



**NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE**  
(NAAC Accredited)



( Approved by AICTE , Affiliated to APJ Abdul Kalam Technological University, Kerala )

**Pampady, Thiruvilwamala(PO), Thrissur(DT), Kerala 680 588**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**LAB MANUAL**



***CS232 FREE AND OPEN SOURCE SOFTWARE LAB***

**VISION OF THE INSTITUTION**

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

**MISSION OF THE INSTITUTION**

**NCERC** is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

## **ABOUT THE DEPARTMENT**

- ◆ Established in: 2002
- ◆ Course offered : B.Tech in Computer Science and Engineering  
M.Tech in Computer Science and Engineering  
M.Tech in Cyber Security
- ◆ Approved by AICTE New Delhi and Accredited by NAAC
- ◆ Certified by ISO 9001-2015
- ◆ Affiliated to A P J Abdul Kalam Technological University, Kerala.

## **DEPARTMENT VISION**

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering professionals to facilitate continuous technological advancement.

## **DEPARTMENT MISSION**

- To Impart Quality Education by creative Teaching Learning Process.
- To promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
- To Inculcate Entrepreneurship Skills among Students.
- To cultivate Moral and Ethical Values in their Profession.

## **PROGRAMME EDUCATIONAL OBJECTIVES**

**PEO1:** Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

**PEO2:** Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

**PEO3:** Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

**PEO4:** Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Team work and leadership qualities.

## **PROGRAM OUTCOMES (POS)**

### **Engineering Graduates will be able to:**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSO)**

**PSO1:** Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

**PSO2:** Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

**PSO3:** Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

**COURSE OUTCOME**

SUBJECT CODE: CS232	
COURSE OUTCOMES	
C232.1	To identify and apply various Linux Commands
C232.2	To develop shell scripts and GUI for specific needs
C232.3	To apply tools like GIT version control
C232.4	To implement basic level application deployment, kernel configuration and installation, packet management and installation etc.
C232.5	To understand formulating scripts for text processing and regular expression using Perl, Awk commands
C232.6	To apply the scripting language to solve real world problems

**MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES**

CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C232.1	3	3	3	2				3	3	3		3
C232.2	2		3		3			3	3	3		3
C232.3	3		3		3			3	3	3		3
C232.4	3		2	2	3			3	3	3		3
C232.5	2		3		3			3	3	3		3
C232.6	3	3	3	2	3			3	3	3		3

**MAPPING OF COURSE OUTCOMES WITH PROGRAM SPECIFIC OUTCOMES**

CO'S	PSO1	PSO2	PSO3
C232.1	3	3	3
C232.2	2	3	3
C232.3	3	2	3
C232.4	3		3
C232.5	2	3	3
C232.6	3	3	3

**PREPARATION FOR THE LABORATORY SESSION**  
**GENERAL INSTRUCTIONS TO STUDENTS**

1. Read carefully and understand the description of the experiment in the lab manual. You may go to the lab at an earlier date to look at the experimental facility and understand it better. Consult the appropriate references to be completely familiar with the concepts and hardware.
2. Make sure that your observation for previous week experiment is evaluated by the faculty member and you have transferred all the contents to your record before entering to the lab/workshop.
3. At the beginning of the class, if the faculty or the instructor finds that a student is not adequately prepared, they will be marked as absent and not be allowed to perform the experiment.
4. Bring necessary material needed (writing materials, graphs, calculators, etc.) to perform the required preliminary analysis. It is a good idea to do sample calculations and as much of the analysis as possible during the session. Faculty help will be available. Errors in the procedure may thus be easily detected and rectified.
5. Please actively participate in class and don't hesitate to ask questions. Please utilize the teaching assistants fully. To encourage you to be prepared and to read the lab manual before coming to the laboratory, unannounced questions may be asked at any time during the lab.
6. Carelessness in personal conduct or in handling equipment may result in serious injury to the individual or the equipment. Do not run near moving machinery/equipment. Always be on the alert for strange sounds.
7. Students must follow the proper dress code inside the laboratory.
8. Maintain silence, order and discipline inside the lab. Don't use cell phones inside the laboratory.
9. Any injury no matter how small must be reported to the instructor immediately.
10. Check with faculty members one week before the experiment to make sure that you have the handout for that experiment and all the apparatus.

