

## **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 **visual** 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**

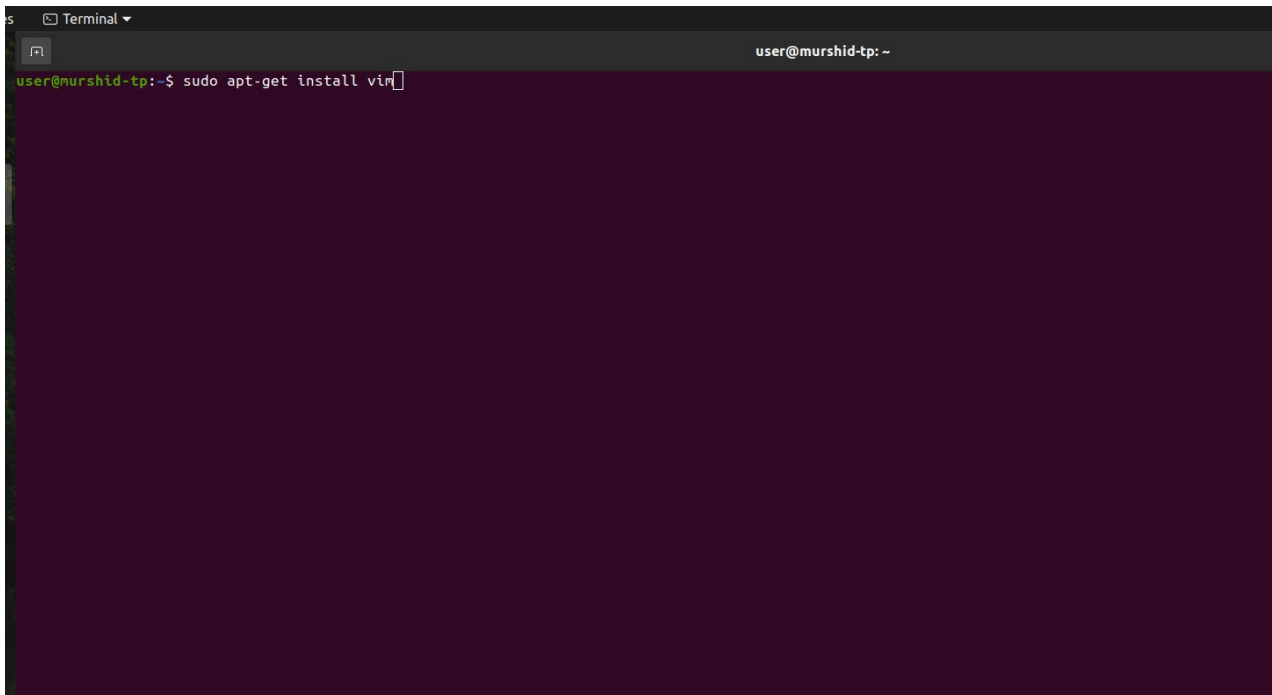
- Its memory footprint is very low
- It is command centric. You can perform complex text related task with few commands
- It is highly configurable and uses simple text file to store its configuration
- There are many plug-in available for Vim. Its functionality can be extended in great manner using 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

