ASSIGNMENT 1:

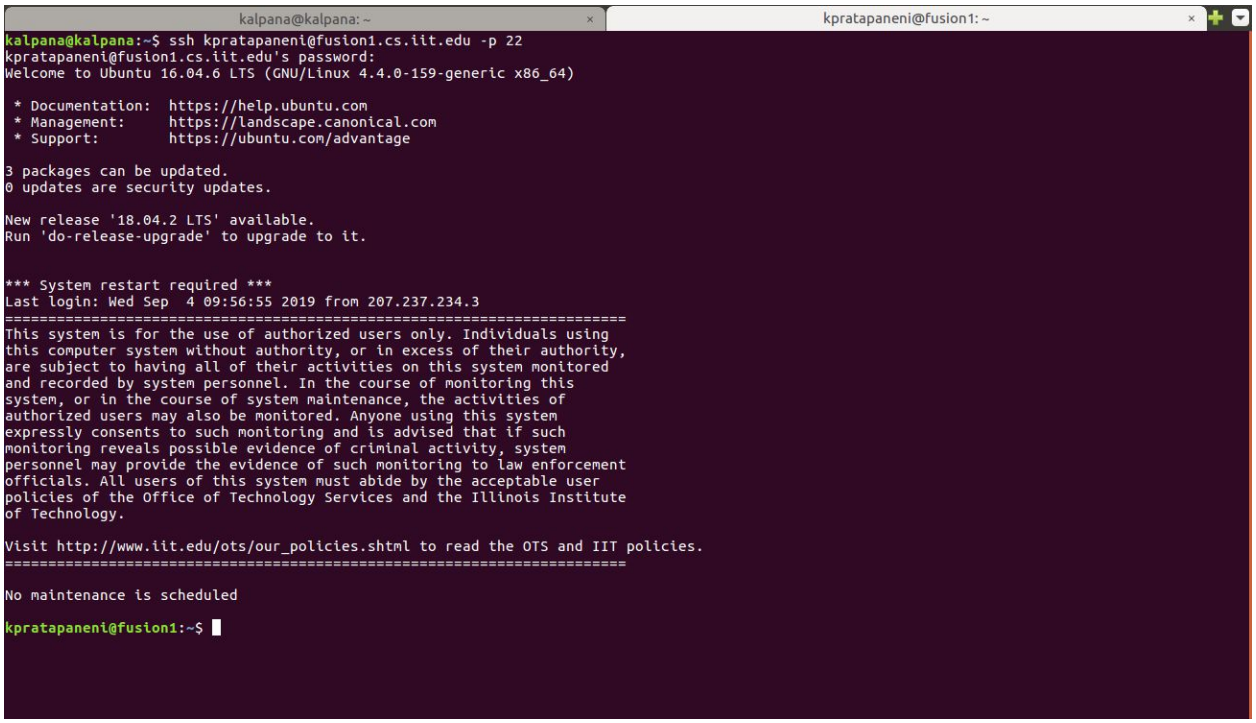
2. Show an example of using the following commands (hint: you can use man to find more information about each one); take screenshots of your commands; make sure to clear the screen between each command; explain in your own words what these commands do:

a) ssh :- It stands for secure shell. It is used for providing secure connections between the host and the remote servers.

Syntax:

$ ssh kpratapaneni@fusion1.cs.iit.edu -p 22

- Connects to server fusion1 for the user “kpratapaneni”



b) ssh-keygen :- It is used to create public and private keypairs. These are used for log-ins and for authenticating users and hosts.

Syntax: $ ssh-keygen

- It is generates public and private key pair for authenticating hosts and users in “.ssh” folder.

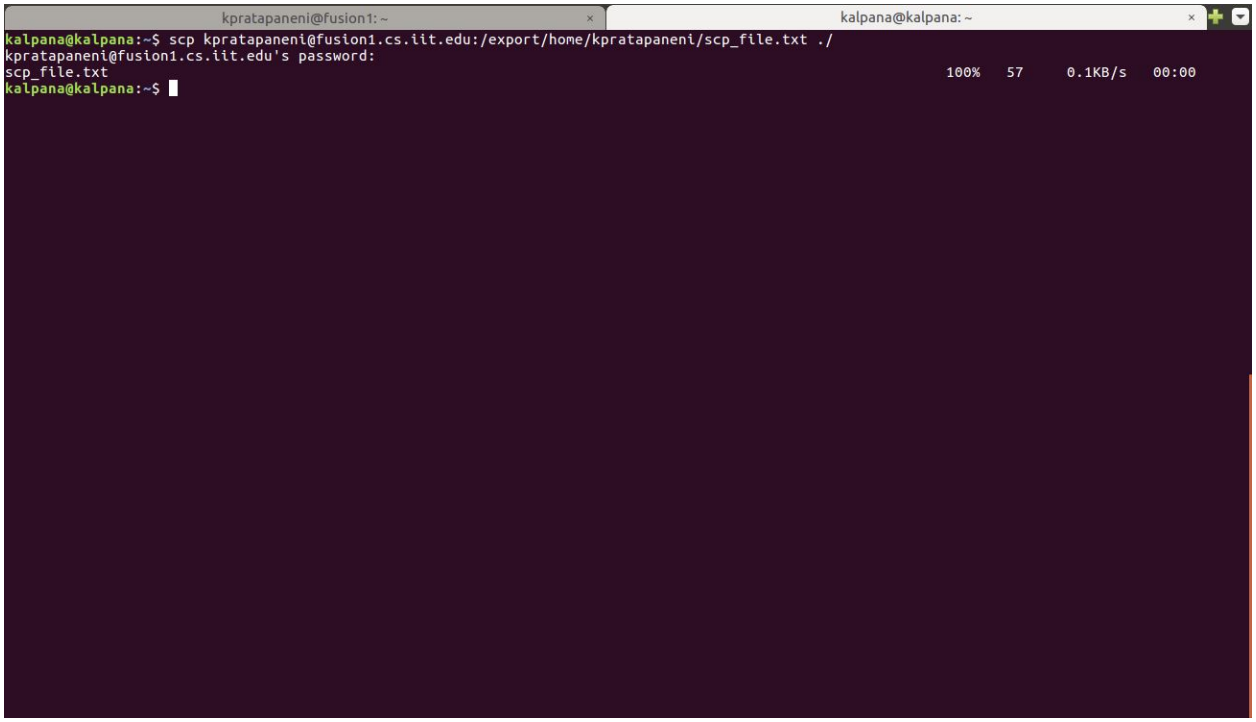
```
kalpana@kalpana: ~                                  kalpana@kalpana: ~
kalpana@kalpana:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kalpana/.ssh/id_rsa):
Created directory '/home/kalpana/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kalpana/.ssh/id_rsa.
Your public key has been saved in /home/kalpana/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:biIH0uKp/oMtXXRmywcAdE3I3h1TxaT0F1+tN6jCzLM kalpana@kalpana
The key's randomart image is:
+---[RSA 2048]----+
|  .oo.+.  .o+o .o|
|    .+ . o. o.  =|
|    . o . o. ..o.|
|   . o * .   .o..|
|  o + = S   .  ..|
| . + o + B .     |
|  * o o + +      |
| + + o o E       |
|o.o..            |
+----[SHA256]-----+
kalpana@kalpana:~$ ls ~/.s
.selected_editor           .ssh/                       .sudo_as_admin_successful
kalpana@kalpana:~$ ls ~/.ssh
id_rsa  id_rsa.pub
kalpana@kalpana:~$
```

c) scp:- It stands for secure copy. It is used for copying files between two servers or between a server and a host.

Syntax:-

$ scp kpratapaneni@fusion1.cs.iit.edu:/export/home/kpratapaneni/scp_file.txt ./

- Copies the file from server in the given folder to current directory



d) History:- It displays the list of commands that have been used in the past in a terminal. We can also re-run, re-check and search for commands.

Syntax:

i) history

- Shows list of all commands used

ii) !2

- Runs second command in the history file .bash_history

iii) history | grep vim

- searching for commands that match a text pattern

```
kalpana@kalpana:~$ history
    1  ls
    2  ls -1
    3  cd Downloads/
    4  cd
    5  history
    6  lspci | grep -i wireless
    7  vim .bash_history
    8  sudo apt install vim
    9  vim .bash_history
   10  ls -1
   11  vim .bash_history
   12  history
kalpana@kalpana:~$ !2
ls -1
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
projects
Public
PycharmProjects
rtlwifi_new
snap
Templates
Videos
VirtualBox VMs
kalpana@kalpana:~$ history | grep vim
    7  vim .bash_history
    8  sudo apt install vim
    9  vim .bash_history
   11  vim .bash_history
   14  history | grep vim
kalpana@kalpana:~$
```

e) sudo:- It stands for superuser do. We use to do administrative tasks.

Syntax :-

$ sudo -i

- login as a super user.

$ sudo reboot

- restarts the computer.

```
kalpana@kalpana:~$ sudo -i
[sudo] password for kalpana:
root@kalpana:~#
```

f) ip:- It stands for internet protocol. It is used for manipulating routing devices and tunnels.

Syntax:

$ ip addr

- It shows the system IP address.

$ sudo ip link set wlp3s0 up

- restart the interface wlp3s0 if it is down

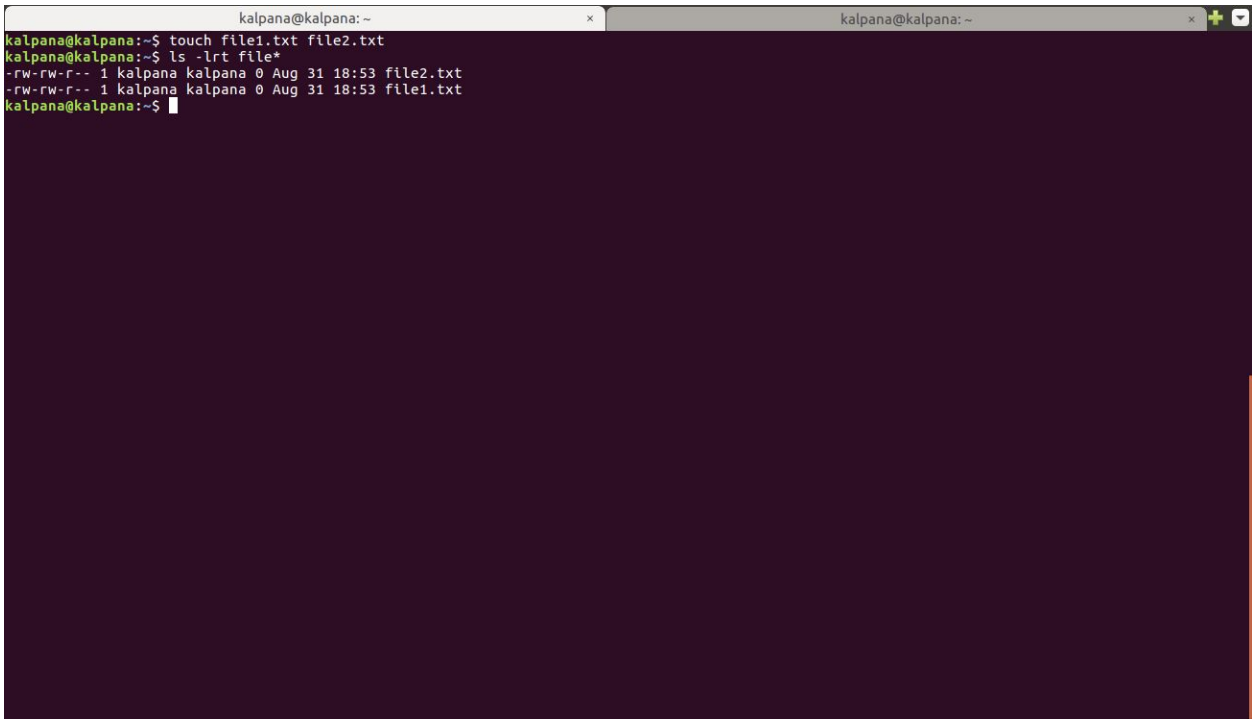
```
kalpana@kalpana: ~        kalpana@kalpana: ~
kalpana@kalpana:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 68:f7:28:6e:7b:f8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.7.40/24 brd 192.168.7.255 scope global dynamic enp2s0
       valid_lft 10079sec preferred_lft 10079sec
    inet6 fe80::3a21:944a:4059:ec0a/64 scope link
       valid_lft forever preferred_lft forever
3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state DORMANT group default qlen 1000
    link/ether 38:b1:db:d6:af:fd brd ff:ff:ff:ff:ff:ff
kalpana@kalpana:~$ sudo ip link set wlp3s0 up
[sudo] password for kalpana:
kalpana@kalpana:~$
```

g) touch:- It is used to create new files without any content and recreates file timestamps like modifying access time.

Syntax:-

$ touch file.txt

- It creates a new file.



h) ls:- It shows all the files and directories along with details like modified date, file size, owner of the file and its permissions.

Syntax:-

$ ls

- shows all files

$ ls -a

- It shows all the files along with hidden files.