## AFTER THE LABORATORY SESSION

1. Clean up your work area.
2. Check with the technician before you leave.
3. Make sure you understand what kind of report is to be prepared and due submission of record is next lab class.
4. Do sample calculations and some preliminary work to verify that the experiment was successful

**MAKE-UPS AND LATE WORK** :Students must participate in all laboratory exercises as scheduled. They must obtain permission from the faculty member for absence, which would be granted only under justifiable circumstances. In such an event, a student must make arrangements for a make-up laboratory, which will be scheduled when the time is available after completing one cycle. Late submission will be awarded less mark for record and internals and zero in worst cases.

## LABORATORY POLICIES

1. Food, beverages & mobile phones are not allowed in the laboratory at any time.
2. Do not sit or place anything on instrument benches.
3. Organizing laboratory experiments requires the help of laboratory technicians and staff. Be punctual.

**SYLLABUS**

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:**

1. Getting started with Linux basic commands for directory operations, displaying directory structure in tree format etc.
2. Linux commands for operations such as redirection, pipes, filters, job control, changing ownership/permissions of files/links/directory.
3. Advanced linux commands curl, wget, ftp, ssh and grep
4. 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
5. 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)
6. Write a shell script to implement a menu driven calculator with following functions
  1. Addition
  2. Subtraction
  3. Multiplication
  4. Division
  5. Modulus
7. 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, kqemu, 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.



**INDEX**

<b>EXP NO.</b>	<b>EXPERIMENT NAME</b>	<b>PAGE NO</b>	<b>MARKS</b>	<b>SIGN</b>
1	GETTING USED TO LINUX BASIC COMMANDS, DIRECTORY STRUCTURE, FILE AND DIRECTORY OPERATIONS	11		
2	LINUX COMMANDS FOR REDIRECTION, PIPES, FILTERS, JOB CONTROL, FILE OWNERSHIP, FILE PERMISSIONS, LINKS AND FILE SYSTEM HIERARCHY	16		
3	ADVANCE LINUX COMMANDS: curl, wget, ftp, ssh and grep.	21		
4	SHELL PROGRAMMING FOR VARIOUS SYSTEM CONFIGURATIONS	25		
5	SHELL PROGRAMMING FOR VARIOUS SYSTEM CONFIGURATIONS	27		
6	SHELL PROGRAMMING TO IMPLEMENT SIMPLE CALCULATOR	29		
7	VIRTUALISATION ENVIRONMENT TO TEST NEW APPLICATIONS AND KERNELS AND ISOLATE APPLICATIONS	32		
8	PERFORM SIMPLE TEXT PROCESSING USING PERL	35		
9	KERNEL CONFIGURATION, COMPILATION AND INSTALLATION	38		

10	SHELL PROGRAMMING TO START ON BOOT-UP AND KILL EVERY PROCESS WHICH USES MORE THAN A SPECIFIED AMOUNT OF MEMORY IN CPU.	40		

**FINAL VERIFICATION BY THE FACULTY**

**TOTAL MARKS:**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**AIM:**

To familiarize with Linux basic commands, directory structure, file and directory operations.

**DESCRIPTION**

**Linux basic commands:**

(a) cd: - It is used to change directory.

- The format of this command is:

`cd <space> <directory_name>`

Eg: `cd bin`

- Here “/” indicates root directory and “~” indicates home directory.

-To move step by step backwards we use:

`cd<space>..`

(b) ls: - It is used to list all the files in the current directory.

-The hidden files in the directory can be listed using the command:

`ls<space>-a`

(c) clear: - It is used to clear the terminal screen.

(d) pwd: - It is used to get the full path to current directory.

(e) cal: - It is used to get the calendar of the current month.

-The whole calendar can be retrieved using: `cal<space>-y`

(f) date: - It is used to get the current time of the machine.

(g) mkdir: - It is used to make directory in the current location.

