Linux Directory Commands

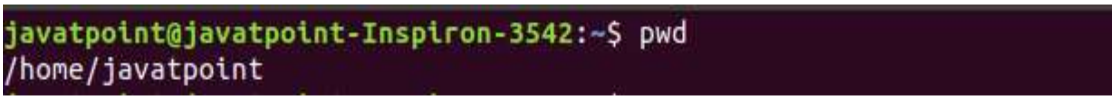
1. pwd Command

The pwd command is used to display the location of the current working directory.

Syntax:

pwd

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ pwd
/home/javatpoint
```

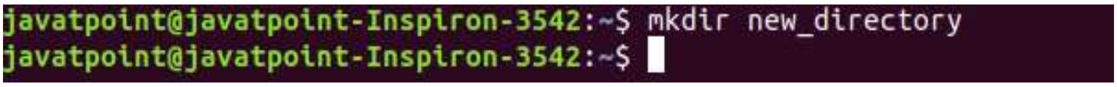
2. mkdir Command

The mkdir command is used to create a new directory under any directory.

Syntax:

mkdir **<directory** name**>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ mkdir new_directory
javatpoint@javatpoint-Inspiron-3542:~$
```

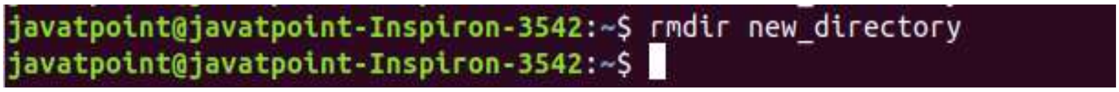
3. rmdir Command

The rmdir command is used to delete a directory.

Syntax:

rmdir **<directory** name**>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ rmdir new_directory
javatpoint@javatpoint-Inspiron-3542:~$
```

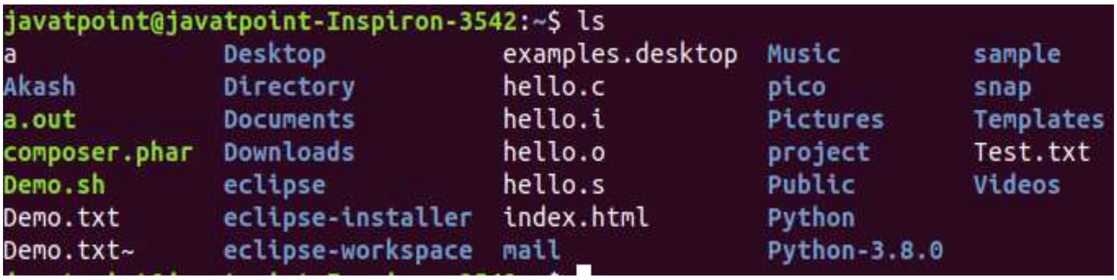
4. ls Command

The ls command is used to display a list of content of a directory.

Syntax:

ls

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ls
a              Desktop            examples.desktop  Music         sample
Akash          Directory          hello.c           pico          snap
a.out          Documents          hello.i           Pictures      Templates
composer.phar  Downloads          hello.o           project       Test.txt
Demo.sh        eclipse            hello.s           Public        Videos
Demo.txt       eclipse-installer  index.html        Python
Demo.txt~      eclipse-workspace  mail              Python-3.8.0
```

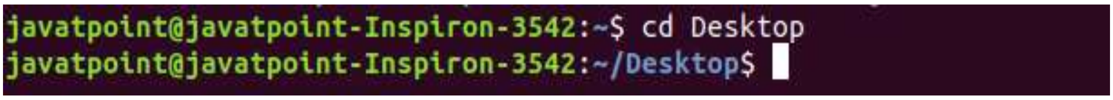
5. cd Command

The cd command is used to change the current directory.

Syntax:

cd <directory name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cd Desktop
javatpoint@javatpoint-Inspiron-3542:~/Desktop$
```

Linux File commands

6. touch Command

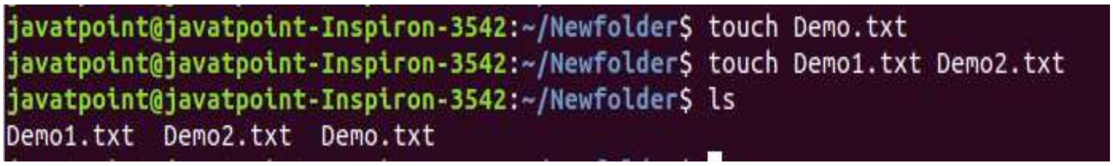
The touch command is used to create empty files. We can create multiple empty files by executing it once.

Syntax:

touch <file name>

touch <file1> <file2> ....

Output:



```
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ touch Demo.txt
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ touch Demo1.txt Demo2.txt
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ ls
Demo1.txt  Demo2.txt  Demo.txt
```

7. cat Command

The cat command is a multi-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file, and more.

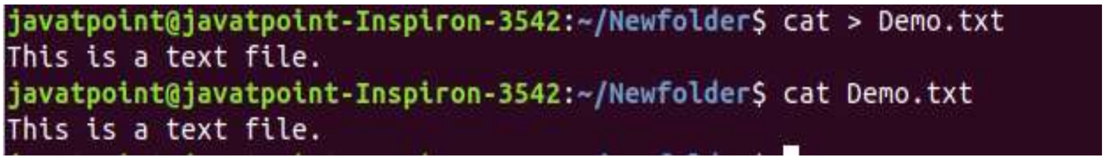
Syntax:

cat [OPTION]... [FILE]..

To create a file, execute it as follows:

cat > <file name>   OR cat <file name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ cat > Demo.txt
This is a text file.
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ cat Demo.txt
This is a text file.
```

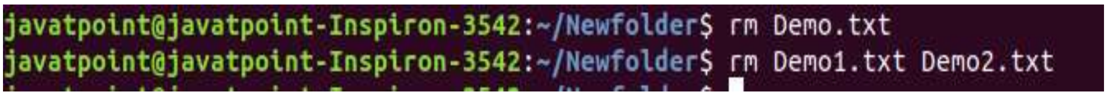
8. rm Command

The rm command is used to remove a file.

Syntax:

rm <file name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ rm Demo.txt
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ rm Demo1.txt Demo2.txt
```

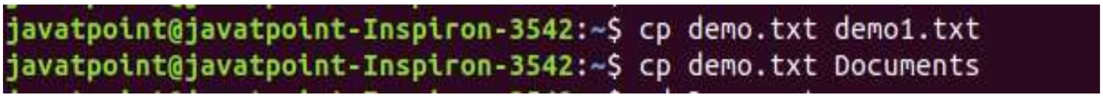
9. cp Command

The cp command is used to copy a file or directory.

Syntax:

cp <existing file name> <new file name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cp demo.txt demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ cp demo.txt Documents
```

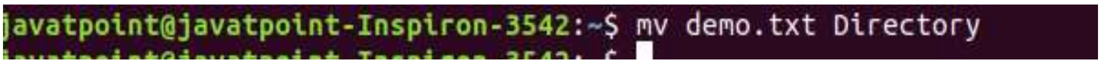
10. mv Command

The mv command is used to move a file or a directory form one location to another location.

