Linux Class

 # Linux Notes

Linux Command

Cp command

- Cp command is used to copy the one or more files.

- **OPTIONS**
  - -i will ask for confirmation to overwrite or not.
  - -r recursively copies all files/subdirectories .

- **For Example**

1) **$cp file1 file2**
   - Copy File1 into file2. If file2 is not exist it create it otherwise overwrite it

2) **$cp file1 dir1/file3**
   - Copy File1 into file3 which is in the dir1.

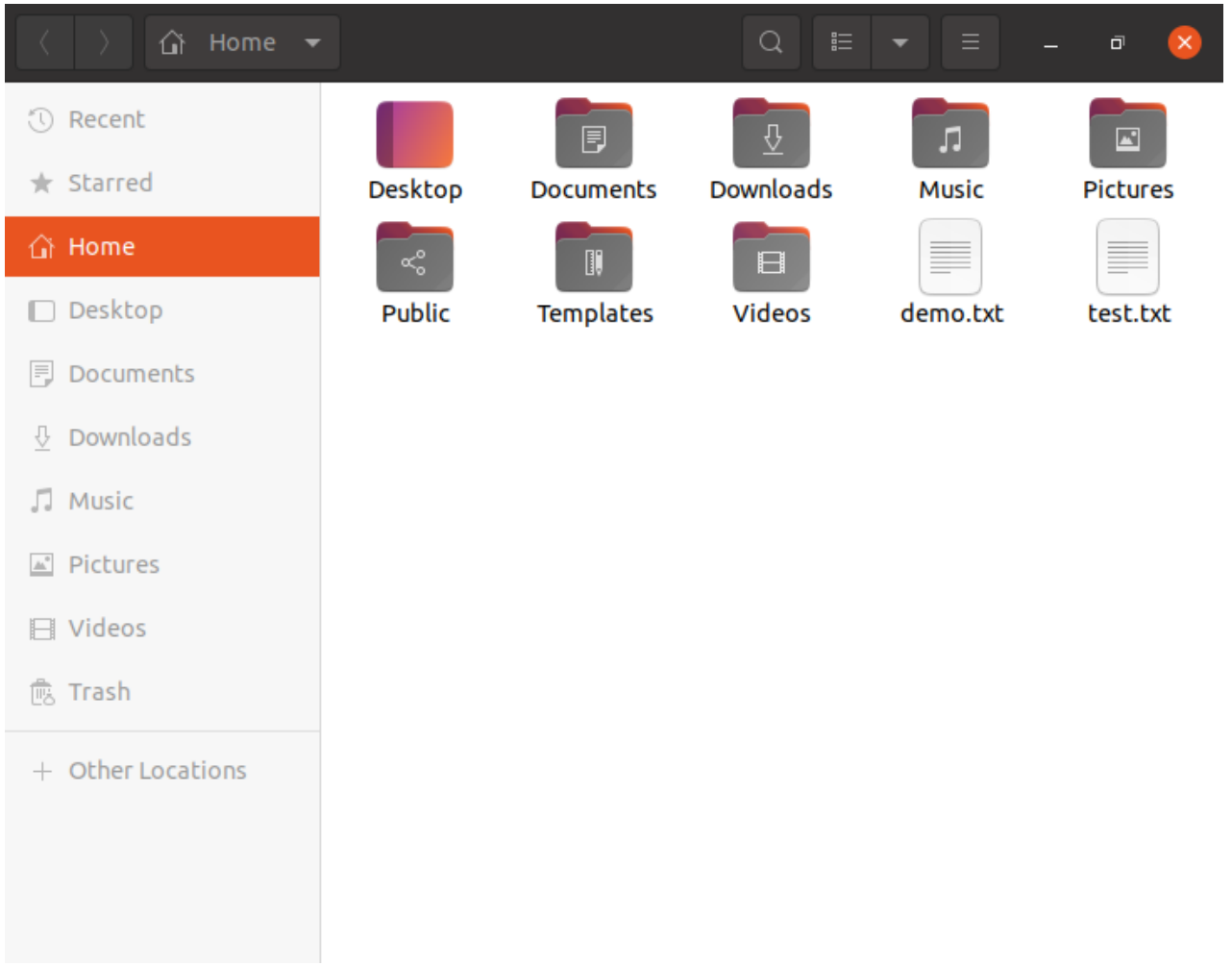
3) **$cp –i file1 file2**

- This options is asked to overwrite or not.

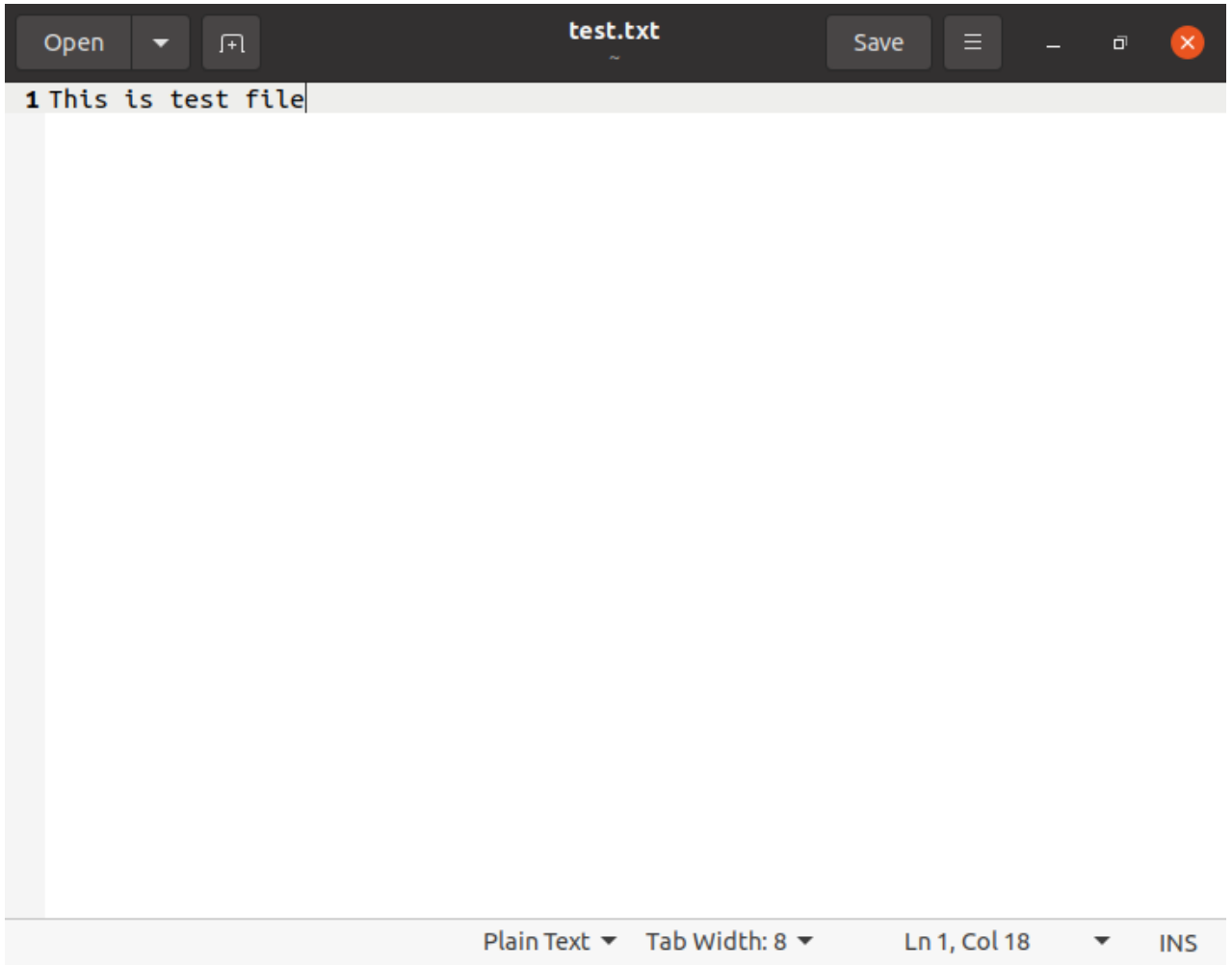
4) **$cp -i -r dir1 dir2**

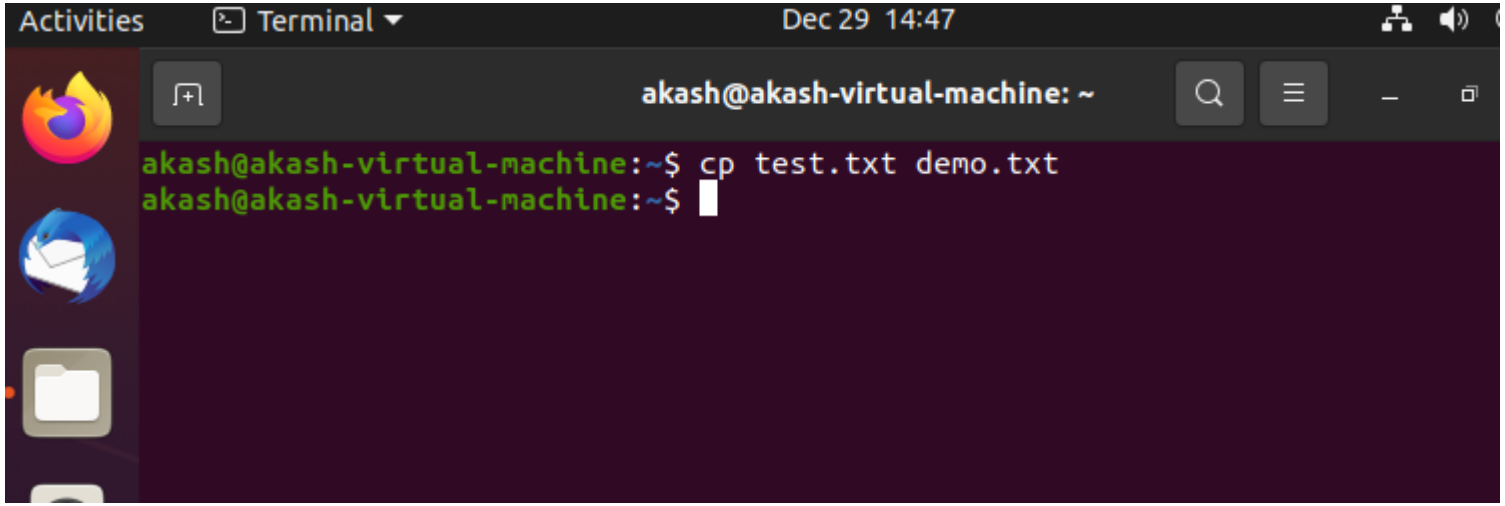
- This command is used to copy all files and directories of dir1 into dir2 If any file or directory of dir1 is already exist in dir2 then it will give prompt to overwrite.

File Stucture

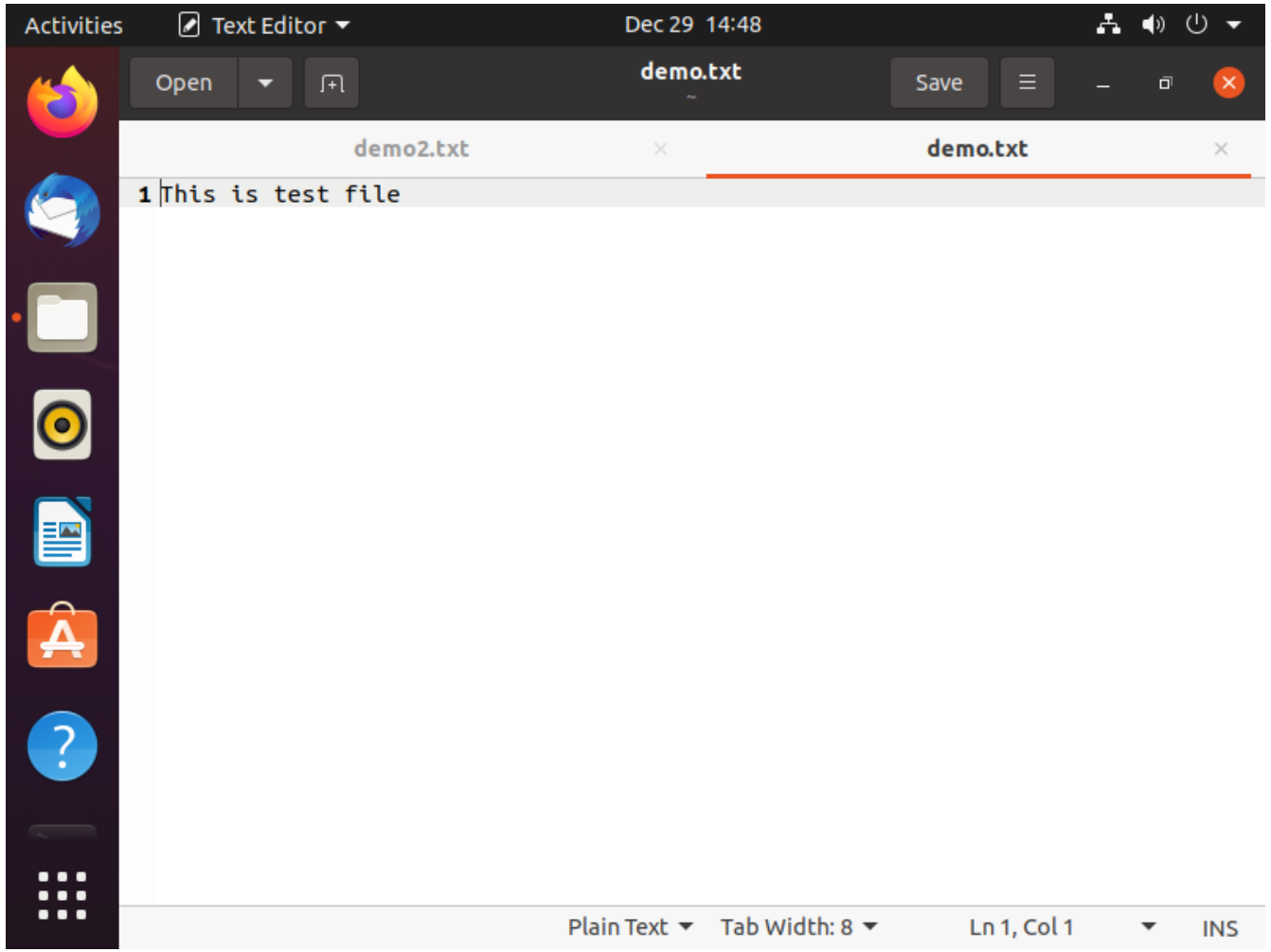


Text.txt


Open
test.txt
Save
1 This is test file
Plain Text
Tab Width: 8
Ln 1, Col 18
INS


Activities
Terminal
Dec 29 14:47
akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ cp test.txt demo.txt
akash@akash-virtual-machine:~$

Demo.txt


Activities
Text Editor
Dec 29 14:48
Open
demo.txt
Save
demo2.txt
demo.txt
1 This is test file
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

Rm Command

- This command is used to delete the file.

- **OPTIONS**
  - -i will ask for confirmation to remove or not.
  - -r Recursively remove all files/subdirectories .

Rm Command

- **For Example**

1) **$rm file1**
   - Delete file1.

2) **$rm file1 file2**
   - Delete multiple files.

3) **$rm file***
   - Delete all the file start with name of file.

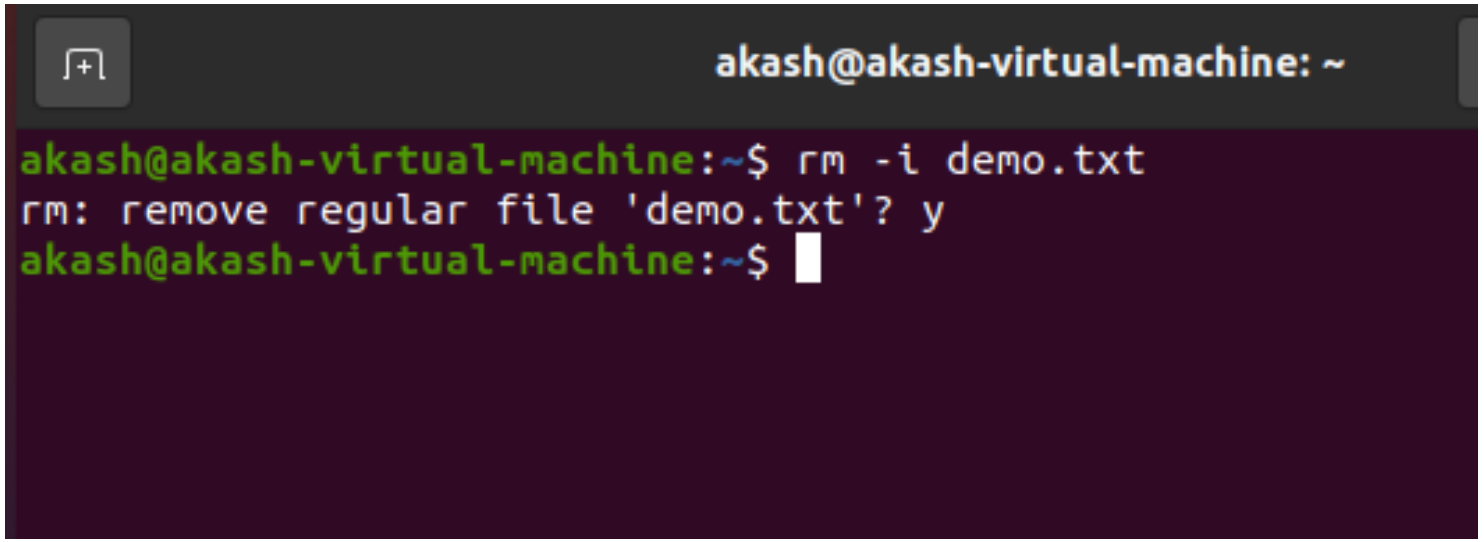
4) **$rm -i file1**

- This command prompt the user for confirmation before delete the file.

