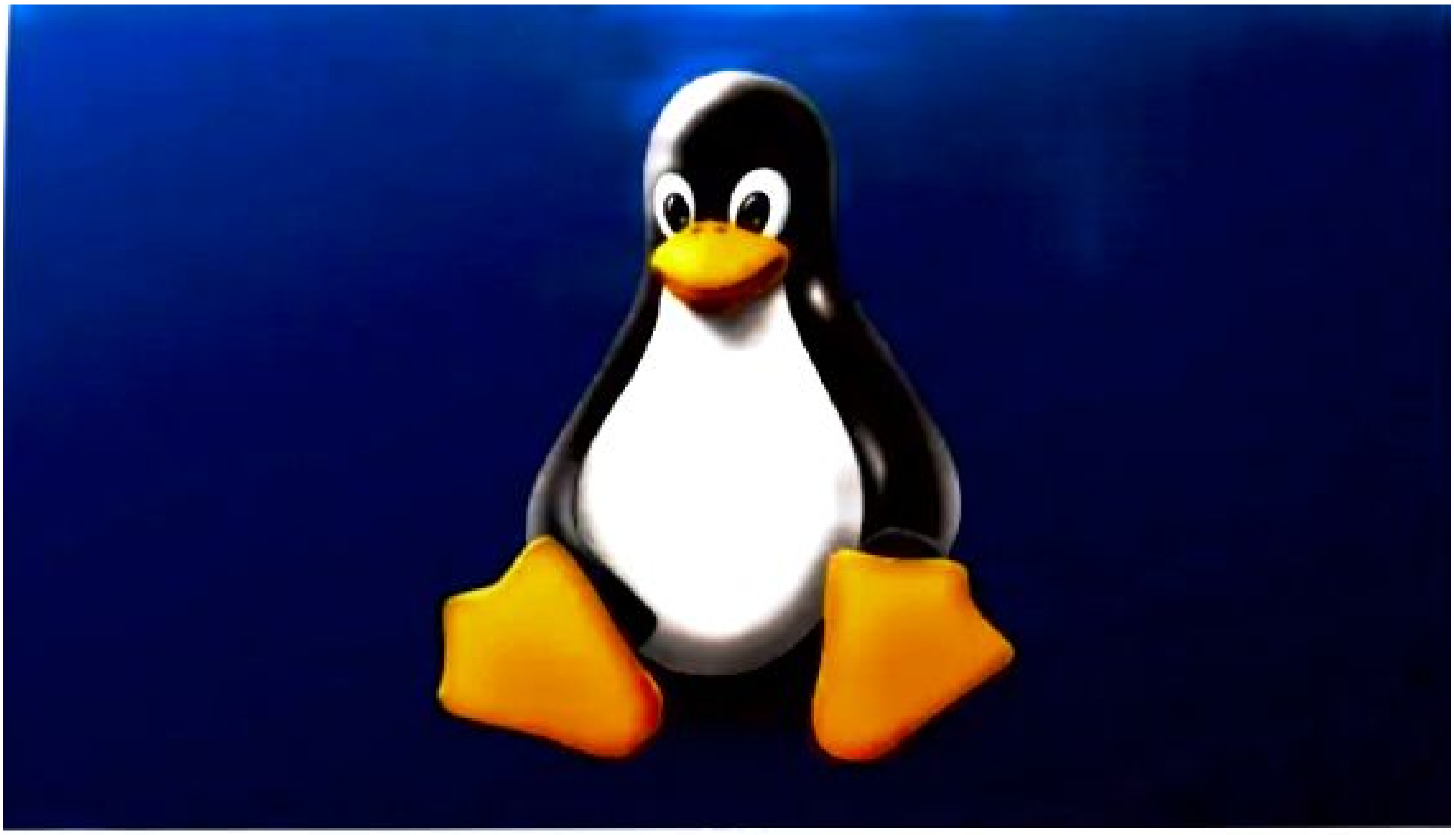
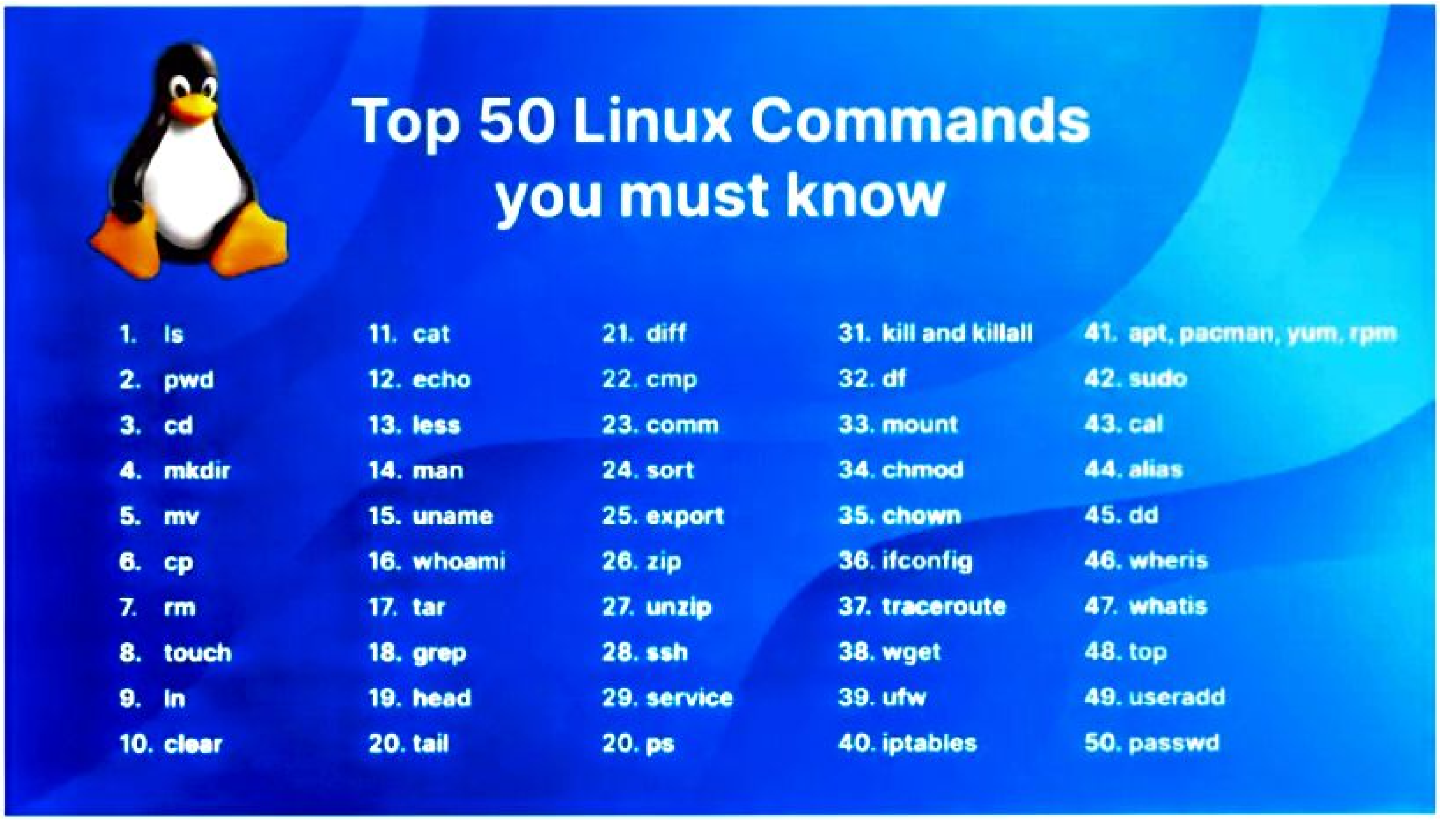




Top 50 Linux Commands you must know

1. ls
2. pwd
3. cd
4. mkdir
5. mv
6. cp
7. rm
8. touch
9. ln
10. clear
11. cat
12. echo
13. less
14. man
15. uname
16. whoami
17. tar
18. grep
19. head
20. tail
21. diff
22. cmp
23. comm
24. sort
25. export
26. zip
27. unzip
28. ssh
29. service
20. ps
31. kill and killall
32. df
33. mount
34. chmod
35. chown
36. ifconfig
37. traceroute
38. wget
39. ufw
40. iptables
41. apt, pacman, yum, rpm
42. sudo
43. cal
44. alias
45. dd
46. wheris
47. whatis
48. top
49. useradd
50. passwd

Ques) What is Linux.

Ans) **Full Form** of LINUX is **Lovable Intellect Not Using XP**. Linux was built by and named after Linus Torvalds. Linux is an open-source operating system for servers, computers, mainframes, mobile systems, and embedded systems. Requests from device software are handled by Linux and relayed to computer hardware.

Step 1: Open Terminal

Step 1: Open Terminal

Linux Commands with Examples

The Linux command is a utility of the Linux operating system. All basic and advanced tasks can be done by executing commands. The commands are executed on the **Linux terminal**. The terminal is a command-line interface to interact with the system, which is similar to the command prompt in the Windows OS. *Commands in Linux are* **case-sensitive**.

Linux terminal is a user-friendly terminal as it provides various support options. To open the Linux terminal, press "CTRL + ALT + T" keys together, and execute a command by pressing the 'ENTER' key.

Linux Directory Commands

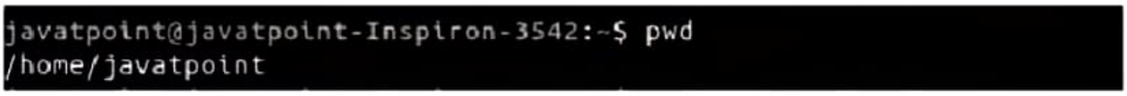
1. pwd Command

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

Syntax:

pwd

Output:



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

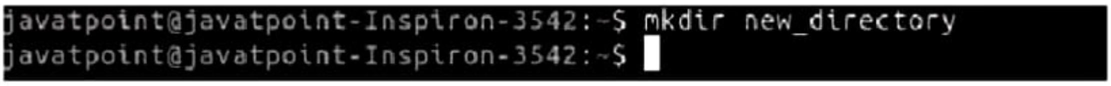
2. mkdir Command

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:~$
```

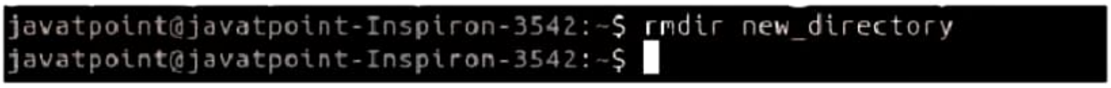
3. rmdir Command

The rmdir

command is used to delete a directory.

Syntax:

rmdir <**directory** name>



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

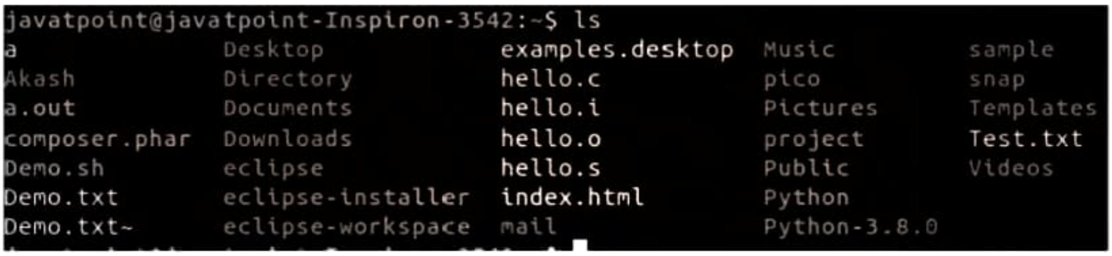
4. ls Command

The ls

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

Syntax:

ls



```
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

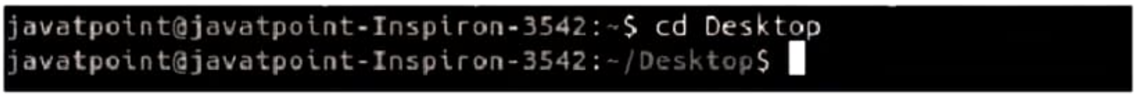
The cd

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

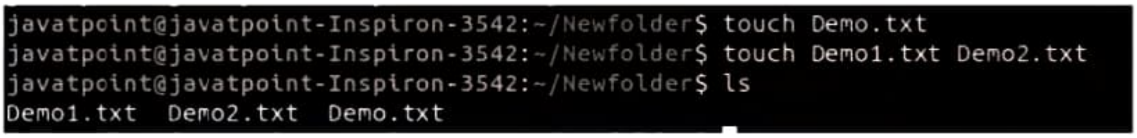
The touch

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

Syntax

1. touch <file name>
2. 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]..