Syntax:

mv <file name> <directory path>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ mv demo.txt Directory
```

11. rename Command

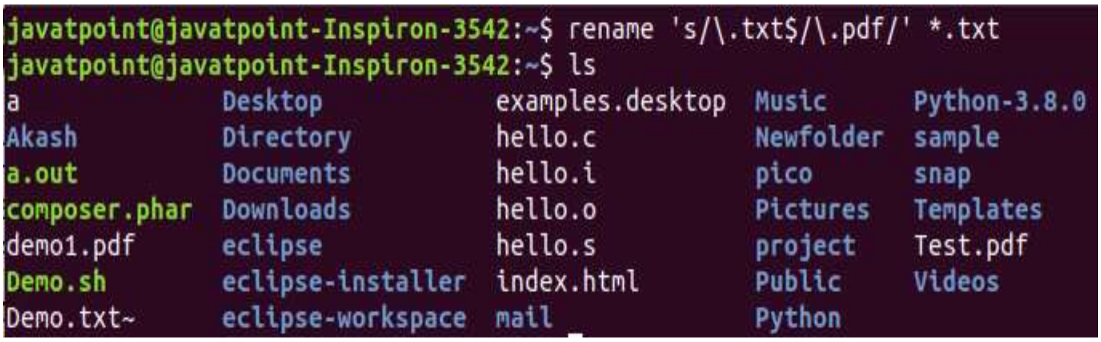
The rename command is used to rename files. It is useful for renaming a large group of files.

Syntax:

rename 's/old-name/new-name/' files

For example, to convert all the text files into pdf files, execute the below command: rename 's/\.txt$/\.pdf/' *.txt

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ rename 's/\.txt$/\.pdf/' *.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a              Desktop            examples.desktop  Music      Python-3.8.0
Akash          Directory          hello.c           Newfolder  sample
a.out          Documents          hello.i           pico       snap
composer.phar  Downloads          hello.o           Pictures   Templates
demo1.pdf      eclipse            hello.s           project    Test.pdf
Demo.sh        eclipse-installer  index.html        Public     Videos
Demo.txt~      eclipse-workspace  mail              Python
```

<u>Linux File Content Commands</u>

12. head Command

The head command is used to display the content of a file. It displays the first 10 lines of a file.

Syntax:

head <file name>

13. tail Command

The tail command is similar to the head command. The difference between both commands is that it displays the last ten lines of the file content. It is useful for reading the error message.

Syntax:

tail <file name>

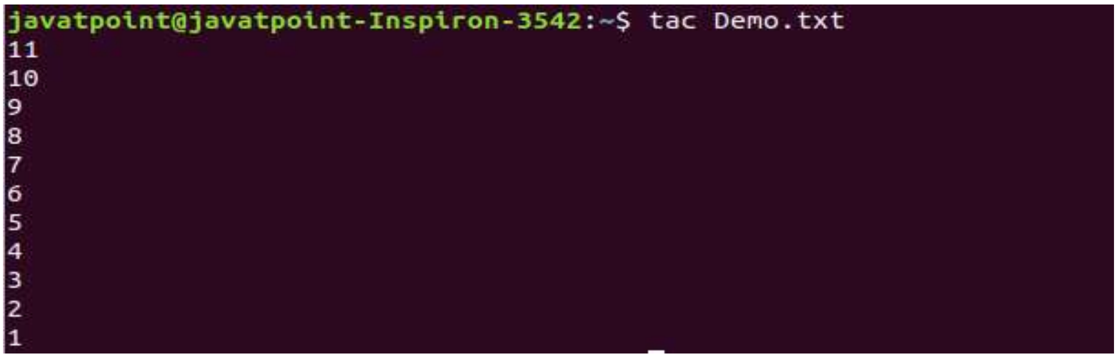
14. tac Command

The tac command is the reverse of cat command, as its name specified. It displays the file content in reverse order (from the last line).

Syntax:

tac <file name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ tac Demo.txt
11
10
9
8
7
6
5
4
3
2
1
```

15. more command

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

In more command, the following keys are used to scroll the page:

ENTER key: To scroll down page by line.

Space bar: To move to the next page.

b key: To move to the previous page.

/ key: To search the string.

Syntax:

more <file name>

16. less Command

The less command is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.' Comparatively, the more command cuts the output in the width of the terminal.

Syntax:

less <file name>

Linux User Commands

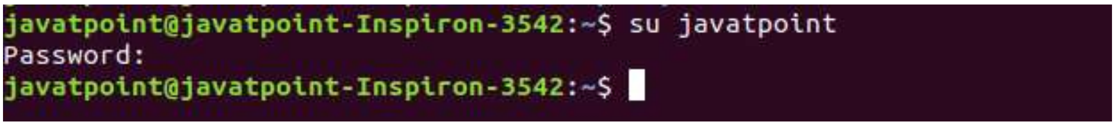
17. su Command

The su command provides administrative access to another user. In other words, it allows access of the Linux shell to another user.

Syntax:

su <user name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ su javatpoint
Password:
javatpoint@javatpoint-Inspiron-3542:~$
```

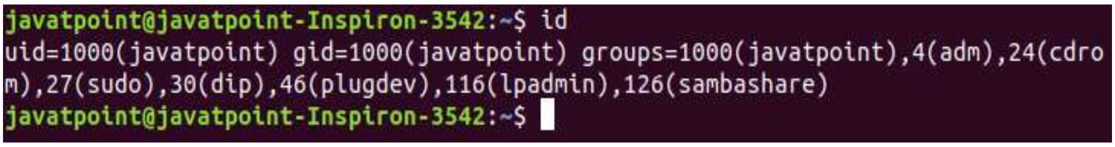
18. id Command

The id command is used to display the user ID (UID) and group ID (GID).

Syntax:

id

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ id
uid=1000(javatpoint) gid=1000(javatpoint) groups=1000(javatpoint),4(adm),24(cdro
m),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
javatpoint@javatpoint-Inspiron-3542:~$
```

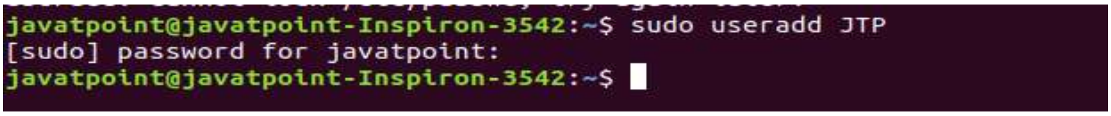
19. useradd Command

The useradd command is used to add or remove a user on a Linux server.

Syntax:

useradd username

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sudo useradd JTP
[sudo] password for javatpoint:
javatpoint@javatpoint-Inspiron-3542:~$
```

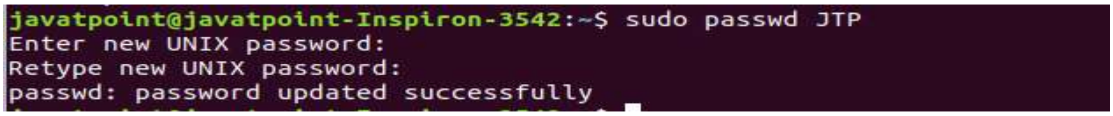
20. passwd Command

The passwd command is used to create and change the password for a user.

Syntax:

passwd <username>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sudo passwd JTP
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

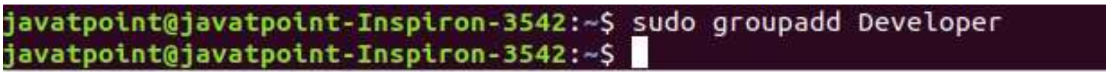
21. groupadd Command

The groupadd command is used to create a user group.