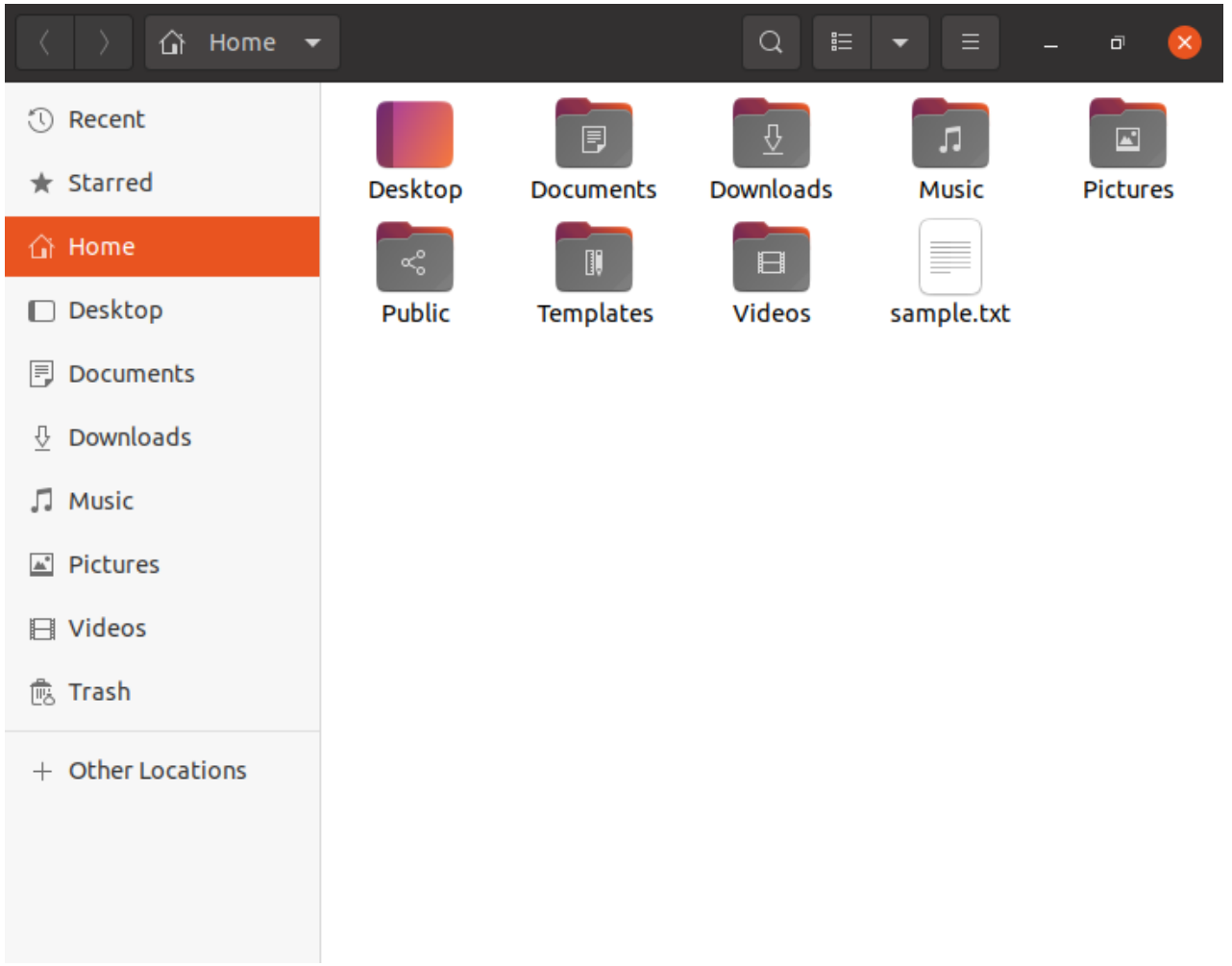
5) **$rm - r ***

- This command delete all files, directories and subdirectories of the current directory.

Rm -i


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ rm -i demo.txt
rm: remove regular file 'demo.txt'? y
akash@akash-virtual-machine:~$

File structure



Mv Command

- This Command is used to rename the file name also directory.
- **For Example**

1)**$mv file1 file2**

- File1 is rename as file2.

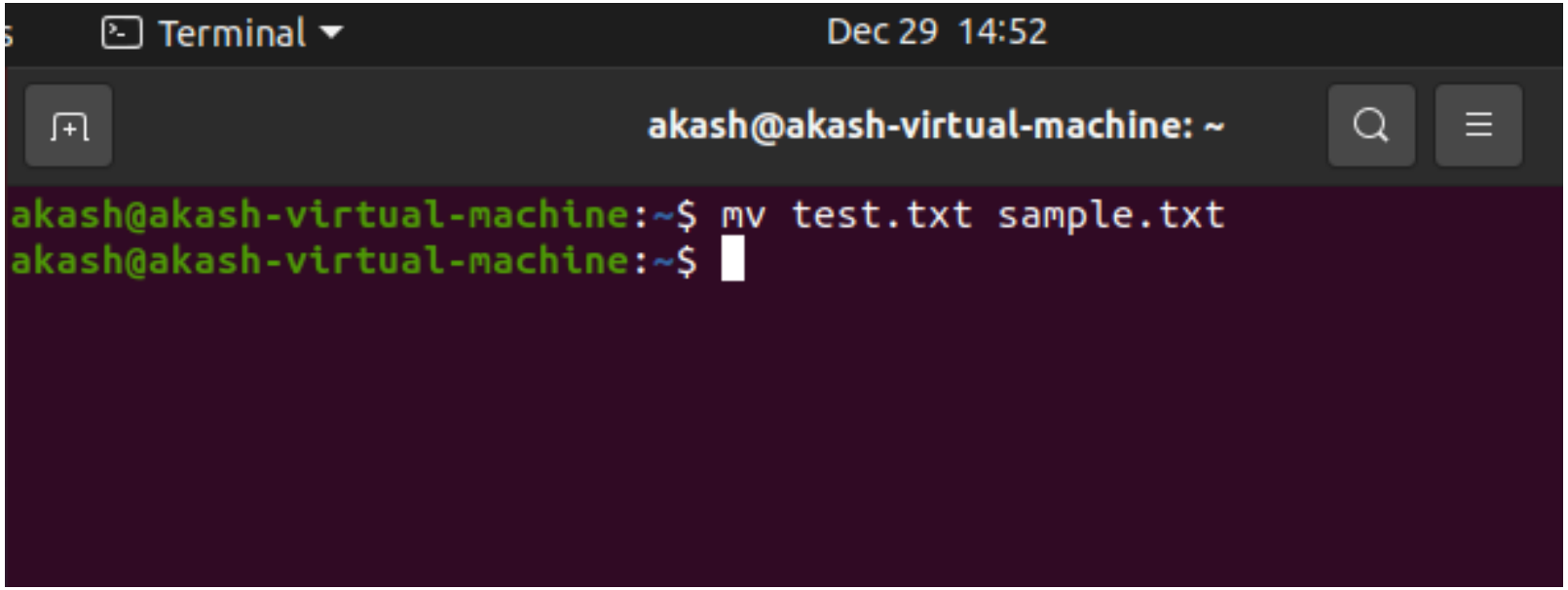
2)**$mv dir1 newdir1**

- Rename directory dir1 as newdir1.

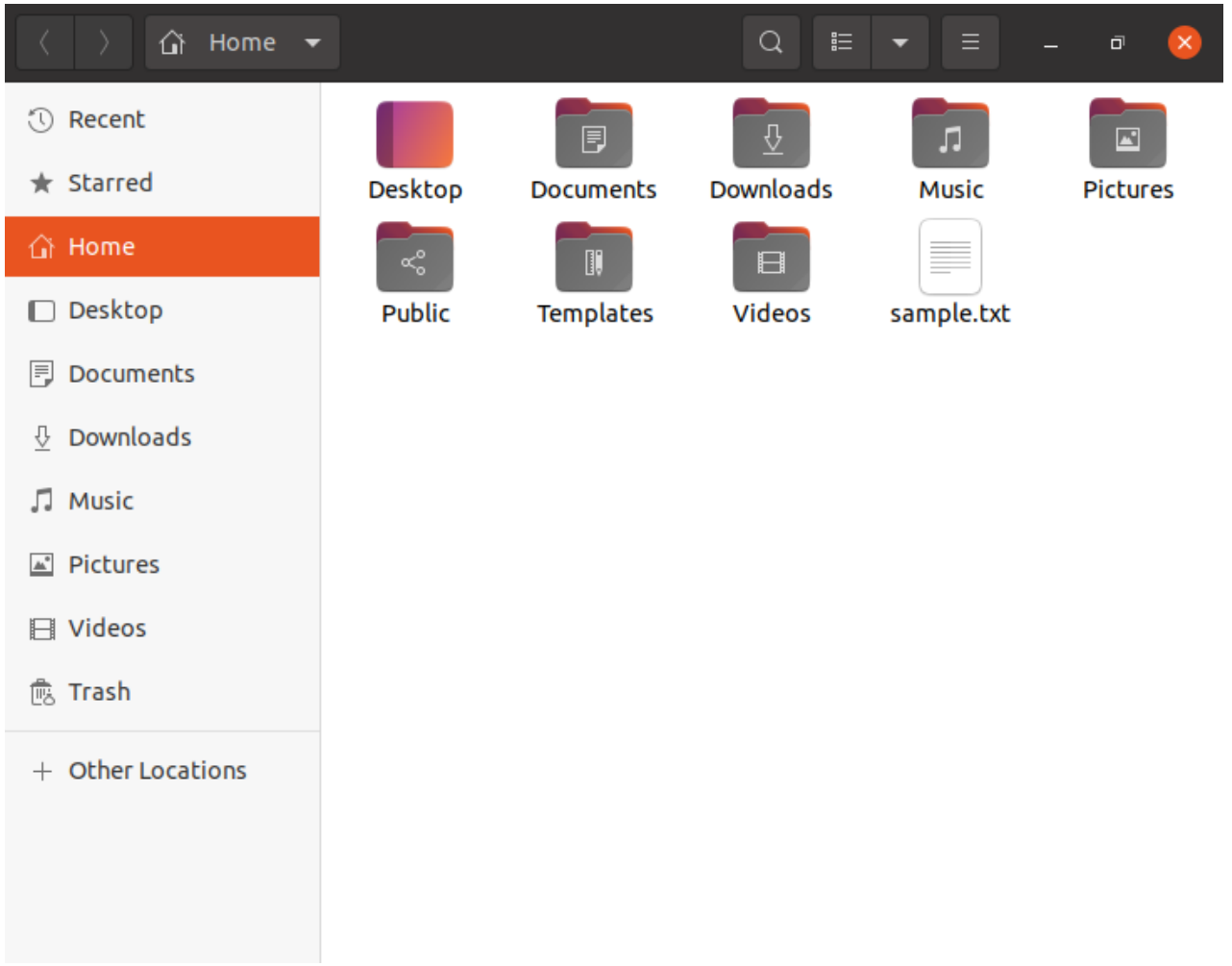
3)**$ mv ../file2 .**

- It will move the file2 from the parent directory to the current directory

Rename file


Terminal
Dec 29 14:52
akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ mv test.txt sample.txt
akash@akash-virtual-machine:~$

File structure



Mkdir Command

- This command is used to create directory.

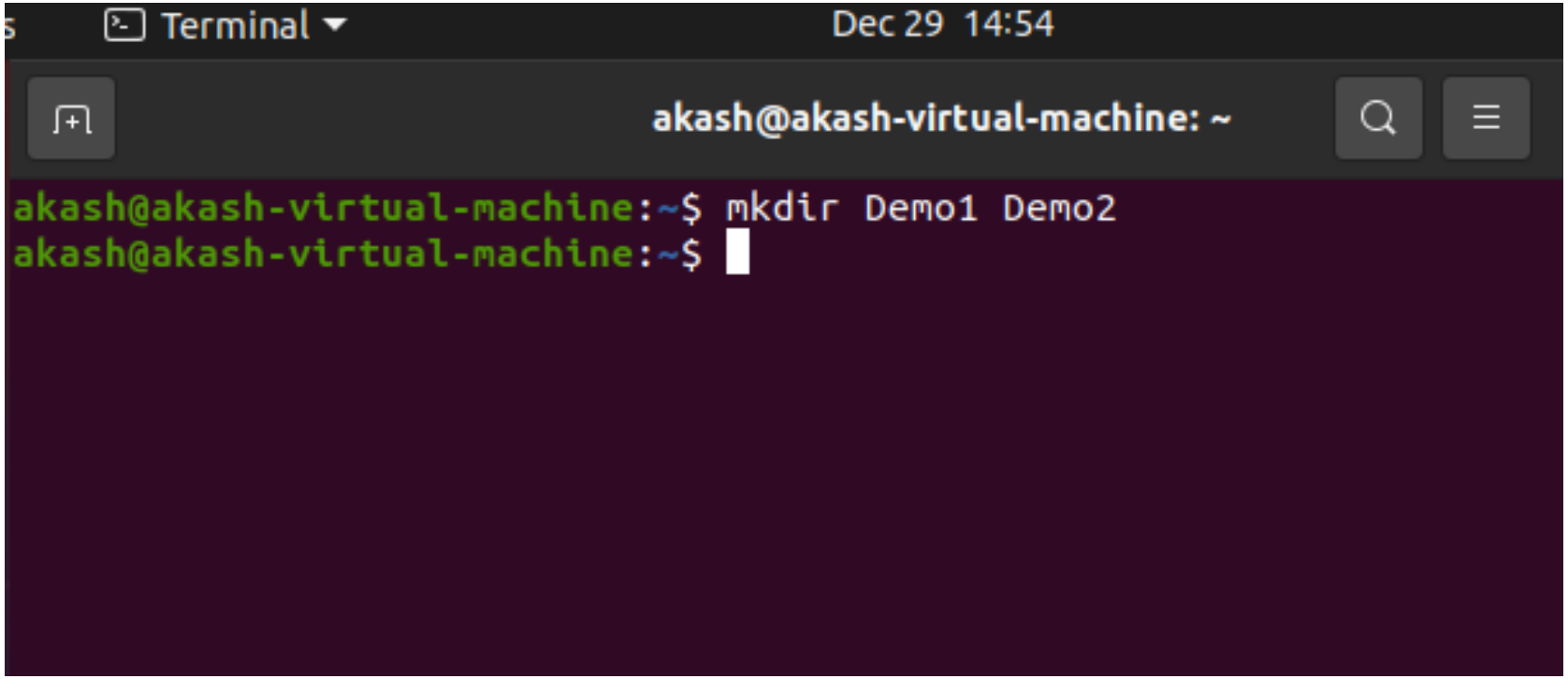
- **For example**

1) **$mkdir dir2**
   - create dir2

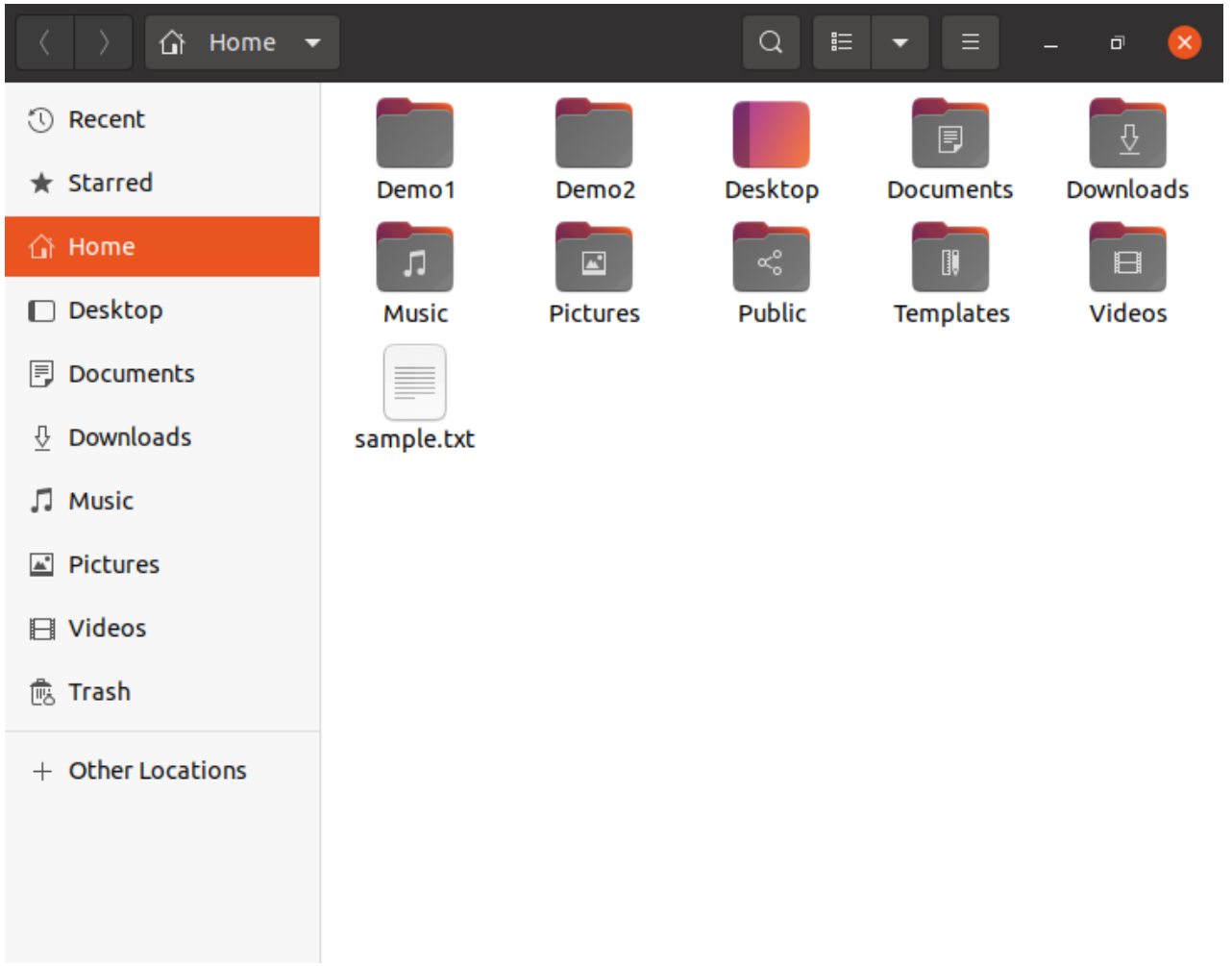
2) **$mkdir dir1 dir2 dir 3**
   - create three directories.

3) **$mkdir dir1 dir1/d1**
   - make a directory tree. It first create dir1 and create d1 directory into dir1.

Create Directory



File structure



ls Command

- The ls command lists all files in the directory that match the name. If name is left blank, it will list all of the files in the directory.

