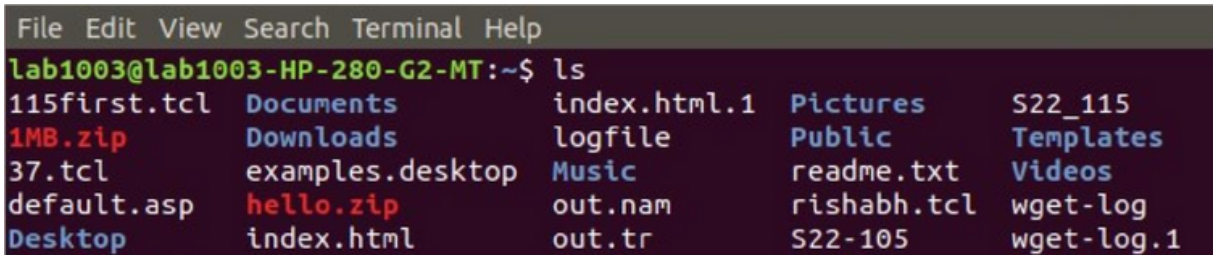
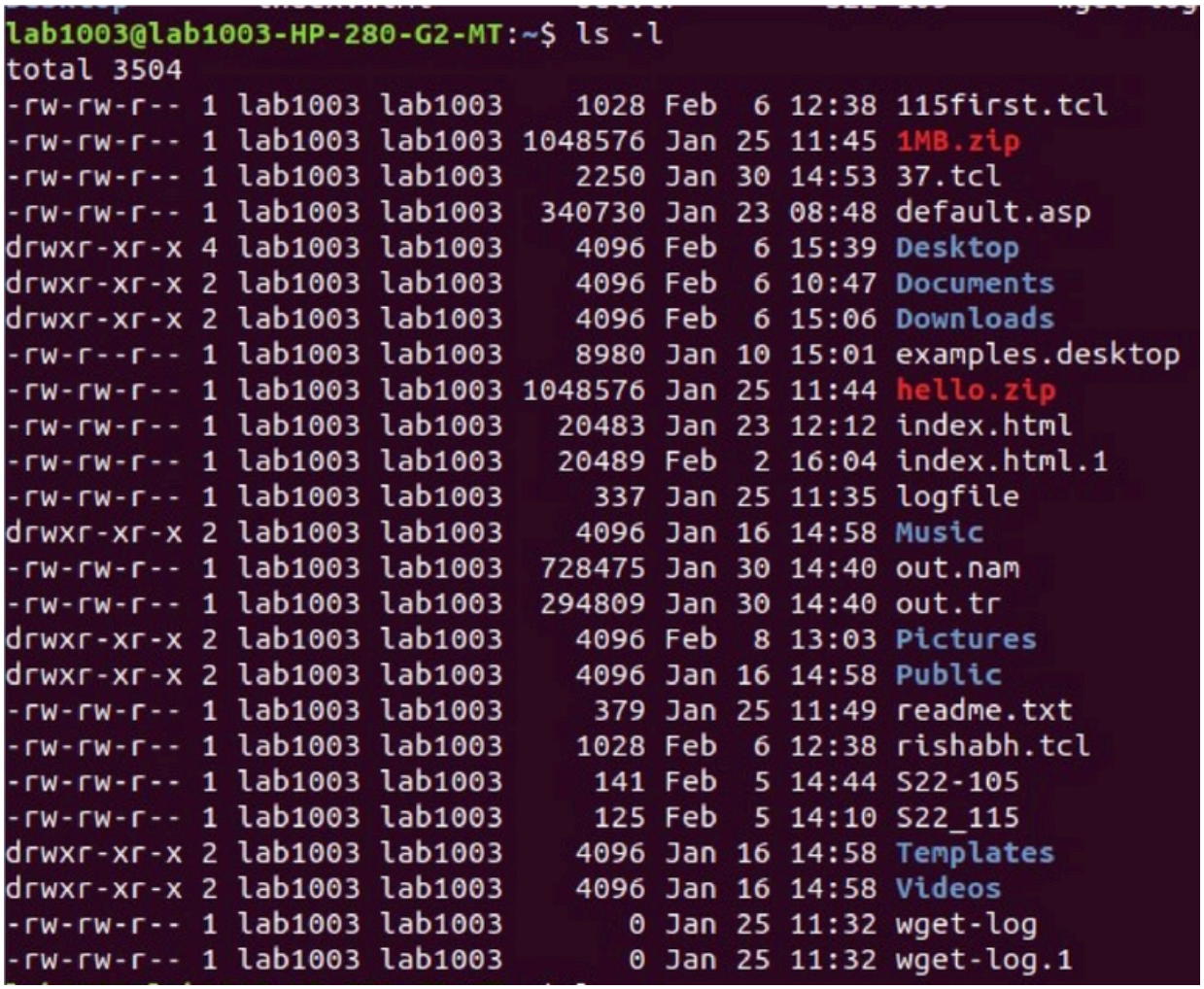
# Assignment 03 -
# To Study File System Management Commands

**1) ls command-ls is a Linux shell command that lists directory contents of files and directories. It provides valuable information about files, directories, and their attributes.**

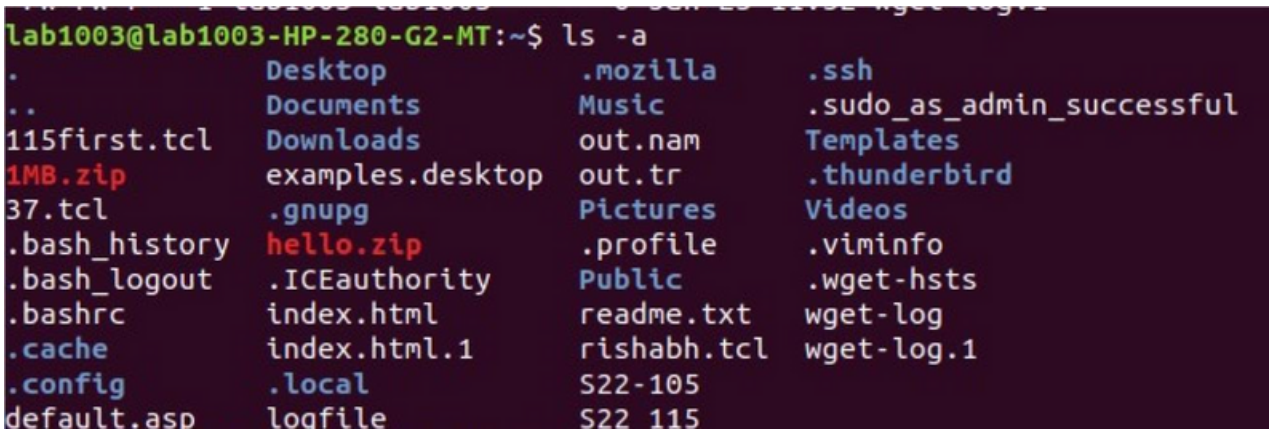



**2) ls -l command - known as a long format**

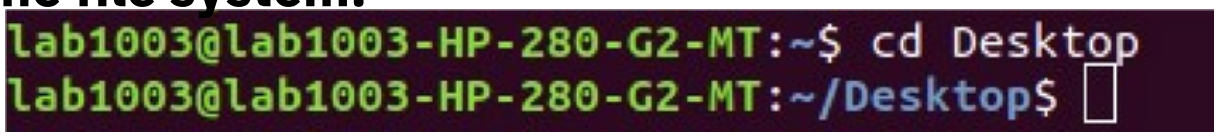



**that displays detailed information about files and directories.**

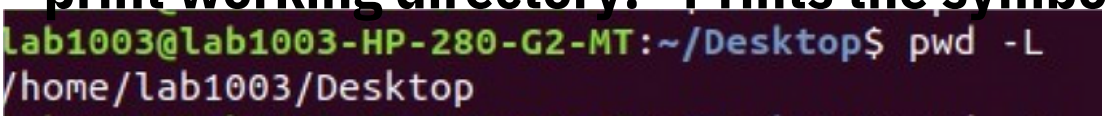
**3) ls -a command - Represent all files Include hidden files and directories in the listing.**

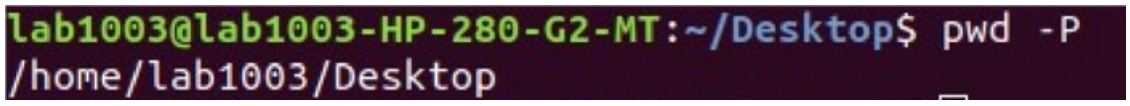



**4) cd command-The 'cd' command allows users to change their current working directory within the file system.**




**5) pwd -L command- The 'pwd,' which stands for "print working directory." Prints the symbolic path.**




**6) pwd -P command -The 'pwd,' which stands for "print working directory." Prints the actual path.**

**7) cat command-The cat command in Linux is more than just a simple tool; it's a versatile companion for various file-related operations, allowing users to view, concatenate, create, copy, merge, and manipulate file contents.**

```
lab1003@lab1003-HP-280-G2-MT:~/Desktop$ cat 1.txt
hello world
```

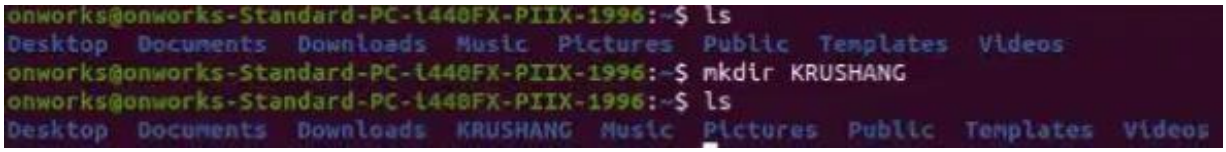
**8) cat -n command -Adding the -n option to cat introduces line numbers, making it convenient to identify and reference specific lines within the file.**

```
lab1003@lab1003-HP-280-G2-MT:~/Desktop$ cat -n 1.txt
     1  hello world
```

**9) cat > file_name command - If you want to create a new file or overwrite an existing file with new content, you can use ‘cat’ with the output redirection (`>`):**

```
lab1003@lab1003-HP-280-G2-MT:~/Desktop$ cat > 2.txt
hello world
this file is created using cat command
```

10)mkdir command- This command can create




multiple directories at once as well as set the permissions for the directories. ‘mkdir’ stands for “make directory,”.

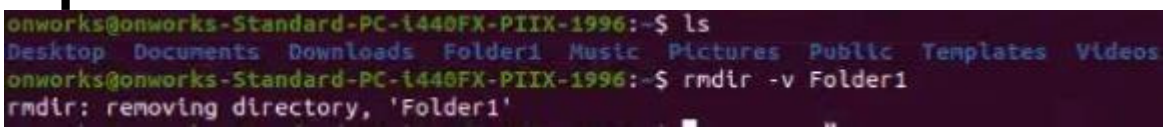
11) mkdir –v command - Enables verbose mode, displaying a message for every directory created. When used with the [directories] argument, it shows the names of the directories being created.


12) mkdir –p command - A flag that allows the creation of parent directories as necessary. If the specified directories already exist, no error is reported. Useful for creating a directory hierarchywithout errors.


13) rmdir command - The rmdir command is useful when you want to remove the empty directories from the filesystem in Linux.

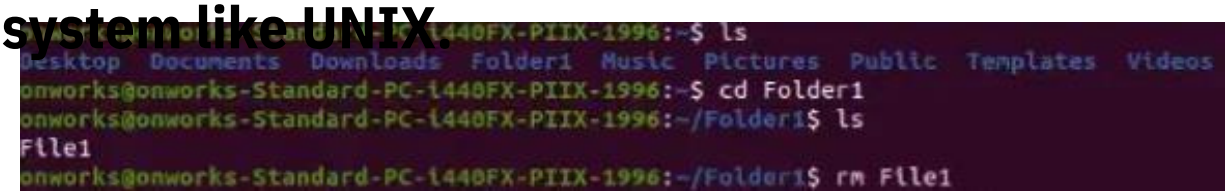

14) rmdir –v command - If you want the terminal to display the message after

removing the directory, you can use the -v option with the rmdir command:




15) rmdir –p command - You can use the -p option with the rmdir command to delete a directory, including all the subdirectories:


16) rm command - 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.