- The format is given by:

mkdir<space><directory\_name>

Eg: mkdir newfile

(h) rmdir: - It is used to remove directory from the current location.

- The format is given by:

rmdir<space><directory name>

Eg: rmdir newfile

- If the directory is not present then it displays error as:

rmdir: cannot remove <directoryname>:No such file or  
directory

- If the directory is not empty, it displays the error as:

rmdir: failed to remove<directoryname>: Directory not  
empty

(i) touch: -It is used to create a file.

-The format is given by:

touch<space><filename>

Eg: touch hi

(j) rm: -It is used to remove a file.

-The format is given by:

rm<space><filename>

Eg: rm hi

(k) mv: -It is used to rename a file.

-The format is given by:

mv<space><oldname><newname>

Eg: mv hi hello

(l) cp: -It is used to copy files between directories.

-The format is given by:

`cp<space><source><destination>`  
Eg: `cp hello /usr`

(m) man: -It is used to get the manual of each of the commands.

-The format is given by:

`man<space><command>`

(n) whoami: -It is used to get the current user

(o) history: -It is used to display the history details of the terminal.

### **Directory structure:**

The Unix operating system consists of a single file system. All the controlling directories is present in the main file system termed as the root directory of the operating system. The different directories in the root are as follows:

(1) bin: -This is used to store binary files.

-All the executable files will be present here.

(2) boot: -This is used to store all booting related files.

-The initial program to be loaded on to the ram and the kernel is situated in this directory.

(3) dev: -This is used to store all device descriptions.

-All the known devices will have a description and the newly installed devices will have an entry in this directory.

(4) etc: -This is used to store the configuration files of the system.

-It includes details like user login, IP address, hostname, etc.

(5) home: -Each user will have a directory here.

-User login is directed to this and all home directory files are here.

(6) lib/lib64/lib32: -This consists of all the system libraries.

(7) media: - Contents of external drives will be stored here.

(8) mnt: -All the mounted drives details will be stored here.

(9) opt: -This is an optional folder for third part programs.

(10) proc: -This is a special folder that resides in RAM.

-All the processes are here and process ids are displayed as sub-directories which contains its resources.

(11) lost+found: -This is an administrative level directory that contains recovery bits for the deleted files or applications.

(12) root: -This is another administrative level directory to store system administrative files.

(13)/sbin: -This also contains system administrative files but can be accessed by all users.

(14) srv: -This contains server related files.

(15) sys: -This contains system related files like power related files.

(16) tmp: -This is used to store temporary files.

(17) usr: -This is used for extra programs like compilers.

(18) var: -This is used to store variable files like system logs.

### **RESULT AND DISCUSSION:**

Familiarized with the basic Linux commands and also the root directory of the operating system.

VIVA QUESTIONS:

1. List and give the purpose of Linux Basic commands.
2. List and give the purpose of Linux directory structure commands.
3. Which command changes a file's group owner
4. Which option of rm command is used to remove a directory with all its sub directories.
5. Which command creates an empty file if file does not exist?

EXP NO.  
2**LINUX COMMANDS FOR REDIRECTION, PIPES, FILTERS, JOB CONTROL, FILE OWNERSHIP, FILE PERMISSIONS, LINKS AND FILE SYSTEM HIERARCHY.****AIM:**

To familiarize with Linux commands for redirection, pipes, filters, job control, file ownership, file permissions, links and file system hierarchy.

**DESCRIPTION****Redirection:**

It is one of the most powerful commands in Linux which is used to redirect the data and is categorized into three:

**(a) Standard Output:**

Most command line programs that display their results do so by sending their results to a facility called standard output. By default, standard output directs its contents to the display. To redirect standard output to a file, the ">" character is used like this:

Eg: `ls > hi` where *hi* is the file to which the output is provided.

