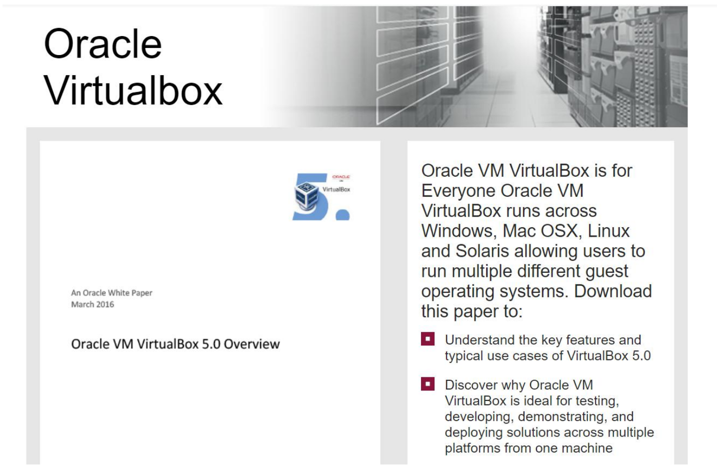
Authors -
Pranjal Naik
Shrey Kothari
Dhaval Patel

Question 1:

Q 1-a Read Oracle VirtualBox White Paper



Oracle Virtualbox

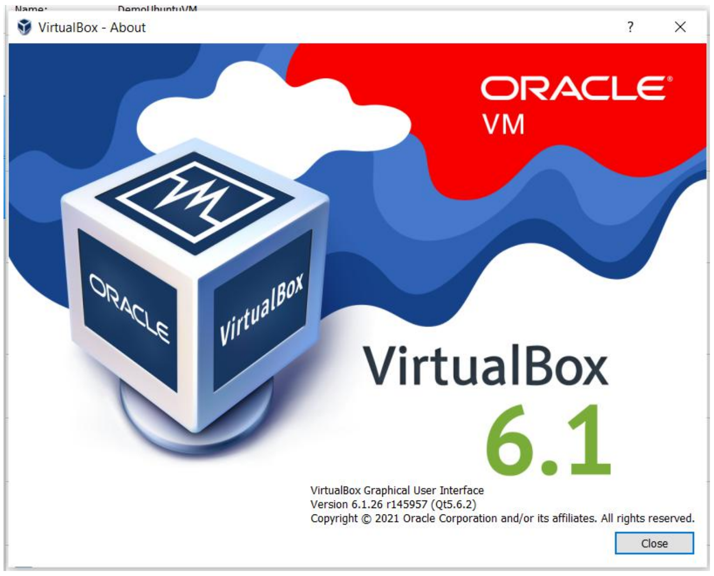
An Oracle White Paper
March 2016

Oracle VM VirtualBox 5.0 Overview

Oracle VM VirtualBox is for Everyone Oracle VM VirtualBox runs across Windows, Mac OSX, Linux and Solaris allowing users to run multiple different guest operating systems. Download this paper to:

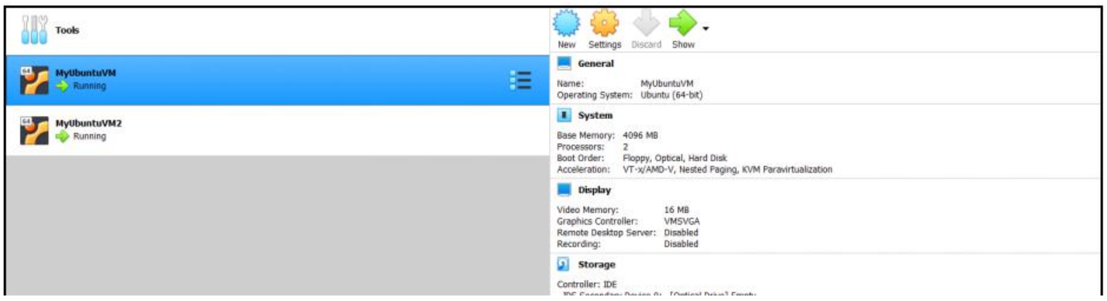
- Understand the key features and typical use cases of VirtualBox 5.0
- Discover why Oracle VM VirtualBox is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms from one machine

Q 1-b Download Oracle VirtualBox 6.1.26



VirtualBox Graphical User Interface
Version 6.1.26 r145957 (Qt5.6.2)
Copyright © 2021 Oracle Corporation and/or its affiliates. All rights reserved.

Q 1-c Install VirtualBox



General

Name: MyUbuntuVM
Operating System: Ubuntu (64-bit)

System

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

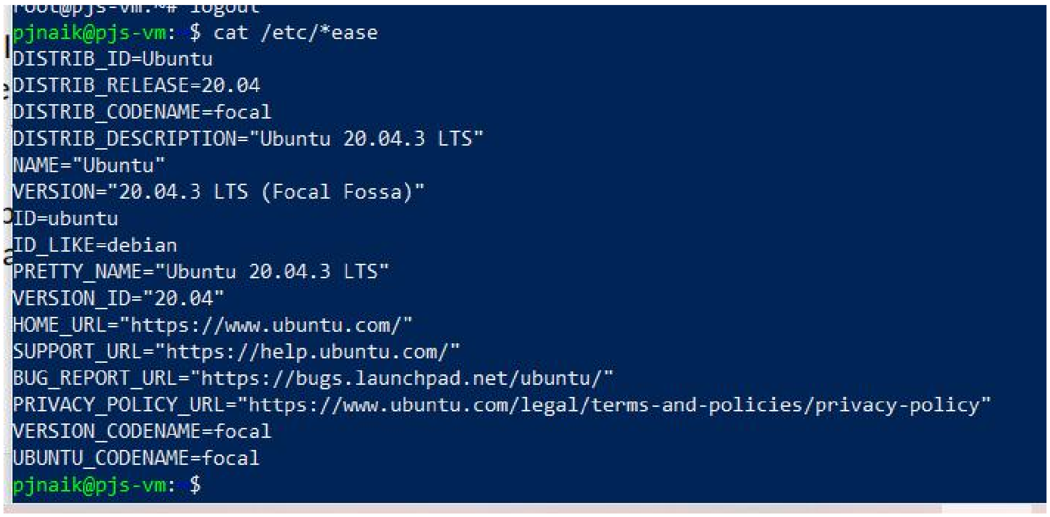
Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

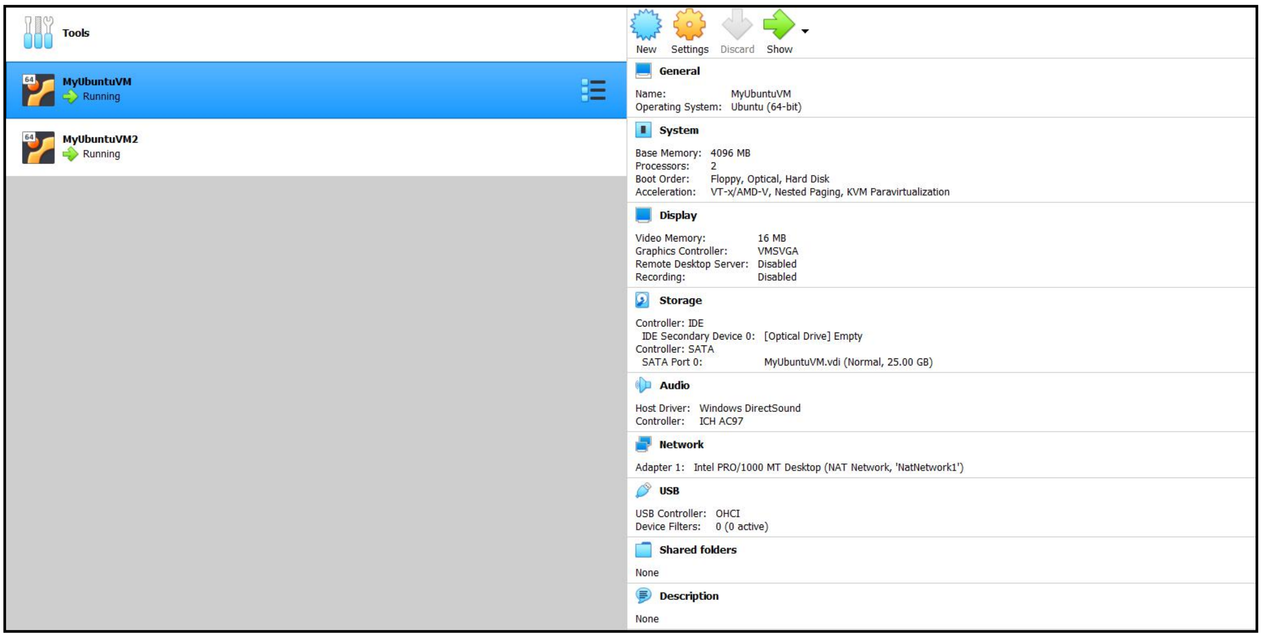
Controller: IDE

Q 1-d Download Ubuntu Desktop 20.04.3 LTS Linux



```
root@pjs-vm:~# logout
pjnaik@pjs-vm:~$ cat /etc/*ease
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.3 LTS"
NAME="Ubuntu"
VERSION="20.04.3 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.3 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
pjnaik@pjs-vm:~$
```

Q 1-e Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB.



General

Name: MyUbuntuVM
Operating System: Ubuntu (64-bit)

System

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: MyUbuntuVM.vdi (Normal, 25.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT Network, 'NatNetwork1')

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

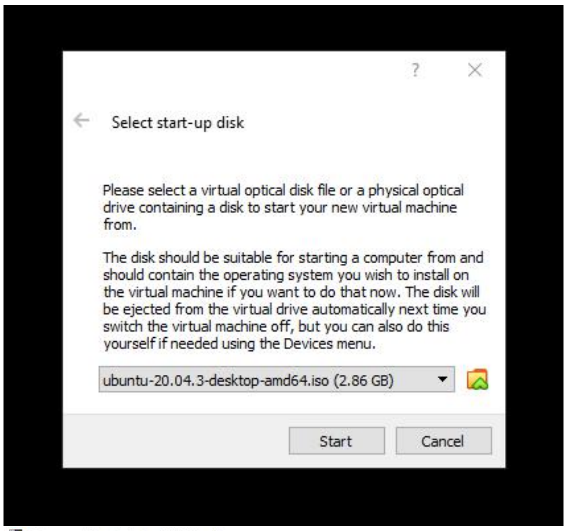
Shared folders

None

Description

None

Q 1-f Install Linux from ISO

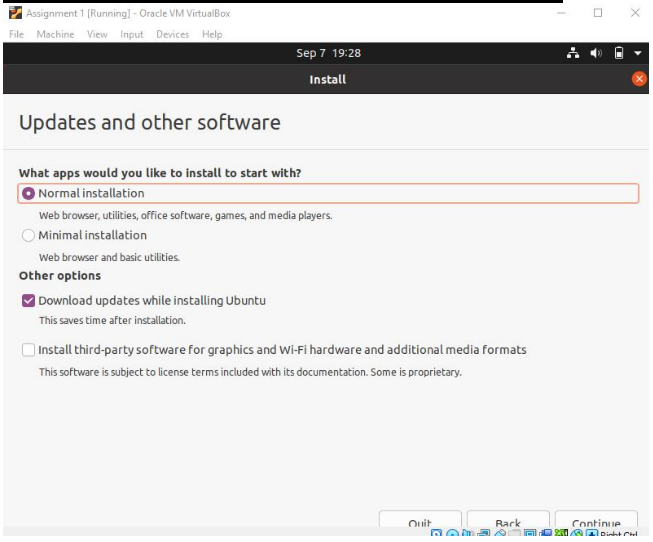


Select start-up disk

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

ubuntu-20.04.3-desktop-amd64.iso (2.86 GB)



Assignment 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Sep 7 19:28
Install

Updates and other software

What apps would you like to install to start with?

[✓ Normal installation]

Web browser, utilities, office software, games, and media players.

Minimal installation

Web browser and basic utilities.

Other options

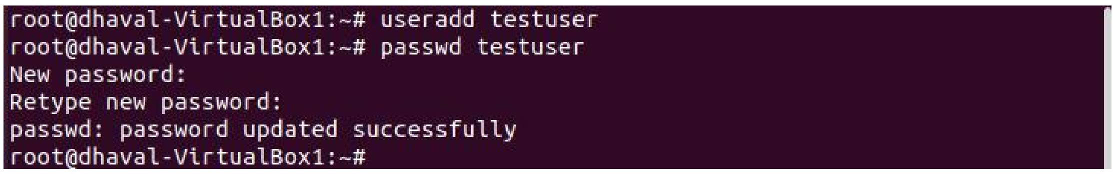
[✓ Download updates while installing Ubuntu]

This saves time after installation.

