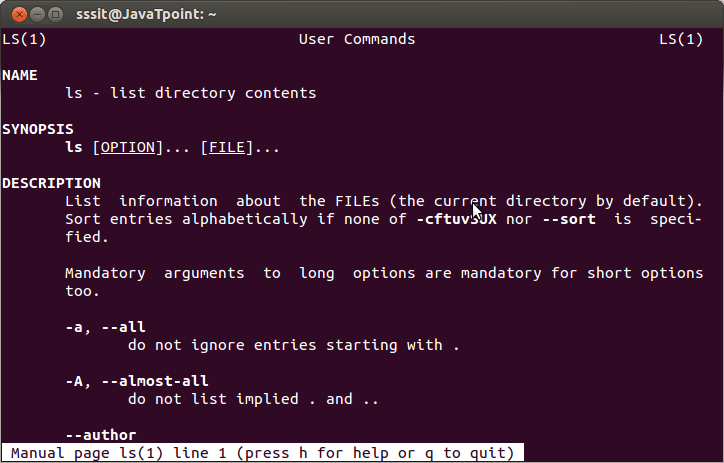
**LAB CYCLE:2 Date:**

**EXPERIMENT NO:4**

**BASIC LINUX COMMANDS**

**Aim:** Familiarity with following commands/operations excepted.

**1.man:** is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.



**2. ls, echo, read**

**ls:** is a Linux shell command that lists directory contents of files and directories. Some practical examples of ls command are shown below.

Options:

-t: It sorts the file by modification time, showing the last edited file first.

-l: To show long listing information about the file/directory.

-lh: To display file size in easy-to-read format.

-a: To show all the hidden files in the directory.

**echo:** used for displaying lines of text or string which are passed as arguments on the command line.

**read:** is used to read from a file descriptor.

**3. more, less, cat**

**more**: reads files and displays the text one screen at a time.

**less:** used for filtering and viewing text files one screen page at a time.

**cat:** used to display the content of a file, copy content from one file to another, concatenate the contents of multiple files, display the line number, display $ at the end of the line, etc.

**4. cd, mkdir, pwd, find**

**cd:** cd command in linux known as change directory command. It is used to change current working directory.

Syntax: cd [directory\_name]

**mkdir:** mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories.

Syntax: mkdir [options...] [directories ...]

**pwd:** pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

Syntax: pwd -L: Prints the symbolic path.

pwd -P: Prints the actual path.

**find:** The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.

Syntax: find [where to start searching from]

[expression determines what to find] [-options] [what to find]

**5. mv, cp, rm, tar**

mv: mv stands for move. mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions:

(i) It renames a file or folder.

(ii) It moves a group of files to a different directory.

Syntax: mv [Option] source destination

**cp:** cp stands for copy. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. cp command require at least two filenames in its arguments.

Syntax: cp [OPTION] Source Destination

**rm:** rm stands for remove here. rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references.

Syntax: rm [OPTION]... FILE...

**tar:** The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

Syntax: tar [options] [archive-file] [file or directory to be archived]

**6.wc, cut, paste**

**wc:** wc stands for word count.It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.

Syntax: wc [OPTION]... [FILE]...

**cut:** The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field.

Syntax: cut OPTION... [FILE]...

**paste:** paste command is one of the useful commands in Unix or Linux operating system. It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output.

Syntax: paste [OPTION]... [FILES]...

**7. head, tail, grep, expr:**

**head:** it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

Syntax: head [OPTION]... [FILE]...

**tail:** it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is precedes by its file name.

Syntax: tail [OPTION]... [FILE]...

**grep:** the grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

Syntax: grep [options] pattern [files]

**8. chmod, chown:**

**chmod:** the chmod command is used to change the access mode of a file.The name is an abbreviation of change mode.

Syntax: chmod [reference][operator][mode] file...

**chown:** it is used to set read, write, execute permission for user. To protect and secure files and directory in Linux we use permissions to control what a user can do with a file or directory.

Syntax: chown [OPTION]… [OWNER][:[GROUP]] FILE…

chown [OPTION]… –reference=RFILE FILE…

**9. Redirections & Piping:**

**Redirection:** whenever an individual runs a command, it can take input, give output, or do both. Redirection helps us redirect these input and output functionalities to the files or folders we want, and we can use special commands or characters to do so.

Append redirection

“>>” standard output

“<<” standard input

Overwrite Redirection

“>” standard output

“<” standard input

**Piping:** Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command’s output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes. The command line programs that do the further processing are referred to as filters.

Syntax: command\_1 | command\_2 | command\_3 | .... | command\_N

**10. useradd, usermod, userdel, passwd:**

**useradd:** is a command in Linux that is used to add user accounts to your system. It is just a symbolic link to adduser command in Linux and the difference between both of them is that useradd is a native binary compiled with system whereas adduser is a Perl script which uses useradd binary in the background.

Syntax: useradd [options] name\_of\_the\_user

**usermod:** is a command in Linux that is used to change the properties of a user in Linux through the command line. After creating a user we have to sometimes change their attributes like password or login directory etc. so in order to do that we use the Usermod command.

Syntax: sudo usermod -c "This is test user" test\_user

**userdel:** is used to delete a user account and related files. This command basically modifies the system account files, deleting all the entries which refer to the username LOGIN. It is a low-level utility for removing the users.

Syntax: userdel [options] LOGIN

**passwd:** passwd command in Linux is used to change the user account passwords. The root user reserves the privilege to change the password for any user on the system, while a normal user can only change the account password for his or her own account.