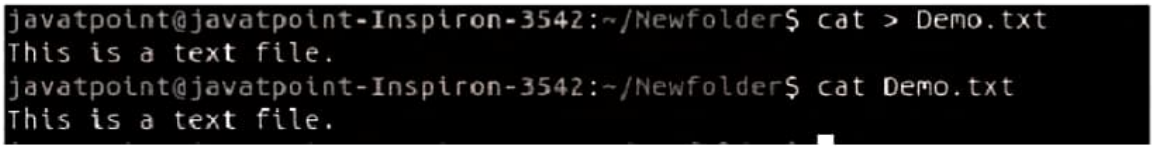
To create a file, execute it as follows

1. cat > <file name>
2. // Enter file content

Press "**CTRL+ D**" keys to save the file. To display the content of the file, execute it as follows:

1. 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

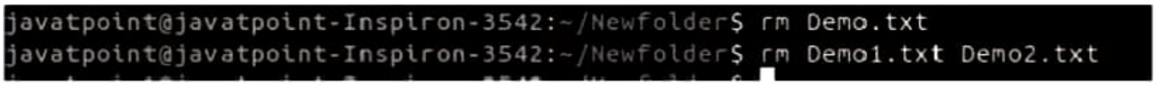
The rm

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.

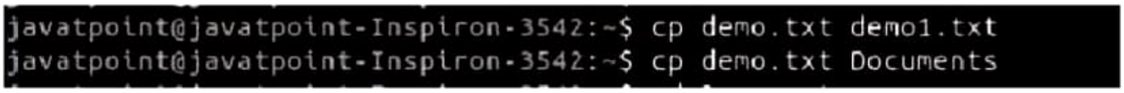
Syntax:

To copy in the same directory:

cp **<existing** file name> **<new** file name>

To copy in a different directory:

Output:



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

10. mv Command

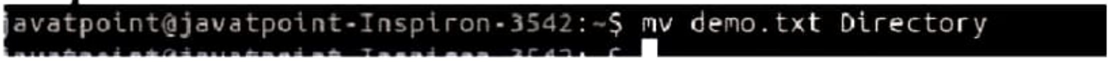
The mv

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.

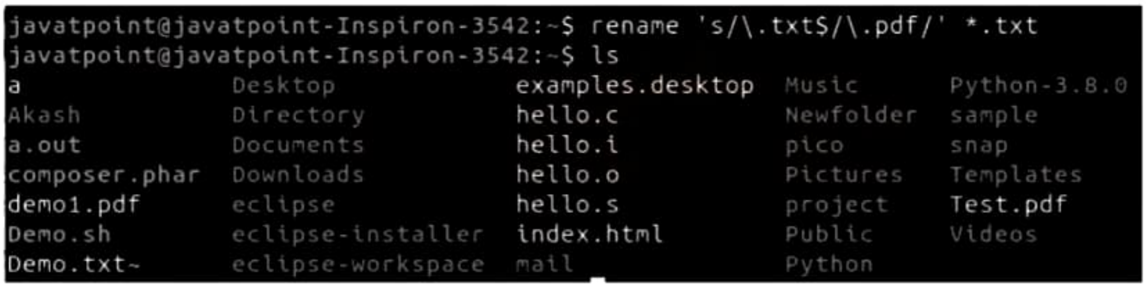
Syntax:

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
```

Linux File Content Commands

12. head Command

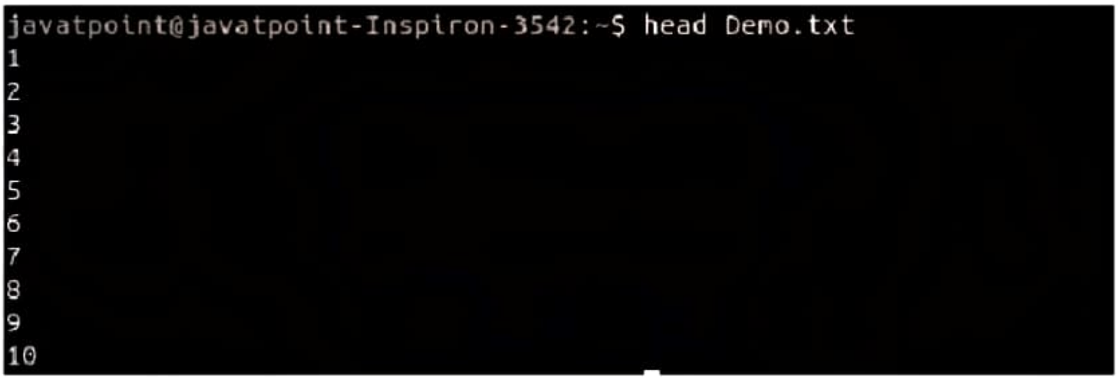
The head

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

Syntax:

head <file name>

Output:



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

13. tail Command

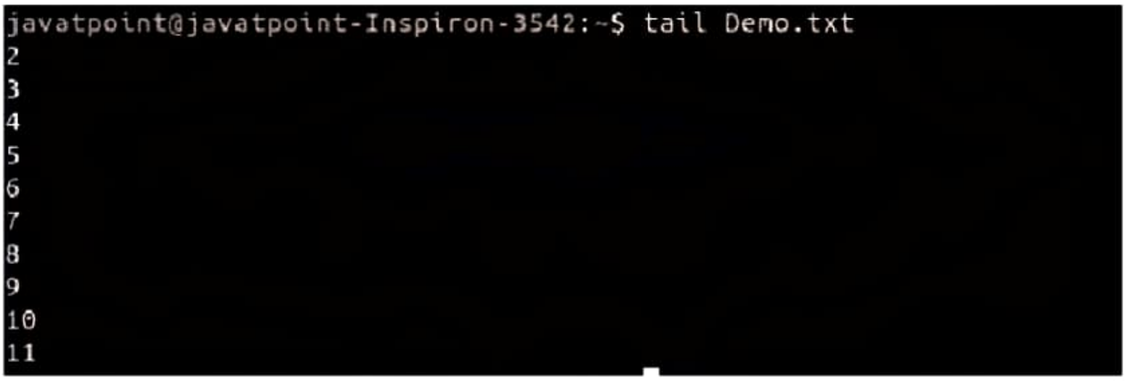
The tail

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>

Output:



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

14. tac Command

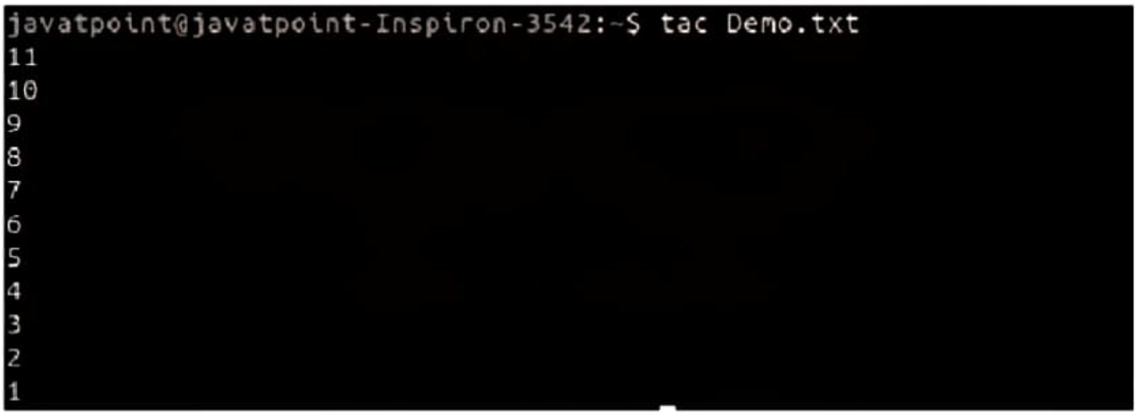
The tac

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.

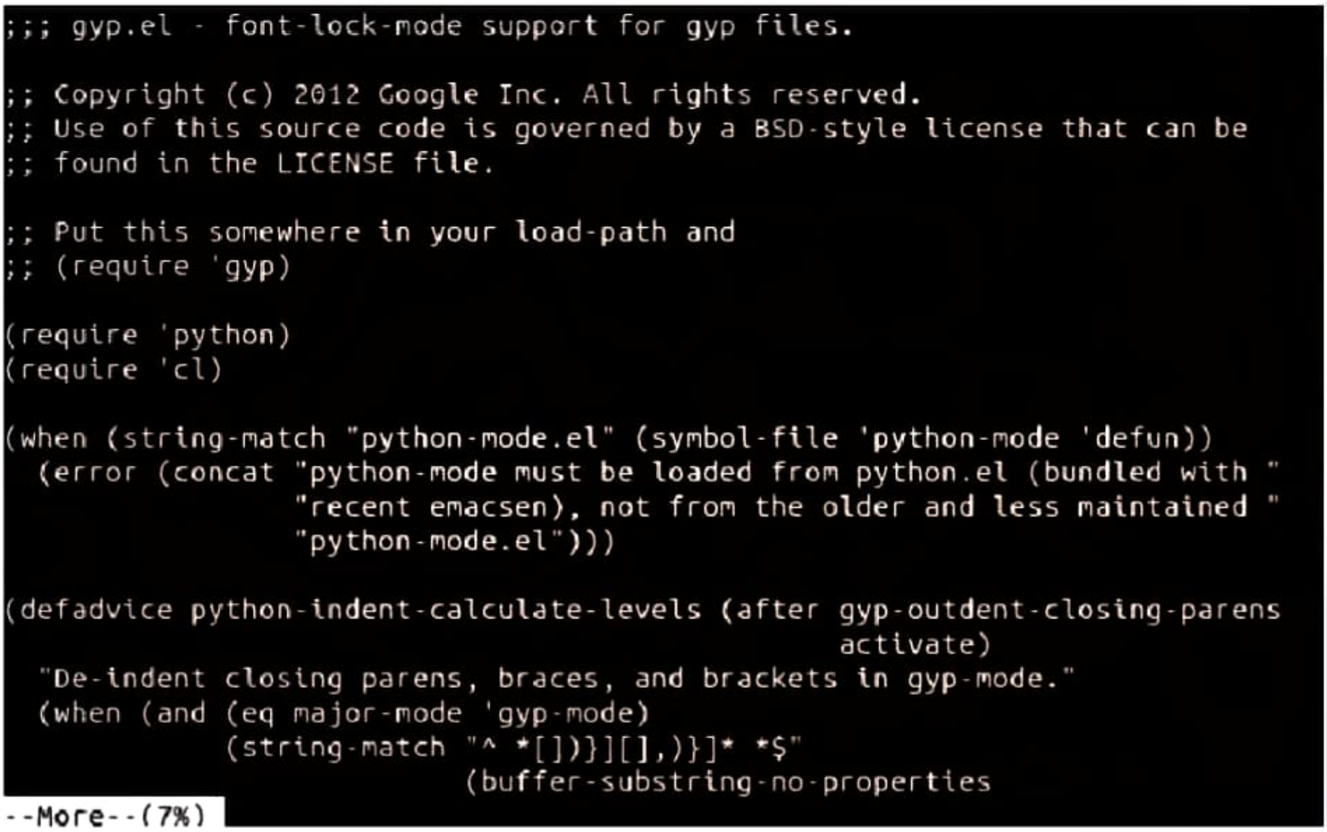
b key: To move to the previous page.

/ key: To search the string.

Syntax:

more <file name>

Output:



```
;;; gyp.el - font-lock-mode support for gyp files.