Install third-party software for graphics and Wi-Fi hardware and additional media formats

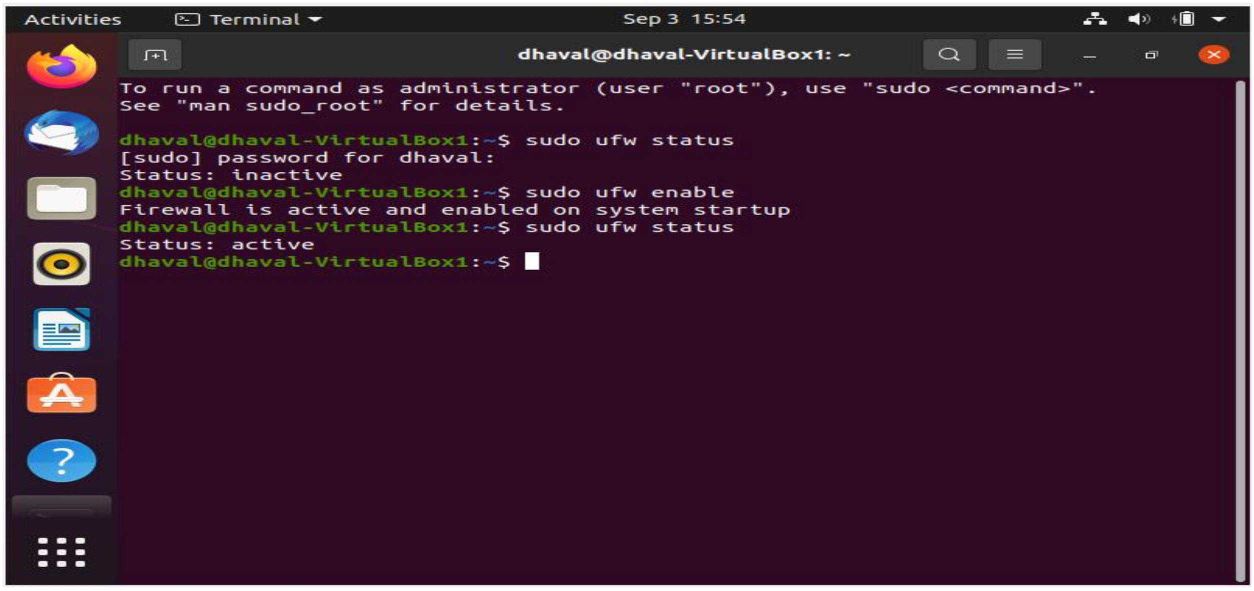
This software is subject to license terms included with its documentation. Some is proprietary.

Q 1-g Create a user id and password



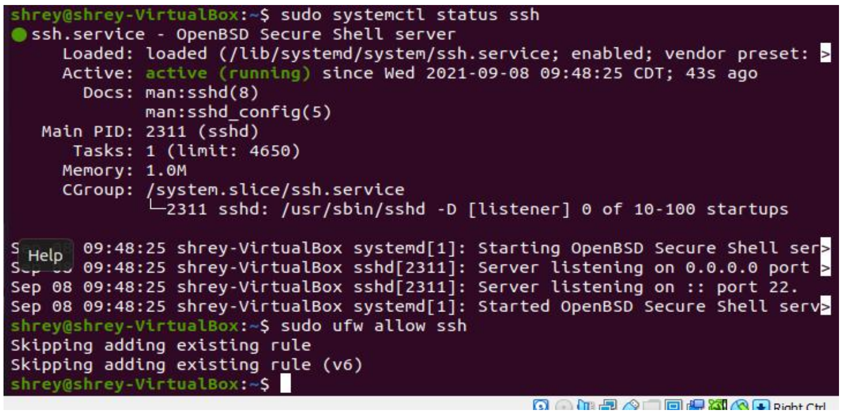
```
root@dhaval-VirtualBox1:~# useradd testuser
root@dhaval-VirtualBox1:~# passwd testuser
New password:
Retype new password:
passwd: password updated successfully
root@dhaval-VirtualBox1:~#
```

Q 1-h Turn on Firewall and block all ports

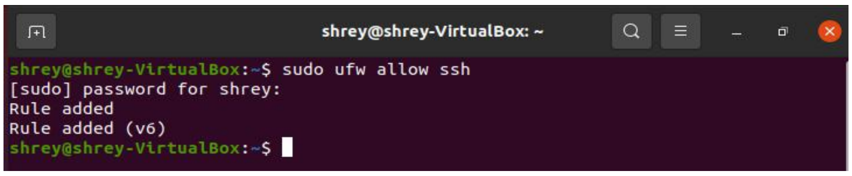


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dhaval@dhaval-VirtualBox1:~$ sudo ufw status
[sudo] password for dhaval:
Status: inactive
dhaval@dhaval-VirtualBox1:~$ sudo ufw enable
Firewall is active and enabled on system startup
dhaval@dhaval-VirtualBox1:~$ sudo ufw status
Status: active
dhaval@dhaval-VirtualBox1:~$
```

Q 1-I Enable SSH access to your new Linux installation; open SSH port in firewall

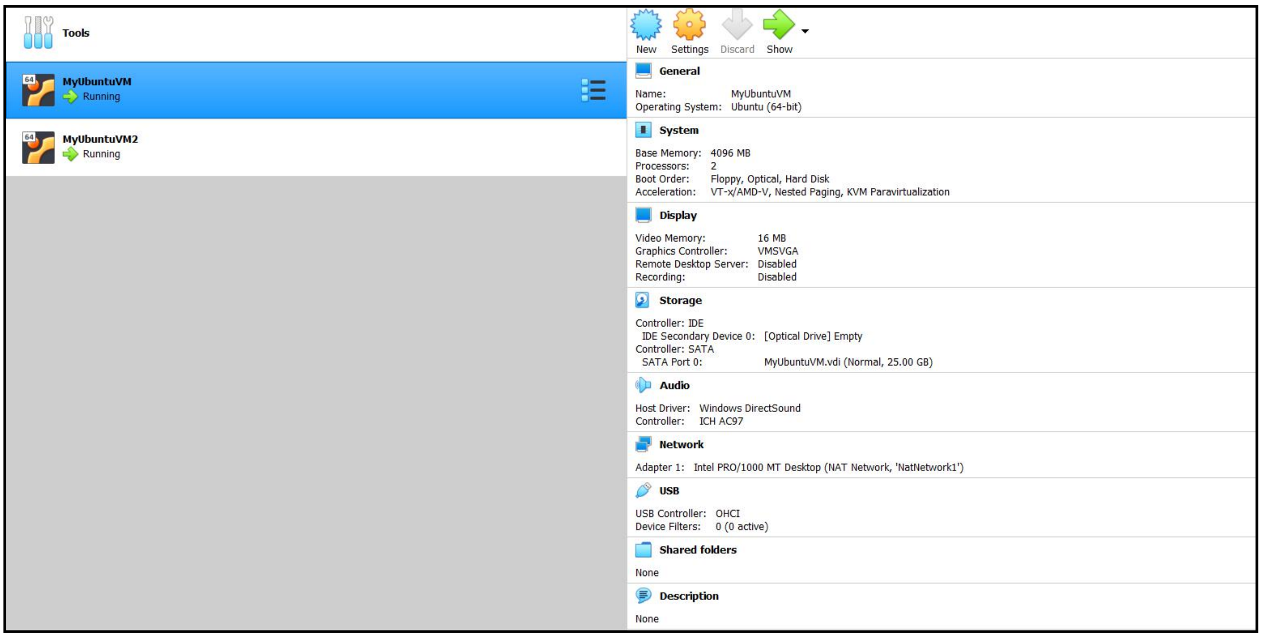


```
shrey@shrey-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Wed 2021-09-08 09:48:25 CDT; 43s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 2311 (sshd)
      Tasks: 1 (limit: 4650)
     Memory: 1.0M
     CGroup: /system.slice/ssh.service
             └─2311 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

S     09:48:25 shrey-VirtualBox systemd[1]: Starting OpenBSD Secure Shell ser>
Sep   09:48:25 shrey-VirtualBox sshd[2311]: Server listening on 0.0.0.0 port >
Sep 08 09:48:25 shrey-VirtualBox sshd[2311]: Server listening on :: port 22.
Sep 08 09:48:25 shrey-VirtualBox systemd[1]: Started OpenBSD Secure Shell serv>
shrey@shrey-VirtualBox:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
shrey@shrey-VirtualBox:~$
```



```
shrey@shrey-VirtualBox:~$ sudo ufw allow ssh
[sudo] password for shrey:
Rule added
Rule added (v6)
shrey@shrey-VirtualBox:~$
```

Q 1-j Repeat steps e through i, and create another VM with the same specifications as the first one



General

Name: MyUbuntuVM
Operating System: Ubuntu (64-bit)

System

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: MyUbuntuVM.vdi (Normal, 25.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT Network, 'NatNetwork1')

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

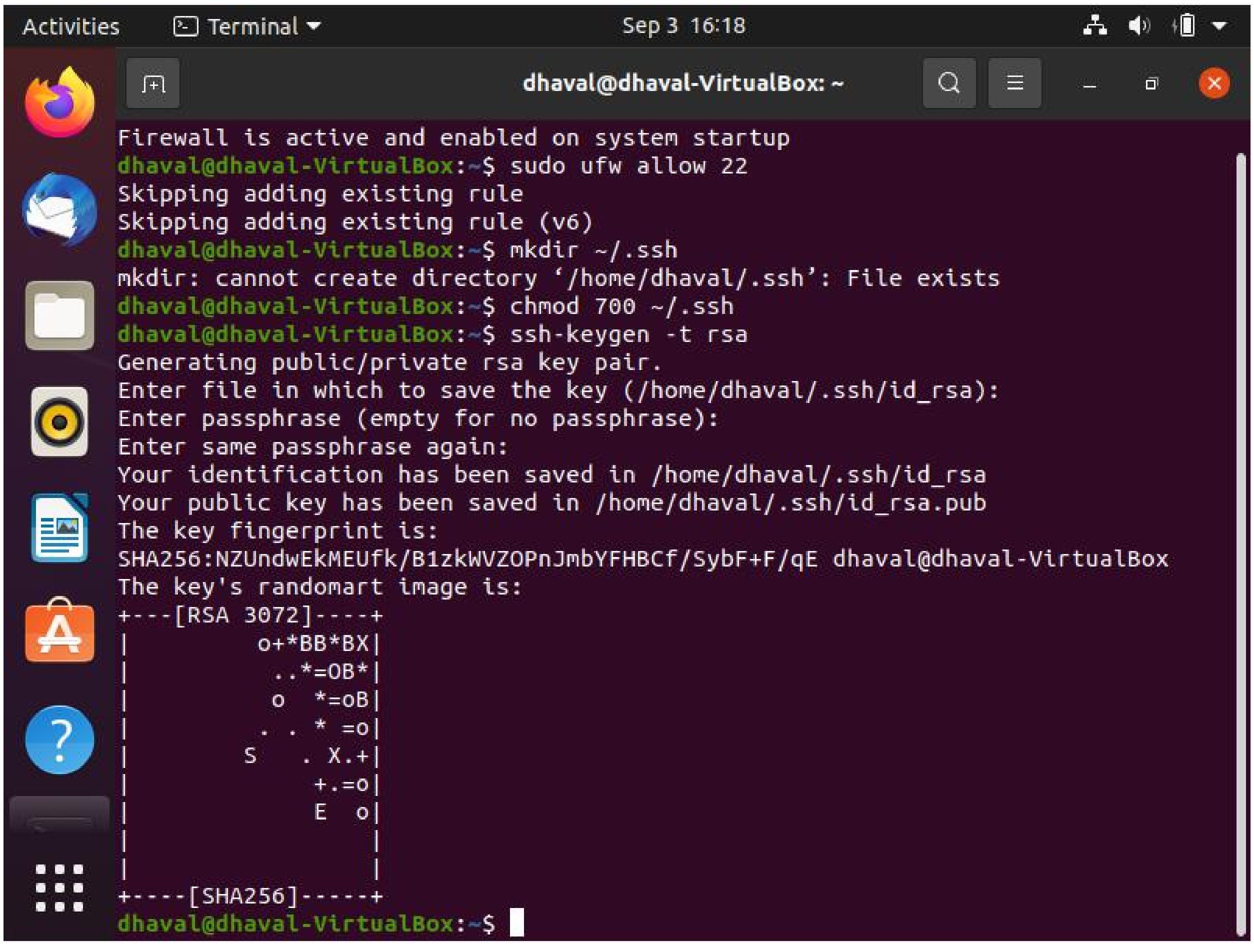
Shared folders

None

Description

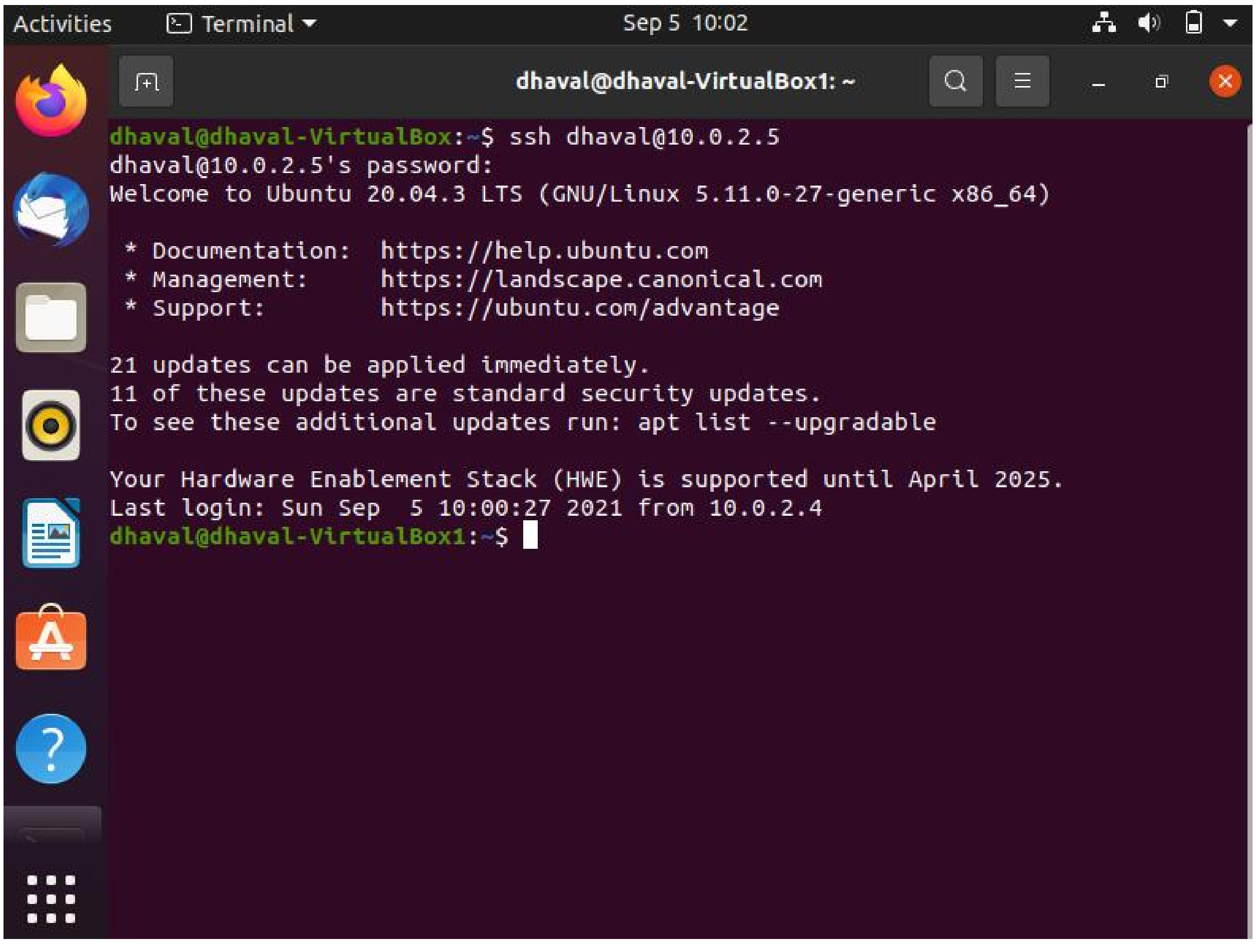
None

Q 1-k Create private/public keys and install them properly in both of your new VMs