Syntax: passwd [options] [username]

**11. df,top, ps:**

**df:** is used to displays the amount of disk space available on the file system containing each file name argument.

Syntax: df [OPTION]...[FILE]...

**top:** top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

**ps:** it allows multiple processes to operate simultaneously without interfering with each other. Process is one of the important fundamental concept of the Linux OS.

Syntax: ps [options]

**12. ssh, scp, ssh-keygen, ssh-copy-id:**

**ssh:** stands for “Secure Shell”. It is a protocol used to securely connect to a remote server/system. ssh is secure in the sense that it transfers the data in encrypted form between the host and the client. It transfers inputs from the client to the host and relays back the output. ssh runs at TCP/IP port 22.

Syntax: ssh user\_name@host(IP/Domain\_name)

**scp:** scp (secure copy) command in Linux system is used to copy file(s) between servers in a secure way. The SCP command or secure copy allows secure transferring of files in between the local host and the remote host or between two remote hosts. It uses the same authentication and security as it is used in the Secure Shell (SSH) protocol. SCP is known for its simplicity, security and pre-installed availability.

Syntax: scp [-346BCpqrTv] [-c cipher] [-F ssh\_config] [-i identity\_file] [-l limit] [-o ssh\_option] [-P port] [-S program] [[user@]host1:]file1 … [[user@]host2:]file2

ssh-keygen: Secure Shell(SSH) is a cryptographic network protocol used for operating remote services securely. It is used for remote operation of devices on secure channels using a client-server architecture that generally operates on Port 22. SSH is the successor of Telnet. SSH uses public and private keys to validate and authenticate users. ssh-keygen is used to generate these key pairs.

**ssh-copy-id:** is a command that copies the public key of the local host to the authorized keys file of the remote host, allowing password-less and automatic login via SSH

**BASIC LINUX COMMANDS ACTICITY QUESTIONS**

1. Command to display the following message as Use “New line).

“God!Bless us.

We are starting Shell Scripting”

**Command:**

echo -e "\”God! Bless us\n We are starting Shell Scripting \”"

1. Get the manual page of ‘ls’ command. Search for the word “alphabet”. Find the next occurrence and then find the previous occurrence.

Command:

man ls

/alphabet

man ls -N

1. Read your name from the keyboard and display it.

Command:

read -p "enter your name:" n

echo $n

1. Create the directory structure dir1/dir4 and dir1/dir2/dir3 with a single command and then change directory to dir3.

Commands:

mkdir -p dir1/dir4 dir1/dir2/dir3

ls

cd dir1/dir3

1. Create some files using Vim.

creating a file using vim:

Start vim by typing vim filename

To insert text press i

Now start editing text. Add new text or delete unwanted text.

One can press Esc and type :wq to save changes to a file and exit from vim]

Command:

syntax: vim test.txt

1. Display the current directory.

Command:

pwd

1. Listing files and folders.
2. Listing the contents of dir1 (Qn.4) and all its descendants.

Command:

ls -R

1. Listing the contents of dir3 (Qn.4) in
   1. Alphabetical order

Command:

ls -l

* 1. Sorted on Time of modification, newest first

Command:

ls -lt

* 1. Sorted on Size

Commands:

ls -S

* 1. Reverse of all above

Command:

ls -r

* 1. Long listing of files Sorted on Size with smallest first and size

Command:

ls -lSr

* 1. Displayed in human readable form

Command:

ls -lhS

1. Execute **ls** and store the output to a file lsoutput

Command:

ls > lsoutput

cat lsoutput

1. Display the file
2. starting with the first 10 lines and

Command:

head -10 num.txt

1. starting with the 10th line with provision for

tail -n +10 num.txt >> out2.txt

1. Scrolling Up

Command:

more -10 out2.txt

1. Scrolling Up and Down

Command:

less -10 out2.txt

1. Execute **ls -l** and add the output to lsoutput, at the end.

Command:

ls -l >> lsoutput.txt

cat lsoutput.txt

1. Execute **ls -l** and feed the result to less command, to scroll through the directory listing.

Command:

ls -l |less

12. Copy the file file1 to newfile.

1. If newfile already exists, it should be replaced.

Command:

cp file1 newfile

1. If newfile already exists, it should not be replaced.

cp -n file2 newfile

1. If newfile already exists, it should be replaced, but only with the consent of the user.

Command:

cp -i file1 newfile

1. If newfile already exists, it should be replaced only if its contents is older than that of file1.

Command:

cp -u file1 newfile

1. Even if newfile is read only.

Command:

chmod u+wx newfile

1. Create a link instead of copying.

Command:

cp -s file3.txt file4.txt

1. Copy the entire directory tree from dir1 of Qn.4 to a new directory dir5

Command:

cp -R dir1 dir5

13. Create a new directory, dir6 inside dir1

1. Move all files in dir5 into it.

Command:

mv dir5 dir1/dir6

1. Delete all files where the name starts with a vowel character, upper or lower case.

Command:

rm [a,e,i,o,u,A,E,I,O,U]\*

1. Delete all files where the name is at least 3 characters long.

Command:

rm ???\*

1. Delete all hidden folders, and files.

Command:

rm -rf \*

14. Create a file testfile1 using Vim

vim testfile1

1. Set line number

vim testfile1

Press esc

Type :set number

1. Type your name and address with district and pincode
2. Copy paste the contents 10 times
3. Replace all occurrence of your district with a neighbouring district