17)rm –v command - If you want the terminal


to display the message after removing the file, youcan use the -v option with the rm command:

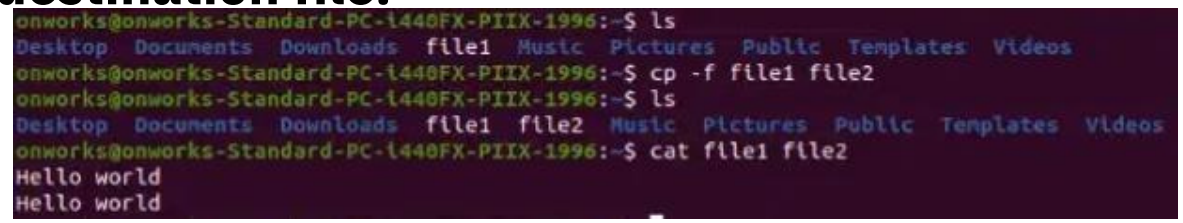
18) rm –i command - If the file is write-protected, rm will ask you to validate its deletion; otherwise, it will delete it without prompting. Using the “-i” flag to force rm to prompt for confirmation before deleting a file.


19) cp command - This command creates a copy of the `source_file` at the specified `destination`. If the destination is a directory, the file is copied into that directory.


20) cp –i command - i stands for Interactive copying. With this option the system first warns the user before overwriting the destination file. cp prompts for a response, if you press y then it

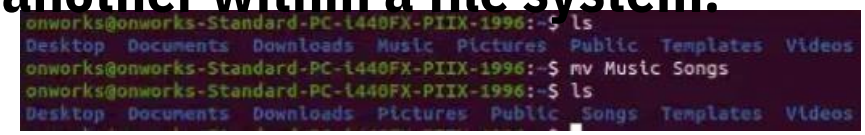
overwrites the file and with any other option leaves it uncopied.


21) cp –f command - -f(force): If the system is unable to open destination file for writing operation because the user doesn't have writing permission for this file then by using -f option with cp command, destination file is deleted first and then copying of content is done from source to
destination file.

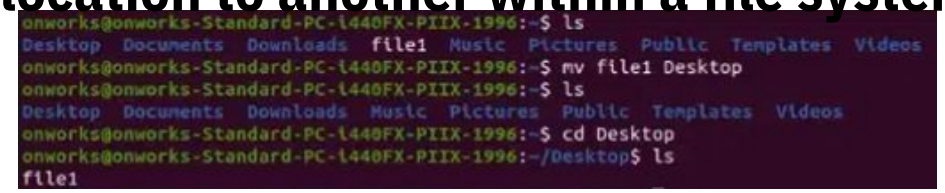


**22) mv command (to rename a file)- mv command in linux is used to rename file directories and move files from one location to another within a file system.**


**23) mv command (to rename a directory)- mv command in linux is used to rename file directories and move files from one location to another within a file system.**




**24) mv command (to move a file to another directory) - mv command in linux is used to rename file directories and move files from one location to another within a file system.**




**25) chmod command - In Unix operating systems, the chmod command is used to change the access mode of a file. The name is an abbreviation of change mode. Which states that every fileand directory has a set of permissions that control the permissions like who can read, write or execute the file. In this the permissions have three categories: read, write, and execute simultaneously represented by `r`, `w` and `x`. These letters**

combine together to form a specific permission for a group of users.The `chmod` command is used to modify this permission so that itcan grant or restrict access to directories and files.

Symbolic mode is the most common method used for specifying fir permissions. In this we must make a combination of letters and operators to set or tell what to do with permissions.

The following operators can be used with the symbolic mode:

| Operators | Definition |
|---|---|
| `+ | Add permissions |
| ` | Remove permissions |
| `-`<br>`=`<br>` | Set the permissions to the specified values |

| Letters | Definition |
| --- | --- |
| `r` | Read permission |
| `w | Write permission |
| ` | Execute permission |

| Reference | Class |
| --- | --- |
| u | Owner |
| g | Group |
| o | Others |
| a | All (owner,groups,others) |


**Examples of Using the Symbolic mode:**

- **Read, write and execute permissions to the file owner:**

**chmod u+rwx [file_name]**

- **Remove write permission for the group and others:**

**chmod go-w [file_name]**

- **Read and write for Owner, and Read-only for**

**the group and other:**
**chmod u+rw,go+r [file_name]**

# Assignment 3

**11,)wc stands for word count. As the name implies, it is mainly used for counting purpose.**

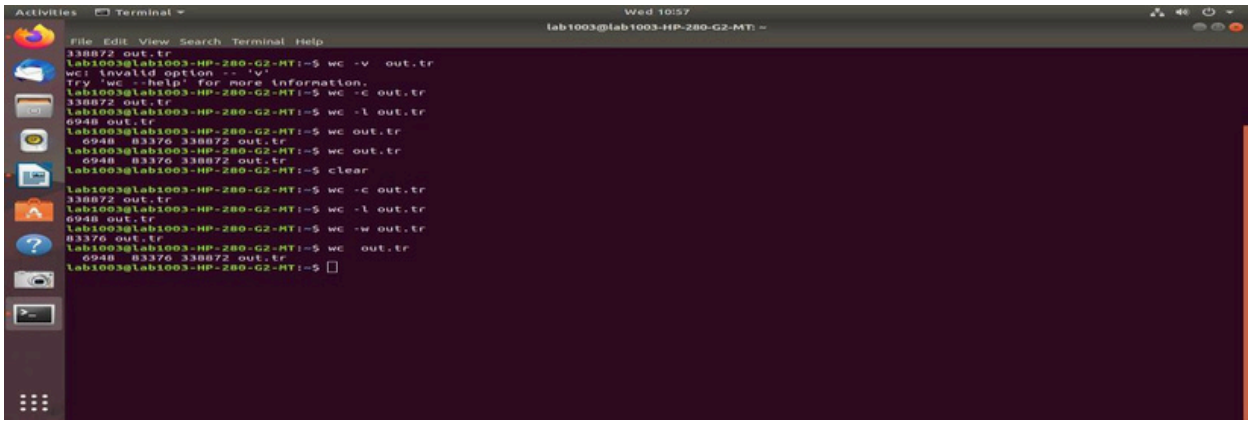
- **It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.**
- **By default it displays four-columnar output.**
  - **First column shows number of lines present in a file specified, second column shows number of**

words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.

. -c: This option displays count of bytes present in a file. With this option it display two-columnar output, 1st column shows number of bytes present in a file and 2nd is the file name.

-w: This option prints the number of words present in a file. With this option wc command displays two-columnar output, 1st column shows number of words present in a file and 2nd is the file name.

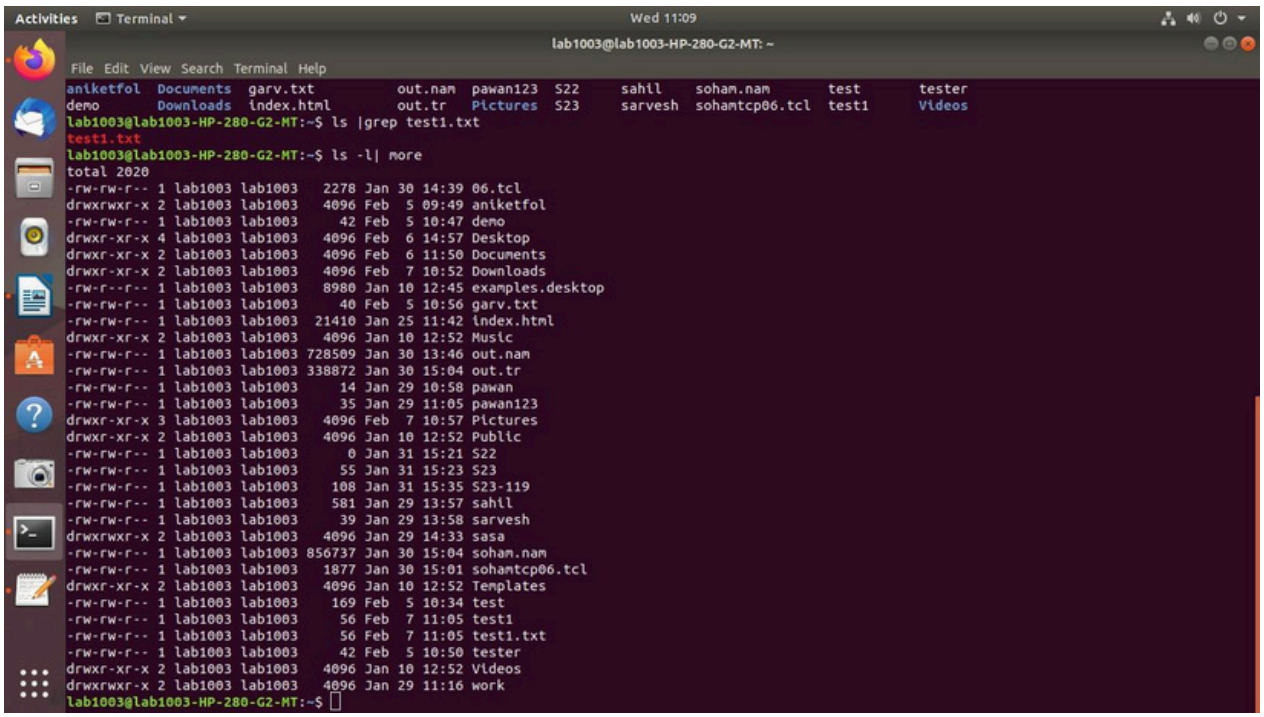
-l: This option prints the number of lines present in a file. With this option wc command displays two-columnar output, 1st column shows number of lines present in a file and 2nd itself represent the file name.



12.) **piping**

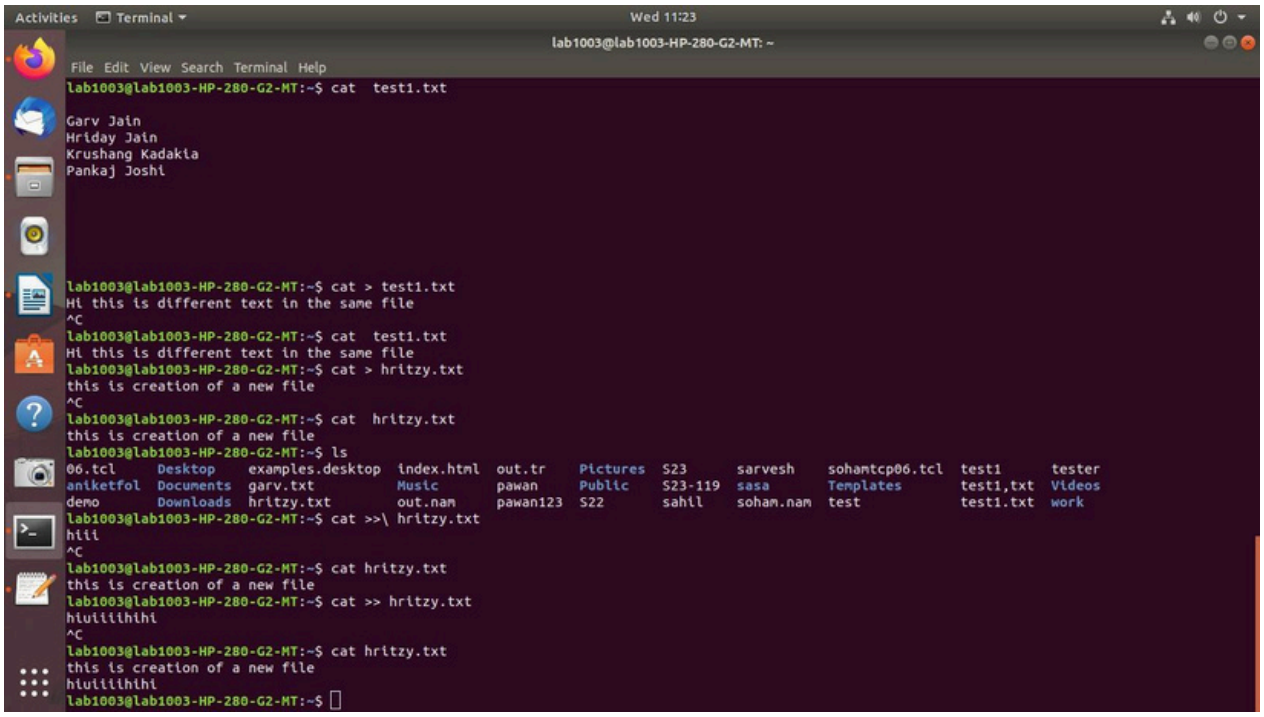
**The pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command, and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes.**