- The syntax for the ls command is:
  - **ls [options] [names]**

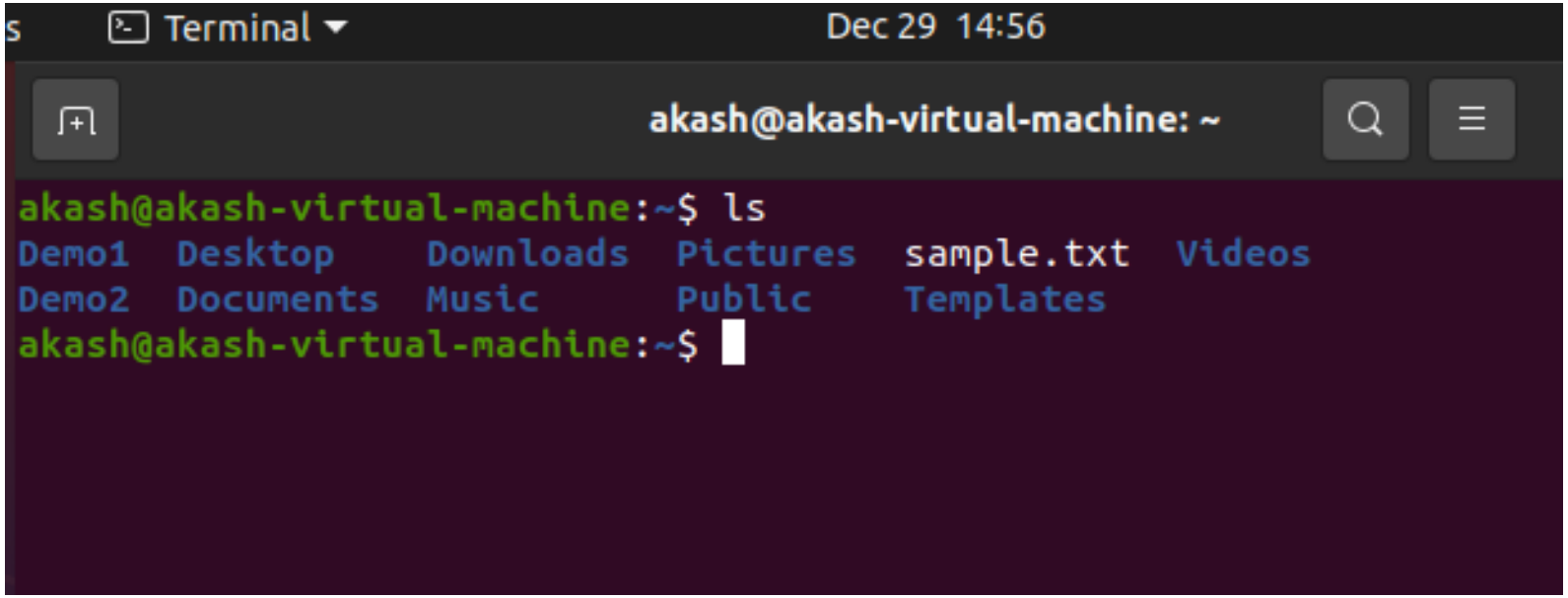
ls Command

- **$ ls –a** Displays all file including . , .. And those beginning with .
- **$ ls –x** Displays files as rows across the screen.
- **$ ls –F** Marks executables with * and directories with /
- **$ ls –r** Displays files in reverse order.
- **$ ls –R** Displays subdirectories as well.
- **$ ls –l** Displays the long format listing. Long listing showing seven attributes of a file.
- **$ ls –D** Displays only directories.

ls Command

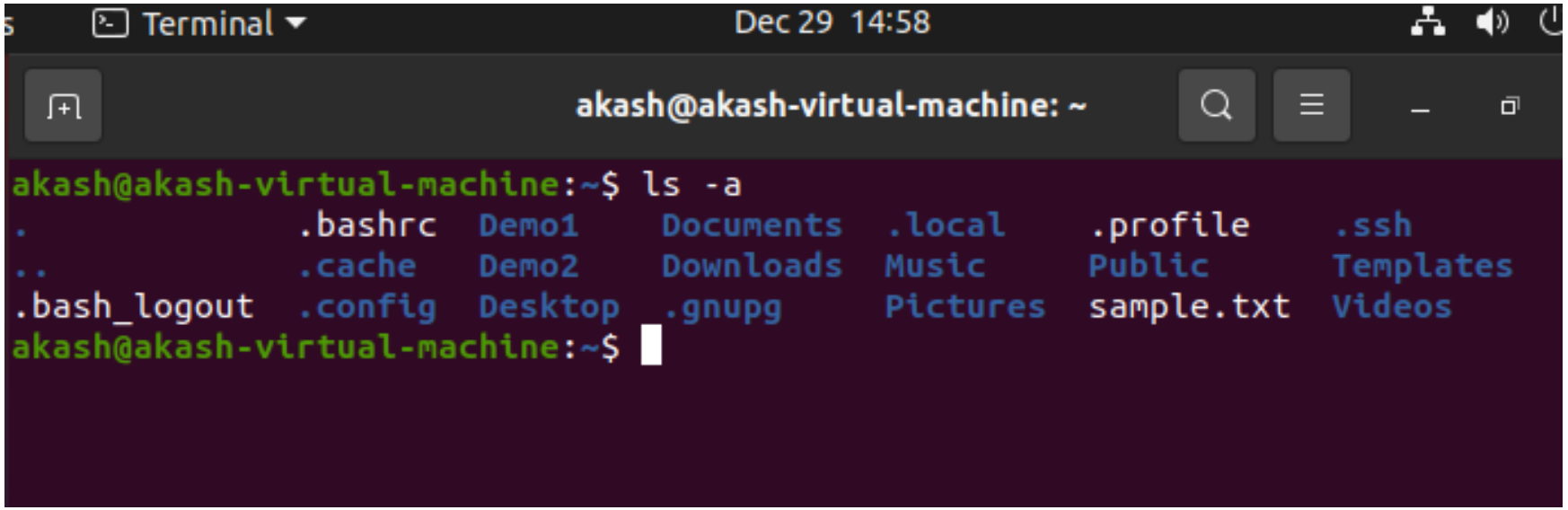
- **$ ls –t** Sorts files by modification time.
- **$ ls –u** Displays files by the file access time.
- **$ ls –i** Displays the inode number for each file.
- **$ ls –m** Displays the names as a comma-separated list.
- **$ ls –p** Displays directories with /

ls


Terminal
Dec 29 14:56
akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ls
Demo1 Desktop Downloads Pictures sample.txt Videos
Demo2 Documents Music Public Templates
akash@akash-virtual-machine:~$

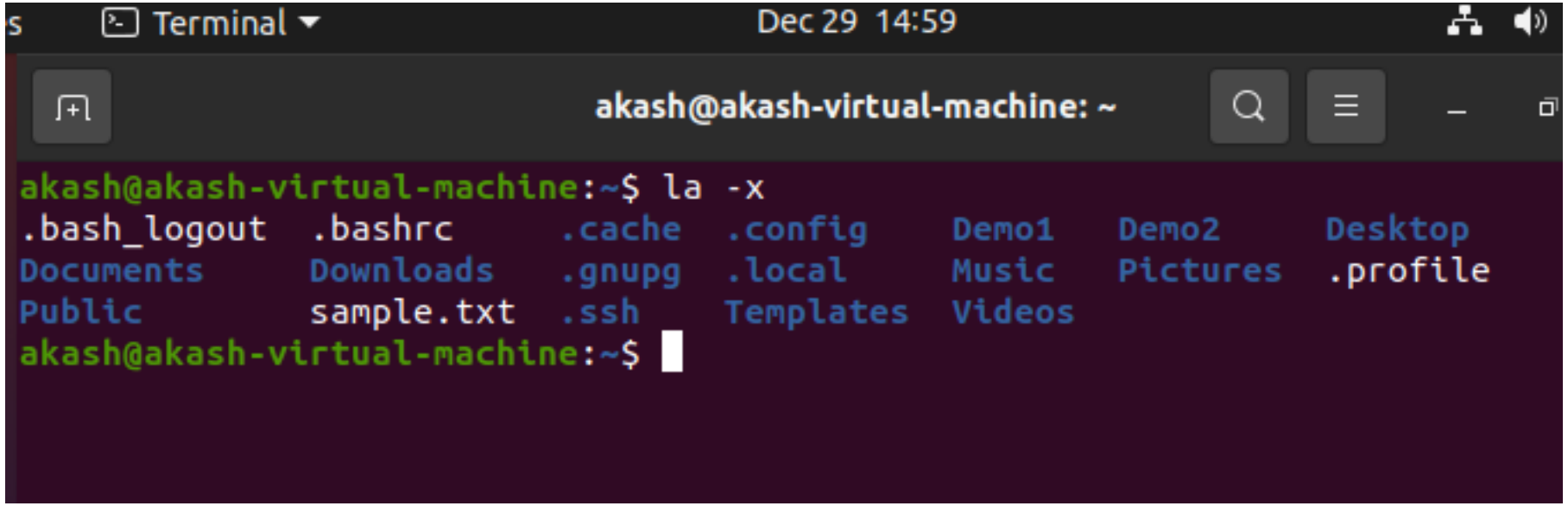
ls –a

$ ls –a Displays all file including . , .. And those beginning with . (DOT)



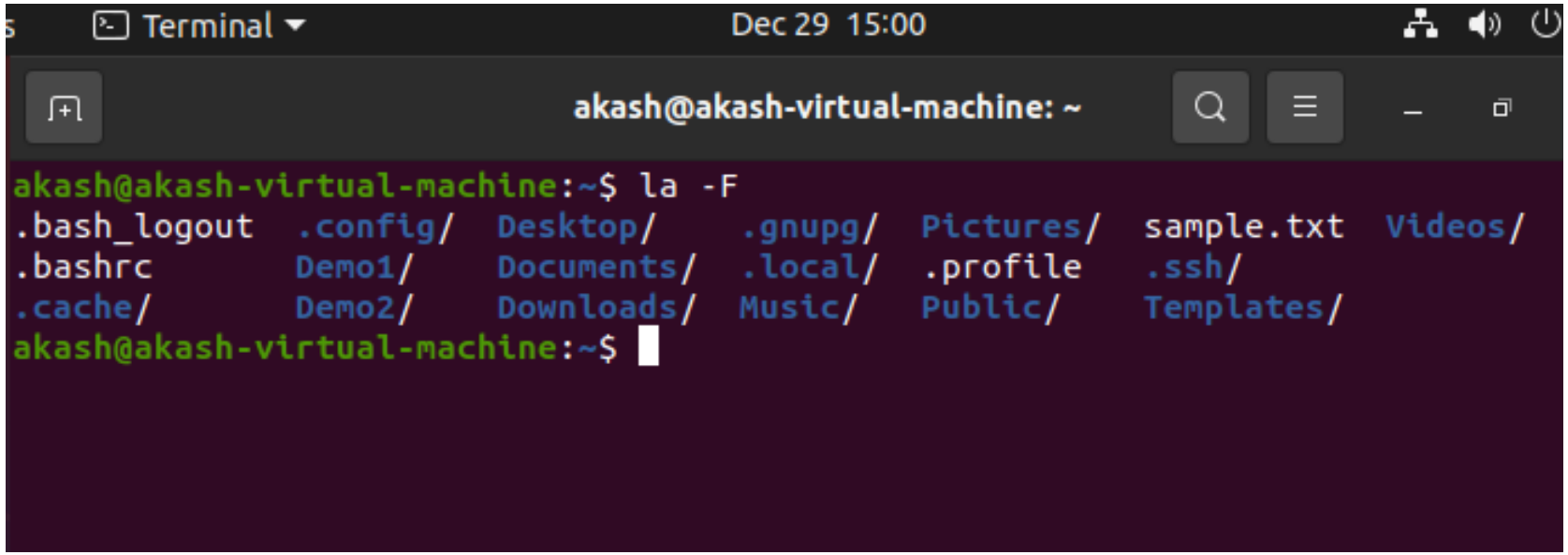
ls –x

$ ls –x Displays files as rows across the screen.



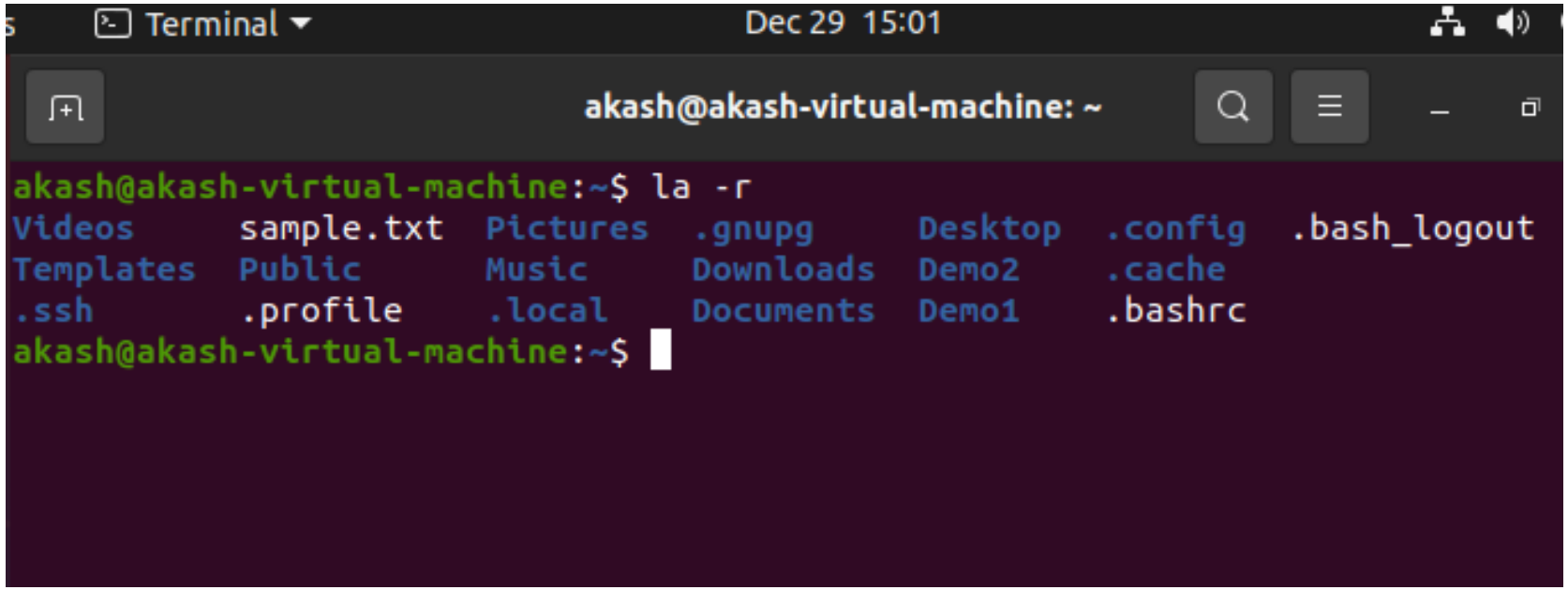
ls –F

$ ls –F Marks executables with * and directories with /



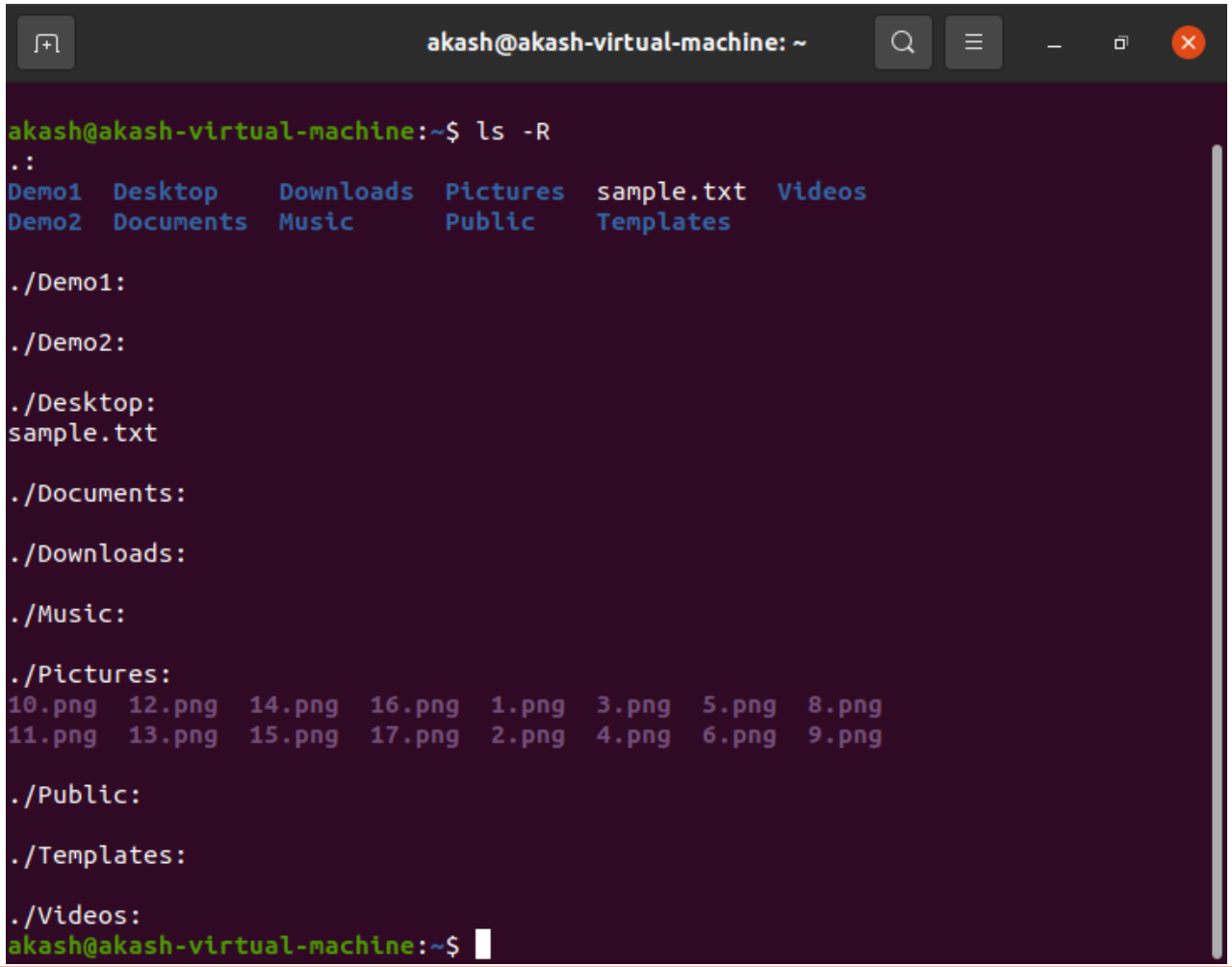
ls –r

$ ls –r Displays files in reverse order.



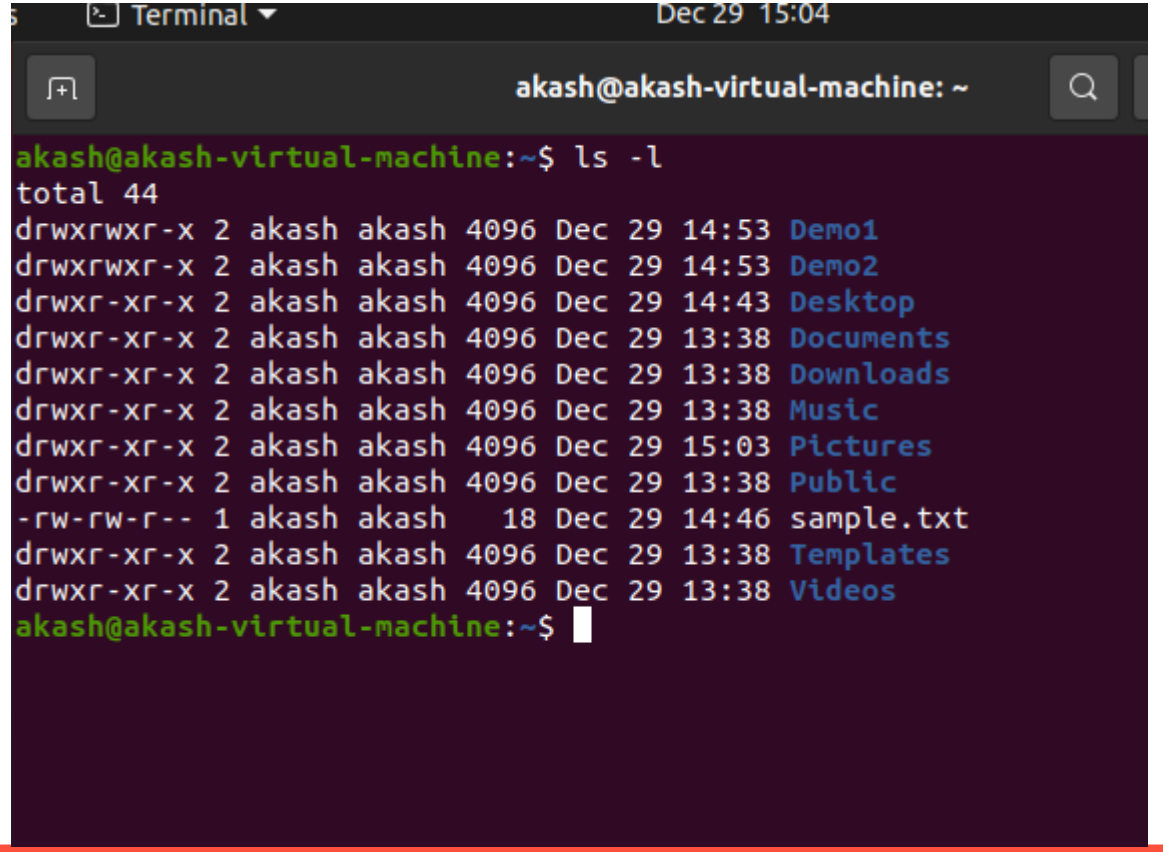
ls –R

$ ls –R Displays subdirectories as well.



