Experiment No : 2

Aim: Study of a terminal based text editor such as Vim or Emacs. (By the end of the course,
students are expected to acquire following skills in using the editor: cursor operations,
manipulate text, search for patterns, global search and replace) Basic Linux commands, familiarity with following commands/operations expected

Text Editor

A text editor is a type of computer program that edits plain text.Text editors are provided with operating systems and software development packages, and can be used to change files such as configuration files, documentation files and programming language source code.
A text editor is a computer program that lets a user enter, change, store, and usually print text(characters and numbers, each encoded by the computer and its input and output devices, arranged to have meaning to users or to other programs). Typically, a text editor provides an "empty" display screen (or "scrollable page") with a fixed-line length and visible line numbers.

A popular text editor in IBM's large or mainframe computers is called XEDIT. In UNIX systems, the two most commonly used text editors are Emacs and Vi . In personal computer systems, word processors are more common than text editors. However, there are variations of mainframe and UNIX text editors that are provided for use on personal computers. An example is KEDIT, which is basically XEDIT for Windows.

Vi Text Editor

A text editor, written by Bill Joy in 1976. Short for Visual Interface.The vi editor is elaborated as **vi**sual editor. It is installed in every Unix system. In other words, it is available in all Linux distros. It is user-friendly and

works same on different distros and platforms. It is a very powerful application. An improved version of vi editor is **vim**.

●It enables fast, simple, and effective text editing mostly based on simple key bindings.

●It provides fast and convenient moving around files and between files.

●One must learn a good number of commands to be proficient in vi.

●We'll be using vim - vi improved

●Written by Bram Moolenaar

●vi is just an alias to vim

VIM

Vim editor is one of the more popular text editors we use today. It is a clone of the Vi editor and is written by Bram Moolenaar. It is cross platform editor and available on most popular platforms like Windows, Linux, Mac and other UNIX variants.

Features of Vim

- Its memory footprint is very low
- It is command centric. You can perform complex text related task with few commands
- It is highly configurable and uses simple text file to store its configuration
- There are many plug-in available for Vim. Its functionality can be extended in great manner using these plug-in
- It supports multiple windows. Using this feature screen can be split into multiple windows
- Same as multiple windows, it also supports multiple buffers
- It supports multiple tabs which allows to work on multiple files

•It supports recording features which allows to record and play Vim commands in repeated manner

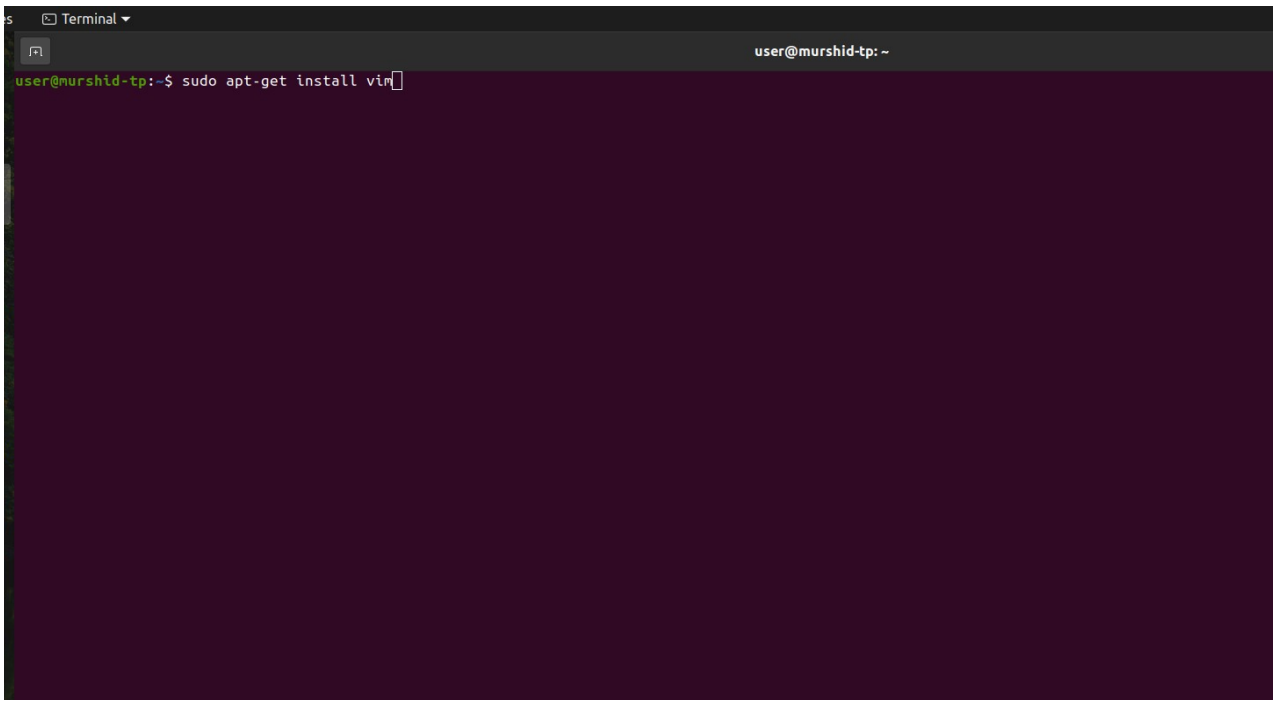
•Excellent for programming due to intelligent language detection.

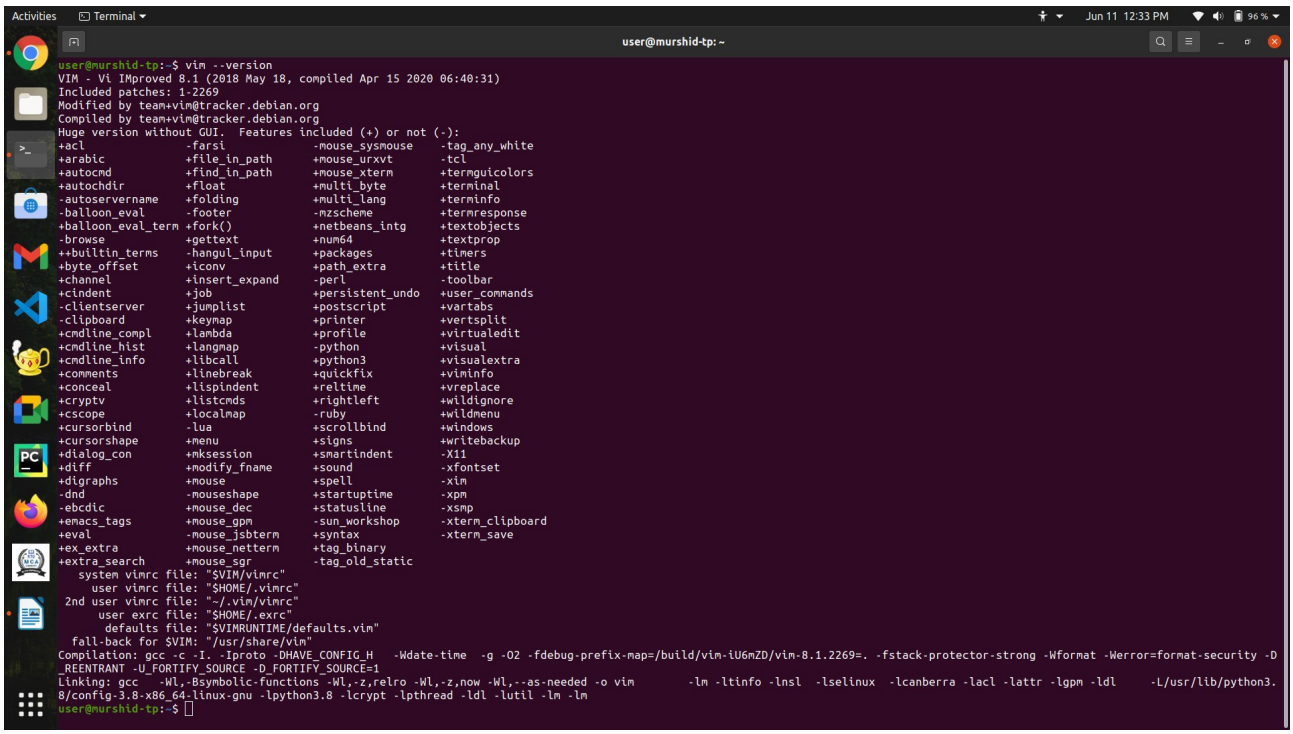
•NOT a formatting tool ... plain text only.