;; Copyright (c) 2012 Google Inc. All rights reserved.
;; Use of this source code is governed by a BSD-style license that can be
;; found in the LICENSE file.

;; Put this somewhere in your load-path and
;; (require 'gyp)

(require 'python)
(require 'cl)

(when (string-match "python-mode.el" (symbol-file 'python-mode 'defun))
  (error (concat "python-mode must be loaded from python.el (bundled with "
                 "recent emacsen), not from the older and less maintained "
                 "python-mode.el")))

(defadvice python-indent-calculate-levels (after gyp-outdent-closing-parens
                                                  activate)
  "De-indent closing parens, braces, and brackets in gyp-mode."
  (when (and (eq major-mode 'gyp-mode)
             (string-match "^ *[])}][],)}]* *$"
                           (buffer-substring-no-properties
--More--(7%)
```

16. less Command

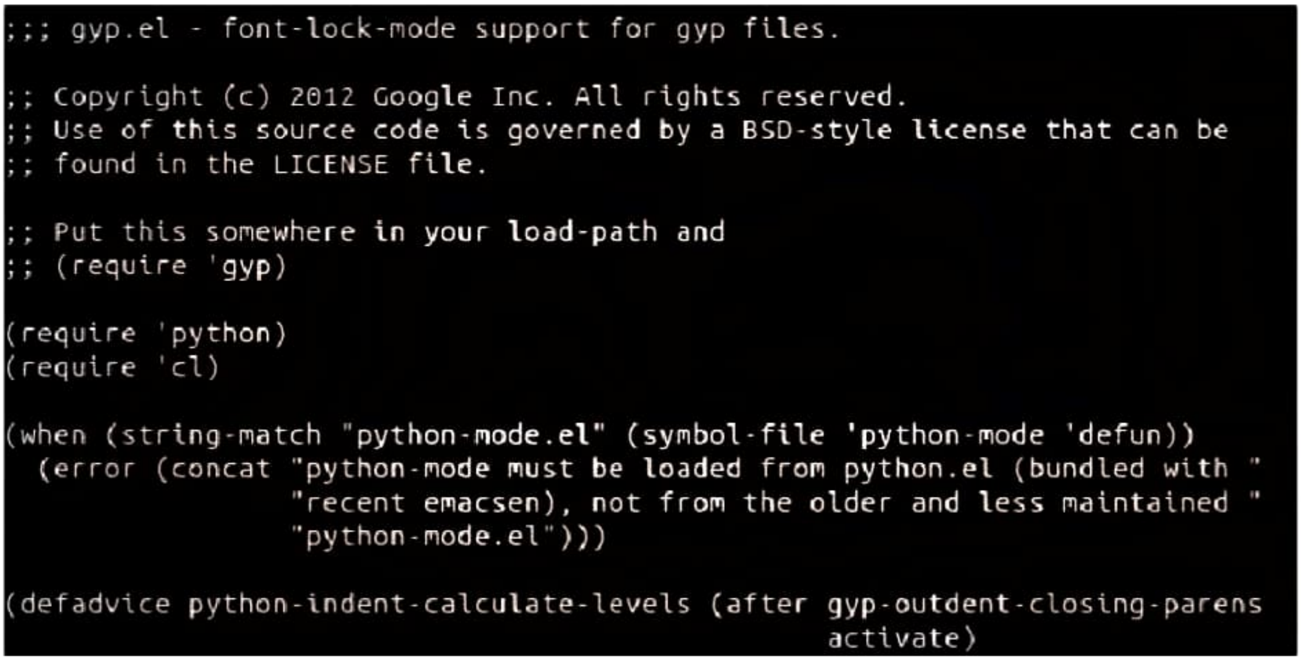
The less

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>

Output:



```
;;; gyp.el - font-lock-mode support for gyp files.

;; Copyright (c) 2012 Google Inc. All rights reserved.
;; Use of this source code is governed by a BSD-style license that can be
;; found in the LICENSE file.

;; Put this somewhere in your load-path and
;; (require 'gyp)

(require 'python)
(require 'cl)

(when (string-match "python-mode.el" (symbol-file 'python-mode 'defun))
  (error (concat "python-mode must be loaded from python.el (bundled with "
                 "recent emacsen), not from the older and less maintained "
                 "python-mode.el")))

(defadvice python-indent-calculate-levels (after gyp-outdent-closing-parens
                                                 activate)
```

Linux User Commands

17. su Command

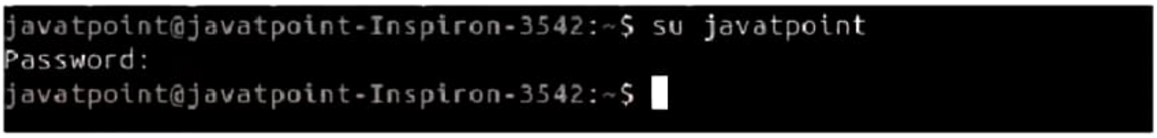
The su

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

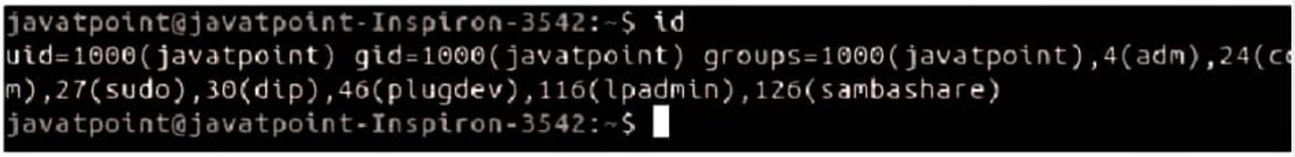
The id

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(c
m),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
javatpoint@javatpoint-Inspiron-3542:~$ 
```