$ ls -lrt

- Here, ” i” stands for owner and its permissions, size and modified date; “r” stands for reverse order and “t” stands for latest modified file.

```
kalpana@kalpana: ~/projects/cs585/assignment1          kalpana@kalpana: ~
kalpana@kalpana:~$ ls
Desktop    Downloads         file1.txt  Music     projects  PycharmProjects  snap       Videos
Documents  examples.desktop  file2.txt  Pictures  Public    rtlwifi_new      Templates  VirtualBox VMs
kalpana@kalpana:~$ ls -a
.                  .cache     Downloads         .gksu.lock     Music     .PyCharm2019.2             Videos
..                 .compiz    examples.desktop  .gnupg         Pictures  PycharmProjects            .viminfo
.bash_history      .config    file1.txt         .ICEauthority  .pki      rtlwifi_new                VirtualBox VMs
.bash_history_old  Desktop    file2.txt         .java          .profile  snap                       .Xauthority
.bash_logout       .dmrc      .gconf            .local         projects  .sudo_as_admin_successful  .xsession-errors
.bashrc            Documents  .gitconfig        .mozilla       Public    Templates                  .xsession-errors.old
kalpana@kalpana:~$ ls -lrt
total 64
-rw-r--r--  1 kalpana kalpana 8980 Aug  5 23:32 examples.desktop
drwxr-xr-x  2 kalpana kalpana 4096 Aug  5 23:46 Videos
drwxr-xr-x  2 kalpana kalpana 4096 Aug  5 23:46 Templates
drwxr-xr-x  2 kalpana kalpana 4096 Aug  5 23:46 Public
drwxr-xr-x  2 kalpana kalpana 4096 Aug  5 23:46 Music
drwxrwxr-x 22 kalpana kalpana 4096 Aug 11 19:17 rtlwifi_new
drwxr-xr-x  2 kalpana kalpana 4096 Aug 30 11:15 Documents
drwxr-xr-x  4 kalpana kalpana 4096 Aug 30 12:34 snap
drwxrwxr-x  3 kalpana kalpana 4096 Aug 30 12:36 PycharmProjects
drwxrwxr-x  3 kalpana kalpana 4096 Aug 31 12:11 VirtualBox VMs
drwxr-xr-x  2 kalpana kalpana 4096 Aug 31 14:04 Desktop
-rw-rw-r--  1 kalpana kalpana    0 Aug 31 18:53 file2.txt
-rw-rw-r--  1 kalpana kalpana    0 Aug 31 18:53 file1.txt
drwxr-xr-x  3 kalpana kalpana 4096 Aug 31 20:06 Pictures
drwxrwxr-x  5 kalpana kalpana 4096 Sep  1 11:44 projects
drwxr-xr-x  5 kalpana kalpana 4096 Sep  1 15:31 Downloads
kalpana@kalpana:~$
```

I) mkdir:- It is used to create new directories.

Syntax:-

$ mkdir new_folder1

- creates new folder with name "new_folder1"

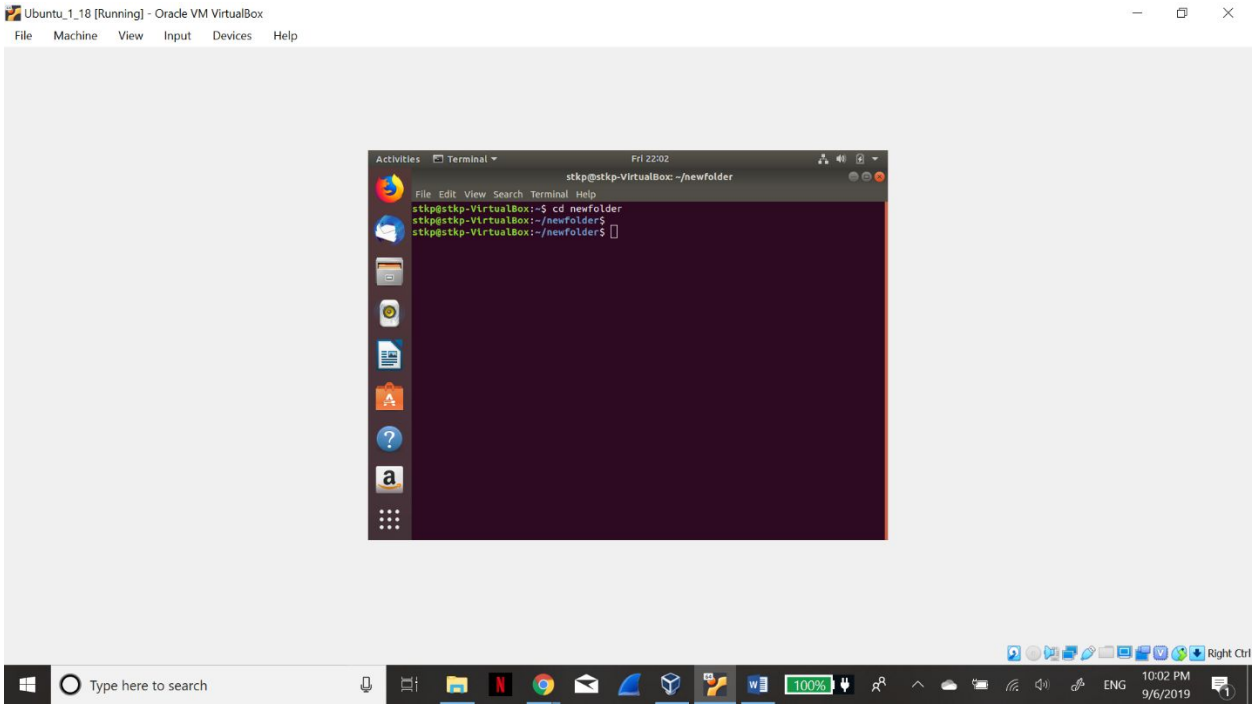
$ mkdir -p new_folder2

- creates new directory if it does not already exist

```
kalpana@kalpana: ~
kalpana@kalpana: ~
kalpana@kalpana:~$ mkdir new_folder1
kalpana@kalpana:~$ mkdir -p new_folder2
kalpana@kalpana:~$ mkdir -p new_folder2
kalpana@kalpana:~$ ls new_folder*
new_folder1:

new_folder2:
kalpana@kalpana:~$
```

j) cd:- It is used to change the directory through the terminal.
Syntax:-
$ cd new_folder

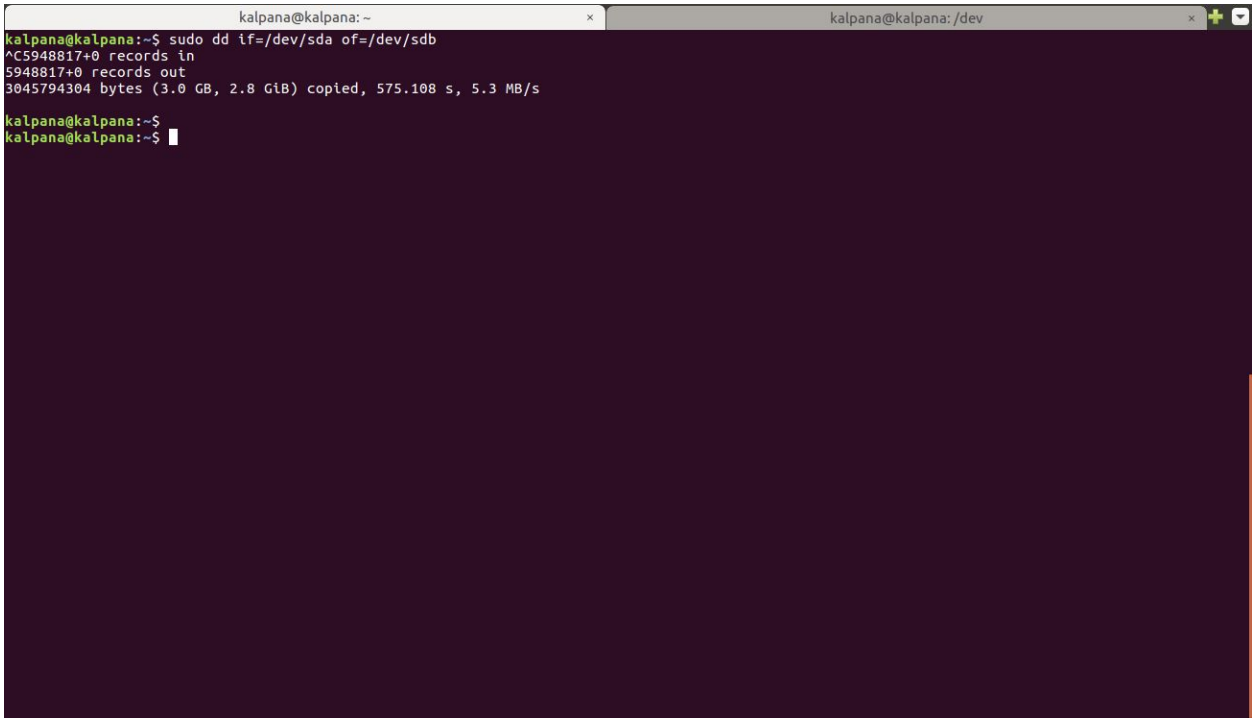
- Move to folder new_folder from the current path



k) dd :- It stands for data duplicator. It is used for copying and converting data.
Syntax:-

$ sudo dd if=/dev/sda of=/dev/sdb

- It backups the entire hard disk copy to another hard disk

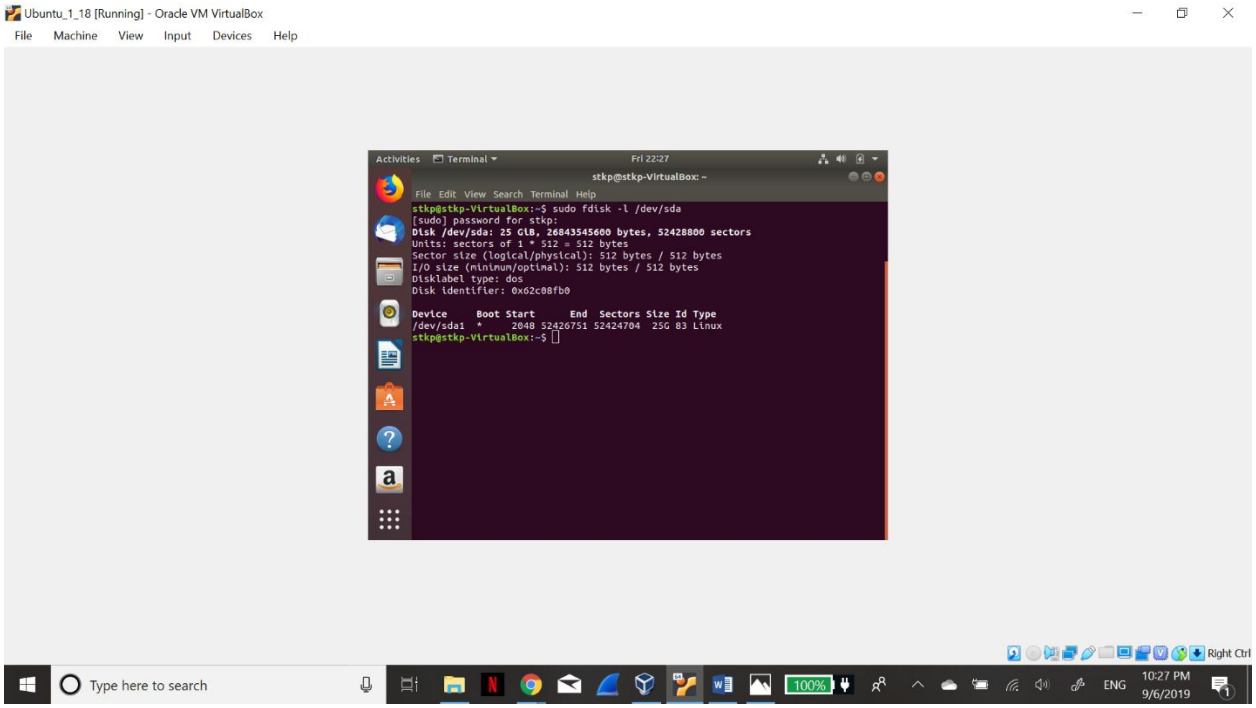


l) fdisk:- It stands for format disk. It is used for creating and manipulating disk partition table.

Syntax:-

$ sudo fdisk -l /dev/sda

- shows all partitions on device dev/sda.



m) apt:- It stands for advanced packaging tool. It is used for installing software packages, upgrading of existing software packages, updating of the package list index and even upgrading the entire Ubuntu system.

Syntax:

$ sudo apt remove vim

- It uninstalls vim.

$ sudo apt install vim

- It installs vim.

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~

kalpana@kalpana:~$ sudo apt remove vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-45 linux-headers-4.15.0-45-generic linux-headers-4.4.0-157 linux-headers-4.4.0-157-generic
  linux-image-4.15.0-45-generic linux-image-4.4.0-157-generic linux-modules-4.15.0-45-generic linux-modules-4.4.0-157-generic
  linux-modules-extra-4.15.0-45-generic linux-modules-extra-4.4.0-157-generic snapd-login-service vim-runtime
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  vim
0 upgraded, 0 newly installed, 1 to remove and 54 not upgraded.
After this operation, 2,458 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 319549 files and directories currently installed.)
Removing vim (2:7.4.1689-3ubuntu1.3) ...
update-alternatives: using /usr/bin/vim.tiny to provide /usr/bin/vi (vi) in auto mode
update-alternatives: using /usr/bin/vim.tiny to provide /usr/bin/view (view) in auto mode
update-alternatives: using /usr/bin/vim.tiny to provide /usr/bin/ex (ex) in auto mode
update-alternatives: using /usr/bin/vim.tiny to provide /usr/bin/rview (rview) in auto mode
kalpana@kalpana:~$ sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-45 linux-headers-4.15.0-45-generic linux-headers-4.4.0-157 linux-headers-4.4.0-157-generic
  linux-image-4.15.0-45-generic linux-image-4.4.0-157-generic linux-modules-4.15.0-45-generic linux-modules-4.4.0-157-generic
  linux-modules-extra-4.15.0-45-generic linux-modules-extra-4.4.0-157-generic snapd-login-service
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim
0 upgraded, 1 newly installed, 0 to remove and 54 not upgraded.
Need to get 1,036 kB of archives.
After this operation, 2,458 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 vim amd64 2:7.4.1689-3ubuntu1.3 [1,036 kB]
Fetched 1,036 kB in 0s (1,216 kB/s)
Selecting previously unselected package vim.
(Reading database ... 319544 files and directories currently installed.)
```

n) vi:- It is an editor which acts as a text editor.

Syntax:

$ vi file1.txt

- It opens a new text file for “file1.txt” if a file doesn’t exist. If a file already exists, it opens the existing file.