VIM Installation

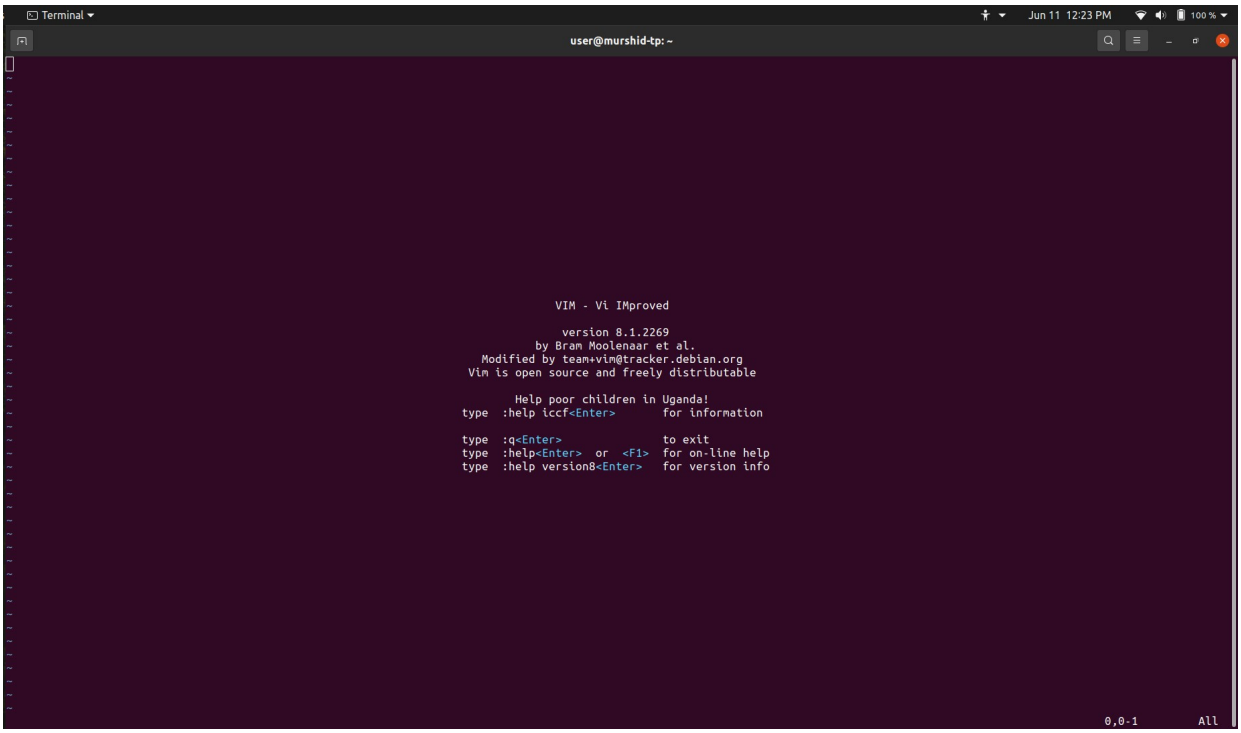
Open terminal application.

1. Update package database by typing the **sudo apt update** command
2. Install vim on Ubuntu Linux, type: **sudo apt install vim**
3. Verify vim installation by typing the **vim --version** command





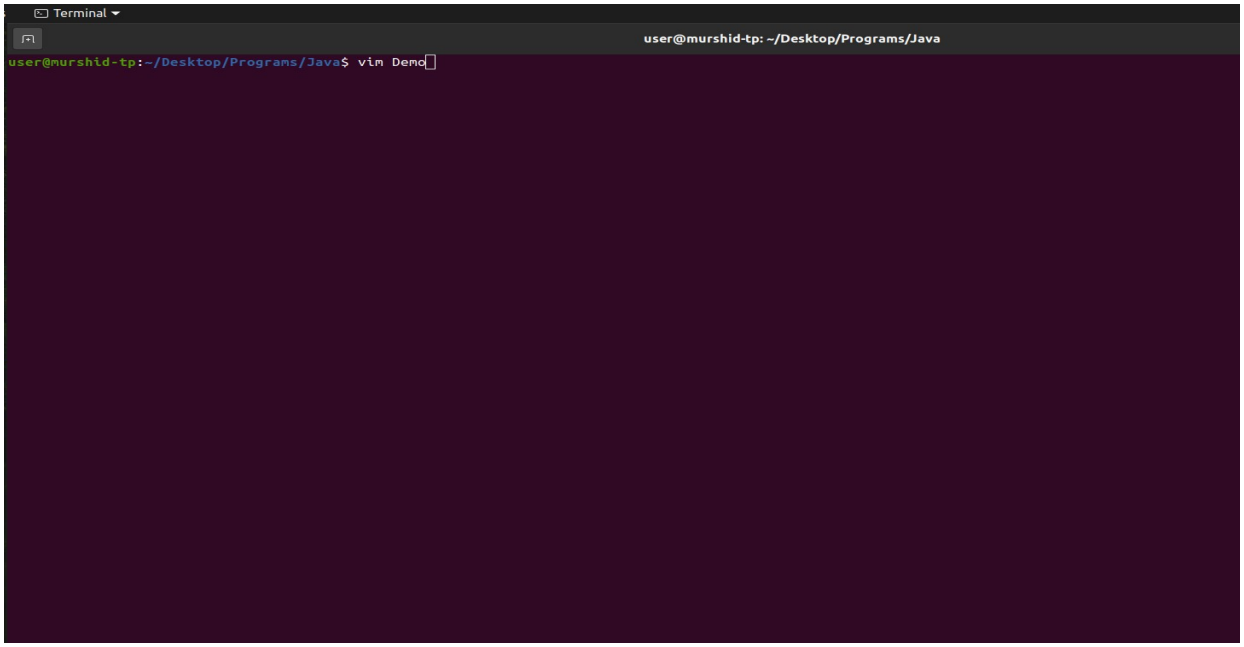
Open Vim **“vim -v”** by this command in terminal

