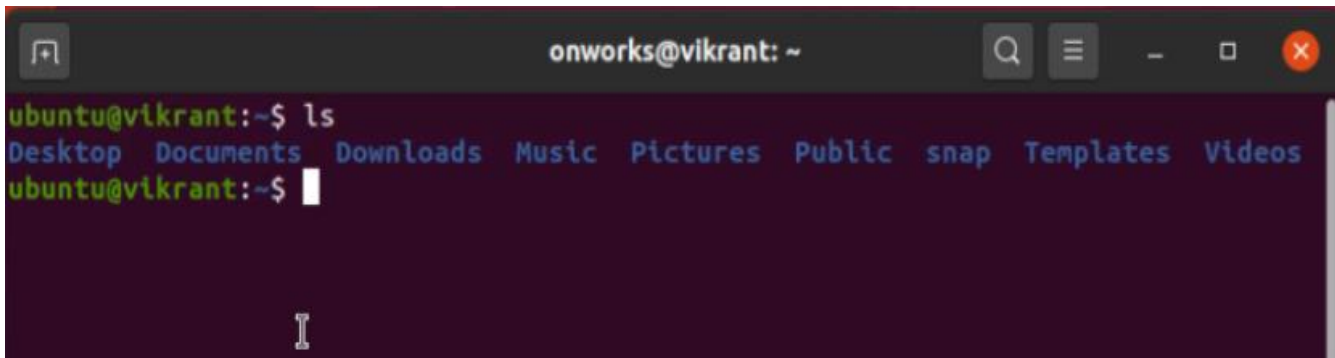


Basic commands used in Linux

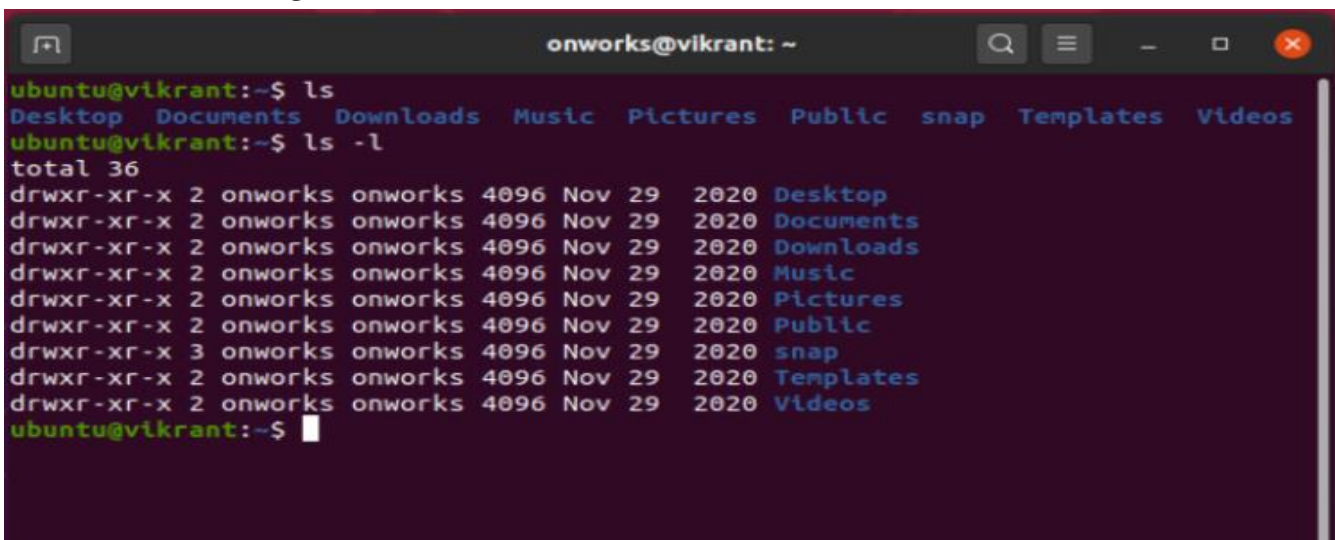
1 . Ls :- The `ls` command is one of the most commonly used commands in daily Linux/UNIX operations. The command is used in listing contents inside a directory and is one of the few command's beginners learn from the onset

Example:

A terminal window titled 'onworks@vikrant: ~' showing the command 'ls' being executed. The output lists the contents of the home directory: Desktop, Documents, Downloads, Music, Pictures, Public, snap, Templates, and Videos.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ ls  
Desktop Documents Downloads Music Pictures Public snap Templates Videos  
ubuntu@vikrant:~$
```

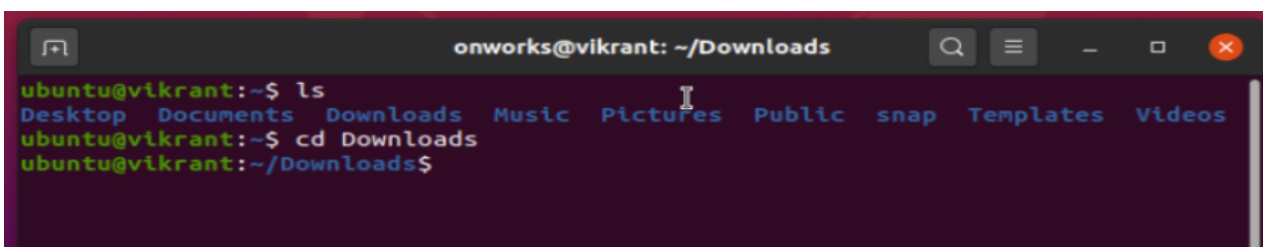
This `ls -l` command is used to list the contents of the directory in atable format with columns including

A terminal window titled 'onworks@vikrant: ~' showing the command 'ls -l' being executed. The output displays a detailed table of directory contents, including permissions, number of links, owner, group, size, date, and name.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ ls  
Desktop Documents Downloads Music Pictures Public snap Templates Videos  
ubuntu@vikrant:~$ ls -l  
total 36  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Desktop  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Documents  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Downloads  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Music  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Pictures  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Public  
drwxr-xr-x 3 onworks onworks 4096 Nov 29 2020 snap  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Templates  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Videos  
ubuntu@vikrant:~$
```

2 . Cd :- Cd command in Linux known as change directory command. It is used to change current working directory. Syntax: `$ cd [directory]` To moveinside a subdirectory : to move inside a subdirectory in Linux we use

`$ cd [directory name]`

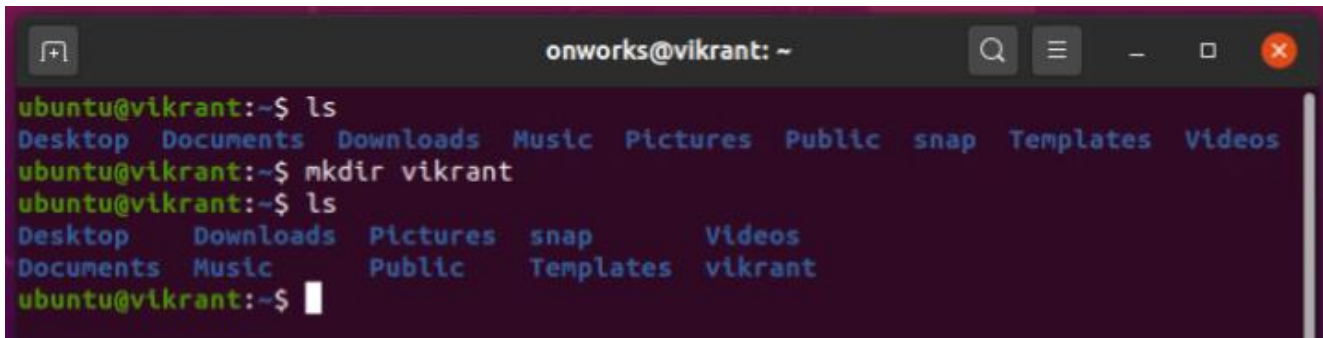
A terminal window titled 'onworks@vikrant: ~/Downloads' showing the command 'cd Downloads' being executed. The prompt changes to '~Downloads\$', indicating the current directory has been changed.

```
onworks@vikrant: ~/Downloads  
ubuntu@vikrant:~$ ls  
Desktop Documents Downloads Music Pictures Public snap Templates Videos  
ubuntu@vikrant:~$ cd Downloads  
ubuntu@vikrant:~/Downloads$
```

3. Mkdir :- *mkdir command in Linux allows the user to create directories(also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories.*

Syntax: mkdir [options...] [directories ...]

Example :

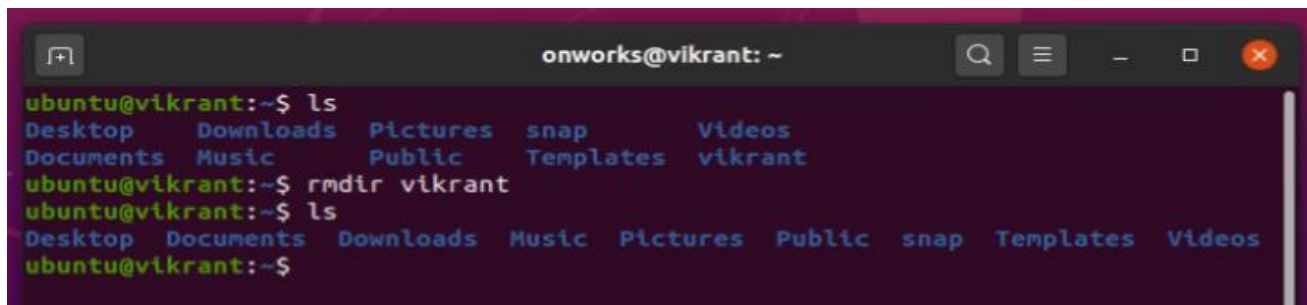
A terminal window titled 'onworks@vikrant: ~' showing the execution of the 'mkdir' command. The user runs 'ls' and sees a list of directories: Desktop, Documents, Downloads, Music, Pictures, Public, snap, Templates, and Videos. Then, the user runs 'mkdir vikrant', creating a new directory. Finally, the user runs 'ls' again, and the new directory 'vikrant' is added to the list.

```
onworks@vikrant: ~
ubuntu@vikrant:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
ubuntu@vikrant:~$ mkdir vikrant
ubuntu@vikrant:~$ ls
Desktop  Downloads  Pictures  snap      Videos
Documents Music      Public    Templates vikrant
ubuntu@vikrant:~$
```

4 Rmdir :- *Rmdir command is used remove empty directories from the filesystem in Linux. The rmdir command removes each and every directory specified in the command line only if these directories are empty. So if the specified directory has some directories or files in it then this cannot be removed by rmdir command.*

Syntax : rmdir

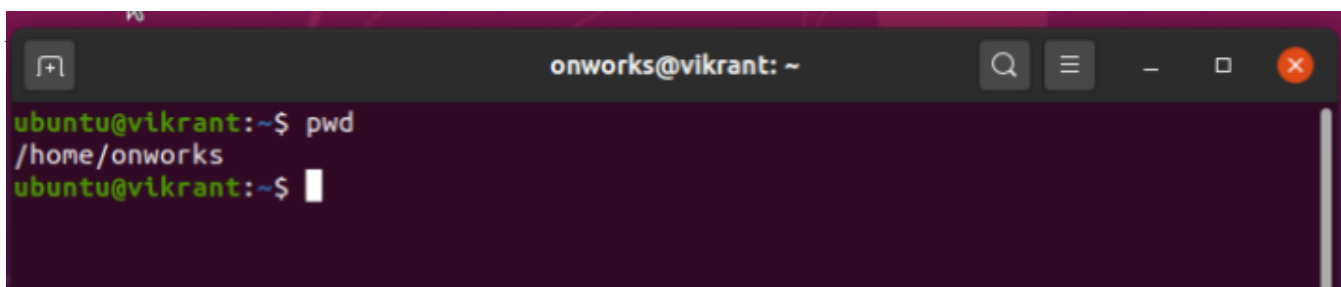
Example :

A terminal window titled 'onworks@vikrant: ~' showing the execution of the 'rmdir' command. The user runs 'ls' and sees the list of directories including 'vikrant'. Then, the user runs 'rmdir vikrant', removing the directory. Finally, the user runs 'ls' again, and 'vikrant' is no longer in the list.

```
onworks@vikrant: ~
ubuntu@vikrant:~$ ls
Desktop  Downloads  Pictures  snap      Videos
Documents Music      Public    Templates vikrant
ubuntu@vikrant:~$ rmdir vikrant
ubuntu@vikrant:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
ubuntu@vikrant:~$
```

5 Pwd :- *Pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root. pwd is shell built-in command(pwd) or an actual binary(/bin/pwd). \$PWD is an environment variable which stores the path of the current directory*

Syntax : pwd [option]

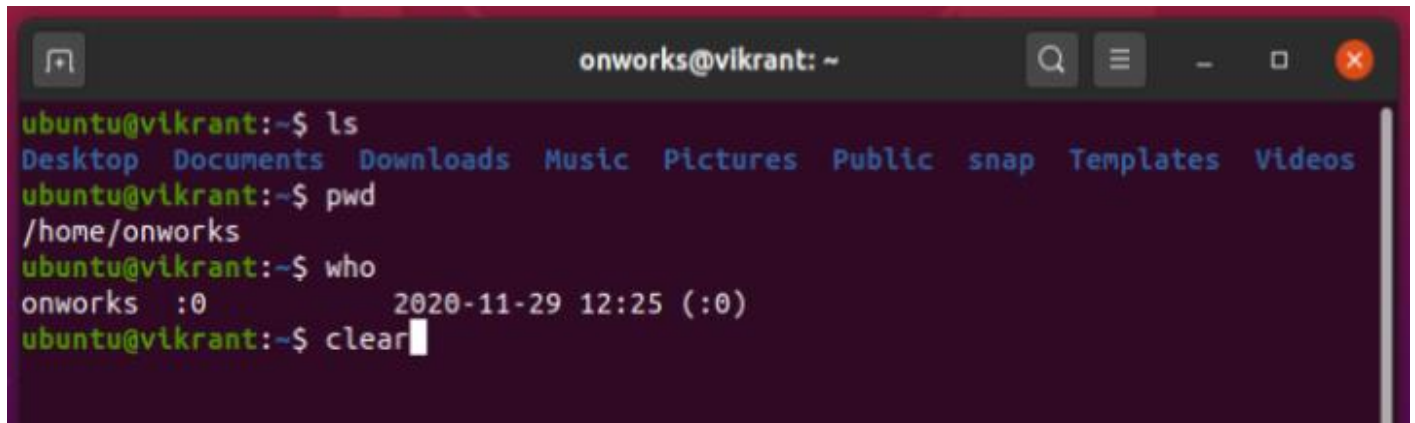
A terminal window titled 'onworks@vikrant: ~' showing the execution of the 'pwd' command. The user runs 'pwd', and the terminal outputs '/home/onworks', which is the absolute path of the current directory.

```
onworks@vikrant: ~
ubuntu@vikrant:~$ pwd
/home/onworks
ubuntu@vikrant:~$
```

6 Clear :- The `clear` command is the go-to tool for clearing the terminal screen in Linux . keyboard shortcut also work for clear command we can use `ctrl + L` to clear screen

Syntax : `clear`

Example :-

A terminal window titled 'onworks@vikrant: ~' with standard Ubuntu window controls. The terminal shows the following commands and output:

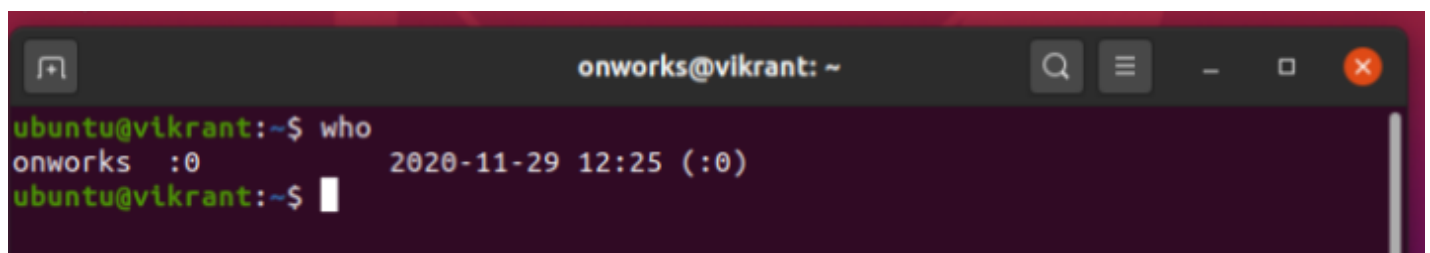
```
ubuntu@vikrant:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
ubuntu@vikrant:~$ pwd
/home/onworks
ubuntu@vikrant:~$ who
onworks  :0                2020-11-29 12:25 (:0)
ubuntu@vikrant:~$ clear
```

7 Who :- The `who` command is used to get information about currently logged in user on to system. Thels

`who` command is related to the command `w` ,which provides the same information but also displays additional data and statistics.