Syntax:

groupadd <group name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sudo groupadd Developer
javatpoint@javatpoint-Inspiron-3542:~$
```

Linux Filter Commands

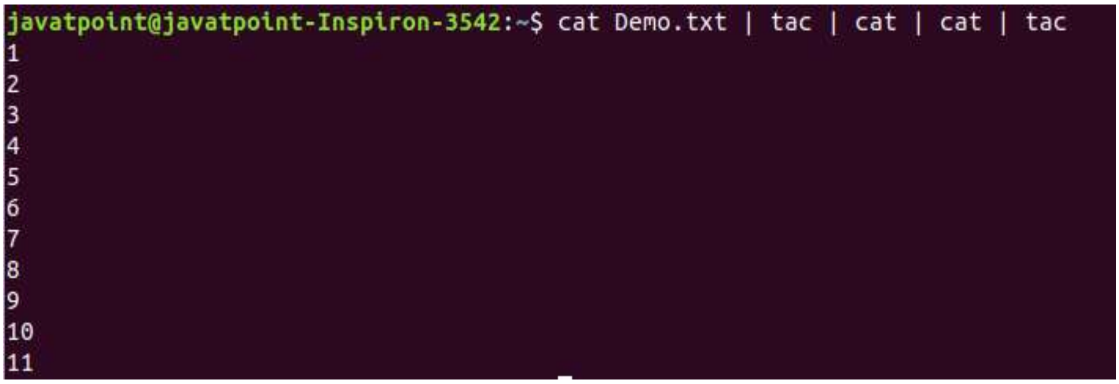
22. cat Command

The cat command is also used as a filter. To filter a file, it is used inside pipes.

Syntax:

cat <fileName> | cat or tac | cat or tac |. . .

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cat Demo.txt | tac | cat | cat | tac
1
2
3
4
5
6
7
8
9
10
11
```

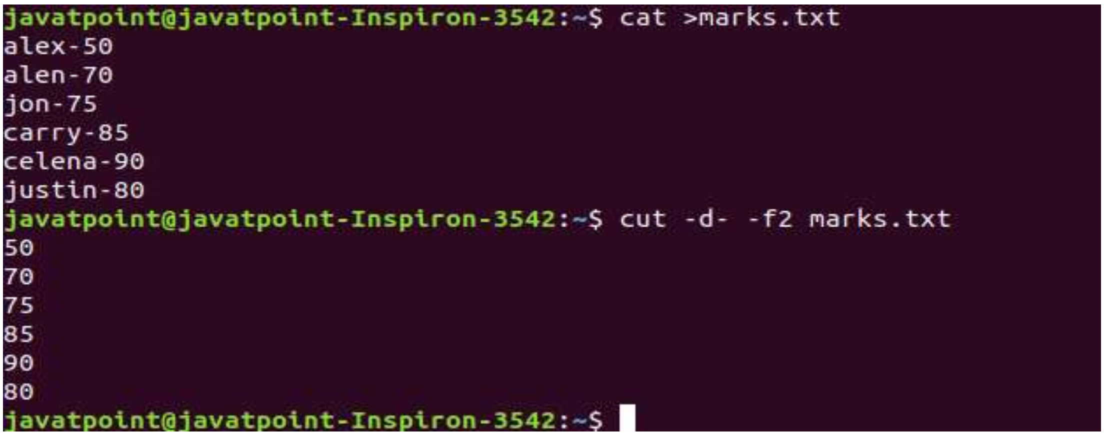
23. cut Command

The cut command is used to select a specific column of a file. The '-d' option is used as a delimiter, and it can be a space (' '), a slash (/), a hyphen (-), or anything else. And, the '-f' option is used to specify a column number.

Syntax:

cut -d(delimiter) -f(columnNumber) <fileName>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cat >marks.txt
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
javatpoint@javatpoint-Inspiron-3542:~$ cut -d- -f2 marks.txt
50
70
75
85
90
80
javatpoint@javatpoint-Inspiron-3542:~$
```

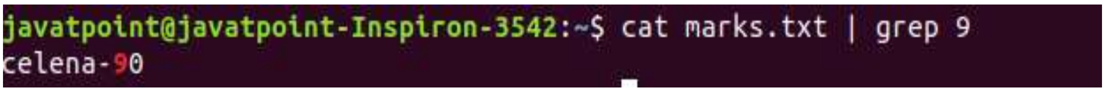
24. grep Command

The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "**global regular expression print**." It is useful for searching the content from a file. Generally, it is used with the pipe.

Syntax:

command | grep <searchWord>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | grep 9
celena-90
```

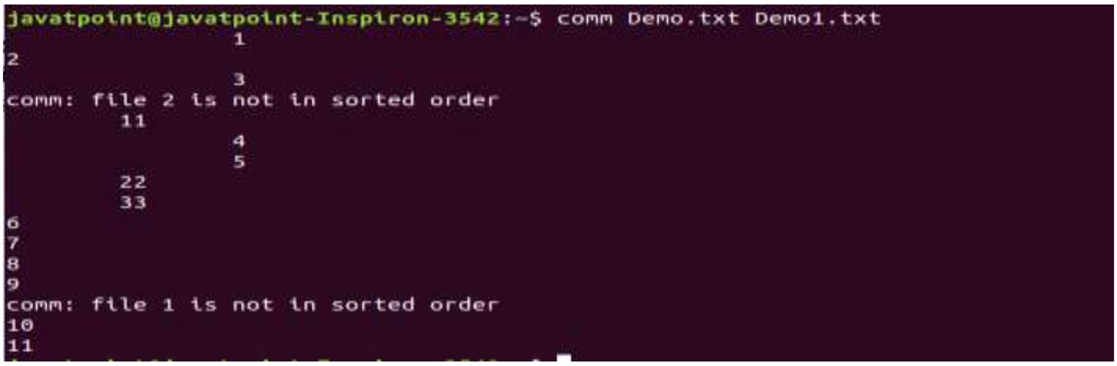
25. comm Command

The 'comm' command is used to compare two files or streams. By default, it displays three columns, first displays non-matching items of the first file, second indicates the non-matching item of the second file, and the third column displays the matching items of both files.

Syntax:

comm **<file1> <file2>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ comm Demo.txt Demo1.txt
                1
2
                3
comm: file 2 is not in sorted order
        11
                4
                5
        22
        33
6
7
8
9
comm: file 1 is not in sorted order
10
11
```

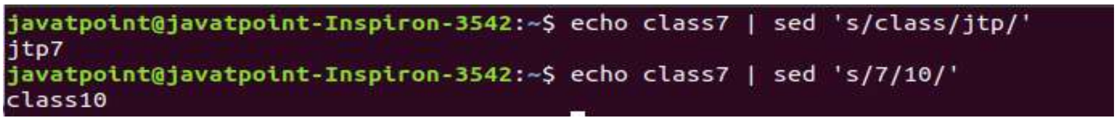
26. sed command

The sed command is also known as **stream editor**. It is used to edit files using a regular expression. It does not permanently edit files; instead, the edited content remains only on display. It does not affect the actual file.

Syntax:

command | sed 's/**<oldWord>**/**<newWord>**/'

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ echo class7 | sed 's/class/jtp/'
jtp7
javatpoint@javatpoint-Inspiron-3542:~$ echo class7 | sed 's/7/10/'
class10
```

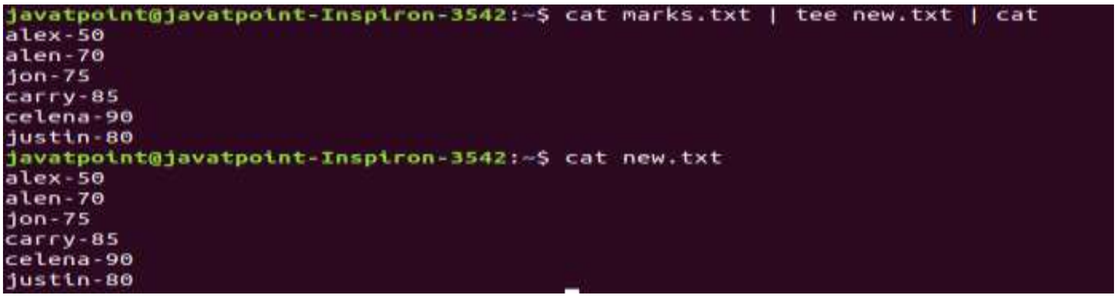
27. tee command

The tee command is quite similar to the cat command. The only difference between both filters is that it puts standard input on standard output and also write them into a file.