```
kalpana@kalpana:~$ vi file1.txt
kalpana@kalpana:~$ vi file1.txt
kalpana@kalpana:~$
```

o) time:- It is used to determine the duration of the execution of a command.

Syntax:-

$ time sleep 3

- creates a dummy job which lasts 3 seconds

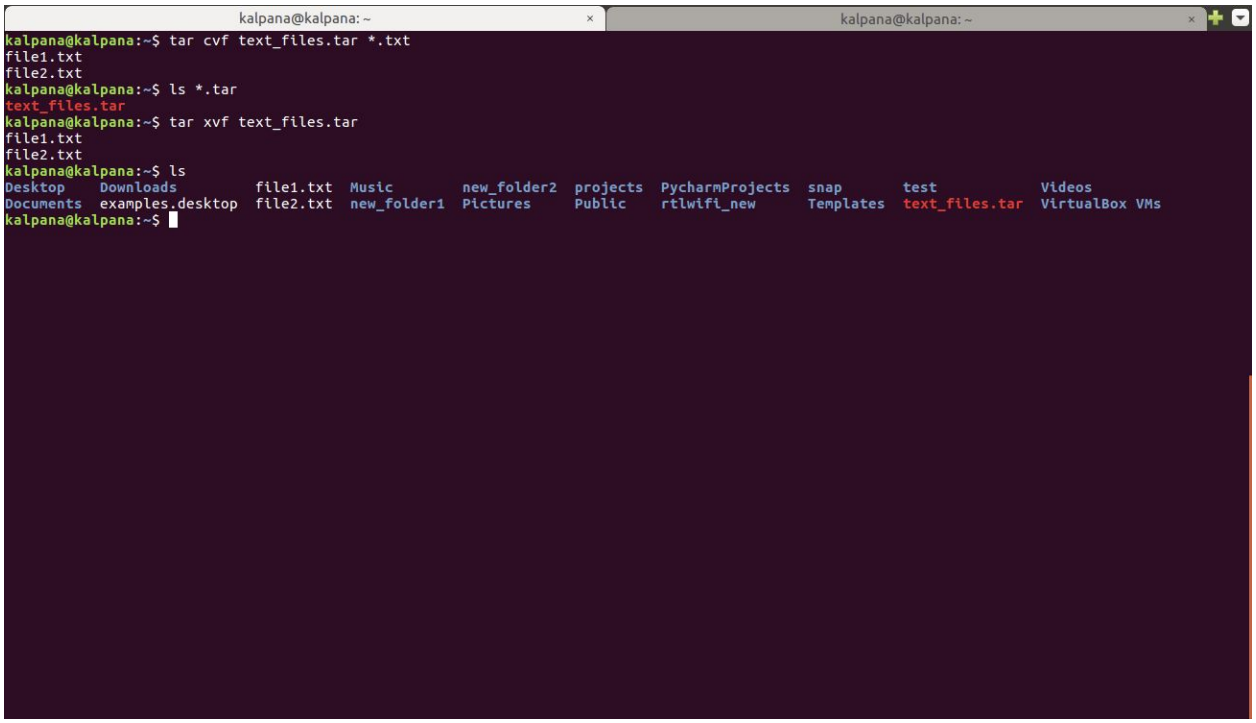
```
kalpana@kalpana:~$ time sleep 3

real    0m3.002s
user    0m0.002s
sys     0m0.000s
kalpana@kalpana:~$
```

p) tar:- Stands for tape archive. It is used to create archive files and extract the archive files.
Syntax:-
$ tar cvf text_files.tar *.txt

- creates archive of all text files

$ tar xvf text_files.tar

- extracts the archive folder text_files.tar to current directory

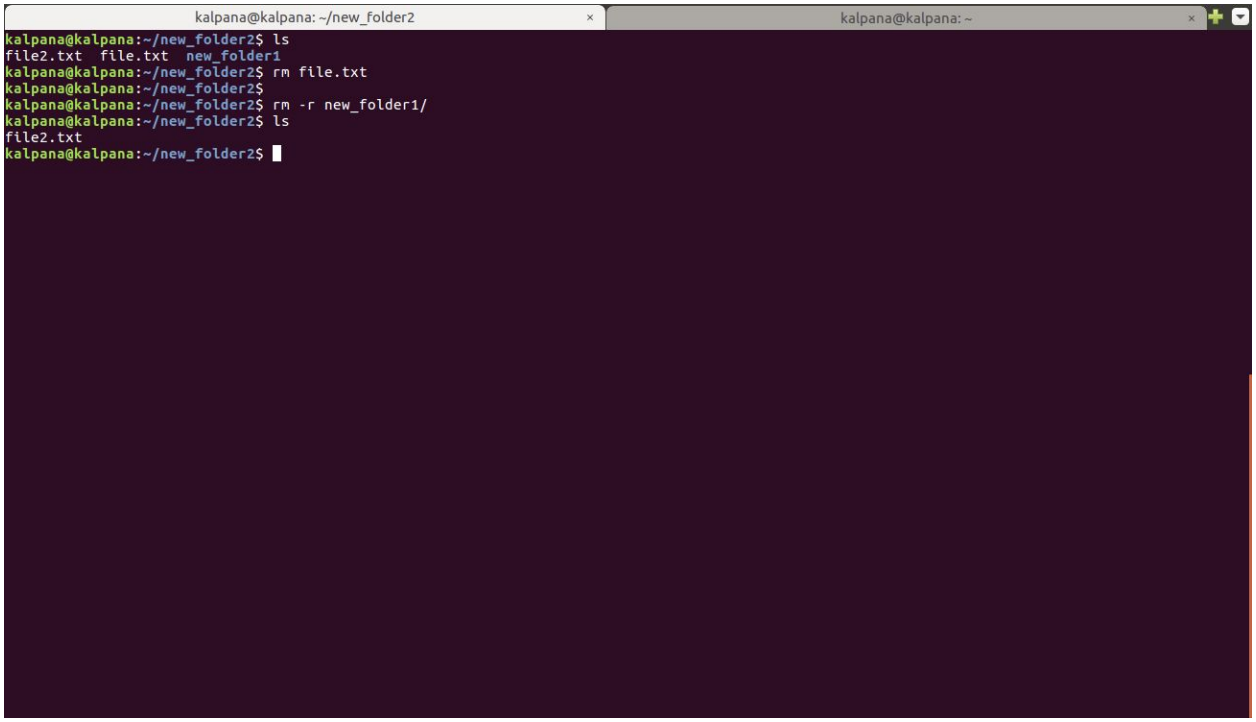


q) rm:- It stands for remove. It is used to remove files, directories and symbolic links.
Syntax:
$ rm file.txt

- removes the file "file.txt"

$ rm -r new_folder1

- removes the folder "new_folder1"



r) cat:- It stands for concatenate. It is used to create files, view file content, redirect output into terminal or file.

Syntax:-

$ cat file1.txt file2.txt

- shows the contents of files file1.txt and file2.txt

$cat file1.txt > file1_cpy.txt

- copies the text of file1 to file1_cpy.

$ cat > newfile.txt

- creates a new file “newfile.txt”

```
kalpana@kalpana: ~/new_folder2        kalpana@kalpana: ~/new_folder2

kalpana@kalpana:~/new_folder2$ cat file1.txt
This is files.txt.
Here I am with some text
kalpana@kalpana:~/new_folder2$ cat file1.txt file2.txt
This is files.txt.
Here I am with some text
Hello this is file2.txt
here is my second line
kalpana@kalpana:~/new_folder2$ cat file1.txt > file1_cpy.txt
kalpana@kalpana:~/new_folder2$ cat file1_cpy.txt
This is files.txt.
Here I am with some text
kalpana@kalpana:~/new_folder2$ cat > newfile.txt
my new file with text
second line of text
end of text
kalpana@kalpana:~/new_folder2$ cat newfile.txt
my new file with text
second line of text
end of text
kalpana@kalpana:~/new_folder2$
```

s) Bash:- It stands for bourne again shell. It is sh-compatible command language interpreter that executes commands read from the standard input or from a file.

Syntax:-

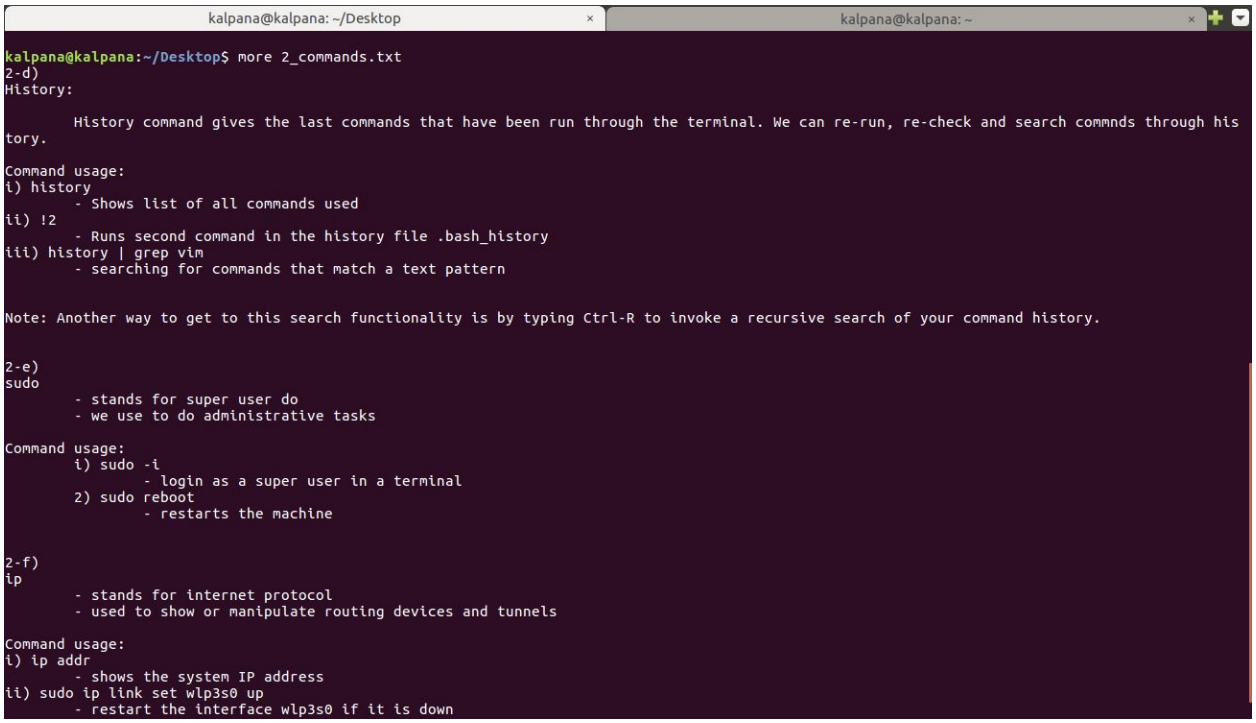
$ ./hello_world.sh

- Runs the above shell file, echoes the output into terminal.

```
kalpana@kalpana:~$ cat hello_world.sh
#!/bin/bash

echo "Hello world!"
kalpana@kalpana:~$ ./hello_world.sh
Hello world!
kalpana@kalpana:~$
```

t) more:- The command helps you to navigate outputs from commands in a user-friendly way. It is also a filter for paging through text one screenful at a time.
Syntax:-
$ more 2_commands.txt

- It displays the content of given text file and we can use keys like enter, space bar, / and q to show up one line, scroll a full screen in one go, search stuff suffixed with keywords and quit respectively.



u) watch:- It runs a command repeatedly and displays its outputs and errors. It watches the program output change over time. By default, the command runs every 2 seconds and watch will run until interrupted.
Syntax:-
$ sleep 5; echo “ hello world”>> ~/newfile.txt

- Run this command on one terminal, which runs after 5 seconds.

$ watch -n 2.5 ls ~

- It watches the ls command change every 2.5 seconds and the newfile.txt appears.

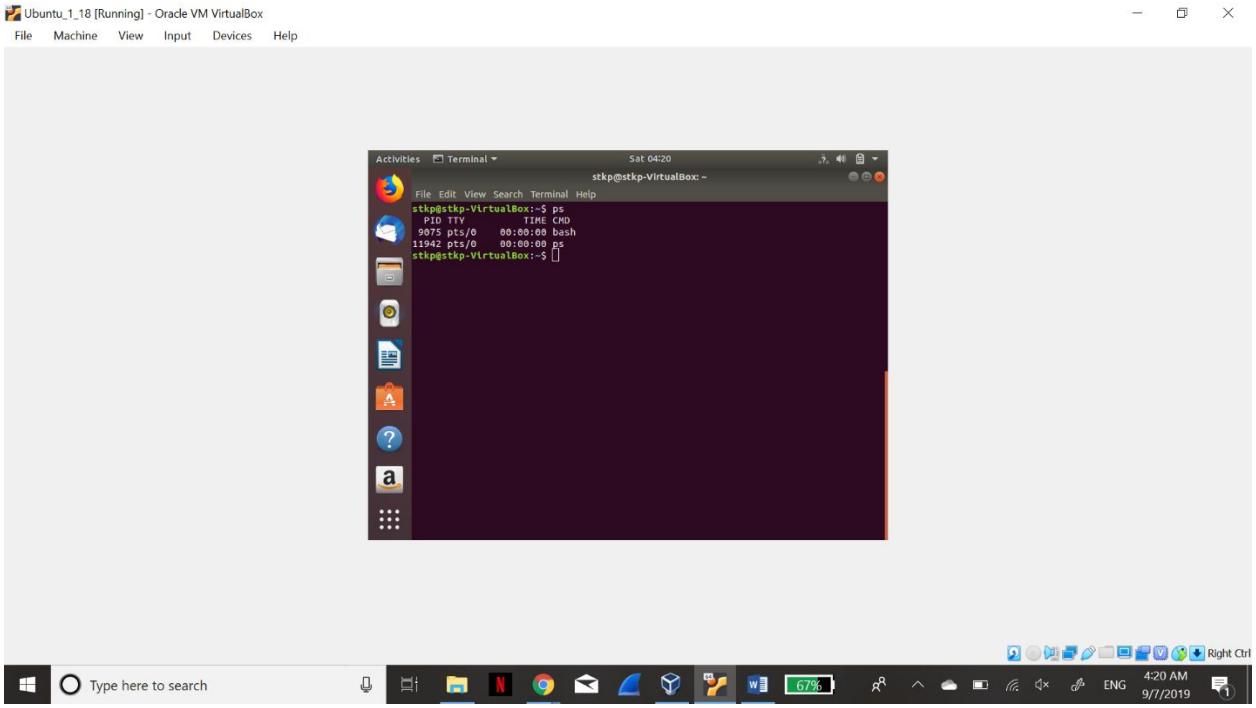
```
kalpana@kalpana:~$ ls *.txt
file2.txt
kalpana@kalpana:~$ sleep 5; echo "hello world" >> ~/newfile.txt
kalpana@kalpana:~$ ls *.txt
file2.txt  newfile.txt
kalpana@kalpana:~$
```