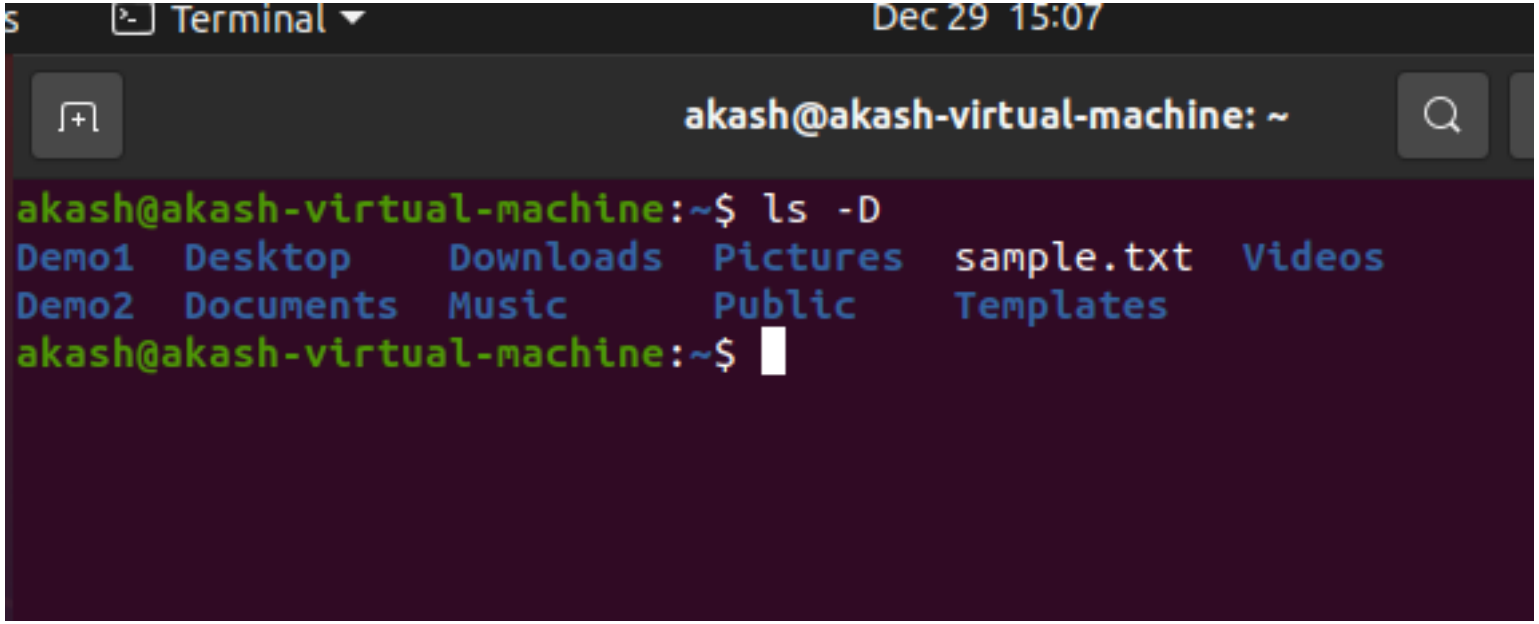
ls –l

$ ls –l Displays the long format listing. Long listing showing seven attributes of a file.



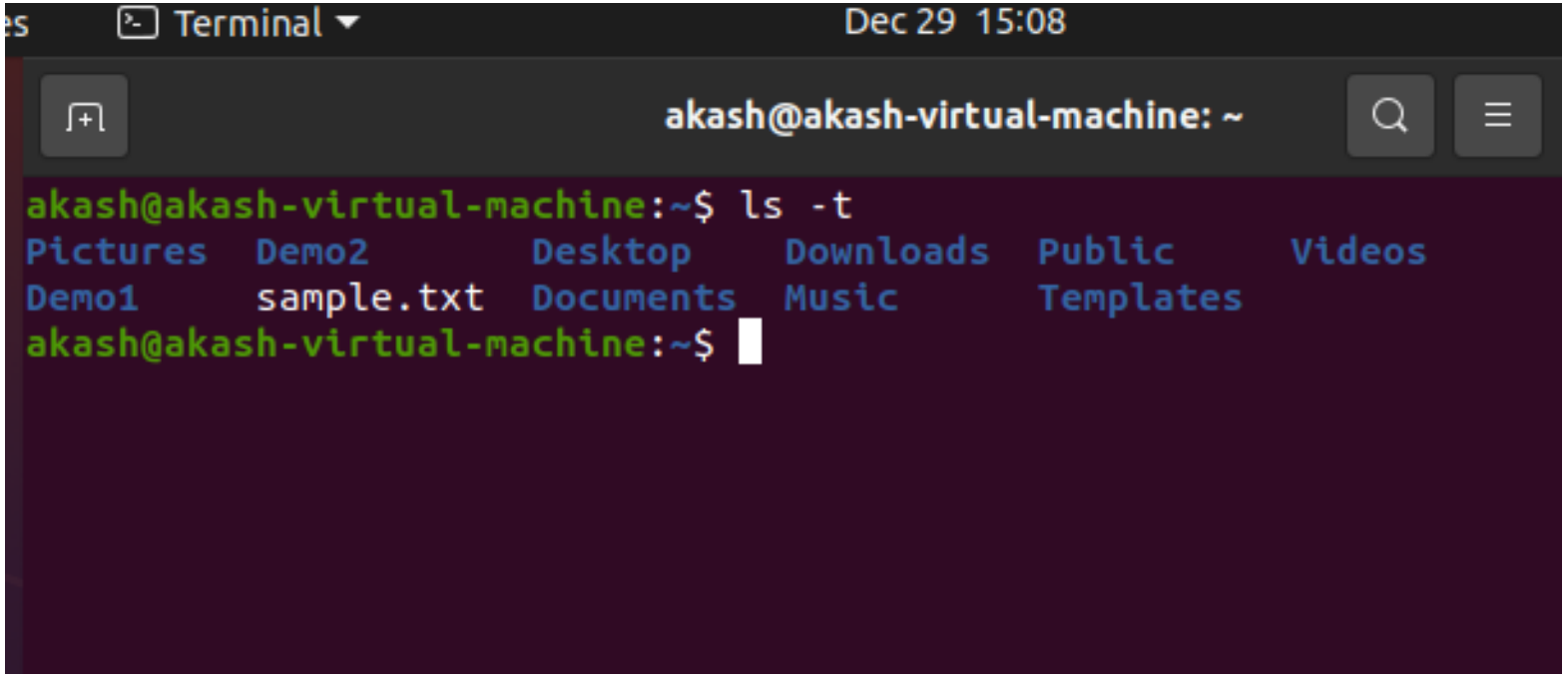
ls –D

$ ls –D Displays only directories.



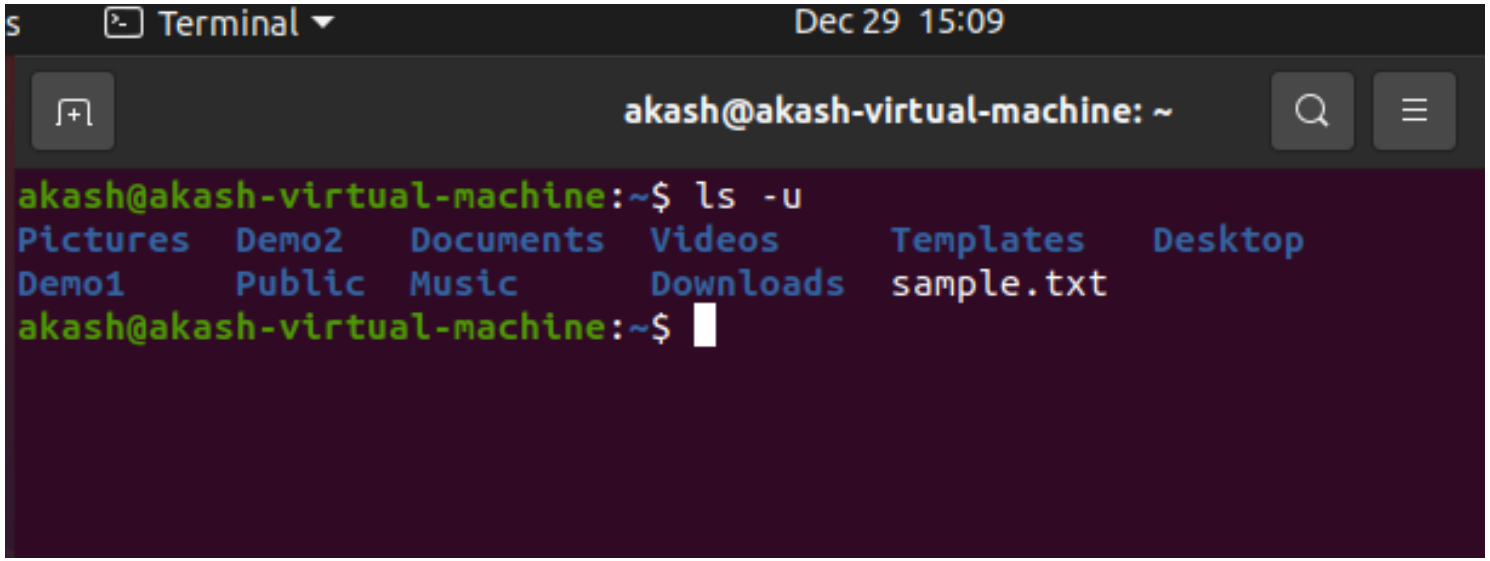
ls –t

$ ls –t Sorts files by modification time.



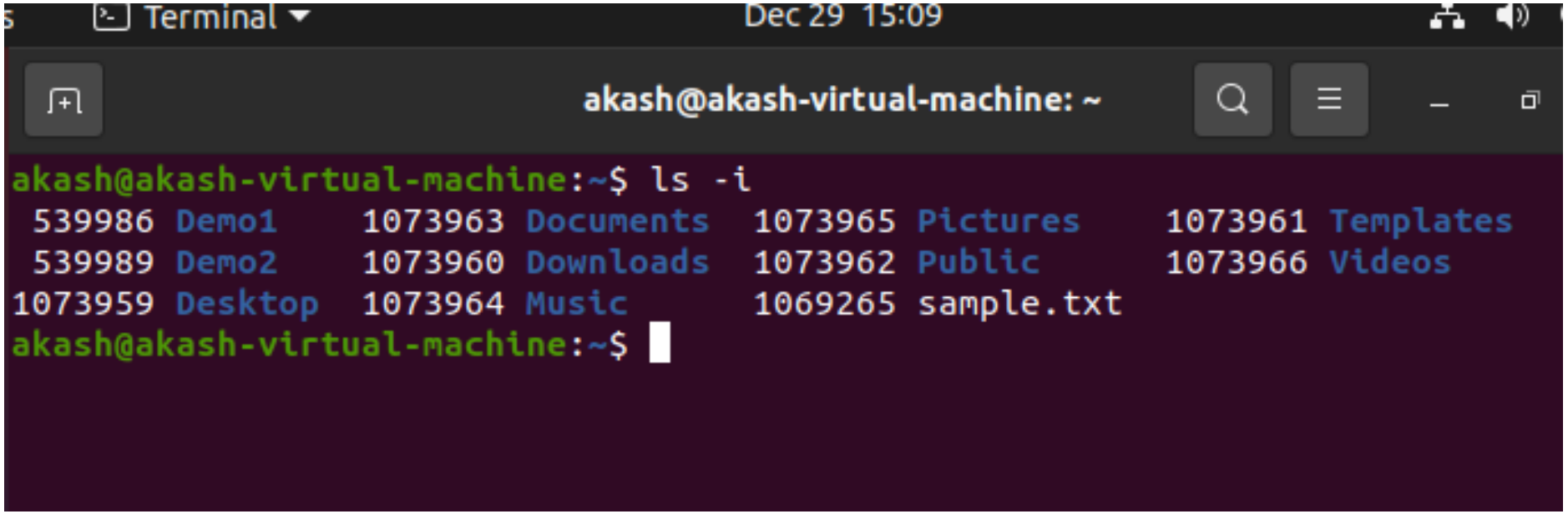
ls –u

ls –u Displays files by the file access time.



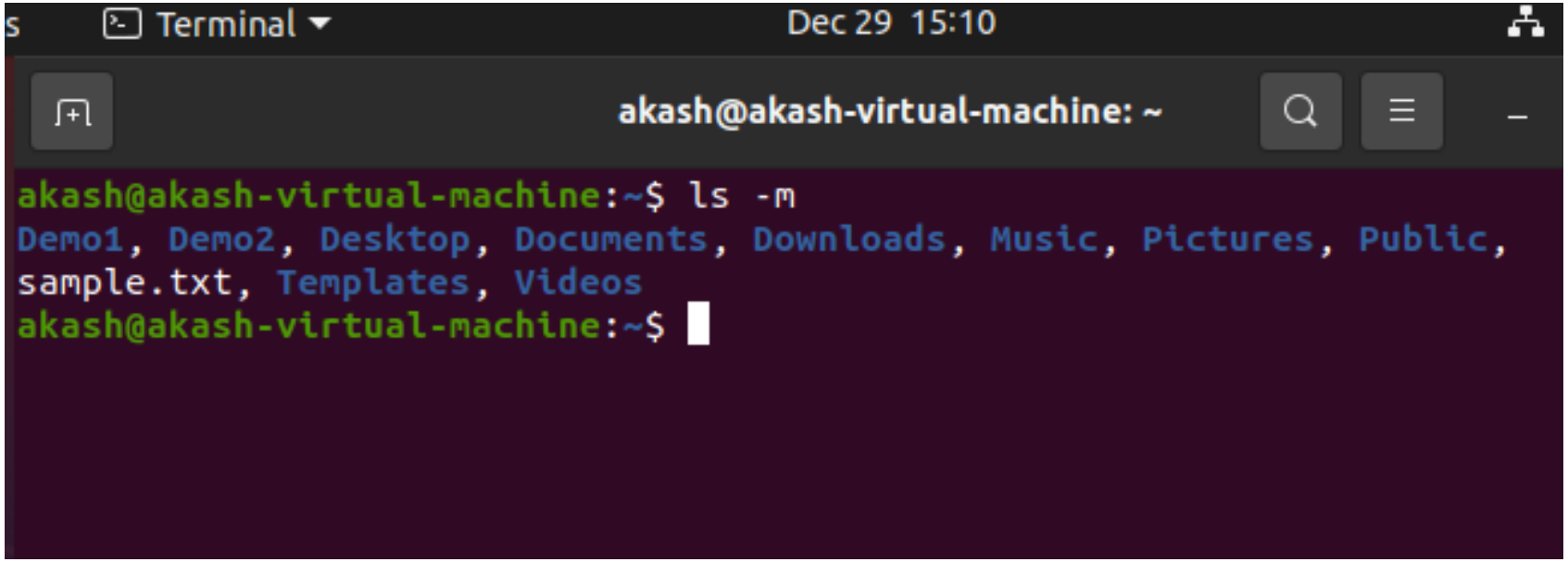
ls –i

ls –i Displays the inode number for each file.



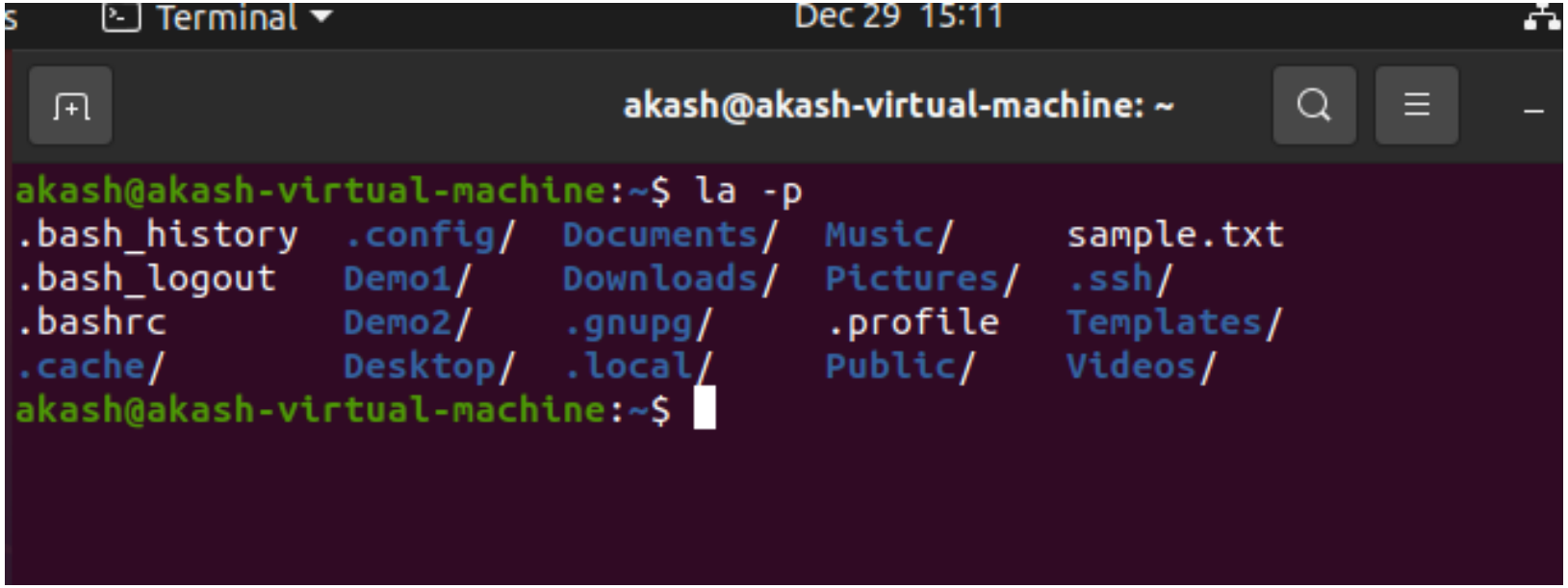
ls –m

$ ls –m Displays the names as a comma-separated list.



ls –p

$ ls –p Displays directories with /



Df command

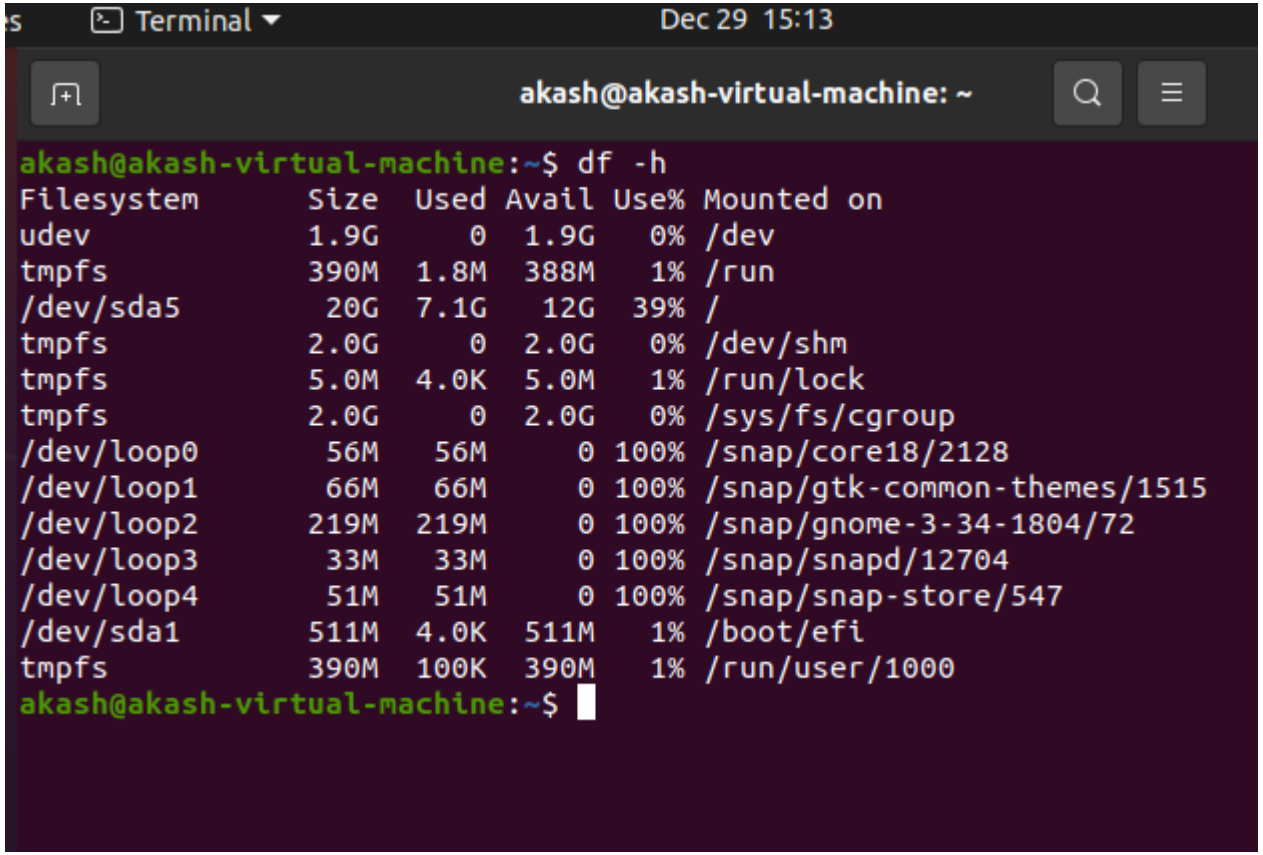
- The “df” command in Linux stands for “disk free.”
- It is used for checking the space available across the different file systems of your Linux system.