Syntax:

cat **<fileName>** | tee **<newFile>** | cat or tac |.....

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | tee new.txt | cat
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
javatpoint@javatpoint-Inspiron-3542:~$ cat new.txt
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
```

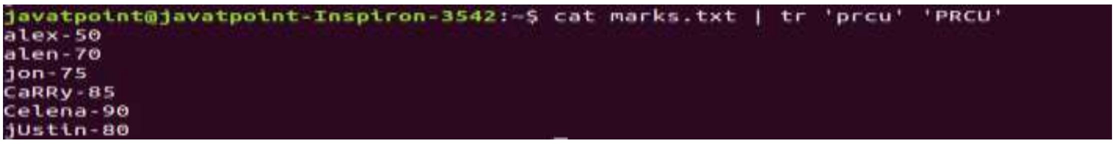
28. tr Command

The tr command is used to translate the file content like from lower case to upper case.

Syntax:

command | tr <'old'> <'new'>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | tr 'prcu' 'PRCU'
alex-50
alen-70
jon-75
CaRRy-85
Celena-90
jUstin-80
```

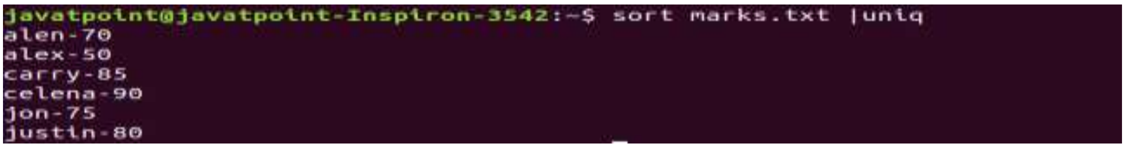
29. uniq Command

The uniq command is used to form a sorted list in which every word will occur only once.

Syntax:

command **<fileName>** | uniq

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sort marks.txt |uniq
alen-70
alex-50
carry-85
celena-90
jon-75
justin-80
```

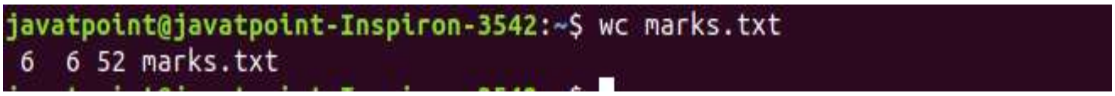
30. wc Command

The wc command is used to count the lines, words, and characters in a file.

Syntax:

wc **<file** name**>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ wc marks.txt
 6  6 52 marks.txt
```

31. od Command

The od command is used to display the content of a file in different s, such as hexadecimal, octal, and ASCII characters.

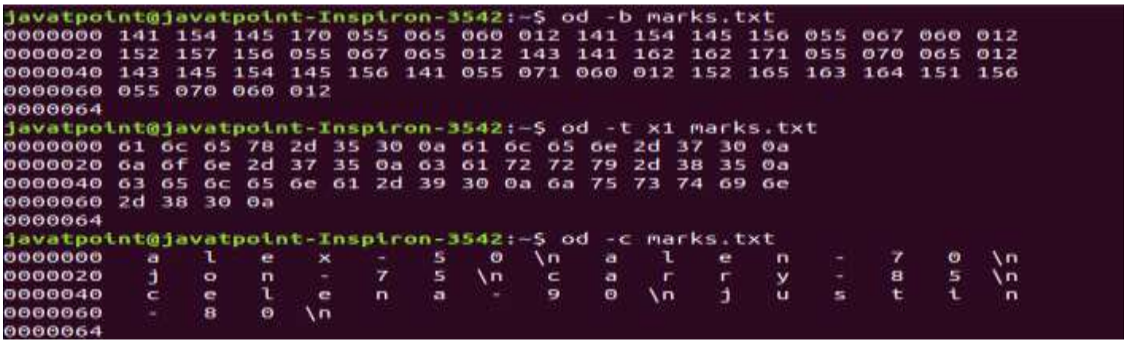
Syntax:

od -b **<fileName>**   // Octal format

od -t x1 **<fileName>**   // Hexa decimal format

od -c **<fileName>**   // ASCII character format

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ od -b marks.txt
0000000 141 154 145 170 055 065 060 012 141 154 145 156 055 067 060 012
0000020 152 157 156 055 067 065 012 143 141 162 162 171 055 070 065 012
0000040 143 145 154 145 156 141 055 071 060 012 152 165 163 164 151 156
0000060 055 070 060 012
0000064
javatpoint@javatpoint-Inspiron-3542:~$ od -t x1 marks.txt
0000000 61 6c 65 78 2d 35 30 0a 61 6c 65 6e 2d 37 30 0a
0000020 6a 6f 6e 2d 37 35 0a 63 61 72 72 79 2d 38 35 0a
0000040 63 65 6c 65 6e 61 2d 39 30 0a 6a 75 73 74 69 6e
0000060 2d 38 30 0a
0000064
javatpoint@javatpoint-Inspiron-3542:~$ od -c marks.txt
0000000   a   l   e   x   -   5   0  \n   a   l   e   n   -   7   0  \n
0000020   j   o   n   -   7   5  \n   c   a   r   r   y   -   8   5  \n
0000040   c   e   l   e   n   a   -   9   0  \n   j   u   s   t   i   n
0000060   -   8   0  \n
0000064
```

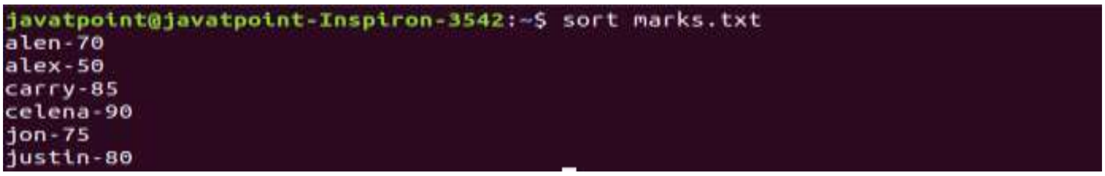
32. sort Command

The sort command is used to sort files in alphabetical order.

Syntax:

sort <file name>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sort marks.txt
alen-70
alex-50
carry-85
celena-90
jon-75
justin-80
```

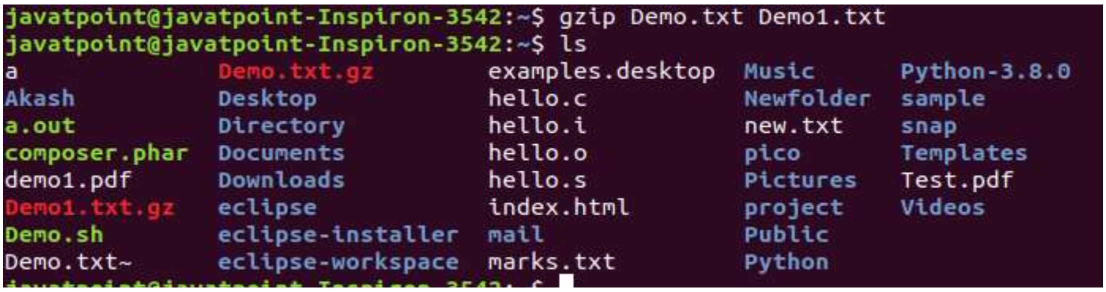
33. gzip Command

The gzip command is used to truncate the file size. It is a compressing tool. It replaces the original file by the compressed file having '.gz' extension.