**Variation of piping are shown below**



13.) **redirection:**

**Each stream uses redirection commands. Single bracket '>' or double bracket '>>' can be used to redirect standard output. If the target file doesn't exist, a new file with the same name will be created.**




**Redirection is denoted by > or >>**

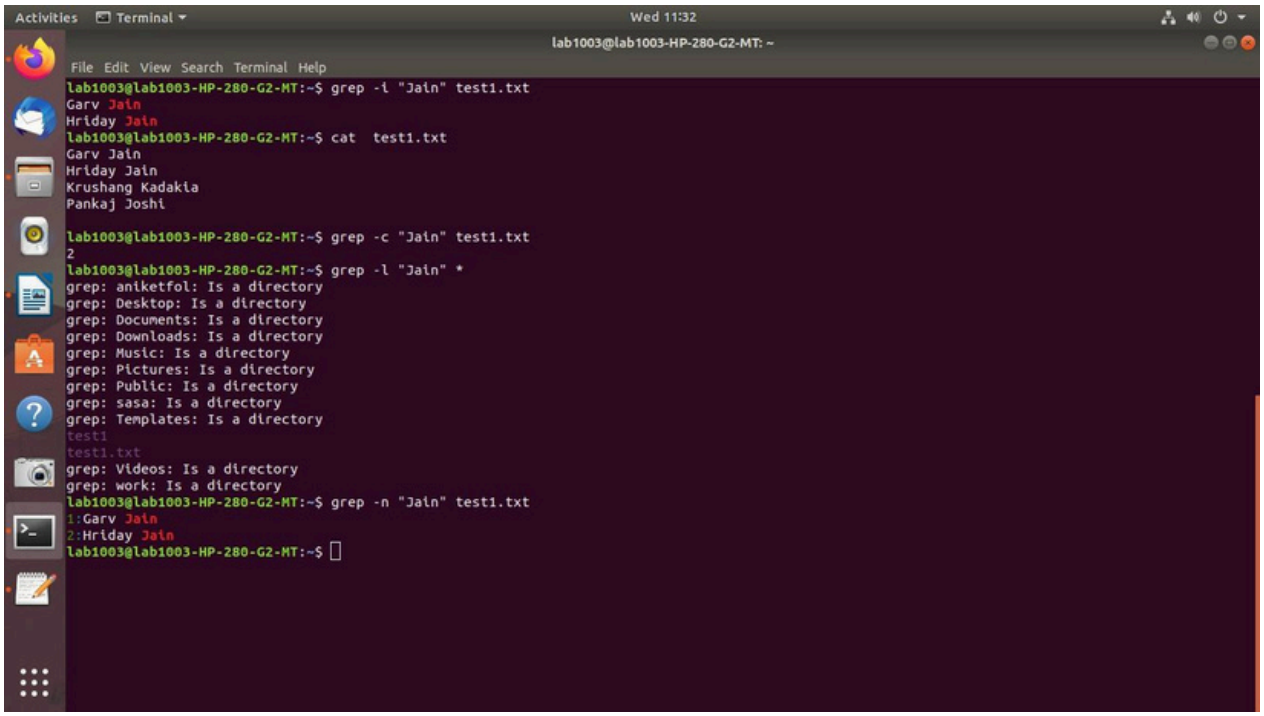
**> - is used to create and edit a new file**

**>>- is used to add to the existing file**

14.) **grep-Grep, short for “global regular expression print”, is a command used for searching and matching text patterns in files contained in the regular expressions.**

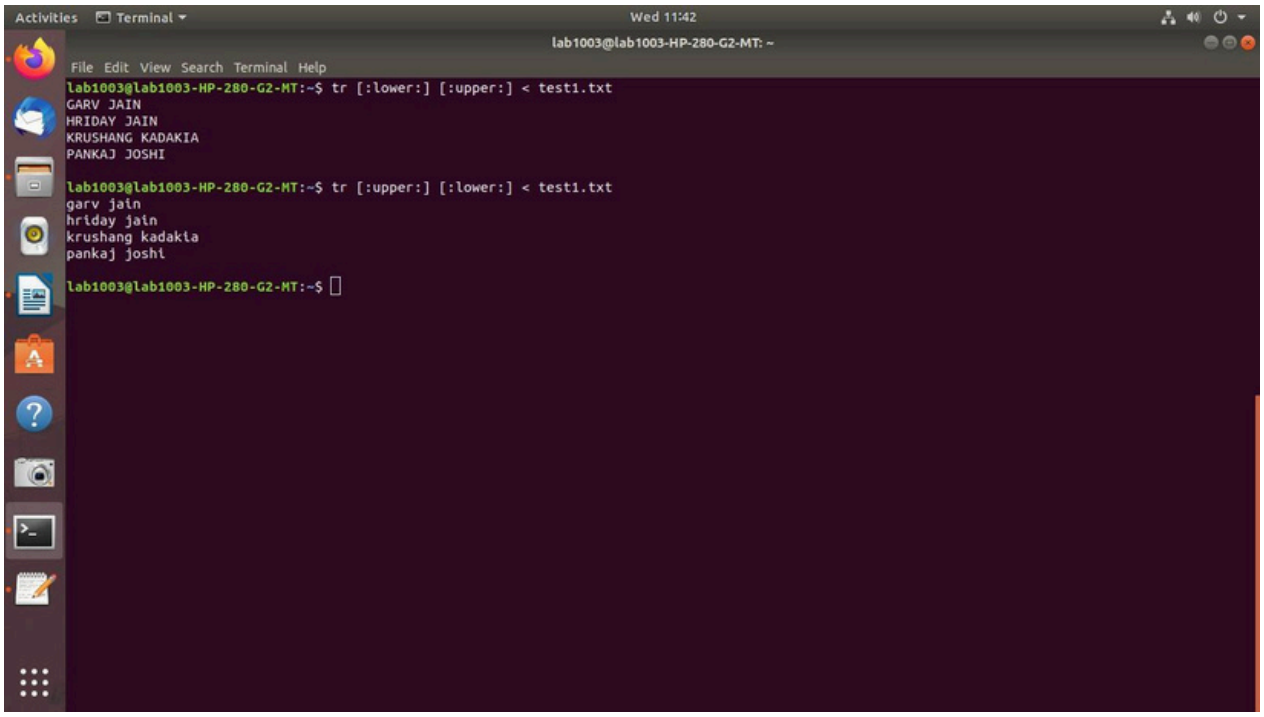
**-c This prints only a count of**

| | |
|---|---|
| | the lines that match a pattern |
| -h | Display the matched lines, but do not display the filenames. |
| –i | Ignores, case for matching |
| -l | Displays list of a filenames only. |
| -n | Display the matched lines and their line numbers. |




15.) **tr**

**The tr command is a UNIX command-line utility for translating or deleting characters. It supports a range of transformations including uppercase to lowercase, squeezing repeating characters, deleting specific characters, and basic find and replace. It can be used with UNIX pipes to support more complex translation. tr stands for translate**




**-d == to delete specified characters. To delete specific characters use the -d option. This option deletes characters in the first set specified.**


**using -c option You can complement the SET1 using -c option.**


Activities
Terminal
Wed 11:49
lab1003@lab1003-HP-280-G2-MT: ~
File Edit View Search Terminal Help
lab1003@lab1003-HP-280-G2-MT:~$ cat test1.txt
Garv Jain ID-55
Hriday Jain ID-56
Krushang Kadakia ID-58
Pankaj Joshi ID-57
lab1003@lab1003-HP-280-G2-MT:~$ tr -cd [:digit:] < test1.txt
55565857lab1003@lab1003-HP-280-G2-MT:~$

16.) **echo**

**The echo command in Linux is a built-in command that allows users to display lines of text or strings that are passed as arguments. It is commonly used in shell scripts and batch files to output status text to the screen or a file.**

**echo *: this command will print all files/folders, similar to ls command.**

**-n: this option is used to omit echoing trailing newline.**

**It can also be used for redirecting amd adding a new line to a existing file**


17.) **sort**

**-o Specifies an output file for the sorted data.**

**Functionally equivalent to redirecting output to a file.**

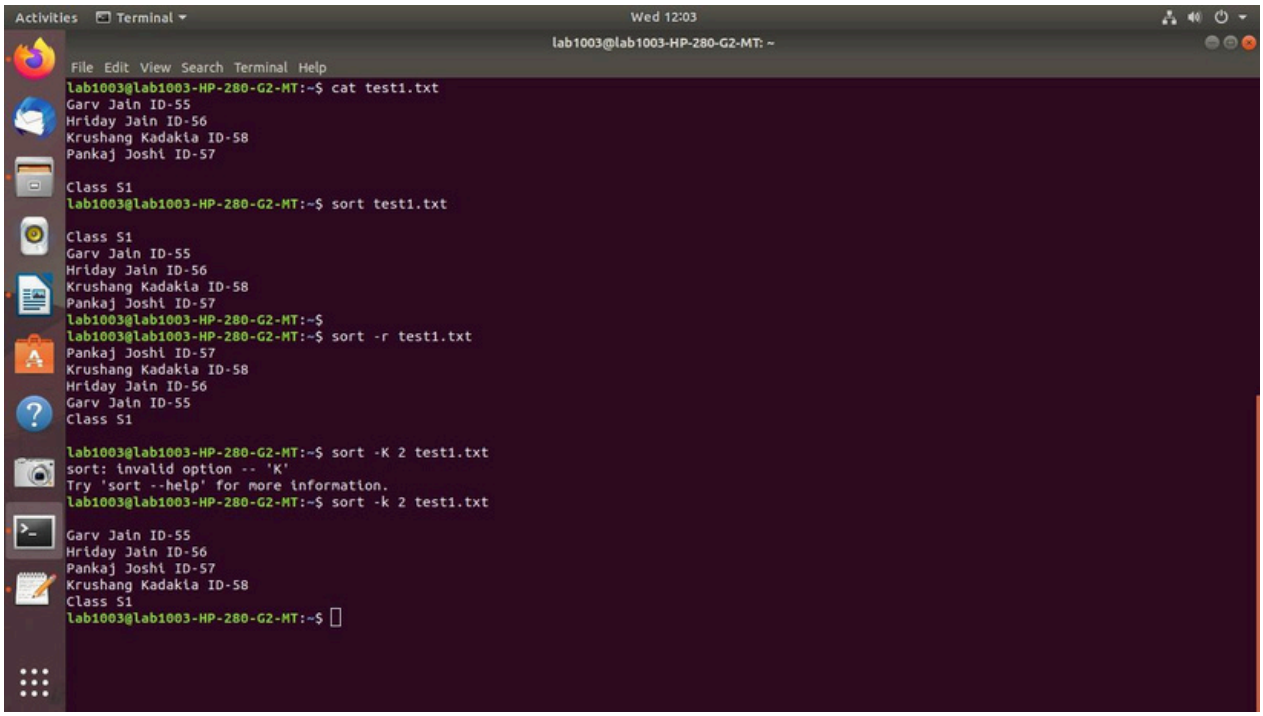
**-r Sorts data in reverse order (descending).**