```
Firewall is active and enabled on system startup
dhaval@dhaval-VirtualBox:~$ sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
dhaval@dhaval-VirtualBox:~$ mkdir ~/.ssh
mkdir: cannot create directory '/home/dhaval/.ssh': File exists
dhaval@dhaval-VirtualBox:~$ chmod 700 ~/.ssh
dhaval@dhaval-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dhaval/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dhaval/.ssh/id_rsa
Your public key has been saved in /home/dhaval/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:NZUndwEkMEUfk/B1zkWVZOPnJmbYFHBCf/SybF+F/qE dhaval@dhaval-VirtualBox
The key's randomart image is:
+---[RSA 3072]----+
|         o+*BB*BX|
|          ..*=OB*|
|          o  *=oB|
|         . . * =o|
|        S   . X.+|
|             +.=o|
|             E  o|
|                 |
|                 |
+----[SHA256]-----+
dhaval@dhaval-VirtualBox:~$
```

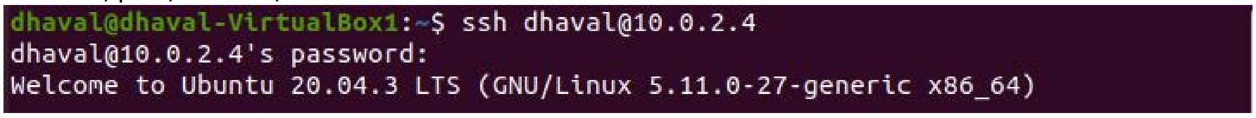
Q 1-l Test that you can connect remotely to your VMs with your keys, from one VM to the other VM



```
dhaval@dhaval-VirtualBox:~$ ssh dhaval@10.0.2.5
dhaval@10.0.2.5's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

21 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Sep  5 10:00:27 2021 from 10.0.2.4
dhaval@dhaval-VirtualBox1:~$
```

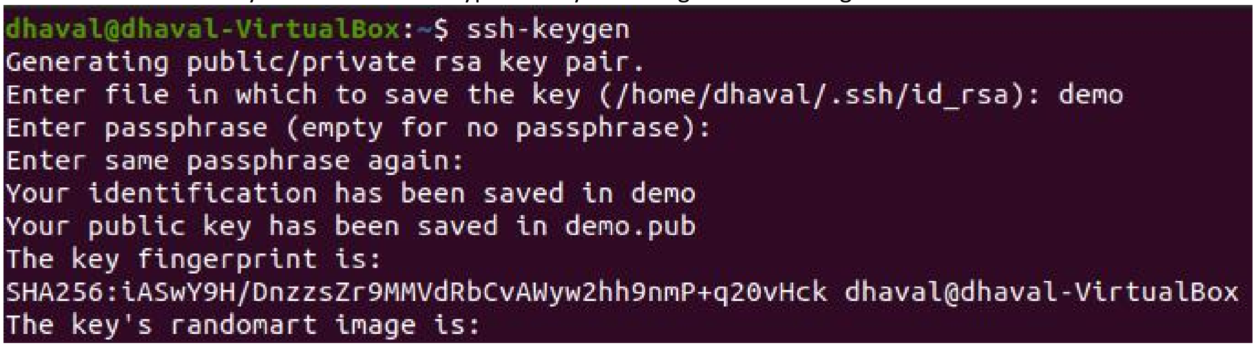
Question 2:

A. Ssh:
This command is used to connect to a machine by specifying various parameters like host, port, address, etc.



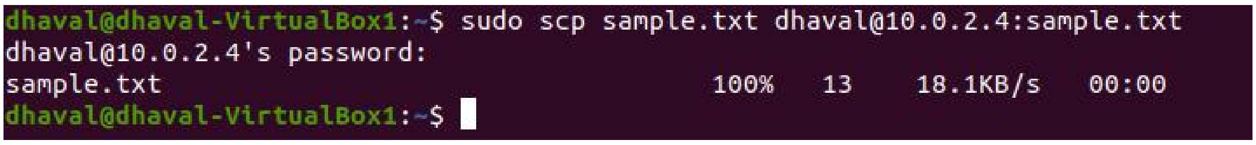
```
dhaval@dhaval-VirtualBox1:~$ ssh dhaval@10.0.2.4
dhaval@10.0.2.4's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-27-generic x86_64)
```

B. **Ssh-keygen**:
This command is used to generate keypair for authentication and is used to connect to other machines remotely via ssh. Different types of keys can be generated using –t .



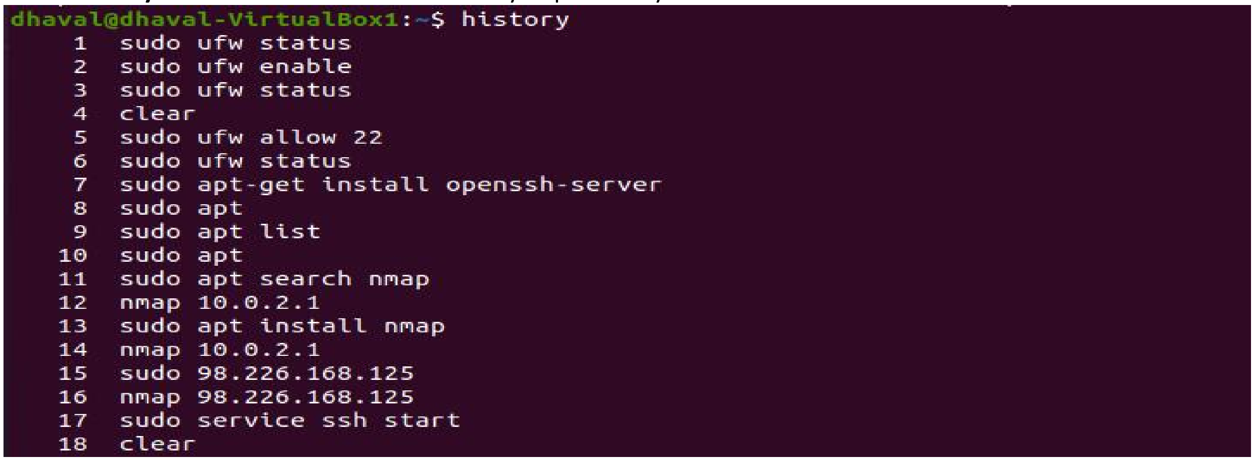
```
dhaval@dhaval-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dhaval/.ssh/id_rsa): demo
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in demo
Your public key has been saved in demo.pub
The key fingerprint is:
SHA256:iASwY9H/DnzzsZr9MMVdRbCvAWyw2hh9nmP+q20vHck dhaval@dhaval-VirtualBox
The key's randomart image is:
```

C. **Scp**:
This command is used to copy files from one host to another host in a same network.



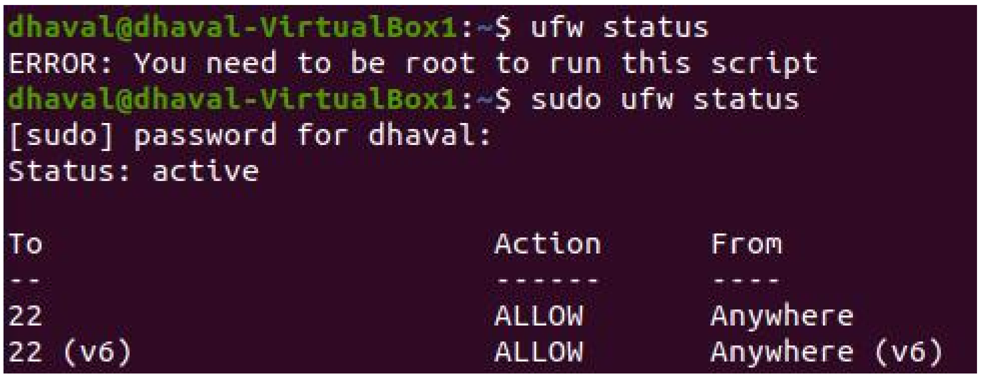
```
dhaval@dhaval-VirtualBox1:~$ sudo scp sample.txt dhaval@10.0.2.4:sample.txt
dhaval@10.0.2.4's password:
sample.txt                                    100%   13    18.1KB/s   00:00
dhaval@dhaval-VirtualBox1:~$
```

D.History: this command shows the history of previously executed commands in the shell.



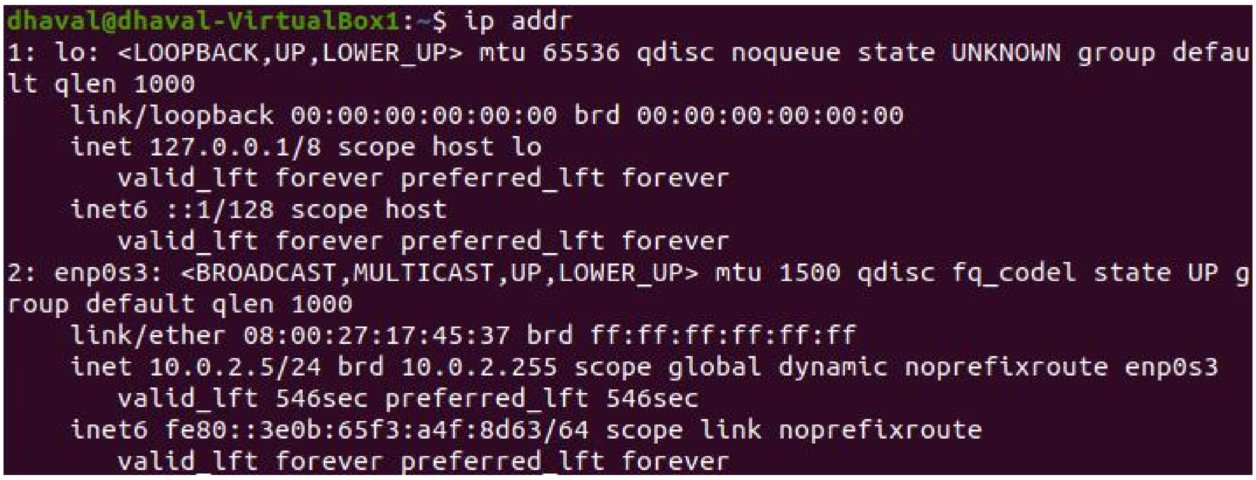
```
dhaval@dhaval-VirtualBox1:~$ history
    1  sudo ufw status
    2  sudo ufw enable
    3  sudo ufw status
    4  clear
    5  sudo ufw allow 22
    6  sudo ufw status
    7  sudo apt-get install openssh-server
    8  sudo apt
    9  sudo apt list
   10  sudo apt
   11  sudo apt search nmap
   12  nmap 10.0.2.1
   13  sudo apt install nmap
   14  nmap 10.0.2.1
   15  sudo 98.226.168.125
   16  nmap 98.226.168.125
   17  sudo service ssh start
   18  clear
```

E.**Sudo** : Authorized user can execute commands as a super user.



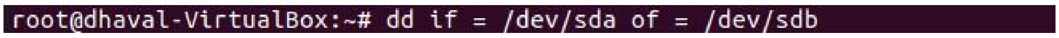
```
dhaval@dhaval-VirtualBox1:~$ ufw status
ERROR: You need to be root to run this script
dhaval@dhaval-VirtualBox1:~$ sudo ufw status
[sudo] password for dhaval:
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
```

F. **Ip**: show various stats related to IP, network and routing



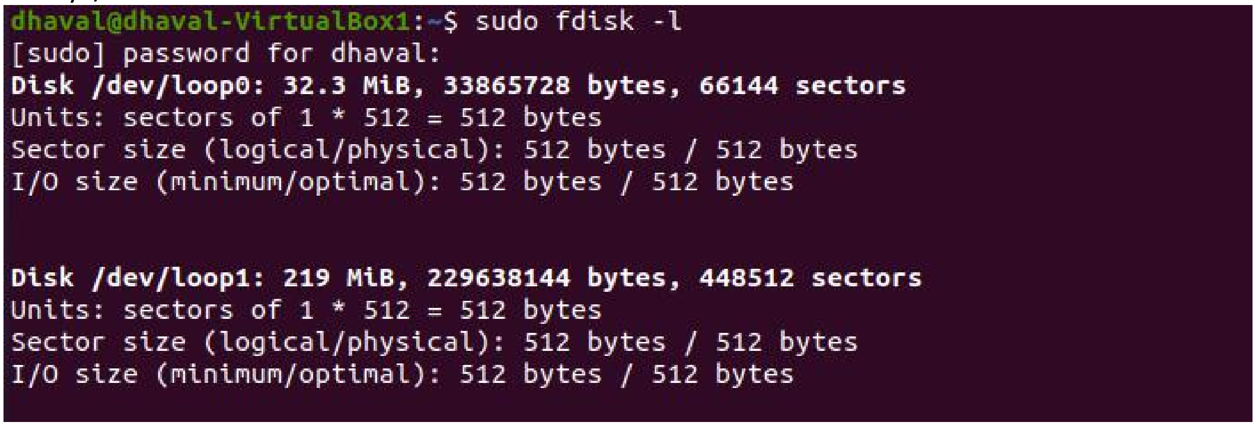
```
dhaval@dhaval-VirtualBox1:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:17:45:37 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 546sec preferred_lft 546sec
    inet6 fe80::3e0b:65f3:a4f:8d63/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