```
Every 2.5s: ls /home/kalpana                                    Tue Sep  3 10:58:24 2019

Desktop
Documents
Downloads
examples.desktop
file2.txt
hello_world.sh
Music
newfile.txt
new_folder2
Pictures
projects
Public
PycharmProjects
rtlwifi_new
snap
Templates
text_files.tar
Videos
VirtualBox VMs
```

v) ps:- It stands for process status. It shows the list of currently running processes along with its respective unique Id, terminal type, amount of CPU in mins, name of the command that launched the process.

Syntax:- ps

- lists out the processes.



w) top :- It displays system information and list of processes/threads that are currently being managed by the Linux Kernel.

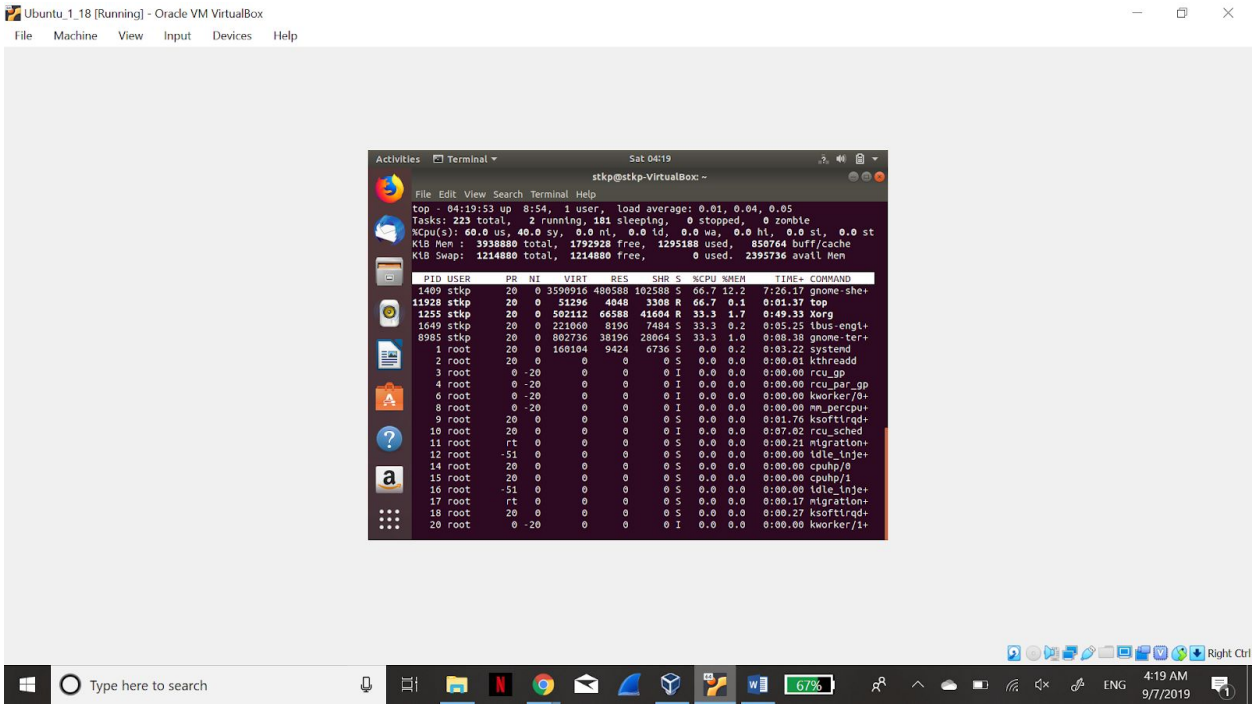
Syntax:

$top

- displays list of all processes

$ top -u root

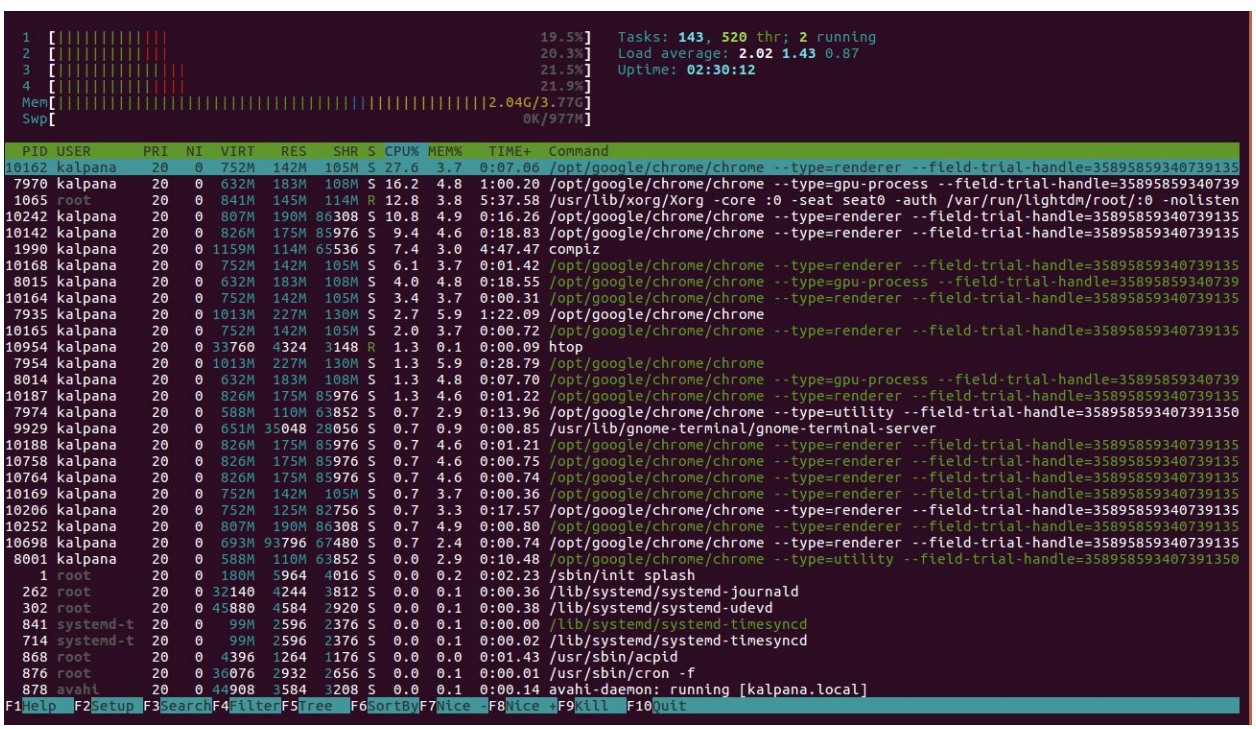
- displays all root processes



x) htop :- It is a system monitoring utility that runs on the terminal. It is a bit similar to top. The usage of htop is easier compared to top. In my ubuntu, I've installed it specifically using apt-get command.

Syntax: htop

- Lists out all processes in readable format. Use Up/Down arrow keys to select a process, and F9 key to kill it, or use F7 and F8 keys to change the priority



y) gcc:- It stands for GNU Compiler Collection (GCC). It is a collection of compilers and libraries for C, C++, Objective-C, Fortran, Ada, Go, and D programming languages. Many open source projects including the GNU tools and the Linux kernel are compiled with GCC.

Syntax:

$ gcc hello.c -o hello

- Compiles hello.c and create a binary file "hello"

z) tail:- It is used to monitor a file, and shows the last few lines.

Syntax:

$ tail 2_commands.txt

- shows last ten lines

$ tail -20 2_commands.txt

- shows last 20 lines

$ tail -f test.log

- renders the latest log

```
kalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$ tail 2_commands.txt
        $ clear
                - clears the screen
jjj. exit
        - Command used to exit the shell where it is currently running.
        Command Usage:
        $ exit
                - exits the terminal
        $ sudo su
        $ exit
                - exit the root directorykalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$ tail -20 2_commands.txt
aaa. less
bbb. su
ccc. ping
ddd. traceroute
eee. date
fff. time
ggg. wget
hhh. wc
iii. clear
        - used to clear the terminal screen
        $ clear
                - clears the screen
jjj. exit
        - Command used to exit the shell where it is currently running.
        Command Usage:
        $ exit
                - exits the terminal
        $ sudo su
        $ exit
                - exit the root directorykalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$
```

aa) grep:- It is a filter and it searches a file for a particular pattern/regular expression and displays all lines that contain that pattern.

Syntax:

$ grep "ls" 2_commands.txt

- Displays the lines that has a match

$ grep -c "ls" 2_commands.txt

- Displays no of lines that has a match

```
kalpana@kalpana:~/Desktop$ grep "ls" 2_commands.txt
        - used to show or manipulate routing devices and tunnels
ls
        - list out the files and folders. It can also show other file details like size, modified date, owner of the file and its permissions.
i)ls
ii)ls -a
iii) ls -lrt
                - uninstalls the vim package
                -Installs vim package
        $ watch -n 2.5 ls ~
        - Watches the ls command change every 2.5 seconds and we can see newfile.txt.
        - The GNU Compiler Collection (GCC) is a collection of compilers and libraries for C, C++, Objective-C, Fortran, Ada, Go, and D program
ming languages. Many open source projects including the GNU tools and the Linux kernel are compiled with GCC.
kalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$
kalpana@kalpana:~/Desktop$ grep -c "ls" 2_commands.txt
12
kalpana@kalpana:~/Desktop$
```

bb) kill:- It is used to terminate processes manually.

Syntax: kill 12775-

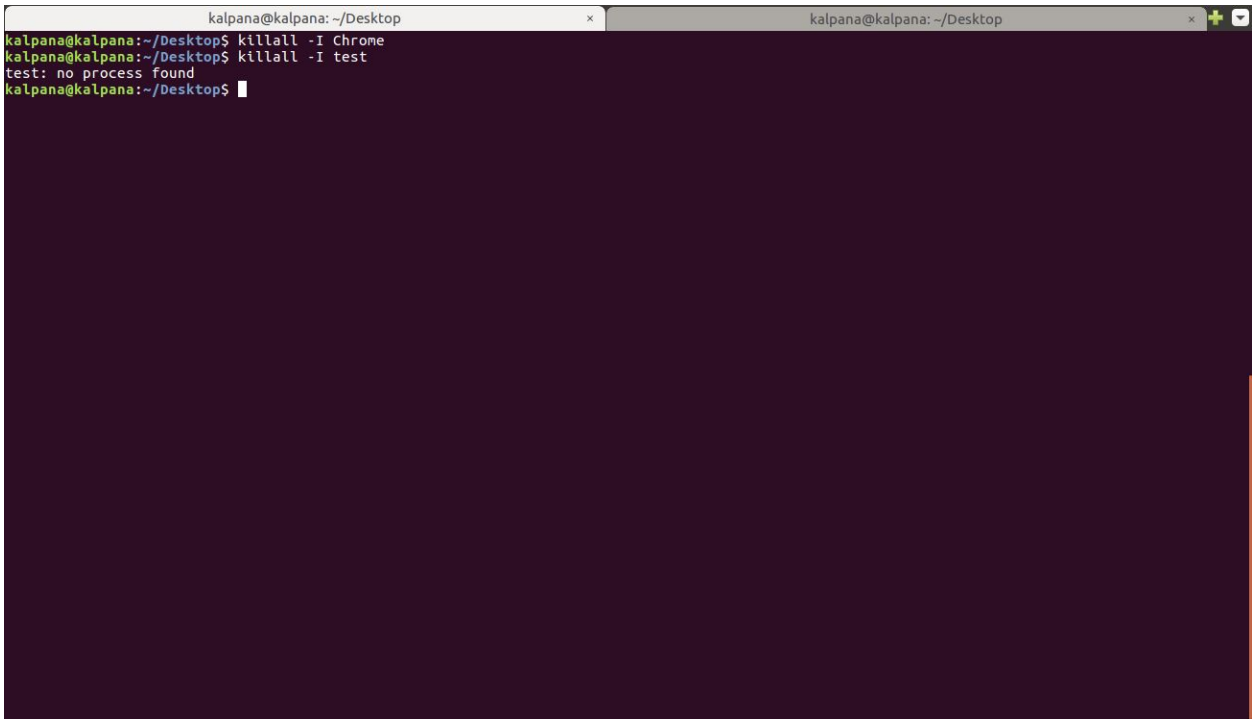
- It kills the process with PID 12775 if it is currently running otherwise shows an error "No such process"

```
kalpana@kalpana: ~/Desktop                    kalpana@kalpana: ~/Desktop
kalpana@kalpana:~/Desktop$ ps
  PID TTY          TIME CMD
 9936 pts/4    00:00:00 bash
12775 pts/4    00:00:00 ps
kalpana@kalpana:~/Desktop$ kill 12775
bash: kill: (12775) - No such process
kalpana@kalpana:~/Desktop$
```

cc) killall:- It is a command that forcibly terminates processes, specified by a name. It sends a signal to all processes running any of the specified commands. If no signal name is specified, SIGTERM is sent.

Syntax:
$ killall -I Chrome

- Kills all chrome tabs if opens otherwise throes error "no process found"



dd) du:- It stands for disk usage. It is used to track files and folders which are consuming space.
Syntax:
$ du -a ./

- shows all files along with space used in bytes in the current folder

$ du -a -h ./

- Shows the bytes in human readable formats like GB, MB, and KB.