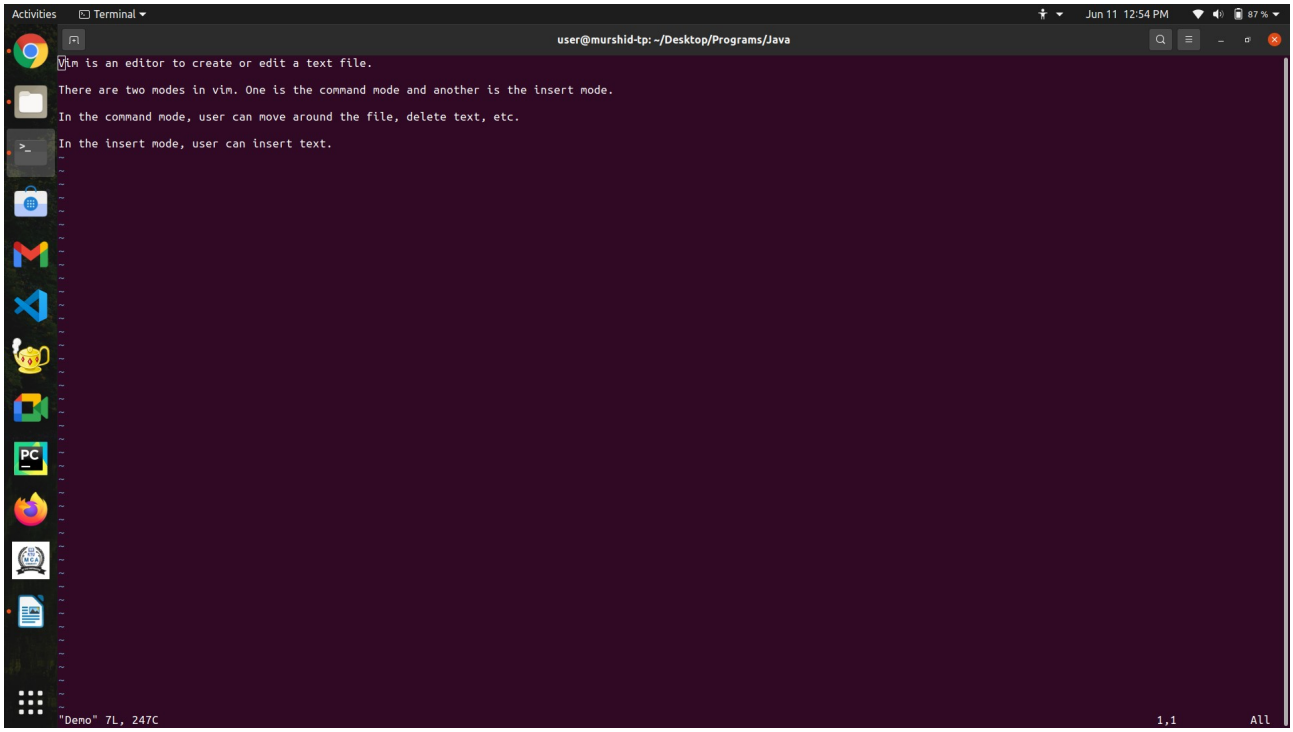


Open an existing file from vim,first go to the file directory and command

vim [your file]

Eg: vim Demo





Vim has three modes

➢ Command

➢ Input

➢ Last Line

1.Command Mode

When you start vim, you begin in Command Mode by default.Hitting ESCAPE will get you back to Command Mode from other modes.In this mode you can issue many vi commands, including commands like *insert*, *append*, and *delete*, and other search and navigation commands that let you move around your file.

2.Insert Mode

This mode allows you to enter text into your document. You can enter insert mode by pressing the **i** key. Keep in mind that to save your document, you'll need to go **back to command mode** since only text input is allowed in this mode.

3.Lastline Mode

The last vi mode is known as *vi last line mode*. You can only get to last line mode from command mode, and you get into last line mode by pressing the **colon key(:)**.After pressing this key, you'll see a colon character appear at the beginning of the last line of your vi editor window, and your cursor will be moved to that position. This indicates that vi is ready for you to type in a "last line command".

Some useful commands for VIM

Cursor movement

h - move cursor left

j - move cursor down

k - move cursor upl - move cursor right

w - jump forwards to the start of a word

W - jump forwards to the start of a word

e - jump forwards to the end of a word

E - jump forwards to the end of a word

b - jump backwards to the start of a word

B - jump backwards to the start of a word

0 - jump to the start of the line

^ - jump to the first non-blank character of the line

$ - jump to the end of the line

Insert mode

i - insert before the cursor

I - insert at the beginning of the line

a - insert (append) after the cursor

A - insert (append) at the end of the line

o - append (open) a new line below the current line

O - append (open) a new line above the current line

ea - insert (append) at the end of the word

Esc - exit insert mode

Editing

r - replace a single character

J - join line below to the current onec

c - change (replace) entire line

cw - change (replace) to the end of the word

c$ - change (replace) to the end of the line

s - delete character and substitute text

S - delete line and substitute text (same as cc)

xp - transpose two letters (delete and paste)

u – undo

Ctrl + r – redo

Cut and paste

yy - yank (copy) a line

2yy - yank (copy) 2 lines

yw - yank (copy) word

y$ - yank (copy) to end of line

p - put (paste) the clipboard after cursor

P - put (paste) before cursor

dd - delete (cut) a line

2dd - delete (cut) 2 lines

dw - delete (cut) word

D - delete (cut) to the end of the line

d$ - delete (cut) to the end of the line

x - delete (cut) character

Exiting commands

:w - write (save) the file, but don't exit

:wq - write (save) and quit

:x - write (save) and quit

:q - quit (fails if there are unsaved changes)

:q! - quit and throw away unsaved changes

Basic LINUX Commands

man

man command in Linux 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.

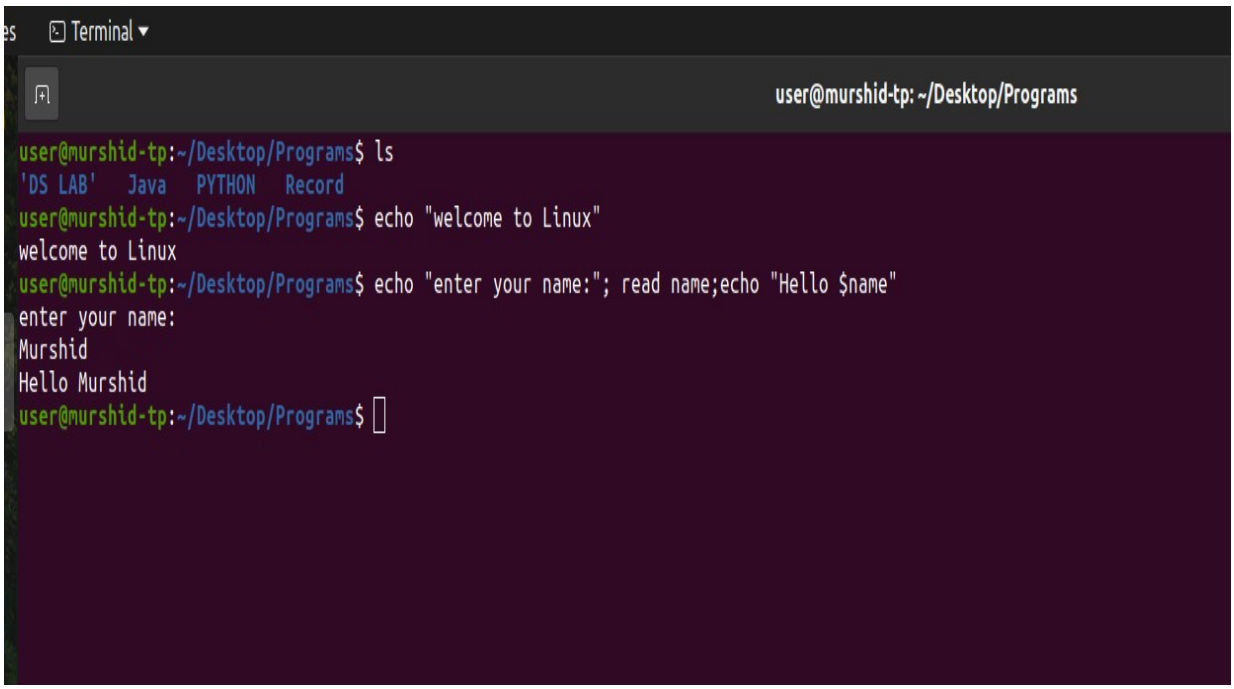
Syntax: $ man [COMMAND NAME]

Example: **$ man ls**

ls: ls is a Linux shell command that lists directory contents of files and directories.

echo : echo command in linux is used to display line of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

read : read command is used to obtain input from users. Understanding the “read” command is key to making your code more interactive. The “read” command is used to obtain inputted information from the user.