Syntax : `who [options] [filename]`

Example :

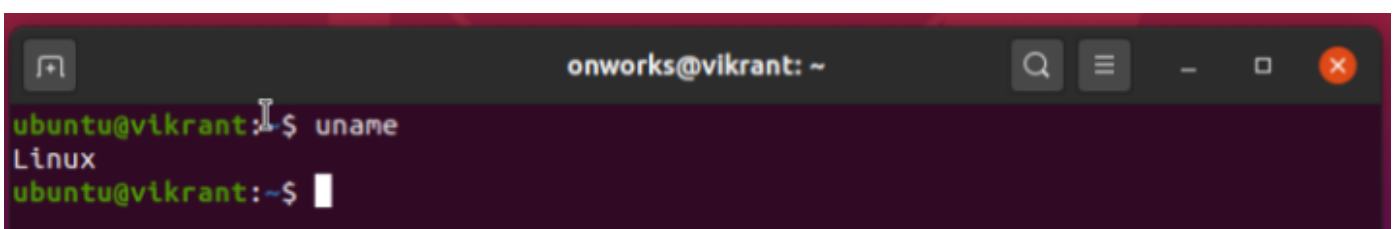
A terminal window titled 'onworks@vikrant: ~' with standard Ubuntu window controls. The terminal shows the following commands and output:

```
ubuntu@vikrant:~$ who
onworks  :0                2020-11-29 12:25 (:0)
ubuntu@vikrant:~$
```

8 Uname :- Uname is a command-line utility that prints basic information about the operating system name and system hardware. The `uname` tool is most commonly used to determine the processor architecture.

Syntax :- `uname [OPTIONS]...`

Example:-

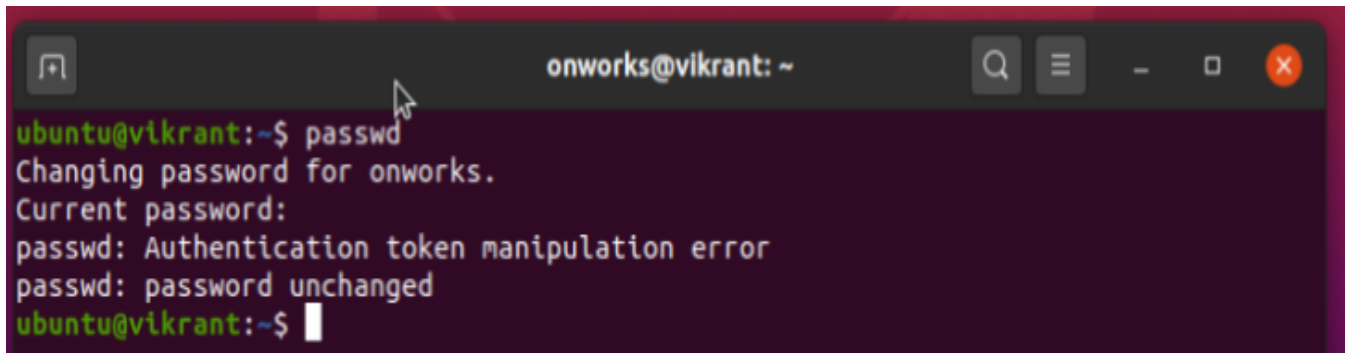
A terminal window titled 'onworks@vikrant: ~' with standard Ubuntu window controls. The terminal shows the following commands and output:

```
ubuntu@vikrant:~$ uname
Linux
ubuntu@vikrant:~$
```

9 Passwd :- *Passwd command in Linux is used to change the user account passwords. The root user reserves the privilege to change the password for any user on the system, while a normal user can only change the account password for his or her own account.*

Syntax : - passwd [options] [username]

Example :-



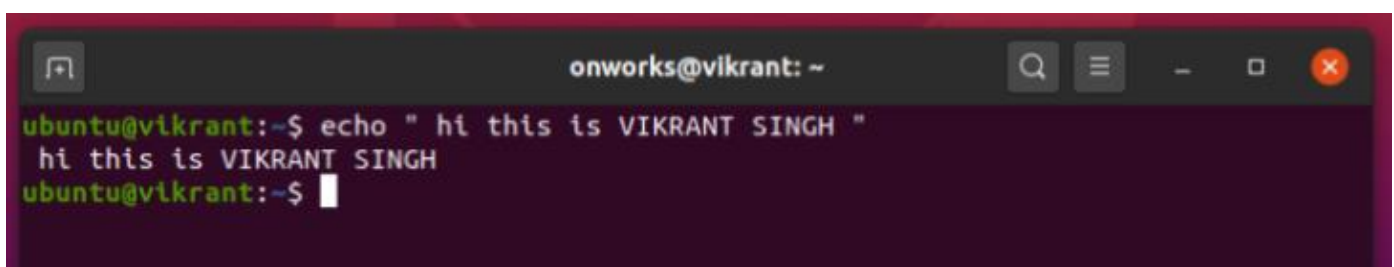
```
onworks@vikrant: ~  
ubuntu@vikrant:~$ passwd  
Changing password for onworks.  
Current password:  
passwd: Authentication token manipulation error  
passwd: password unchanged  
ubuntu@vikrant:~$
```

10 Echo: - *Echo command in Linux is used to display line of text/string that are passed as an argument. This is a builtin command that is mostly used in shell scripts and batch files to output status text to the screen or a file.*

Syntax :

echo [option] [string]

Example :-



```
onworks@vikrant: ~  
ubuntu@vikrant:~$ echo " hi this is VIKRANT SINGH "  
hi this is VIKRANT SINGH  
ubuntu@vikrant:~$
```

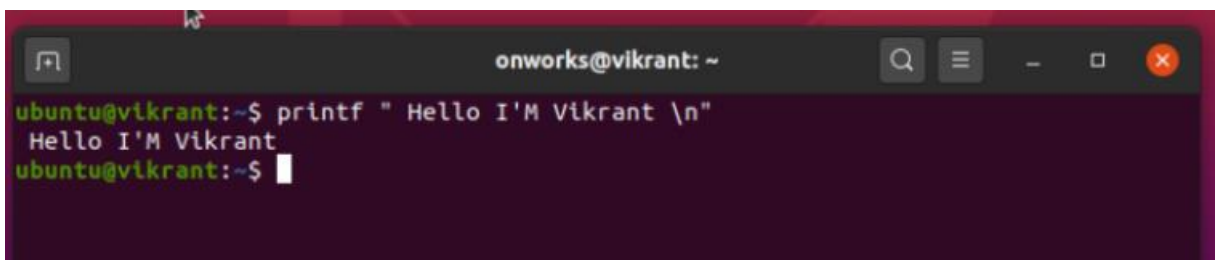
11. Printf:- *“Printf” command in Linux is used to display the given string, number or any other format specifier on the terminal window*

Syntax:

`$printf [-v var] format [arguments]`

Printf can have format specifiers, escape sequences or ordinary characters.

Example

A terminal window titled 'onworks@vikrant: ~' showing the execution of the printf command. The prompt is 'ubuntu@vikrant:~\$'. The command entered is 'printf " Hello I'M Vikrant \n"'. The output is 'Hello I'M Vikrant' followed by a new line. The prompt returns to 'ubuntu@vikrant:~\$' with a cursor.

```
onworks@vikrant: ~
ubuntu@vikrant:~$ printf " Hello I'M Vikrant \n"
Hello I'M Vikrant
ubuntu@vikrant:~$
```

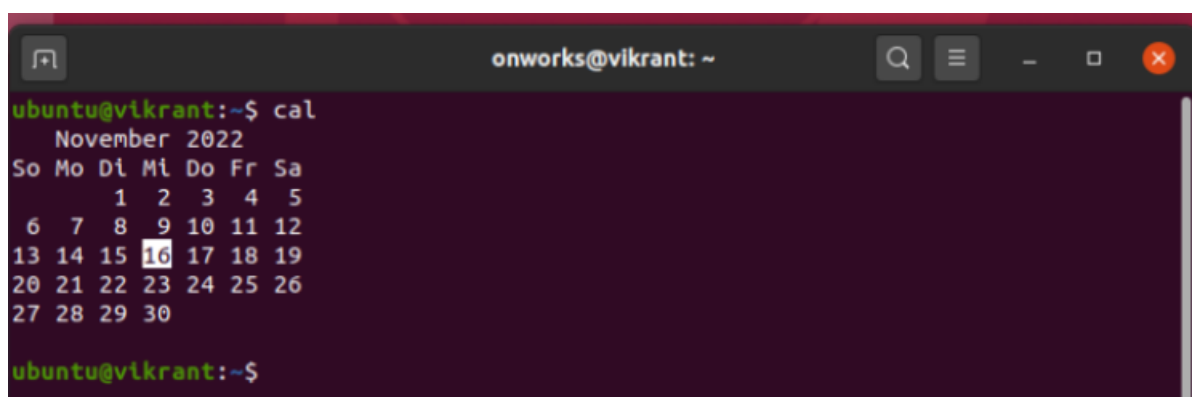
12 Cal :- *Cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year .*

Syntax:

`cal [[month] year]`

Rectangular bracket means it is optional, so if used without option, it will display a calendar of current month and year.

Example:

A terminal window titled 'onworks@vikrant: ~' showing the execution of the cal command. The prompt is 'ubuntu@vikrant:~\$'. The command entered is 'cal'. The output shows the calendar for November 2022, with days of the week as headers and dates 1 through 30. The date 16 is highlighted in blue. The prompt returns to 'ubuntu@vikrant:~\$' with a cursor.

```
onworks@vikrant: ~
ubuntu@vikrant:~$ cal
November 2022
So Mo Di Mi Do 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
ubuntu@vikrant:~$
```

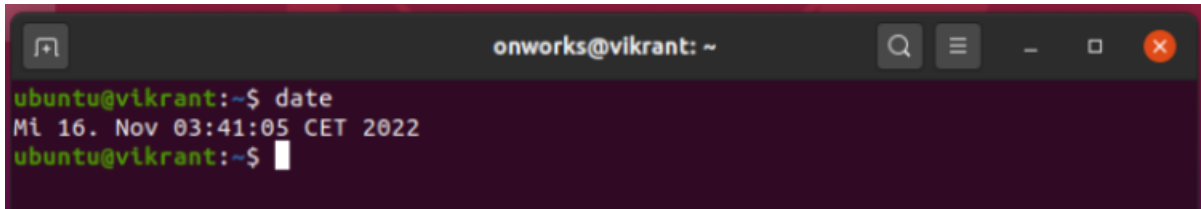

12 Date : Date command is used to display the system date and time.

date command is also used to set date and time of the system. By default, the date command displays the date in the time zone on which Unix/Linux operating system is configured. You must be the super-user (root) to change the date and time

Syntax: date [OPTION]... [+FORMAT]

date [-u/--utc/--universal] [MMDDhhmm[[CC]YY] [. ss]]

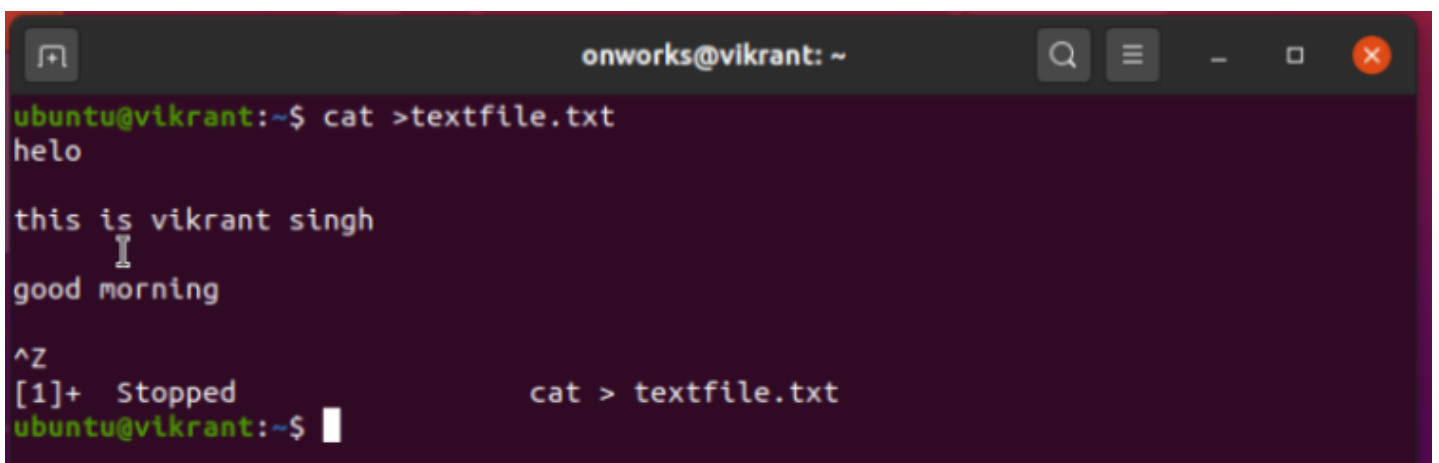
Example

A terminal window titled 'onworks@vikrant: ~' showing the command 'date' being executed. The output is 'Mi 16. Nov 03:41:05 CET 2022'.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ date  
Mi 16. Nov 03:41:05 CET 2022  
ubuntu@vikrant:~$
```

14 CAT :- concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

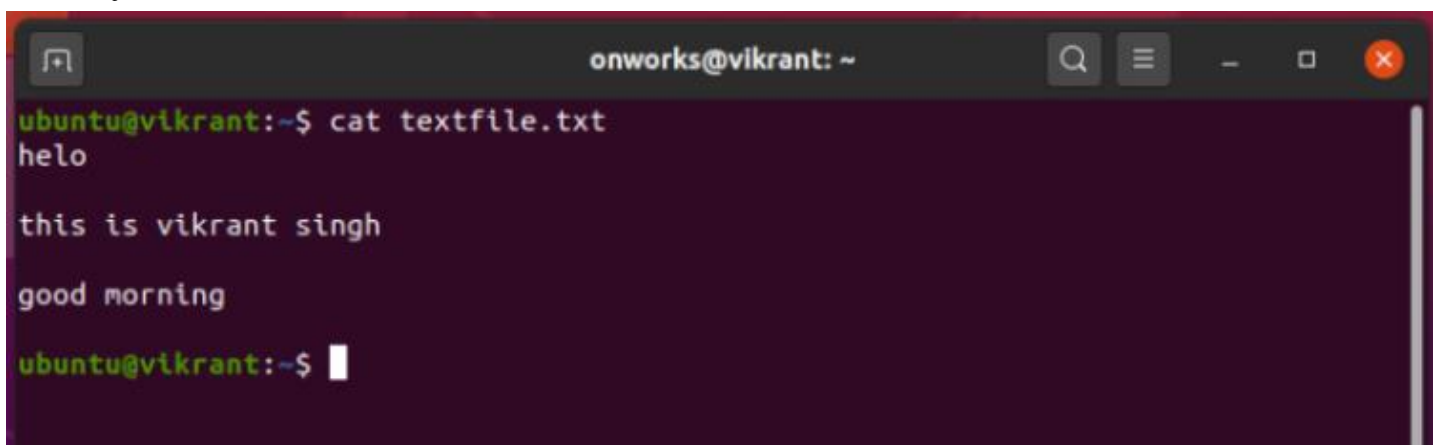
Create a file Command: \$ cat > filename

A terminal window titled 'onworks@vikrant: ~' showing the command 'cat > textfile.txt' being executed. The user enters 'helo', 'this is vikrant singh', and 'good morning' on separate lines. The prompt changes to '[1]+ Stopped cat > textfile.txt' when the user presses Ctrl+Z.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ cat > textfile.txt  
helo  
  
this is vikrant singh  
good morning  
  
^Z  
[1]+  Stopped                  cat > textfile.txt  
ubuntu@vikrant:~$
```