```
kalpana@kalpana: ~/new_folder2          kalpana@kalpana: ~/Desktop
kalpana@kalpana:~/new_folder2$ du -a ./
4       ./newfile.txt
4       ./move_me.txt
4       ./file2.txt
4       ./file1.txt
4       ./test
4       ./file1_cpy.txt
28      ./
kalpana@kalpana:~/new_folder2$ du -h -a ./
4.0K    ./newfile.txt
4.0K    ./move_me.txt
4.0K    ./file2.txt
4.0K    ./file1.txt
4.0K    ./test
4.0K    ./file1_cpy.txt
28K     ./
kalpana@kalpana:~/new_folder2$
```

ee) df:- It stands for disk free. It displays the space available on all currently mounted file systems.
Syntax: du- displays the space available on all currently mounted file systems.

$ df

- display the size in power of 1024.

$ df -h

- display the size in the human readable format.

```
kalpana@kalpana: ~          kalpana@kalpana: ~/Desktop
kalpana@kalpana:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1946052        0   1946052   0% /dev
tmpfs             395260     6288    388972   2% /run
/dev/sda2      959801768 36431408 874592172   4% /
tmpfs            1976284    74312   1901972   4% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            1976284        0   1976284   0% /sys/fs/cgroup
/dev/loop1         69248    69248         0 100% /snap/sublime-text/69
/dev/loop0        430848   430848         0 100% /snap/pycharm-professional/151
/dev/loop2         90880    90880         0 100% /snap/core/7396
/dev/sda1         523248     6240    517008   2% /boot/efi
tmpfs             395260       72    395188   1% /run/user/1000
kalpana@kalpana:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           386M  6.2M  380M   2% /run
/dev/sda2       916G   35G  835G   4% /
tmpfs           1.9G   73M  1.9G   4% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/loop1       68M   68M     0 100% /snap/sublime-text/69
/dev/loop0      421M  421M     0 100% /snap/pycharm-professional/151
/dev/loop2       89M   89M     0 100% /snap/core/7396
/dev/sda1       511M  6.1M  505M   2% /boot/efi
tmpfs           386M   72K  386M   1% /run/user/1000
kalpana@kalpana:~$
```

ff) screen:- It is used to launch and use multiple shell sessions from a single ssh session. When a process is started with "screen", the process can be detached from session & then can reattach the session at a later time

Syntax: screen

- It will start a new window within the screen

$ screen -ls

- display the currently opened screens including those running in the background.

```
kalpana@kalpana:~$ screen -ls
There are screens on:
        17257.pts-4.kalpana     (Tuesday 03 September 2019 12:40:02  CDT)       (Attached)
        17180.pts-4.kalpana     (Tuesday 03 September 2019 12:39:29  CDT)       (Detached)
2 Sockets in /var/run/screen/S-kalpana.
kalpana@kalpana:~$
```

gg) vim :- It is a text editor used in command line interface. All available modes are save, quit, copy, paste. w for saving, q for quit, q! for quitting without saving, i for edit, y for copy, yy for copy a line, p for paste, d for cut, dd for cut a line.

Syntax:

$ vim file2.txt

- opens the new or existing file file2.txt in vim editor

```
file 1 txt
~
:wq
```

hh) chmod:- It stands for change mode. It is used to change the access mode of a file.
Syntax: chmod 777 file2.txt

- Above command gives read, write and execution access to all users.

```
kalpana@kalpana:~$ ls -lrt file2.txt
-rw-rw-r-- 1 kalpana kalpana 11 Sep  3 12:49 file2.txt
kalpana@kalpana:~$ chmod 777 file2.txt
kalpana@kalpana:~$ ls -lrt file2.txt
-rwxrwxrwx 1 kalpana kalpana 11 Sep  3 12:49 file2.txt
kalpana@kalpana:~$ :
```

II) chown:- It is used to change the owner and group-related information for a file or directory.
Syntax:
$ sudo chown root file2.txt

- It changes the owner from kalpana to root.

```
kalpana@kalpana:~$ ls -lrt *.txt
-rw-rw-r-- 1 kalpana kalpana 12 Sep  3 10:58 newfile.txt
-rwxrwxrwx 1 kalpana kalpana 11 Sep  3 12:49 file2.txt
kalpana@kalpana:~$ sudo chown root file2.txt
kalpana@kalpana:~$ ls -lrt *.txt
-rw-rw-r-- 1 kalpana kalpana 12 Sep  3 10:58 newfile.txt
-rwxrwxrwx 1 root    kalpana 11 Sep  3 12:49 file2.txt
kalpana@kalpana:~$ sudo chown :root file2.txt
kalpana@kalpana:~$ ls -lrt *.txt
-rw-rw-r-- 1 kalpana kalpana 12 Sep  3 10:58 newfile.txt
-rwxrwxrwx 1 root    root    11 Sep  3 12:49 file2.txt
kalpana@kalpana:~$
```

jj) useradd:- It is used to create user accounts on Linux with some specific properties, limitations or comments

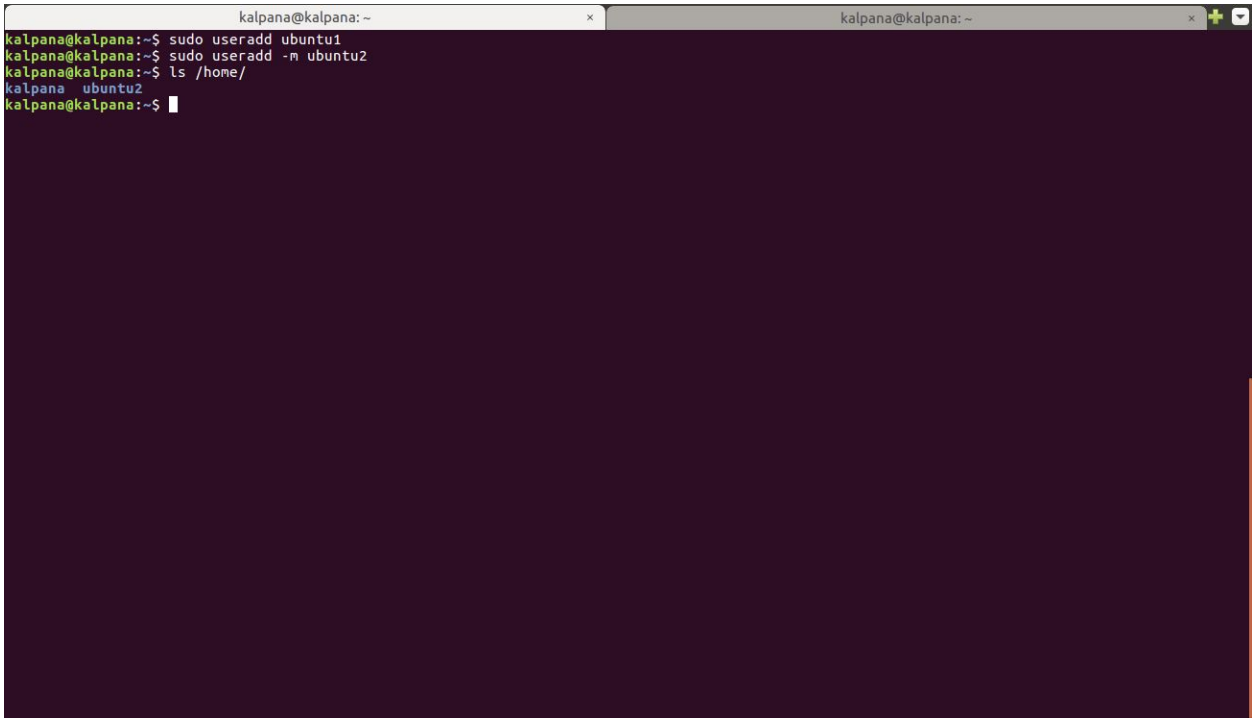
Syntax:

$ useradd ubuntu1

- Creates a new user with name ubuntu1 if the same user name does not exist.

$ useradd -m ubuntu2

- Creates the new user's home directory and copies files from /etc/skel directory to it.



kk) mv:- It is used to move and rename the files and folders

Syntax:

$ mv move_me.txt new_folder2/

- Moves "move_me.txt" to the folder "new_folder2"

$ mv file1.txt file2.txt

- Renames the file "file1.txt" to "file2.txt"

$ mv test new_folder2/

- Moves "test" folder to another folder "new_folder2"

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~
kalpana@kalpana:~$ mv move_me.txt new_folder2/
kalpana@kalpana:~$ ls new_folder2/move_me.txt
new_folder2/move_me.txt
kalpana@kalpana:~$
kalpana@kalpana:~$
kalpana@kalpana:~$ mv file1.txt file2.txt
kalpana@kalpana:~$ ls file*
file2.txt
kalpana@kalpana:~$ mv test new_folder2/
kalpana@kalpana:~$
kalpana@kalpana:~$ ls new_folder2/
file1_cpy.txt  file1.txt  file2.txt  move_me.txt  newfile.txt  test
kalpana@kalpana:~$
kalpana@kalpana:~$
```

II) man :- This command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

Syntax:

$ man ls

- Displays the manual for the command ls

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~
LS(1)                                User Commands                                LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort is
       specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576 bytes;  see  SIZE  format
              below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with  -lt:  sort  by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by
              name; otherwise: sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
              colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below
 Manual page ls(1) line 1 (press h for help or q to quit)
```

mm) locate:- It is used to find the files by name

Syntax:

$ locate file1.txt

- Searches for file file1.txt and displays the path.

$ locate *.txt

- Searches for all text files and displays its paths.