- The basic syntax of the “df” command is as follows:
  - **$ df [option]**

Df command

- The disk space can also be displayed in a human-readable format in Linux in the following manner:
  - **$ df –h**

Example


Terminal
Dec 29 15:13
akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ df -h
Filesystem Size Used Avail Use% Mounted on
udev 1.9G 0 1.9G 0% /dev
tmpfs 390M 1.8M 388M 1% /run
/dev/sda5 20G 7.1G 12G 39% /
tmpfs 2.0G 0 2.0G 0% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 2.0G 0 2.0G 0% /sys/fs/cgroup
/dev/loop0 56M 56M 0 100% /snap/core18/2128
/dev/loop1 66M 66M 0 100% /snap/gtk-common-themes/1515
/dev/loop2 219M 219M 0 100% /snap/gnome-3-34-1804/72
/dev/loop3 33M 33M 0 100% /snap/snapd/12704
/dev/loop4 51M 51M 0 100% /snap/snap-store/547
/dev/sda1 511M 4.0K 511M 1% /boot/efi
tmpfs 390M 100K 390M 1% /run/user/1000
akash@akash-virtual-machine:~$

zip command

- zip is a command-line utility that helps you create Zip archives.

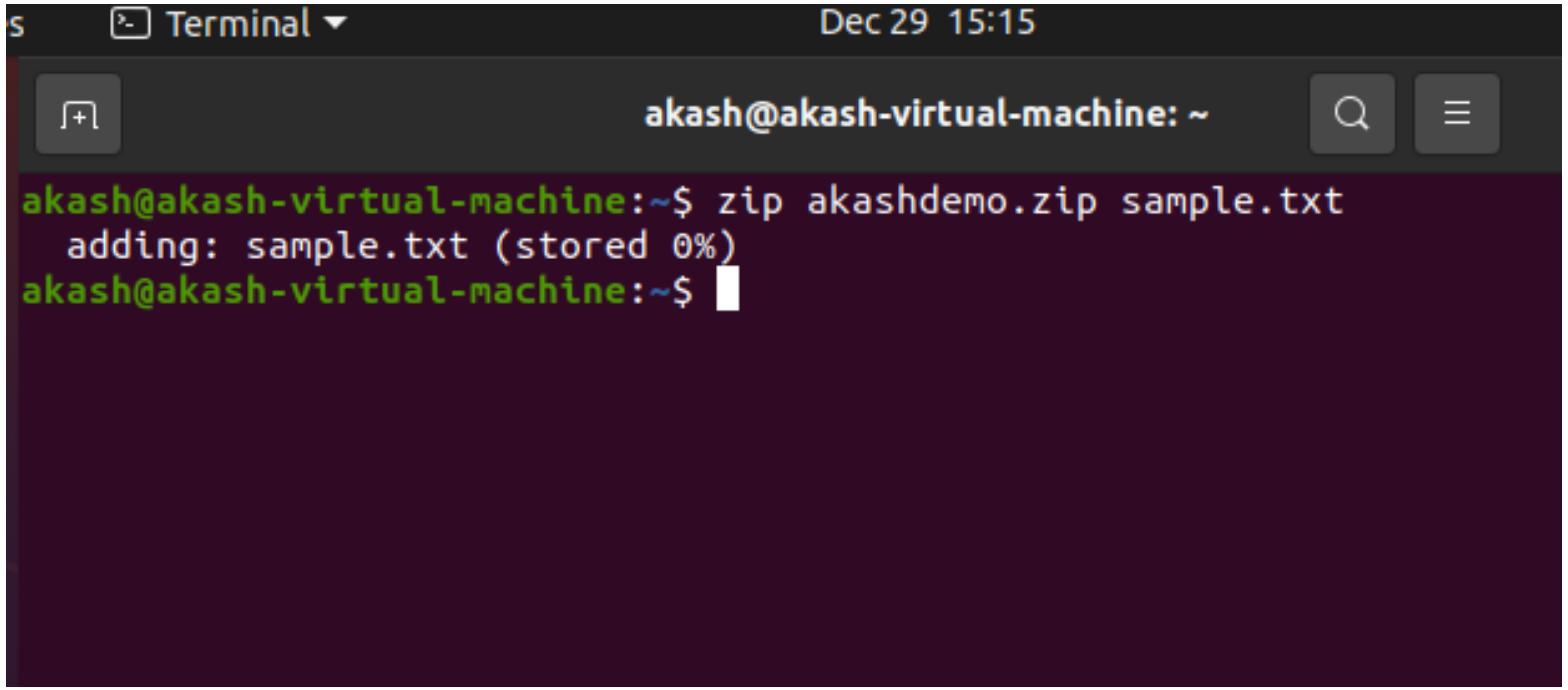
- The zip command takes the following syntax form:

zip ARCHIVE_NAME FILES

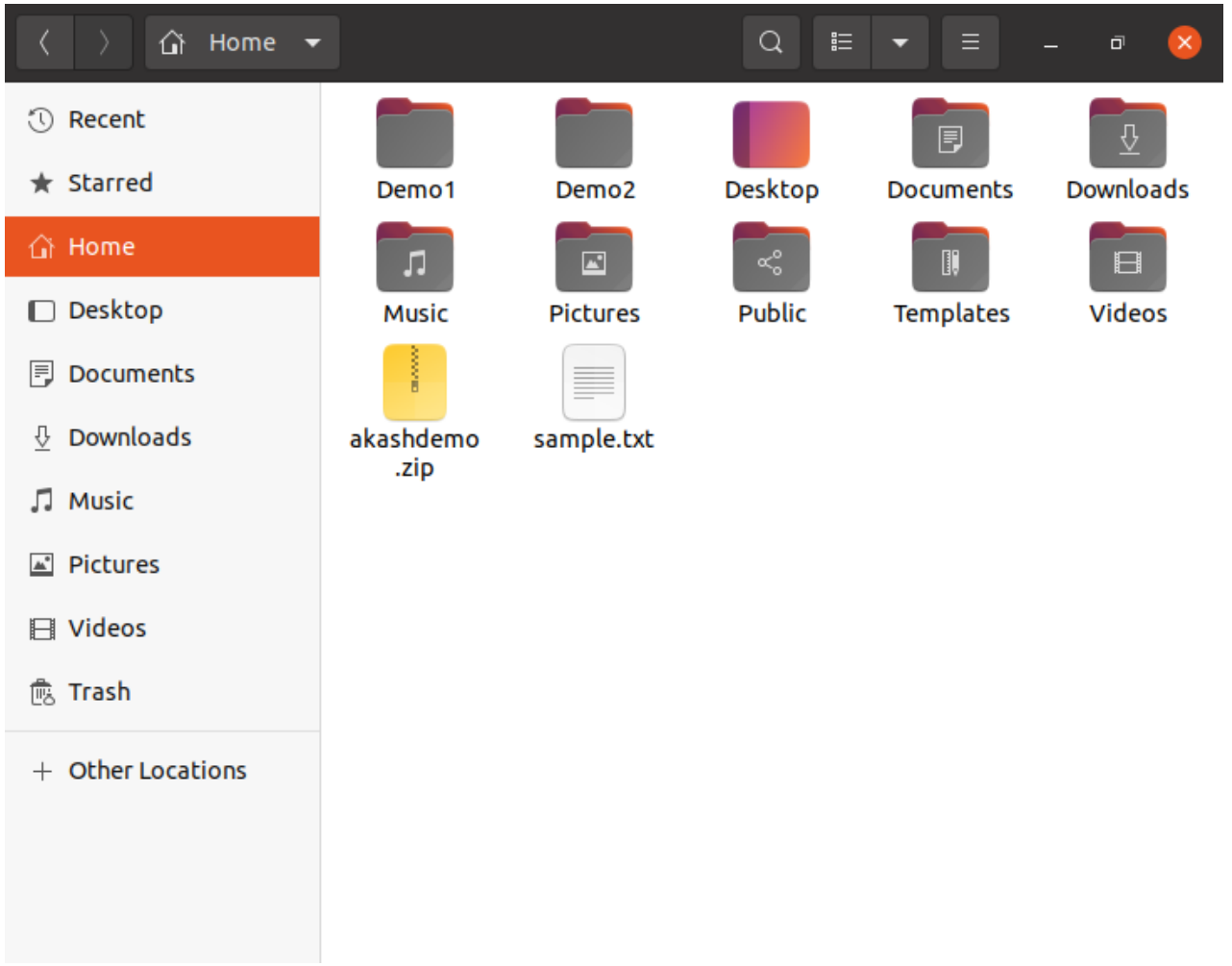
- **Example:**
- **$zip myfile.zip filename.txt**

Zip Example

$zip myfile.zip filename.txt



File Structure



Vi Command

- The default editor that comes with the UNIX operating system is called vi (visual editor).
- Using vi editor, we can edit an existing file or create a new file from scratch. we can also use this editor to just read a text file.

- **Syntax:**
  - **vi filename**

vi Command mode:

- The vi editor opens in this mode, and it only understands commands.
- In this mode, you can, move the cursor and cut, copy, paste the text.
- This mode also saves the changes you have made to the file.
- Commands are case sensitive. You should use the right letter case.

vi Editor Insert mode:

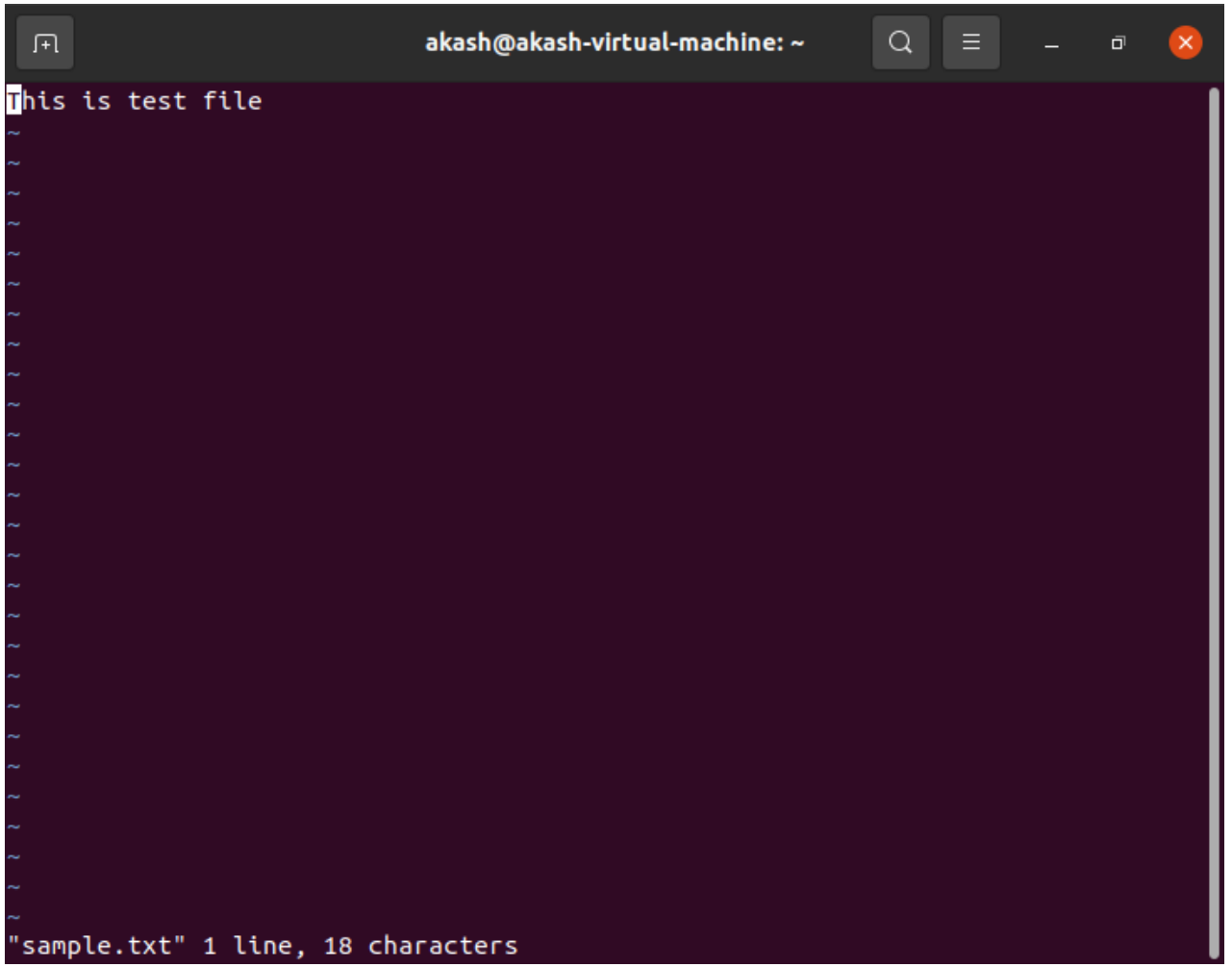
- This mode is for inserting text in the file.
- You can switch to the Insert mode from the command mode by pressing ‘i’ on the keyboard.
- Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
- To return to the command mode and save the changes you have made you need to press the Esc key.

Reading File in Vi Editor

```
[1]+  Stopped                 vi
akash@akash-virtual-machine:~$ vi sample.txt

[2]+  Stopped                 vi sample.txt
akash@akash-virtual-machine:~$
```

Output


akash@akash-virtual-machine: ~
This is test file
"sample.txt" 1 line, 18 characters

VI Editing commands

- i – Insert at cursor (goes into insert mode)
- a – Write after cursor (goes into insert mode)
- A – Write at the end of line (goes into insert mode)
- ESC – Terminate insert mode
- u – Undo last change
- U – Undo all changes to the entire line
- o – Open a new line (goes into insert mode)
- dd – Delete line
- 3dd – Delete 3 lines.
- D – Delete contents of line after the cursor
- dw – Delete word
- C – Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
- 4dw – Delete 4 words
- cw – Change word
- x – Delete character at the cursor
- r – Replace character
- R – Overwrite characters from cursor onward
- s – Substitute one character under cursor continue to insert
- S – Substitute entire line and begin to insert at the beginning of the line
- ~ – Change case of individual character

VI Note

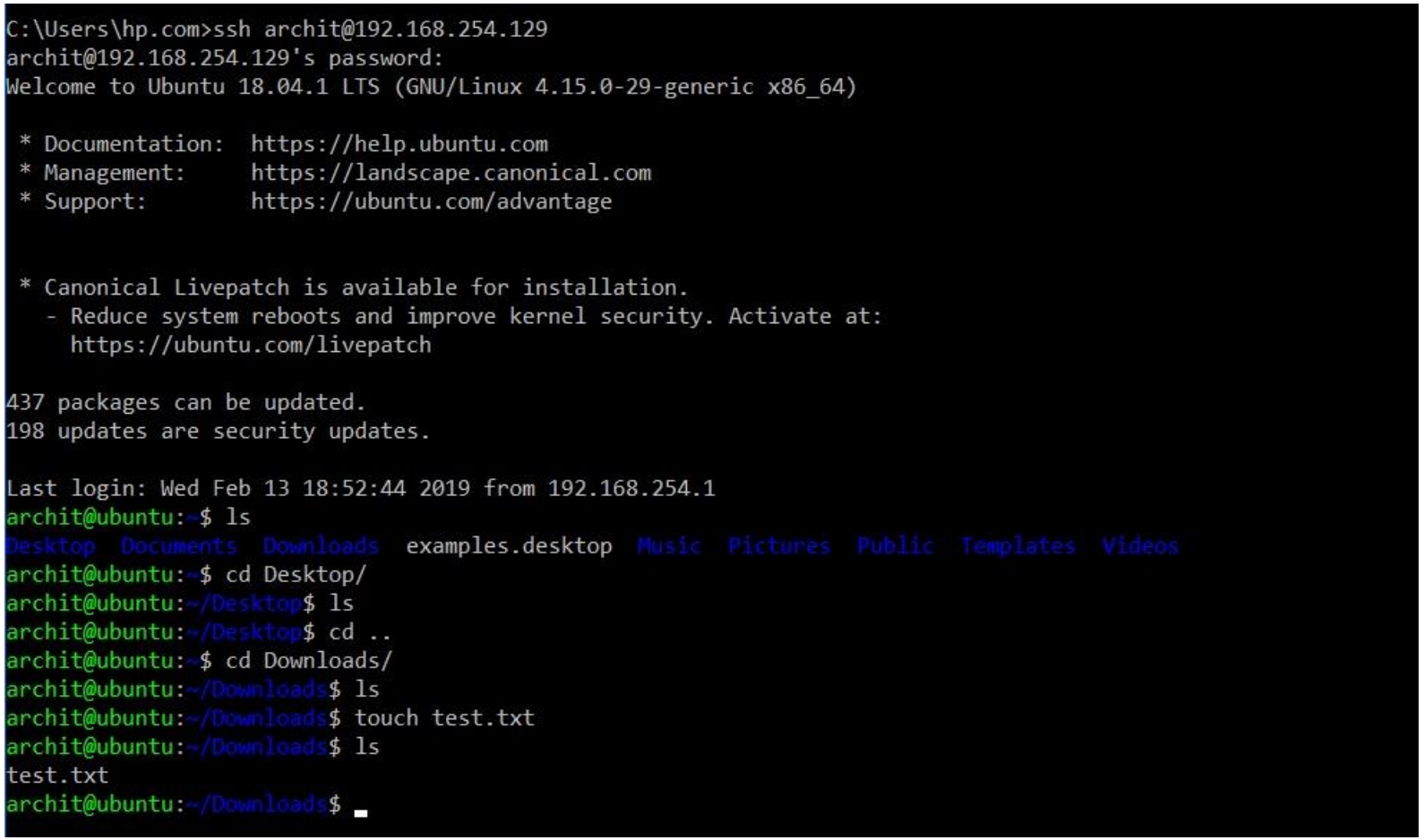
- Note: You should be in the “command mode” to execute these commands. VI editor is case-sensitive so make sure you type the commands in the right letter-case.

ssh command

- ssh stands for “Secure Shell”. It is a protocol used to securely connect to a remote server/system.
- ssh is secure in the sense that it transfers the data in encrypted form between the host and the client.
- It transfers inputs from the client to the host and relays back the output.
- ssh runs at TCP/IP port 22.
- **Syntax:**
  - **ssh user_name@host(IP/Domain_name)**

Example

- Example: Accessing ubuntu machine via windows10 command prompt using ssh



ssh

- ssh command consists of 3 different parts:
  - **ssh command** instructs the system to establish an encrypted secure connection with the host machine.
  - **user_name** represents the account that is being accessed on the host.
  - **host** refers to the machine which can be a computer or a router that is being accessed. It can be an IP address (e.g. 192.168.1.24) or domain name(e.g. www.domainname.com).