To view the content in the file command:

\$ cat file name

A terminal window titled 'onworks@vikrant: ~' showing the command 'cat textfile.txt' being executed. The output displays the content of the file: 'helo', 'this is vikrant singh', and 'good morning'.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ cat textfile.txt  
helo  
  
this is vikrant singh  
good morning  
  
ubuntu@vikrant:~$
```

15 .CP : - *cp* stands for copy . This command is used to copy files or groups of files or directory . It creates an exact image of a file on a disk with a different file name. *cp* command requires at least two filenames in its arguments To copy a file to another destination Command : : \$ *cp -v* [filename] [destination]

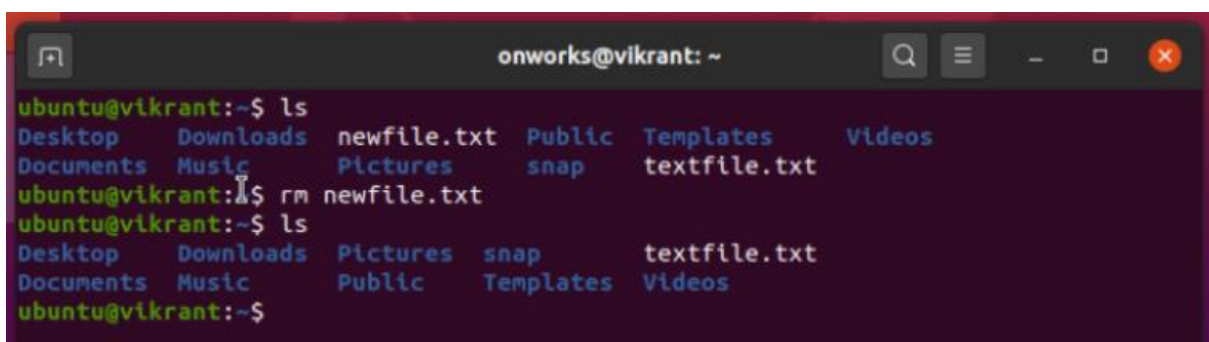
Suppose there is a directory named newFile having a text file new.txt and a directory name folder in which we are going to copy that file .

A terminal window titled 'onworks@vikrant: ~' showing the execution of the 'cp' command. The user runs 'ls' and lists files: Desktop, Downloads, newfile.txt, Public, Templates, Videos, Documents, Music, Pictures, snap, and textfile.txt. Then, they run 'cp textfile.txt newfile.txt'. Finally, they run 'cat newfile.txt' and see the output: 'helo', 'this is vikrant singh', and 'good morning'.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ ls  
Desktop  Downloads  newfile.txt  Public  Templates  Videos  
Documents Music      Pictures    snap    textfile.txt  
ubuntu@vikrant:~$ cp textfile.txt newfile.txt  
ubuntu@vikrant:~$ cat newfile.txt  
helo  
  
this is vikrant singh  
good morning  
ubuntu@vikrant:~$
```

16 rm :- *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. To be more precise, *rm* removes references to objects from the filesystem, where those objects might have had multiple references.

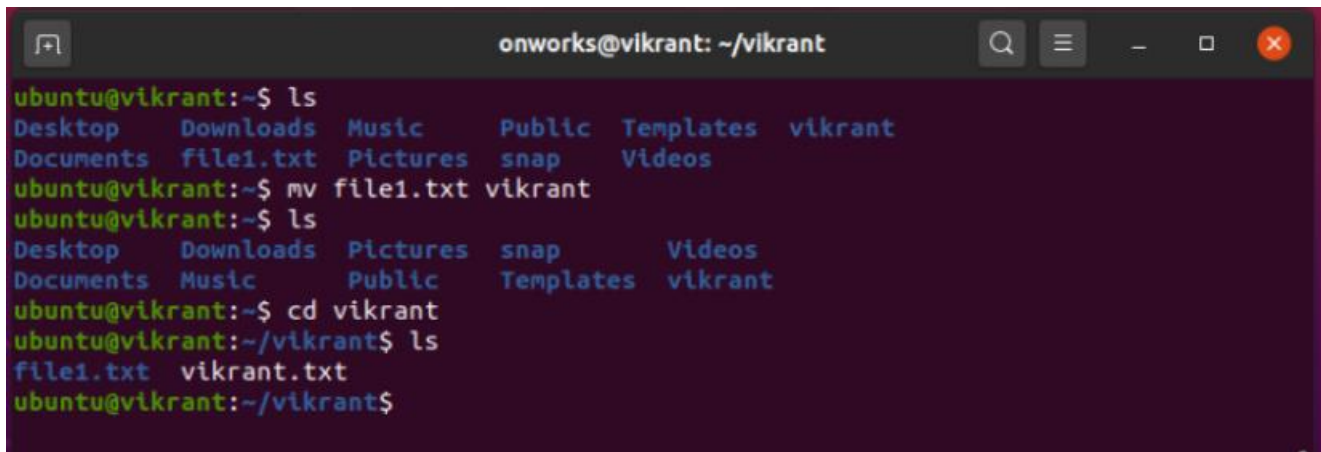
To remove file Command : \$ *rm* [filename]

A terminal window titled 'onworks@vikrant: ~' showing the execution of the 'rm' command. The user runs 'ls' and lists files: Desktop, Downloads, newfile.txt, Public, Templates, Videos, Documents, Music, Pictures, snap, and textfile.txt. Then, they run 'rm newfile.txt'. Finally, they run 'ls' again and see the updated list of files, where 'newfile.txt' has been removed.

```
onworks@vikrant: ~  
ubuntu@vikrant:~$ ls  
Desktop  Downloads  newfile.txt  Public  Templates  Videos  
Documents Music      Pictures    snap    textfile.txt  
ubuntu@vikrant:~$ rm newfile.txt  
ubuntu@vikrant:~$ ls  
Desktop  Downloads  Pictures    snap    textfile.txt  
Documents Music      Public      Templates Videos  
ubuntu@vikrant:~$
```

17 MV :- *mv stands for move. mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions: (i) It moves files to a different directory.*

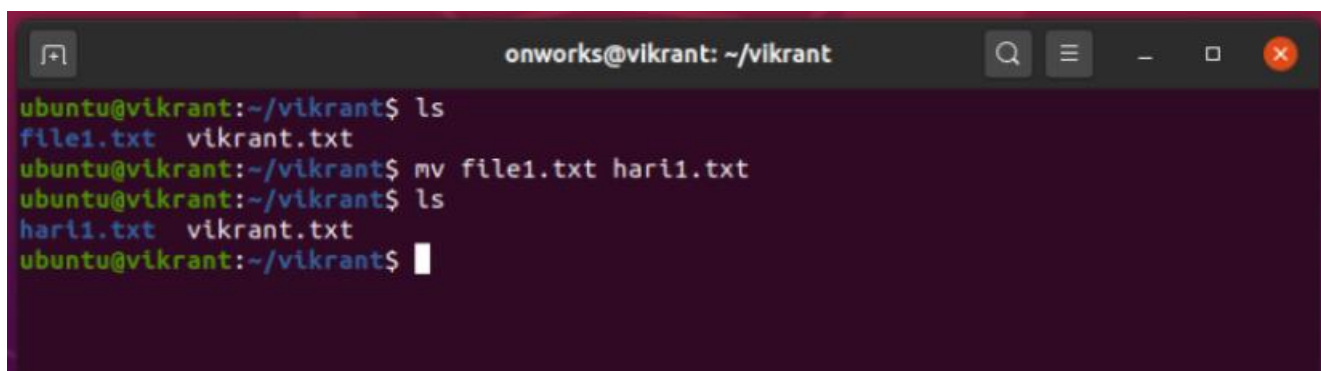
Command : 5 mv -v [filename] [destination]



```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~$ ls
Desktop  Downloads  Music      Public    Templates  vikrant
Documents file1.txt  Pictures   snap      Videos
ubuntu@vikrant:~$ mv file1.txt vikrant
ubuntu@vikrant:~$ ls
Desktop  Downloads  Pictures   snap      Videos
Documents Music      Public     Templates  vikrant
ubuntu@vikrant:~$ cd vikrant
ubuntu@vikrant:~/vikrant$ ls
file1.txt  vikrant.txt
ubuntu@vikrant:~/vikrant$
```

(ii) It renames a file or folder. Command

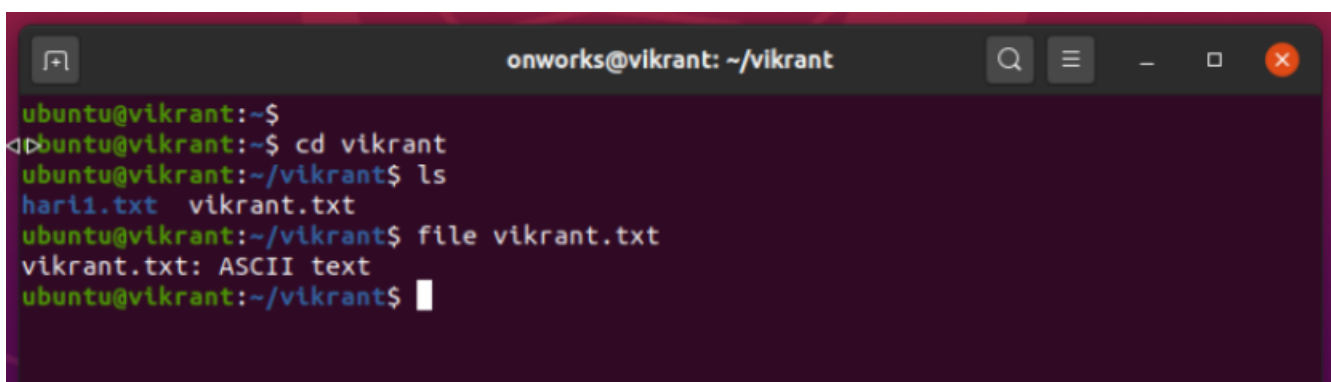
:\$ mv [filename] [new_name_of_file]



```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~/vikrant$ ls
file1.txt  vikrant.txt
ubuntu@vikrant:~/vikrant$ mv file1.txt hari1.txt
ubuntu@vikrant:~/vikrant$ ls
hari1.txt  vikrant.txt
ubuntu@vikrant:~/vikrant$
```

18.File:- *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-asciii").*

\$ file [filename]



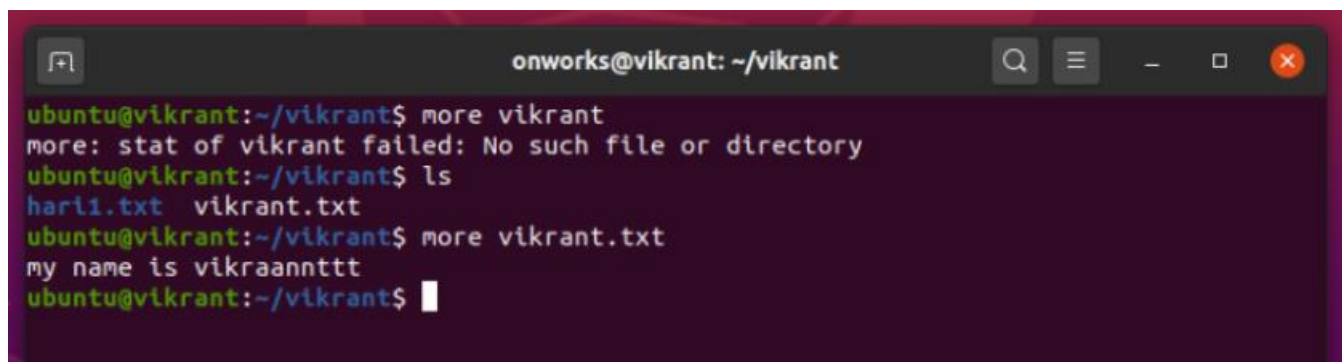
```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~$
ubuntu@vikrant:~$ cd vikrant
ubuntu@vikrant:~/vikrant$ ls
hari1.txt  vikrant.txt
ubuntu@vikrant:~/vikrant$ file vikrant.txt
vikrant.txt: ASCII text
ubuntu@vikrant:~/vikrant$
```


19. more:- *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.*

we can use more commands to see output one by one.