19. useradd Command

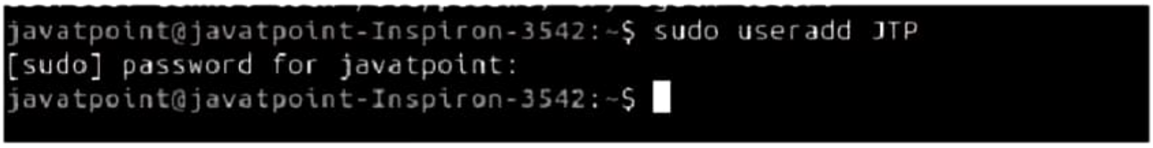
The useradd

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

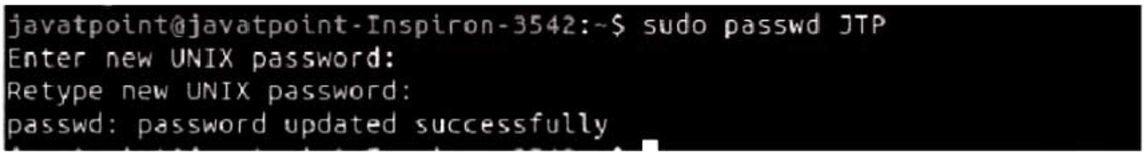
The passwd

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

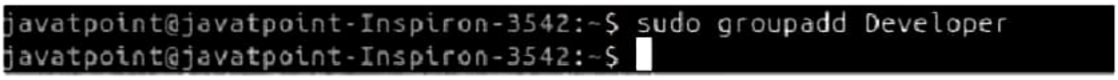
The groupadd

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

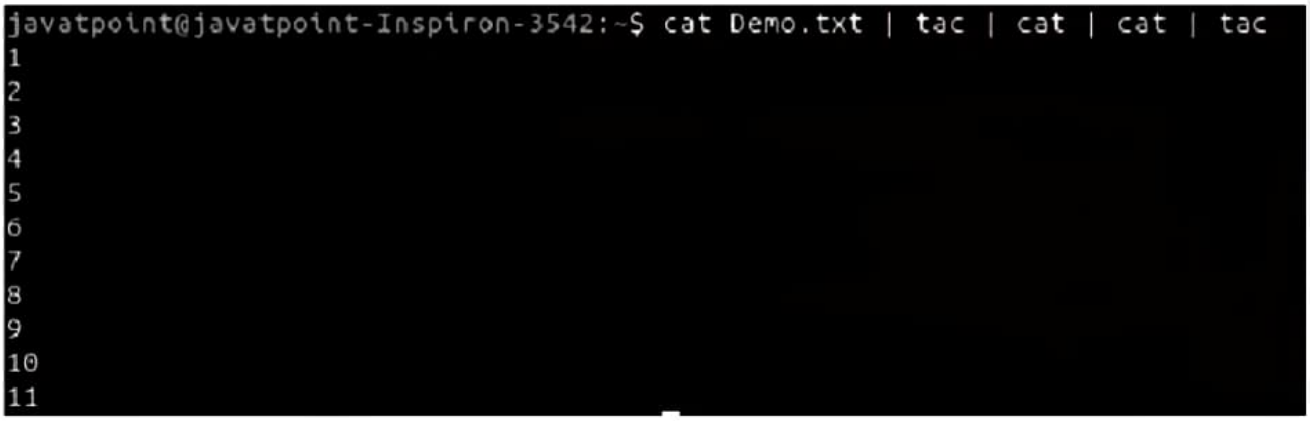
The cat

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

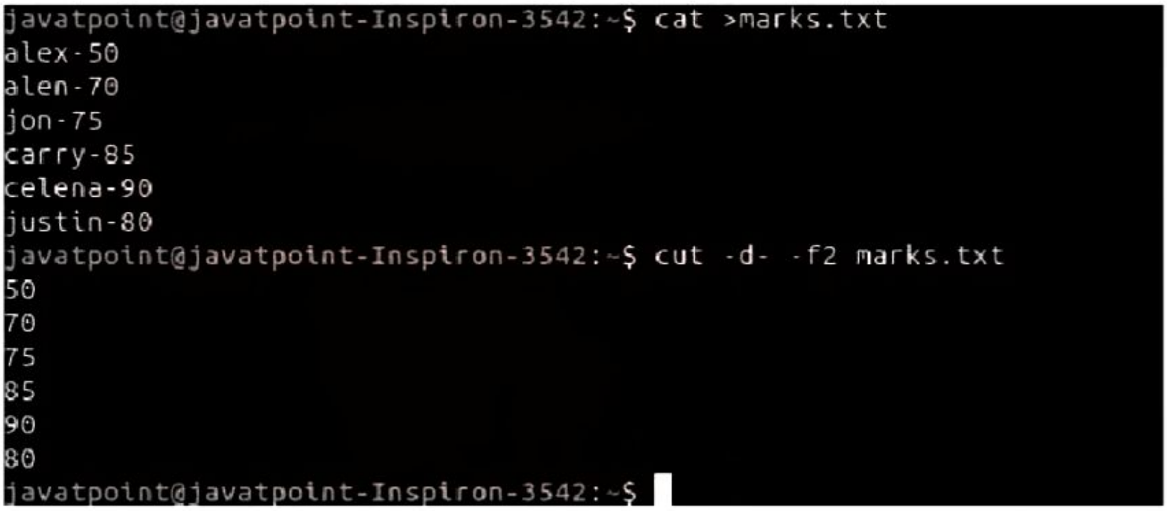
The cut

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

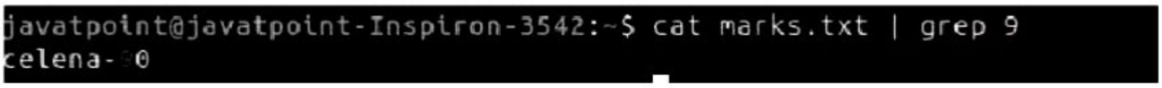
The grep

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

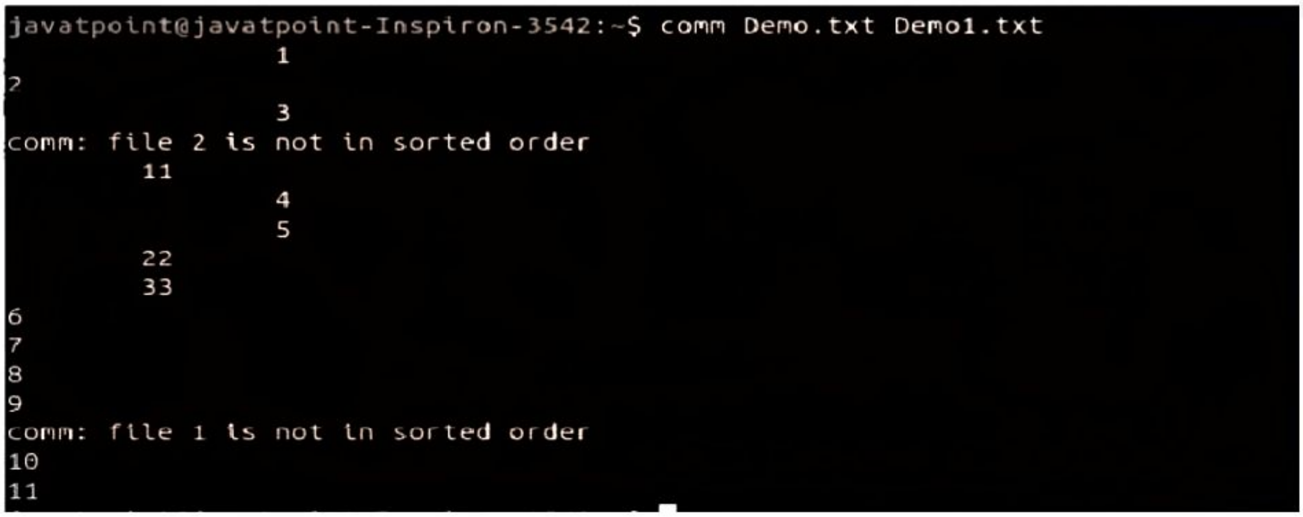
The 'comm'

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

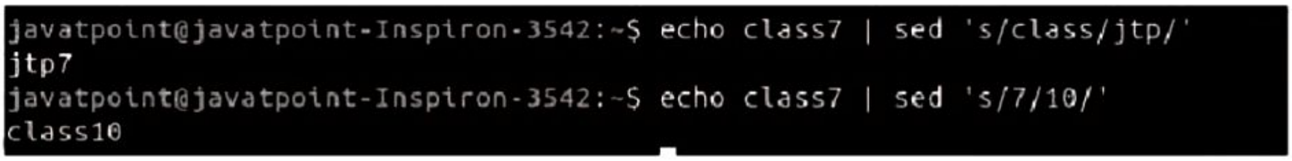
The sed

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

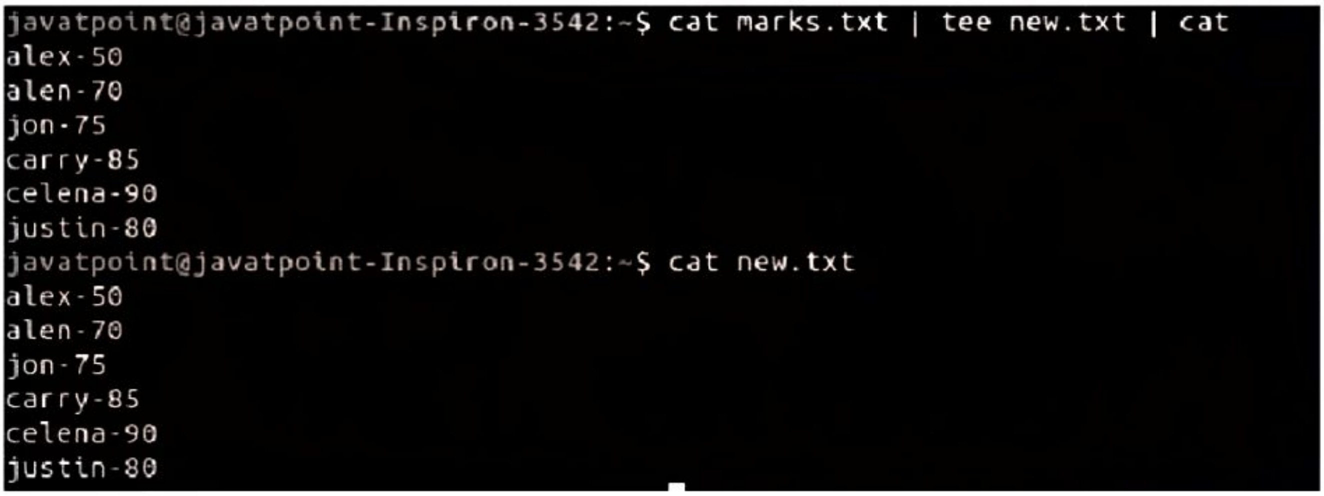
The tee

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

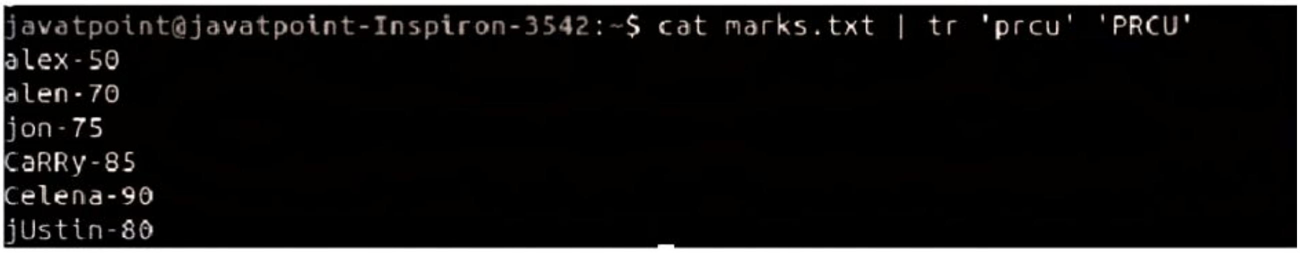
The tr

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

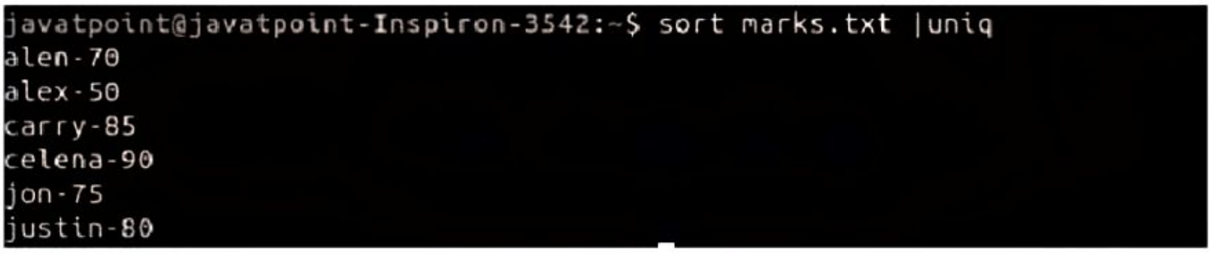
The uniq

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

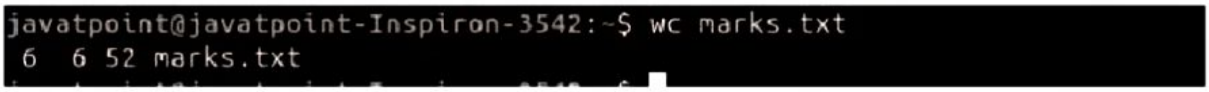
The wc

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.

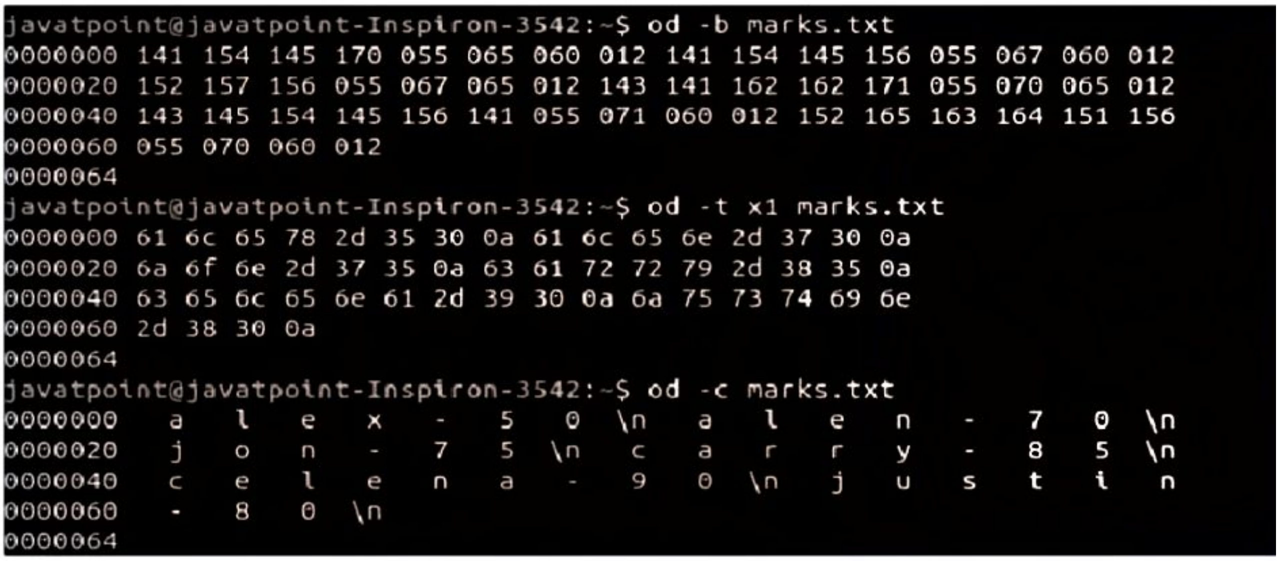
Syntax:

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

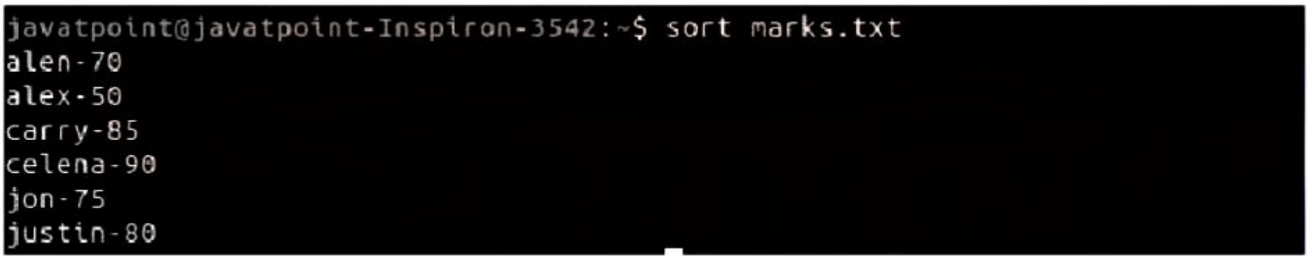
The sort

command is used to sort files in alphabetical order.

