts on system boot up and kills every process which uses more than a specified amount of memory or CPU.
10. Introduction to packet management system : Given a set of RPM or DEB, build and maintain, and serve packages over http or ftp. Configure client systems to access the package repository.
11. Perform simple text processing using Perl, Awk.
12. Running PHP : simple applications like login forms after setting up a LAMP stack
13. Virtualisation environment (e.g., xen, qemu, virtualbox or lguest) to test applications, new kernels and isolate applications. It could also be used to expose students to other alternate OS such as freeBSD
14. 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.,
15. 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
16. GUI Programming: Create scientific calculator – using any one of Gambas, GTK, QT
17. Installing various software packages. Either the package is yet to be installed or an older version is present. The student can practice installing the latest version. ( Internet access is needed).
- Install samba and share files to windows
  - Install Common Unix Printing System(CUPS)
18. Set up the complete network interface by configuring services such as gateway, DNS, IP tables etc. using *ifconfig*

**Expected outcome:**

The 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.