Syntax:

gzip <file1> <file2> <file3>...

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ gzip Demo.txt Demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a              Demo.txt.gz        examples.desktop  Music      Python-3.8.0
Akash          Desktop            hello.c           Newfolder  sample
a.out          Directory          hello.i           new.txt    snap
composer.phar  Documents          hello.o           pico       Templates
demo1.pdf      Downloads          hello.s           Pictures   Test.pdf
Demo1.txt.gz   eclipse            index.html        project    Videos
Demo.sh        eclipse-installer  mail              Public
Demo.txt~      eclipse-workspace  marks.txt         Python
javatpoint@javatpoint-Inspiron-3542:~$
```

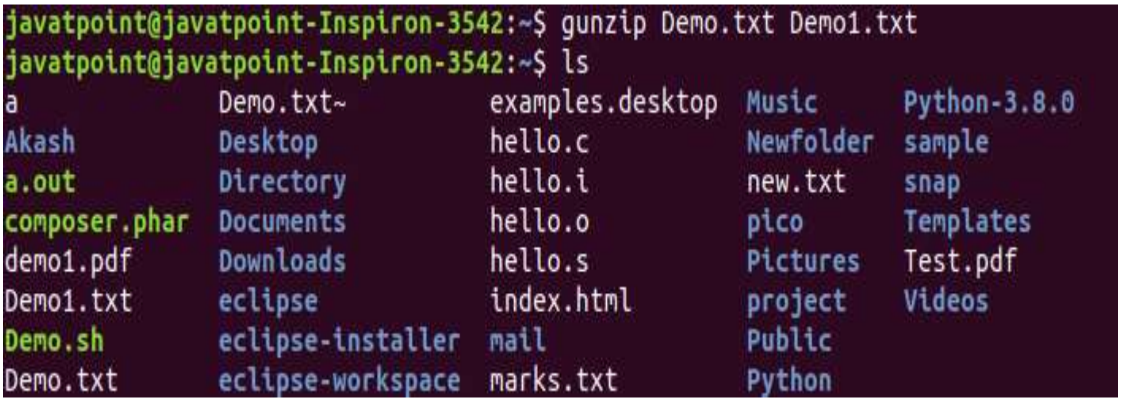
34. gunzip Command

The gunzip command is used to decompress a file. It is a reverse operation of gzip command.

Syntax:

gunzip <file1> <file2> <file3>. .

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ gunzip Demo.txt Demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a              Demo.txt~          examples.desktop  Music      Python-3.8.0
Akash          Desktop            hello.c           Newfolder  sample
a.out          Directory          hello.i           new.txt    snap
composer.phar  Documents          hello.o           pico       Templates
demo1.pdf      Downloads          hello.s           Pictures   Test.pdf
Demo1.txt      eclipse            index.html        project    Videos
Demo.sh        eclipse-installer  mail              Public
Demo.txt       eclipse-workspace  marks.txt         Python
```

Linux Utility Commands

35. find Command

The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date, and more.

The following symbols are used after the find command:

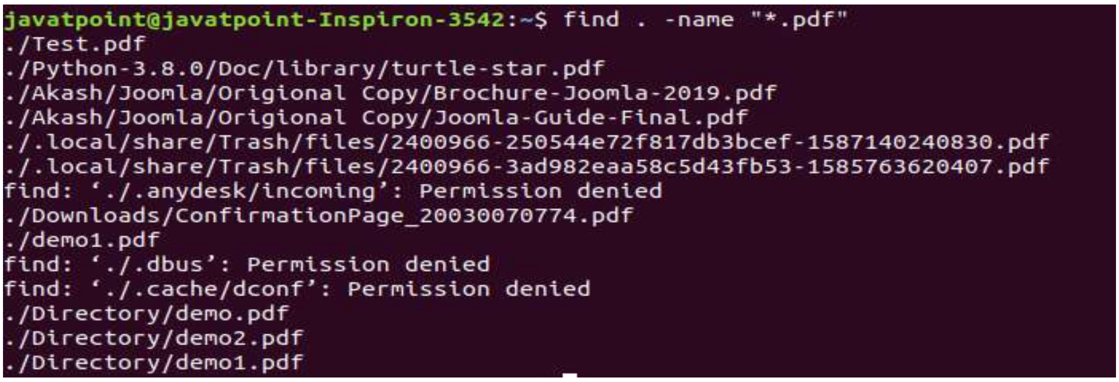
(.) : For current directory name

(/) : For root

Syntax:

```
find . -name "*.pdf"
```

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ find . -name "*.pdf"
./Test.pdf
./Python-3.8.0/Doc/library/turtle-star.pdf
./Akash/Joomla/Origional Copy/Brochure-Joomla-2019.pdf
./Akash/Joomla/Origional Copy/Joomla-Guide-Final.pdf
./.local/share/Trash/files/2400966-250544e72f817db3bcef-1587140240830.pdf
./.local/share/Trash/files/2400966-3ad982eaa58c5d43fb53-1585763620407.pdf
find: './.anydesk/incoming': Permission denied
./Downloads/ConfirmationPage_20030070774.pdf
./demo1.pdf
find: './.dbus': Permission denied
find: './.cache/dconf': Permission denied
./Directory/demo.pdf
./Directory/demo2.pdf
./Directory/demo1.pdf
```

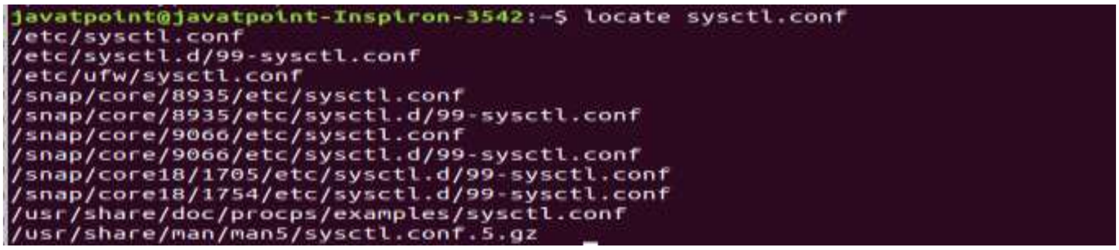
36. locate Command

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.

Syntax:

```
locate <file name>
```

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ locate sysctl.conf
/etc/sysctl.conf
/etc/sysctl.d/99-sysctl.conf
/etc/ufw/sysctl.conf
/snap/core/8935/etc/sysctl.conf
/snap/core/8935/etc/sysctl.d/99-sysctl.conf
/snap/core/9066/etc/sysctl.conf
/snap/core/9066/etc/sysctl.d/99-sysctl.conf
/snap/core18/1705/etc/sysctl.d/99-sysctl.conf
/snap/core18/1754/etc/sysctl.d/99-sysctl.conf
/usr/share/doc/procps/examples/sysctl.conf
/usr/share/man/man5/sysctl.conf.5.gz
```