Syntax:

1. 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

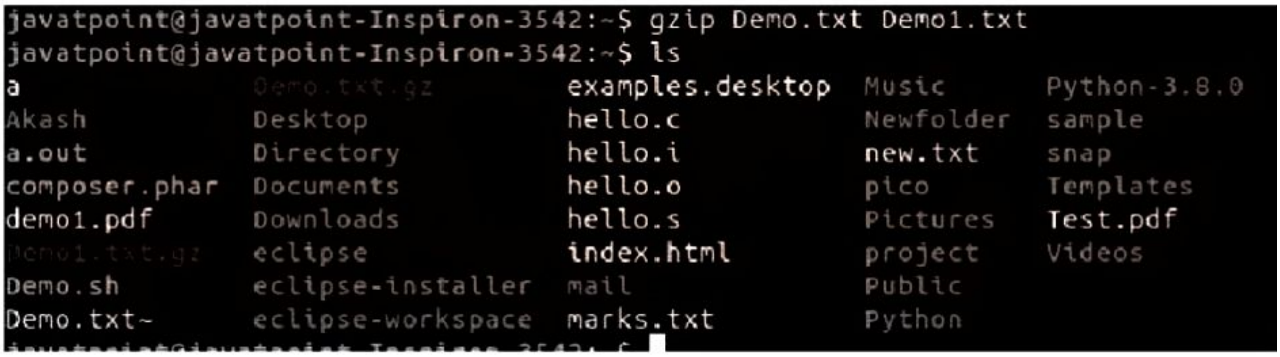
The gzip

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:

1. 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
```

34. gunzip Command

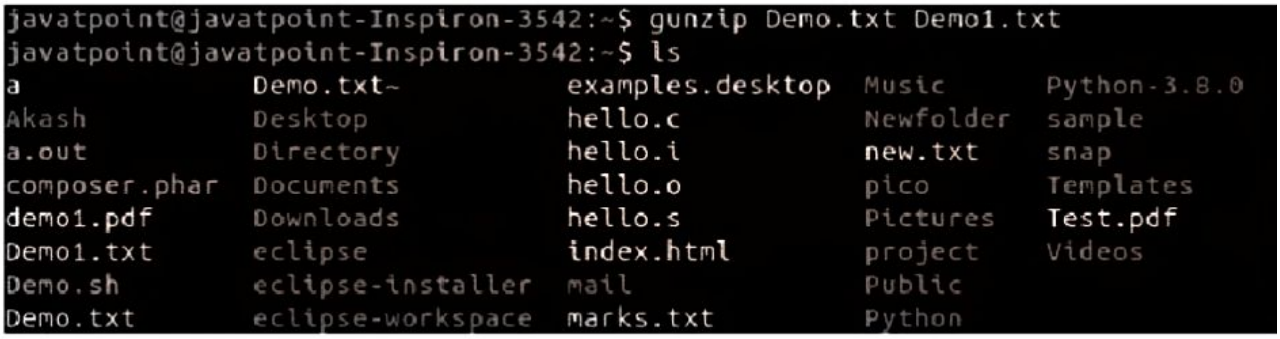
The gunzip

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

Syntax:

1. 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:

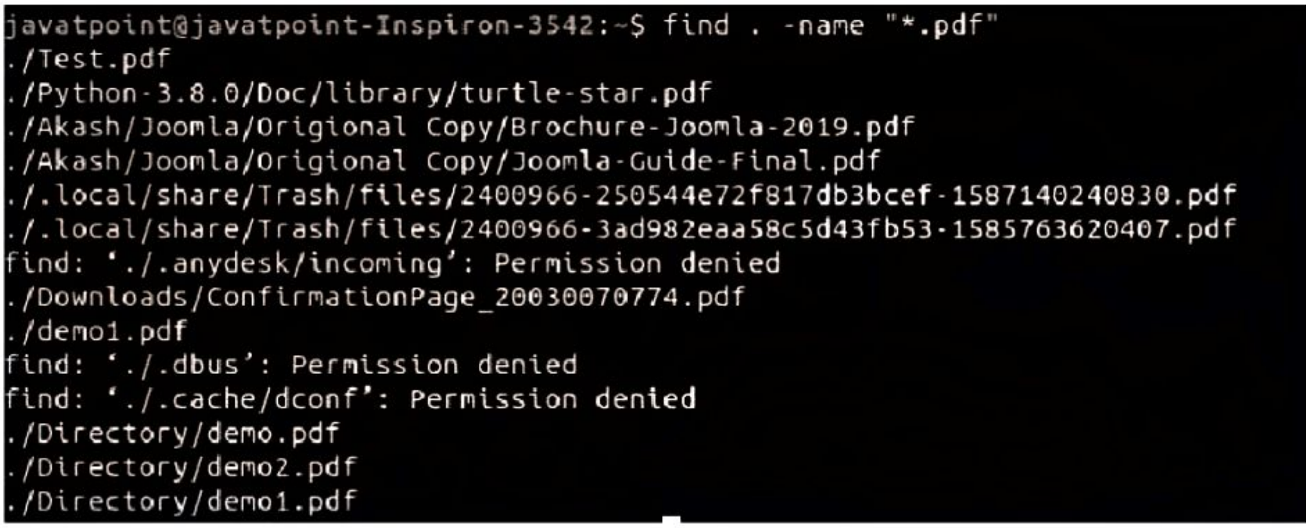
(.) : For current directory name

(/) : For root

Syntax:

1. 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

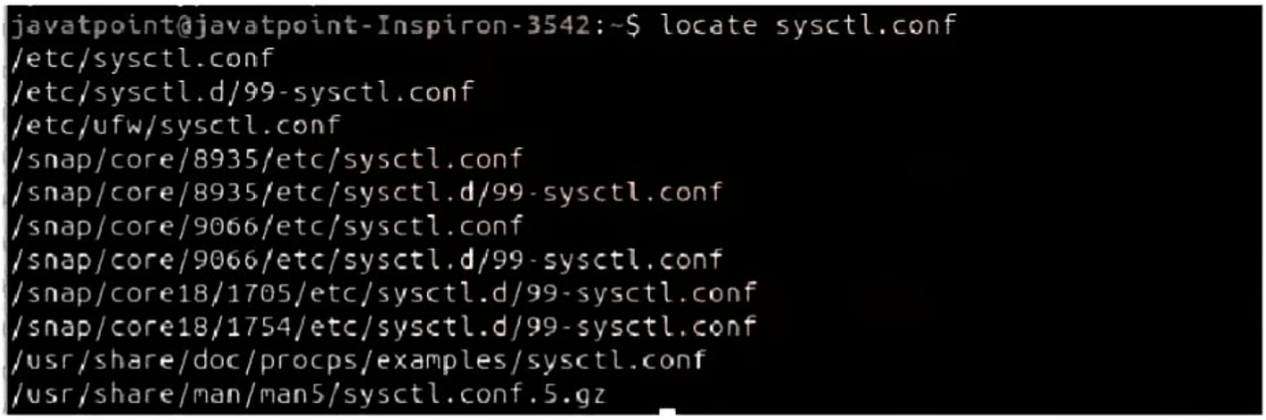
The locate

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:

1. 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

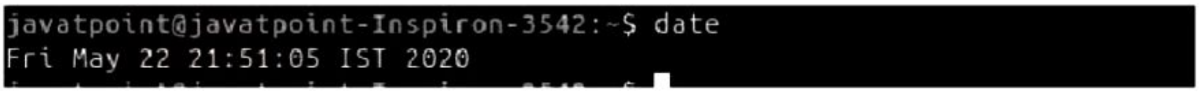
The date

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

Syntax:

1. date

Output:



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

38. cal Command

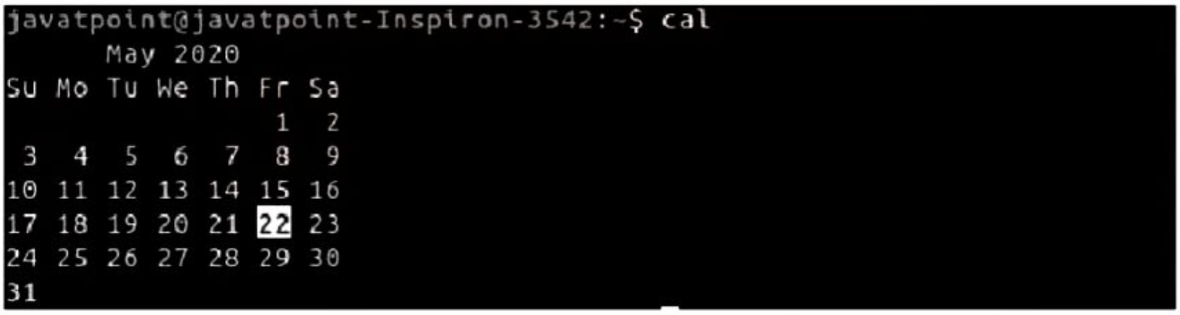
The cal

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

Syntax:

1. 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

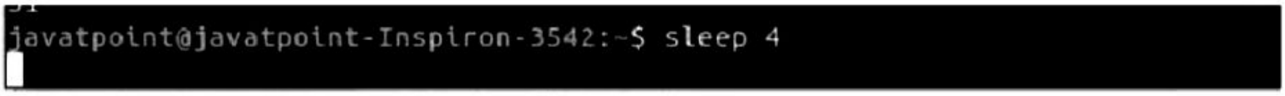
The sleep

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

Syntax:

1. sleep **<time>**

Output:



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

40. time Command

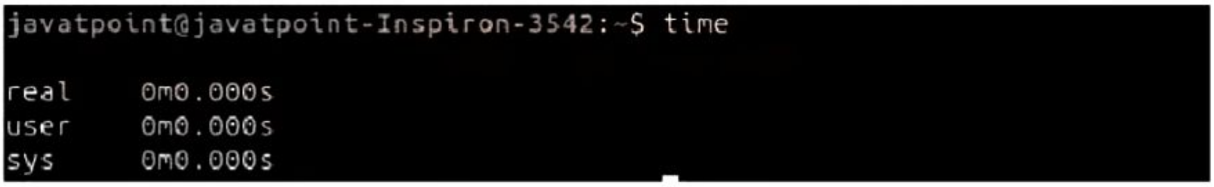
The time

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

Syntax:

1. time

Output:



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

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

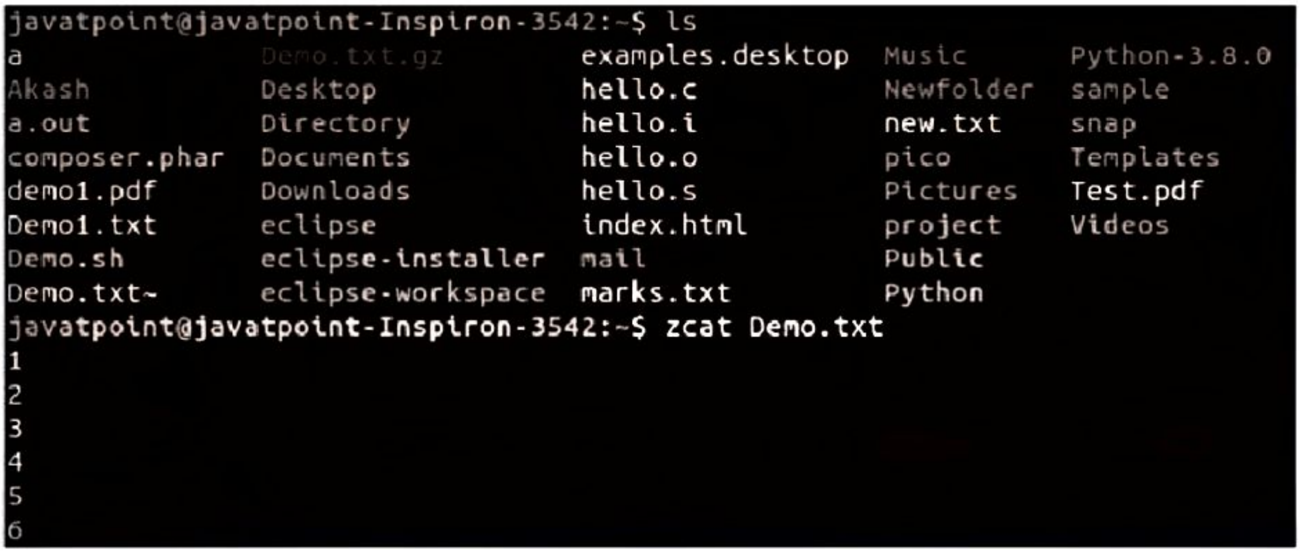
41. zcat Command

The zcat command is used to display the compressed files.

Syntax:

1. 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

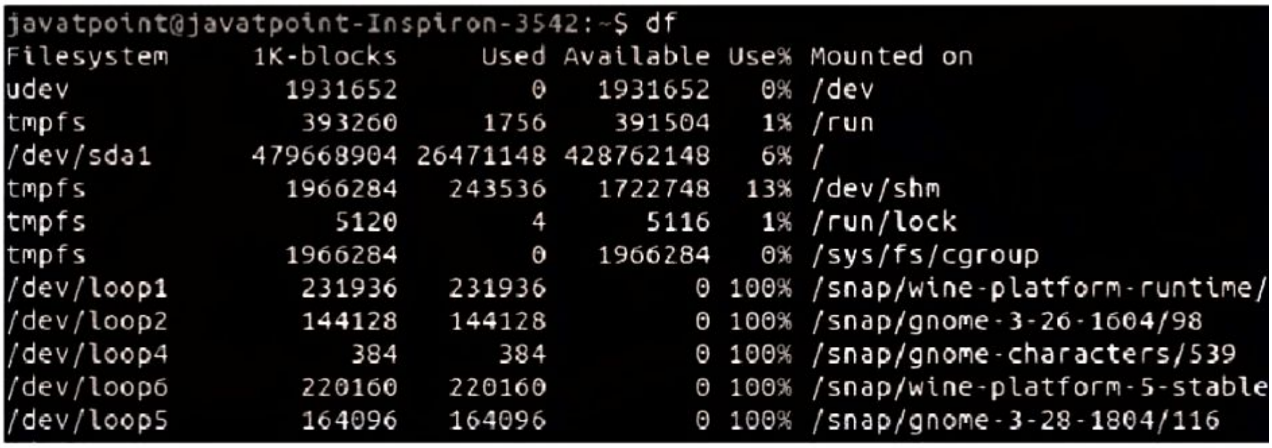
The df

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:

1. 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/
/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
/dev/loop5         164096   164096         0 100% /snap/gnome-3-28-1804/116
```