**(b) Standard Input:**

Many commands can accept input from a facility called standard input. By default, standard input gets its contents from the keyboard, but like standard output, it can be redirected. To redirect standard input from a file instead of the keyboard, the "<" character is used like this:

Eg: `sort < hi` so here the input from the file "hi" will be sorted

**(c) Standard error**

The errors that comes in terminal can be redirected to a different file using the standard error redirection method. To perform error redirection, the "2>" is used like this:



Eg: `rm 0 2> hi` *where we are trying to remove a file named "0" and the error will be redirected to the file "hi"*

### **Pipe:**

It is used for page by page display of contents in terminal. Pipe command is executed using "|" character. Two parameters are used with pipe namely more and less. If more is used, scrolling backward is not possible. But if less is used, there is provision for scrolling upwards and downwards through the pages.

Eg: `ls | more`

`ls | less`

### **Filters:**

(a) `wc` : It is termed as word count and is used to filter the number of lines, number of words and number of character bytes in a file. The format used is:

`wc <space> <filename>`

Eg: `wc hi`

- More filters can be used in `wc` to filter out individual components:

`-l` : for number of lines

`-w` : for number of words

`-c` or `-m` : for number of character bytes

(b) `head`: It is used to filter out initial n number of lines from a file. The format used is:

`head <space> -n <space> <filename>`

Eg: `head -2 hi` *which will take first two lines of file named "hi"*

(c) tail: It is used to filter out last n number of lines from a file. The format used is:

```
tail<space>-n<space><filename>
```

Eg: tail -3 hi *which will take final 3 lines of file named "hi"*

### **File ownership and file permission:**

- Once a file is created it will have three types of ownerships:
  - (a) Owner permission
  - (b) Group permission
  - (c) Other permission
  
- Owner permission is for the one who has created the file. Group permission is for a set of people who can access it. Other permission is for all others.
  
- Permissions of a file can be seen by the command:

```
ls<space>-l
```

Eg: drwxr-xr--x 2 root root 4096 Sep 22 0:15 bin

- In the above example, the first part is the permission bytes, next is the number of links, third part is the owner name, fourth part is group name, fifth part is memory bytes, sixth part is date and time, and the last part is the filename.
  
- The permission byte consists of 1byte of file type (like d) which can be directory, file, or link. The next 3 bytes of owner permission (rwx), the next 3 for group (-xr) and the last 3 for others (--r).

- Three permissions are mainly used: r for read, w for write and x for execute.
- The permissions can be manipulated by “chmod” command.

`chmod<space>u/g/o+/-r/w/x`

- Chagemode is followed by u,g,o for user or owner, group or others respectively. +/- indicates whether to add permission or to remove it. And the last session shows which kind of permission r/w/x.

Eg: If we want to add w permission for group in above example

`chmod g + w`

- If we remove all permissions, then we can deny access to that file.

### **Link:**

- There are two types of links: (a) soft-link and (b) hard-link.
- Hard-link is used to access same data in multiple ways. We can create different links for same data.
- The format used is :

`ln<space><oldname><space><newname>`

Eg: *ln hi hello, so a new link for hi will be created with name hello*

- Soft-link is used to create shortcut to a file. The format used is:

`ln<space>-s<space><oldname><softname>`

Eg: In –s hi hai, *so a shortcut for hi will be created with name hai*

### **Job Control:**

- It can be used to control jobs. “ps” is used to get all the process and “ps –x” is used to get all the background running processes.

### **RESULT AND DISCUSSION:**

Familiarized with advanced and more powerful commands in Linux like redirection, pipe, file ownership and permission, filters, links and job control.

### **VIVA QUESTIONS:**

1. Explain about pipes and give the commands used for input and output.
2. Explain about filter and grep command.
3. What are the file ownerships and permissions available in Linux .
4. Which command changes a file’s group owner
5. True statement about chown command in Linux is
  - A. Creates a thread
  - B. Creates a child process
  - C. Changes the mode of operation to kernel mode
  - D. Changes the users and/or group ownership of each given file

EXP NO.  
3

**ADVANCE LINUX COMMANDS: curl, wget, ftp, ssh and grep.**

**AIM:**

To familiarize with advance Linux commands like curl, wget, ftp, ssh and grep.