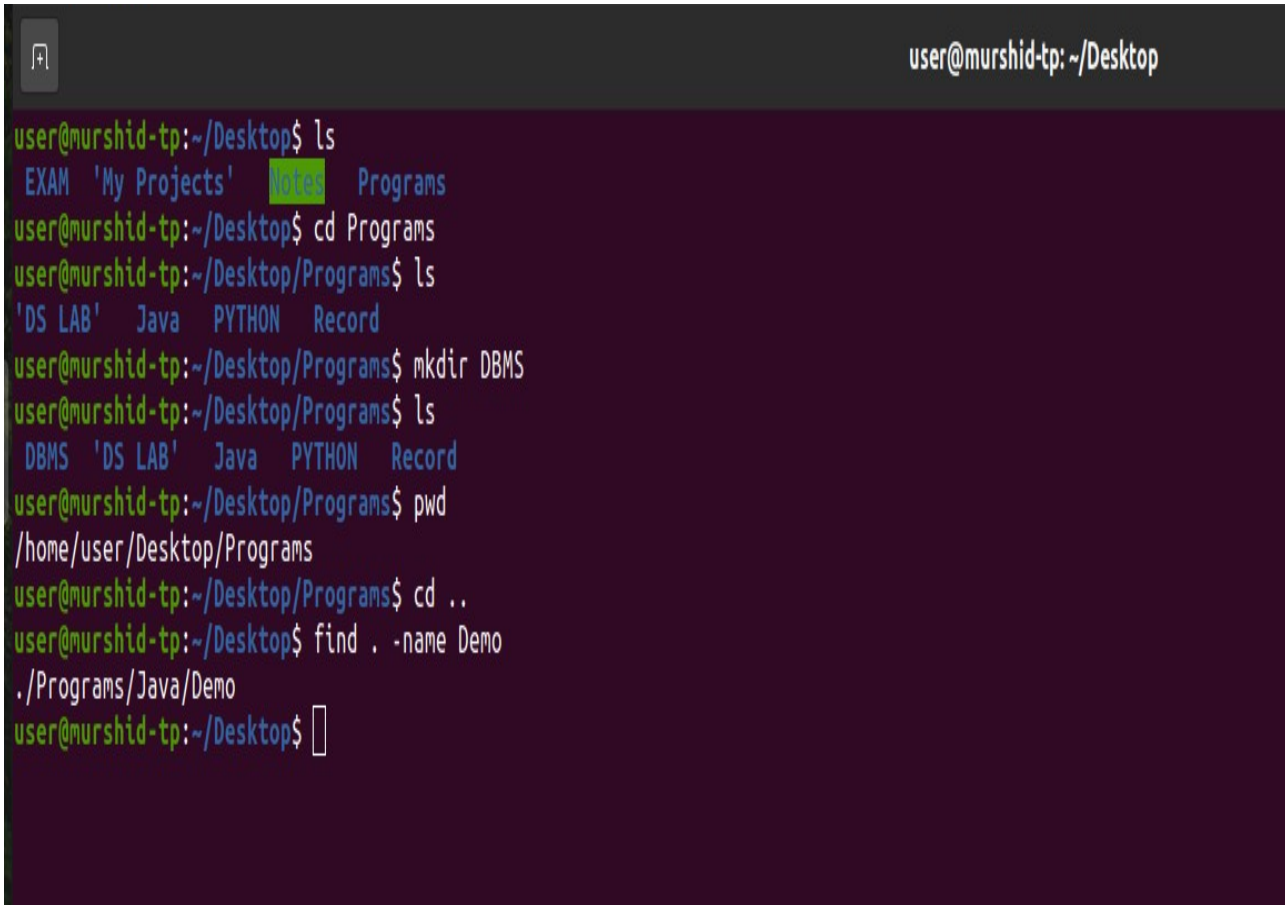
cd : cd command in linux known as change directory command. It is used to change current working directory.

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.

pwd : pwd stands for **P**rint **W**orking **D**irectory. It prints the path of the working directory, starting from the root.

find : It seaches for files and directories in a directory heirarchy based on a user given expression and can perform user specified action on each matched file.

Syntax: **$ find [where to start searching from][exprssion determines what to find] [-options] [what to find]**

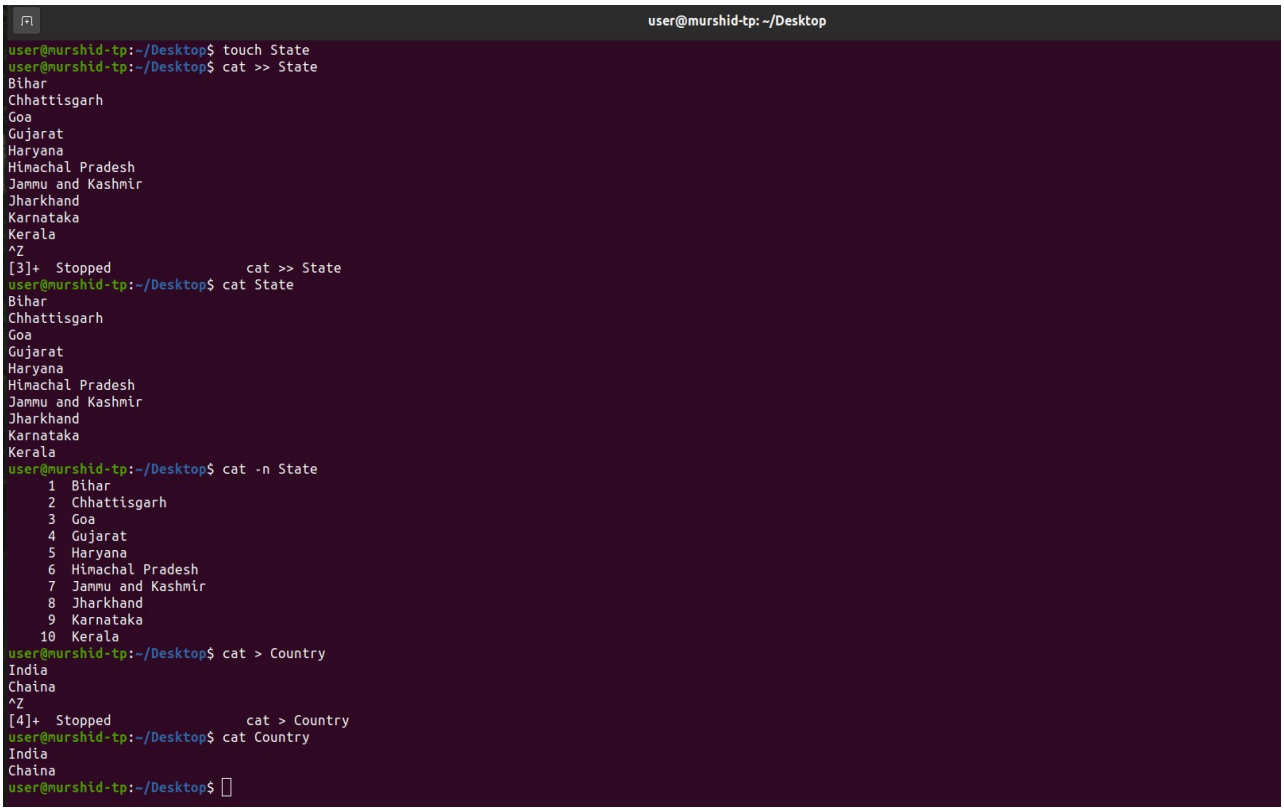


touch : It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.

cat : Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

Linux cat command usage

Option	Function
cat > [fileName]	To create a file.
cat [oldfile] > [newfile]	To copy content from older to new file.
cat [file1 file2 and so on] > [new file name]	To concatenate contents of multiple files into one.
cat -n/cat -b [fileName]	To display line numbers.
cat -e [fileName]	To display $ character at the end of each line.
cat [fileName] <<EOF	Used as page end marker.



locate : The locate command is used to search a file by file name. It is quite similar to find command; the difference is that it is a background process. It searches the file in the database, whereas the find command searches in the file system. It is faster than the find command. To find the file with the locates command, keep your database updated.

date and cal : The date command is used to display date, time, time zone, and more.