G. Dd



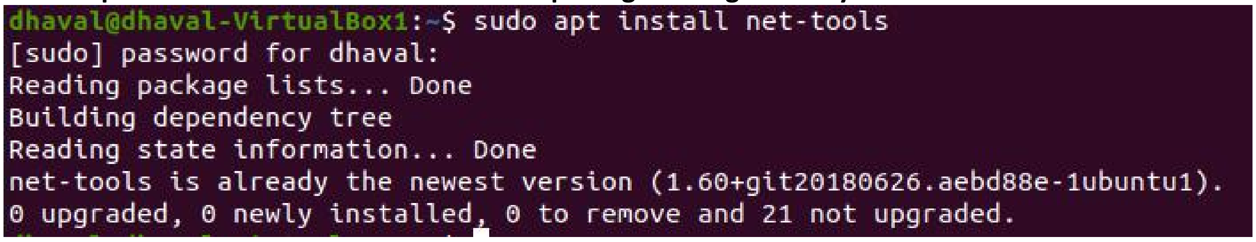
```
root@dhaval-VirtualBox:~# dd if = /dev/sda of = /dev/sdb
```

H. Fdisk: it is used to create and manipulate disk partition table. Default partition is done by I/O limits.



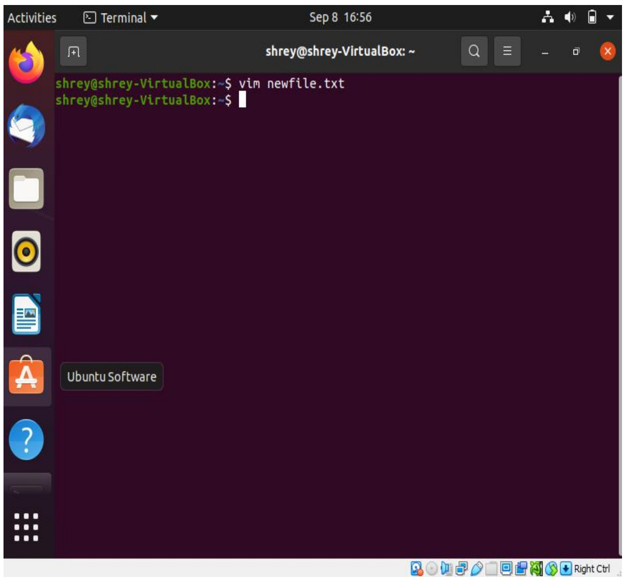
```
dhaval@dhaval-VirtualBox1:~$ sudo fdisk -l
[sudo] password for dhaval:
Disk /dev/loop0: 32.3 MiB, 33865728 bytes, 66144 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 219 MiB, 229638144 bytes, 448512 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

I. **Apt: It is a command line interface for package management system**

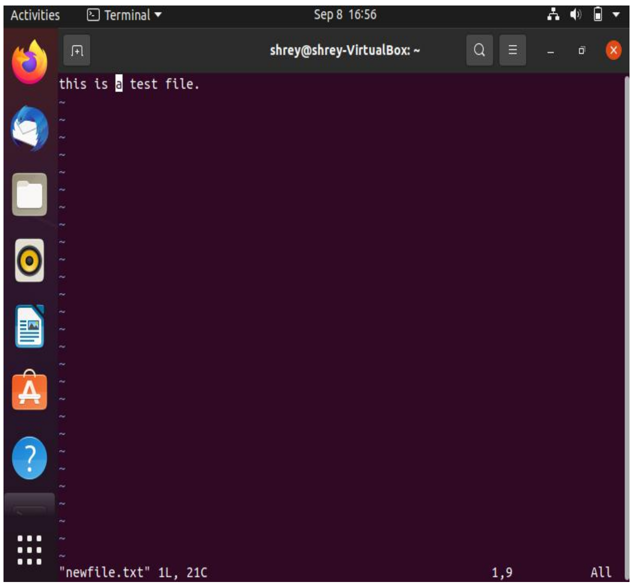


```
dhaval@dhaval-VirtualBox1:~$ sudo apt install net-tools
[sudo] password for dhaval:
Reading package lists... Done
Building dependency tree
Reading state information... Done
net-tools is already the newest version (1.60+git20180626.aebd88e-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 21 not upgraded.
```

J. **Vi: it is a text editor in ubuntu.**

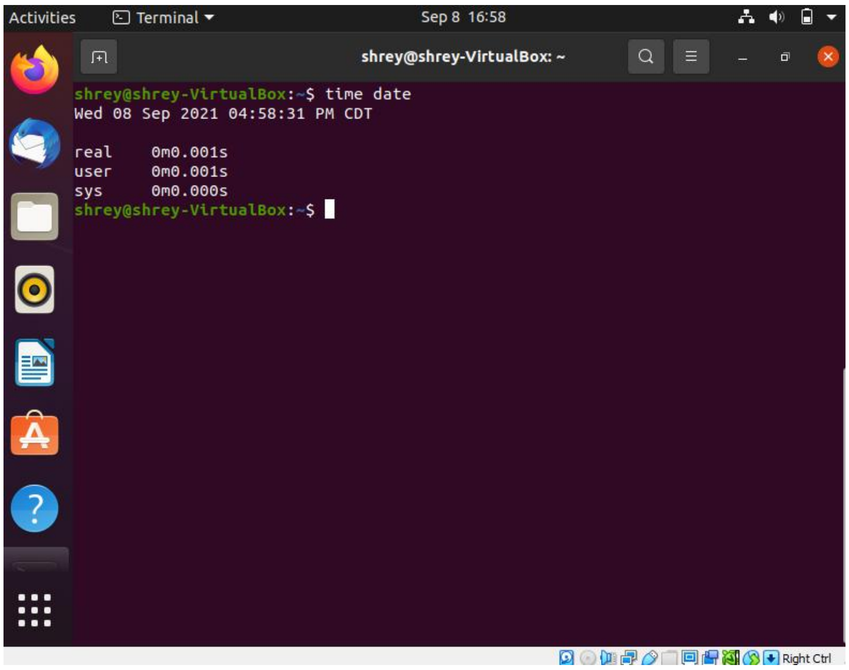


```
shrey@shrey-VirtualBox:~$ vim newfile.txt
shrey@shrey-VirtualBox:~$
```



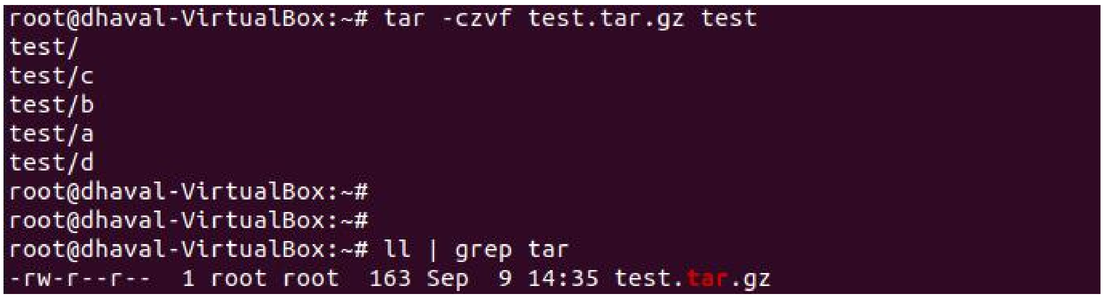
```
this is a test file.
~
~
"newfile.txt" 1L, 21C                                   1,9           All
```

K. Time: Show for how long the command runs in real time, user time and system time.



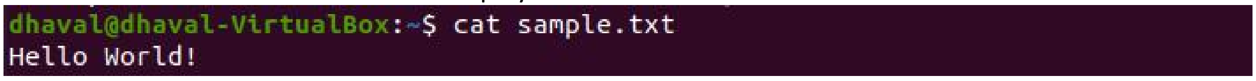
```
shrey@shrey-VirtualBox:~$ time date
Wed 08 Sep 2021 04:58:31 PM CDT

real    0m0.001s
user    0m0.001s
sys     0m0.000s
shrey@shrey-VirtualBox:~$
```

L. Tar: This command is used to archive a file.



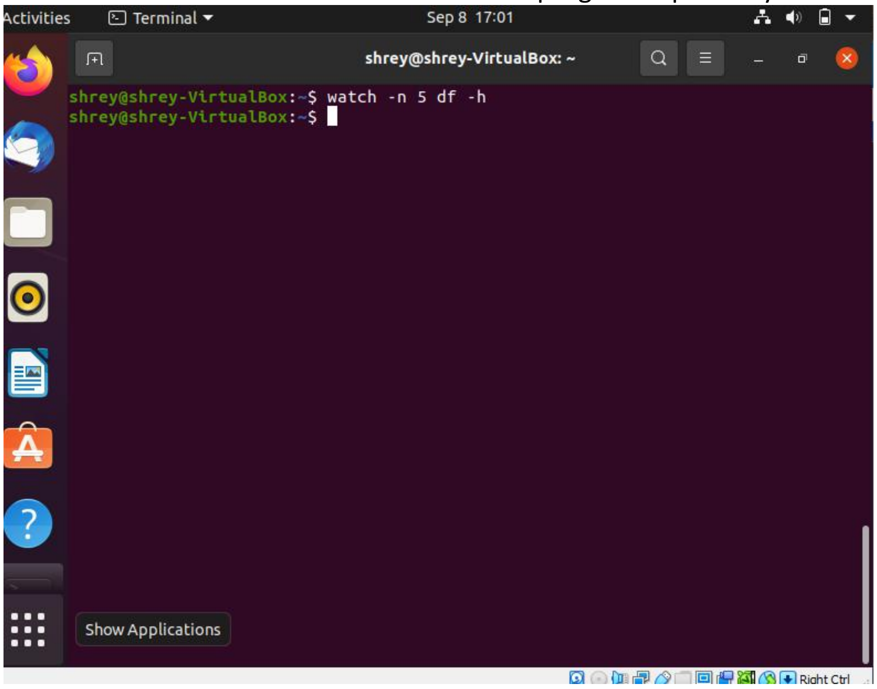
```
root@dhaval-VirtualBox:~# tar -czvf test.tar.gz test
test/
test/c
test/b
test/a
test/d
root@dhaval-VirtualBox:~#
root@dhaval-VirtualBox:~#
root@dhaval-VirtualBox:~# ll | grep tar
-rw-r--r--  1 root root  163 Sep  9 14:35 test.tar.gz
```

M. **Cat**: This command is used to display the content of a file

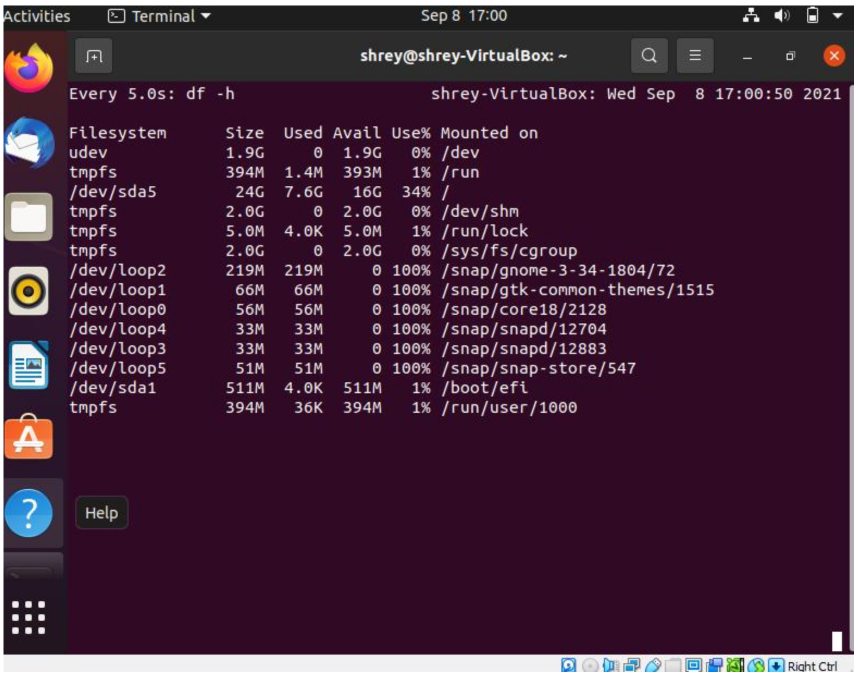