Syntax: more (-options) [-num) (+/pattern) (+linenum) (file_name)

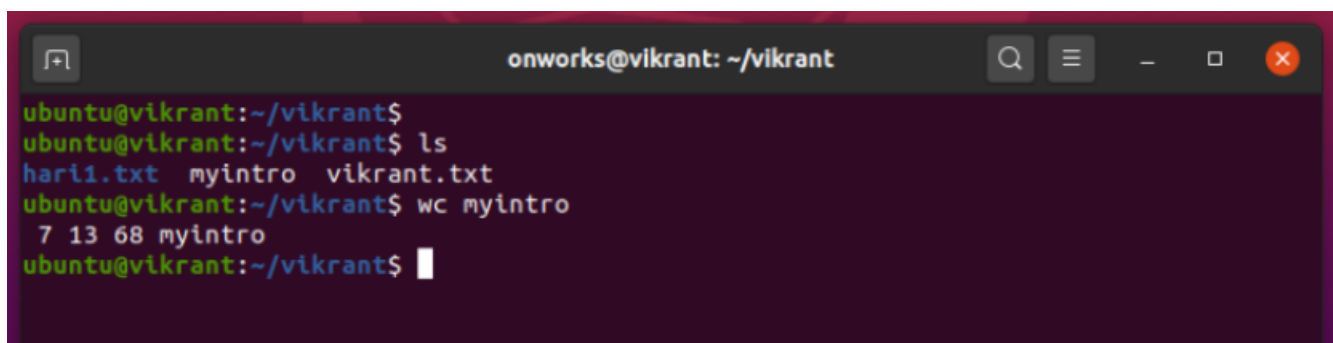
- [options): any option that you want to use in order to change the way the file is displayed. Choose any one from the followings: (-d, 4 -l. p. e, -u)*
 - num): type the number of lines that you want to display per screen.*
 - [pattern) replace the pattern with any string that you want to find in the text file*
 - Flenum): use the Nine number from where you want to start displaying the text content. file_name) name of the file containing the text that you want to display on the screen*
- \$ more [filename]*

A terminal window titled 'onworks@vikrant: ~/vikrant' showing the execution of the 'more' command. The user enters 'more vikrant', which results in an error: 'more: stat of vikrant failed: No such file or directory'. Then, the user enters 'ls', listing files 'hari1.txt' and 'vikrant.txt'. Finally, the user enters 'more vikrant.txt', which displays the content 'my name is vikraannttt' on a single screen.

```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~/vikrant$ more vikrant
more: stat of vikrant failed: No such file or directory
ubuntu@vikrant:~/vikrant$ ls
hari1.txt  vikrant.txt
ubuntu@vikrant:~/vikrant$ more vikrant.txt
my name is vikraannttt
ubuntu@vikrant:~/vikrant$
```

20. WC:- *Wc Command in Linux (Count Number of Lines, Words, and Characters) On Linux and Unix-like operating systems, the wc command allows you to count the number of lines, words, characters, and bytes of each given file or standard input and print the result.*

\$ wc [filename]

A terminal window titled 'onworks@vikrant: ~/vikrant' showing the execution of the 'wc' command. The user enters 'wc myintro', which outputs '7 13 68 myintro', indicating 7 lines, 13 words, and 68 characters in the file 'myintro'.

```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~/vikrant$ wc myintro
7 13 68 myintro
ubuntu@vikrant:~/vikrant$
```

21 df :- The df command (short for disk free), is used to display information related to file systems about total space and available space.

Options for df command:

-3, -all: includes pseudo, duplicate and inaccessible file systems
-k, -block-size=SIZE : Scales sizes by SIZE before printing them,

-h, -human-readable : print sizes in power of 1024

\$ df-h

```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~/vikrant$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1467132         0   1467132   0% /dev
tmpfs           299064      1368    297696   1% /run
/dev/sda5       30313412 6992668  21757864  25% /
tmpfs           1495304         0   1495304   0% /dev/shm
tmpfs            5120          4      5116   1% /run/lock
tmpfs           1495304         0   1495304   0% /sys/fs/cgroup
/dev/loop1       261760    261760         0 100% /snap/gnome-3-34-1804/36
/dev/loop2        63616    63616         0 100% /snap/gtk-common-themes/1506
/dev/loop0        56320    56320         0 100% /snap/core18/1880
/dev/loop3        51072    51072         0 100% /snap/snap-store/467
/dev/loop4       30720    30720         0 100% /snap/snapd/8542
/dev/sda1        523248          4    523244   1% /boot/efi
tmpfs            299060         40    299020   1% /run/user/1000
ubuntu@vikrant:~/vikrant$
```

But If we want display the information in human readable format Sdu -ha

```
onworks@vikrant: ~/vikrant
ubuntu@vikrant:~/vikrant$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1,4G   0  1,4G   0% /dev
tmpfs           293M   1,4M 291M   1% /run
/dev/sda5       29G   6,7G  21G   25% /
tmpfs           1,5G   0  1,5G   0% /dev/shm
tmpfs           5,0M   4,0K 5,0M   1% /run/lock
tmpfs           1,5G   0  1,5G   0% /sys/fs/cgroup
/dev/loop1      256M  256M   0 100% /snap/gnome-3-34-1804/36
/dev/loop2       63M   63M   0 100% /snap/gtk-common-themes/1506
/dev/loop0       55M   55M   0 100% /snap/core18/1880
/dev/loop3       50M   50M   0 100% /snap/snap-store/467
/dev/loop4       30M   30M   0 100% /snap/snapd/8542
/dev/sda1       511M   4,0K 511M   1% /boot/efi
tmpfs           293M   40K 293M   1% /run/user/1000
ubuntu@vikrant:~/vikrant$
```

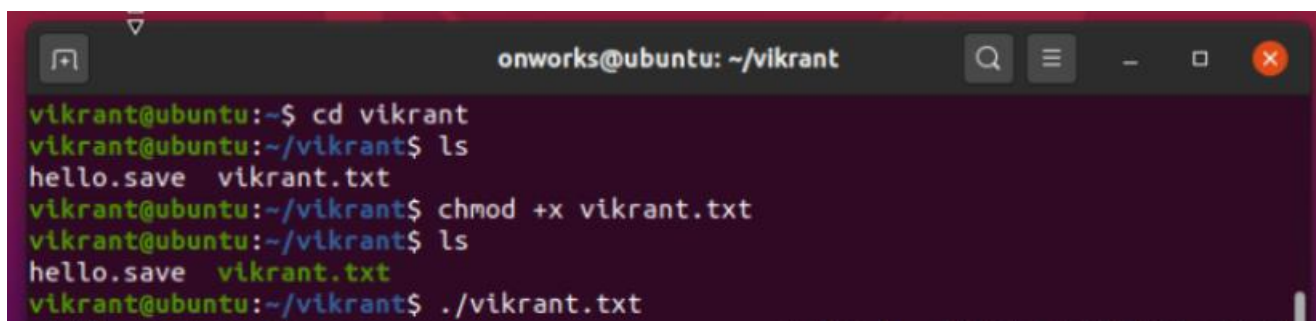
22.chmod:- In Unix-like operating systems, the *chmod* command is used to change the access mode of a file. The name is an abbreviation of change mode.

The modes indicate which permissions are to be granted or removed from the specified classes. There are three basic modes which correspond to the basic permissions

: Permission to read the file.

Permission to write (or delete) the file. Permission to execute the file, or, in the case of a directory, search it.

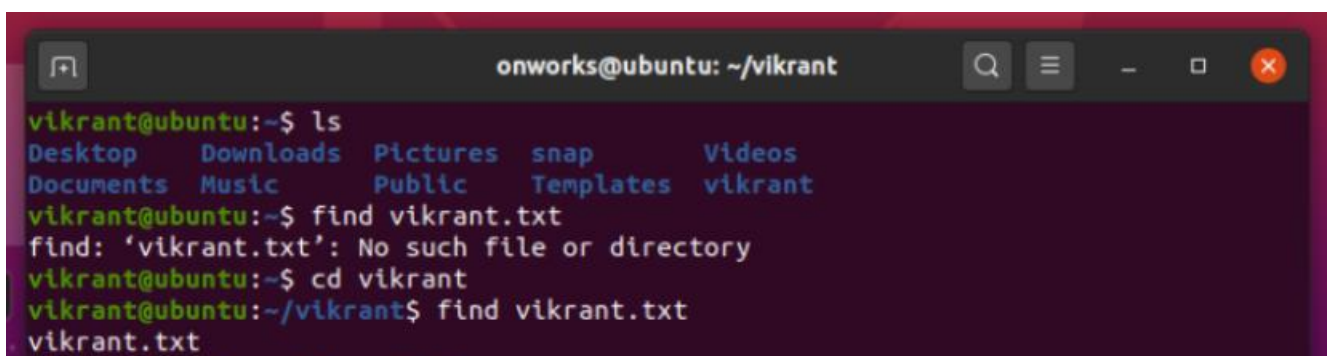
\$ *chmod* [reference] [operator] (mode) file

A terminal window titled 'onworks@ubuntu: ~/vikrant' showing a series of commands. The user navigates to the 'vikrant' directory, lists files ('hello.save' and 'vikrant.txt'), changes permissions of 'vikrant.txt' to '+x', lists files again, and finally executes './vikrant.txt'.

```
onworks@ubuntu: ~/vikrant
vikrant@ubuntu:~$ cd vikrant
vikrant@ubuntu:~/vikrant$ ls
hello.save  vikrant.txt
vikrant@ubuntu:~/vikrant$ chmod +x vikrant.txt
vikrant@ubuntu:~/vikrant$ ls
hello.save  vikrant.txt
vikrant@ubuntu:~/vikrant$ ./vikrant.txt
```

23 find :- The *find* command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.

\$ *find* (where to start searching from) [expression determines what to find] [/options] [what to find]

A terminal window titled 'onworks@ubuntu: ~/vikrant' showing the 'find' command. The user lists the root directory, then searches for 'vikrant.txt' from the root (resulting in 'No such file or directory'), then changes to the 'vikrant' directory and successfully finds 'vikrant.txt'.

```
onworks@ubuntu: ~/vikrant
vikrant@ubuntu:~$ ls
Desktop  Downloads  Pictures  snap      Videos
Documents Music      Public    Templates vikrant
vikrant@ubuntu:~$ find vikrant.txt
find: 'vikrant.txt': No such file or directory
vikrant@ubuntu:~$ cd vikrant
vikrant@ubuntu:~/vikrant$ find vikrant.txt
vikrant.txt
```


24. free :- free command can be used to check exactly what amount of storage is free and used in physical as well as swap memory in the system. There are also some options to use with free command such as you can use free -b to view the results in bytes, free -k to display the available and used in memory in kilobytes, free -m to view in megabytes, free -g to view results in gigabytes and free -tera to view the results in terabytes.

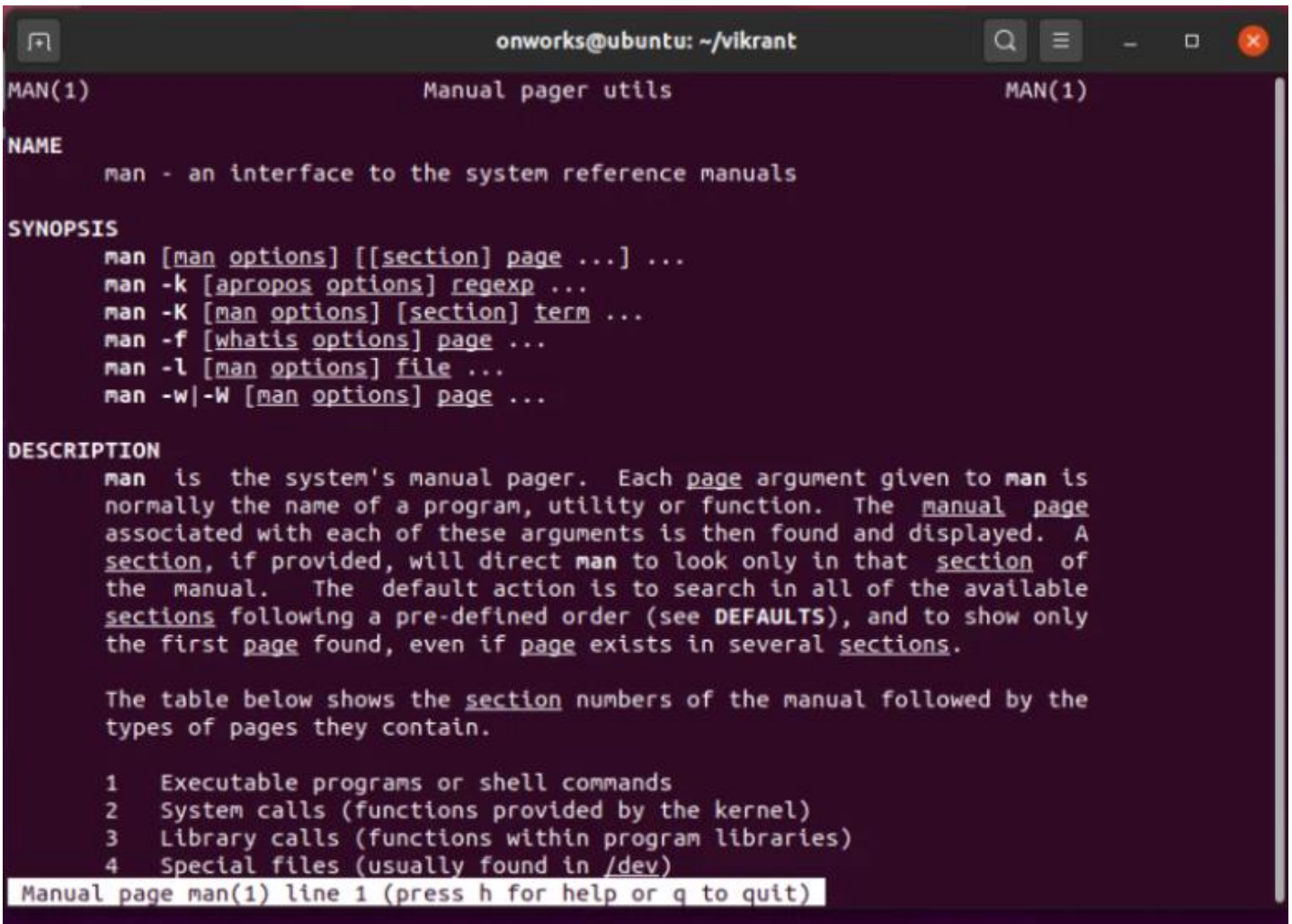
```
Thunderbird Mail onworks@ubuntu: ~
vkrant@ubuntu:~$ free
              total        used        free      shared  buff/cache   available
Mem:           2990612      1029804      189964        10828      1770844      1764768
Swap:          1435264          524      1434740
vkrant@ubuntu:~$
```