The **cal** command is used to display the current month's calendar with the current date highlighted.

ls -l : It will show the list in a long list format.It includes,

•number of links to the content

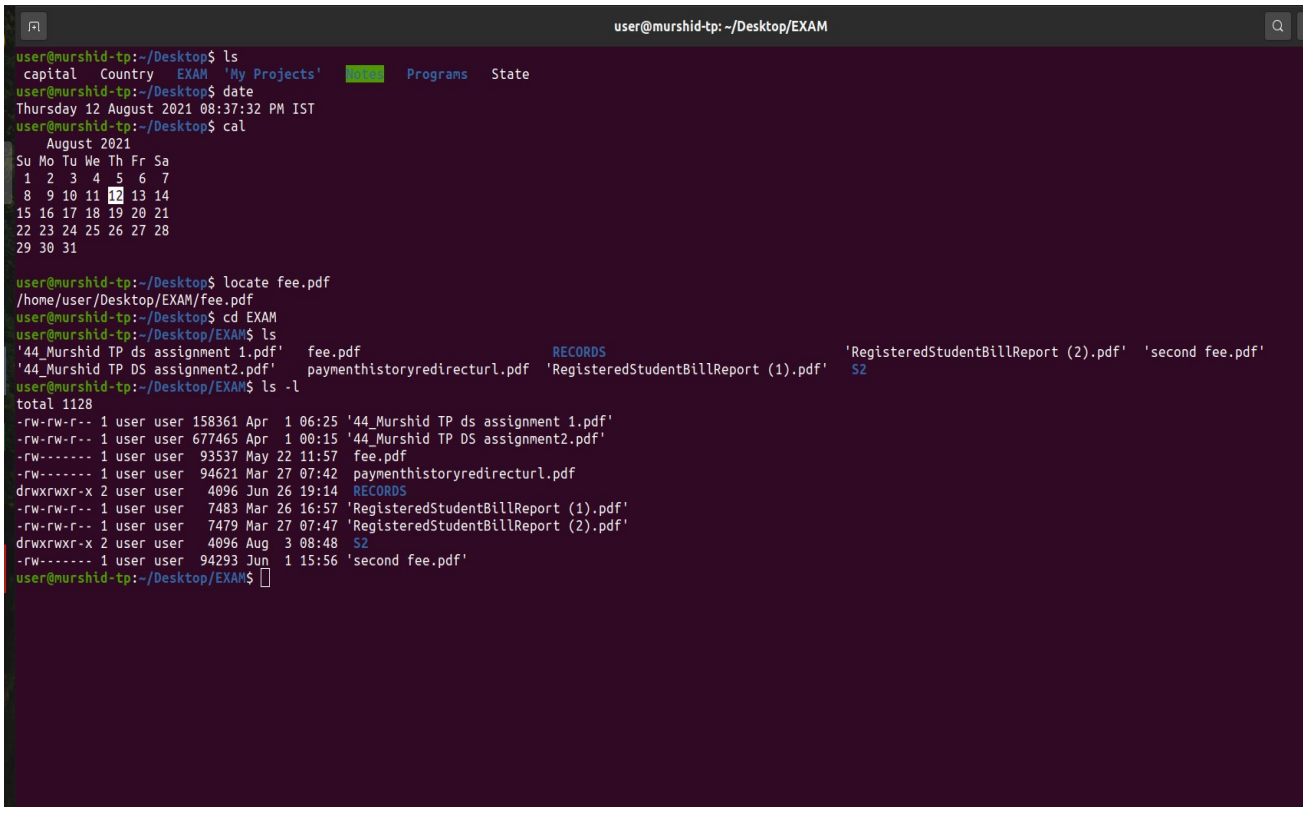
•owner of the content

•group owner of the content

•size of the content in bytes

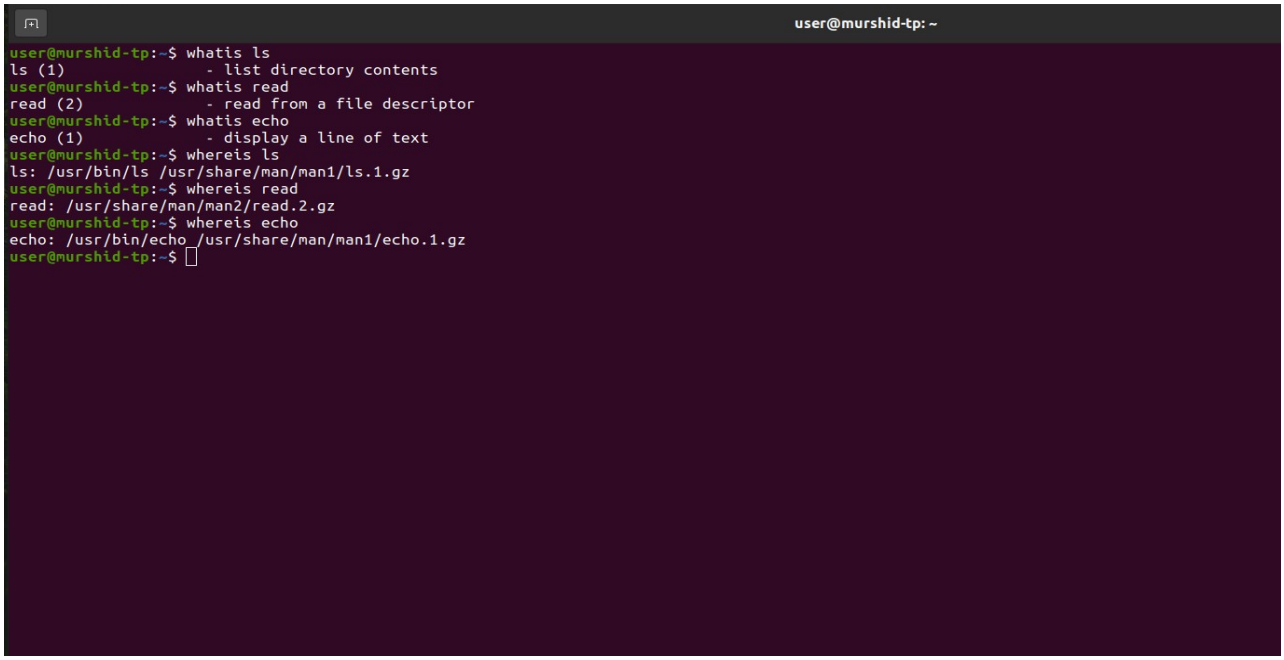
•last modified date / time of the content