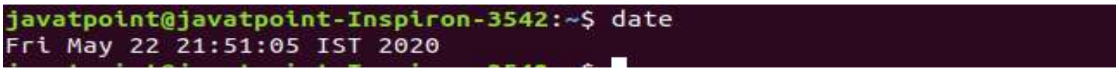
37. date Command

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

Syntax:

```
date
```

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ date
Fri May 22 21:51:05 IST 2020
```

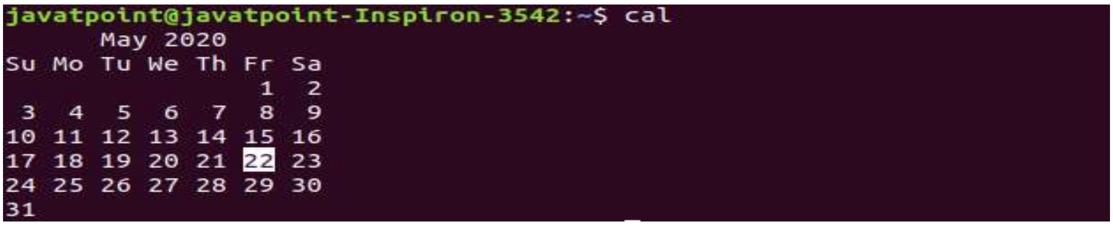
38. cal Command

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

Syntax:

cal<

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ cal
      May 2020
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
```

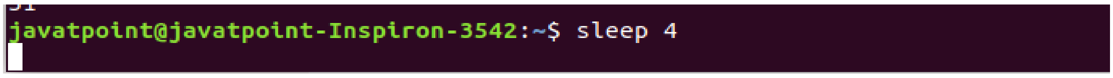
39. sleep Command

The sleep command is used to hold the terminal by the specified amount of time. By default, it takes time in seconds.

Syntax:

sleep **<time>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sleep 4
```

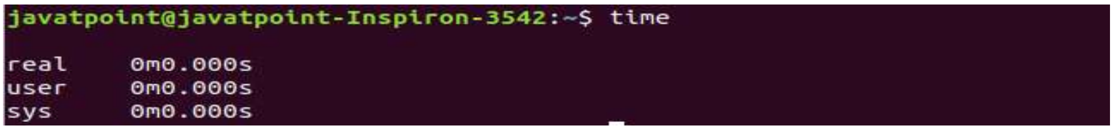
40. time Command

The time command is used to display the time to execute a command.

Syntax:

time

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
```

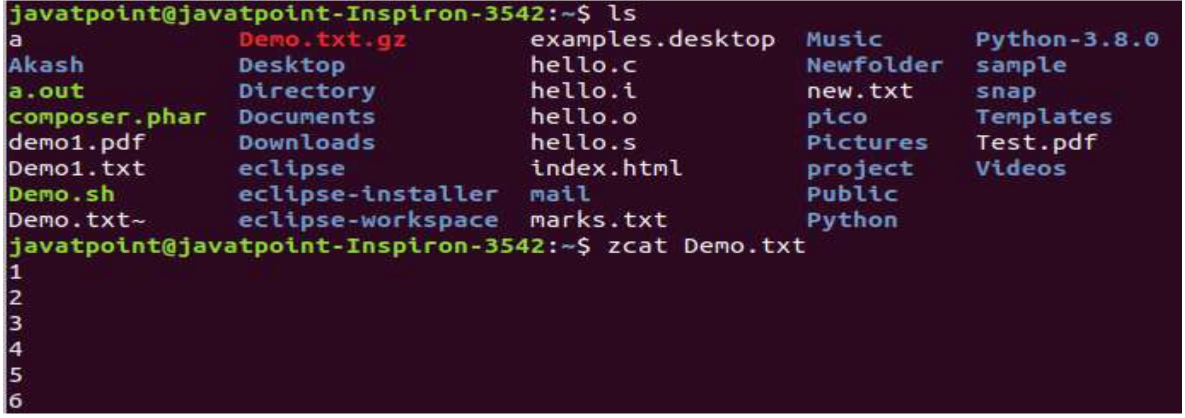
41. zcat Command

The zcat command is used to display the compressed files.

Syntax:

zcat **<file** name**>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ls
a               Demo.txt.gz        examples.desktop  Music      Python-3.8.0
Akash           Desktop            hello.c           Newfolder  sample
a.out           Directory          hello.i           new.txt    snap
composer.phar   Documents          hello.o           pico       Templates
demo1.pdf       Downloads          hello.s           Pictures   Test.pdf
Demo1.txt       eclipse            index.html        project    Videos
Demo.sh         eclipse-installer  mail              Public
Demo.txt~       eclipse-workspace  marks.txt         Python
javatpoint@javatpoint-Inspiron-3542:~$ zcat Demo.txt
1
2
3
4
5
6
```

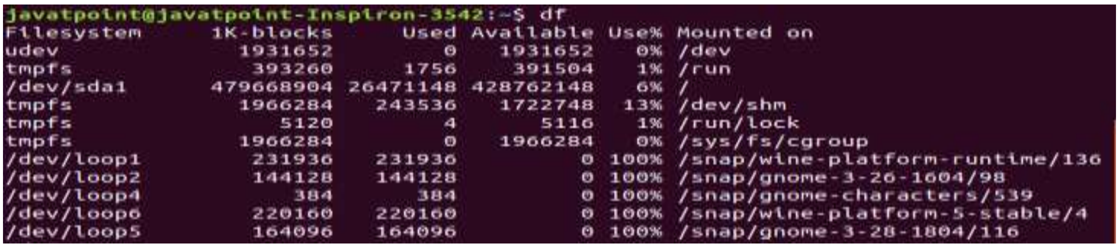
42. df Command

The df command is used to display the disk space used in the file system. It displays the output as in the number of used blocks, available blocks, and the mounted directory.

Syntax:

df

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1931652        0   1931652   0% /dev
tmpfs             393260     1756    391504   1% /run
/dev/sda1      479668904 26471148 428762148   6% /
tmpfs            1966284   243536   1722748  13% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            1966284        0   1966284   0% /sys/fs/cgroup
/dev/loop1        231936   231936         0 100% /snap/wine-platform-runtime/136
/dev/loop2        144128   144128         0 100% /snap/gnome-3-26-1604/98
/dev/loop4           384      384         0 100% /snap/gnome-characters/539
/dev/loop6        220160   220160         0 100% /snap/wine-platform-5-stable/4
/dev/loop5        164096   164096         0 100% /snap/gnome-3-28-1804/116
```

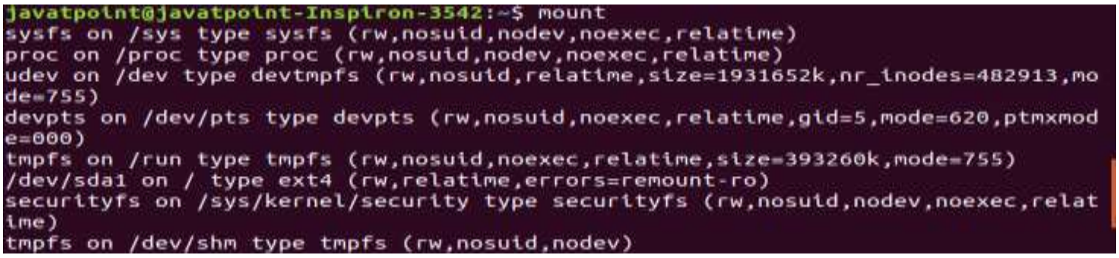
43. mount Command

The mount command is used to connect an external device file system to the system's file system.