```
dhaval@dhaval-VirtualBox:~$ cat sample.txt
Hello World!
```

N. **Watch**: this command allows to run a program repeatedly after every certain period.

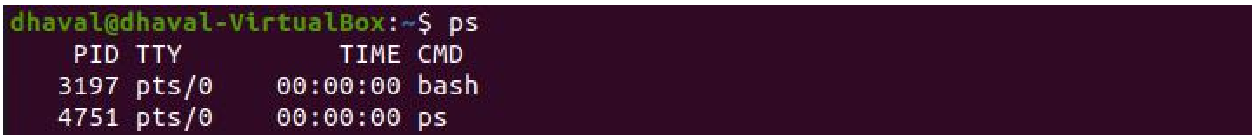


```
shrey@shrey-VirtualBox:~$ watch -n 5 df -h
shrey@shrey-VirtualBox:~$
```



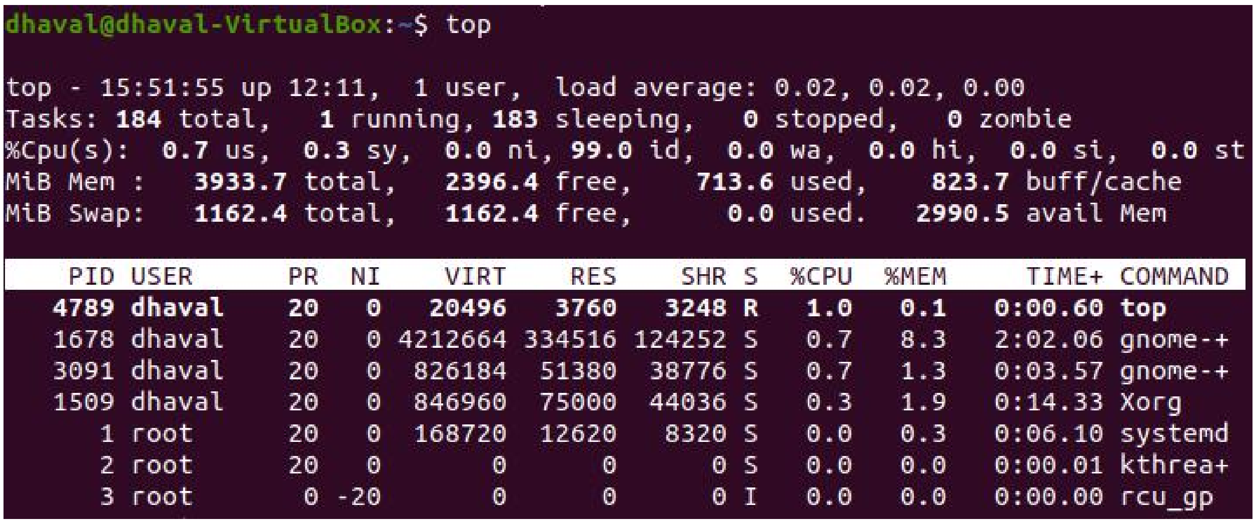
```
Every 5.0s: df -h                    shrey-VirtualBox: Wed Sep  8 17:00:50 2021

Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           394M  1.4M  393M   1% /run
/dev/sda5        24G  7.6G   16G  34% /
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/loop2      219M  219M     0 100% /snap/gnome-3-34-1804/72
/dev/loop1       66M   66M     0 100% /snap/gtk-common-themes/1515
/dev/loop0       56M   56M     0 100% /snap/core18/2128
/dev/loop4       33M   33M     0 100% /snap/snapd/12704
/dev/loop3       33M   33M     0 100% /snap/snapd/12883
/dev/loop5       51M   51M     0 100% /snap/snap-store/547
/dev/sda1       511M  4.0K  511M   1% /boot/efi
tmpfs           394M   36K  394M   1% /run/user/1000
```

O. **Ps**: It is used to get a snapshot of the current processes at that particular instance.



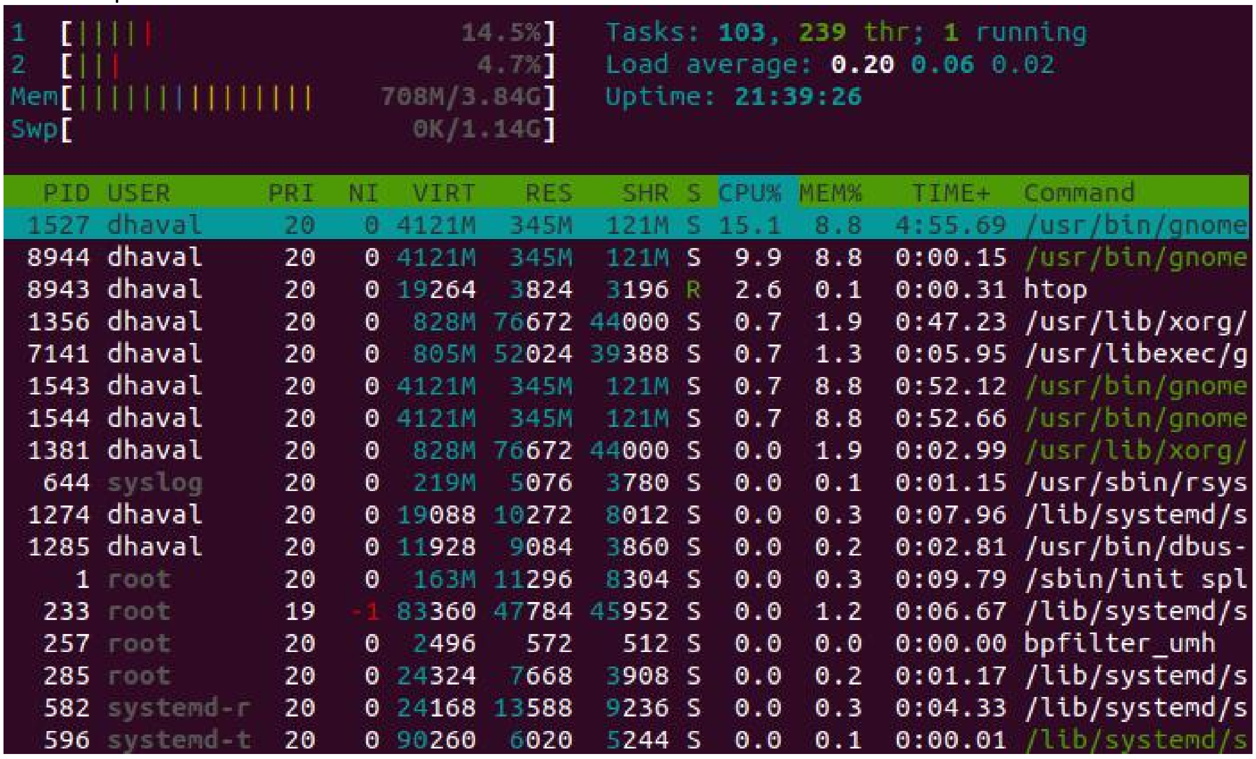
```
dhaval@dhaval-VirtualBox:~$ ps
    PID TTY          TIME CMD
   3197 pts/0    00:00:00 bash
   4751 pts/0    00:00:00 ps
```

P. TOP: this command is used to get the resources used by each process in actual time(i.e. it gets updated every second)



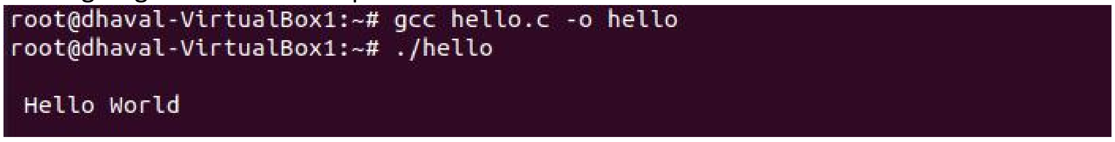
```
dhaval@dhaval-VirtualBox:~$ top

top - 15:51:55 up 12:11,  1 user,  load average: 0.02, 0.02, 0.00
Tasks: 184 total,   1 running, 183 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  0.3 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  3933.7 total,  2396.4 free,   713.6 used,   823.7 buff/cache
MiB Swap:  1162.4 total,  1162.4 free,     0.0 used.  2990.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   4789 dhaval    20   0   20496   3760   3248 R   1.0   0.1   0:00.60 top
   1678 dhaval    20   0 4212664 334516 124252 S   0.7   8.3   2:02.06 gnome-+
   3091 dhaval    20   0  826184  51380  38776 S   0.7   1.3   0:03.57 gnome-+
   1509 dhaval    20   0  846960  75000  44036 S   0.3   1.9   0:14.33 Xorg
      1 root      20   0  168720  12620   8320 S   0.0   0.3   0:06.10 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthrea+
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
```

Q. htop: it functions the same as htop but it is a more interactive to user and we can view processes with their command lines.



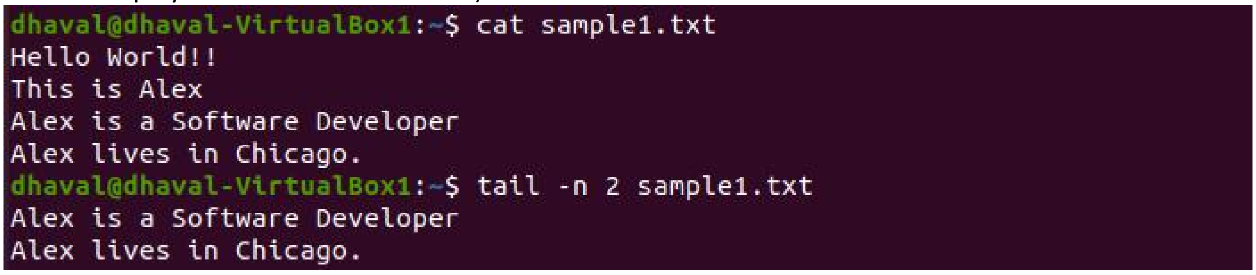
```
1  [|||||                   14.5%]   Tasks: 103, 239 thr; 1 running
2  [|||                      4.7%]   Load average: 0.20 0.06 0.02
Mem[|||||||||||||||    708M/3.84G]   Uptime: 21:39:26
Swp[                      0K/1.14G]

  PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 1527 dhaval      20   0 4121M  345M  121M S 15.1  8.8  4:55.69 /usr/bin/gnome
 8944 dhaval      20   0 4121M  345M  121M S  9.9  8.8  0:00.15 /usr/bin/gnome
 8943 dhaval      20   0 19264  3824  3196 R  2.6  0.1  0:00.31 htop
 1356 dhaval      20   0  828M 76672 44000 S  0.7  1.9  0:47.23 /usr/lib/xorg/
 7141 dhaval      20   0  805M 52024 39388 S  0.7  1.3  0:05.95 /usr/libexec/g
 1543 dhaval      20   0 4121M  345M  121M S  0.7  8.8  0:52.12 /usr/bin/gnome
 1544 dhaval      20   0 4121M  345M  121M S  0.7  8.8  0:52.66 /usr/bin/gnome
 1381 dhaval      20   0  828M 76672 44000 S  0.0  1.9  0:02.99 /usr/lib/xorg/
  644 syslog      20   0  219M  5076  3780 S  0.0  0.1  0:01.15 /usr/sbin/rsys
 1274 dhaval      20   0 19088 10272  8012 S  0.0  0.3  0:07.96 /lib/systemd/s
 1285 dhaval      20   0 11928  9084  3860 S  0.0  0.2  0:02.81 /usr/bin/dbus-
    1 root        20   0  163M 11296  8304 S  0.0  0.3  0:09.79 /sbin/init spl
  233 root        19  -1 83360 47784 45952 S  0.0  1.2  0:06.67 /lib/systemd/s
  257 root        20   0  2496   572   512 S  0.0  0.0  0:00.00 bpfilter_umh
  285 root        20   0 24324  7668  3908 S  0.0  0.2  0:01.17 /lib/systemd/s
  582 systemd-r   20   0 24168 13588  9236 S  0.0  0.3  0:04.33 /lib/systemd/s
  596 systemd-t   20   0 90260  6020  5244 S  0.0  0.1  0:00.01 /lib/systemd/s
```

R. gcc: gcc is used to compile C files in linux.



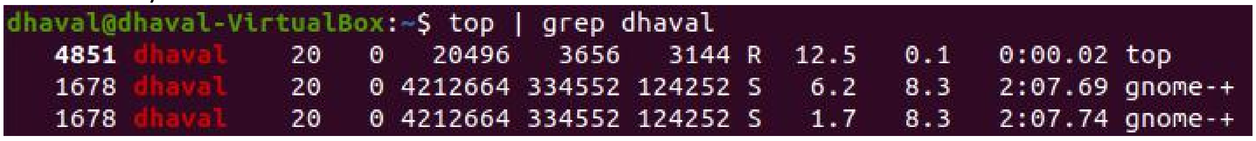
```
root@dhaval-VirtualBox1:~# gcc hello.c -o hello
root@dhaval-VirtualBox1:~# ./hello

 Hello World
```

S. tail: It is used to display the last few lines of a particular file(i.e. if the value of n is 2, it will display last two lines of the file.)



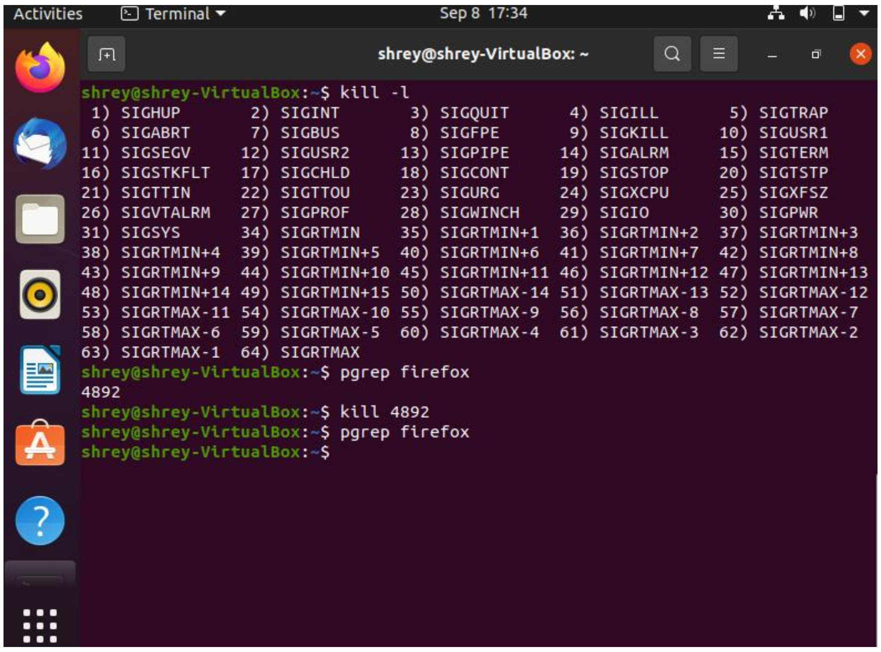
```
dhaval@dhaval-VirtualBox1:~$ cat sample1.txt
Hello World!!
This is Alex
Alex is a Software Developer
Alex lives in Chicago.
dhaval@dhaval-VirtualBox1:~$ tail -n 2 sample1.txt
Alex is a Software Developer
Alex lives in Chicago.
```

T. grep: this command is used to find a particular pattern from any file, output or a directory.



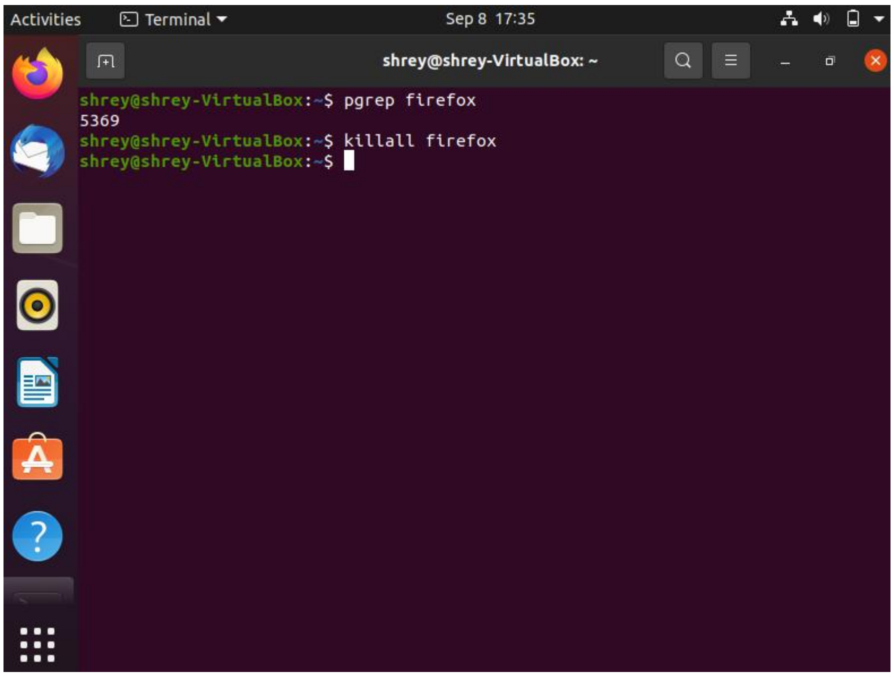
```
dhaval@dhaval-VirtualBox:~$ top | grep dhaval
   4851 dhaval    20   0   20496   3656   3144 R  12.5   0.1   0:00.02 top
   1678 dhaval    20   0 4212664 334552 124252 S   6.2   8.3   2:07.69 gnome-+
   1678 dhaval    20   0 4212664 334552 124252 S   1.7   8.3   2:07.74 gnome-+
```

U. kill:



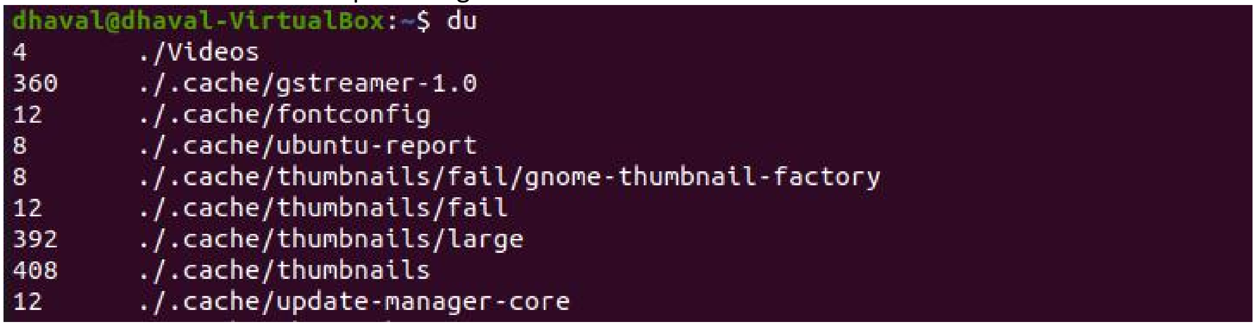
```
shrey@shrey-VirtualBox:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) 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
shrey@shrey-VirtualBox:~$ pgrep firefox
4892
shrey@shrey-VirtualBox:~$ kill 4892
shrey@shrey-VirtualBox:~$ pgrep firefox
shrey@shrey-VirtualBox:~$
```

V. killall: It is used to kill processes by name.



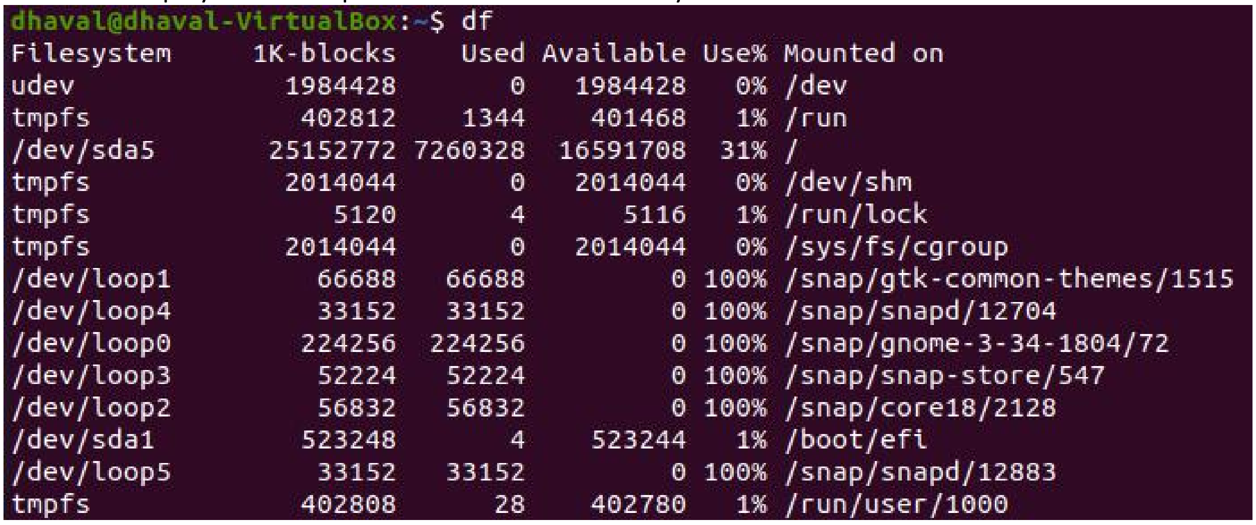
```
shrey@shrey-VirtualBox:~$ pgrep firefox
5369
shrey@shrey-VirtualBox:~$ killall firefox
shrey@shrey-VirtualBox:~$
```

W. Du: estimate the file space usage.



```
dhaval@dhaval-VirtualBox:~$ du
4       ./Videos
360     ./.cache/gstreamer-1.0
12      ./.cache/fontconfig
8       ./.cache/ubuntu-report
8       ./.cache/thumbnails/fail/gnome-thumbnail-factory
12      ./.cache/thumbnails/fail
392     ./.cache/thumbnails/large
408     ./.cache/thumbnails
12      ./.cache/update-manager-core
```

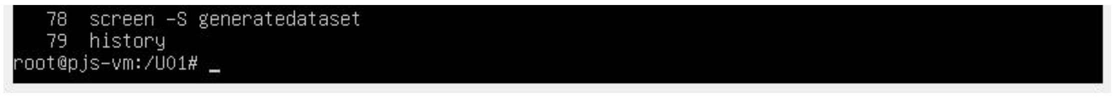
X. Df: displays the disk space available on each file system.



```
dhaval@dhaval-VirtualBox:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
udev             1984428       0   1984428   0% /dev
tmpfs             402812    1344    401468   1% /run
/dev/sda5       25152772 7260328  16591708  31% /
tmpfs            2014044       0   2014044   0% /dev/shm
tmpfs               5120       4      5116   1% /run/lock
tmpfs            2014044       0   2014044   0% /sys/fs/cgroup
/dev/loop1         66688   66688         0 100% /snap/gtk-common-themes/1515
/dev/loop4         33152   33152         0 100% /snap/snapd/12704
/dev/loop0        224256  224256         0 100% /snap/gnome-3-34-1804/72
/dev/loop3         52224   52224         0 100% /snap/snap-store/547
/dev/loop2         56832   56832         0 100% /snap/core18/2128
/dev/sda1         523248       4    523244   1% /boot/efi
/dev/loop5         33152   33152         0 100% /snap/snapd/12883
tmpfs             402808      28    402780   1% /run/user/1000
```

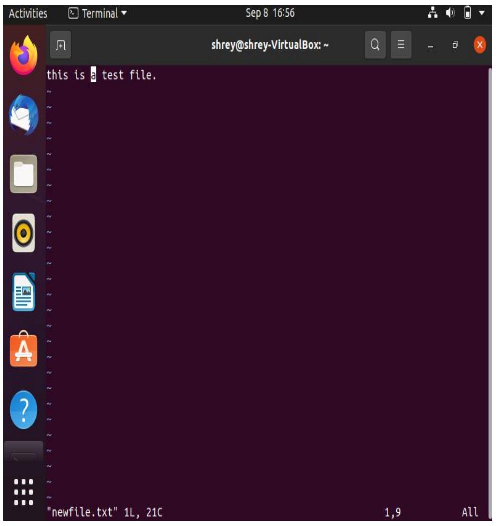
Y. Screen:

Screen is used to run commands in background. The advantage of this command is that the command will continue to execute even If the shell or ssh session fails.

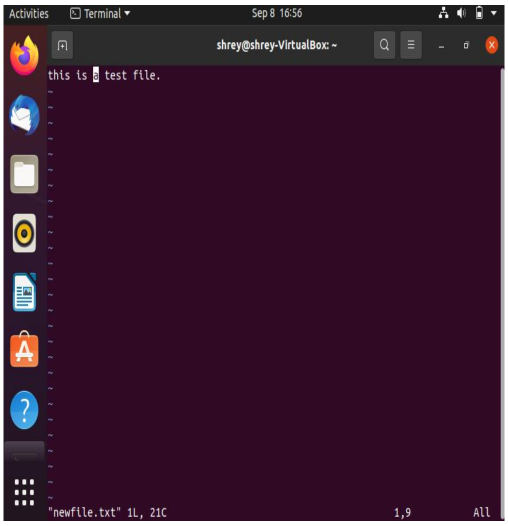