25. top :- top is simple but useful command to monitor all the ongoing processes on the Linux system with the user name, priority level, unique process id and shared memory by each task.

```
onworks@ubuntu: ~
top - 11:10:15 up 20 min, 1 user, load average: 1,31, 1,21, 0,91
Tasks: 178 total, 2 running, 176 sleeping, 0 stopped, 0 zombie
%Cpu(s): 40,2 us, 9,3 sy, 0,0 ni, 49,3 id, 0,8 wa, 0,0 hi, 0,2 st, 0,2 st
MiB Mem : 2920,5 total, 424,6 free, 1087,9 used, 1408,0 buff/cache
MiB Swap: 1401,6 total, 1399,9 free, 1,8 used. 1643,0 avail Mem

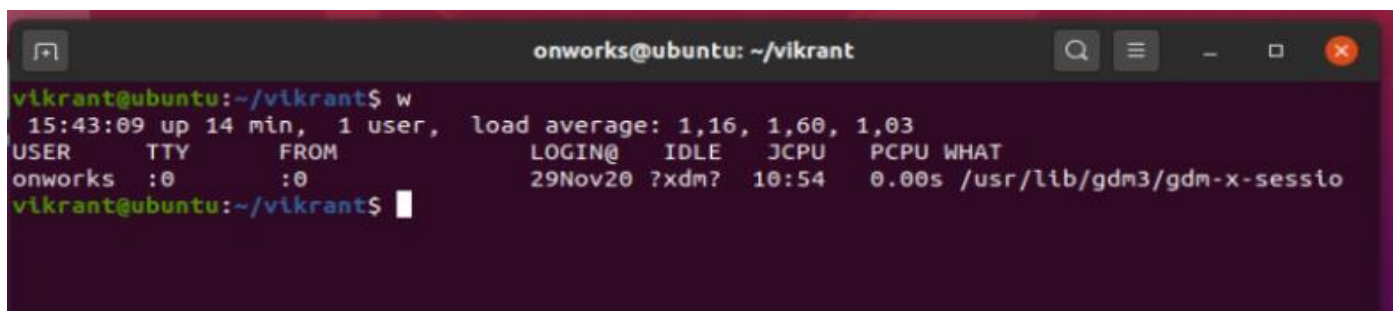
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 15639 root        20   0   395904   164088   7872 R   30,6   5,5   0:00.92 unattended-upgr
  3101 root        20   0   363136   192132   67992 S    3,0   6,4  11:38.58 unattended-upgr
   182 root         0 -20       0        0        0 I    0,7   0,0   0:02.18 kworker/0:1H-kblockd
     1 root        20   0   103132   12808    8420 S    0,3   0,4   0:03.47 systemd
    10 root        20   0         0         0        0 I    0,3   0,0   0:01.11 rcu_sched
   192 root        20   0         0         0        0 S    0,3   0,0   0:00.75 jbd2/sda5-8
   773 onworks     20   0   318936   10100    8744 S    0,3   0,3   0:00.11 goa-identity-se
  1052 onworks     20   0 4149172 318952 115420 S    0,3  10,7   0:25.89 gnome-shell
  2382 root        20   0    2608    1584    1488 S    0,3   0,1   0:00.05 sh
     2 root        20   0         0         0        0 S    0,0   0,0   0:00.00 kthreadd
     3 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 rcu_gp
     4 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 rcu_par_gp
     6 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 kworker/0:0H-kblockd
     8 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 mm_percpu_wq
     9 root        20   0         0         0        0 S    0,0   0,0   0:00.11 ksoftirqd/0
    11 root        rt    0         0         0        0 S    0,0   0,0   0:00.01 migration/0
    12 root       -51   0         0         0        0 S    0,0   0,0   0:00.00 idle_inject/0
    14 root        20   0         0         0        0 S    0,0   0,0   0:00.00 cpuhp/0
    15 root        20   0         0         0        0 S    0,0   0,0   0:00.00 cpuhp/1
    16 root       -51   0         0         0        0 S    0,0   0,0   0:00.00 idle_inject/1
    17 root        rt    0         0         0        0 S    0,0   0,0   0:00.17 migration/1
    18 root        20   0         0         0        0 S    0,0   0,0   0:00.20 ksoftirqd/1
    20 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 kworker/1:0H-kblockd
    21 root        20   0         0         0        0 S    0,0   0,0   0:00.00 kdevtmpfs
    22 root         0 -20       0         0        0 I    0,0   0,0   0:00.00 netns
```

26. man :- Here man stands for user manual and as the name suggests man <command name> will display the user manual for the particular command. It will display name of the command, ways in which command can be used and description of the command.



```
onworks@ubuntu: ~/vikrant
MAN(1) Manual pager utils MAN(1)
NAME
  man - an interface to the system reference manuals
SYNOPSIS
  man [man options] [[section] page ...] ...
  man -k [apropos options] regexp ...
  man -K [man options] [section] term ...
  man -f [whatis options] page ...
  man -l [man options] file ...
  man -w|-W [man options] page ...
DESCRIPTION
  man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.
  The table below shows the section numbers of the manual followed by the types of pages they contain.
  1 Executable programs or shell commands
  2 System calls (functions provided by the kernel)
  3 Library calls (functions within program libraries)
  4 Special files (usually found in /dev)
Manual page man(1) line 1 (press h for help or q to quit)
```

27. w :- w is the short and simple command which will help you view the list of currently logged in users.



```
onworks@ubuntu: ~/vikrant
vikrant@ubuntu:~/vikrant$ w
 15:43:09 up 14 min,  1 user,  load average: 1.16, 1.60, 1.03
USER    TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
onworks  :0        :0            29Nov20  ?xdm?  10:54   0.00s  /usr/lib/gdm3/gdm-x-sessio
vikrant@ubuntu:~/vikrant$
```

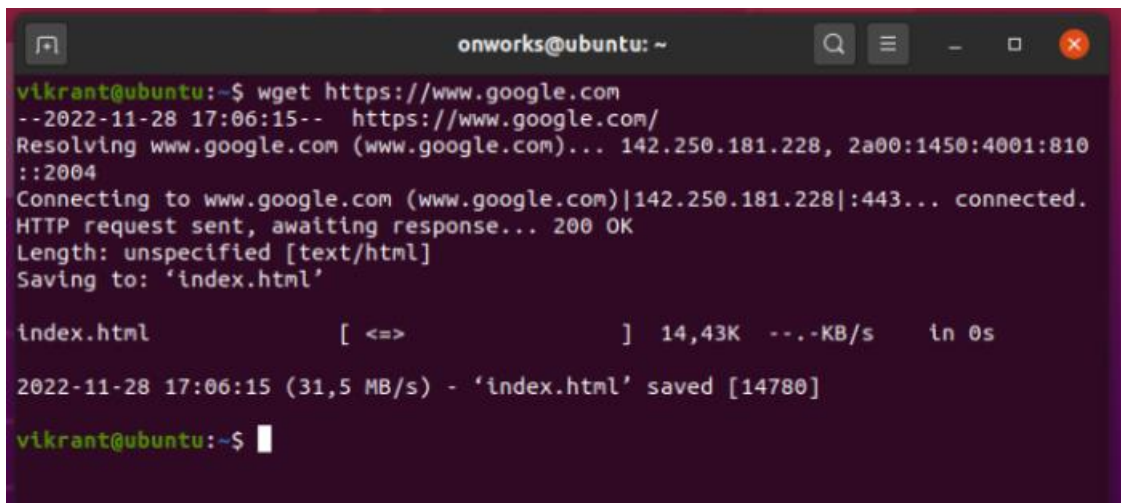

28. *lscpu* :- This command will display all the CPU architecture information such as threads, sockets, cores and CPU count.

```
onworks@ubuntu: ~/vikrant
vikrant@ubuntu:~/vikrant$ lscpu
Architecture:                x86_64
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
Address sizes:               40 bits physical, 48 bits virtual
CPU(s):                      2
On-line CPU(s) list:         0,1
Thread(s) per core:          1
Core(s) per socket:          1
Socket(s):                   2
NUMA node(s):                1
Vendor ID:                   GenuineIntel
CPU family:                   15
Model:                       6
Model name:                  Common KVM processor
Stepping:                    1
CPU MHz:                     3399.994
BogoMIPS:                    6799.98
Hypervisor vendor:           KVM
Virtualization type:         full
L1d cache:                   64 KiB
L1i cache:                   64 KiB
L2 cache:                    8 MiB
L3 cache:                    32 MiB
NUMA node0 CPU(s):           0,1
Vulnerability Itlb multihit:  KVM: Vulnerable
Vulnerability L1tf:           Mitigation; PTE Inversion
Vulnerability Mds:            Vulnerable: Clear CPU buffers attempted, no microcode; SMT
                               Host state unknown
Vulnerability Meltdown:       Mitigation; PTI
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1:     Mitigation; usercopy/swapgs barriers and __user pointer sa
                               nitization
```

29. *lshw* :- *sudo lshw* command can be used to invoke detailed hardware information of the system on which Linux is running. It gives you every small detail about hardware, just try it.

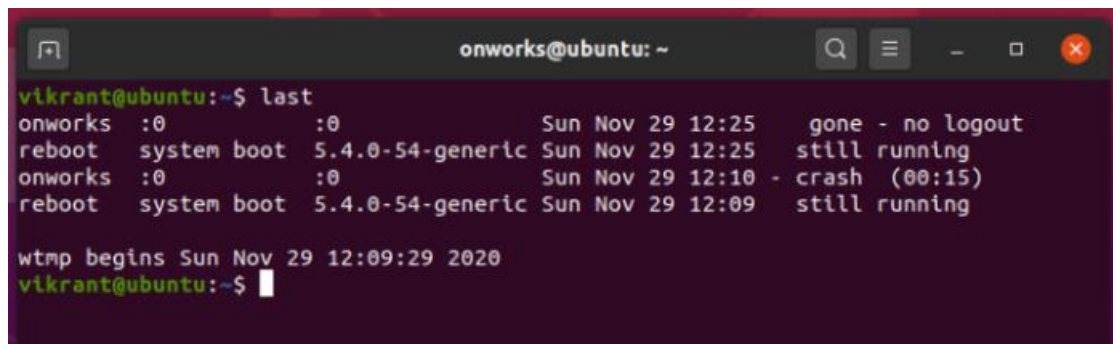
```
onworks@ubuntu: ~
vikrant@ubuntu:~$ lshw
WARNING: you should run this program as super-user.
ubuntu
  description: Computer
  width: 64 bits
  capabilities: smp vsyscall32
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: System memory
  physical id: 0
  size: 3GiB
*-cpu:0
  product: Common KVM processor
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0
  width: 64 bits
  capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic
sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx x86-64 con
stant_tsc nopl xtopology cpuid tsc_known_freq pni cx16 x2apic hypervisor pti
*-cpu:1
  product: Common KVM processor
  vendor: Intel Corp.
  physical id: 2
  bus info: cpu@1
  width: 64 bits
  capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic
sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx x86-64 con
stant_tsc nopl xtopology cpuid tsc_known_freq pni cx16 x2apic hypervisor pti
*-pci
  description: Host bridge
```

30. wget :- *wget <file path> is very useful command to download any file from the internet and best part is download works in background so that you can continue working on your task.*



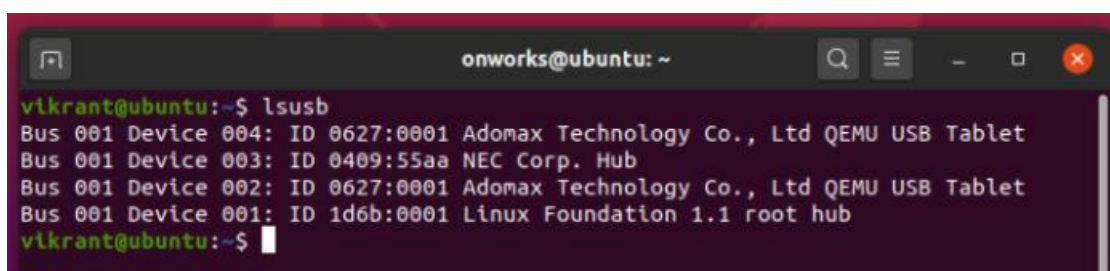
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ wget https://www.google.com  
--2022-11-28 17:06:15-- https://www.google.com/  
Resolving www.google.com (www.google.com)... 142.250.181.228, 2a00:1450:4001:810::2004  
Connecting to www.google.com (www.google.com)|142.250.181.228|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/html]  
Saving to: 'index.html'  
  
index.html          [ <=>          ] 14,43K  --.-KB/s   in 0s  
  
2022-11-28 17:06:15 (31,5 MB/s) - 'index.html' saved [14780]  
  
vikrant@ubuntu:~$
```

31. last :- *When executed last command will display the list of last logged in users into the system as an output in Linux Terminal.*



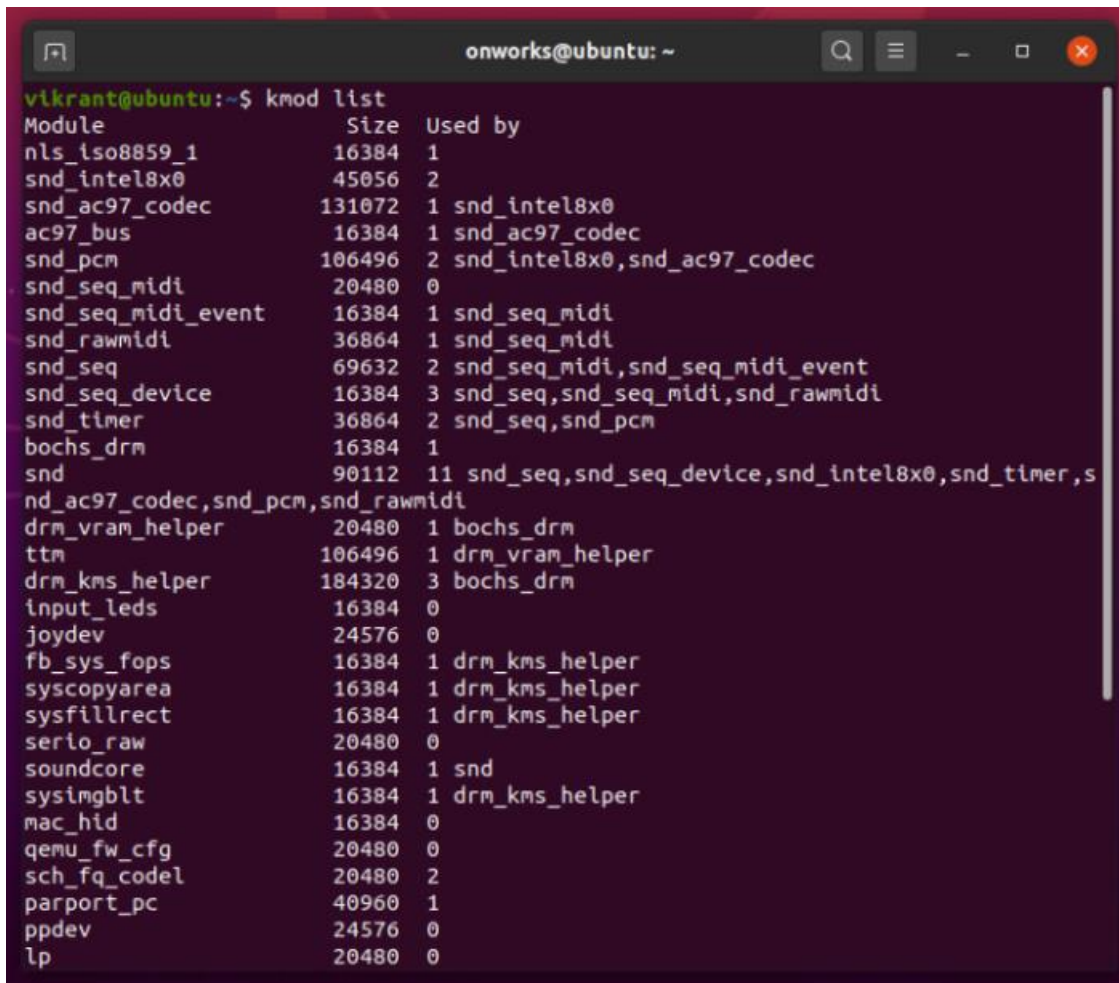
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ last  
onworks  :0      :0      Sun Nov 29 12:25   gone - no logout  
reboot   system boot  5.4.0-54-generic Sun Nov 29 12:25   still running  
onworks  :0      :0      Sun Nov 29 12:10   crash (00:15)  
reboot   system boot  5.4.0-54-generic Sun Nov 29 12:09   still running  
  
wtmp begins Sun Nov 29 12:09:29 2020  
vikrant@ubuntu:~$
```

32. lsusb :- *lsusb command will show information about all the USB buses connected to the hardware and external USB devices connected to them as you can see in screenshot below.*