Syntax:

mount -t type **<device> <directory>**

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1931652k,nr_inodes=482913,mo
de=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmod
e=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=393260k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relat
ime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```

44. exit Command

Linux exit command is used to exit from the current shell. It takes a parameter as a number and exits the shell with a return of status number.

Syntax:

exit

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ exit
```

After pressing the ENTER key, it will exit the terminal.

45. clear Command

Linux **clear** command is used to clear the terminal screen.

Syntax:

clear

Linux Networking Commands

46. ip Command

Linux ip command is an updated version of the ipconfig command. It is used to assign an IP address, initialize an interface, disable an interface.

Syntax:

ip a or ip addr

47. ssh Command

Linux ssh command is used to create a remote connection through the ssh protocol.

Syntax:

ssh user_name@host(IP/Domain_name)</p>

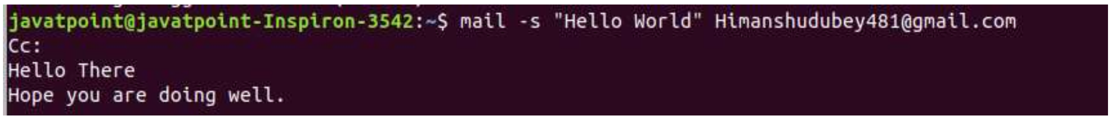
48. mail Command

The mail command is used to send emails from the command line.

Syntax:

mail -s "Subject" <recipient address>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ mail -s "Hello World" Himanshudubey481@gmail.com
Cc:
Hello There
Hope you are doing well.
```

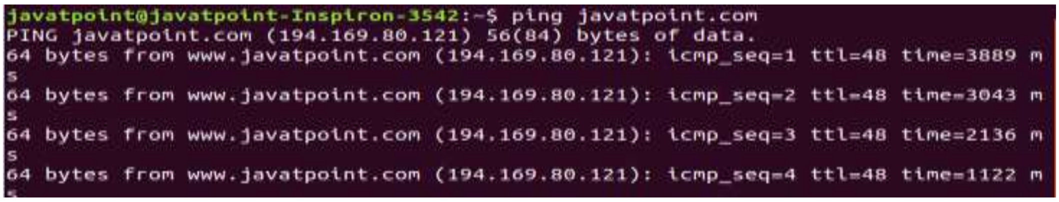
49. ping Command

The ping command is used to check the connectivity between two nodes, that is whether the server is connected. It is a short form of "Packet Internet Groper."

Syntax:

ping <destination>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ping javatpoint.com
PING javatpoint.com (194.169.80.121) 56(84) bytes of data.
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=1 ttl=48 time=3889 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=2 ttl=48 time=3043 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=3 ttl=48 time=2136 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=4 ttl=48 time=1122 m
```

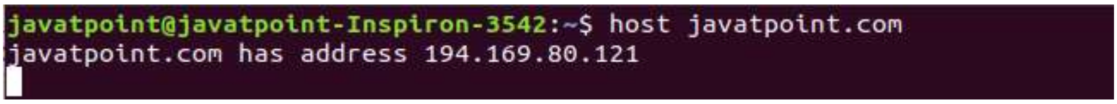
50. host Command

The host command is used to display the IP address for a given domain name and vice versa. It performs the DNS lookups for the DNS Query.

Syntax:

host <domain name> or <ip address>

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ host javatpoint.com
javatpoint.com has address 194.169.80.121
```

gcc command in Linux

GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language. It can also be used to compile Objective C and Objective C++. The most important option required while compiling a source code file is the name of the source program, rest every argument is optional like a warning, debugging, linking libraries, object file etc. The different options of *gcc* command allow the user to stop the compilation process at different stages.

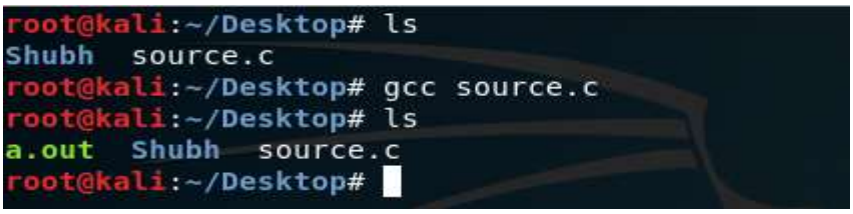
Syntax:

gcc [-c|-S|-E] [-std=standard]

Example:

This will compile the *source.c* file and give the output file as *a.out* file which is default name of output file given by gcc compiler, which can be executed using **./a.out**

gcc source.c



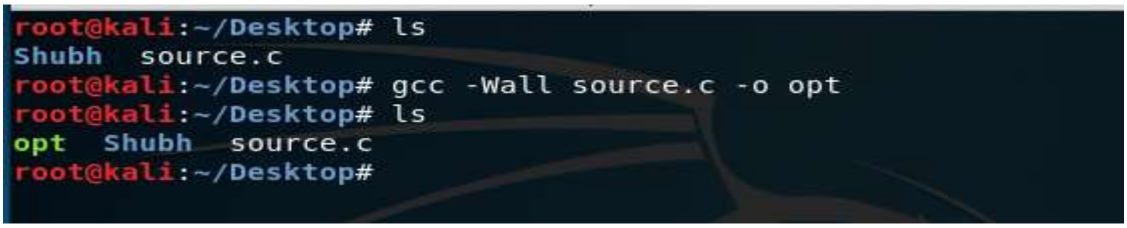
```
root@kali:~/Desktop# ls
Shubh   source.c
root@kali:~/Desktop# gcc source.c
root@kali:~/Desktop# ls
a.out   Shubh   source.c
root@kali:~/Desktop# 
```

Most Useful Options with Examples: Here *source.c* is the C program code file.

- **-o opt:**

  This will compile the source.c file but instead of giving default name hence executed using **./opt**, it will give output file as opt. *-o* is for output file option.

  gcc source.c -o opt

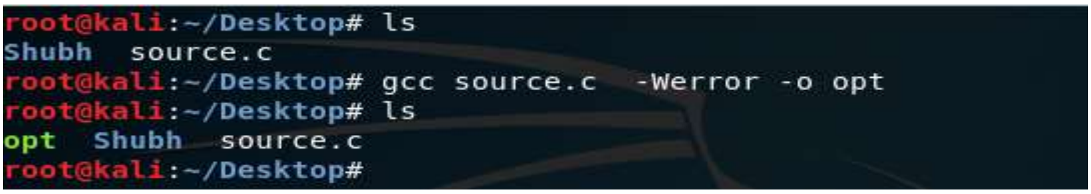


```
root@kali:~/Desktop# ls
Shubh   source.c
root@kali:~/Desktop# gcc -Wall source.c -o opt
root@kali:~/Desktop# ls
opt   Shubh   source.c
root@kali:~/Desktop#
```

- **-Werror:**

  This will compile the source and show the warning if any error is there in the program, -W is for giving warnings.

  gcc source.c -Werror -o opt

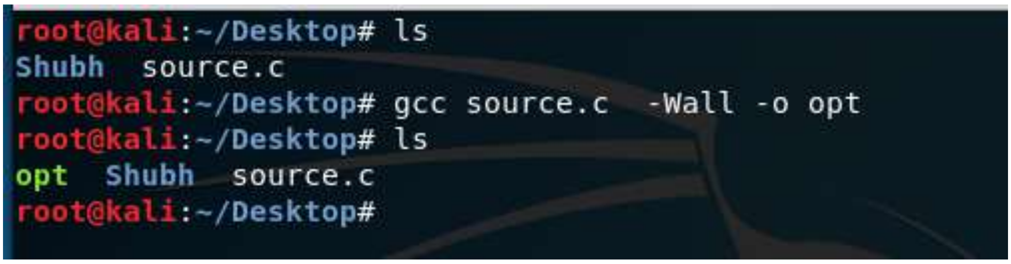


```
root@kali:~/Desktop# ls
Shubh   source.c
root@kali:~/Desktop# gcc source.c  -Werror -o opt
root@kali:~/Desktop# ls
opt   Shubh   source.c
root@kali:~/Desktop#
```