Chown command

- chown command is used to change the file Owner or group. Whenever you want to change ownership you can use chown command.
- **Ownership and Permissions:** To protect and secure files and directory in Linux we use permissions to control what a user can do with a file or directory**. Linux uses three types of permissions:**

Linux Users

- Read:
  - This permission allows the user to read files and in directories, it lets the user read directories and subdirectories stores in it.

- Write:
  - This permission allows a user to modify and delete a file.
  - Also it allows a user to modify its contents (create, delete and rename files in it) for the directories.
  - Unless the execute permission is not given to directories changes does do affect them.

- Execute:
  - The write permission on a file allows it to get executed.
  - For example, if we have a file named php.sh so unless we don’t give it execute permission it won’t run.

Permission

- Types of file Permissions:
- **User:** These type of file permission affect the owner of the file.
- **Group:** These type of file permission affect the group which owns the file. Instead of the group permissions, the user permissions will apply if the owner user is in this group.
- **Other:** These type of file permission affect all other users on the system.

chown

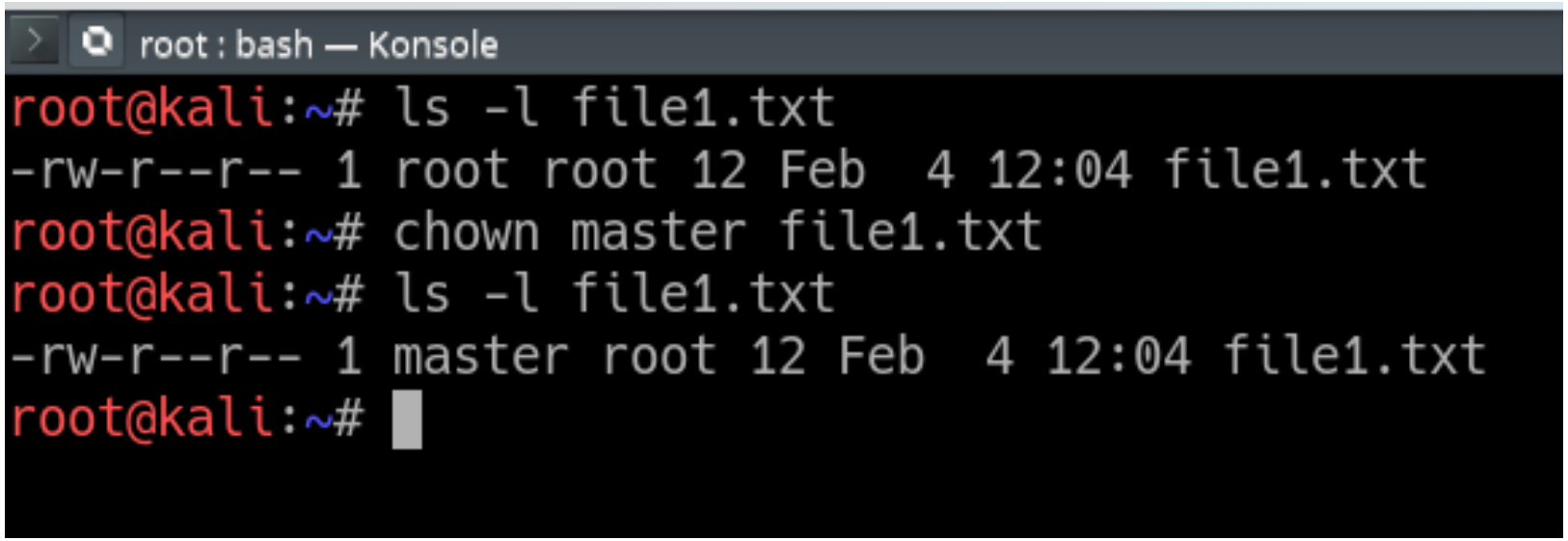
- **Note: To view the permissions we use:**
  - ls –l

- **Syntax:**
  - chown [OPTION]… [OWNER][:[GROUP]] FILE…
  - chown [OPTION]… –reference=RFILE FILE…

Chown

- Example: To change owner of the file:
  - **chown owner_name file_name**

Output


root : bash — Konsole
root@kali:~# ls -l file1.txt
-rw-r--r-- 1 root root 12 Feb 4 12:04 file1.txt
root@kali:~# chown master file1.txt
root@kali:~# ls -l file1.txt
-rw-r--r-- 1 master root 12 Feb 4 12:04 file1.txt
root@kali:~#

Chmod Command

- To change file mode Chmod command is use.
- Options of chmod are

Symbol	Meaning
U	User
G	Group
O	Other
A	All
R	Read
W	write (and delete)
X	execute (and access directory)
+	add permission
-	take away permission

chmod

- Let's say you are the owner of a file named myfile, and you want to set its permissions so that:
- the **user can read, write, and execute** it;
- members of your **group can read and execute** it; and
- **others** may **only read** it.

- This command does the trick:
- chmod u=rwx,g=rx,o=r myfile

- This example uses symbolic permissions notation. The letters u, g, and o stand for "user", "group", and "other".
- The equals sign ("=") means "set the permissions exactly like this," and the letters "r", "w", and "x" stand for "read", "write", and "execute", respectively.
- The commas separate the different classes of permissions, and there are no spaces between them.

- chmod 754 myfile

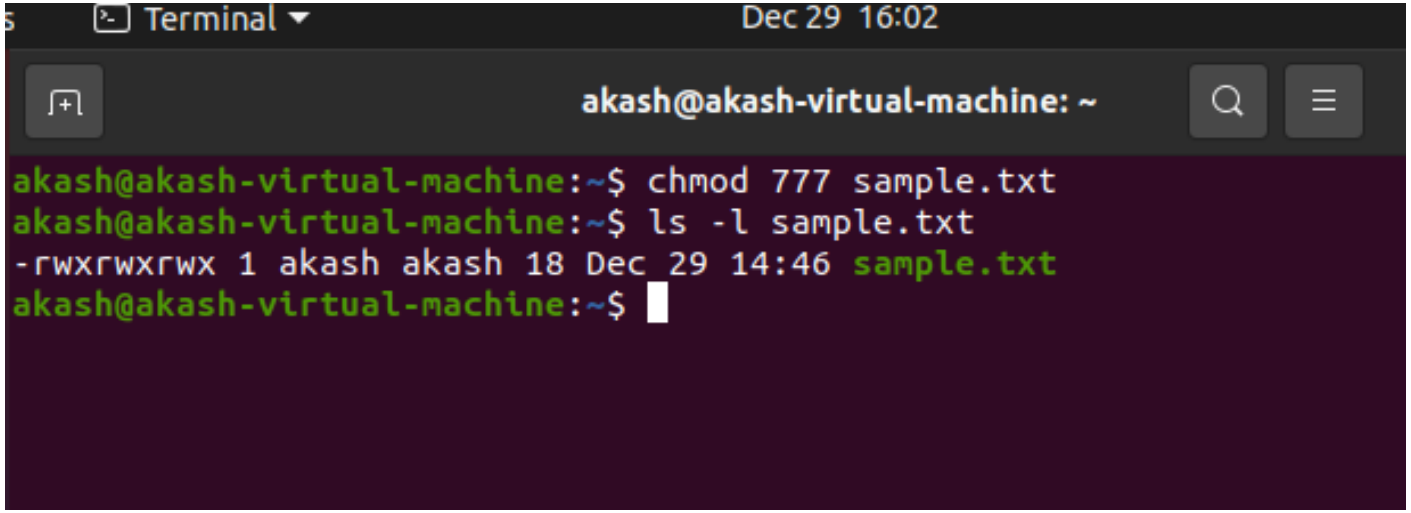
- Here the digits 7, 5, and 4 each individually represent the permissions for the user, group, and others, in that order. Each digit is a combination of the numbers 4, 2, 1, and 0:
  - 4 stands for "read",
  - 2 stands for "write",
  - 1 stands for "execute", and
  - 0 stands for "no permission.“

- So 7 is the combination of permissions 4+2+1 (read, write, and execute), 5 is 4+0+1 (read, no write, and execute), and 4 is 4+0+0 (read, no write, and no execute).

- Every one of the three digits on the mode number corresponds to one of the three permission triplets. (u, g and o)
- Each permission bit in a triplet corresponds to a value: 4 for r, 2 for w, 1 for x.
- So if a file has rwx-xr-x permissions we do
- The following calculation:
  - for u: rwx => 4 + 2 + 1 = 7
  - for g: r-x => 4 + 0 + 1 = 5
  - for o: r-x => 4 + 0 + 1 = 5
  - Which makes : 755

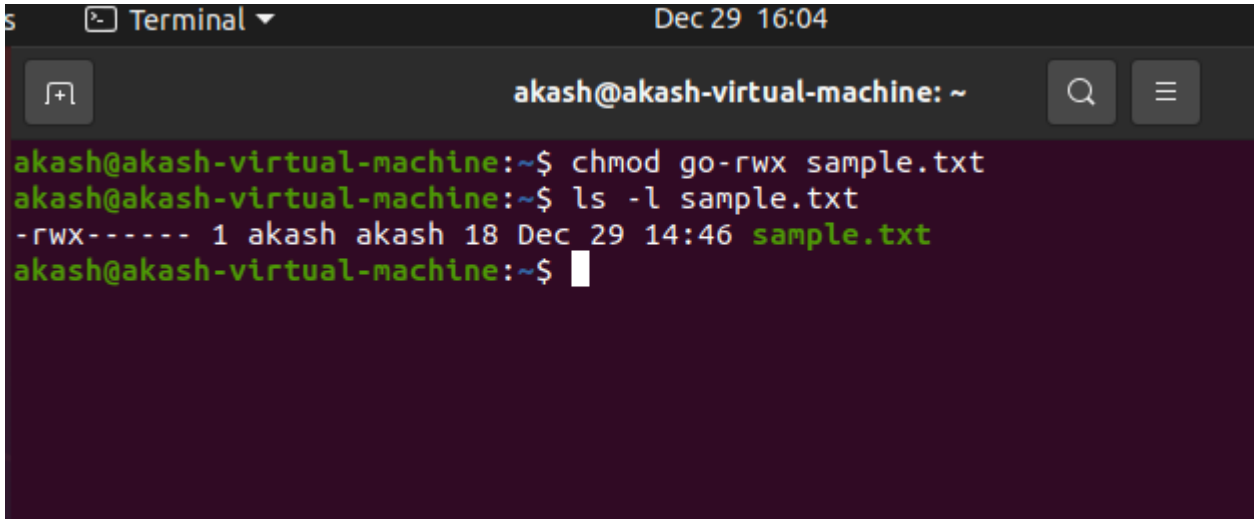
Example 1

- 777 means 'everyone has full access to this file’
  - $ chmod 777 file1(or)$ chmod ugo+rwx file



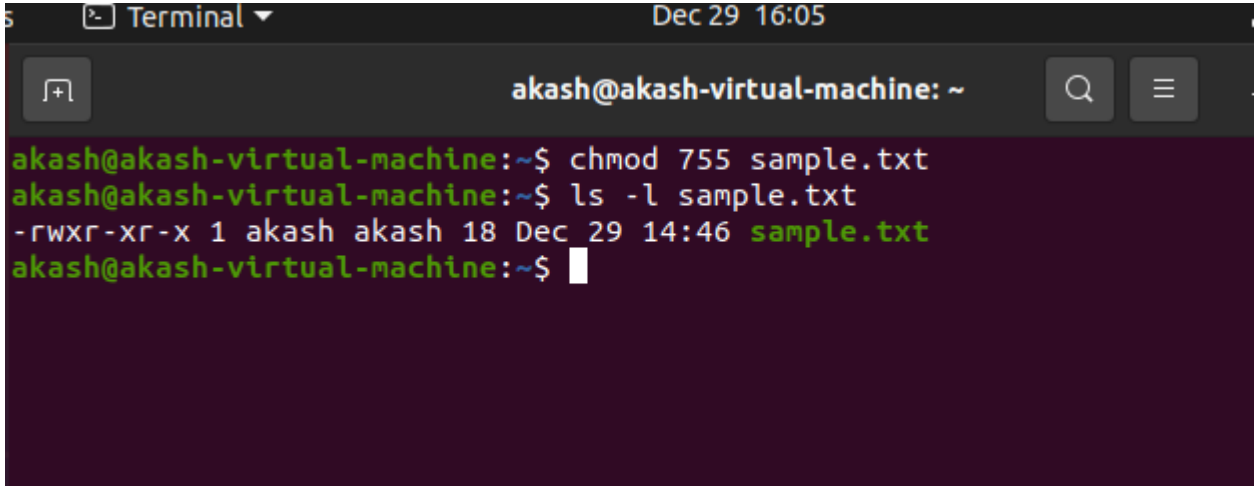
Example 2

- To remove read write and execute permissions on the file file1 for the group and others, type.
  - $ chmod go-rwx file1



Example 3

- Give full permission to user and give read and executable permission to group and others.
  - $chmod 755 file1



Ping command

- Ping is short for Packet Internet Groper.
- This command is mainly used for checking the network connectivity among host/server and host.
- The ping command takes the URL or IP address as input and transfers the data packet to a specified address along with a "PING" message.
- Then, it will get a reply from the host/server. This time is known as "latency".

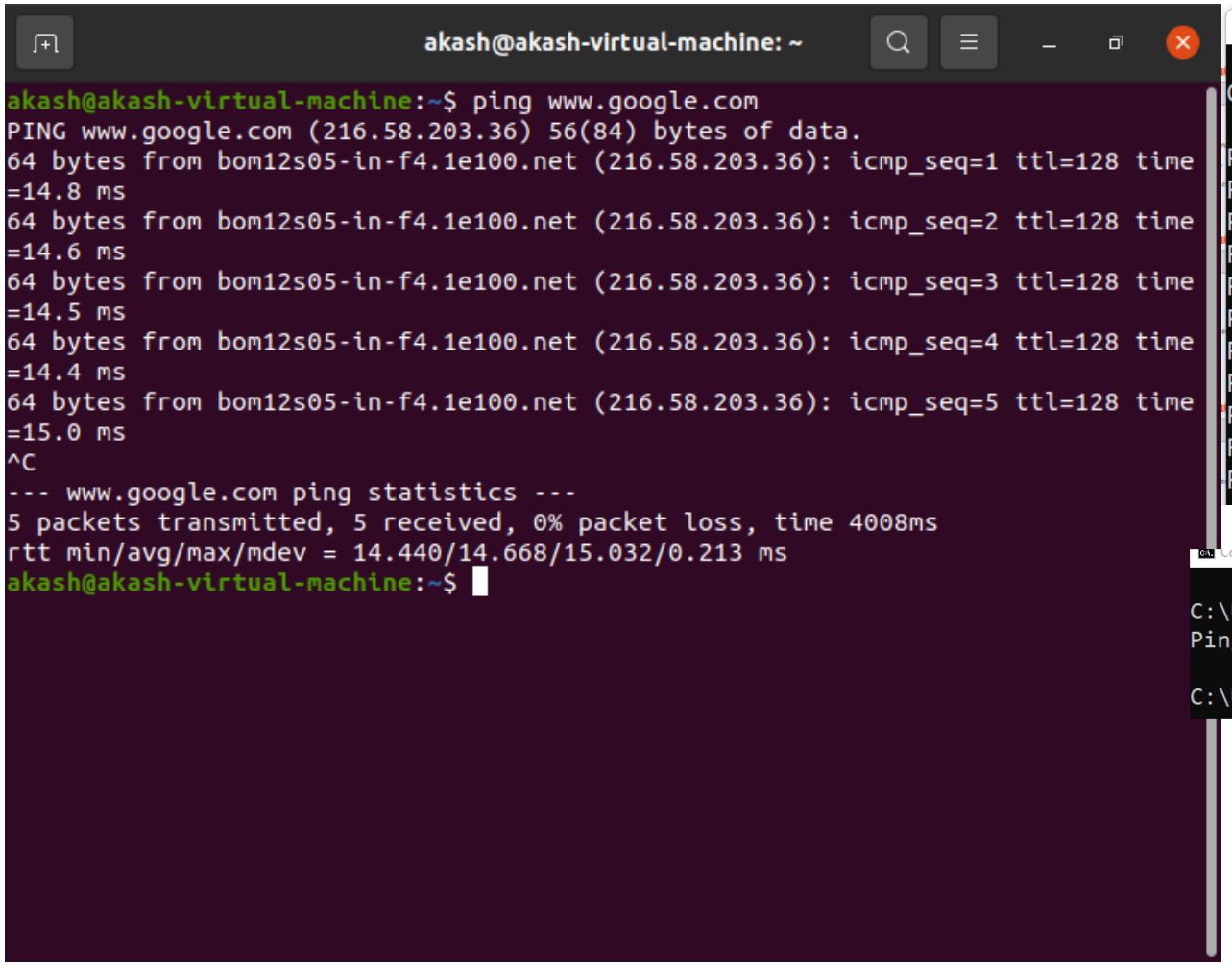
Working of Ping Command

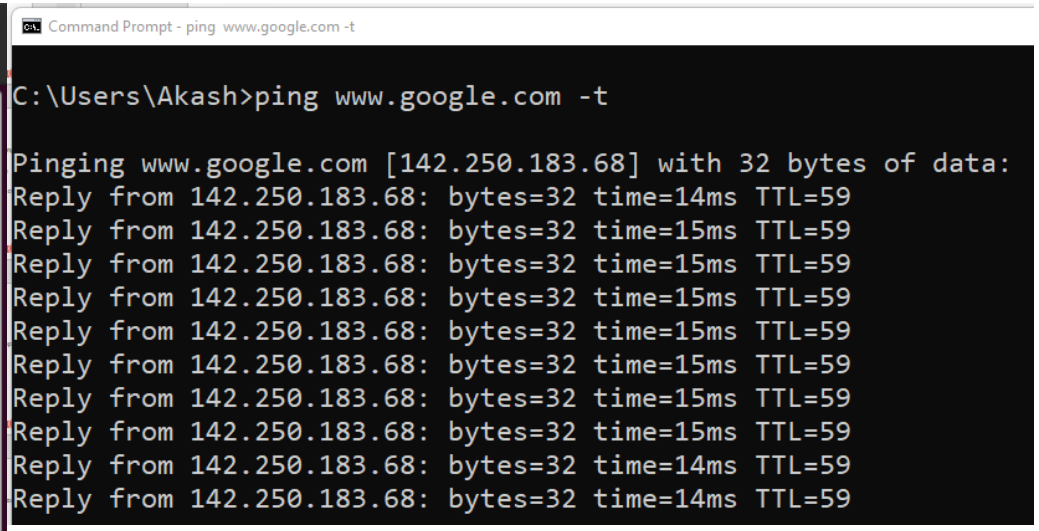
- The ping command permits us to:
  - Test our Internet connection.
  - Check if the remote machine is active.
  - Analyze when there are network problems such as high latency or dropped packages.

- **Syntax :**
  - **ping [options] hostname or IP address**
  - **For stopping the process, we can use the Ctrl+C keys.**

- **from:** It tells the target and its IP address.
- **Important:** The IP address might be different for any website depending on our geographical location.
- **ttl=52:** It tells the value, i.e., Time to Live from 1-255. Also, it indicates network number hops a packet could take before any router removes it.
- **icmp_seq=1:** It tells the all ICMP packet's sequence number. It increases by a single number for all subsequent echo requests.
- **time=7.68 ms:** It tells the Time that it took any packet for reaching the target and come back to the origin. It expressed in ms (milliseconds).