**-n Sorts a file numerically (interprets data as numbers).**

**-nr Sorts a file with numeric data in reverse order.**

**Combines -n and -r options.**

**-k Sorts a table based on a specific column numbe**

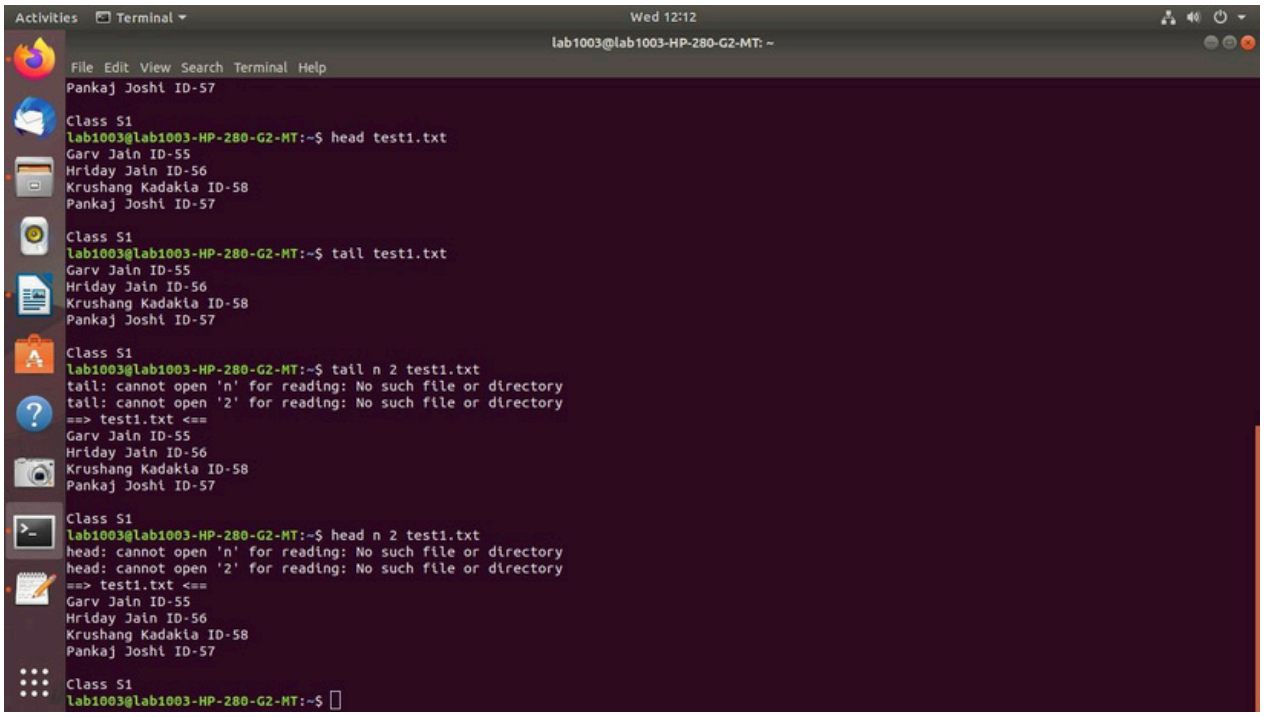



## 18.) **head**

**It is the complementary of Tail command. 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.**

nd Pratical Examples of Tail Command in

| Short Options | Long Options |
|---|---|
| -n | --lines |
| -c | --bytes |
| -q | --quiet |
| -v | --verbose |
| -f | --follow |



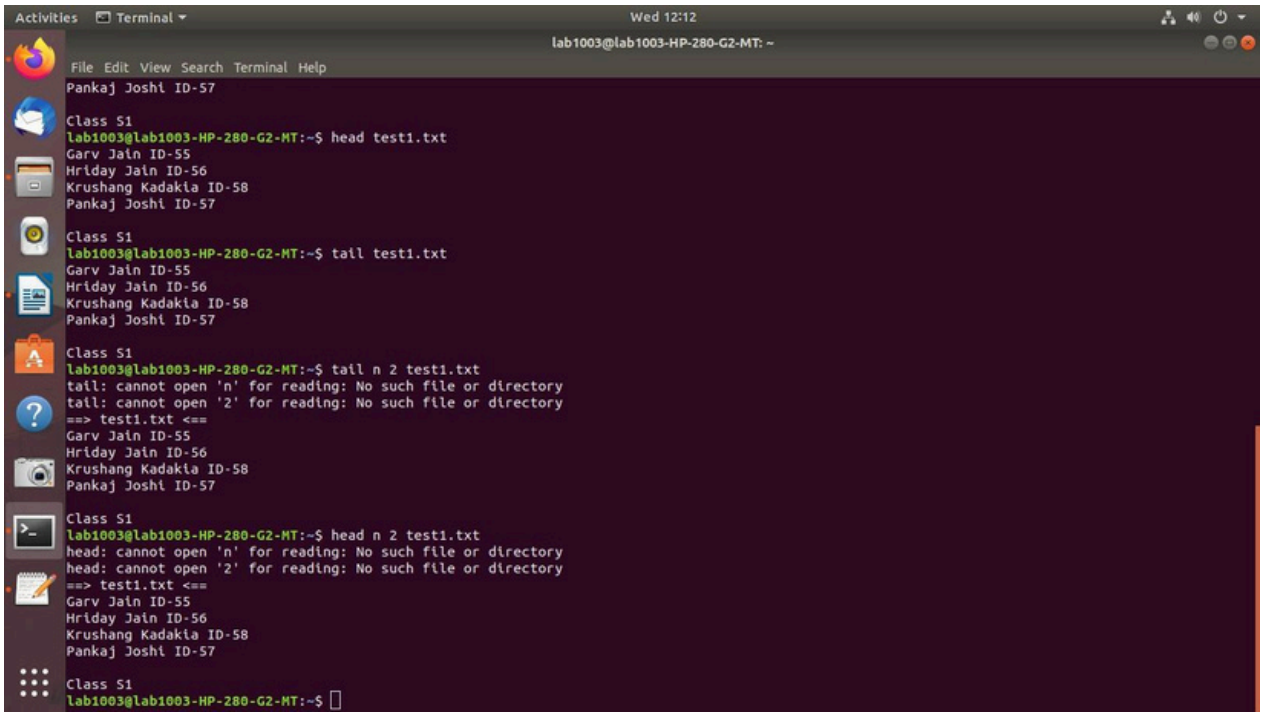

19.) **tail**

**It is the complementary of head command. The tail command, as the name implies, prints 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 preceded by its file name.**

nd Pratical Examples of Tail Command in

| Short Options | Long Options |
| --- | --- |
| -n | --lines |
| -c | --bytes |
| -q | --quiet |
| -v | --verbose |
| -f | --follow |

Option in Tail Command in Linux




## 21.)Less

**Less command is a Unix utility that can be used to read the contents of a text file one page (one screen) at a time. It has faster access because if a file is large, it doesn't access the complete file, but accesses it page by page.**

**For example, if it's a large file and you are reading it using any text editor, then the complete file will be loaded to the main memory. The less command doesn't load the entire file but loads it part by part which makes it faster.**

**Syntax of `less` command in Unix**

**The basic syntax of the less**

**command is as follows: less**

**[options] filename**

**Options Description**

**-E Automatically exit when reaching the end of the file.**

**-f Force non-regular files to be opened.**

**-F Exit if the entire file can be displayed on the first screen.**

-g Highlight the string that was found by the last search command.

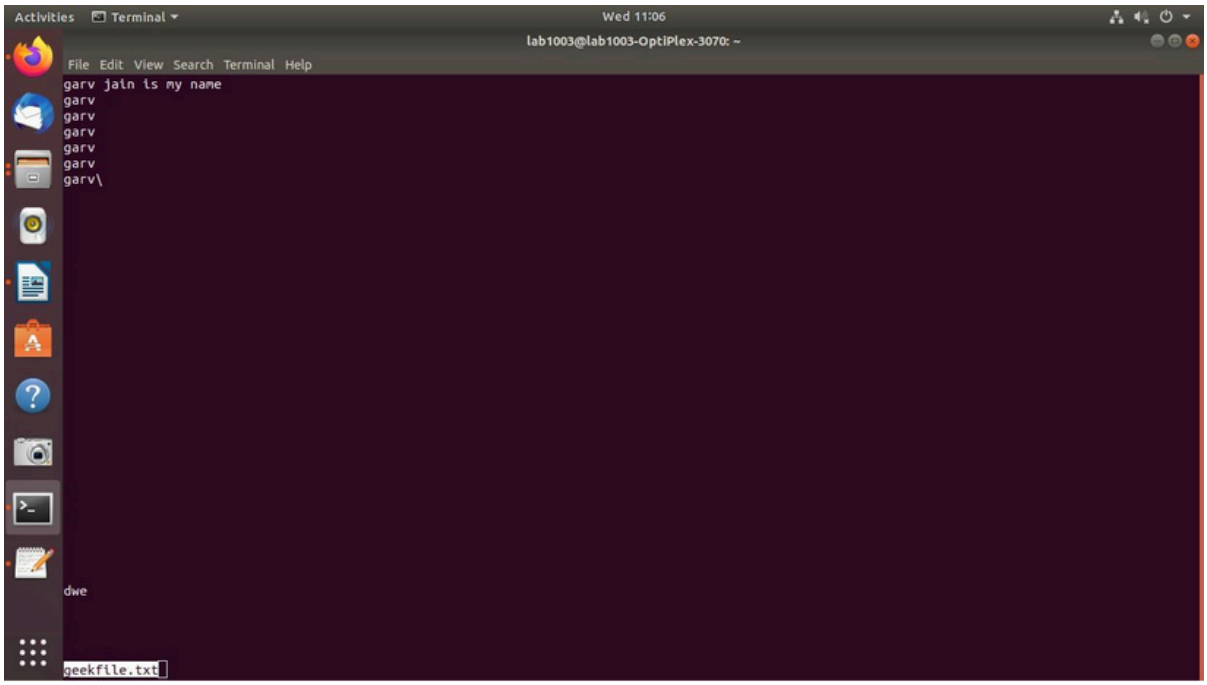
-G Suppress highlighting of search matches.

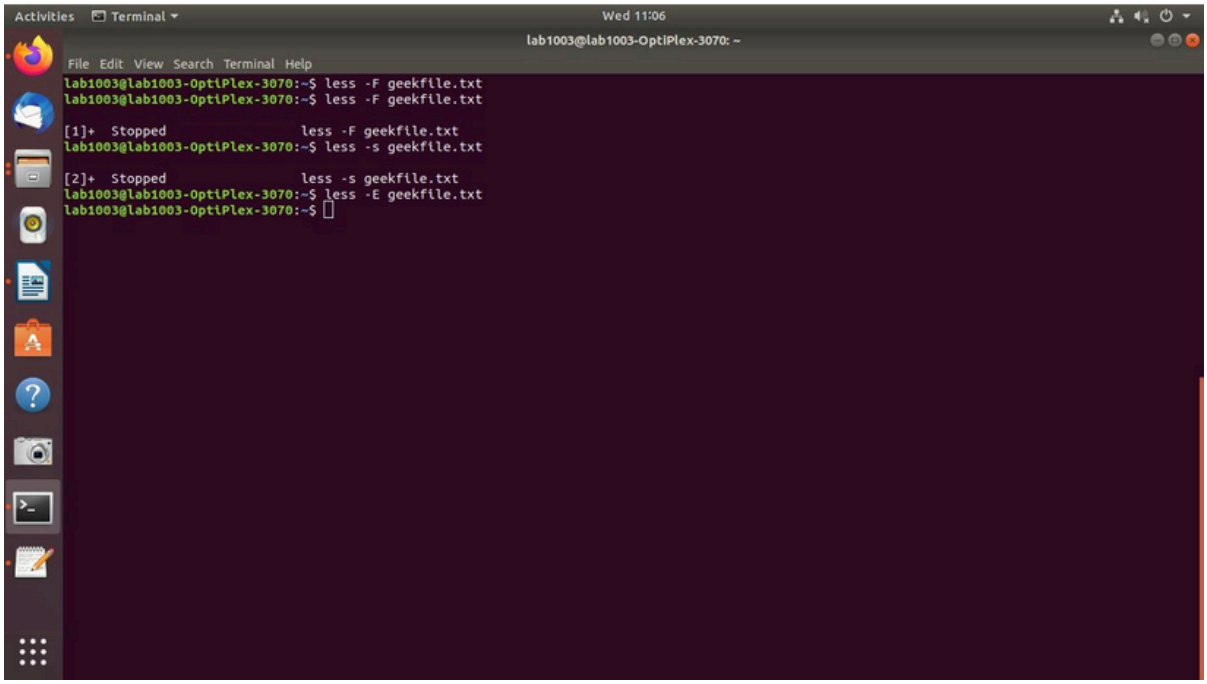
-i Ignore cases when searching.

-n Suppress line numbers.

-p Start at the first occurrence of the patterspecified pattern in the file.
n

-s Squeeze consecutive blank lines into a single line.