- **-Wall:**

  This will check not only for errors but also for all kinds warning like unused variables errors, it is good practice to use this flag while compiling the code.

  gcc source.c -Wall -o opt

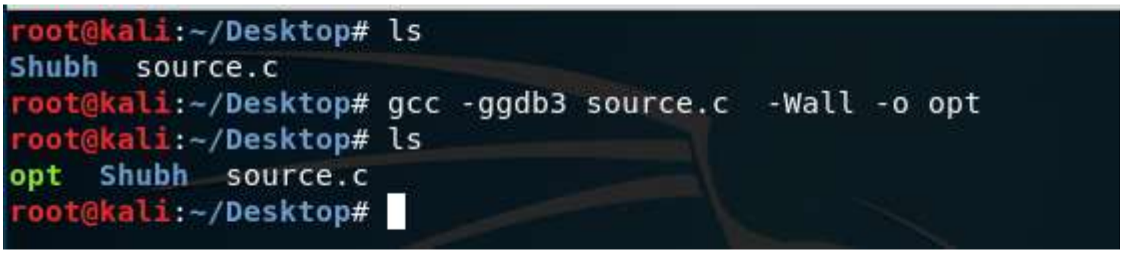


```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc source.c  -Wall -o opt
root@kali:~/Desktop# ls
opt  Shubh  source.c
root@kali:~/Desktop#
```

- **-ggdb3:**

  This command give us permissions to debug the program using gdb which will be described later, *-g* option is for debugging.

  gcc -ggdb3 source.c -Wall -o opt

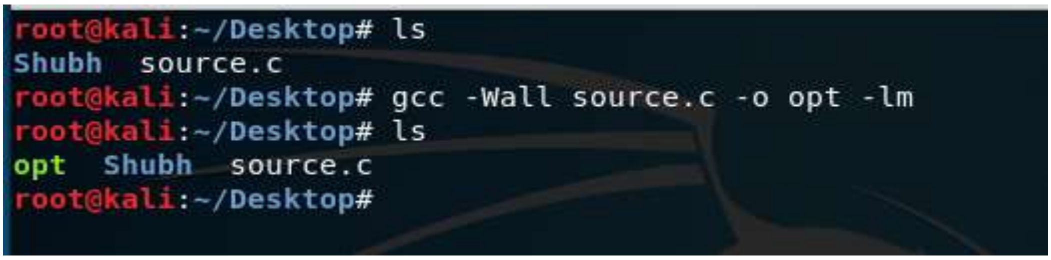


```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc -ggdb3 source.c  -Wall -o opt
root@kali:~/Desktop# ls
opt  Shubh  source.c
root@kali:~/Desktop# █
```

- **-lm :**

  This command link *math.h* library to our source file, -l option is used for linking particular library, for math.h we use -lm.

  gcc -Wall source.c -o opt -lm

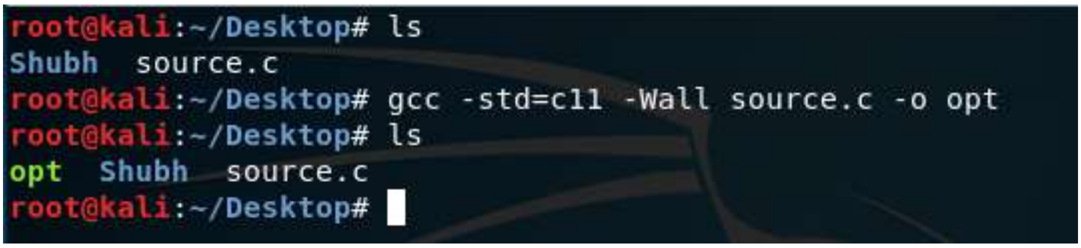


```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc -Wall source.c -o opt -lm
root@kali:~/Desktop# ls
opt  Shubh  source.c
root@kali:~/Desktop#
```

- **-std=c11 :**

  This command will use the c11 version of standards for compiling the *source.c* program, which allows to define variable under loop initializations also using newer standards version is preferred.

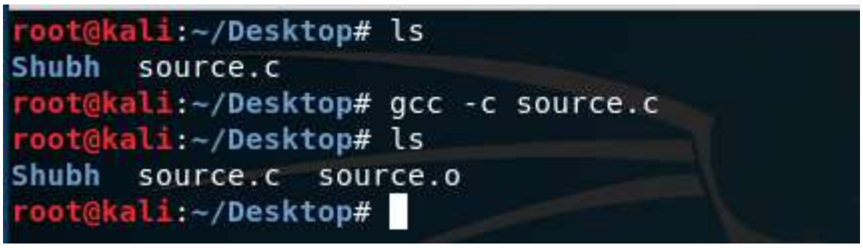
  gcc -Wall -std=c11 source.c -o opt



```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc -std=c11 -Wall source.c -o opt
root@kali:~/Desktop# ls
opt  Shubh  source.c
root@kali:~/Desktop# █
```

- **-c :**

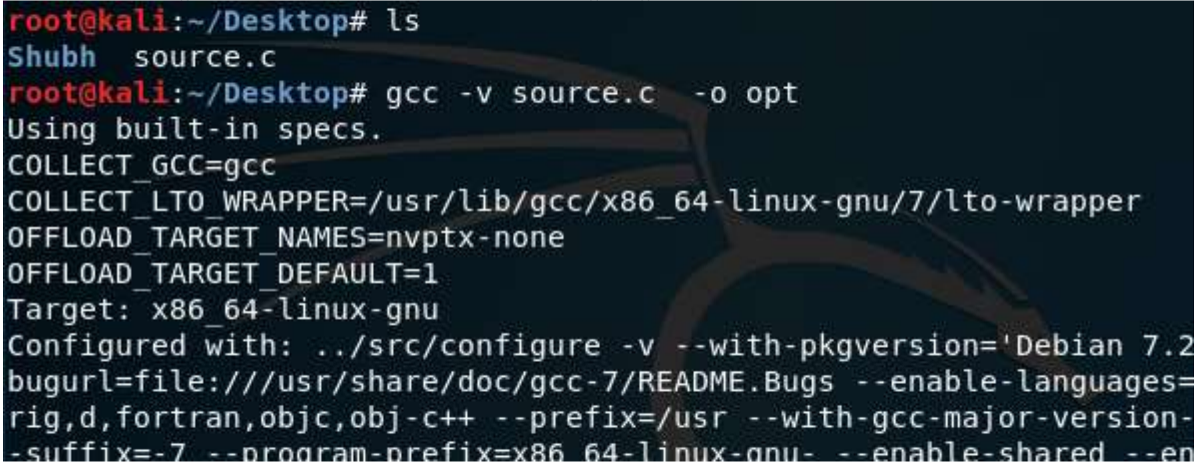
  This command compile the program and give the object file as output, which is used to make libraries.



```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc -c source.c
root@kali:~/Desktop# ls
Shubh  source.c  source.o
root@kali:~/Desktop# 
```

- **-v :**

  This option is used for the verbose purpose.



```
root@kali:~/Desktop# ls
Shubh  source.c
root@kali:~/Desktop# gcc -v source.c  -o opt
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Debian 7.2
bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=
rig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-
-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --en
```