**DESCRIPTION:**

**curl**

- It is easy to use to send and receive files and support major IPs.
- Installation is done by the command

```
sudo apt-get install curl
```

- To run the command use the method in the following example:

```
curl sample.com
```

- This shows the entire http content on that sample.com url.
- To save the output into a file use:

```
curl sample.com > sample.html
```

*or*

```
curl -o sample.html sample.com
```

*where .html is the output file and .com is the url*

- To download multiple files we can use:

```
curl -o sample.html sample.com -o sample1.html  
sample1.com
```

*where .html is the output file and .com is the url*

**wget:**

- It retrieves file from www using protocols like http,https,ftp.

(i) Single file download

- This is used to download single file and stores in current directory.
- It shows download progress, size, date, etc.
- Format used is:

```
wget<space> URL
```

(ii) Download file with other name

- Format used is:

```
wget<space>-O<space>filename<space>URL
```

(iii) Download multiple files

- There is provision to download multiple files using http and ftp protocol in single wget command.
- Format used is:

```
wget<space>http_URL<space>ftp_URL
```

(iv) Read URLs from a file

- Number of URLs can be in a text file and can download with `-i` option.
- Format used is:

```
wget<space>-i<space>file_location
```

(v) Resume an uncompleted file

- Download of a file can be resumed with `-c`.
- Format used is:

```
wget<space>-c<space>URL
```

**sftp:**

- It is used as secured file transport protocol.
- The format used is:  
sftp<space>username@remotehostname or IP  
Eg: sftp [ubuntu@192.168.1.168](#)
- Then press yes to continue and type user password to log in.

- To get a remote file use the command:  
get<space>filename
- To put a remote file use the command:  
put<space>filename

## ssh

- It stands for secured shell and is used to securely log onto remote systems.
- It is common way to access remote system.
- If telnet is used, no encryption happens and data can be easily hacked.
- But ssh is secured as it has encrypted data.

## grep

- It is used to search or filter.
  - (i) It can be used to search a word.  
Eg: `dpkg -l | grep -i python`  
*Here it is used to search all packages and filter out and return everything with "python" in it.*
  - (ii) It can be used to avoid or skip some characters or words.  
Eg: `grep -v "#" /etc/adduser.conf`  
*Here it avoids or skips all line with "#" in adduser.conf*
  - (iii) find : It can used to filter by providing possible constrains.  
Eg: `find -name "*.mp3" | grep -i rock | grep -vi "remix"`  
*- Here we provide the pattern name and find all files with .mp3 extension.*  
*-Then pipe to grep and filter out and print all files with name "rock".*

- Another piping is done to grep and filter out to avoid printing all files with string "remix".

(iv) It is used to search file by given string

Eg: `grep -n "main" setup.py`

*which retrieves line number of string to be searched, i.e, "main" in the file "setup.py"*

(iv) It is also used to search a string recursively in all directions.

Eg: `grep -r "function" *`

*this searches all directories and subdirectories searching "function" recursively (-r). -rn can be used to search recursively with line number.*

## **RESULT AND DISCUSSION:**

Familiarized with advanced Linux commands like curl, wget, sftp, ssh and grep.

## **VIVA QUESTIONS:**

- 1.What is the difference between curl and wget?
- 2.How to use curl command to download a file?
- 3.Give the use of sftp command.
- 4.Give the uses of grep command?



EXP NO.  
4

**SHELL PROGRAMMING FOR VARIOUS SYSTEM CONFIGURATIONS (SET 1)**

**AIM:**

To implement a shell script to show various system configurations like (a) currently logged user and his login name, (b) Your current shell, (c) Your home directory, (d) Your operating system type, (e) Your current path setting, (f) Your current working directory, and (g) Number of users currently logged in.

**DESCRIPTION:**

Step 1: Start

Step 2: Allocate variables for the commands to retrieve the necessary system configurations.

Step 3: Use the commands `who` and `whoami` within back-quotes to retrieve currently logged user and his login name respectively.

Step 4: Use the special command `$SHELL` within double-quotes to retrieve the current shell.

Step 5: Use the character `~` to retrieve the home directory.

Step 6: Use the command `uname -a` within back-quotes to retrieve operating system type.

Step 7: Use the special command `$PATH` within double-quotes to retrieve current path setting.