A screenshot of a terminal window with a dark purple background. The window title is "Terminal". The prompt is "user@murshid-tp: ~". The command "sudo apt-get install vim" has been entered, and the cursor is at the end of the line.

```
user@murshid-tp:~$ sudo apt-get install vim
```

```
user@murshid-tp:~$ vim --version
VIM - Vi IMproved 8.1 (2018 May 18, compiled Apr 15 2020 06:40:31)
Included patches: 1-2269
Modified by team+vim@tracker.debian.org
Compiled by team+vim@tracker.debian.org
Huge version without GUI.  Features included (+) or not (-):
+acl                -farsi              -mouse_sysmouse    -tag_any_white
+arabic             +file_in_path       -mouse_urxvt       +tcl
+autocmd             +find_in_path        +mouse_xterm       +termguicolors
+autocmdnr          +float              +multi_byte        +terminal
+autoservername     +folding             +multi_lang        +terminfo
+balloon_eval        -footer              +mzscheme          +termresponse
+balloon_eval_term  +gettext             +netbeans_intg     +textobjects
+browse              +gettext             +num64             +textprop
+builtin_terms       +hangul_input        +packages          +timers
+byte_offset         +iconv              +path_extra        +title
+channel             +insert_expand       -perl              -toolbar
+cinindent           +job                 +persistent_undo   +user_commands
+clientserver        +jumplist            +postscript        +various
+clipboard           +keymap              +printer           +vertical
+cmdline_compl       +lambda              +profile            +visual
+cmdline_hist        +langmap             +python            +visual
+cmdline_info        +libcall             +python3           +visualextra
+comments            +linebreak           +quickfix          +viminfo
+conceal             +listindent          +reltime           +vreplace
+cryptv             +listends            +rightleft         +wildignore
+cscope              +localmap            -ruby              +wildmenu
+cursordb            -lua                 +scrollbind        +windows
+cursortype          +menu                +signs             +writebackup
+dialog_con          +mksession           +smartindent       -X11
+diff               +modify_fname        +sound             -xfontset
+digraphs            +mouse               +spell            -xim
+dnd                 -mousethrow         +startuptime       -xpm
+ebcdic              +mouse_dec           +statusline        -xsnm
+emacs_tags          +mouse_gpm           -sun_workshop      +xterm_clipboard
+eval                +mouse_jsbterm       +syntax            -xterm_save
+ex_extra            +mouse_netterm       +tag_binary
+extra_search        +mouse_sgr           +tag_old_static

system vimrc file: "$VIM/vimrc"
user vimrc file: "$HOME/.vimrc"
2nd user vimrc file: "~/.vim/vimrc"
user exrc file: "$HOME/.exrc"
defaults file: "$VIMRUNTIME/defaults.vim"
fall-back for $VIM: "/usr/share/vim"
Compilation: gcc -c -I. -Iproto -DHAVE_CONFIG_H -Wdate-time -g -O2 -fdebug-prefix-map=/build/vim-iU6nZD/vim-8.1.2269=. -fstack-protector-strong -Wformat -Werror=format-security -D
REENTRANT -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=1
Linking: gcc -Wl,-Bsymbolic-functions -Wl,-z,relro -Wl,-z,now -Wl,--as-needed -o vim -lm -ltinfo -lnsl -lselinux -lcanberra -lacl -lattrib -lgpm -ldl -L/usr/lib/python3.8/config-3.8-x86_64-linux-gnu -lpthread -ldl -lutil -lm -lm
user@murshid-tp:~$
```

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

```
user@murshid-tp:~$ vim -v
VIM - Vi IMproved
      version 8.1.2269
  by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

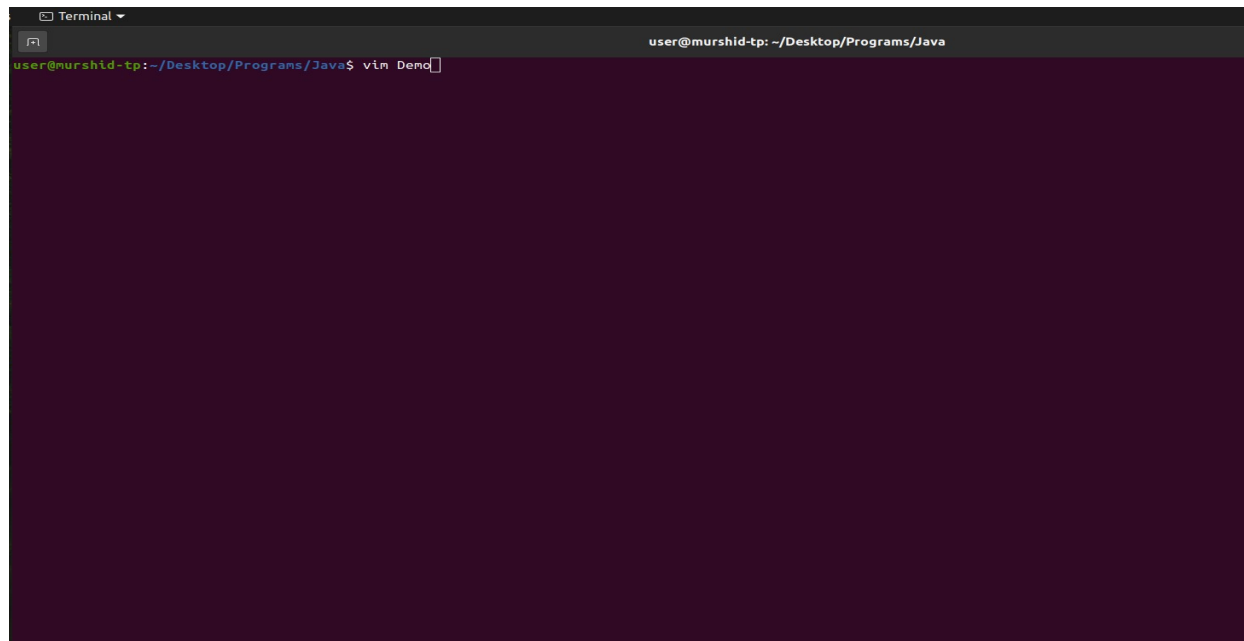
  Help poor children in Uganda!
type :help iccf<Enter>      for information
type :q<Enter>              to exit
type :help<Enter> or <F1>  for on-line help
type :help version8<Enter> for version info

0,0-1 All
```