Activities
Terminal
Wed 11:06
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
garv jain is my name
garv
garv
garv
garv
garv
garv\
dwe
geekfile.txt


Activities
Terminal
Wed 11:06
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
lab1003@lab1003-OptiPlex-3070:~$ less -F geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$ less -F geekfile.txt
[1]+ Stopped less -F geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$ less -s geekfile.txt
[2]+ Stopped less -s geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$ less -E geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$

**more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user do scroll up and down through the page. The syntax along with options and command is as follows. Another application of more is to use it with some other command after a pipe. When the output is large, we can use more command to see output one by one**

**Options:**

**-d : Use this command in order to help the user to navigate. It displays “[Press space to continue, ‘q’ to quit.]” and displays “[Press ‘h’ for instructions.]” when wrong key is pressed.**

**Example:**

**more -d sample.txt**
**-p : This option clears the screen and then displays the text.**

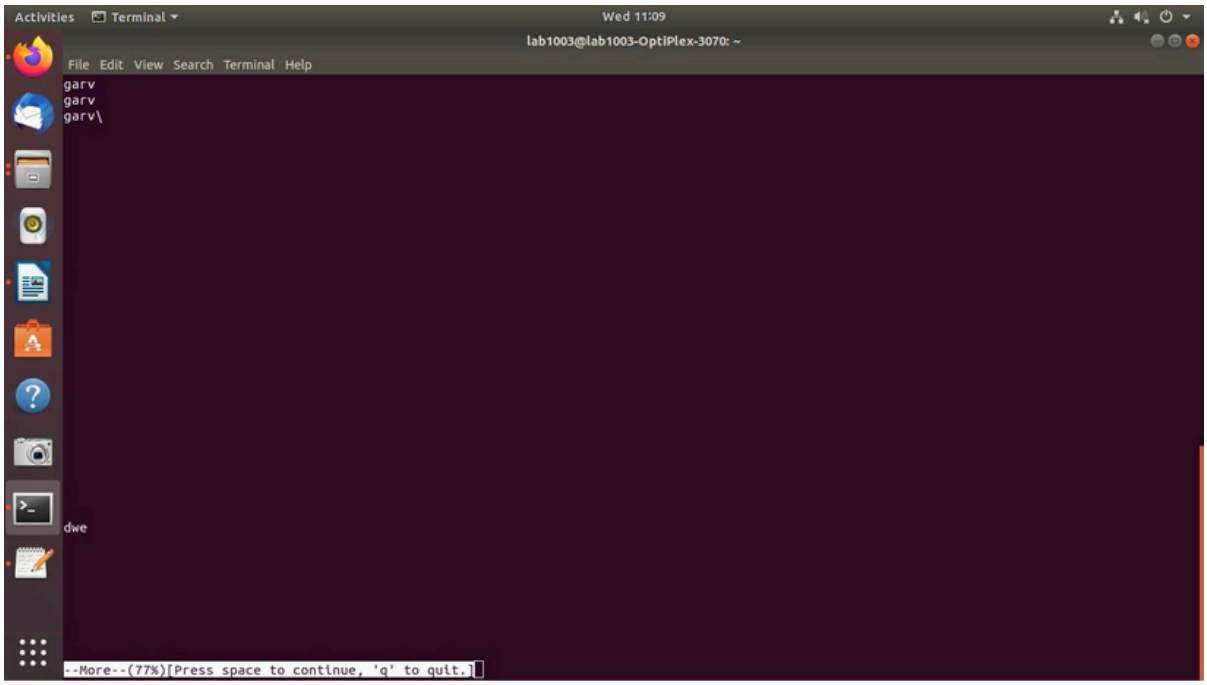
**Example:**

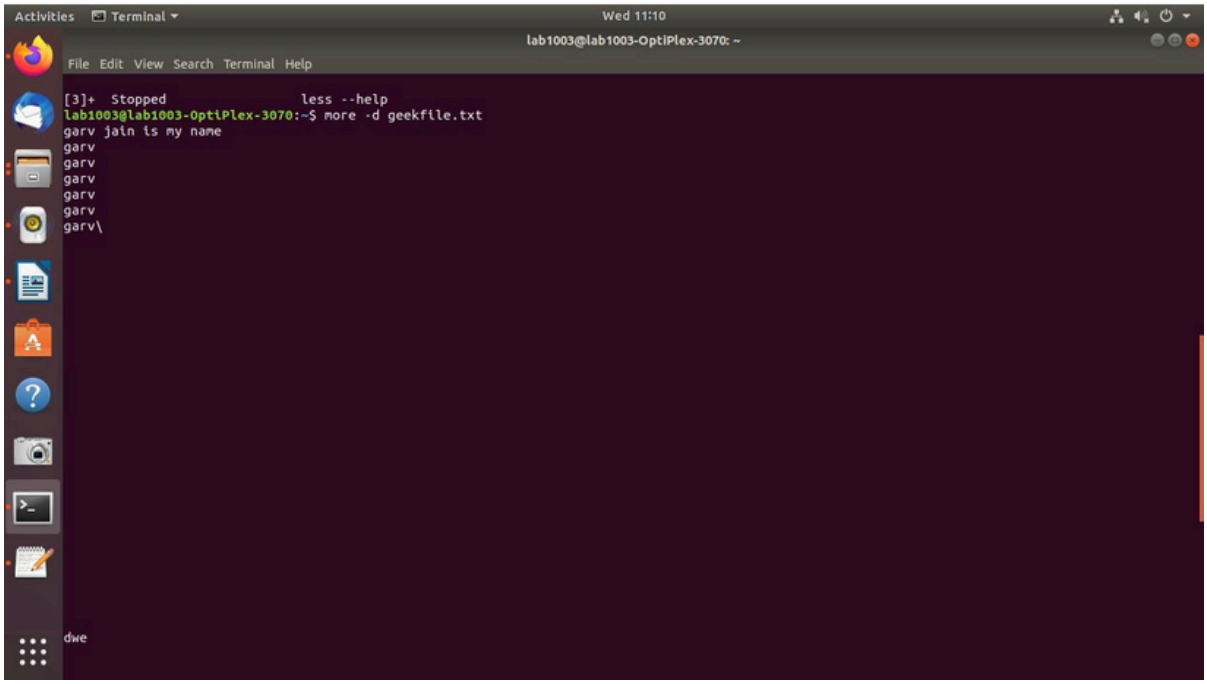
**more -p sample.txt**

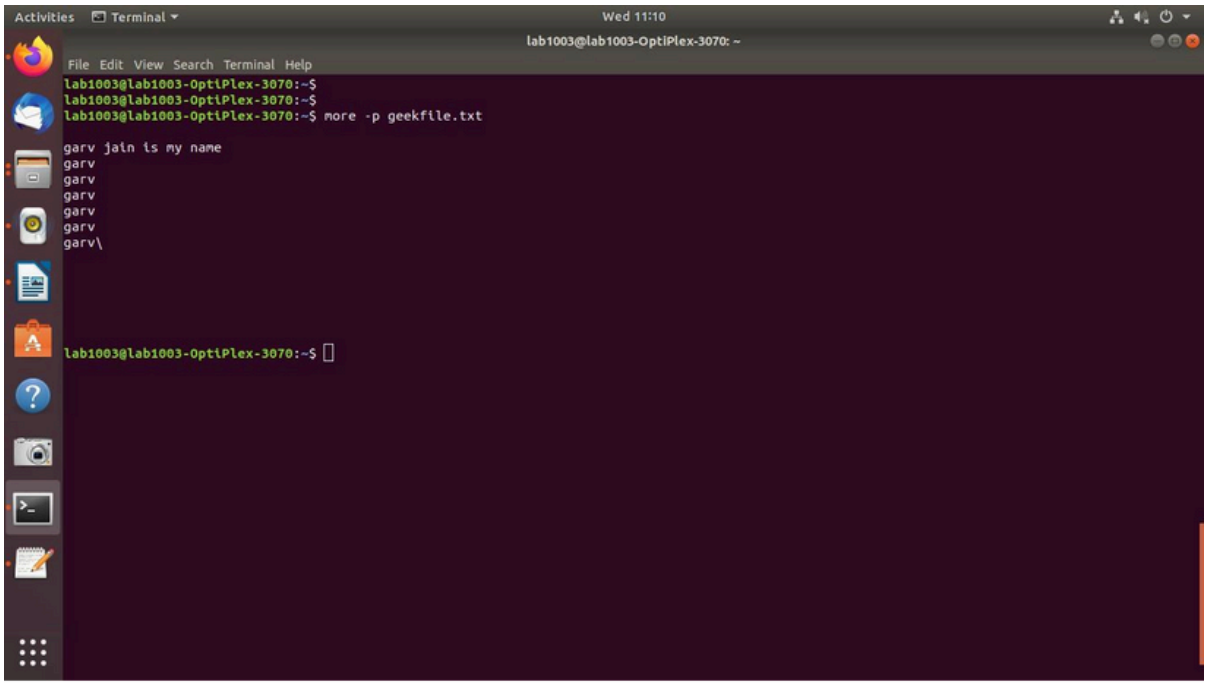
**-s : This option squeezes multiple blank lines into one single blank line.**

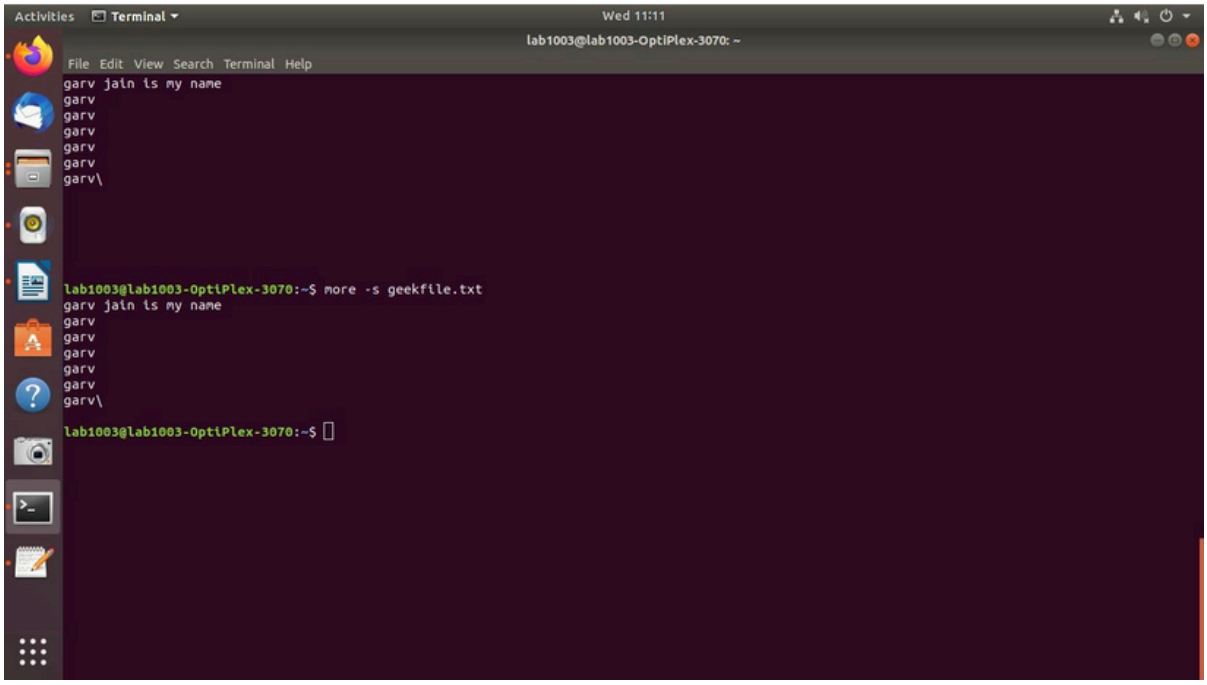
**Example:**

**more -s sample.txt**


Activities
Terminal
Wed 11:09
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
garv
garv
garv\
dwe
--More--(77%)[Press space to continue, 'q' to quit.]