43. mount Command

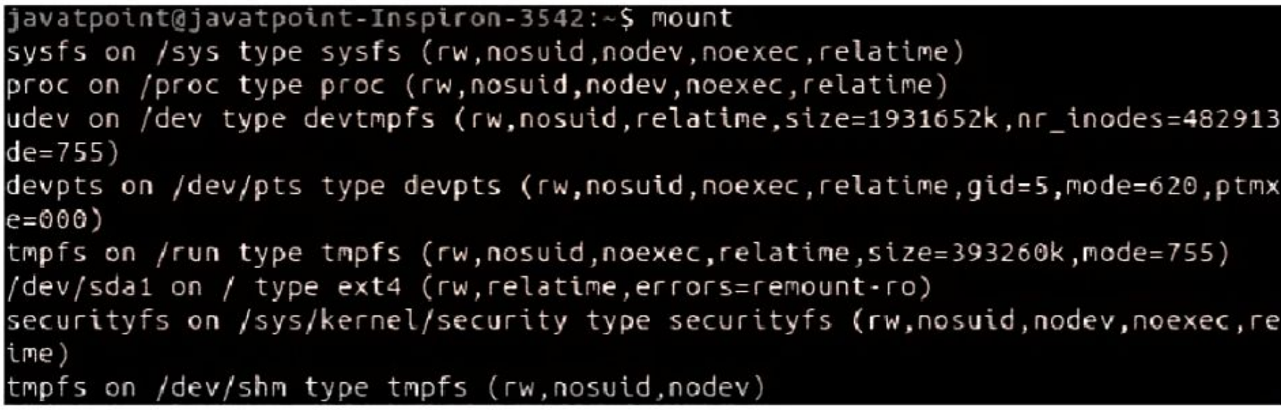
The mount

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

Syntax:

1. 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
de=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmx
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,re
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:

1. exit

Output:

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

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

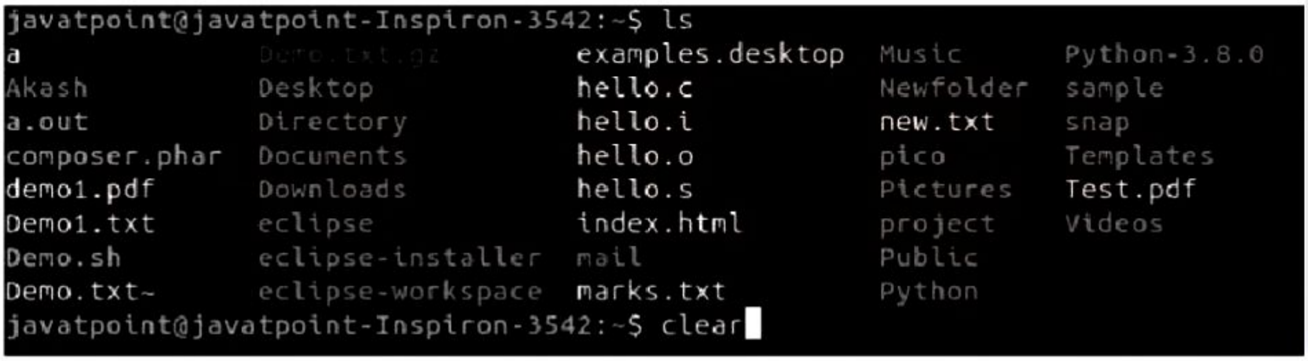
45. clear Command

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

Syntax:

1. clear

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:~$ clear
```

After pressing the ENTER key, it will clear the terminal screen.

Linux Networking Commands

46. ip Command

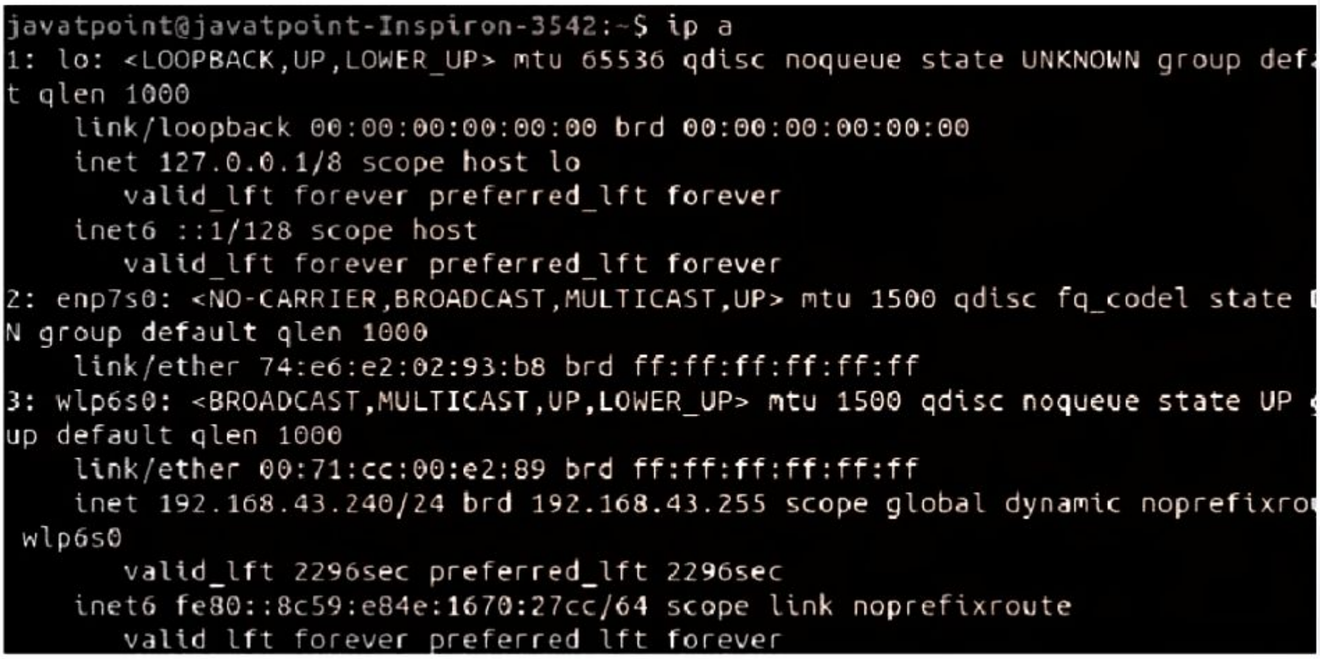
Linux ip

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

Syntax:

1. ip a or ip addr

Output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state
N group default qlen 1000
    link/ether 74:e6:e2:02:93:b8 brd ff:ff:ff:ff:ff:ff
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
up default qlen 1000
    link/ether 00:71:cc:00:e2:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.43.240/24 brd 192.168.43.255 scope global dynamic noprefixro
 wlp6s0
       valid_lft 2296sec preferred_lft 2296sec
    inet6 fe80::8c59:e84e:1670:27cc/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

47. ssh Command

Linux ssh

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

Syntax:

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

48. mail Command

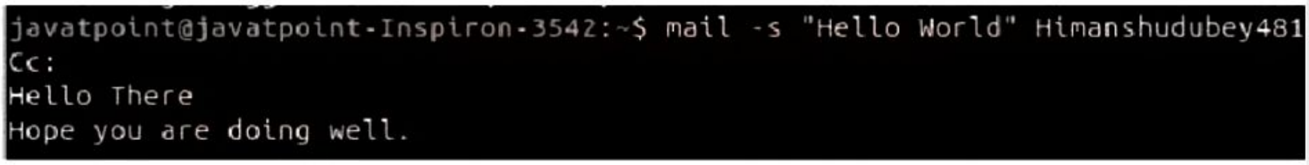
The mail

command is used to send emails from the command line.

Syntax:

1. mail -s "Subject" <recipient address>

Output:



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

49. ping Command

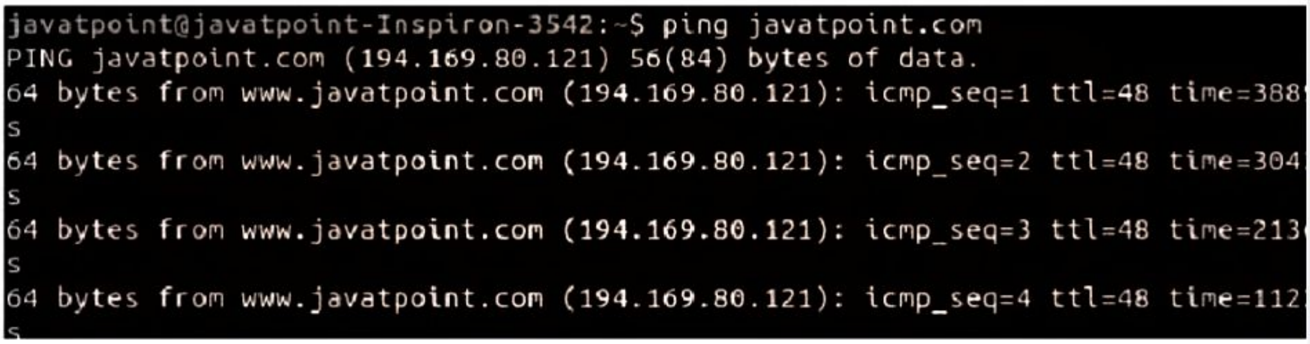
The ping

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:

1. 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=388
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=2 ttl=48 time=304
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=3 ttl=48 time=213
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=4 ttl=48 time=112
s
```

50. host Command

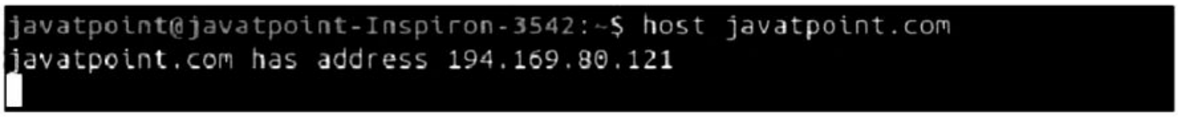
The host

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:

1. host **<domain** name> or **<ip** address>

   Output:

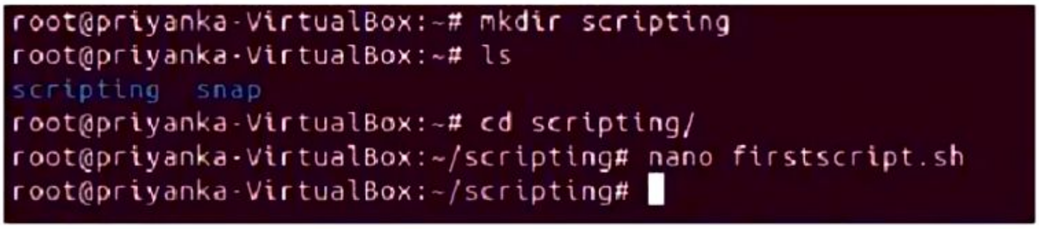


```
javatpoint@javatpoint-Inspiron-3542:~$ host javatpoint.com
javatpoint.com has address 194.169.80.121
```

Index

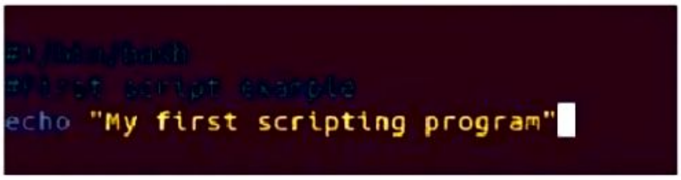
Shell Programming

1.Program to print string.