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

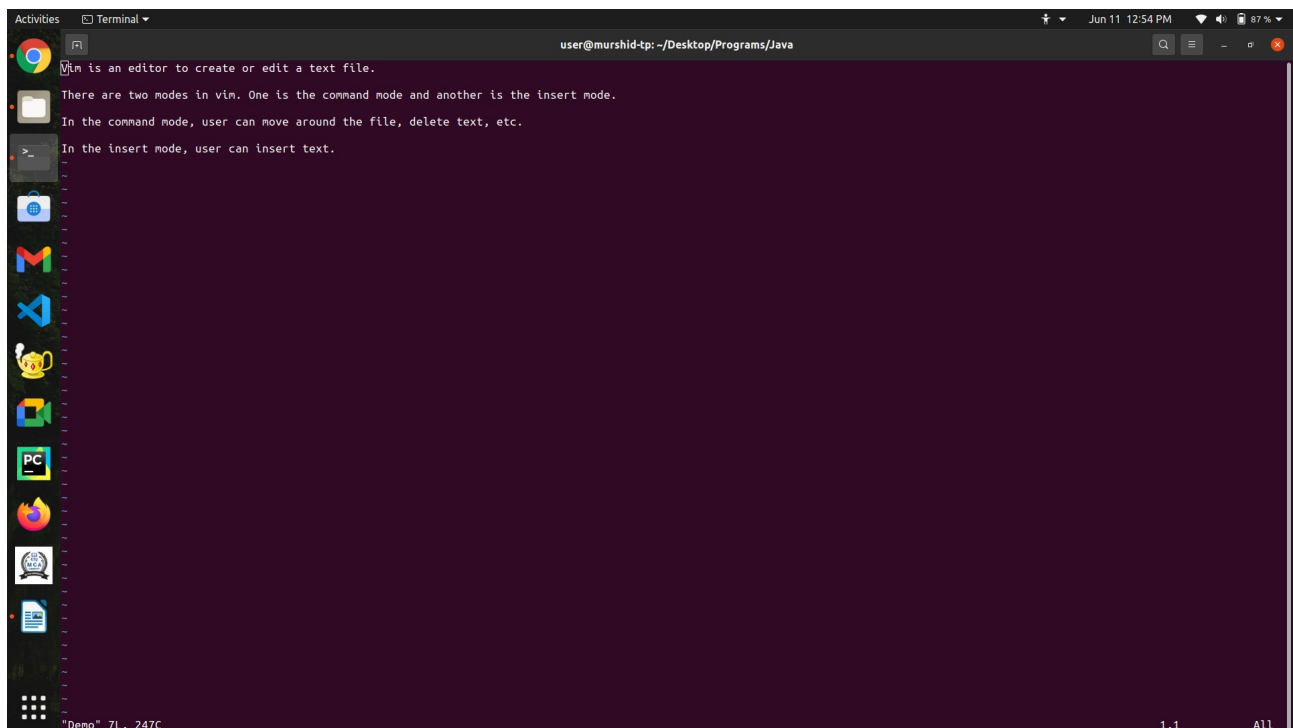
**vim [your file]**

Eg: vim Demo



```
user@murshid-tp: ~/Desktop/Programs/Java$ vim Demo
```

A terminal window titled "Terminal" with a dark background. The prompt is "user@murshid-tp: ~/Desktop/Programs/Java\$". The command "vim Demo" has been entered, and the cursor is at the end of the line.