Activities
Terminal
Wed 11:10
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
[3]+ Stopped less --help
lab1003@lab1003-OptiPlex-3070:~$ more -d geekfile.txt
garv jain is my name
garv
garv
garv
garv
garv
garv\
dwe


Activities
Terminal
Wed 11:10
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
lab1003@lab1003-OptiPlex-3070:~$
lab1003@lab1003-OptiPlex-3070:~$
lab1003@lab1003-OptiPlex-3070:~$ more -p geekfile.txt
garv jain is my name
garv
garv
garv
garv
garv
garv\
lab1003@lab1003-OptiPlex-3070:~$


Activities
Terminal
Wed 11:11
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
garv jain is my name
garv
garv
garv
garv
garv
garv\
lab1003@lab1003-OptiPlex-3070:~$ more -s geekfile.txt
garv jain is my name
garv
garv
garv
garv
garv
garv\
lab1003@lab1003-OptiPlex-3070:~$

**File Command**

**file command is used to determine the type of a file. .*file* type may be of human-readable(e.g. 'ASCII text') or MIME type(e.g. 'text/plain; charset=us-ascii'). This command tests each argument in an attempt to categorize it.**

**file [option] [filename**

**-b, –brief : This is used to display just file type in brief mode.**

**Syntax:**

**file -b filename**

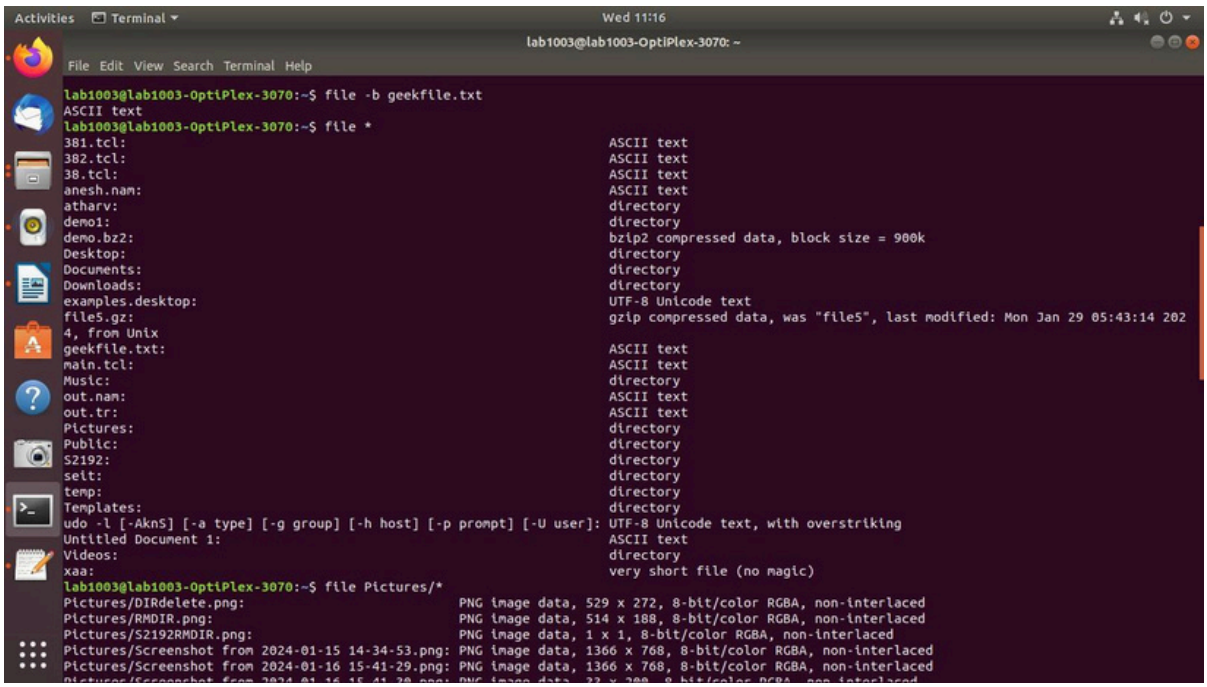
*** option : Command displays**

**the all files's file type. file ***

**directoryname/* option : This is used to display all files filetypes in particular directory.**

**Syntax:**

**file directoryname/***


Activities
Terminal
Wed 11:16
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
lab1003@lab1003-OptiPlex-3070:~$ file -b geekfile.txt
ASCII text
lab1003@lab1003-OptiPlex-3070:~$ file *
381.tcl: ASCII text
382.tcl: ASCII text
38.tcl: ASCII text
anesh.nam: ASCII text
atharv: directory
demo1: directory
demo.bz2: bzip2 compressed data, block size = 900k
Desktop: directory
Documents: directory
Downloads: directory
examples.desktop: UTF-8 Unicode text
file5.gz: gzip compressed data, was "file5", last modified: Mon Jan 29 05:43:14 202
4, from Unix
geekfile.txt: ASCII text
main.tcl: ASCII text
Music: directory
out.nam: ASCII text
out.tr: ASCII text
Pictures: directory
Public: directory
S2192: directory
seit: directory
temp: directory
Templates: directory
udo -l [-AknS] [-a type] [-g group] [-h host] [-p prompt] [-U user]: UTF-8 Unicode text, with overstriking
Untitled Document 1: ASCII text
Videos: directory
xaa: very short file (no magic)
lab1003@lab1003-OptiPlex-3070:~$ file Pictures/*
Pictures/DIRdelete.png: PNG image data, 529 x 272, 8-bit/color RGBA, non-interlaced
Pictures/RMDIR.png: PNG image data, 514 x 188, 8-bit/color RGBA, non-interlaced
Pictures/S2192RMDIR.png: PNG image data, 1 x 1, 8-bit/color RGBA, non-interlaced
Pictures/Screenshot from 2024-01-15 14-34-53.png: PNG image data, 1366 x 768, 8-bit/color RGBA, non-interlaced
Pictures/Screenshot from 2024-01-16 15-41-29.png: PNG image data, 1366 x 768, 8-bit/color RGBA, non-interlaced

**Type**

**The type command is used to describe how its argument would be translated if used as commands. It is also used to find out whether it is built-in or external binary file.-a : This option is used to find out whether it is an alias, keyword or a function and it also displays the path of an executable, if available.**

**Example:**

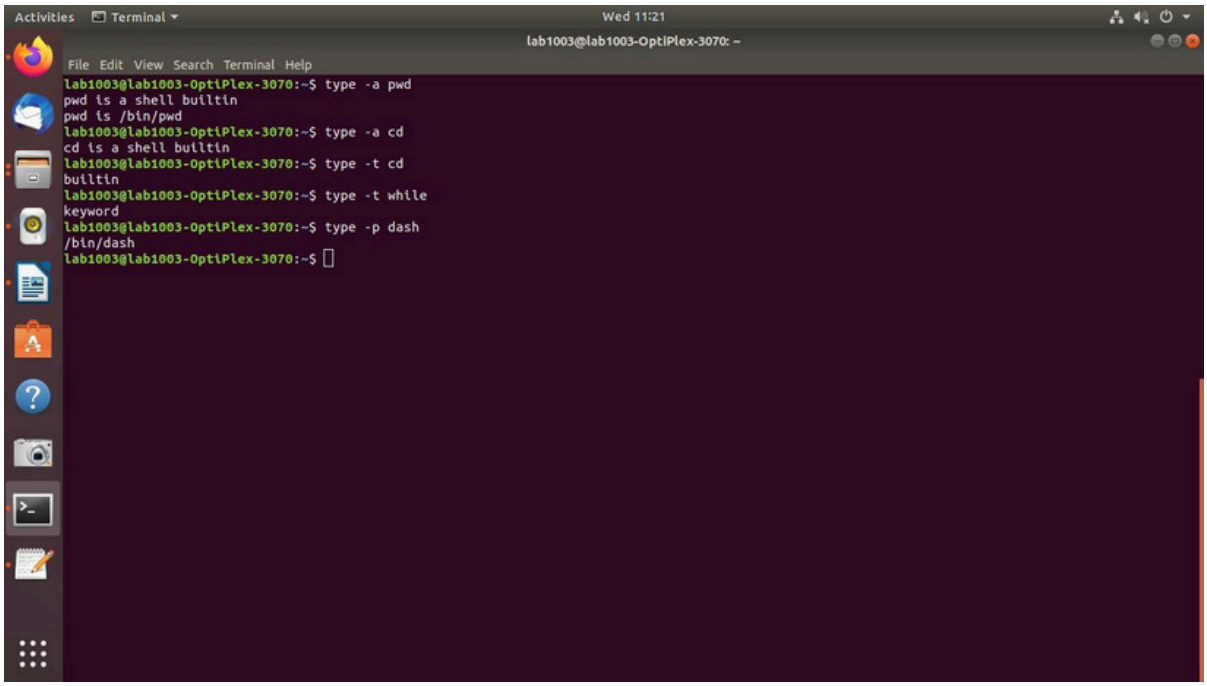
**-a :**

**This option is used to find out whether it is an alias, keyword or a function and it also displays the path of an executable, if available.**

**-t : This option will display a single word as an output.**

- alias – if command is a shell alias
- keyword – if command is a shell reserved word
- builtin – if command is a shell builtin
- function – if command is a shell function
- file – if command is a disk file

**-p : This option displays the name of the disk file which would be executed by the shell. It will return nothing if the command is not a disk file.**




## Split

**Split command in Unix is used to split large files into smaller files. It splits the files into 1000 lines per file(by default) and even allows users to change the number of lines as per requirement.**

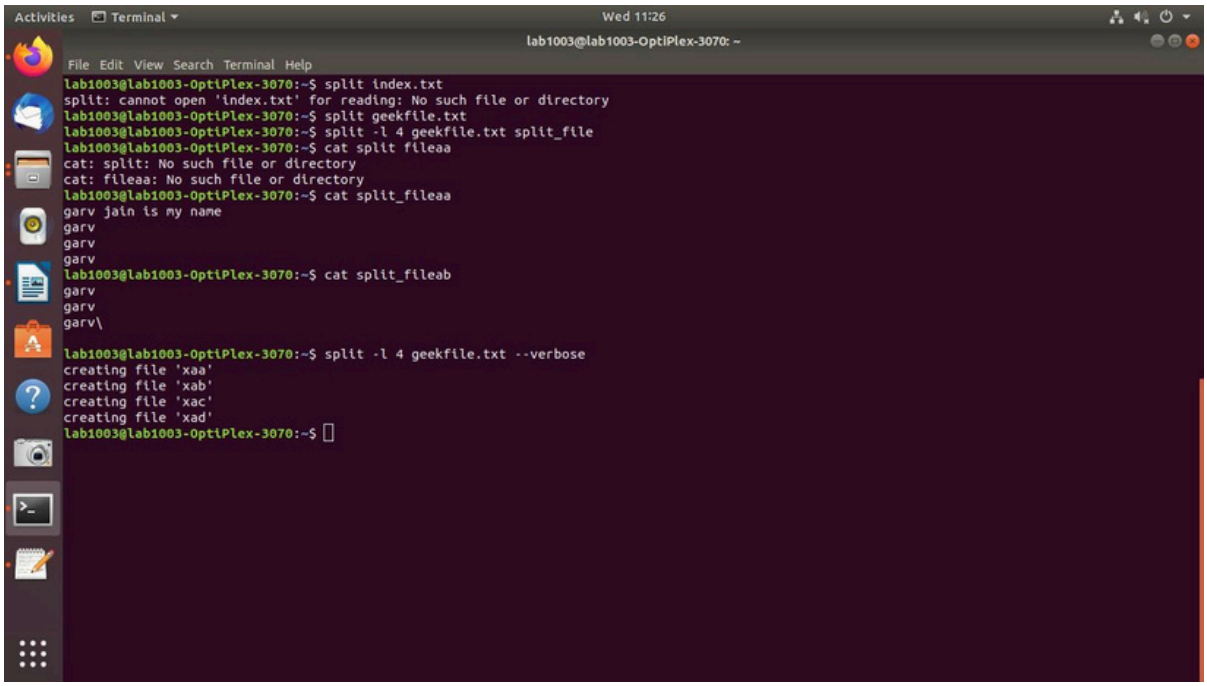
**The names of the files are PREFIXaa, PREFIXab, PREFIXac, and so on. By default the PREFIX of files name is x and the default size of each split file is 1000 lines per file and both the parameters can be changed with ease. It is generally used with log and archive files**

**as they are very large and have a lot of lines, So in order to break them into small files for analysis split command is used.**

**1. Split file into short files. Assume a file name with name index.txt. Use below split command to break it into pieces.**

**2. Split file based on number of lines.**

**3. Split command with verbose option. We can also run split command in verbose mode by using '–verbose'. It will give a diagnostic message each time a new split file is created.**




## Cmp

**cmp command in Unix/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.**

**● When cmp is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no**

**difference is found *i.e* the files compared are identical.**

- **cmp displays no message and simply returns the prompt if the files compared are identical.**