Step 8: Use the command `pwd` within back-quotes to retrieve the current working directory.

Step 9: Use the command `who | wc -l` within back-quotes to retrieve the number of users currently logged in.

Step 10: Stop

## **RESULT AND DISCUSSION:**

Shell program for system configurations executed correctly and output was verified.

## **VIVA QUESTIONS**

1. Which is the core of the operating system?
2. Applications communicate with kernel by using:
  - a) System Calls
  - b) C Programs
  - c) Shell Script
  - d) Shell
3. Which command is used to retrieve the current shell?
4. Which command is used to retrieve the current working directory?
5. Which command is used to retrieve the number of currently logged users?

EXP NO.  
5

**SHELL PROGRAMMING FOR VARIOUS SYSTEM CONFIGURATION (SET 2)**

**AIM:**

To implement a shell script to show various system configurations like (a) your OS and version, release number, kernel version, (b) all available shells, (c) computer CPU information like processor type, speed, (d) memory information (e) hard disk information like size of hard-disk, cache memory, model, and (f) File system (Mounted).

**DESCRIPTION:**

Step 1: Start

Step 2: Allocate variables for the commands to retrieve the necessary system configurations.

Step 3: Use the commands `uname -o`, `uname -k` and `uname -v` within back-quotes to retrieve OS and version, release number and kernel version respectively.

Step 4: Use the command `cat /etc/shells` within back-quotes to retrieve all available shells.

Step 5: Use the command `cat /proc/cpuinfo` within back-quotes to retrieve the CPU information.

Step 6: Use the command `cat /proc/meminfo` within back-quotes to retrieve the memory information.

Step 7: Use the command `cat /proc/diskstats` within back-quotes to retrieve the hard-disk information.

Step 8: Use the command `cat /proc/mounts` within back-quotes to retrieve the file systems(mounted) information.

Step 9: Stop

## **RESULT AND DISCUSSION:**

Shell program for system configurations executed correctly and output was verified.

### **VIVA QUESTIONS:**

1. Give the use of these commands `uname -o`, `uname -k` and `uname -v`.
2. Give the use of `cat` command.
3. Which command is used to retrieve the memory information of the system.
4. Which command is used to retrieve the hard disk information of the system

EXP NO.  
6

**SHELL PROGRAMMING TO IMPLEMENT SIMPLE CALCULATOR**

**AIM:**

To implement a shell script for a menu driven calculator using the functions like addition, subtraction, multiplication, division and modulus.

**DESCRIPTION:**

Step 1: Start

Step 2: Allocate variables to store the numbers from user.

Step 3: Print the menu for the user using echo command.

Step 4: Allocate a variable to store the choice from user.

Step 5: If the choice is 1, then find sum of the two numbers and print the result.

Step 6: If the choice is 2, then find difference of the two numbers and print the result.

Step 7: If the choice is 3, then find product of the two numbers and print the result.

Step 8: If the choice is 4, then divide the two numbers and print the result.

Step 9: If the choice is 5, find modulus of two numbers and print the result.

Step 10: If the choice is “y” from user then again print the menu otherwise exit.

Step 11: Stop

**PROGRAM**

```
#!/bin/bash
```

```
clear
```

```
sum=0
i="y"
echo " Enter first no."
read n1
echo "Enter second no."
read n2
while [ $i = "y" ]
do
echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo "Enter your choice"
read ch
case $ch in
    1)sum=`expr $n1 + $n2`
    echo "Sum ="$sum;;
    2)sum=`expr $n1 - $n2`
    echo "Sub = "$sum;;
    3)sum=`expr $n1 \* $n2`
    echo "Mul = "$sum;;
    4)sum=`echo "scale=2;$n1/$n2"|bc`
    echo "div=" $sum;;
```

```
*)echo "Invalid choice";;  
  
esac  
  
echo "Do u want to continue ?[y/n]"  
  
read i  
  
if [ $i != "y" ]  
  
then  
  
    exit  
  
fi  
  
done
```

### **RESULT AND DISCUSSION:**

Shell program for simple calculator executed correctly and output was verified.