•file or directory name

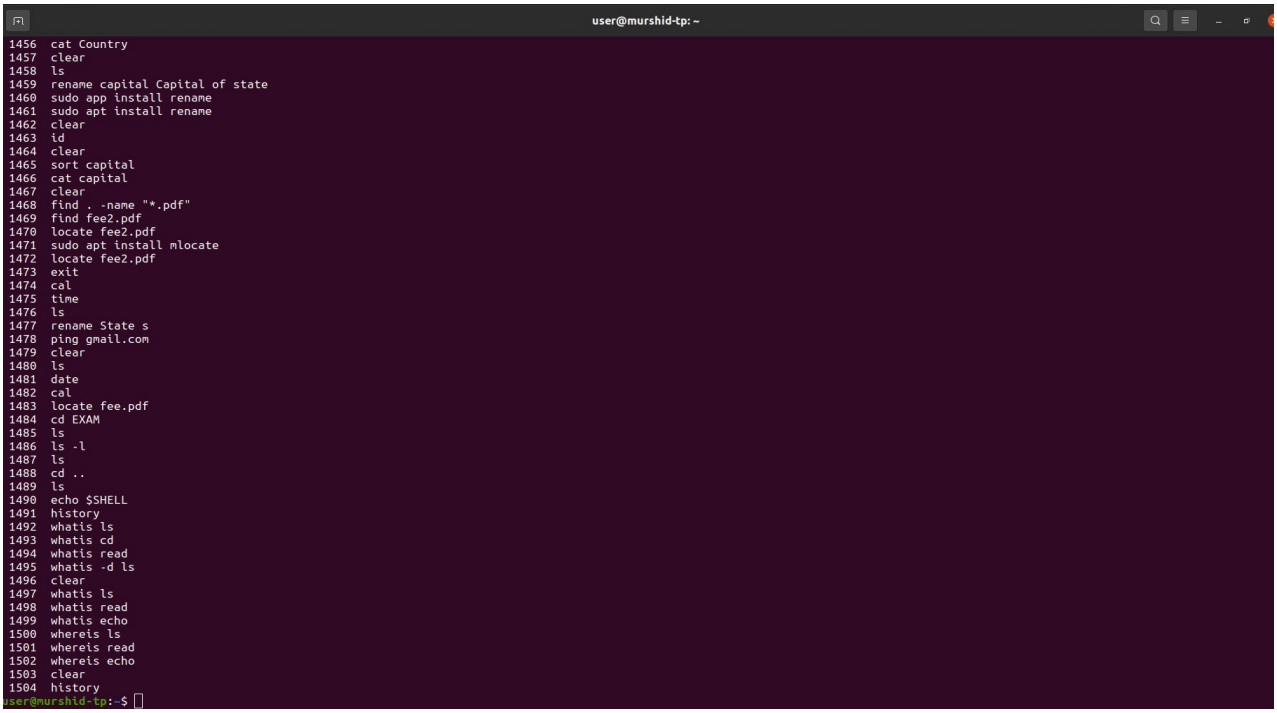


whatis :command in Linux is used to get a one-line manual page descriptions. In Linux, each manual page has some sort of description within it. So this command search for the manual pages names and show the manual page description of the specified filename or argument.

whereis : command is used to find the location of source/binary file of a command and manuals sections for a specified file in Linux system.

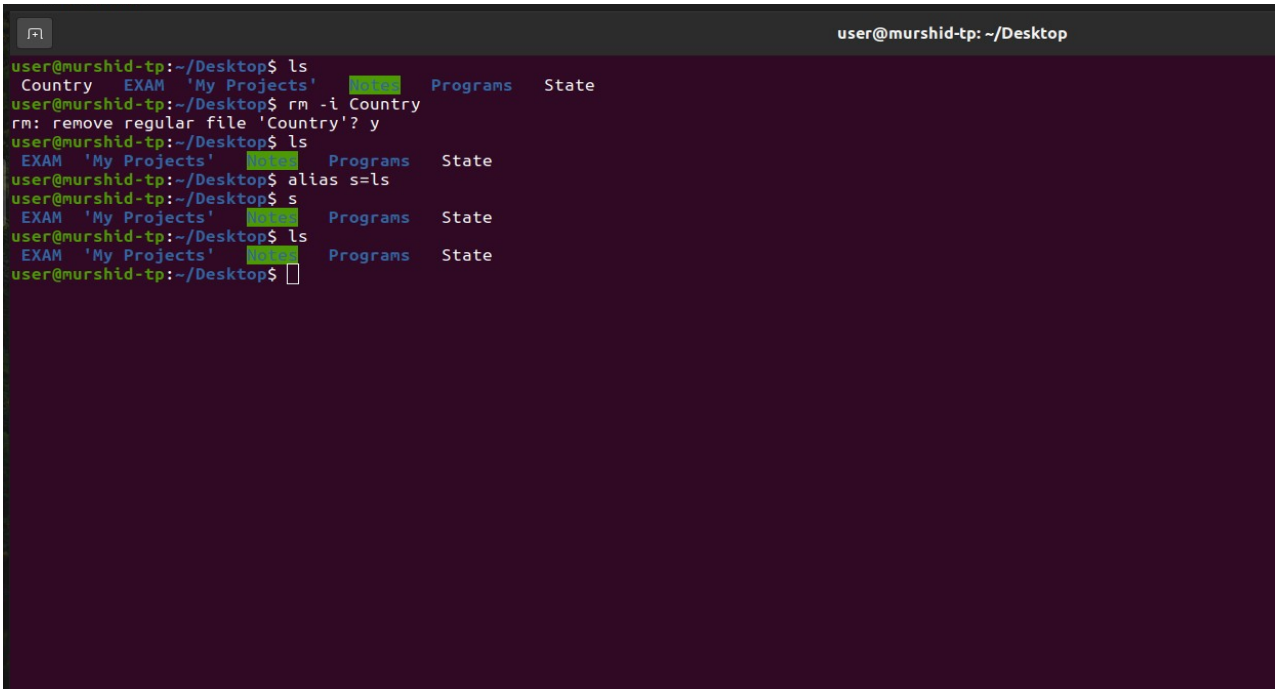


history : *history* command is used to view the previously executed command.

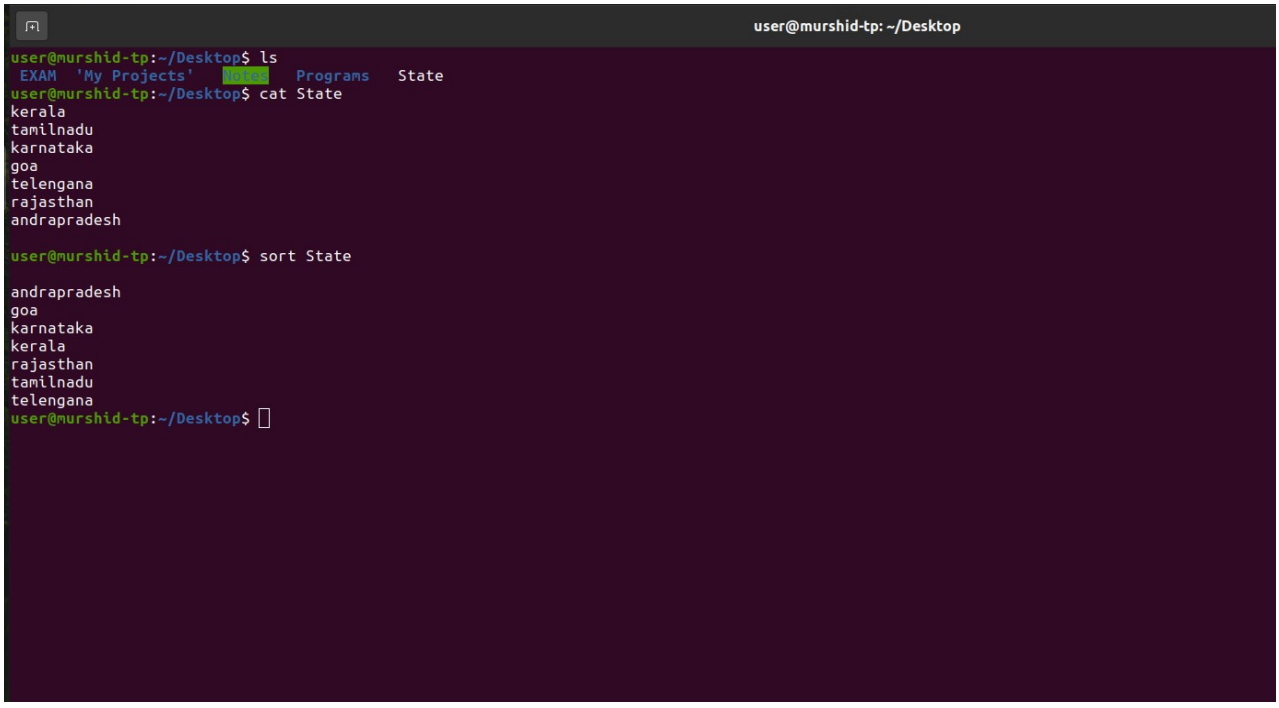


alias : Linux 'alias' command replaces one string from the shell with another string. It is a shell built-in command. It converts a complicated command into a simpler command or in other words, it creates a shortcut by replacing it with the simpler one.

rm -i : prompt before every removal.