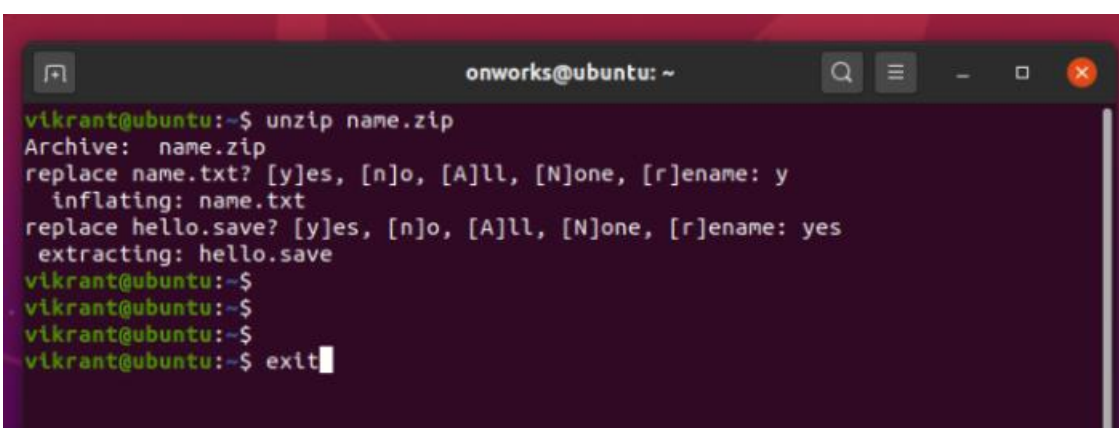
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ lsusb  
Bus 001 Device 004: ID 0627:0001 Adomax Technology Co., Ltd QEMU USB Tablet  
Bus 001 Device 003: ID 0409:55aa NEC Corp. Hub  
Bus 001 Device 002: ID 0627:0001 Adomax Technology Co., Ltd QEMU USB Tablet  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
vikrant@ubuntu:~$
```

33. kmod :- You can use *kmod list* command to manage all the Linux Kernel modules as this command will display all the currently loaded modules on the system.



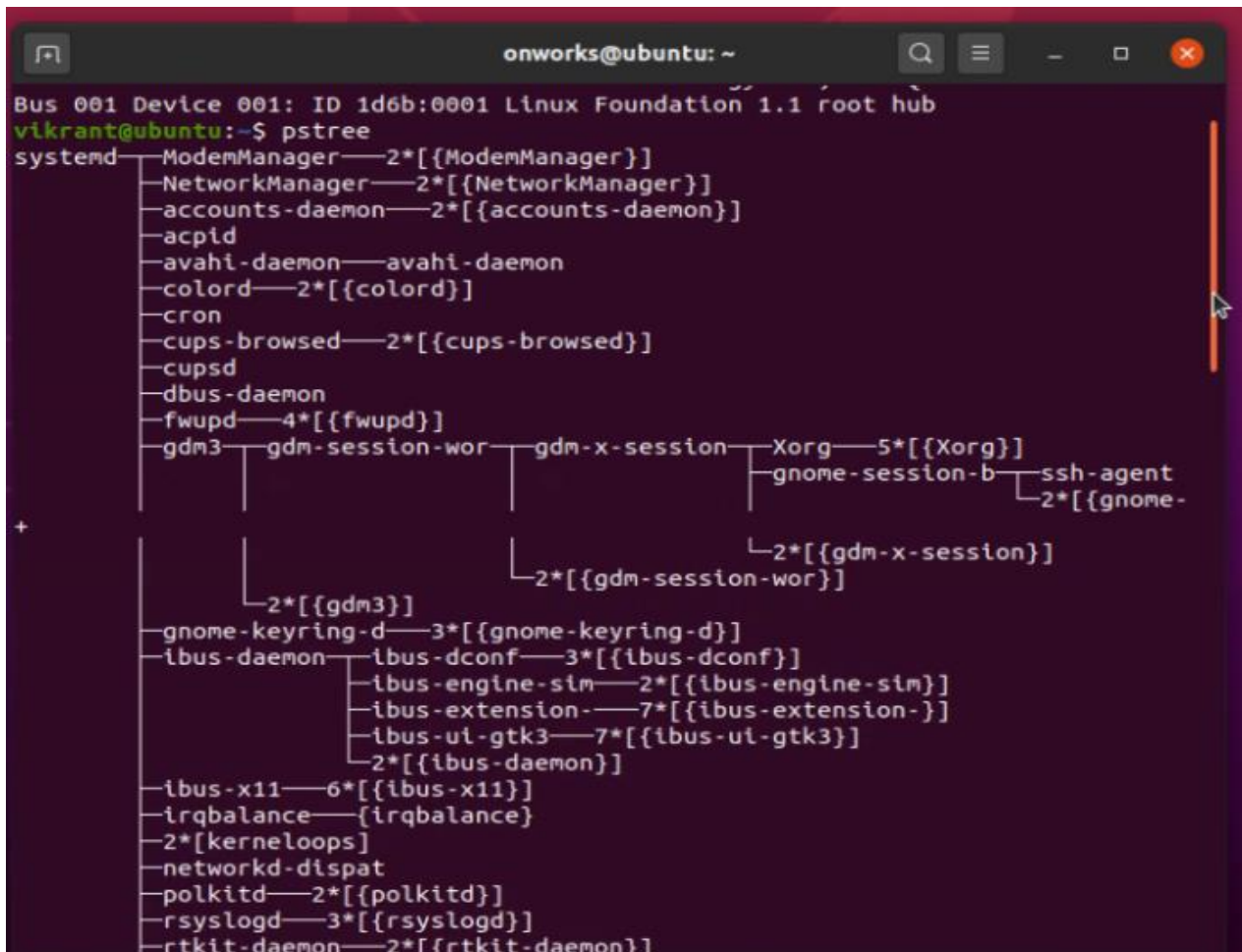
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ kmod list  
Module      Size      Used by  
nls_iso8859_1 16384      1  
snd_intel8x0 45056      2  
snd_ac97_codec 131072     1 snd_intel8x0  
ac97_bus      16384      1 snd_ac97_codec  
snd_pcm       106496     2 snd_intel8x0,snd_ac97_codec  
snd_seq_midi  20480      0  
snd_seq_midi_event 16384      1 snd_seq_midi  
snd_rawmidi   36864      1 snd_seq_midi  
snd_seq       69632     2 snd_seq_midi,snd_seq_midi_event  
snd_seq_device 16384      3 snd_seq,snd_seq_midi,snd_rawmidi  
snd_timer     36864      2 snd_seq,snd_pcm  
bochs_drm     16384      1  
snd           90112     11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi  
drm_vram_helper 20480      1 bochs_drm  
ttm           106496     1 drm_vram_helper  
drm_kms_helper 184320     3 bochs_drm  
input_leds    16384      0  
joydev        24576      0  
fb_sys_fops   16384      1 drm_kms_helper  
syscopyarea   16384      1 drm_kms_helper  
sysfillrect   16384      1 drm_kms_helper  
serio_raw     20480      0  
soundcore     16384      1 snd  
sysimgblt     16384      1 drm_kms_helper  
mac_hid       16384      0  
qemu_fw_cfg   20480      0  
sch_fq_codel   20480      2  
parport_pc    40960      1  
ppdev         24576      0  
lp            20480      0
```

34. exit:- *exit* command can be used to close the Terminal shell window directly from the command-line.



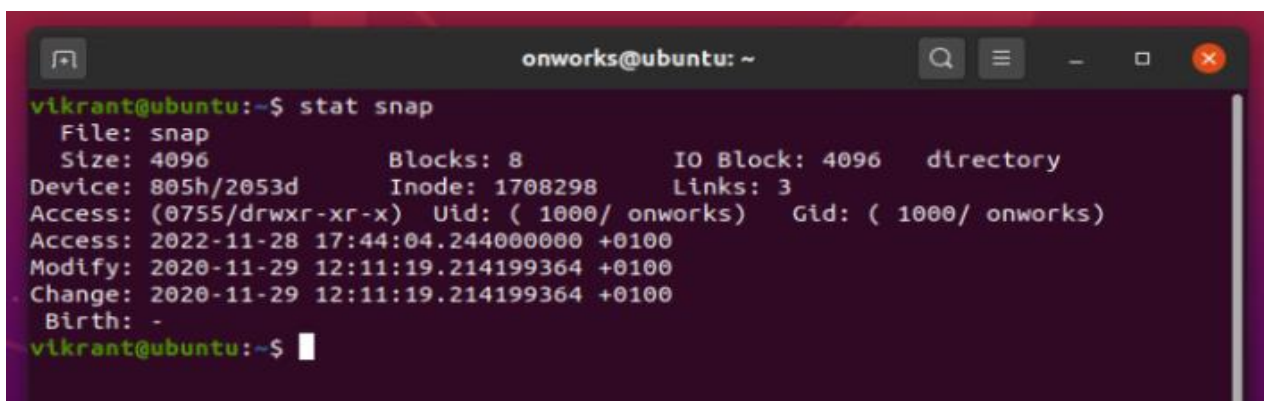
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ unzip name.zip  
Archive: name.zip  
replace name.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y  
  inflating: name.txt  
replace hello.save? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes  
  extracting: hello.save  
vikrant@ubuntu:~$  
vikrant@ubuntu:~$  
vikrant@ubuntu:~$  
vikrant@ubuntu:~$ exit
```


35. pstree: - *pstree command displays all the currently running processes in the tree format on Linux Terminal window.*



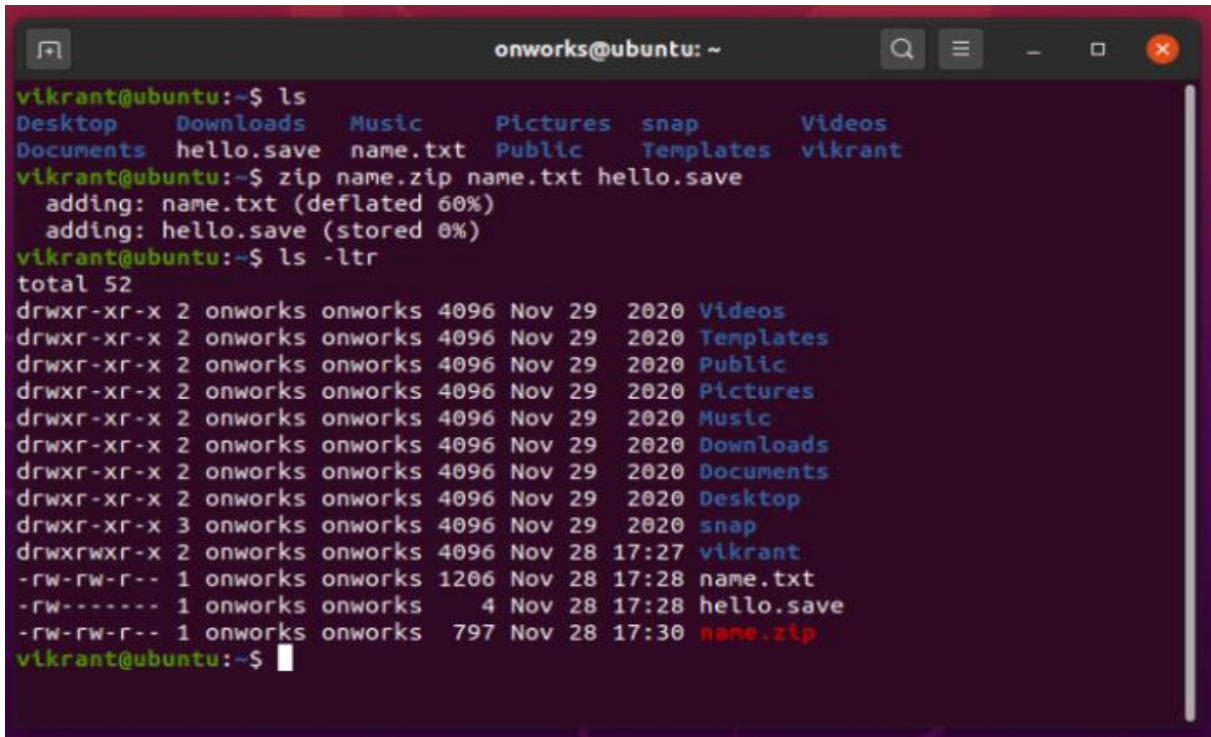
```
onworks@ubuntu: ~
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
vikrant@ubuntu:~$ pstree
systemd--ModemManager--2*[{ModemManager}]
      |NetworkManager--2*[{NetworkManager}]
      |accounts-daemon--2*[{accounts-daemon}]
      |acpid
      |avahi-daemon--avahi-daemon
      |colord--2*[{colord}]
      |cron
      |cups-browsed--2*[{cups-browsed}]
      |cupsd
      |dbus-daemon
      |fwupd--4*[{fwupd}]
      |gdm3--gdm-session-wor--gdm-x-session--Xorg--5*[{Xorg}]
      |      |      |      |gnome-session-b--ssh-agent
      |      |      |      |      |2*[{gnome-
      |      |      |      |      |      |
      |      |      |      |      |2*[{gdm-x-session}]
      |      |      |      |      |2*[{gdm-session-wor}]
      |      |      |      |      |2*[{gdm3}]
      |      |      |      |      |gnome-keyring-d--3*[{gnome-keyring-d}]
      |      |      |      |      |ibus-daemon--ibus-dconf--3*[{ibus-dconf}]
      |      |      |      |      |      |ibus-engine-sim--2*[{ibus-engine-sim}]
      |      |      |      |      |      |ibus-extension--7*[{ibus-extension-}]
      |      |      |      |      |      |ibus-ui-gtk3--7*[{ibus-ui-gtk3}]
      |      |      |      |      |      |2*[{ibus-daemon}]
      |      |      |      |      |ibus-x11--6*[{ibus-x11}]
      |      |      |      |      |irqbalance--[{irqbalance}]
      |      |      |      |      |2*[{kerneloops}]
      |      |      |      |      |networkd-dispat
      |      |      |      |      |polkitd--2*[{polkitd}]
      |      |      |      |      |rsyslogd--3*[{rsyslogd}]
      |      |      |      |      |rtkit-daemon--2*[{rtkit-daemon}]
```

36. stat :- *You can view the status of a file or an entire file system using stat <file or file system name> command in Linux Terminal. You can also use other options as listed in the screenshot.*



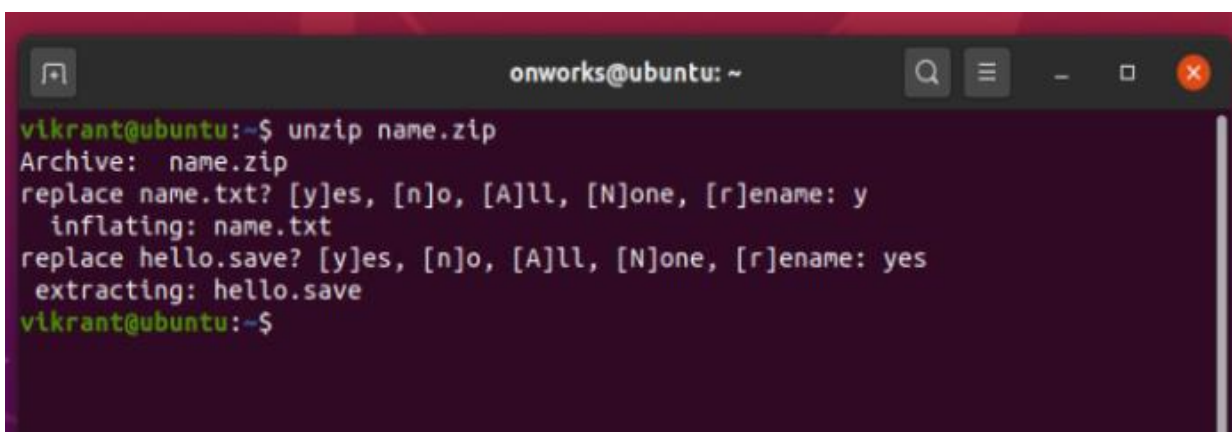
```
onworks@ubuntu: ~
vikrant@ubuntu:~$ stat snap
File: snap
Size: 4096      Blocks: 8      IO Block: 4096   directory
Device: 805h/2053d Inode: 1708298 Links: 3
Access: (0755/drwxr-xr-x)  Uid: ( 1000/ onworks)   Gid: ( 1000/ onworks)
Access: 2022-11-28 17:44:04.244000000 +0100
Modify: 2020-11-29 12:11:19.214199364 +0100
Change: 2020-11-29 12:11:19.214199364 +0100
Birth: -
vikrant@ubuntu:~$
```

37. zip :- You can use zip command to compress one or more files as you can see in the screenshot below. It is simple but useful command to compress any number of files in a go.