### **VIVA QUESTIONS:**

1. What is a shell?
2. Explain the working of the program?
3. Which command is used for printing output?

EXP NO.  
7

**VIRTUALISATION ENVIRONMENT TO TEST NEW APPLICATIONS AND KERNELS  
AND ISOLATE APPLICATIONS.**

**AIM:**

To implement virtualization concept in Linux by installing a new kernel within the existing using Virtualbox.

**DESCRIPTION**

Ubuntu's even versions are called as Long Term Support as its mission is a longer one than the odd versions. One of the most successful of them is Debian which is mainly used for security. Virtualization environment can be implemented using many tools like Virtualbox, QEMU, VMware, etc used according to the purpose. The implementation virtualization environment by installing Debian OS through Virtualbox is as follows:

**Install Virtualbox:**

Step 1: To install Virtualbox use the command:

```
sudo apt-get install Virtualbox
```

If the package is not present then use: `sudo apt-get update` and then type above command.

Step 2: Once installation is complete, type:

```
sudo Virtualbox
```

Step 3: First select *new* and then give the name and operating system.



Step 4: Give the memory size of RAM. Optimum size is around 2GB, but should not be less than 512Mb as it affects the Virtual machine.

Step 5: Select *create VHD now* in hard drive options. You can select *use existing VHD now*, if you require double boot in virtual machine.

Step 6: Select hard drive file type. As we are using iso image use *CD or iso image file*.

Step 7: Then choose storage type. It's better to select *Dynamic* because if specified memory is not enough, it automatically allocates the required quantity. If we choose *Fixed size*, you need to know the actual size required.

Step 8: Then choose file location and size. This is used to provide hard-disk space for the new operating system. The normal range of file location is around 16 Gb if the actual hard disk is around 2Tb.

Step 9: In settings, you can edit all settings related to system, processor, etc. if needed. Till now we have just created the environment. From *storage* browse the folders to select the iso image of the new OS.

Step 10: Click *start* to start the installation.

### **Installation of Debian:**

Step 1: Select *graphical installation* and then language as *English*, country as *India* and keyboard language as *American English*.

Step 2: Then provide the host name which will be saved in the host directory that reside within the etc directory of root.

Step 3: Then provide the domain name for the new OS.

Step 4: Provide the root password for the operating system.

Step 5: Then provide user name and log-in password. The new version of Debian allows both root password and log-in password to be same.

Step 6: Select *guide-use entire disk* and then select *All files in one partition*.

Step 7: Then select *Finish partition* and then press *yes*.

Step 8: If you don't want any latest version then type *No*.

Step 9: Then press *yes* to install grub.

Step 10: Select */dev/sda/...* to complete the installation process.

### **RESULT AND DISCUSSION:**

Installation of Virtualbox and Debian operating system was done and worked properly.

### **VIVA QUESTIONS:**

1. Which command is used for installing Virtual box.
2. Explain the steps for Virtualbox installation?
3. What is the purpose of Debian OS?

<b>EXP NO.</b> 8	<b>PERFORM SIMPLE TEXT PROCESSING USING PERL</b>
---------------------	--

**AIM:**

To perform simple text processing using perl.

**DESCRIPTION**

Perl is a programming language developed by Larry Wall, especially designed for text processing. It stands for Practical Extraction and Report Language. It runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

**Basic Commands:**

## 1) Scalar Variables

A scalar is a single unit of data. That data might be an integer number, floating point, a character, a string, a paragraph, or an entire web page.

Eg: \$age = 25;

## 2) Array Variables

An array is a variable that stores an ordered list of scalar values. Array variables are preceded by an "at" (@) sign.

Eg: @ages = (25, 30, 40);

## 3) Adding and Removing Elements in Array

Perl provides a number of useful functions to add and remove elements in an array.

## 1. push @ARRAY, LIST

Pushes the values of the list onto the end of the array.

2. `pop @ARRAY`

Pops off and returns the last value of the array.

3. `shift @ARRAY`

Shifts the first value of the array off and returns it, shortening the array by 1 and moving everything down.