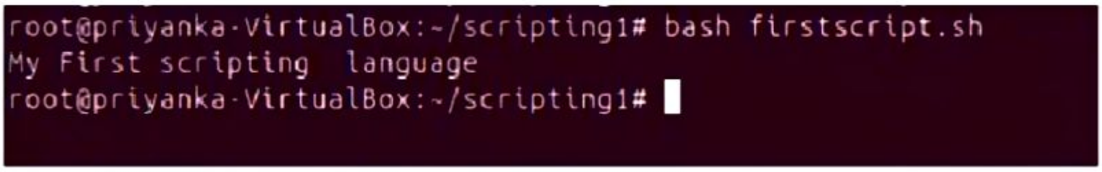
```
root@priyanka-VirtualBox:~# mkdir scripting
root@priyanka-VirtualBox:~# ls
scripting  snap
root@priyanka-VirtualBox:~# cd scripting/
root@priyanka-VirtualBox:~/scripting# nano firstscript.sh
root@priyanka-VirtualBox:~/scripting#
```

Create a shell script named firstscript.sh



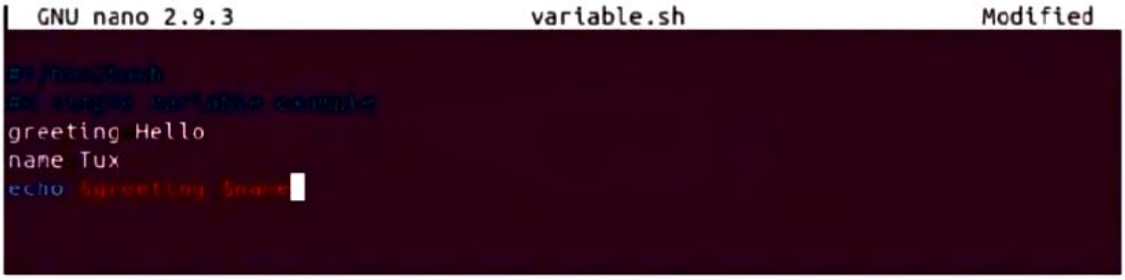
```
#!/bin/bash
#first script example
echo "My first scripting program"
```

Run the firstscript.sh



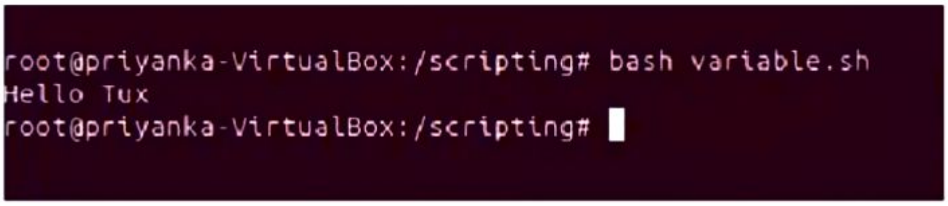
```
root@priyanka-VirtualBox:~/scripting1# bash firstscript.sh
My First scripting  language
root@priyanka-VirtualBox:~/scripting1#
```

2.Program to print variable name.



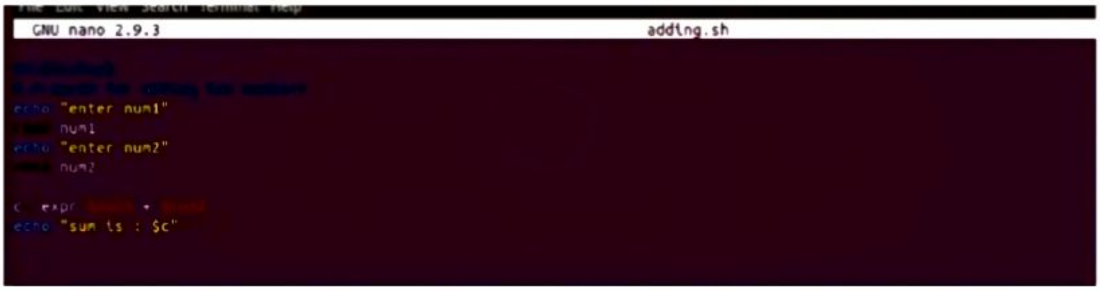
```
  GNU nano 2.9.3                     variable.sh                     Modified

#!/bin/bash
#A simple variable example
greeting Hello
name Tux
echo $greeting $name
```

Run the variable.sh script with this command



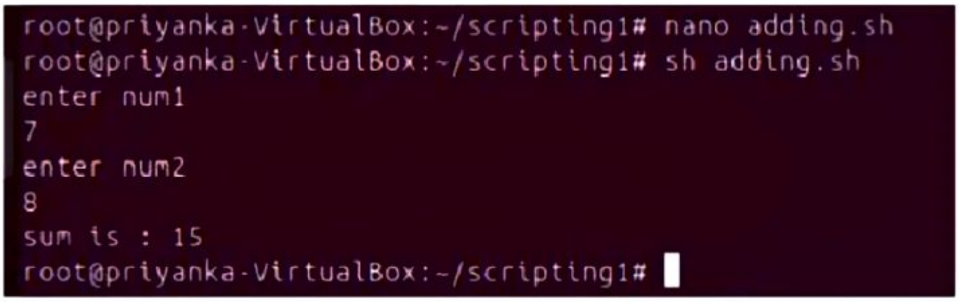
```
root@priyanka-VirtualBox:/scripting# bash variable.sh
Hello Tux
root@priyanka-VirtualBox:/scripting#
```

3.Program to read two integers number and print the addition of both variable.



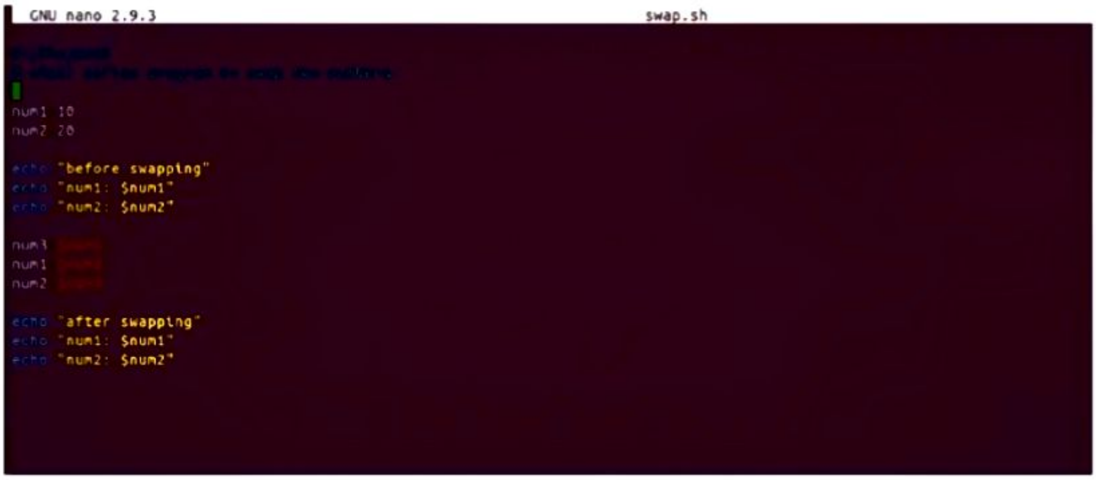
```
GNU nano 2.9.3                                      adding.sh

[illegible]
[illegible]
echo "enter num1"
[illegible] num1
echo "enter num2"
[illegible] num2

c  expr [illegible] + [illegible]
echo "sum is : $c"
```

Output



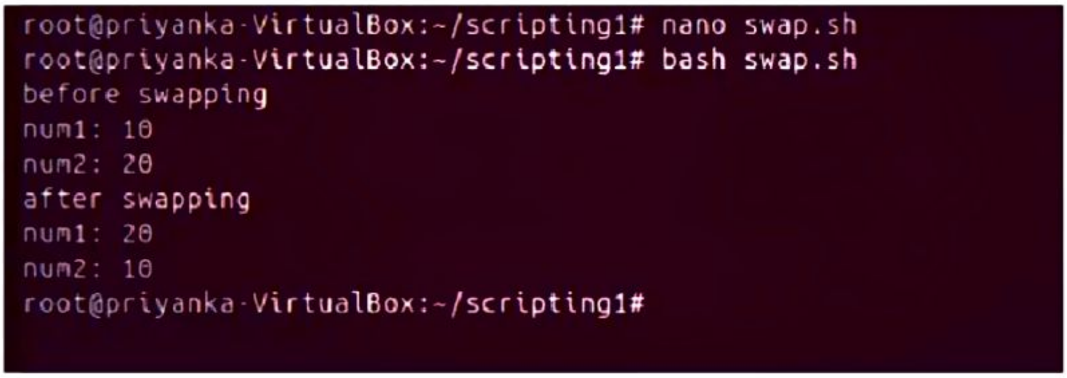
```
root@priyanka-VirtualBox:~/scripting1# nano adding.sh
root@priyanka-VirtualBox:~/scripting1# sh adding.sh
enter num1
7
enter num2
8
sum is : 15
root@priyanka-VirtualBox:~/scripting1#
```

4.Program to swap two numbers.