```
kalpana@kalpana: ~                                    kalpana@kalpana: ~

kalpana@kalpana:~$ locate file1.txt
/home/kalpana/file1.txt
kalpana@kalpana:~$
kalpana@kalpana:~$ locate "*.txt"
/etc/X11/rgb.txt
/etc/brltty/Input/ba/all.txt
/etc/brltty/Input/bd/all.txt
/etc/brltty/Input/bl/18.txt
/etc/brltty/Input/bl/40_m20_m40.txt
/etc/brltty/Input/ec/all.txt
/etc/brltty/Input/ec/spanish.txt
/etc/brltty/Input/eu/all.txt
/etc/brltty/Input/lb/all.txt
/etc/brltty/Input/lt/all.txt
/etc/brltty/Input/mb/all.txt
/etc/brltty/Input/mn/all.txt
/etc/brltty/Input/tn/all.txt
/etc/brltty/Input/tt/all.txt
/etc/brltty/Input/vd/all.txt
/etc/brltty/Input/vr/all.txt
/etc/brltty/Input/vs/all.txt
/home/kalpana/file1.txt
/home/kalpana/file2.txt
/home/kalpana/.PyCharm2019.2/system/log/threadDumps-freeze-20190830-123516-PY-192.6262.63-WelcomeFrame.showNow-16sec/threadDump-20190830-123522
.txt
/home/kalpana/.PyCharm2019.2/system/log/threadDumps-freeze-20190830-123516-PY-192.6262.63-WelcomeFrame.showNow-16sec/threadDump-20190830-123528
.txt
/home/kalpana/.PyCharm2019.2/system/log/threadDumps-freeze-20190830-123643-PY-192.6262.63-RandomAccessFile.seek0-7sec/threadDump-20190830-12364
9.txt
/home/kalpana/.PyCharm2019.2/system/log/threadDumps-freeze-20190830-123703-PY-192.6262.63-AccessController.doPrivileged-9sec/threadDump-2019083
0-123705.txt
/home/kalpana/.PyCharm2019.2/system/log/threadDumps-freeze-20190830-123703-PY-192.6262.63-AccessController.doPrivileged-9sec/threadDump-2019083
0-123710.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/0bf6ab7f94a21cdc9c1649f884333ec20f40a544/index.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/18ae9b1c3e70c6b6064b6bc10544cb86296e48ae/index.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/4c237d5e33167c88df3e45d9c8b59fdd4d727472/index.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/579544fd7d0441717f082c9eb123588966aa57ac/index.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/5b32c3d2f81e7666f4c97a5ef68e9aa2898bf3df/index.txt
/home/kalpana/.config/google-chrome/Default/Service Worker/CacheStorage/76ef9e5953a1ba4578548bb32235240a9f0e0ca2/index.txt
```

nn) find:- It is 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

Syntax:

$ find ./new_folder2 -name file1.txt

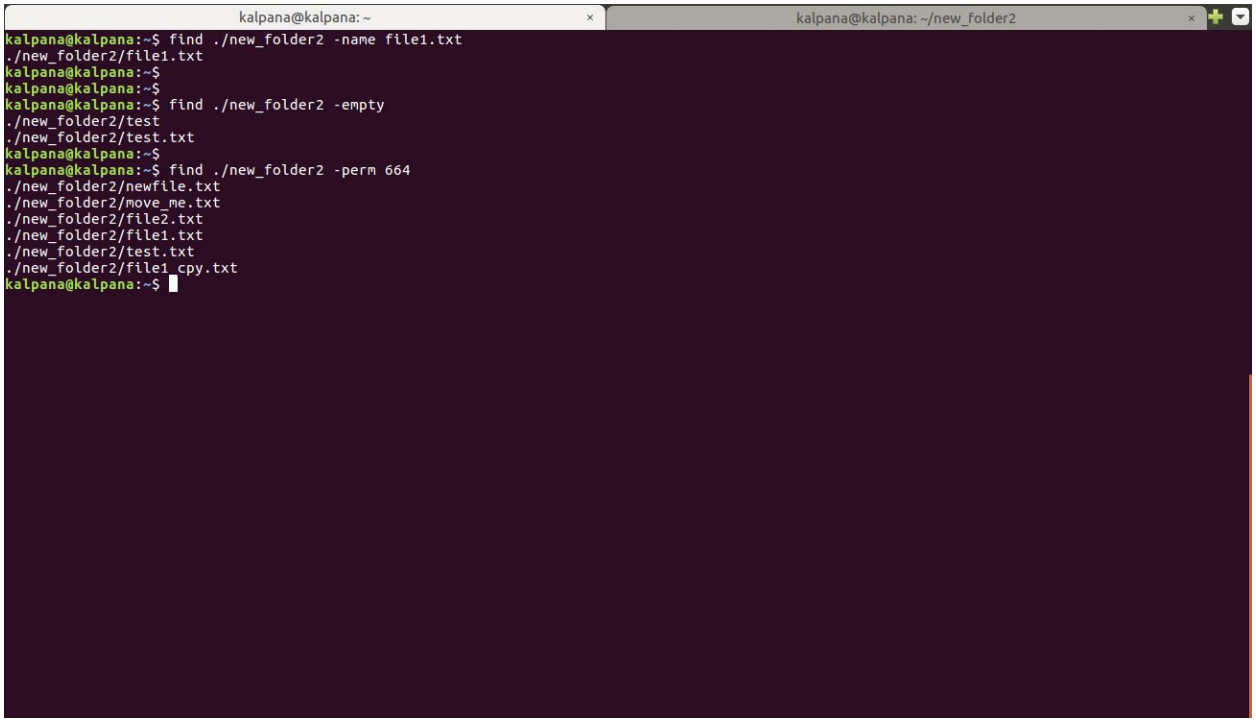
- search for file1.txt in new_folder2 directory.

$ find ./new_folder2 -empty

- Finds all empty folders and files in the entered directory or sub-directories.

$ find ./new_folder2 -perm 664

- Finds all the files with the given permission



oo) sed:- It stands for stream editor. It uses functions on file like searching, find and replace, insertion or deletion. It is most commonly used for substitution and to find and replace.

Syntax:

$ sed "s/text/test/g" newfile.txt

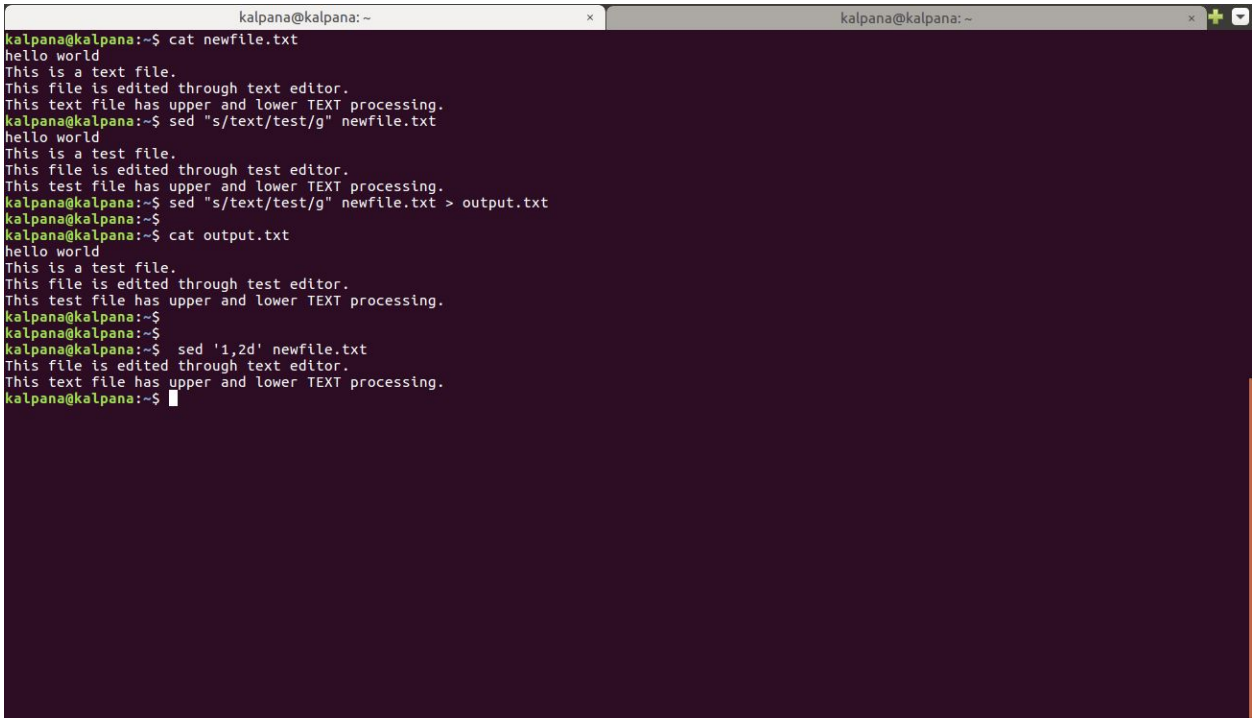
- Replaces the word "text" with test

$ sed "s/text/test/g" newfile.txt > output.txt

- Modifies the file and saves the input into "output.txt"

$ sed '1,2d' newfile.txt

- Deletes lines from 1 to 2 and displays the output on the terminal



pp) awk – It is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

Syntax:

$ awk '{print}' newfile.txt

- It prints all th lines of the file

$ awk '/TEXT/ {print}' newfile.txt

- Searches and prints the lines which has key "TEXT"

$ awk '{print $1,$NF}' newfile.txt

- Prints first and last words.

$ awk 'BEGIN { for(i=1;i<=6;i++) print "square of", i, "is",i*i; }'

- Sample aws script to print square root of numbers from 1 to 6.

```
kalpana@kalpana:~$ awk '{print}' newfile.txt
hello world
This is a text file.
This file is edited through text editor.
This text file has upper and lower TEXT processing.
kalpana@kalpana:~$
kalpana@kalpana:~$  awk '/TEXT/ {print}' newfile.txt
This text file has upper and lower TEXT processing.
kalpana@kalpana:~$
kalpana@kalpana:~$ awk '{print $1,$NF}' newfile.txt
hello world
This file.
This editor.
This processing.
kalpana@kalpana:~$
kalpana@kalpana:~$
kalpana@kalpana:~$ awk 'BEGIN { for(i=1;i<=6;i++) print "square of", i, "is",i*i; }'
square of 1 is 1
square of 2 is 4
square of 3 is 9
square of 4 is 16
square of 5 is 25
square of 6 is 36
kalpana@kalpana:~$
kalpana@kalpana:~$
```

qq) diff – It stands for difference. It shows the differences in the files by comparing line by line.
Syntax:
$ diff newfile.txt output.txt

- Shows the difference line by line.

```
kalpana@kalpana:~$ diff newfile.txt output.txt
2,4c2,4
< This is a text file.
< This file is edited through text editor.
< This text file has upper and lower TEXT processing.
---
> This is a test file.
> This file is edited through test editor.
> This test file has upper and lower TEXT processing.
kalpana@kalpana:~$
```

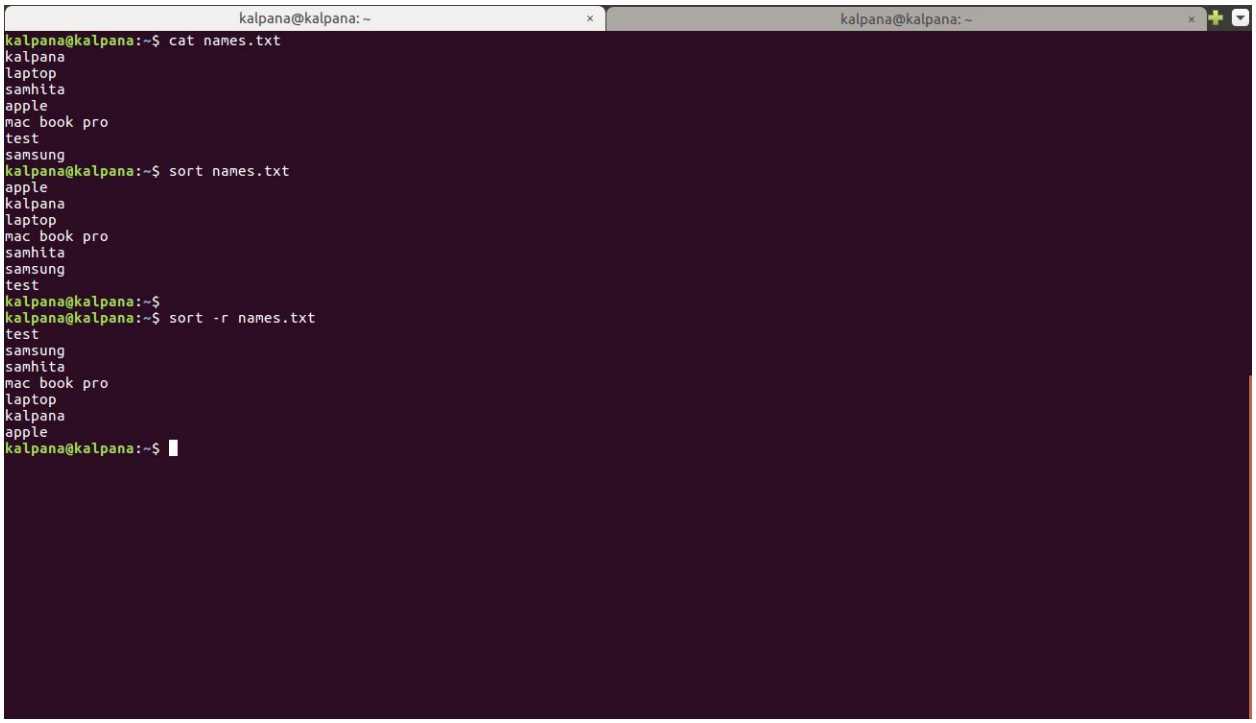
rr) sort:- It sorts the contents of a text file, line by line in the ASCII order.

Syntax:
$ sort names.txt

- Sorts the content line by line and displays on the terminal.

$ sort -r names.txt

- Sorts in the reverse order



ss) export:-It is used to export a variable or function to the environment of all the child processes running in the current shell.
Syntax:
$ export ENV=local
Exports the ENV variable as local into the current shell.

```
kalpana@kalpana: ~
kalpana@kalpana: ~
kalpana@kalpana:~$ echo $ENV

kalpana@kalpana:~$ export ENV="local"
kalpana@kalpana:~$
kalpana@kalpana:~$ echo $ENV
local
kalpana@kalpana:~$
```

tt) pwd:- It stands for Print Working Directory.It prints the path of the working directory, starting from the root.

Syntax:

$ pwd

- shows the present working directory from the root.

```
kalpana@kalpana: ~
kalpana@kalpana: ~/new_folder1
kalpana@kalpana:~$ pwd
/home/kalpana
kalpana@kalpana:~$ cd new_folder1/
kalpana@kalpana:~/new_folder1$ pwd
/home/kalpana/new_folder1
kalpana@kalpana:~/new_folder1$
```

uu) crontab:- It opens the cron table for editing. The cron table is the list of tasks scheduled to run at regular time intervals on the system using regular expressions.

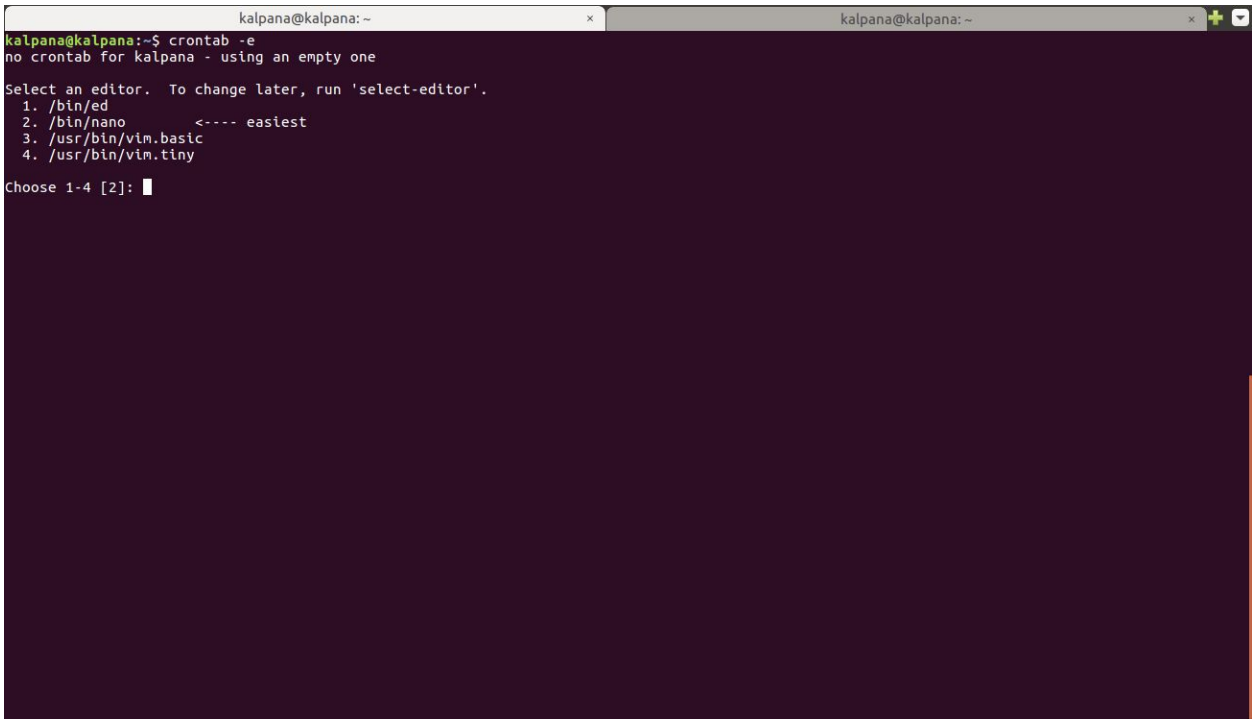
Syntax :

$ crontab -l

- List out the cron jobs

$ crontab -e

- opens the crontab jobs list in an editor