4. `unshift @ARRAY, LIST`

Prepends list to the front of the array, and returns the number of elements in the new array.

#### 4) Slicing Array Elements

You can also extract a "slice" from an array - that is, you can select more than one item from an array in order to produce another array.

```
#!/usr/bin/perl
@days = qw/Mon Tue Wed Thu Fri Sat Sun/;
@weekdays = @days[3,4,5];
print "@weekdays\n";
```

#### 5) Transform Strings to Arrays

```
split [ PATTERN [ , EXPR [ , LIMIT ] ] ]
```

This function splits a string into an array of strings, and returns it. If `LIMIT` is specified, splits into at most that number of fields. If `PATTERN` is omitted, splits on whitespace.

## 6) Transform Arrays to Strings

`join EXPR, LIST`

This function joins the separate strings of `LIST` into a single string with fields separated by the value of `EXPR`, and returns the string.

## 7) Sorting Arrays

The `sort()` function sorts each element of an array according to the ASCII Numeric standards.

`sort [ SUBROUTINE ] LIST`

This function sorts the `LIST` and returns the sorted array value. If `SUBROUTINE` is specified then specified logic inside the `SUBROUTINE` is applied while sorting the elements.

## **RESULT AND DISCUSSION:**

Familiarized with simple text processing using perl.

## VIVA QUESTIONS:

1. How to perform text operations using perl?
2. Give two examples of array processing commands in perl?
3. Which function is used to sort array example. Give an example.

EXP NO. 9	KERNEL CONFIGURATION, COMPILATION AND INSTALLATION
--------------	--

**AIM:**

To familiarize with Kernel configuration, compilation and installation by downloading the source code and the compiling and installing the same in the local system.

**DESCRIPTION:**

The kernel is indeed a computer program which forms the core of the operating system. The complete control of the system is handled by this set of code and it is the first program loaded on start-up. It handles the rest of start-up as well as input/output requests from software, translating them into data-processing instructions for the central processing unit. It handles memory and peripherals like keyboards, monitors, printers, speakers. The steps in its installation are as follows:

Step1: First download the source file of the kernel by the following command:

```
sudo apt-get source linux-image-$(uname -r)
```

Step2: Now we need to install the dependencies concerned with the kernel source code and to do it use the following command:

```
sudo apt-get build dep linux-image-$(uname -r)
```

Three files will be downloaded after the dependencies are installed and extract the compressed file.

Step 3: Change the current directory to extracted location and then quick build.

Step 4: Now install each of .deb extension files as follows:

```
sudo dpkg -I linux 4.8.0-17.19* .deb
```

Step 5: Now reboot the system and the system will be loaded with the newly installed kernel.

Step 6: You can customize the kernel configuration like we can disable USB drives from the local machine by editing the kernel file and once this kernel is rebooted the next time system will be customized accordingly.

### **RESULT AND DISCUSSION:**

Kernel configuration, compilation and installation was done and it worked properly.

### **VIVA QUESTIONS:**

1. Explain about the Kernel Installation Steps?
2. Give the command used to installing the kernel.

**EXP NO.**  
10

**SHELL PROGRAMMING TO START ON BOOT-UP AND KILL EVERY PROCESS WHICH USES MORE THAN A SPECIFIED AMOUNT OF MEMORY IN CPU.**

**AIM:**

To implement a shell script to start on boot-up and kill every process which uses more than a specified amount of memory in CPU.

**DESCRIPTION:**

Step 1: Start

Step 2: Find the CPU usage from the fields corresponding to uptime command.

Step 3: Set a maximum threshold value.

Step 4: Find the running processes using ps command and then sort them according to the amount of CPU usage.

Step 5: Check if the CPU usage is higher than threshold value, and if true then kill such processes.

Step 6: Repeat step 4 and 5 until the top most process in the list has CPU usage below the threshold value.

Step 7: Stop

**RESULT AND DISCUSSION:**

Shell program executed correctly and output was verified.

**VIVA QUESTIONS:**

1. How to implement a boot-up process using SHELL Script ?
2. How to kill process using SHELL Script?