```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ ls  
Desktop Downloads Music Pictures snap Videos  
Documents hello.save name.txt Public Templates vikrant  
vikrant@ubuntu:~$ zip name.zip name.txt hello.save  
adding: name.txt (deflated 60%)  
adding: hello.save (stored 0%)  
vikrant@ubuntu:~$ ls -ltr  
total 52  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Videos  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Templates  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Public  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Pictures  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Music  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Downloads  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Documents  
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Desktop  
drwxr-xr-x 3 onworks onworks 4096 Nov 29 2020 snap  
drwxrwxr-x 2 onworks onworks 4096 Nov 28 17:27 vikrant  
-rw-rw-r-- 1 onworks onworks 1206 Nov 28 17:28 name.txt  
-rw----- 1 onworks onworks 4 Nov 28 17:28 hello.save  
-rw-rw-r-- 1 onworks onworks 797 Nov 28 17:30 name.zip  
vikrant@ubuntu:~$
```

38. unzip :- To extract files from compressed zip file use unzip <file name> command in Terminal shell. You can also use this command to extract files from multiple compressed files from the particular directory.



```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ unzip name.zip  
Archive: name.zip  
replace name.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y  
inflating: name.txt  
replace hello.save? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes  
extracting: hello.save  
vikrant@ubuntu:~$
```

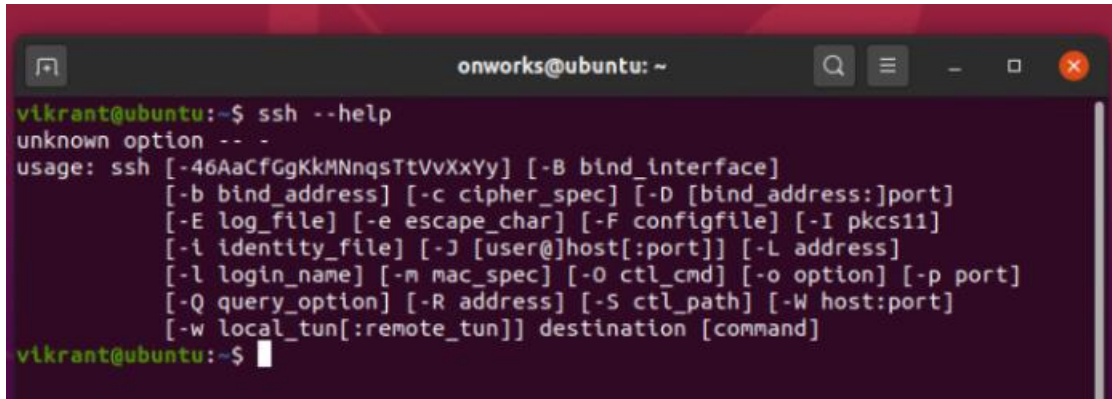

39. usermod :- If you want edit or modify attributes of already created user account then `usermod <options> login` is the best command for you.

```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ usermod  
Usage: usermod [options] LOGIN  
  
Options:  
-b, --badnames          allow bad names  
-c, --comment COMMENT   new value of the GECOS field  
-d, --home HOME_DIR     new home directory for the user account  
-e, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE  
-f, --inactive INACTIVE set password inactive after expiration to INACTIVE  
  
-g, --gid GROUP          force use GROUP as new primary group  
-G, --groups GROUPS     new list of supplementary GROUPS  
-a, --append             append the user to the supplemental GROUPS mentioned by the -G option without removing the user from other groups  
  
-h, --help              display this help message and exit  
-l, --login NEW_LOGIN   new value of the login name  
-L, --lock              lock the user account  
-m, --move-home         move contents of the home directory to the new location (use only with -d)  
  
-o, --non-unique         allow using duplicate (non-unique) UID  
-p, --password PASSWORD use encrypted password for the new password  
-R, --root CHROOT_DIR   directory to chroot into  
-P, --prefix PREFIX_DIR prefix directory where are located the /etc/* fi
```

40. pmap :- `pmap <pid>` command display the memory map of the pid you provide. You can also view memory map for multiple processes.

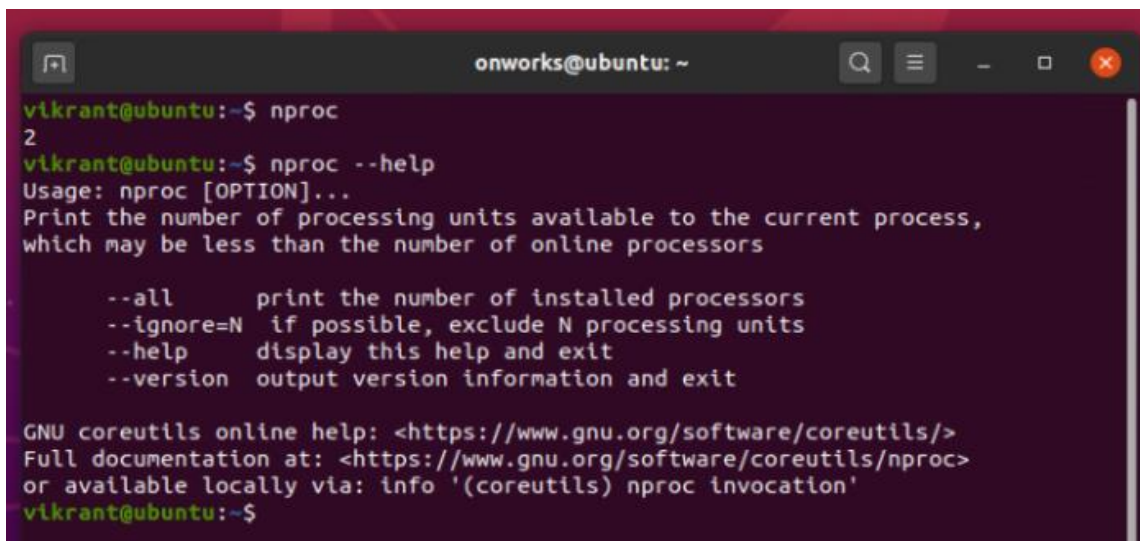
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ pmap  
Usage:  
pmap [options] PID [PID ...]  
  
Options:  
-x, --extended          show details  
-X                      show even more details  
WARNING: format changes according to /proc/PID/smaps  
-XX                     show everything the kernel provides  
-c, --read-rc           read the default rc  
-C, --read-rc-from=<file> read the rc from file  
--create-rc             create new default rc  
--create-rc-to=<file>   create new rc to file  
NOTE: pid arguments are not allowed with -n, -N  
-d, --device            show the device format  
-q, --quiet            do not display header and footer  
-p, --show-path         show path in the mapping  
-A, --range=<low>[,<high>] limit results to the given range  
  
-h, --help              display this help and exit  
-V, --version           output version information and exit  
  
For more details see man(1)
```

41. ssh :- *ssh acronym for Secure Shell is protocol which is used to securely connect to host system. ssh username@host<IP/Domain Name> is the command to connect to host computer as a user.*



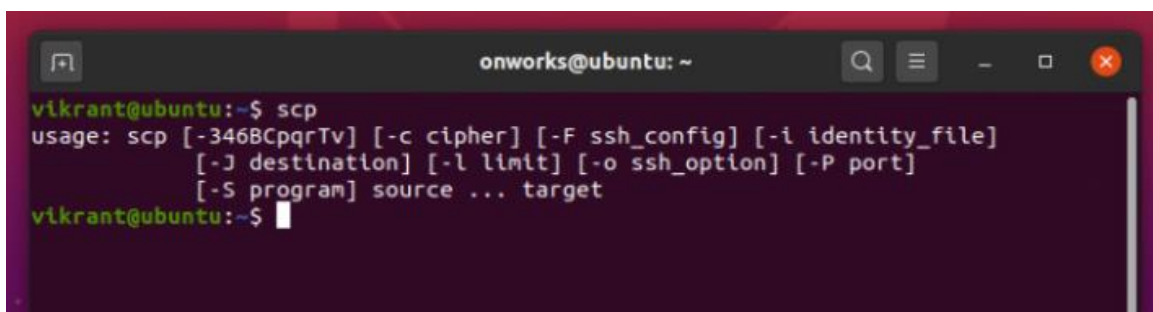
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ ssh --help  
unknown option --  
usage: ssh [-46AaCfGgKkMnQsTtVvXxYy] [-B bind_interface]  
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]  
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]  
          [-i identity_file] [-J [user@]host[:port]] [-L address]  
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]  
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]  
          [-w local_tun[:remote_tun]] destination [command]  
vikrant@ubuntu:~$
```

42. nproc :- *nproc [option] command will display the number of processing units allotted to the currently running process.*



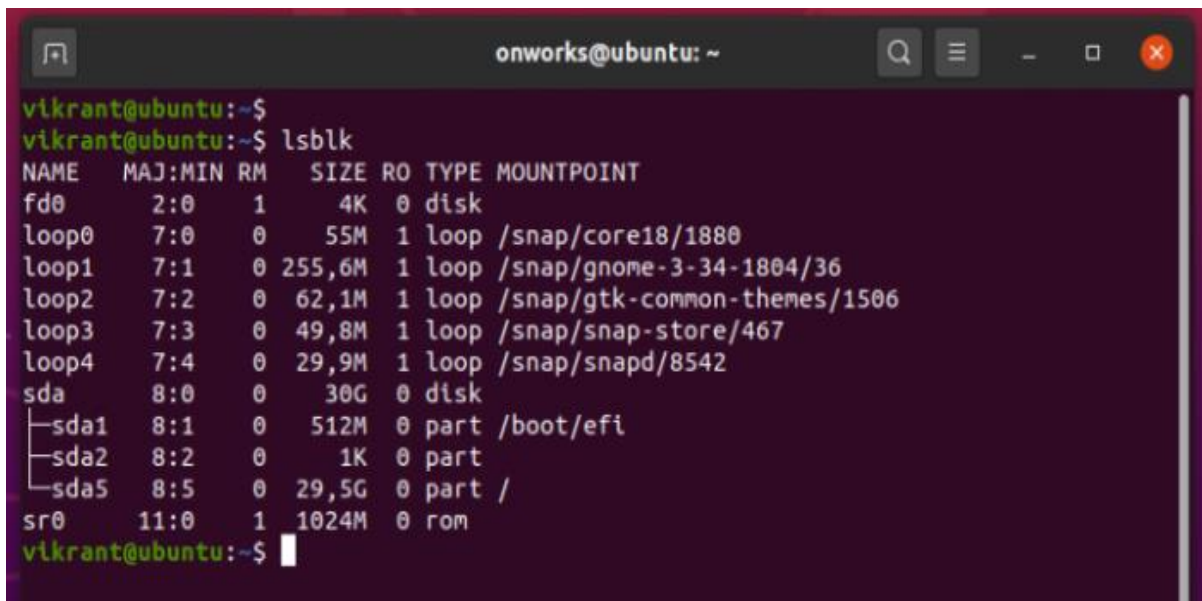
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ nproc  
2  
vikrant@ubuntu:~$ nproc --help  
Usage: nproc [OPTION]...  
Print the number of processing units available to the current process,  
which may be less than the number of online processors  
  
    --all          print the number of installed processors  
    --ignore=N    if possible, exclude N processing units  
    --help        display this help and exit  
    --version      output version information and exit  
  
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>  
Full documentation at: <https://www.gnu.org/software/coreutils/nproc>  
or available locally via: info '(coreutils) nproc invocation'  
vikrant@ubuntu:~$
```

43. scp :- *scp acronym for Secure Copy is the Linux command which can be used to copy files and directories between hosts on the network.*



```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ scp  
usage: scp [-346BCpqRTv] [-c cipher] [-F ssh_config] [-i identity_file]  
          [-J destination] [-l limit] [-o ssh_option] [-P port]  
          [-S program] source ... target  
vikrant@ubuntu:~$
```

44. lsblk :- *lsblk command reads the sysfs filesystem and displays the block device information on the Terminal window.*