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ping www.google.com
PING www.google.com (216.58.203.36) 56(84) bytes of data.
64 bytes from bom12s05-in-f4.1e100.net (216.58.203.36): icmp_seq=1 ttl=128 time=14.8 ms
64 bytes from bom12s05-in-f4.1e100.net (216.58.203.36): icmp_seq=2 ttl=128 time=14.6 ms
64 bytes from bom12s05-in-f4.1e100.net (216.58.203.36): icmp_seq=3 ttl=128 time=14.5 ms
64 bytes from bom12s05-in-f4.1e100.net (216.58.203.36): icmp_seq=4 ttl=128 time=14.4 ms
64 bytes from bom12s05-in-f4.1e100.net (216.58.203.36): icmp_seq=5 ttl=128 time=15.0 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 14.440/14.668/15.032/0.213 ms
akash@akash-virtual-machine:~$


Command Prompt - ping www.google.com -t
C:\Users\Akash>ping www.google.com -t
Pinging www.google.com [142.250.183.68] with 32 bytes of data:
Reply from 142.250.183.68: bytes=32 time=14ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=15ms TTL=59
Reply from 142.250.183.68: bytes=32 time=14ms TTL=59
Reply from 142.250.183.68: bytes=32 time=14ms TTL=59


Command Prompt
C:\Users\Akash>ping www.g.com -t
Ping request could not find host www.g.com. Please check the name and try again.
C:\Users\Akash>

ifconfig command

- ifconfig(interface configuration) command is used to configure the kernel-resident network interfaces.
- It is used at the boot time to set up the interfaces as necessary.
- After that, it is usually used when needed during debugging or when you need system tuning.
- Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.
- **Syntax :-**
  - **ifconfig [...OPTIONS] [INTERFACE]**

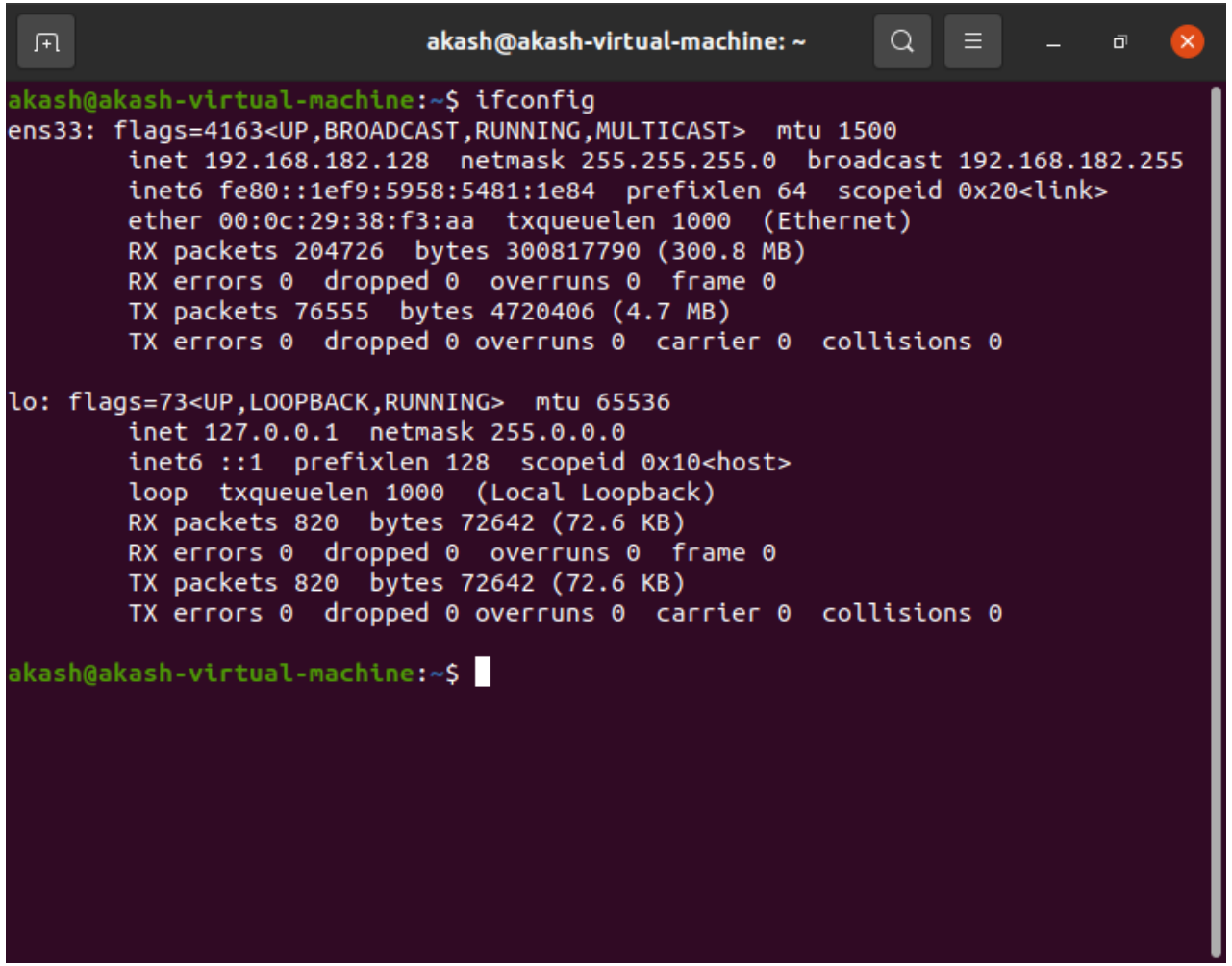
- Newer versions of some Linux distributions don’t have ifconfig command pre-installed.
- So, in case, there is an error “ifconfig: command not found”, Then execute the following command to install ifconfig.

sudo apt install net-tools

- For CentOS or RPM(RedHat Package Manager) based Linux

yum install net-tools

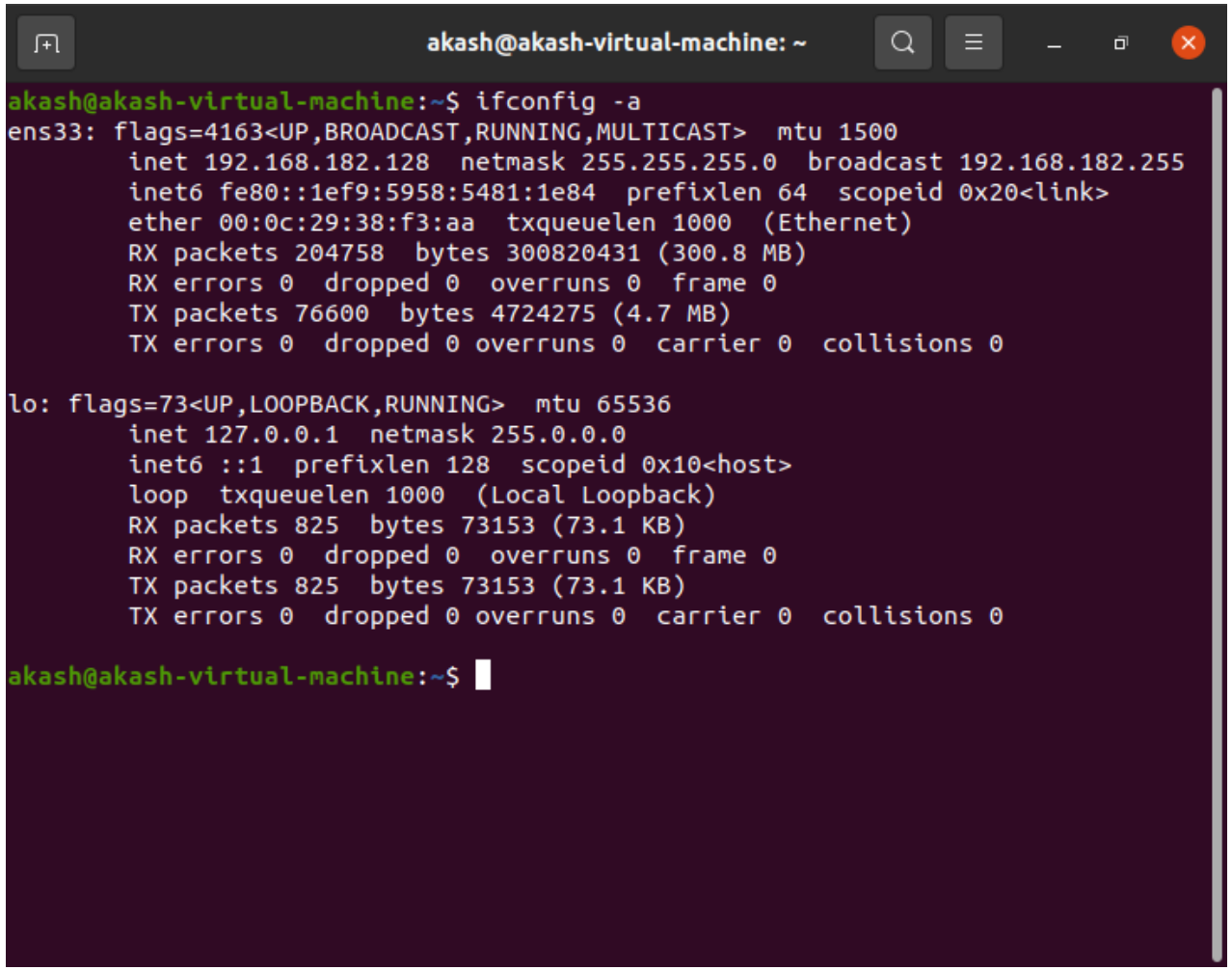
Ifconfig Example



Options

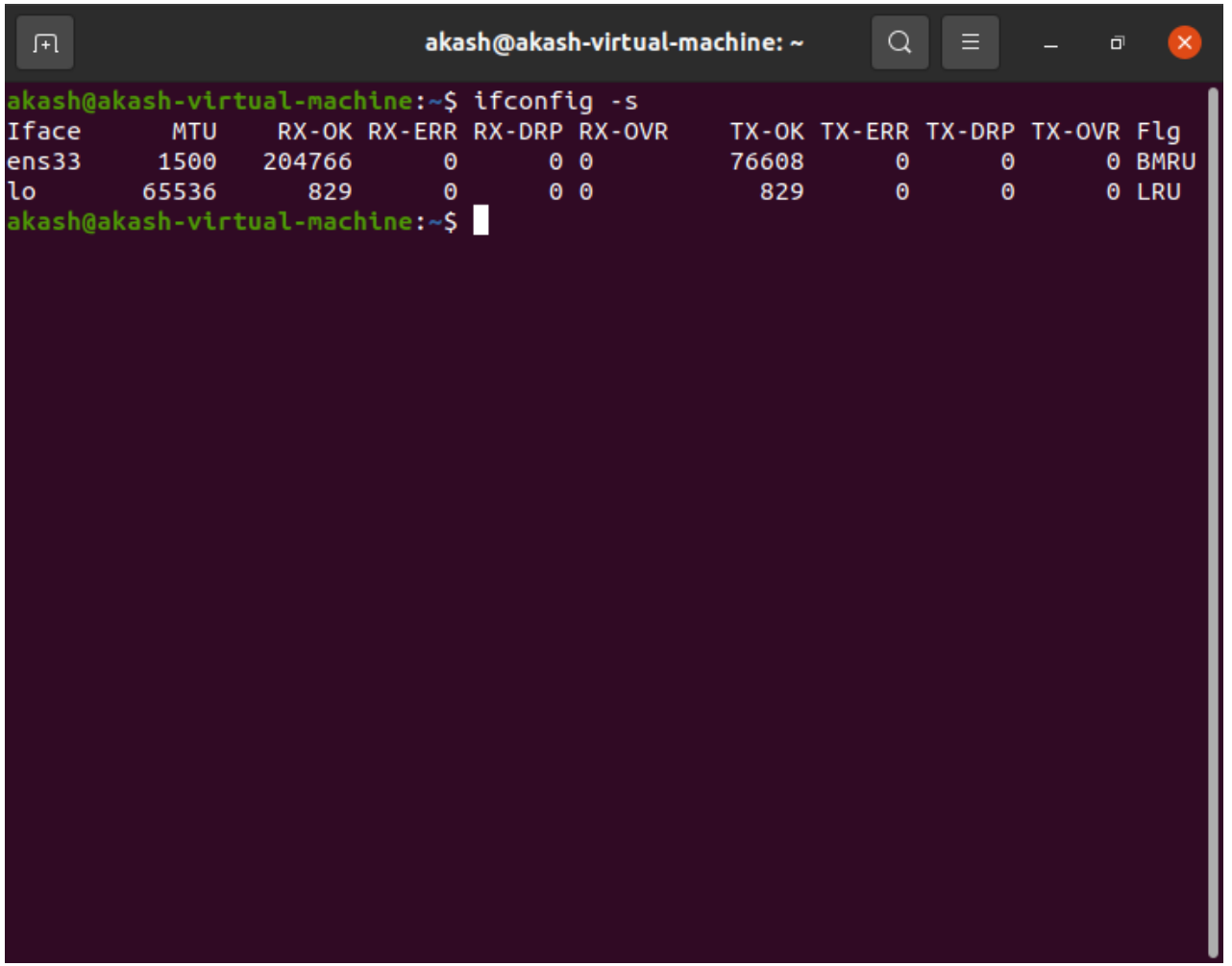
- a : This option is used to display all the interfaces available, even if they are down.
- Syntax:
- ifconfig -a

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.182.128 netmask 255.255.255.0 broadcast 192.168.182.255
inet6 fe80::1ef9:5958:5481:1e84 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:38:f3:aa txqueuelen 1000 (Ethernet)
RX packets 204758 bytes 300820431 (300.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 76600 bytes 4724275 (4.7 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 825 bytes 73153 (73.1 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 825 bytes 73153 (73.1 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
akash@akash-virtual-machine:~$

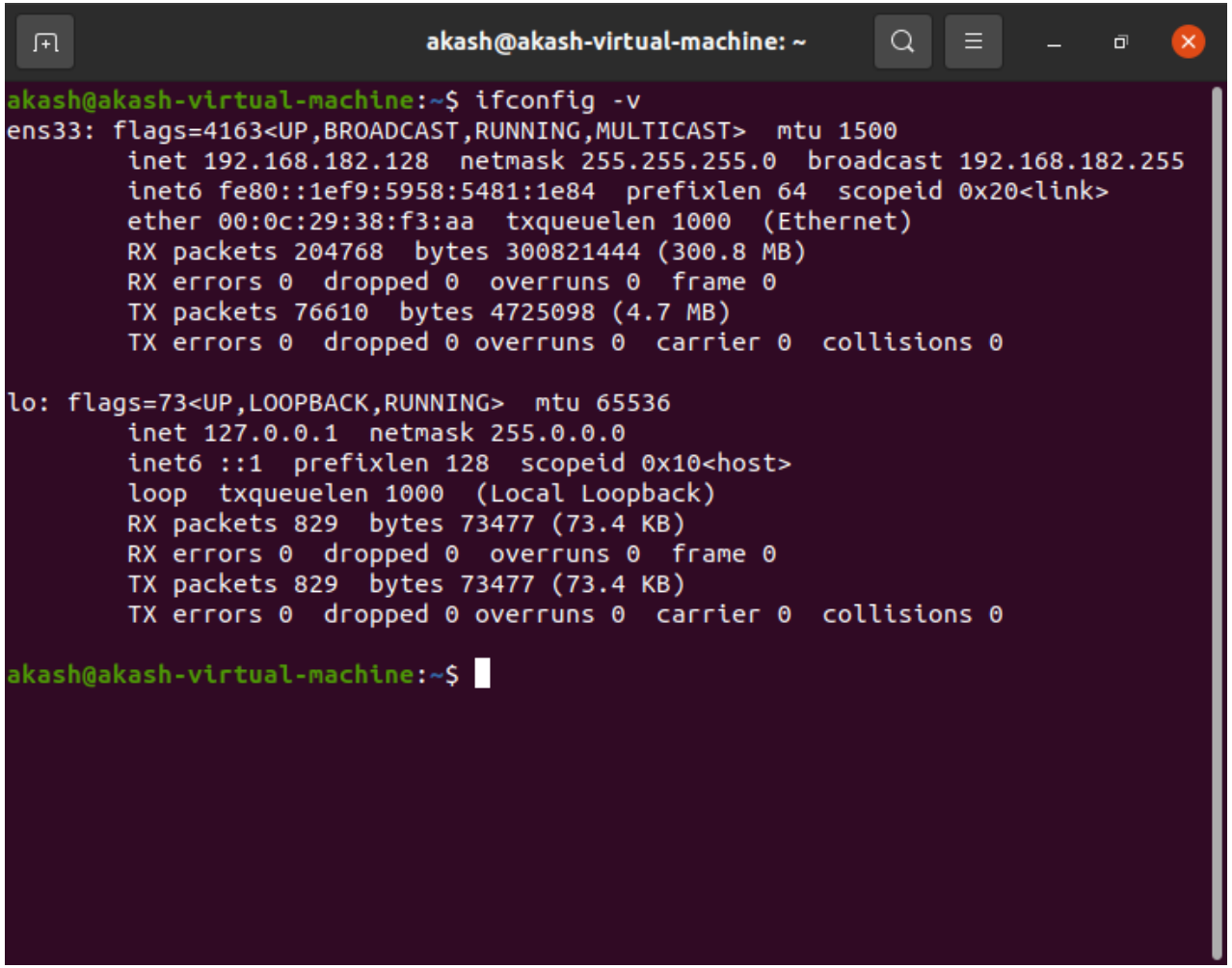
- -s : Display a short list, instead of details.
- Syntax:
- ifconfig -s

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ifconfig -s
Iface MTU RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
ens33 1500 204766 0 0 0 76608 0 0 0 BMRU
lo 65536 829 0 0 0 829 0 0 0 LRU
akash@akash-virtual-machine:~$

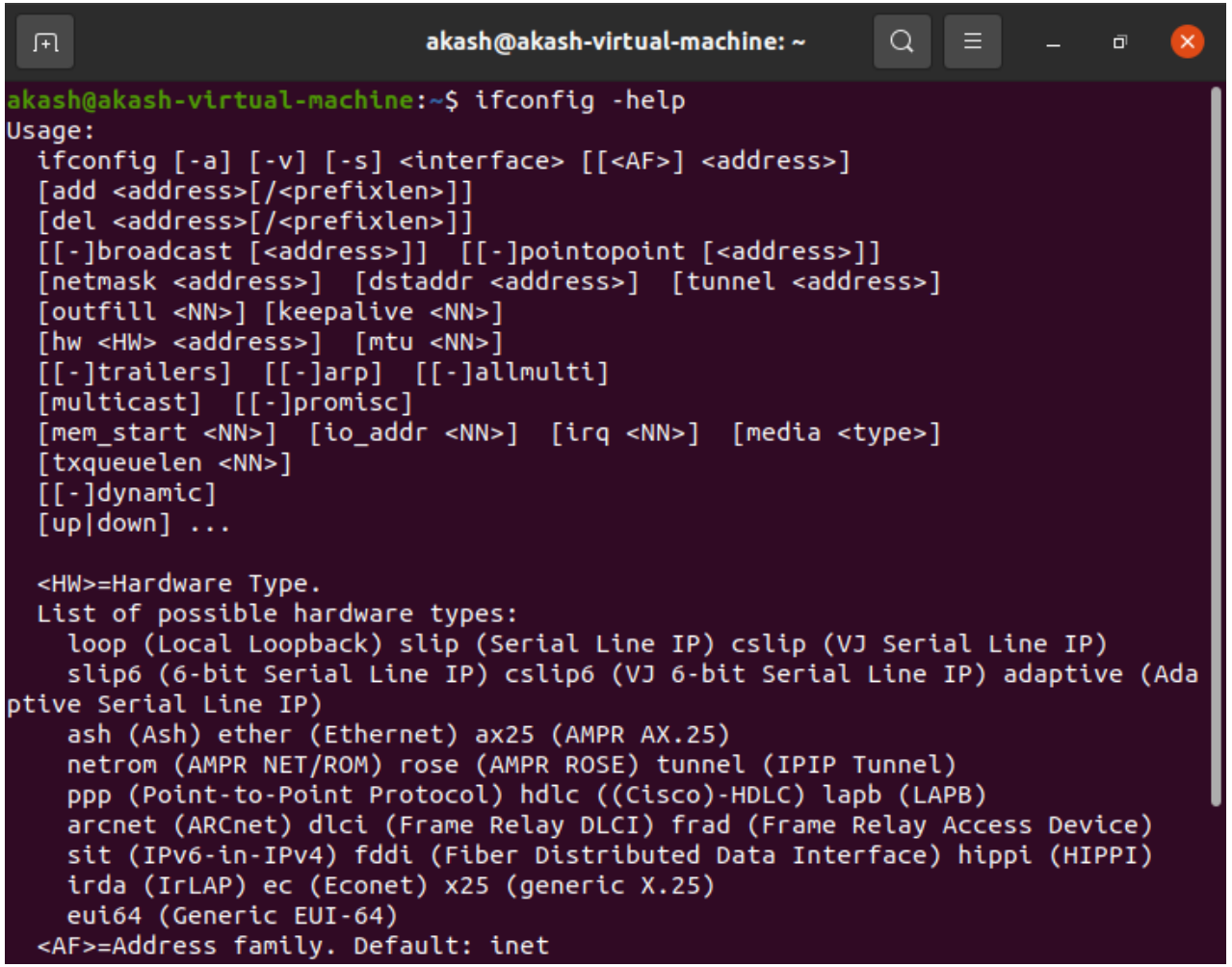
- -v : Run the command in verbose mode – log more details about execution.
- Syntax:
- ifconfig -v