```
kalpana@kalpana:~$ crontab -l
no crontab for kalpana
kalpana@kalpana:~$ crontab -e
no crontab for kalpana - using an empty one
crontab: installing new crontab
kalpana@kalpana:~$
```

vv) mount:- The mount command serves to attach the file system found on some device to the big file tree.

Syntax:

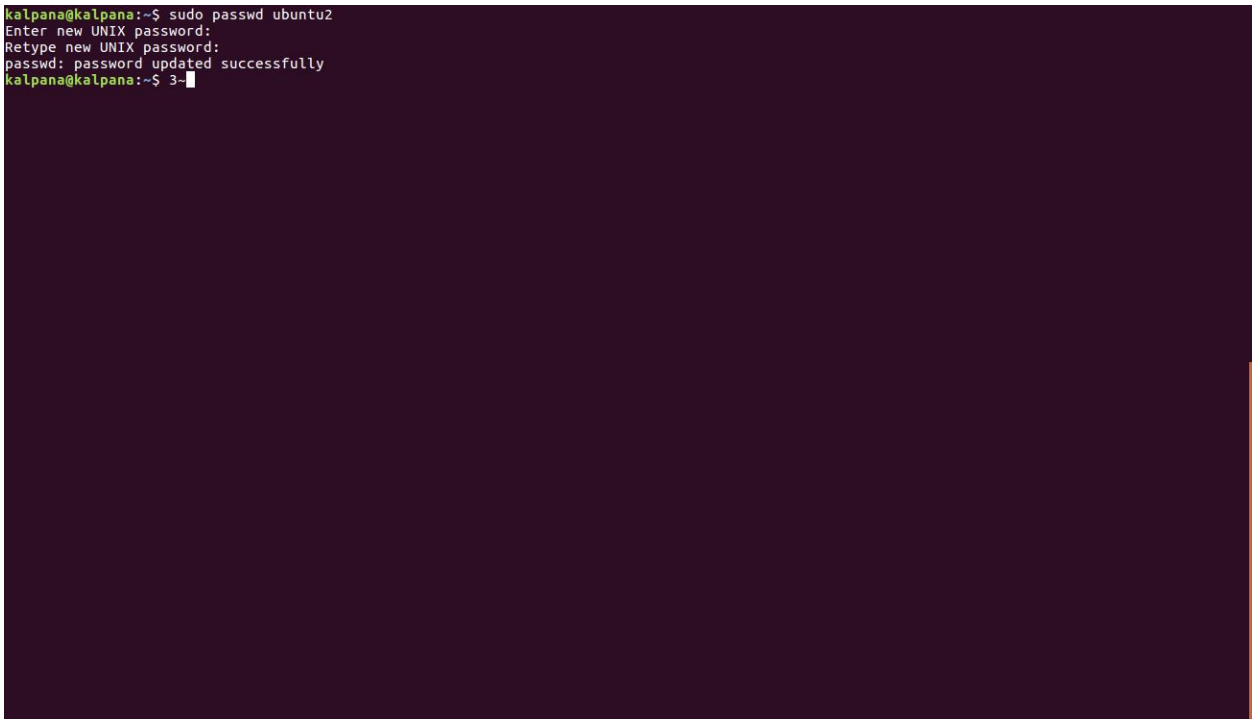
$ mount -t vfat /dev/sdb1 /media/usbstick

- It mounts the pen drive to “usbstick” folder

```
kalpana@kalpana:~$ sudo mkdir /media/usbstick
kalpana@kalpana:~$ sudo mount -t vfat /dev/sdb1 /media/usbstick
kalpana@kalpana:~$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop1    7:1    0  67.6M  1 loop /snap/sublime-text/69
sdb      8:16   1  28.9G  0 disk
├─sdb2   8:18   1  28.7G  0 part /media/kalpana/KP
└─sdb1   8:17   1   200M  0 part /media/usbstick
sr0     11:0    1  1024M  0 rom
loop2    7:2    0  88.7M  1 loop /snap/core/7396
loop0    7:0    0 420.7M  1 loop /snap/pycharm-professional/151
sda      8:0    0 931.5G  0 disk
├─sda2   8:2    0 930.1G  0 part /
├─sda3   8:3    0   977M  0 part [SWAP]
└─sda1   8:1    0   512M  0 part /boot/efi
kalpana@kalpana:~$
kalpana@kalpana:~$
```

ww) passwd:- It is used to create/change the user account passwords
Syntax:
$ sudo passwd ubuntu2

- Changes/creates the password for user “ubuntu2”



xx) uname:- It displays the information about the system
Syntax:
$ uname -a

- It prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system

$ uname -n

- It prints the hostname of the network node

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~/new_folder2
kalpana@kalpana:~$ uname
Linux
kalpana@kalpana:~$
kalpana@kalpana:~$
kalpana@kalpana:~$ uname -a
Linux kalpana 4.15.0-58-generic #64~16.04.1-Ubuntu SMP Wed Aug 7 14:10:35 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
kalpana@kalpana:~$
kalpana@kalpana:~$
kalpana@kalpana:~$ uname -n
kalpana
kalpana@kalpana:~$
```

yy) whereis:- It is used to find the location of source/binary file of a command and manuals sections for a specified file in Linux system.

Syntax:- whereis python3

- Shows the bin path2 to python3

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~/new_folder2
kalpana@kalpana:~$ whereis python3
python3: /usr/bin/python3.5m /usr/bin/python3.5 /usr/bin/python3 /usr/lib/python3.5 /usr/lib/python3 /etc/python3.5 /etc/python3 /usr/local/lib
/python3.5 /usr/include/python3.5m /usr/share/python3 /usr/share/man/man1/python3.1.gz
kalpana@kalpana:~$
kalpana@kalpana:~$
kalpana@kalpana:~$
```

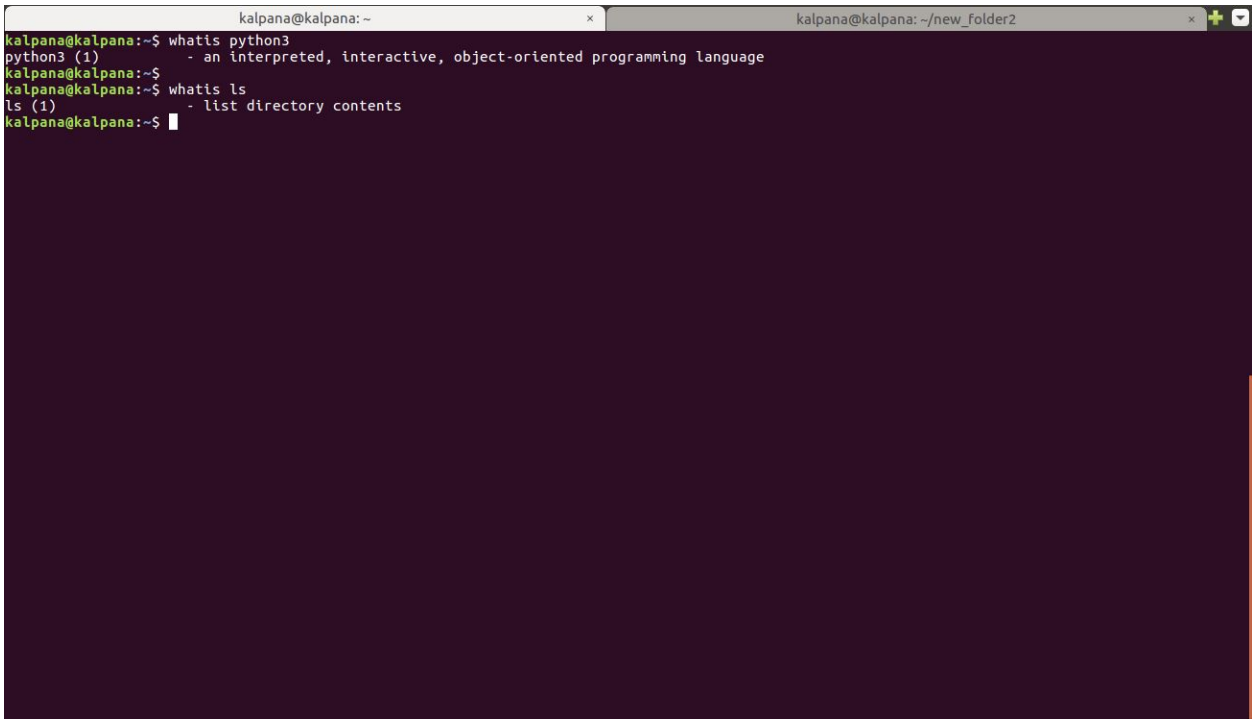
zz) whatis:- This command in Linux is used to get a one-line manual page descriptions.

Syntax:
$ whatis python3

- Displays brief information about python3

$ whatis ls

- Displays info about ls



aaa) less: - Less command is a linux utility which can be used to read contents of text file one page(one screen) per time. It has faster access because if a file is large, it doesn't access complete file, but it does page by page.

Syntax:
$ less Desktop/2_commands.txt

- Displays the first page of the file, and we can use up, down and q keys to move up, move down and quit.

```
kalpana@kalpana: ~                                  kalpana@kalpana: ~/new_folder2
2-d)
History:

        History command gives the last commands that have been run through the terminal. We can re-run, re-check and search commnds through his
tory.

Command usage:
i) history
        - Shows list of all commands used
ii) !2
        - Runs second command in the history file .bash_history
iii) history | grep vim
        - searching for commands that match a text pattern


Note: Another way to get to this search functionality is by typing Ctrl-R to invoke a recursive search of your command history.


2-e)
sudo
        - stands for super user do
        - we use to do administrative tasks

Command usage:
        i) sudo -i
                - login as a super user in a terminal
        2) sudo reboot
                - restarts the machine


2-f)
ip
        - stands for internet protocol
        - used to show or manipulate routing devices and tunnels

Command usage:
i) ip addr
        - shows the system IP address
ii) sudo ip link set wlp3s0 up
        - restart the interface wlp3s0 if it is down

Desktop/2_commands.txt
```

bbb) su:- su is used to switch from one account to another.

Syntax:

$ sudo su - ubuntu2

- Switches to another user ubuntu2