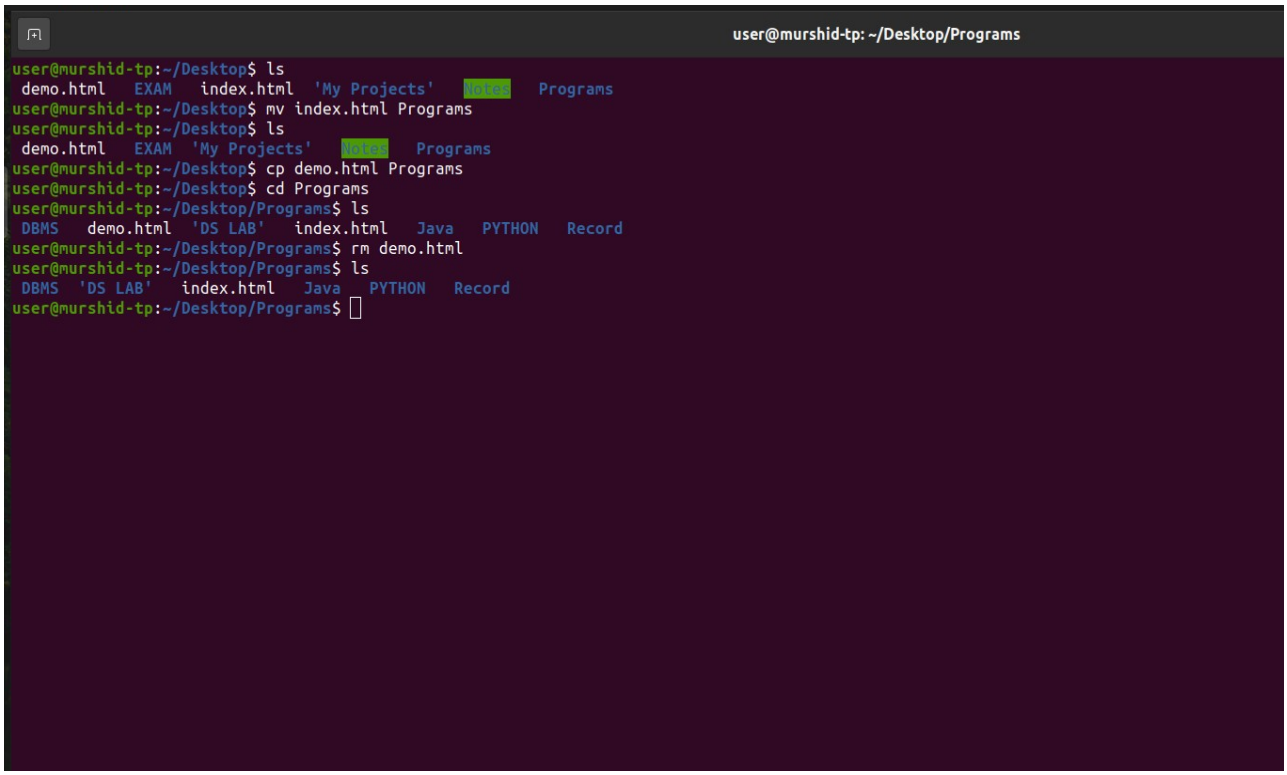
Sort : SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. It sort alphabeticaly.



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.

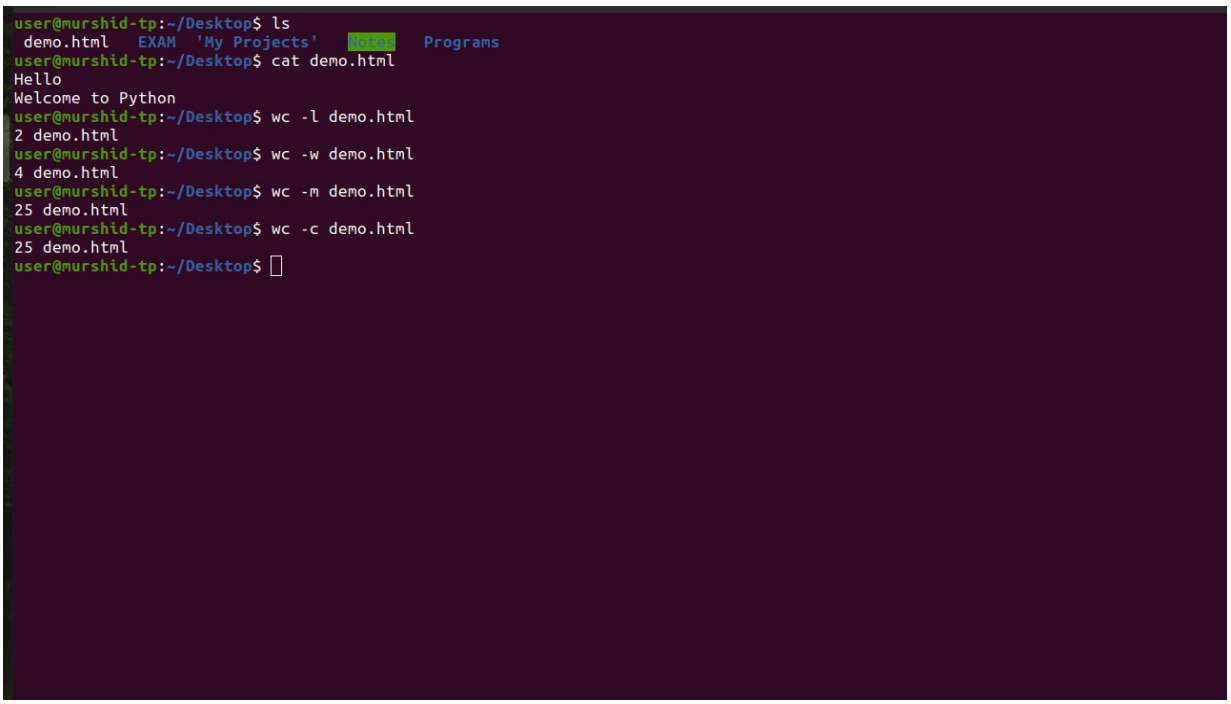
cp : To copy one or more files to another location.

rm : rm command is used to remove objects such as files, directories, symbolic links and so on from the file system.



wc : The **wc** (**word count**) command in Unix/Linux operating systems is used to find out number of **newline count**, **word count**, **byte and characters** count in a files specified by the file arguments.

wc -l : Prints the number of lines in a file.
wc -w : prints the number of words in a file.
wc -c : Displays the count of bytes in a file.
wc -m : prints the count of characters from a file.
wc -L : prints only the length of the longest line in a file.



cut : Linux cut command is useful for selecting a specific column of a file. It is used to cut a specific sections by byte position, character, and field and writes them to standard output. It cuts a line and extracts the text data. It is necessary to pass an argument with it; otherwise, it will throw an error message.