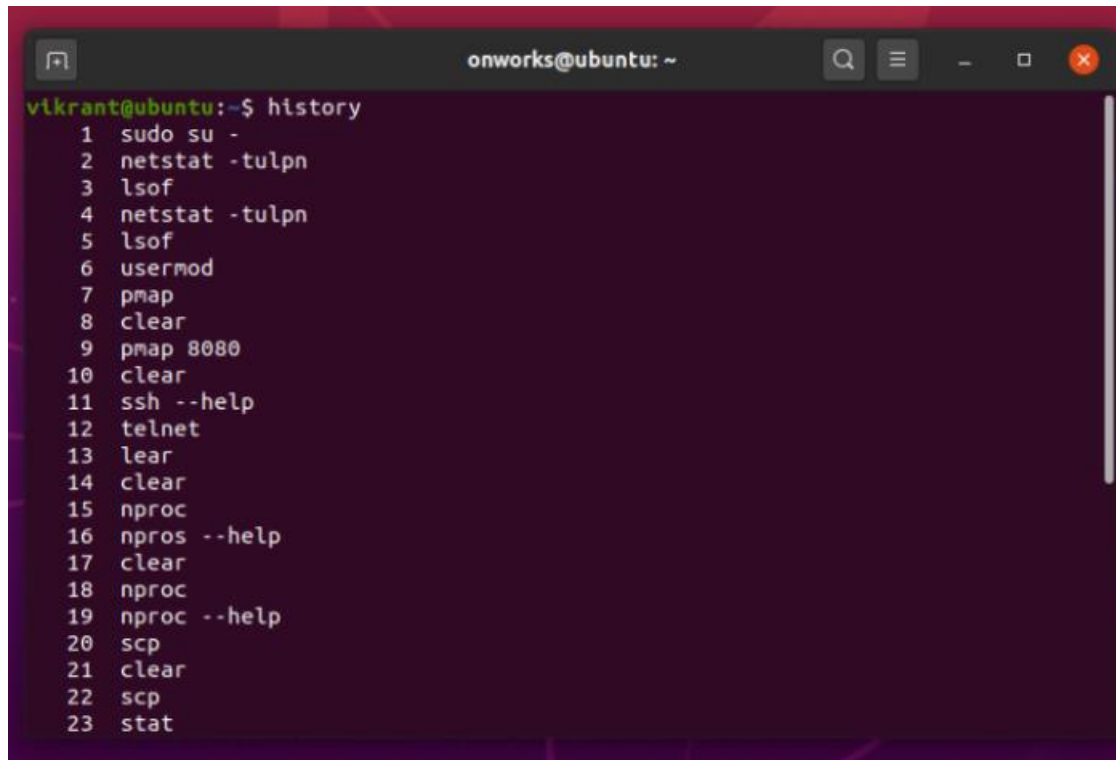
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$  
vikrant@ubuntu:~$ lsblk  
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT  
fd0          2:0    1     4K  0 disk  
loop0        7:0    0    55M  1 loop /snap/core18/1880  
loop1        7:1    0 255,6M  1 loop /snap/gnome-3-34-1804/36  
loop2        7:2    0   62,1M  1 loop /snap/gtk-common-themes/1506  
loop3        7:3    0   49,8M  1 loop /snap/snap-store/467  
loop4        7:4    0   29,9M  1 loop /snap/snapd/8542  
sda          8:0    0    30G  0 disk  
├─sda1       8:1    0   512M  0 part /boot/efi  
├─sda2       8:2    0      1K  0 part  
└─sda5       8:5    0   29,5G  0 part /  
sr0         11:0    1  1024M  0 rom  
vikrant@ubuntu:~$
```

45.hdparm :- *Using hdparm command you can handle hard disk and other disk devices in the Linux using Terminal shell.*



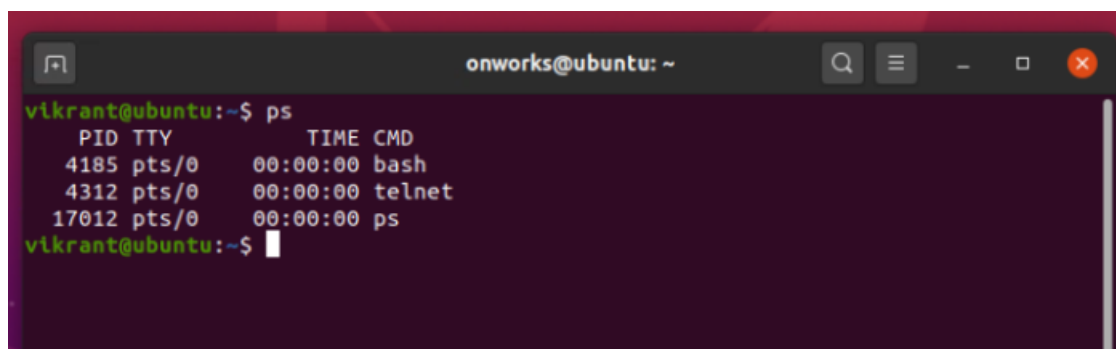
```
onworks@ubuntu: ~  
vikrant@ubuntu:~$ hdparm  
  
hdparm - get/set hard disk parameters - version v9.58, by Mark Lord.  
  
clue=6  
Usage:  hdparm [options] [device ...]  
  
Options:  
-a  Get/set fs readahead  
-A  Get/set the drive look-ahead flag (0/1)  
-b  Get/set bus state (0 == off, 1 == on, 2 == tristate)  
-B  Set Advanced Power Management setting (1-255)  
-c  Get/set IDE 32-bit IO setting  
-C  Check drive power mode status  
-d  Get/set using_dma flag  
-D  Enable/disable drive defect management  
-E  Set cd/dvd drive speed  
-f  Flush buffer cache for device on exit  
-F  Flush drive write cache  
-g  Display drive geometry  
-h  Display terse usage information  
-H  Read temperature from drive (Hitachi only)  
-i  Display drive identification  
-I  Detailed/current information directly from drive
```


46. history :- When fired into Terminal shell, history command will list all the commands used by you in serial numbered form. Using exclamation mark ! and serial number of the command will help you execute that particular command without need to writing whole command in the terminal.

A terminal window titled 'onworks@ubuntu: ~' showing the output of the 'history' command. The output is a list of 23 commands, each preceded by a line number from 1 to 23. The commands are: 1 sudo su -, 2 netstat -tulpn, 3 lsof, 4 netstat -tulpn, 5 lsof, 6 usermod, 7 pmap, 8 clear, 9 pmap 8080, 10 clear, 11 ssh --help, 12 telnet, 13 lear, 14 clear, 15 nproc, 16 npros --help, 17 clear, 18 nproc, 19 nproc --help, 20 scp, 21 clear, 22 scp, 23 stat.

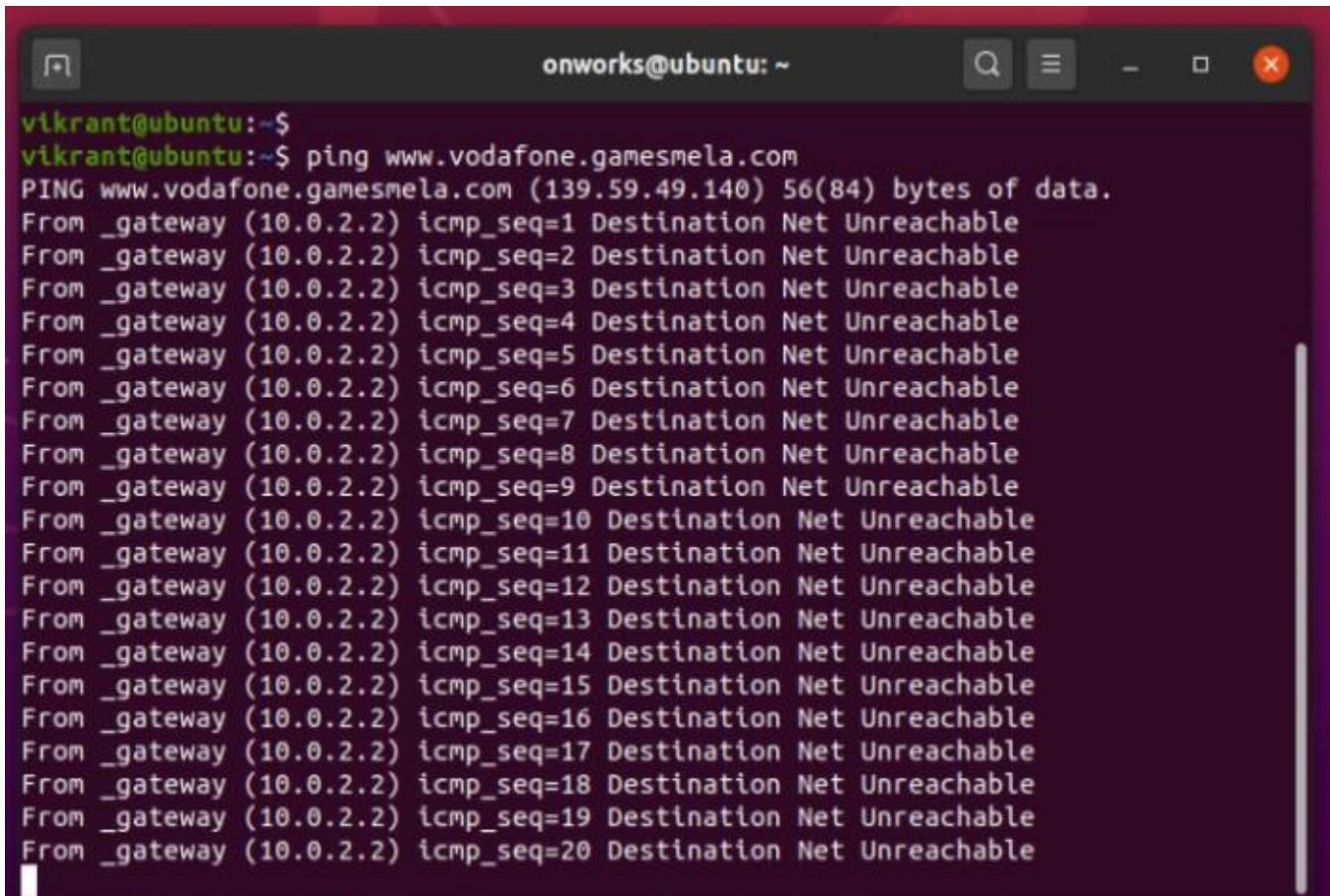
```
onworks@ubuntu: ~  
vkrant@ubuntu:~$ history  
1  sudo su -  
2  netstat -tulpn  
3  lsof  
4  netstat -tulpn  
5  lsof  
6  usermod  
7  pmap  
8  clear  
9  pmap 8080  
10 clear  
11 ssh --help  
12 telnet  
13 lear  
14 clear  
15 nproc  
16 npros --help  
17 clear  
18 nproc  
19 nproc --help  
20 scp  
21 clear  
22 scp  
23 stat
```

47.ps :- If you want to see the list of processes that are currently running for your session or for other users on the system then ps command is for you as it shows processes with their process identification numbers and in detail as well when you use ps -u command.

A terminal window titled 'onworks@ubuntu: ~' showing the output of the 'ps' command. The output is a table with four columns: PID, TTY, TIME, and CMD. The table lists three processes: PID 4185, TTY pts/0, TIME 00:00:00, CMD bash; PID 4312, TTY pts/0, TIME 00:00:00, CMD telnet; and PID 17012, TTY pts/0, TIME 00:00:00, CMD ps.

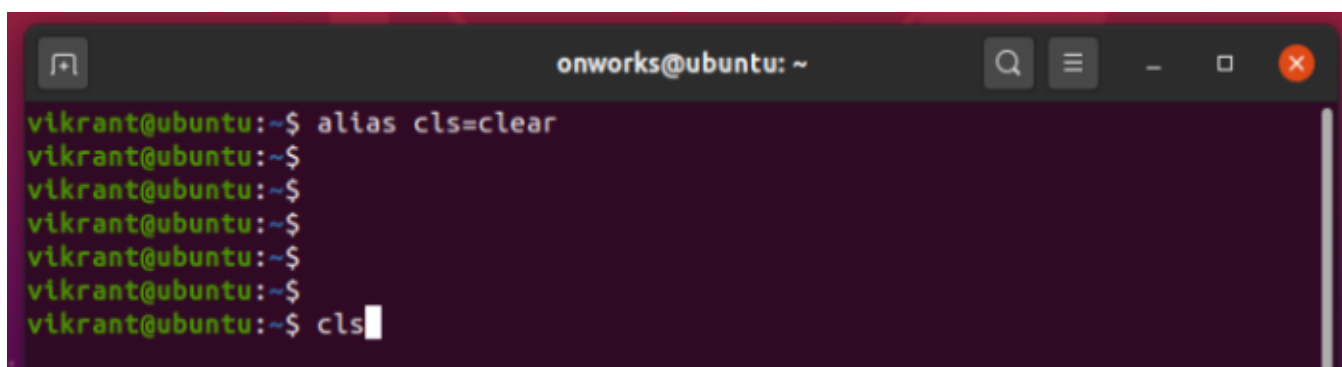
```
onworks@ubuntu: ~  
vkrant@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 4185 pts/0    00:00:00 bash  
 4312 pts/0    00:00:00 telnet  
17012 pts/0    00:00:00 ps  
vkrant@ubuntu:~$
```

48.ping :- PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message “PING” and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection.

A terminal window titled 'onworks@ubuntu: ~' showing a user named 'vikrant' attempting to ping 'www.vodafone.gamesmela.com'. The command 'ping www.vodafone.gamesmela.com' is entered, and the output shows 20 consecutive 'Destination Net Unreachable' messages from the gateway (10.0.2.2) with increasing ICMP sequence numbers from 1 to 20. The window has standard Ubuntu window controls (minimize, maximize, close) and a search icon in the title bar.

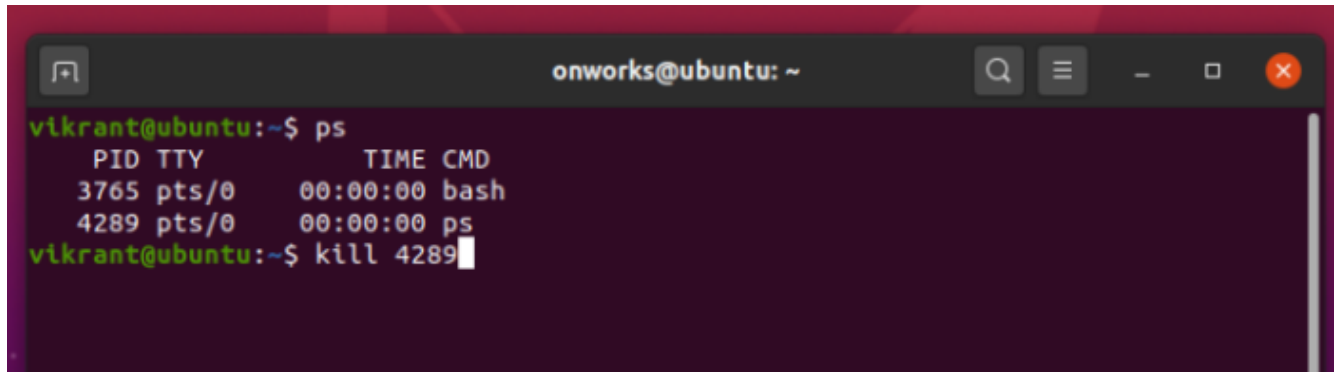
```
onworks@ubuntu: ~
vikrant@ubuntu:~$
vikrant@ubuntu:~$ ping www.vodafone.gamesmela.com
PING www.vodafone.gamesmela.com (139.59.49.140) 56(84) bytes of data.
From _gateway (10.0.2.2) icmp_seq=1 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=2 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=3 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=4 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=5 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=6 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=7 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=8 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=9 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=10 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=11 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=12 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=13 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=14 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=15 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=16 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=17 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=18 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=19 Destination Net Unreachable
From _gateway (10.0.2.2) icmp_seq=20 Destination Net Unreachable
```

49 . alias :- The command Alias is an amazing way to personalize and organize all your commands. It allows users to designate a name to a single command or even a string of commands. So programmers can give a short name before executing it. Here is an example that uses an alias:

A terminal window titled 'onworks@ubuntu: ~' showing a user named 'vikrant' creating an alias 'cls' for the 'clear' command. The user enters 'alias cls=clear' and then several empty lines. Finally, the user enters 'cls' at the prompt. The window has standard Ubuntu window controls and a search icon in the title bar.

```
onworks@ubuntu: ~
vikrant@ubuntu:~$ alias cls=clear
vikrant@ubuntu:~$
vikrant@ubuntu:~$
vikrant@ubuntu:~$
vikrant@ubuntu:~$
vikrant@ubuntu:~$
vikrant@ubuntu:~$ cls
```


50 kill :- The kill command offers the liberty to end the process from the command line. It is useful for those monitoring CPU processes as it makes it easy to terminate processes without working on it. The user is only required to enter the Process ID(PID). Users must be cautious not to kill process randomly or haphazardly. Only use the kill command if the process or task must terminate or has locked up. To obtain the process ID, users have to use the ps and grep command.

A terminal window titled 'onworks@ubuntu: ~' with search, menu, and window control icons. The user 'vikrant@ubuntu' runs 'ps', displaying a table of running processes. Then, they run 'kill 4289' to terminate the process with PID 4289.

```
vikrant@ubuntu:~$ ps
  PID TTY          TIME CMD
 3765 pts/0    00:00:00 bash
 4289 pts/0    00:00:00 ps
vikrant@ubuntu:~$ kill 4289
```