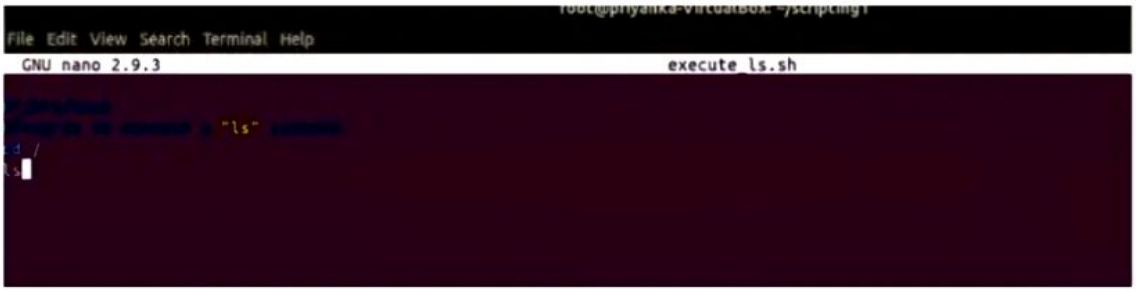


```
GNU nano 2.9.3                                        swap.sh

[illegible]
[illegible]

num1 10
num2 20

echo "before swapping"
echo "num1: $num1"
echo "num2: $num2"

num3 [illegible]
num1 [illegible]
num2 [illegible]

echo "after swapping"
echo "num1: $num1"
echo "num2: $num2"
```

Output:

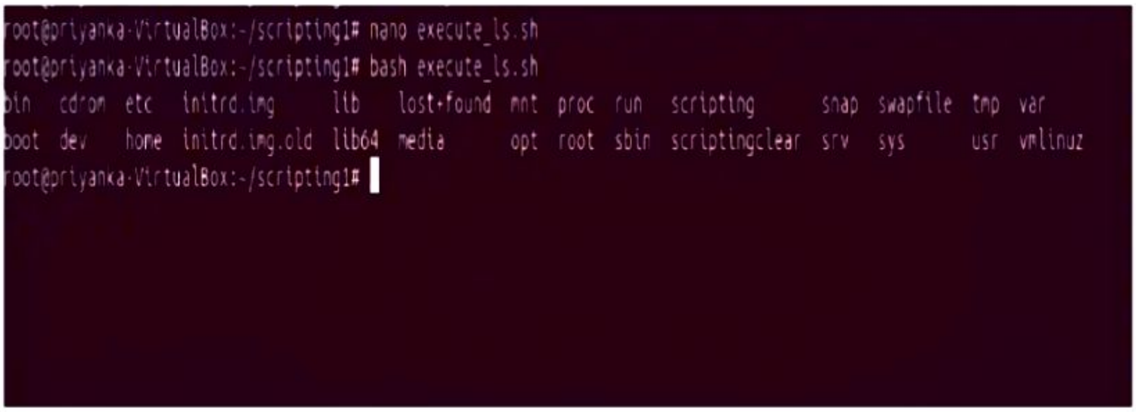


```
root@priyanka-VirtualBox:~/scripting1# nano swap.sh
root@priyanka-VirtualBox:~/scripting1# bash swap.sh
before swapping
num1: 10
num2: 20
after swapping
num1: 20
num2: 10
root@priyanka-VirtualBox:~/scripting1#
```



```
root@priyanka-VirtualBox: ~/scripting1
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                   execute_ls.sh

#!/bin/bash
[illegible] "ls" [illegible]
cd /
ls
```

Output:



```
root@priyanka-VirtualBox:~/scripting1# nano execute_ls.sh
root@priyanka-VirtualBox:~/scripting1# bash execute_ls.sh
bin   cdrom  etc   initrd.img      lib    lost+found  mnt  proc  run   scripting       snap  swapfile  tmp  var
boot  dev    home  initrd.img.old  lib64  media       opt  root  sbin  scriptingclear  srv   sys       usr  vmlinuz
root@priyanka-VirtualBox:~/scripting1# 
```

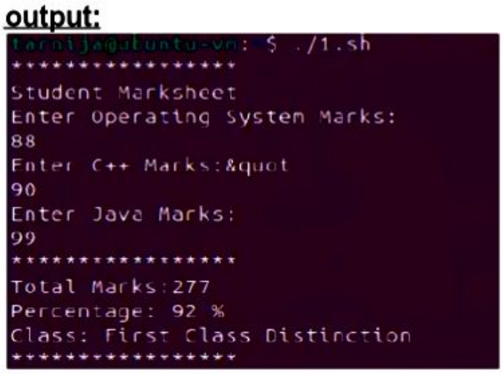
Practical-3

Aim:Write a shell script to generate marksheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.

Program:

```
echo "*****************"
echo "Student Marksheet"
echo "*****************"
echo "Enter Operating System Marks:"
read os
echo "Enter C++ Marks:"
read cpp
echo "Enter Java Marks:"
read java
echo "*****************"
total=`expr $os + $cpp + $java`
echo "Total Marks:"$total
percentage=`expr $total / 3`
echo "Percentage:" $percentage %
if [ $percentage -ge 60 ]
then
echo "Class: First Class Distinction"
elif [ $percentage -ge 50 ]
then
echo "Class: First class"
elif [ $percentage -ge 40 ]
then
echo "Class: Second class"
else
echo "Class: Fail"
fi
echo "*****************"
```

output:



```
tarnija@ubuntu-vm:~$ ./1.sh
*****************
Student Marksheet
Enter Operating System Marks:
88
Enter C++ Marks:&quot
90
Enter Java Marks:
99
*****************
Total Marks:277
Percentage: 92 %
Class: First Class Distinction
*****************
```

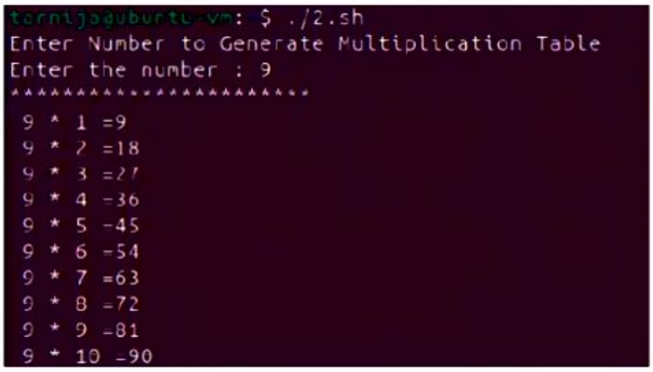
Practical-4

Aim:-Write a shell script to display multiplication table of given numbe

Program:-

```
echo "Enter Number to Generate Multiplication Table"
read -p "Enter the number : " number
echo "***********************"
i=1
while [ $i -le 10 ]
do
echo " $number * $i =`expr $number \* $i ` "
i=`expr $i + 1`
done
echo "***********************"
```

Output:



```
tarnija@ubuntu-vm: $ ./2.sh
Enter Number to Generate Multiplication Table
Enter the number : 9
************************
 9 * 1 =9
 9 * 2 =18
 9 * 3 =27
 9 * 4 =36
 9 * 5 =45
 9 * 6 =54
 9 * 7 =63
 9 * 8 =72
 9 * 9 =81
 9 * 10 =90
```

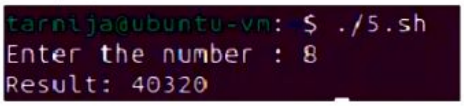
Practical-5

Aim:Write a shell script to find factorial of given number n.

Program:

```
read -p
 "Enter the number : " number
 fact=1
while [ $number -gt 1 ]
do
fact=$((fact * number))
 number=$((number - 1))
 done echo "Result:" $fact
```

Output



```
tarnija@ubuntu-vm: $ ./5.sh
Enter the number : 8
Result: 40320
```

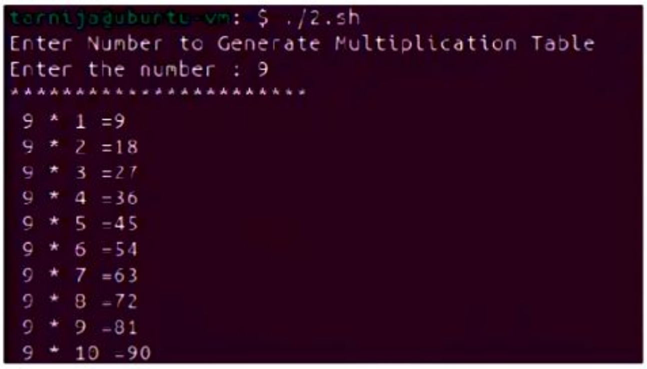
Practical-4

Aim:-Write a shell script to display multiplication table of given numbe

Program:-

```
echo "Enter Number to Generate Multiplication Table"
read -p "Enter the number : " number
echo "************************"
i=1
while [ $i -le 10 ]
do
echo " $number * $i =`expr $number \* $i ` "
i=`expr $i + 1`
done
echo "************************"
```

Output:



```
tarnija@ubuntu-vm: $ ./2.sh
Enter Number to Generate Multiplication Table
Enter the number : 9
************************
 9 * 1 =9
 9 * 2 =18
 9 * 3 =27
 9 * 4 =36
 9 * 5 =45
 9 * 6 =54
 9 * 7 =63
 9 * 8 =72
 9 * 9 =81
 9 * 10 =90
```

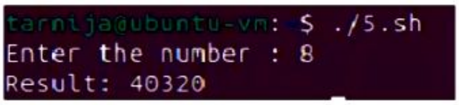
Practical-5

Aim:Write a shell script to find factorial of given number n.

Program:

```
read -p
 "Enter the number : " number
 fact=1
while [ $number -gt 1 ]
do
fact=$((fact * number))
 number=$((number - 1))
 done echo "Result:" $fact
```

Output



```
tarnija@ubuntu-vm: $ ./5.sh
Enter the number : 8
Result: 40320
```

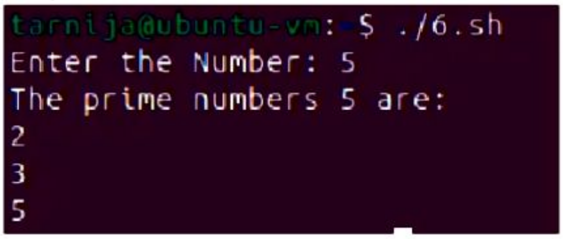
Practical 6:

Aim:Write a shell script which will accept a number b and display first n prime numbers as output

Program:

```
read -p "Enter the Number: " n
echo "The prime numbers $n are: "
m=2
while [ $m -le $n ]
do
i=2
flag=0
while [ $i -le `expr $m / 2` ]
do
if [ `expr $m % $i` -eq 0 ]
then
flag=1
break
fi
i=`expr $i + 1`
done
if [ $flag -eq 0 ]
then
echo $m
fi
m=`expr $m + 1`
done
```

Output:



```
tarnija@ubuntu-vm:~$ ./6.sh
Enter the Number: 5
The prime numbers 5 are:
2
3
5
```

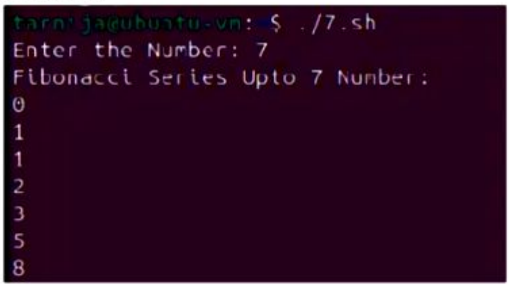
Practical 7:

Aim:Write a shell script which will generate first n fibonnacci numbers like: 1, 1, 2, 3, 5, 13, ...

Program:

```
read -p "Enter the Number: " number
x=0
y=1
i=2
echo "Fibonacci Series Upto $number Number: "
echo "$x"
echo "$y"
while [ $i -lt $number ]
do
i=`expr $i + 1`
z=`expr $x + $y`
echo "$z"
x=$y
y=$z
done
```

Output:



```
tarnija@ubuntu-vm:~$ ./7.sh
Enter the Number: 7
Fibonacci Series Upto 7 Number:
0
1
1
2
3
5
8
```

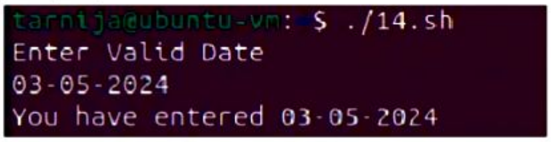
Practical 14:

Aim: Write a shell script to validate the entered date. (eg. Date format is : dd-mm-yyyy)

Program:

```
echo "Enter Valid Date"
read date
echo "You have entered $date"
date -d $date
if [ $? -eq 0 ]
then
echo "Enter Date is Valid"
else
echo "Enter Date is Invalid"
if
```

Output:



```
tarnija@ubuntu-vm:~$ ./14.sh
Enter Valid Date
03-05-2024
You have entered 03-05-2024
```

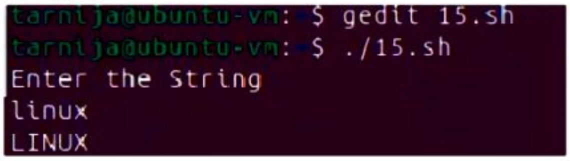
Practical 15:

Aim:Write an awk program using function, which convert each word in a given text into capital

Program:

```
echo "Enter the String"
a=$(awk 'BEGIN{
getline str;
print toupper(str);
}')
echo $a
```

Output:



```
tarnija@ubuntu-vm:~$ gedit 15.sh
tarnija@ubuntu-vm:~$ ./15.sh
Enter the String
linux
LINUX
```