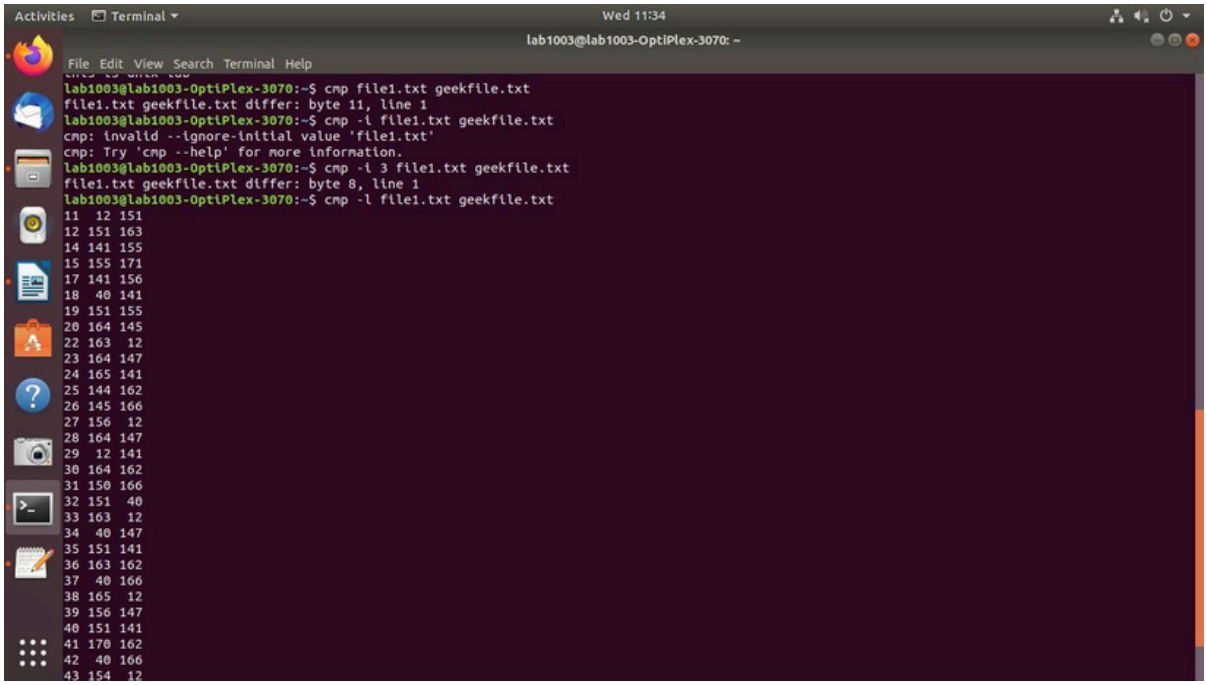
**Syntax:**

**cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]**

**1. -b(print-bytes) : If you want cmp displays the differing bytes in the output when used with -b option.**

**2. -i [bytes-to-be-skipped] : Now, this option when used with cmp command helps to skip a particular number of initial bytes from both the files and then after skipping it compares the files. This can be done by specifying the number of bytes as argument to the -i command line option.**

**3. -l option : This option makes the cmp command print byte position and byte value for all differing bytes.**




**Find**

**Unix, renowned for its robust command-line interface, provides a suite of powerful tools for efficient file and**

**directory management. Among these, the “find” command stands out as an indispensable asset, offering unparalleled versatility in searching for files based on diverse criteria. This article explores the prowess of the find command, shedding light on its capabilities and how it serves as a go-to tool for Unix users aiming to locate files swiftly and effectively.**

## 2. How to Search Files with a Pattern Using `find` Command in Unix

**This command is tailored for discovering files within a directory that adhere to a specific naming pattern. In this case, it identifies files ending with ‘.txt’ within the “GFG” directory.**

**The command looks for files with names ending in ‘.txt’ within the “GFG” directory, presenting a list of matching files.**

## 3. Search for Empty Files and Directories Using `find` Command in Unix

**This query is tailored for discovering and listing empty files and directories within a specified directory.**

```
find ./GFG -empty
```

**The `find` command identifies and lists all empty folders and files within the “GFG” directory or its subdirectories.**

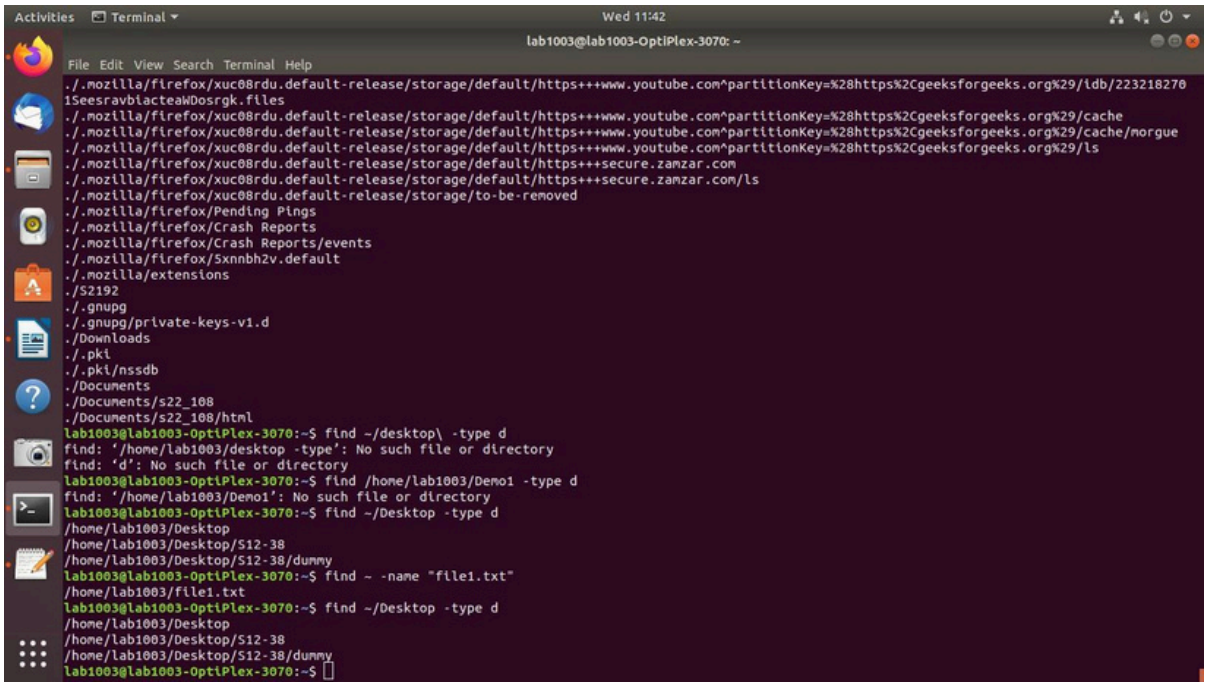
## 6. Display Repository Hierarchy Using `find` Command in Unix

**This command is utilized to display the hierarchical structure of repositories and sub-repositories within a given directory.**

```
find . -type d
```

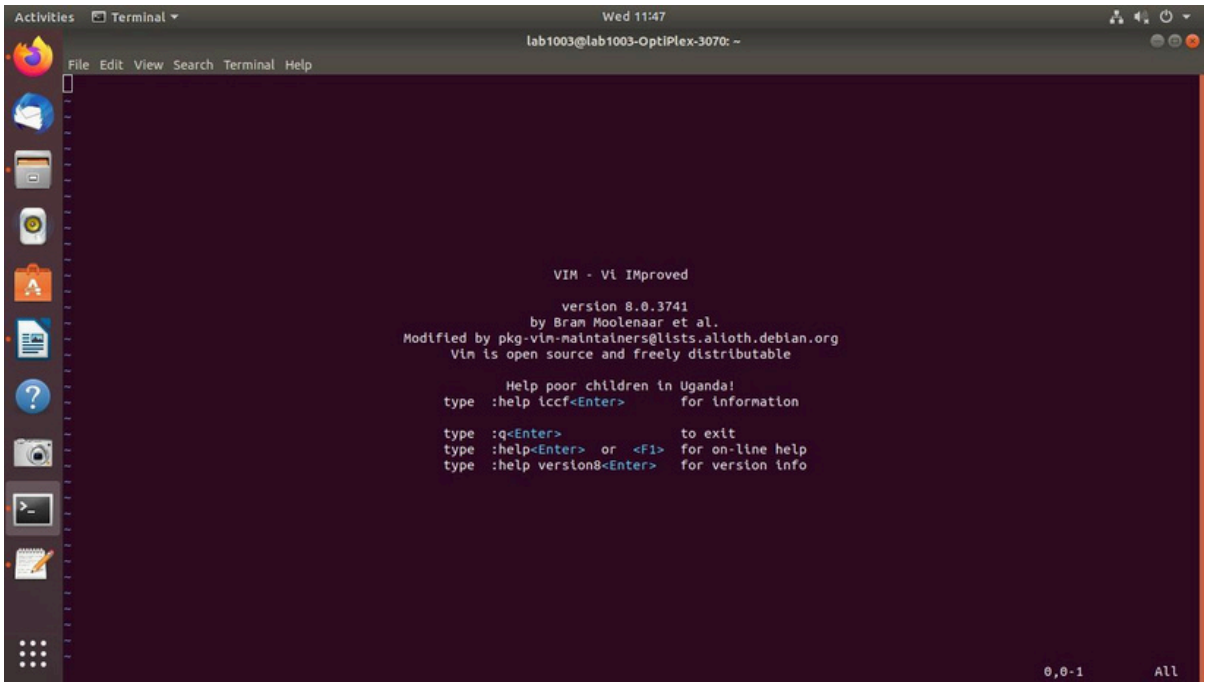
**This command displays all the repositories and sub-repositories present in the current repository. In the below example, we are currently in a repository namely “GeeksforGeeks” which contains a repo “Unix”, which contains sub-repo “UnixCmds” which further contains a repo “FindCmd”. The ouput of**

**below cmd is simply displaying this info. Please note that in this case if you will use “ls” cmd then it will only show “/Unix”.**




## 27. IM

**Vim, short for “Vi Improved,” is a highly efficient and powerful text editor that is favored by many developers and system administrators. One of Vim’s distinguishing features is its unique approach to editing, which involves six distinct modes. Understanding these modes is crucial for mastering Vim and unleashing its full potential. In this article, we’ll explore each of the six modes in detail, providing examples and explanations to help you grasp their functionality.**

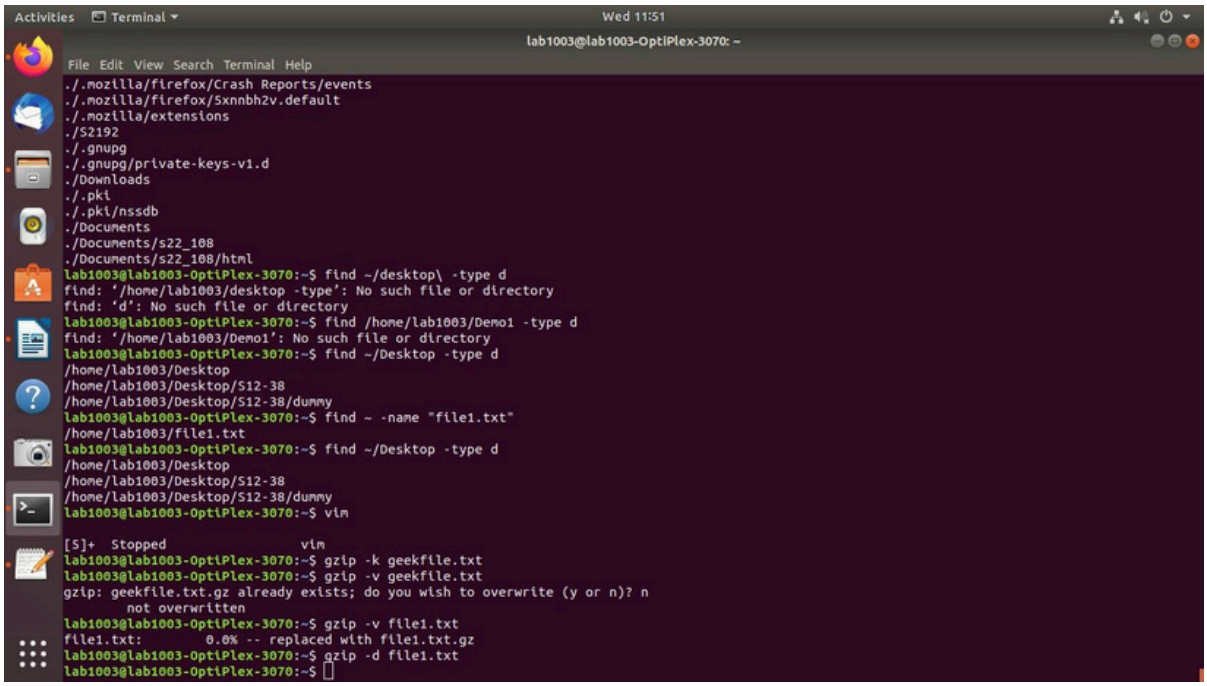



## 28. **Gzip**

**gzip command compresses files. Each single file is compressed into a single file. The compressed file consists of a GNU zip header and deflated data. If given a file as an argument, gzip compresses the file, adds a “.gz” suffix, and deletes the original file. With no arguments, gzip compresses the standard input and writes the compressed file to the standard output.**

| Option | Description |
| --- | --- |
| -f | Forcefully compress a file even if a compressed version with the same name already exists. |
| -k | Compress a file and keep the original file, resulting in both the compressed and original files. |
| -L | Display the gzip license for the software. |

| | |
|---|---|
| **-r** | **Recursively compress all files in a folder and its subfolders.** |
| **-v** | **Display the name and percentage reduction for each file compressed or decompressed.** |
| **-d** | **Decompress a file that was compressed using the gzip command.** |