## 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 up **l** - 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 one

**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**

```
Terminal
LS(1) User Commands LS(1) user@murshid-tp: ~
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
fied.
Mandatory arguments to long options are mandatory for short options
too.
-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
with -l, print the author of each file
-b, --escape
print C-style escapes for nongraphic characters
--block-size=SIZE
with -l, scale sizes by SIZE when printing them; e.g.,
'--block-size=M'; see SIZE format below
-B, --ignore-backups
do not list implied entries ending with ~
-c
with -lt: sort by, and show, ctime (time of last modification of
file status information); with -l: show ctime and sort by name;
otherwise: sort by ctime, newest first
-C
list entries by columns
--color[=WHEN]
colorize the output; WHEN can be 'always' (default if omitted),
'auto', or 'never'; more info below
-d, --directory
list directories themselves, not their contents
-D, --dired
Manual page ls(1) line 1 (press h for help or q to quit)
```

**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.

```
Terminal
user@murshid-tp: ~/Desktop/Programs
user@murshid-tp:~/Desktop/Programs$ ls
'DS LAB' Java PYTHON Record
user@murshid-tp:~/Desktop/Programs$ echo "welcome to Linux"
welcome to Linux
user@murshid-tp:~/Desktop/Programs$ echo "enter your name:"; read name;echo "Hello $name"
enter your name:
Murshid
Hello Murshid
user@murshid-tp:~/Desktop/Programs$
```


**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 searches for files and directories in a directory hierarchy based on a user given expression and can perform user specified action on each matched file.

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

A terminal window with a dark purple background. The title bar shows a window icon and the text 'user@murshid-tp: ~/Desktop'. The terminal content shows a series of commands and their outputs: 'ls' lists 'EXAM', 'My Projects', 'Notes', and 'Programs'; 'cd Programs' changes the directory; 'ls' lists 'DS LAB', 'Java', 'PYTHON', and 'Record'; 'mkdir DBMS' creates a new directory; 'ls' lists 'DBMS', 'DS LAB', 'Java', 'PYTHON', and 'Record'; 'pwd' shows the full path '/home/user/Desktop/Programs'; 'cd ..' returns to the parent directory; 'find . -name Demo' finds the file './Programs/Java/Demo'; and the prompt returns to the user's home directory.

```
user@murshid-tp:~/Desktop$ ls
EXAM 'My Projects' Notes Programs
user@murshid-tp:~/Desktop$ cd Programs
user@murshid-tp:~/Desktop/Programs$ ls
'DS LAB' Java PYTHON Record
user@murshid-tp:~/Desktop/Programs$ mkdir DBMS
user@murshid-tp:~/Desktop/Programs$ ls
DBMS 'DS LAB' Java PYTHON Record
user@murshid-tp:~/Desktop/Programs$ pwd
/home/user/Desktop/Programs
user@murshid-tp:~/Desktop/Programs$ cd ..
user@murshid-tp:~/Desktop$ find . -name Demo
./Programs/Java/Demo
user@murshid-tp:~/Desktop$
```

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

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ touch State
user@murshid-tp:~/Desktop$ cat >> State
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
^Z
[3]+  Stopped                  cat >> State
user@murshid-tp:~/Desktop$ cat State
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
user@murshid-tp:~/Desktop$ cat -n State
 1 Bihar
 2 Chhattisgarh
 3 Goa
 4 Gujarat
 5 Haryana
 6 Himachal Pradesh
 7 Jammu and Kashmir
 8 Jharkhand
 9 Karnataka
10 Kerala
user@murshid-tp:~/Desktop$ cat > Country
India
China
^Z
[4]+  Stopped                  cat > Country
user@murshid-tp:~/Desktop$ cat Country
India
China
user@murshid-tp:~/Desktop$
```

**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