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ifconfig -v
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.182.128 netmask 255.255.255.0 broadcast 192.168.182.255
inet6 fe80::1ef9:5958:5481:1e84 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:38:f3:aa txqueuelen 1000 (Ethernet)
RX packets 204768 bytes 300821444 (300.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 76610 bytes 4725098 (4.7 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 829 bytes 73477 (73.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 829 bytes 73477 (73.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
akash@akash-virtual-machine:~$

- help : Display help related to ifconfig command.
- Syntax:
- ifconfig --help

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ ifconfig -help
Usage:
ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
[add <address>[/<prefixlen>]]
[del <address>[/<prefixlen>]]
[[-]broadcast [<address>]] [[-]pointopoint [<address>]]
[netmask <address>] [dstaddr <address>] [tunnel <address>]
[outfill <NN>] [keepalive <NN>]
[hw <HW> <address>] [mtu <NN>]
[[-]trailers] [[-]arp] [[-]allmulti]
[multicast] [[-]promisc]
[mem_start <NN>] [io_addr <NN>] [irq <NN>] [media <type>]
[txqueuelen <NN>]
[[-]dynamic]
[up|down] ...
<HW>=Hardware Type.
List of possible hardware types:
loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Ada
ptive Serial Line IP)
ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
irda (IrLAP) ec (Econet) x25 (generic X.25)
eui64 (Generic EUI-64)
<AF>=Address family. Default: inet

Up and Down Network

- ifconfig eth0 up
- ifconfig eth0 down
- note : eth0 -> Network name

Wget

- Wget is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

- wget [option] [URL]

Example

- To simply download a webpage:
  - wget http://example.com/sample.php
- To download the file in background
  - wget -b http://www.example.com/samplepage.php
- To resume a partially downloaded file
  - wget -c http://example.com/samplefile.tar.gz
- To try a given number of times
  - wget --tries=10 http://example.com/samplefile.tar.gz

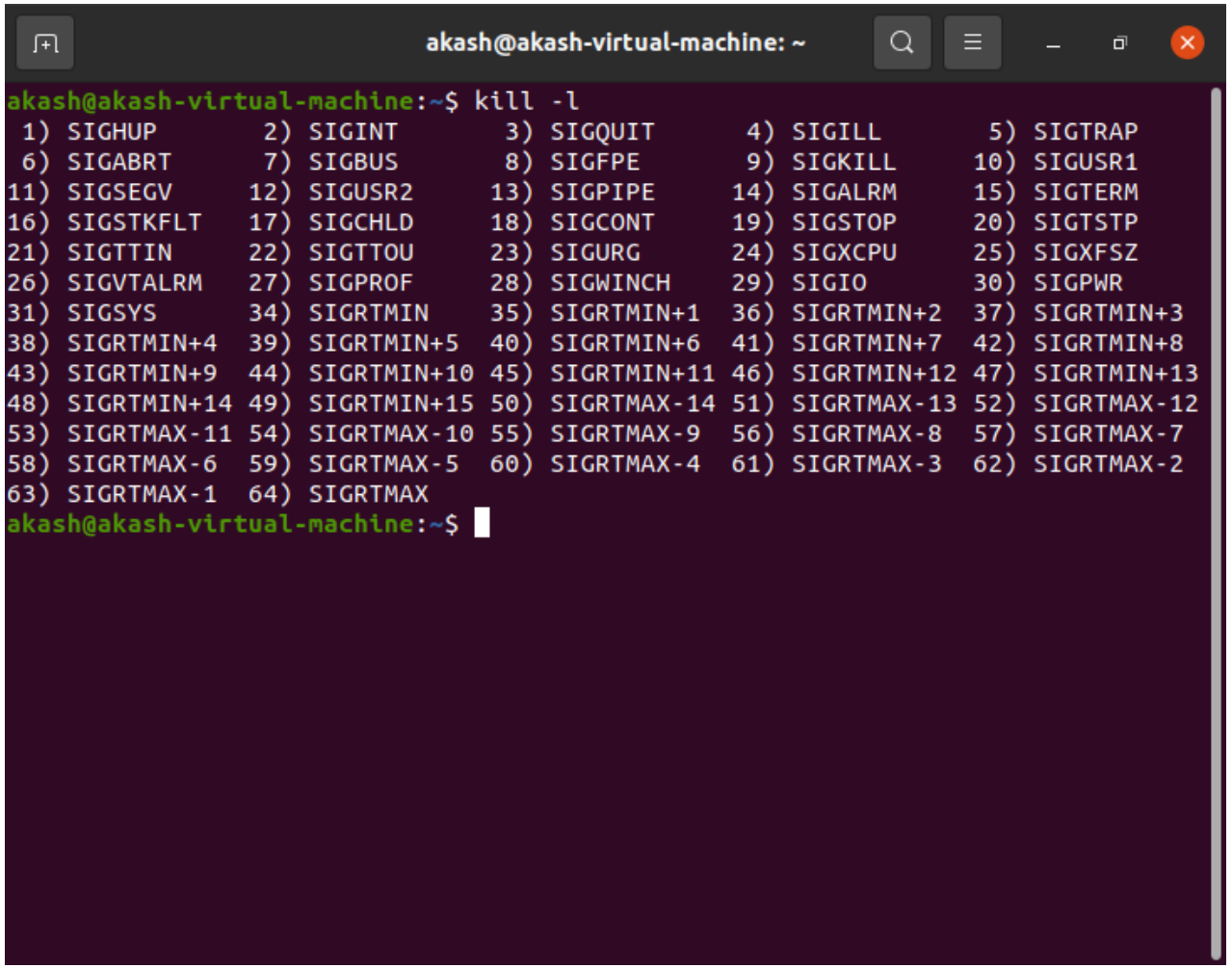
Kill command

- kill command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually.
- kill command sends a signal to a process which terminates the process.
- If the user doesn’t specify any signal which is to be sent along with kill command then default TERM signal is sent that terminates the process.

Option

- kill -l :To display all the available signals you can use below command option:
- Syntax:
- $kill -l

Example


akash@akash-virtual-machine: ~
akash@akash-virtual-machine:~$ kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT   4) SIGILL   5) SIGTRAP
 6) SIGABRT   7) SIGBUS   8) SIGFPE   9) SIGKILL   10) SIGUSR1
11) SIGSEGV   12) SIGUSR2   13) SIGPIPE   14) SIGALRM   15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD   18) SIGCONT   19) SIGSTOP   20) SIGTSTP
21) SIGTTIN   22) SIGTTOU   23) SIGURG   24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF   28) SIGWINCH   29) SIGIO   30) SIGPWR
31) SIGSYS   34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
akash@akash-virtual-machine:~$

lsof command

- The lsof command in Linux displays in its output information about files that are opened by processes.
- Syntax :
- $lsof

Install Linux lsof command

- Most Linux distributions come with lsof pre-installed. If it is not installed on your system yet, use the following commands:

- **On Ubuntu and Debian:**
- **$ sudo apt install lsof**

- **list files based on their Internet address**
- The tool lets you list files based on their Internet address. This can be done using the -i command-line option.
- For example, if you want, you can have IPv4 and IPv6 files displayed separately. For IPv4, run the following command:
- **lsof -i 4**
- **lsof -i 6**

Example remain

Htop Command

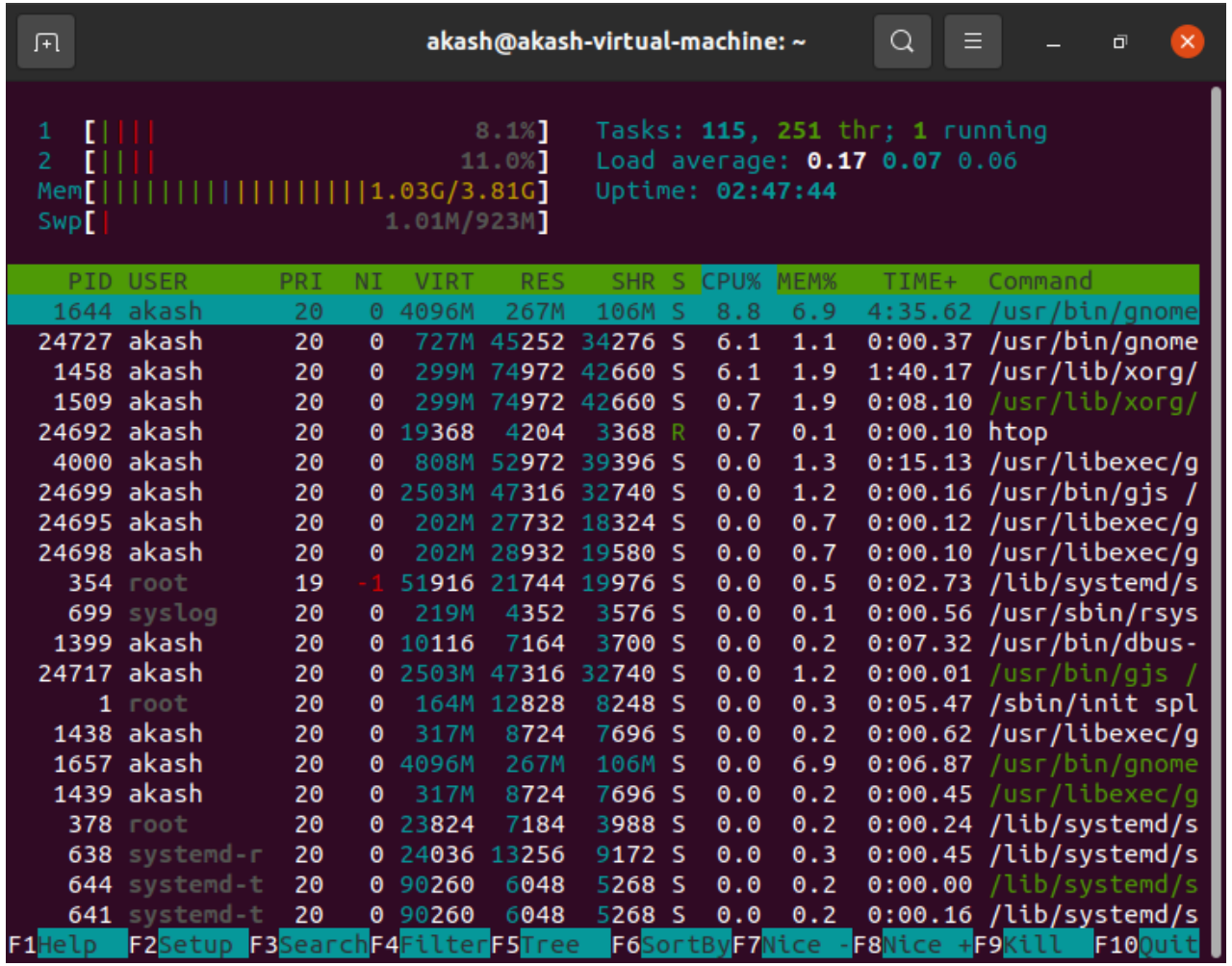
- htop is a ncurses based program for viewing processes in a system running Linux. htop is quite similar to the top command.
- However, since htop is a newer program compared to top, it offers many improvements.
- htop supports mouse operation, uses color in its output and gives visual indications about processor, memory and swap usage.
- htop also prints full command lines for processes and allows one to scroll both vertically and horizontally for processes and command lines respectively.

Install Linux htop command

- **sudo apt-get install htop**

- **RUNNING htop**
- htop can be run from the command line,
- $ htop

Example


akash@akash-virtual-machine: ~
1 [|||| 8.1%]
2 [|||| 11.0%]
Mem[|||||||||||||||||1.03G/3.81G]
Swp[| 1.01M/923M]
Tasks: 115, 251 thr; 1 running
Load average: 0.17 0.07 0.06
Uptime: 02:47:44
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1644 akash 20 0 4096M 267M 106M S 8.8 6.9 4:35.62 /usr/bin/gnome
24727 akash 20 0 727M 45252 34276 S 6.1 1.1 0:00.37 /usr/bin/gnome
1458 akash 20 0 299M 74972 42660 S 6.1 1.9 1:40.17 /usr/lib/xorg/
1509 akash 20 0 299M 74972 42660 S 0.7 1.9 0:08.10 /usr/lib/xorg/
24692 akash 20 0 19368 4204 3368 R 0.7 0.1 0:00.10 htop
4000 akash 20 0 808M 52972 39396 S 0.0 1.3 0:15.13 /usr/libexec/g
24699 akash 20 0 2503M 47316 32740 S 0.0 1.2 0:00.16 /usr/bin/gjs /
24695 akash 20 0 202M 27732 18324 S 0.0 0.7 0:00.12 /usr/libexec/g
24698 akash 20 0 202M 28932 19580 S 0.0 0.7 0:00.10 /usr/libexec/g
354 root 19 -1 51916 21744 19976 S 0.0 0.5 0:02.73 /lib/systemd/s
699 syslog 20 0 219M 4352 3576 S 0.0 0.1 0:00.56 /usr/sbin/rsys
1399 akash 20 0 10116 7164 3700 S 0.0 0.2 0:07.32 /usr/bin/dbus-
24717 akash 20 0 2503M 47316 32740 S 0.0 1.2 0:00.01 /usr/bin/gjs /
1 root 20 0 164M 12828 8248 S 0.0 0.3 0:05.47 /sbin/init spl
1438 akash 20 0 317M 8724 7696 S 0.0 0.2 0:00.62 /usr/libexec/g
1657 akash 20 0 4096M 267M 106M S 0.0 6.9 0:06.87 /usr/bin/gnome
1439 akash 20 0 317M 8724 7696 S 0.0 0.2 0:00.45 /usr/libexec/g
378 root 20 0 23824 7184 3988 S 0.0 0.2 0:00.24 /lib/systemd/s
638 systemd-r 20 0 24036 13256 9172 S 0.0 0.3 0:00.45 /lib/systemd/s
644 systemd-t 20 0 90260 6048 5268 S 0.0 0.2 0:00.00 /lib/systemd/s
641 systemd-t 20 0 90260 6048 5268 S 0.0 0.2 0:00.16 /lib/systemd/s
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit

Color coding for CPU

- As it is quite visible to the user, there are a multiple colours used to describe the bar. Each colour has a specific meaning:

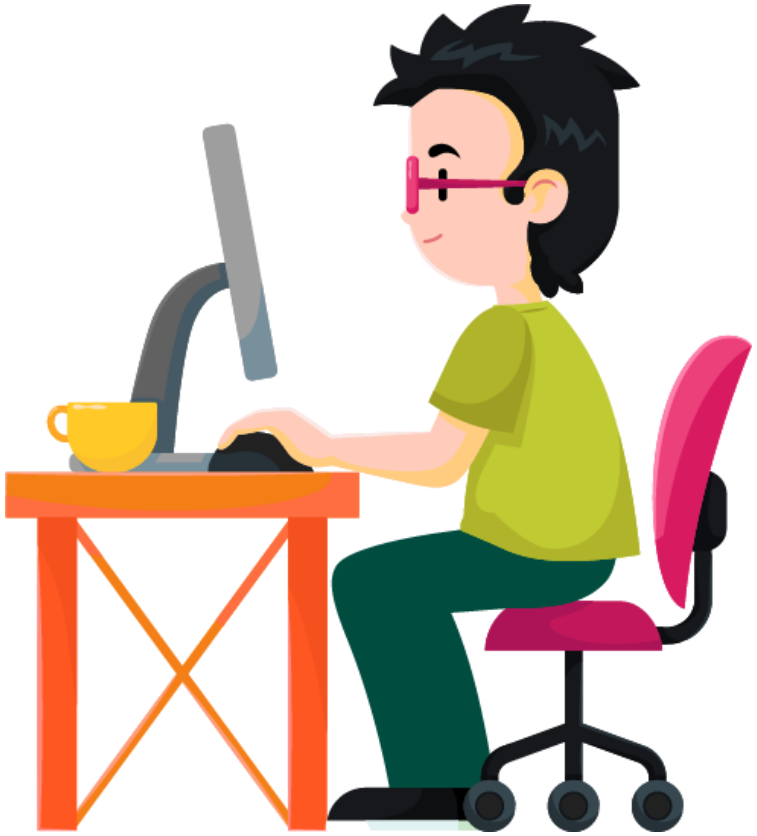
  - **Green – Amount of CPU consumed by the user's processes.**
  - **Red – Amount of CPU used by system processes.**
  - **Grey – Amount of CPU used for Input/Output based processes.**

Color coding for Memory

- The panel also contains information related to the amount of memory being used at every instant.
- The ‘mem’ bar represents the main memory (or RAM), whereas ‘swp’ refers to the swap memory.

- Color coding for Memory
  - **Green – Percentage of RAM being used for running processes in the system.**
  - **Blue – Percentage of RAM being consumed by buffer pages.**
  - **Orange – Percentage of RAM being used for cache memory**.

Get Exclusive Video Tutorials



www.aptutorials.com
https://www.youtube.com/user/Akashtips

Get More Details

www.akashsir.com

If You Liked It !

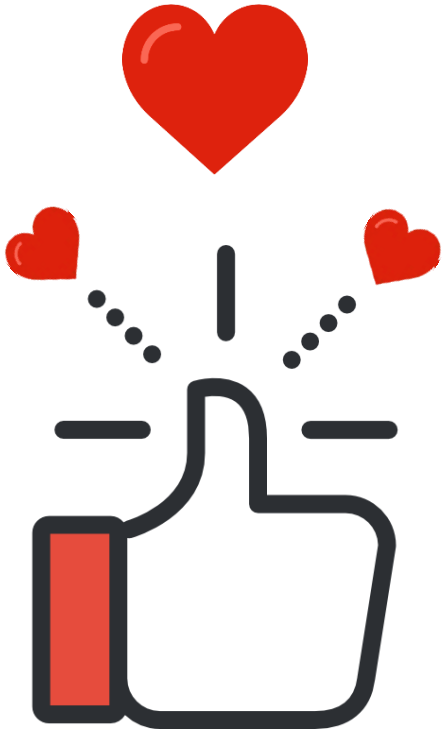
Rating Us Now

Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET

Sulekha

https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad



Connect With Me

Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechnolabs.com
www.akashpadhiyar.com
www.aptutorials.com

Social Info

Akash.padhiyar

Akashpadhiyar

Akash_padhiyar

+91 99786-21654

#Akashpadhiyar
#aptutorials