```
   78  screen -S generatedataset
   79  history
root@pjs-vm:/U01# _
```

Z. Vim: It is a text editor in linux shell.

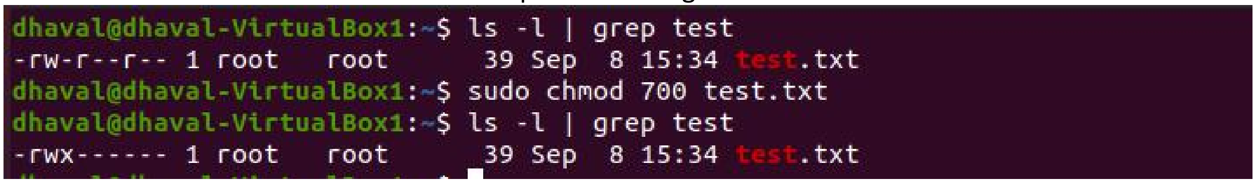


```
this is a test file.
"newfile.txt" 1L, 21C                                    1,9           All
```



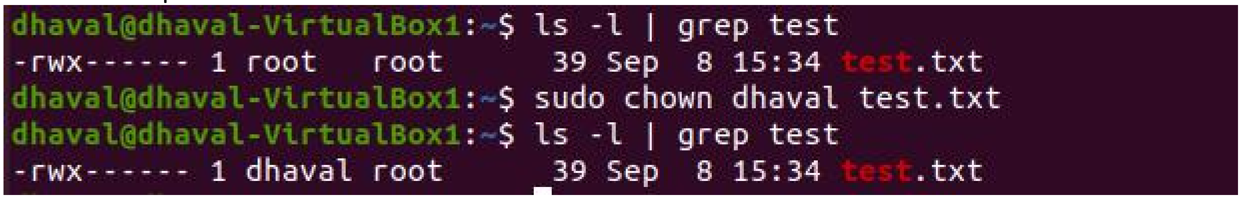
```
this is a test file.
"newfile.txt" 1L, 21C                                    1,9           All
```

AA. **CHMOD** : This command is used to perform change in bits of a file



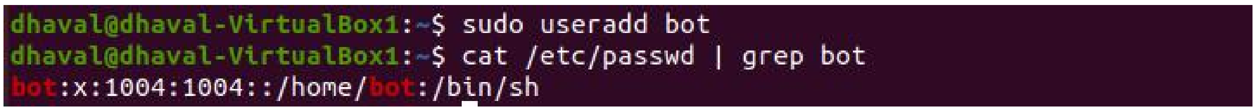
```
dhaval@dhaval-VirtualBox1:~$ ls -l | grep test
-rw-r--r-- 1 root   root      39 Sep  8 15:34 test.txt
dhaval@dhaval-VirtualBox1:~$ sudo chmod 700 test.txt
dhaval@dhaval-VirtualBox1:~$ ls -l | grep test
-rwx------ 1 root   root      39 Sep  8 15:34 test.txt
```

AB. chown: It is used to change the owner of a file. Also we can provide or remove group ownership of a file.



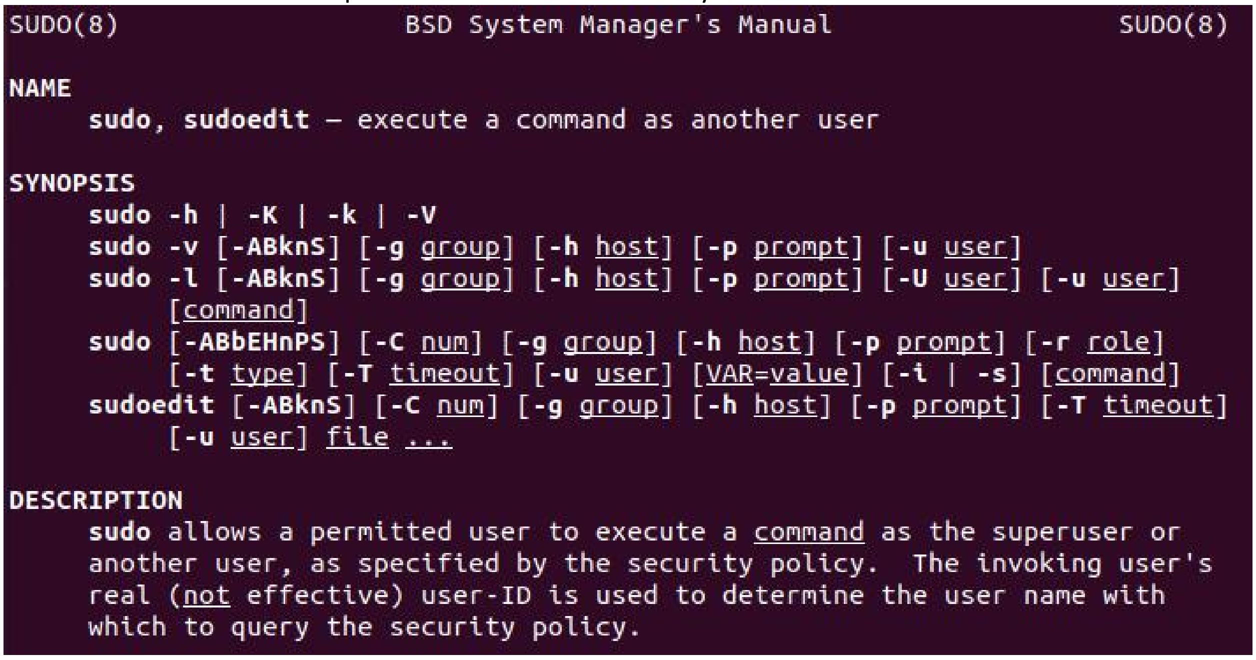
```
dhaval@dhaval-VirtualBox1:~$ ls -l | grep test
-rwx------ 1 root   root      39 Sep  8 15:34 test.txt
dhaval@dhaval-VirtualBox1:~$ sudo chown dhaval test.txt
dhaval@dhaval-VirtualBox1:~$ ls -l | grep test
-rwx------ 1 dhaval root      39 Sep  8 15:34 test.txt
```

AC. **useradd:** it is used to add a user



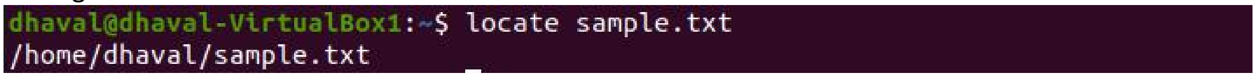
```
dhaval@dhaval-VirtualBox1:~$ sudo useradd bot
dhaval@dhaval-VirtualBox1:~$ cat /etc/passwd | grep bot
bot:x:1004:1004::/home/bot:/bin/sh
```

AD. man: this command provides a user manual for any of the commands.



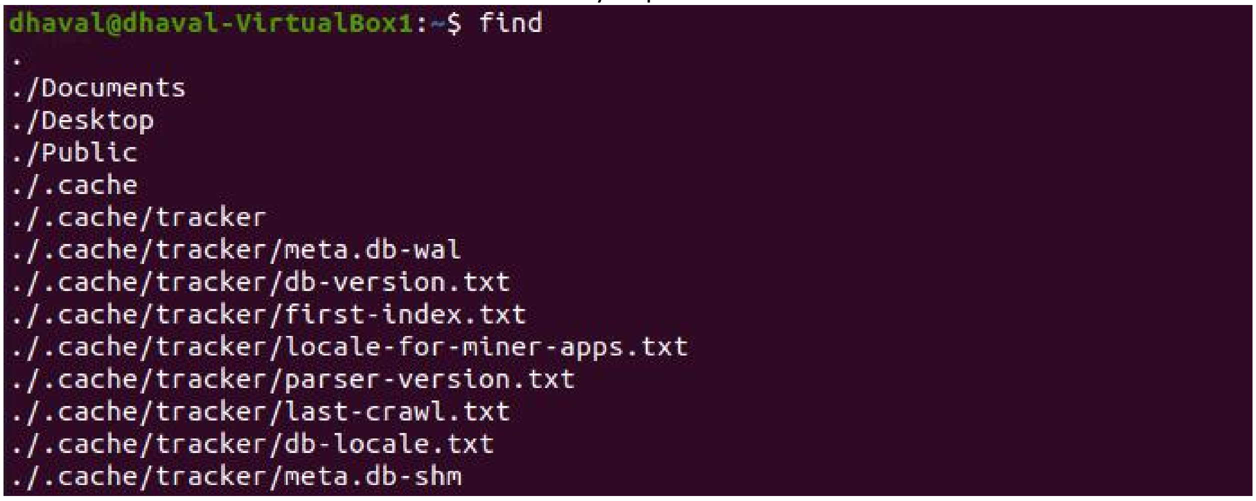
```
SUDO(8)                  BSD System Manager's Manual                 SUDO(8)

NAME
     sudo, sudoedit — execute a command as another user

SYNOPSIS
     sudo -h | -K | -k | -V
     sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
     sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
          [command]
     sudo [-ABbEHnPS] [-C num] [-g group] [-h host] [-p prompt] [-r role]
          [-t type] [-T timeout] [-u user] [VAR=value] [-i | -s] [command]
     sudoedit [-ABknS] [-C num] [-g group] [-h host] [-p prompt] [-T timeout]
          [-u user] file ...

DESCRIPTION
     sudo allows a permitted user to execute a command as the superuser or
     another user, as specified by the security policy.  The invoking user's
     real (not effective) user-ID is used to determine the user name with
     which to query the security policy.
```

AE. locate: this command is used to locate a specific file or directory mentioned in the argument.



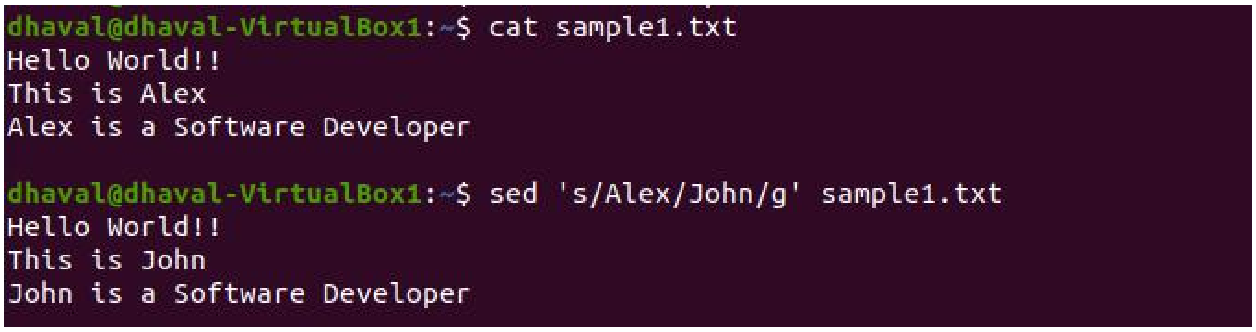
```
dhaval@dhaval-VirtualBox1:~$ locate sample.txt
/home/dhaval/sample.txt
```

AF. find: it is used to search a file in a directory. It provides the tree structure.



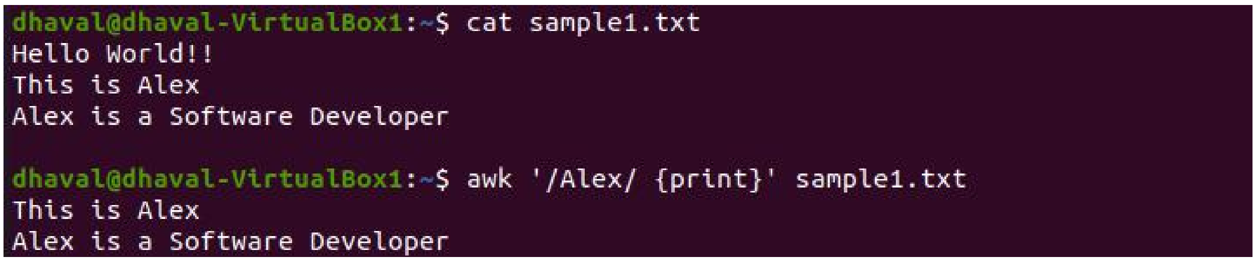
```
dhaval@dhaval-VirtualBox1:~$ find
.
./Documents
./Desktop
./Public
./.cache
./.cache/tracker
./.cache/tracker/meta.db-wal
./.cache/tracker/db-version.txt
./.cache/tracker/first-index.txt
./.cache/tracker/locale-for-miner-apps.txt
./.cache/tracker/parser-version.txt
./.cache/tracker/last-crawl.txt
./.cache/tracker/db-locale.txt
./.cache/tracker/meta.db-shm
```

AG. sed: it is a stream editor for performing and transforming text.



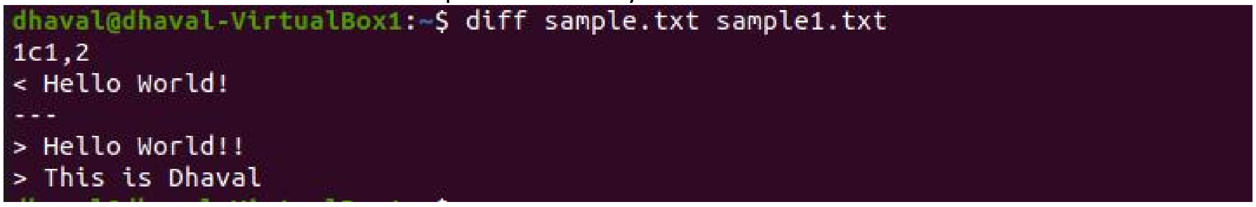
```
dhaval@dhaval-VirtualBox1:~$ cat sample1.txt
Hello World!!
This is Alex
Alex is a Software Developer