```
user@murshid-tp: ~/Desktop/EXAM
user@murshid-tp:~/Desktop$ ls
capital Country EXAM 'My Projects' Programs State
user@murshid-tp:~/Desktop$ date
Thursday 12 August 2021 08:37:32 PM IST
user@murshid-tp:~/Desktop$ cal
      August 2021
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

user@murshid-tp:~/Desktop$ locate fee.pdf
/home/user/Desktop/EXAM/fee.pdf
user@murshid-tp:~/Desktop$ cd EXAM
user@murshid-tp:~/Desktop/EXAM$ ls
'44_Murshid TP ds assignment 1.pdf'  fee.pdf  RECORDS  'RegisteredStudentBillReport (2).pdf'  'second fee.pdf'
'44_Murshid TP DS assignment2.pdf'  paymenthistoryredirecturl.pdf  'RegisteredStudentBillReport (1).pdf'  S2
user@murshid-tp:~/Desktop/EXAM$ ls -l
total 1128
-rw-rw-r-- 1 user user 158361 Apr  1 06:25 '44_Murshid TP ds assignment 1.pdf'
-rw-rw-r-- 1 user user 677465 Apr  1 00:15 '44_Murshid TP DS assignment2.pdf'
-rw-rw-r-- 1 user user 93537 May 22 11:57 fee.pdf
-rw-rw-r-- 1 user user 94621 Mar 27 07:42 paymenthistoryredirecturl.pdf
drwxrwxr-x 2 user user 4096 Jun 26 19:14 RECORDS
-rw-rw-r-- 1 user user 7483 Mar 26 16:57 'RegisteredStudentBillReport (1).pdf'
-rw-rw-r-- 1 user user 7479 Mar 27 07:47 'RegisteredStudentBillReport (2).pdf'
drwxrwxr-x 2 user user 4096 Aug  3 08:48 S2
-rw-rw-r-- 1 user user 94293 Jun  1 15:56 'second fee.pdf'
user@murshid-tp:~/Desktop/EXAM$
```

**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.

```
user@murshid-tp: ~  
user@murshid-tp:~$ whatis ls  
ls (1) - list directory contents  
user@murshid-tp:~$ whatis read  
read (2) - read from a file descriptor  
user@murshid-tp:~$ whatis echo  
echo (1) - display a line of text  
user@murshid-tp:~$ whereis ls  
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz  
user@murshid-tp:~$ whereis read  
read: /usr/share/man/man2/read.2.gz  
user@murshid-tp:~$ whereis echo  
echo: /usr/bin/echo /usr/share/man/man1/echo.1.gz  
user@murshid-tp:~$
```

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

```
user@murshid-tp: ~  
1456 cat Country  
1457 clear  
1458 ls  
1459 rename capital Capital of state  
1460 sudo apt install rename  
1461 sudo apt install rename  
1462 clear  
1463 id  
1464 clear  
1465 sort capital  
1466 cat capital  
1467 clear  
1468 find . -name "*.pdf"  
1469 find fee2.pdf  
1470 locate fee2.pdf  
1471 sudo apt install nlocate  
1472 locate fee2.pdf  
1473 exit  
1474 cal  
1475 time  
1476 ls  
1477 rename State s  
1478 ping gmail.com  
1479 clear  
1480 ls  
1481 date  
1482 cal  
1483 locate fee.pdf  
1484 cd EXAM  
1485 ls  
1486 ls -l  
1487 ls  
1488 cd ..  
1489 ls  
1490 echo $SHELL  
1491 history  
1492 whatis ls  
1493 whatis cd  
1494 whatis read  
1495 whatis -d ls  
1496 clear  
1497 whatis ls  
1498 whatis read  
1499 whatis echo  
1500 whereis ls  
1501 whereis read  
1502 whereis echo  
1503 clear  
1504 history  
user@murshid-tp:~$
```

**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.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ ls
Country  EXAM  'My Projects'  Notes  Programs  State
user@murshid-tp:~/Desktop$ rm -i Country
rm: remove regular file 'Country'? y
user@murshid-tp:~/Desktop$ ls
EXAM  'My Projects'  Notes  Programs  State
user@murshid-tp:~/Desktop$ alias s=ls
user@murshid-tp:~/Desktop$ s
EXAM  'My Projects'  Notes  Programs  State
user@murshid-tp:~/Desktop$ ls
EXAM  'My Projects'  Notes  Programs  State
user@murshid-tp:~/Desktop$
```