29. **Bzip2**

***bzip2*** **command in Unix is used to compress and decompress the files i.e. it helps in binding the files into a single file which takes less storage space as the original file use to take. It has a slower decompression time and higher memory use. It uses Burrows-Wheeler block sorting text compression algorithm, and Huffman Coding. Each file is replaced by a compressed version of itself, with the name original name of the file followed by extension bz2.**

**Syntax:**

**bzip2 [OPTIONS] filenames ...**

**k: This option does compression but does not deletes the original file.**

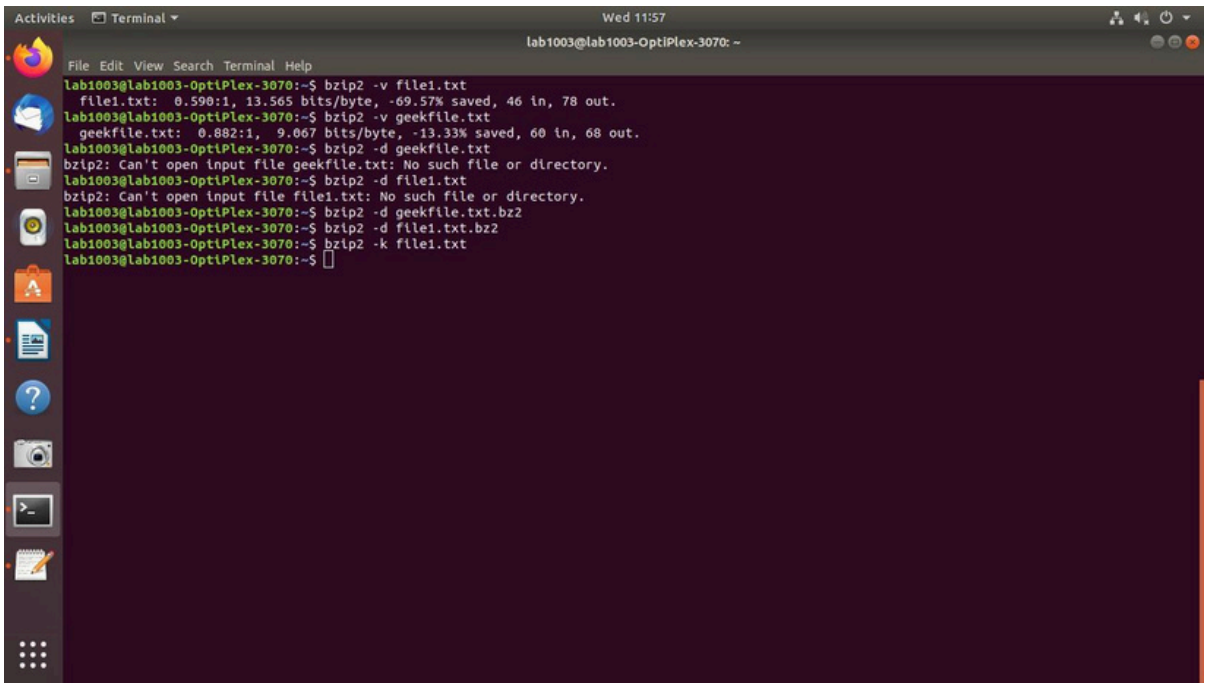
**$ bzip2 -k input.txt**

**-d : This option is used for decompression of compressed files.**

**$ bzip2 -d input.txt.bz2**

**-v : Verbose mode show the compression ratio for each file processed. It also increases the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.**

# $ bzip2 -v input.txt



30. **Locate**

**locate command in Unix is used to find the files by name. There are two most widely used file-searching utilities accessible to users called to find and locate. The locate utility works better and faster than the find command counterpart because instead of searching the file system when a file search is initiated, it would look through a database. This database contains bits and parts of files and their corresponding paths on your system. By default, locate command does not check whether the files found in the database still exist and it never reports files created after the most recent update of the relevant database**

## OPTIONS DESCRIPTION

**-b, –basename Match only the base name against the specified patterns,**

**which is the opposite of *–wholename*.**

**-c, –count Instead of writing file names on standard output, write the number of matching entries only.**

**-d, –database DBPAT Replace the default database with DBPATH. DBPATH is a : (colon) separated list of database file names. If more than one *–database* option is specified, the resulting path is a concatenation of the separate paths. An empty database file name is replaced by the default database. A database file name – refers to the standard input. Note that a database can be read from the standard input only once.**

| | |
|---|---|
| **-e, –existing** | **Print only entries that refer to files existing at the time locate is run.** |

**-L, –follow** When checking whether files exist (if the *–existing* option is specified), follow trailing symbolic links. This causes bro ken symbolic links to be omitted from the output. This option is the default behavior. The opposite can be specified using –*nofollow*.

**-h, –help** Write a summary of the available options to standard output and exit successfully.

**-i, –ignore-case** Ignore case distinctions when matching patterns.

**-l, –limit, -n LIMIT** Exit successfully after finding LIMIT entries. If the *–count* option is specified, the resulting count is also limited to

LIMIT.

-m, –mmap Ignored, but included for compatibility with BSD and GNU locate.

-P, –nofollow, -H When checking whether files exist (if the –existing option is specified), do not follow trailing symbolic links. This causes broken symbolic links to be reported like other files. This option is the opposite of –follow.

-0, –null Separate the entries on output using the ASCII NULL character instead of writing each entry on a separate line. This option is designed for interoperability with the –null option of GNU

| | |
|---|---|
| | xargs. |
| -S, –statistics | Write statistics about each read database to standard output instead of searching for files and exit successfully. |
| q, –quiet | Write no messages about errors encountered while reading and processing databases. |

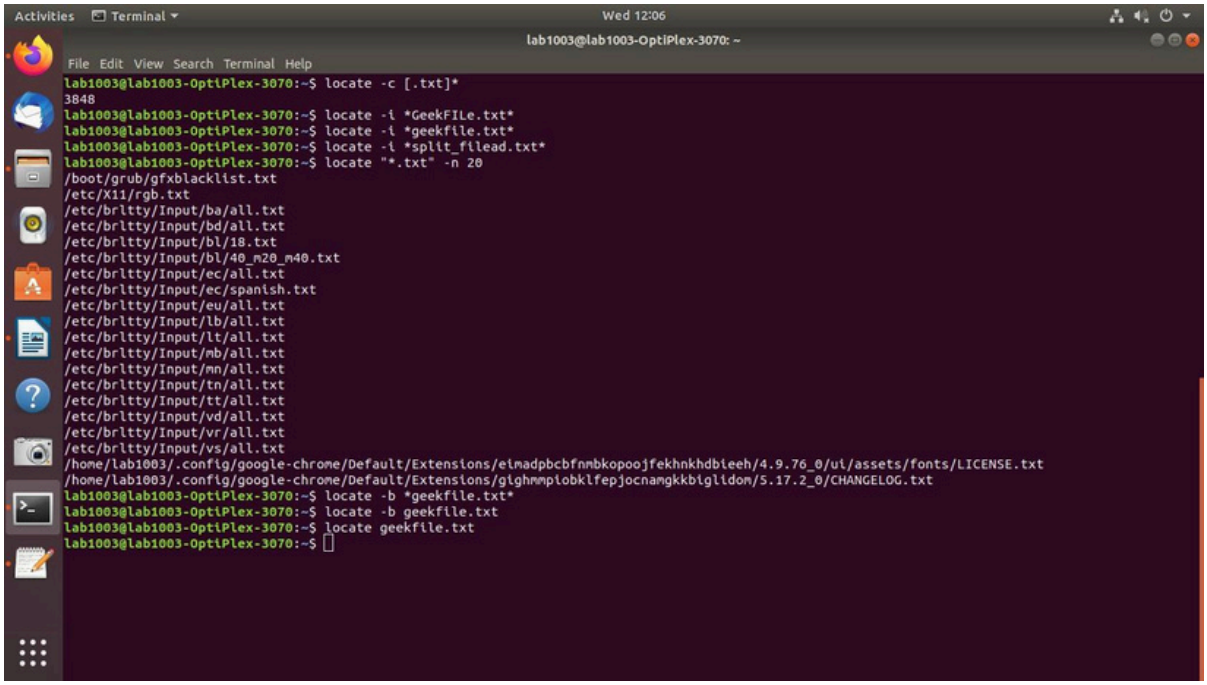
**-r, –regexp REGEXP Search for a basic regexp REGEXP. No PATTERNs are allowed if this option is used, but this option can be specified multiple times.**

**–regex Interpret all PATTERNs as extended regexps.**

**-s, –stdio Ignored, for compatibility with BSD and GNU locate.**

**-V, –version Write information about the version and license of locate on standard output and exit successfully.**

**-w, –wholename Match only the whole path name against the specified patterns. This option is the default behavior. The opposite can be specified using –*basename*.**


Activities
Terminal
Wed 12:06
lab1003@lab1003-OptiPlex-3070: ~
File Edit View Search Terminal Help
lab1003@lab1003-OptiPlex-3070:~$ locate -c [.txt]*
3848
lab1003@lab1003-OptiPlex-3070:~$ locate -i *GeekFILe.txt*
lab1003@lab1003-OptiPlex-3070:~$ locate -i *geekfile.txt*
lab1003@lab1003-OptiPlex-3070:~$ locate -i *split_filead.txt*
lab1003@lab1003-OptiPlex-3070:~$ locate "*.txt" -n 20
/boot/grub/gfxblacklist.txt
/etc/X11/rgb.txt
/etc/brltty/Input/ba/all.txt
/etc/brltty/Input/bd/all.txt
/etc/brltty/Input/bl/18.txt
/etc/brltty/Input/bl/40_m20_m40.txt
/etc/brltty/Input/ec/all.txt
/etc/brltty/Input/ec/spanish.txt
/etc/brltty/Input/eu/all.txt
/etc/brltty/Input/lb/all.txt
/etc/brltty/Input/lt/all.txt
/etc/brltty/Input/mb/all.txt
/etc/brltty/Input/mn/all.txt
/etc/brltty/Input/tn/all.txt
/etc/brltty/Input/tt/all.txt
/etc/brltty/Input/vd/all.txt
/etc/brltty/Input/vr/all.txt
/etc/brltty/Input/vs/all.txt
/home/lab1003/.config/google-chrome/Default/Extensions/eimadpbcbfnmbkopoojfekhnkhdbieeh/4.9.76_0/ui/assets/fonts/LICENSE.txt
/home/lab1003/.config/google-chrome/Default/Extensions/gighmmpiobklfepjocnamgkkbiglidom/5.17.2_0/CHANGELOG.txt
lab1003@lab1003-OptiPlex-3070:~$ locate -b *geekfile.txt*
lab1003@lab1003-OptiPlex-3070:~$ locate -b geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$ locate geekfile.txt
lab1003@lab1003-OptiPlex-3070:~$