```
ubuntu2@kalpana: ~                                  kalpana@kalpana: ~
kalpana@kalpana:~$ ls /home/
kalpana  ubuntu2
kalpana@kalpana:~$ sudo su - ubuntu2
ubuntu2@kalpana:~$
```

ccc) ping:- It stands for Packet Internet Groper. It is used to check the network connectivity between host and server/host.

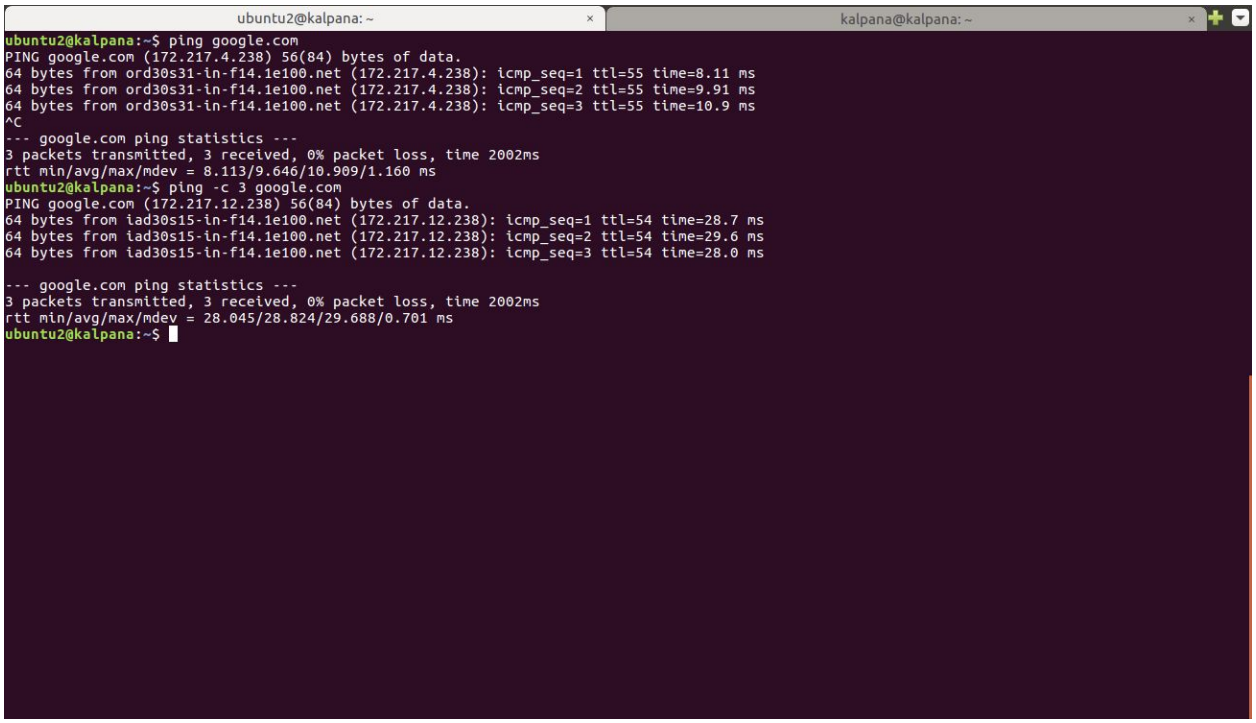
Syntax:

$ ping google.com

- Checks the connection to google. To stop pinging we should use ctrl+c otherwise it will keep on sending packets.

$ ping -c 3 google.com

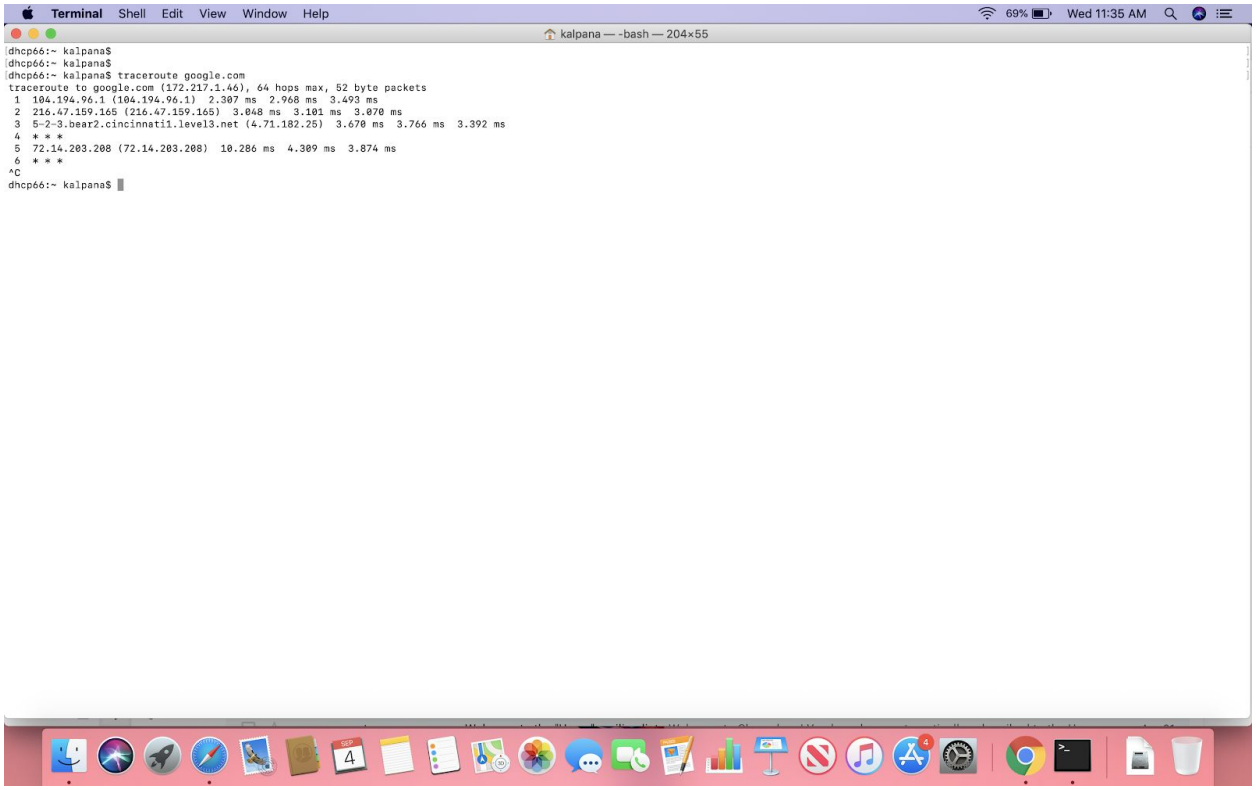
- Pings google for three times



ddd) traceroute:- This command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.

Syntax:

$ traceroute google.com



eee) date:- date command is used to display the system date and time

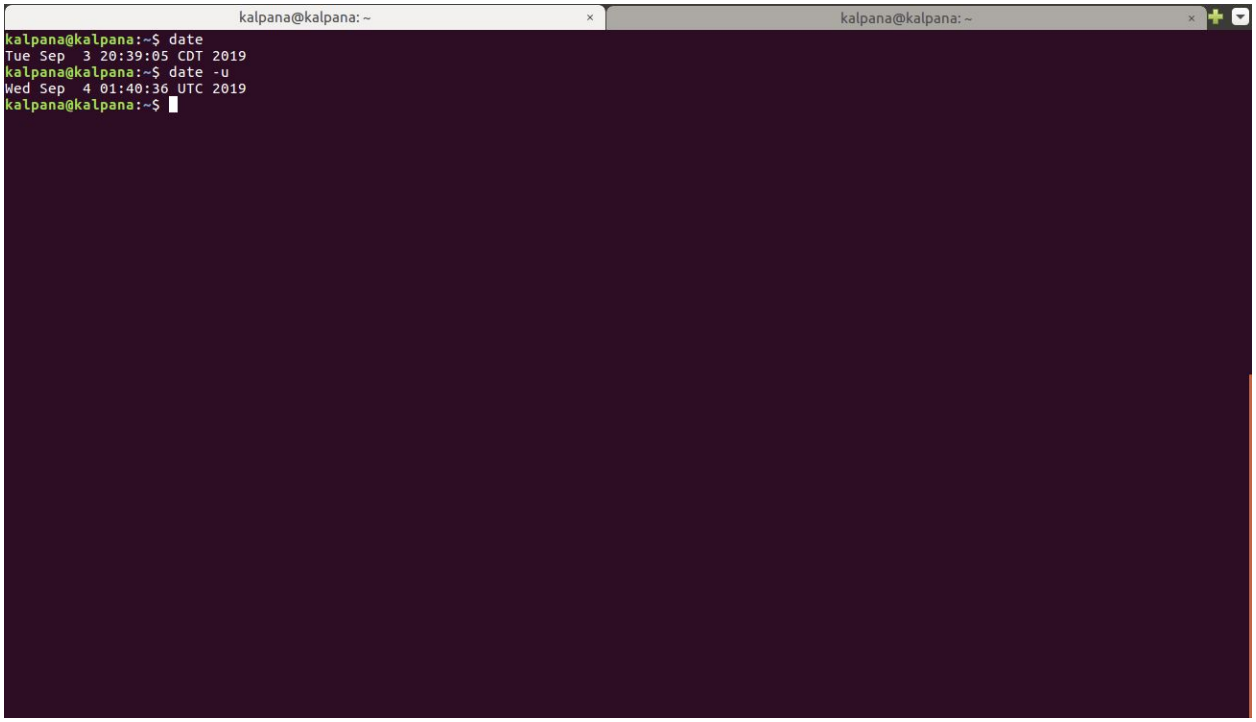
Syntax:

$ date

- Displays the system date

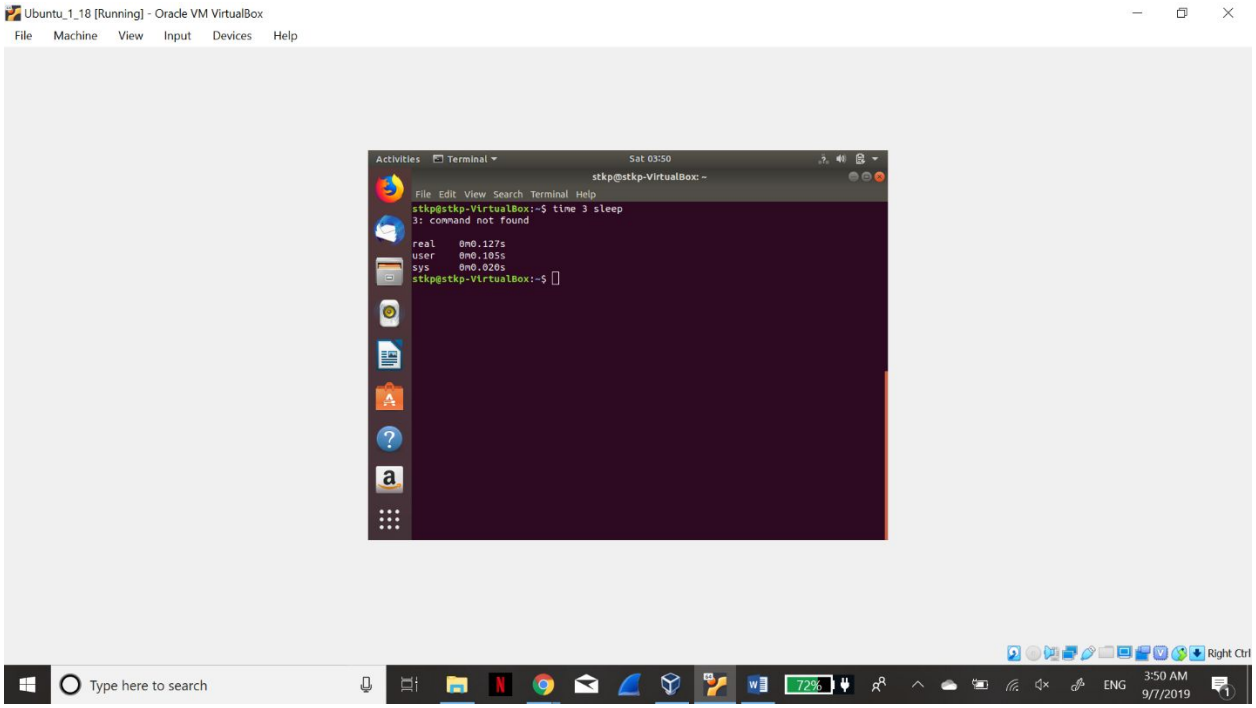
$ date -u

- Displays the time in GMT(Greenwich Mean Time)/UTC(Coordinated Universal Time) time zone.



fff) time:- time command is used to determine how long a given command takes to run
Syntax:-
$ time sleep 3

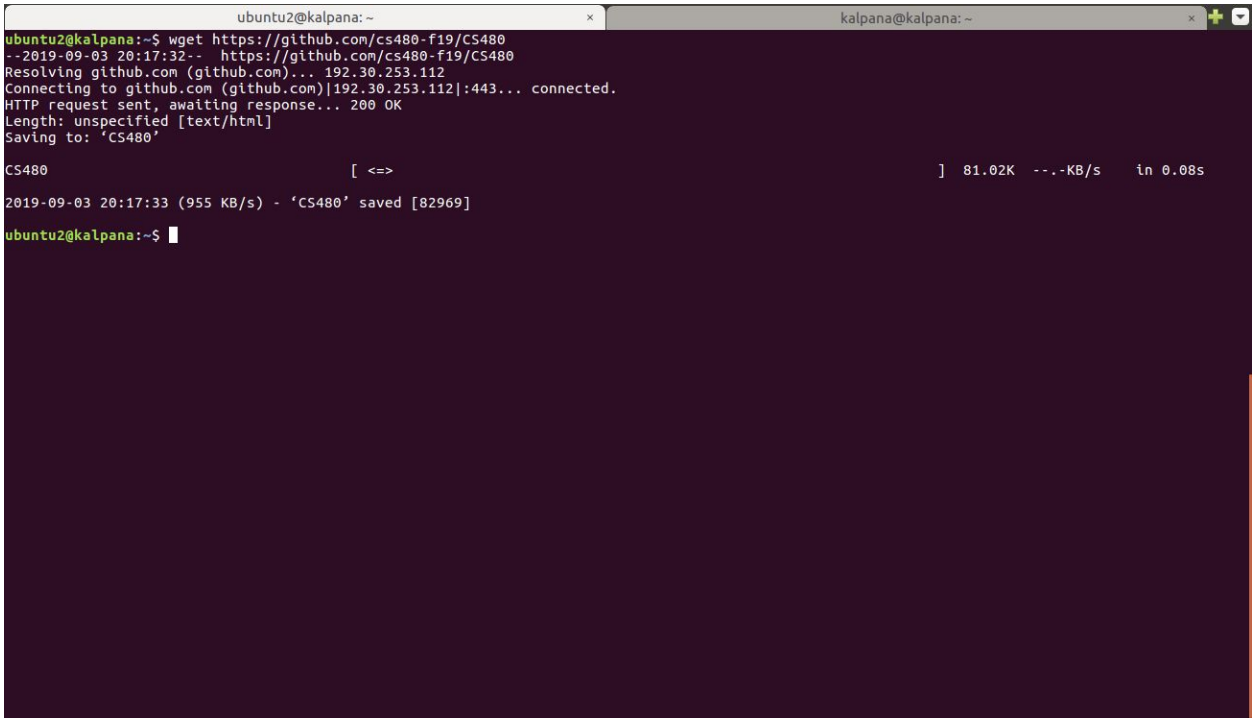
- creates a dummy job which lasts 3 seconds



ggg) wget:- Wget is the non-interactive network downloader, downloads the files from the server even when the user has not logged on to the system

Syntax: wget https://github.com/cs480-f19/CS480

- Downloads the files from github



hhh) wc:- wc stands for word count. As the name implies, it is mainly used for counting purpose.

Syntax:-

$ wc names.txt

- Displays the row, words and total letters count.

$ wc file2.txt

- Displays the word count

```
kalpana@kalpana:~$ cat file2.txt
file 1 txt
kalpana@kalpana:~$ cat names.txt
kalpana
laptop
samhita
apple
mac book pro
test
samsung
kalpana@kalpana:~$ wc file2.txt
 1  3 11 file2.txt
kalpana@kalpana:~$ wc names.txt
 7  9 55 names.txt
kalpana@kalpana:~$
```

iii) clear:- used to clear the terminal screen

Syntax:

$ clear

- clears the screen

```
kalpana@kalpana:~$ clear
```

jjj. exit

Command used to exit the shell where it is currently running.

Syntax:

$ exit

- exits the terminal

$ sudo -i

$ exit

- exit the root directory