**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 alphabetically.

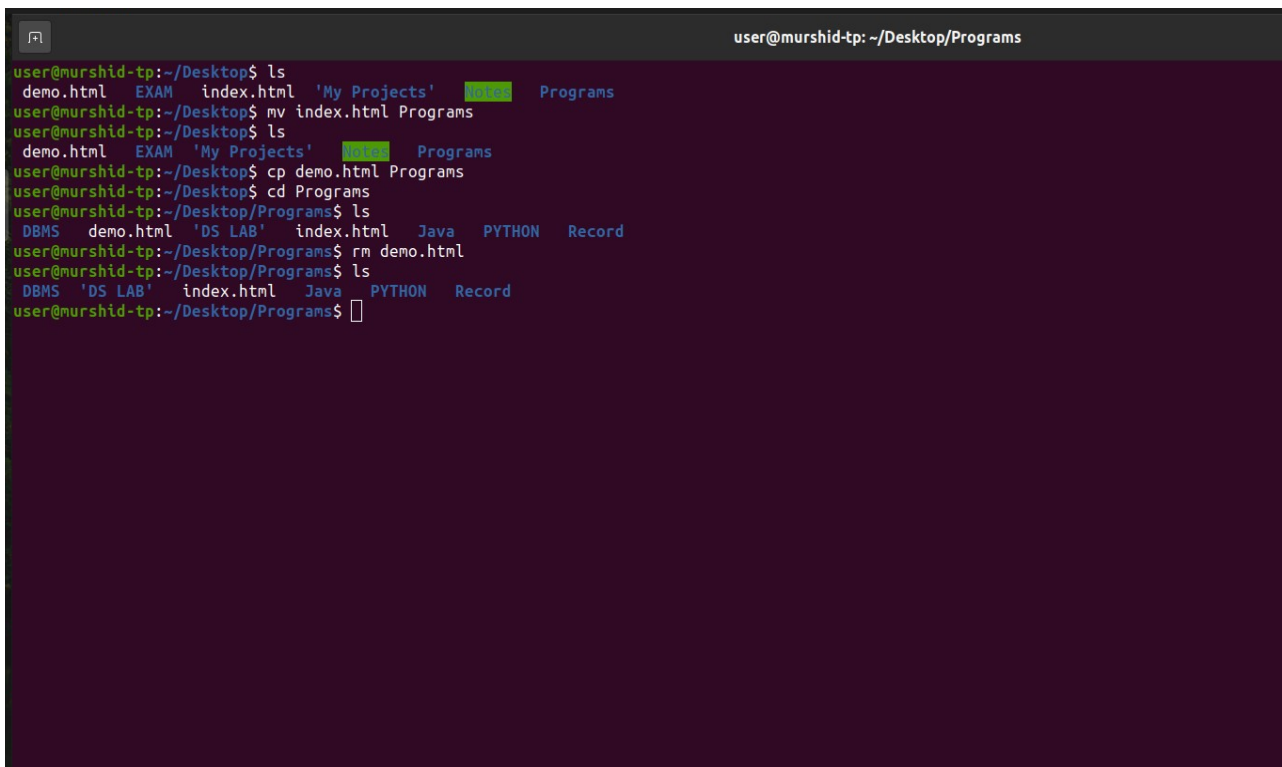
```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ ls
EXAM  'My Projects'  Notes  Programs  State
user@murshid-tp:~/Desktop$ cat State
kerala
tamilnadu
karnataka
goa
telengana
rajasthan
andrapradesh
user@murshid-tp:~/Desktop$ sort State
andrapradesh
goa
karnataka
kerala
rajasthan
tamilnadu
telengana
user@murshid-tp:~/Desktop$
```