To cut a specific section, it is necessary to specify the delimiter. A delimiter will decide how the sections are separated in a text file. Delimiters can be a space (' '), a hyphen (-), a slash (/), or anything else. After '-f' option, the column number is mentioned.

-b, --bytes=LIST: It is used to cut a specific section by bytes.

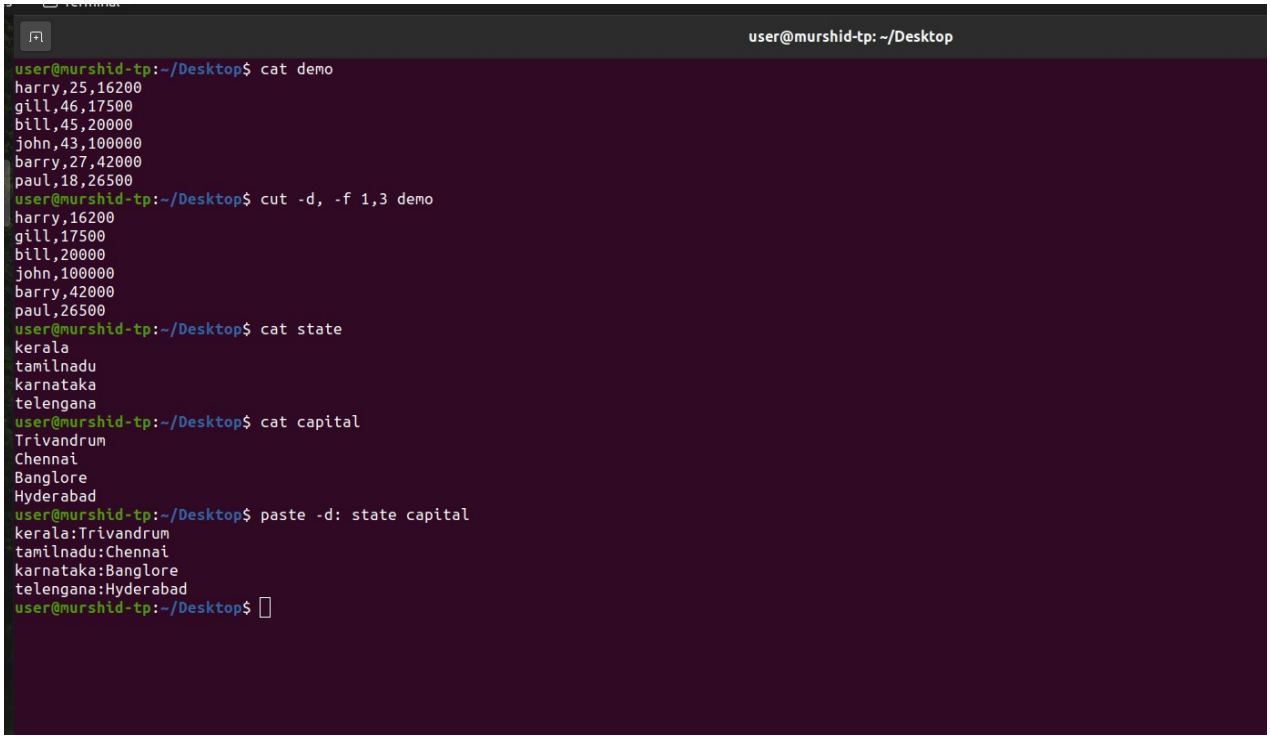
-c, --characters=LIST: It is used to select the specified characters.

-d, --delimiter=DELIM: It is used to cut a specific section by a delimiter.

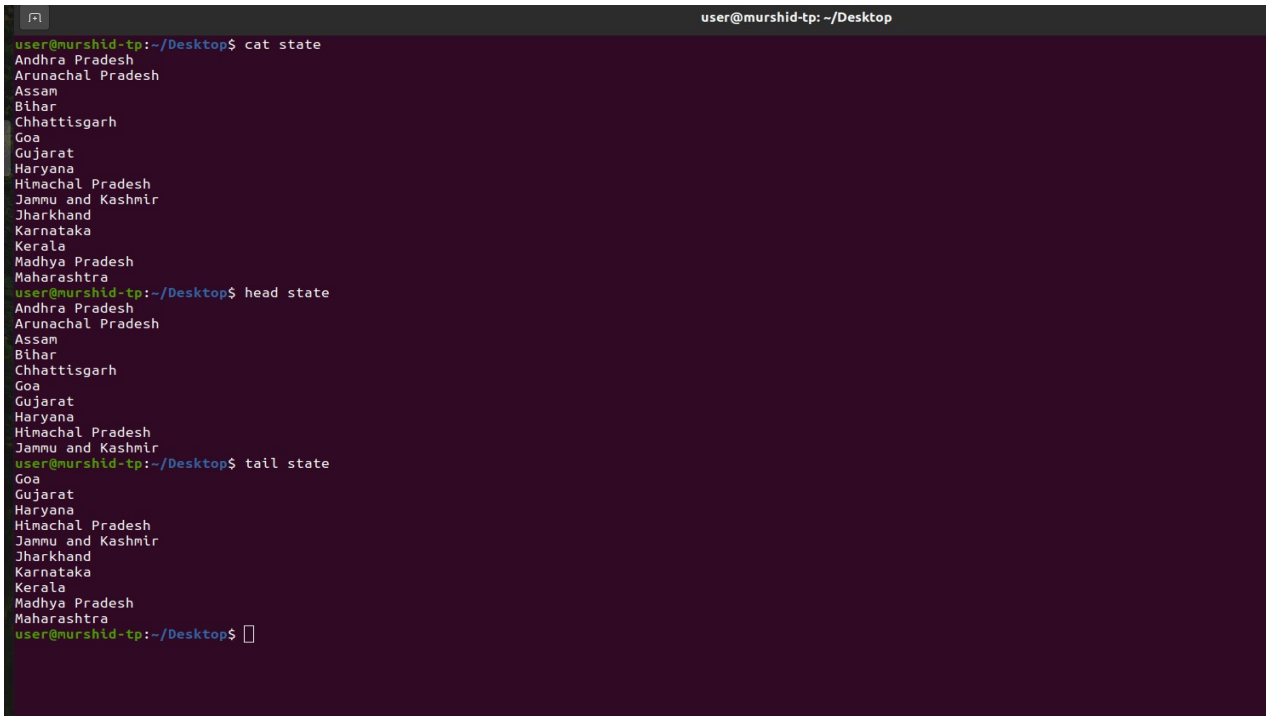
-f, --fields=LIST: It is used to select the specific fields. It also prints any line that does not contain any delimiter character, unless the -s option is specified.

-n: It is used to ignore any option.

paste : The paste command is useful for merging files together. The first line of each file is joined separated by a Tab character. It is possible to specify a different delimiter with the -d parameter.



<u>head and tail</u> : The head command, as the name implies, print the top N number of data of the given input. By default, 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.The tail command, as the name implies, print the last N number of data of the given input. By default 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.



expr and grep : The **expr command** is used to evaluate a given expression and display its standard output. Each separated expression is considered as an argument. These expressions could be integer and string expressions, including regular expressions. If expressions are not passed properly, it will prevent the execution of the command.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.

df,top,ps : Linux df command is used to display the **disk space used in the file system**. The 'df' stands for **"disk filesystem**." It defines the number of blocks used, the number of blocks available, and the directory where the file system is mounted.

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.The ps command is used to view currently running processes on the system. It helps us to determine which process is doing what in our system, how much memory it is using, how much CPU space it occupies, user ID, command name, etc .The **ps** command may display different results for different systems because it displays information about the currently running process of a system.