**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.

A terminal window titled 'user@murshid-tp: ~/Desktop/Programs' showing a series of commands and their outputs. The user starts in the Desktop directory, lists files, moves 'index.html' to the 'Programs' directory, copies 'demo.html' to 'Programs', changes to the 'Programs' directory, lists files, and finally removes 'demo.html'.

```
user@murshid-tp:~/Desktop$ ls
demo.html  EXAM  index.html  'My Projects'  Notice  Programs
user@murshid-tp:~/Desktop$ mv index.html Programs
user@murshid-tp:~/Desktop$ ls
demo.html  EXAM  'My Projects'  Notice  Programs
user@murshid-tp:~/Desktop$ cp demo.html Programs
user@murshid-tp:~/Desktop$ cd Programs
user@murshid-tp:~/Desktop/Programs$ ls
DBMS  demo.html  'DS LAB'  index.html  Java  PYTHON  Record
user@murshid-tp:~/Desktop/Programs$ rm demo.html
user@murshid-tp:~/Desktop/Programs$ ls
DBMS  'DS LAB'  index.html  Java  PYTHON  Record
user@murshid-tp:~/Desktop/Programs$
```

**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.



```
user@murshid-tp:~/Desktop$ ls
demo.html  EXAM  'My Projects'  Programs
user@murshid-tp:~/Desktop$ cat demo.html
Hello
Welcome to Python
user@murshid-tp:~/Desktop$ wc -l demo.html
2 demo.html
user@murshid-tp:~/Desktop$ wc -w demo.html
4 demo.html
user@murshid-tp:~/Desktop$ wc -m demo.html
25 demo.html
user@murshid-tp:~/Desktop$ wc -c demo.html
25 demo.html
user@murshid-tp:~/Desktop$
```

**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.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ cat demo
harry,25,16200
gill,46,17500
bill,45,20000
john,43,100000
barry,27,42000
paul,18,26500
user@murshid-tp:~/Desktop$ cut -d, -f 1,3 demo
harry,16200
gill,17500
bill,20000
john,100000
barry,42000
paul,26500
user@murshid-tp:~/Desktop$ cat state
kerala
tamilnadu
karnataka
telengana
user@murshid-tp:~/Desktop$ cat capital
Trivandrum
Chennai
Banglore
Hyderabad
user@murshid-tp:~/Desktop$ paste -d: state capital
kerala:Trivandrum
tamilnadu:Chennai
karnataka:Banglore
telengana:Hyderabad
user@murshid-tp:~/Desktop$
```

**head and tail** : 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.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ cat state
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
user@murshid-tp:~/Desktop$ head state
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
user@murshid-tp:~/Desktop$ tail state
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
user@murshid-tp:~/Desktop$
```

**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.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ expr --version
expr (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Mike Parker, James Youngman, and Paul Eggert.
user@murshid-tp:~/Desktop$ expr 5 + 2
7
user@murshid-tp:~/Desktop$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1940816         0   1940816   0% /dev
tmpfs           394088         1876   392212    1% /run
/dev/sda8       92098232 25630400 61746476  30% /
tmpfs           1970428 169748   1800680    9% /dev/shm
tmpfs           5120          4     5116    1% /run/lock
tmpfs           1970428         0   1970428   0% /sys/fs/cgroup
/dev/loop0       56832    56832         0 100% /snap/core18/2066
/dev/loop1       56832    56832         0 100% /snap/core18/2074
/dev/loop4       168832   168832         0 100% /snap/gnome-3-28-1804/161
/dev/loop3       66432    66432         0 100% /snap/gtk-common-themes/1514
/dev/loop2       166784   166784         0 100% /snap/gnome-3-28-1804/145
/dev/loop6       33152    33152         0 100% /snap/snapd/12398
/dev/loop7       33152    33152         0 100% /snap/snapd/12704
/dev/loop5       66688    66688         0 100% /snap/gtk-common-themes/1515
/dev/loop8       310400   310400         0 100% /snap/wine-platform-5-stable/16
/dev/sda7        456744   213148   209408   51% /boot
/dev/sda1        262144   58440   203704   23% /boot/efi
tmpfs            394084         60   394024    1% /run/user/1000
user@murshid-tp:~/Desktop$ ps
  PID TTY          TIME CMD
  87522 pts/1    00:00:00 bash
  91873 pts/1    00:00:00 ps
user@murshid-tp:~/Desktop$
```