dhaval@dhaval-VirtualBox1:~$ sed 's/Alex/John/g' sample1.txt
Hello World!!
This is John
John is a Software Developer
```

AH. awk: This command is used for pattern scanning in multiple files. It scans for the word or pattern given in the argument and search for the same in the files mentioned and prints the lines where the pattern matches.



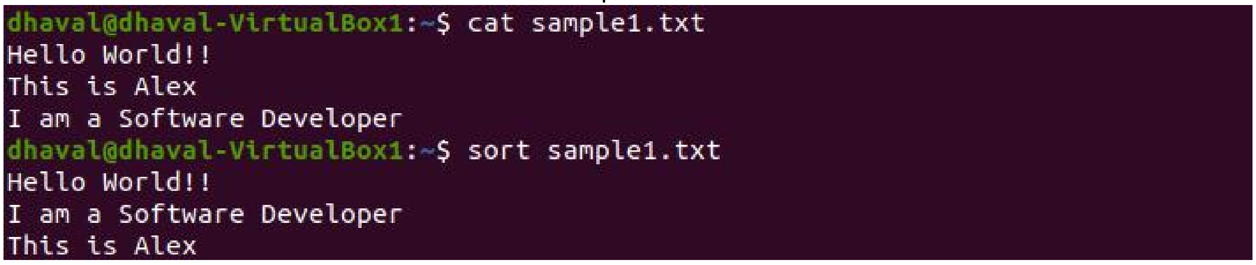
```
dhaval@dhaval-VirtualBox1:~$ cat sample1.txt
Hello World!!
This is Alex
Alex is a Software Developer

dhaval@dhaval-VirtualBox1:~$ awk '/Alex/ {print}' sample1.txt
This is Alex
Alex is a Software Developer
```

AI. diff: this command is used to compare files line by line.



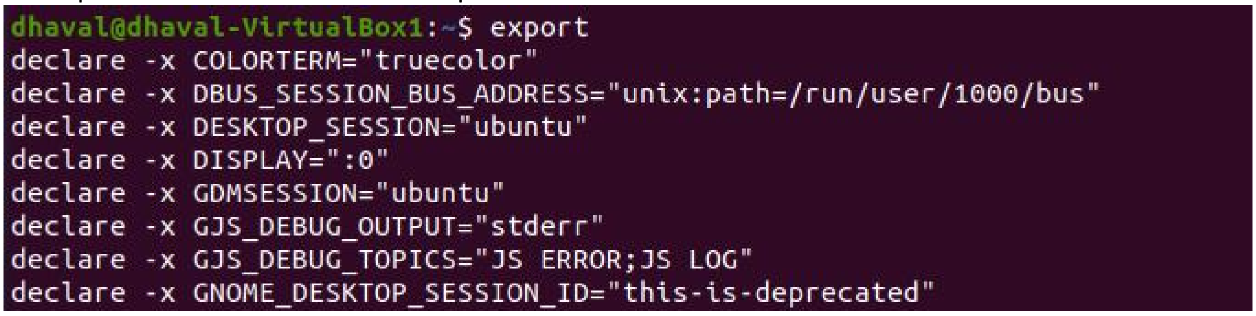
```
dhaval@dhaval-VirtualBox1:~$ diff sample.txt sample1.txt
1c1,2
< Hello World!
---
> Hello World!!
> This is Dhaval
```

AJ.sort: this command is used to sort content of a particular file.



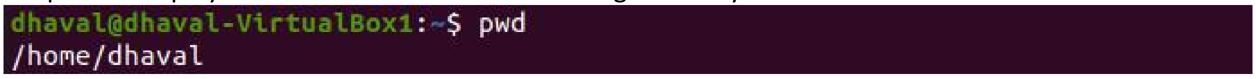
```
dhaval@dhaval-VirtualBox1:~$ cat sample1.txt
Hello World!!
This is Alex
I am a Software Developer
dhaval@dhaval-VirtualBox1:~$ sort sample1.txt
Hello World!!
I am a Software Developer
This is Alex
```

AK. export: It is used to view all the exported variables in a shell.



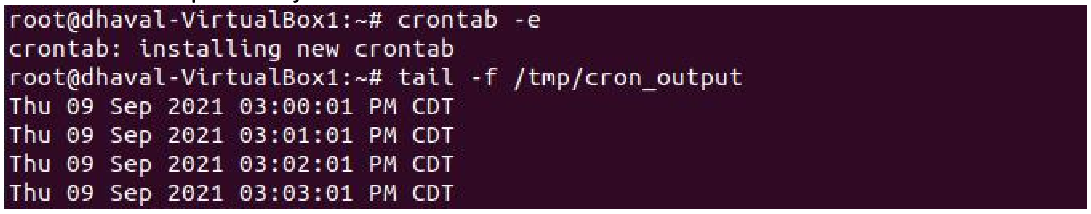
```
dhaval@dhaval-VirtualBox1:~$ export
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GJS_DEBUG_OUTPUT="stderr"
declare -x GJS_DEBUG_TOPICS="JS ERROR;JS LOG"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
```

AL. pwd: It displays the name of the current working directory.

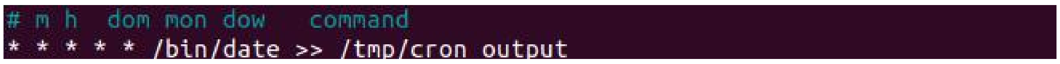


```
dhaval@dhaval-VirtualBox1:~$ pwd
/home/dhaval
```

AM. Crontab:
It is used to set periodic jobs in linux.

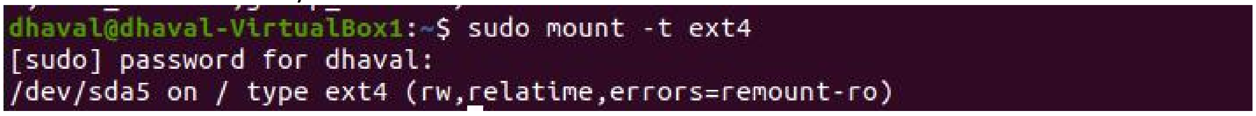


```
root@dhaval-VirtualBox1:~# crontab -e
crontab: installing new crontab
root@dhaval-VirtualBox1:~# tail -f /tmp/cron_output
Thu 09 Sep 2021 03:00:01 PM CDT
Thu 09 Sep 2021 03:01:01 PM CDT
Thu 09 Sep 2021 03:02:01 PM CDT
Thu 09 Sep 2021 03:03:01 PM CDT
```



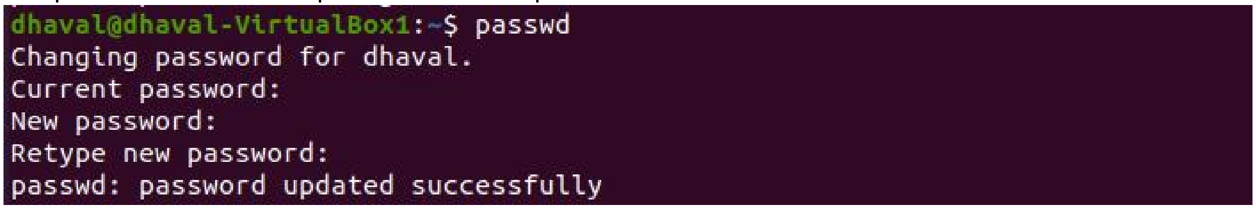
```
# m h  dom mon dow   command
* * * * * /bin/date >> /tmp/cron_output
```

AN. mount: mounts filesystem towards



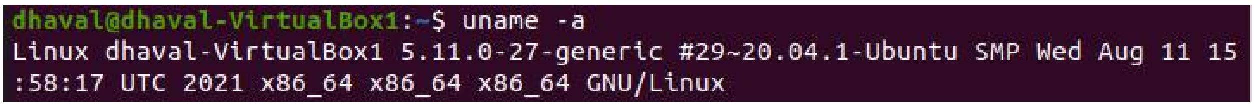
```
dhaval@dhaval-VirtualBox1:~$ sudo mount -t ext4
[sudo] password for dhaval:
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
```

AO. passwd: It is used to update the current password of the user.



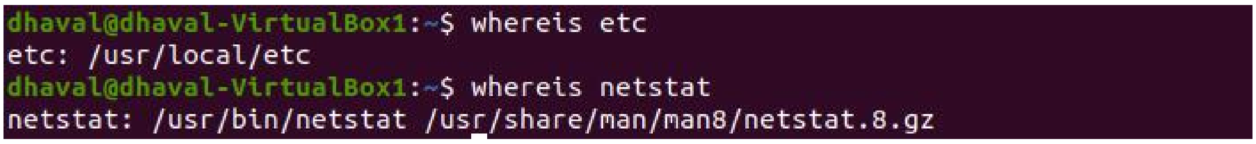
```
dhaval@dhaval-VirtualBox1:~$ passwd
Changing password for dhaval.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

AP. uname: It is used to print system information.



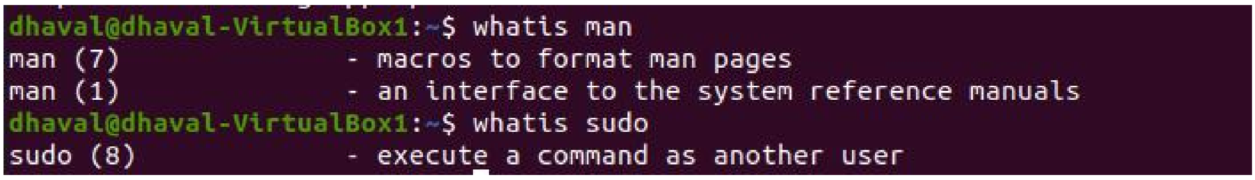
```
dhaval@dhaval-VirtualBox1:~$ uname -a
Linux dhaval-VirtualBox1 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15
:58:17 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

AQ. whereis: It is used to locate the binary and manual files of the command



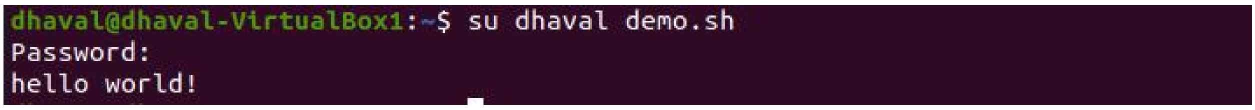
```
dhaval@dhaval-VirtualBox1:~$ whereis etc
etc: /usr/local/etc
dhaval@dhaval-VirtualBox1:~$ whereis netstat
netstat: /usr/bin/netstat /usr/share/man/man8/netstat.8.gz
```

AR. whatis:



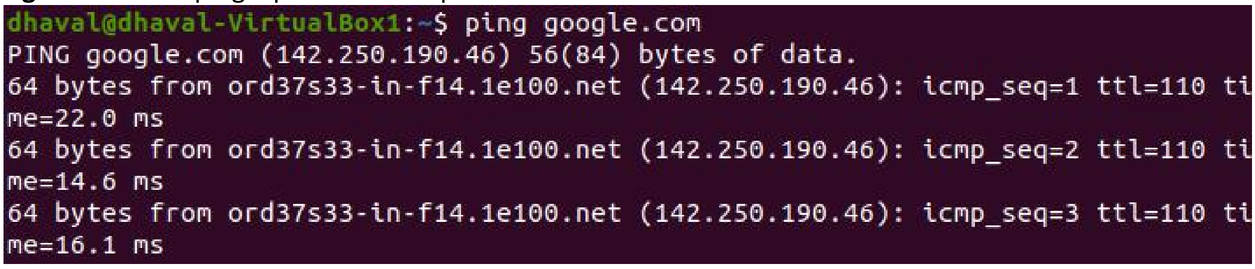
```
dhaval@dhaval-VirtualBox1:~$ whatis man
man (7)              - macros to format man pages
man (1)              - an interface to the system reference manuals
dhaval@dhaval-VirtualBox1:~$ whatis sudo
sudo (8)             - execute a command as another user
```

AS. su: it lets you execute command as a different user.



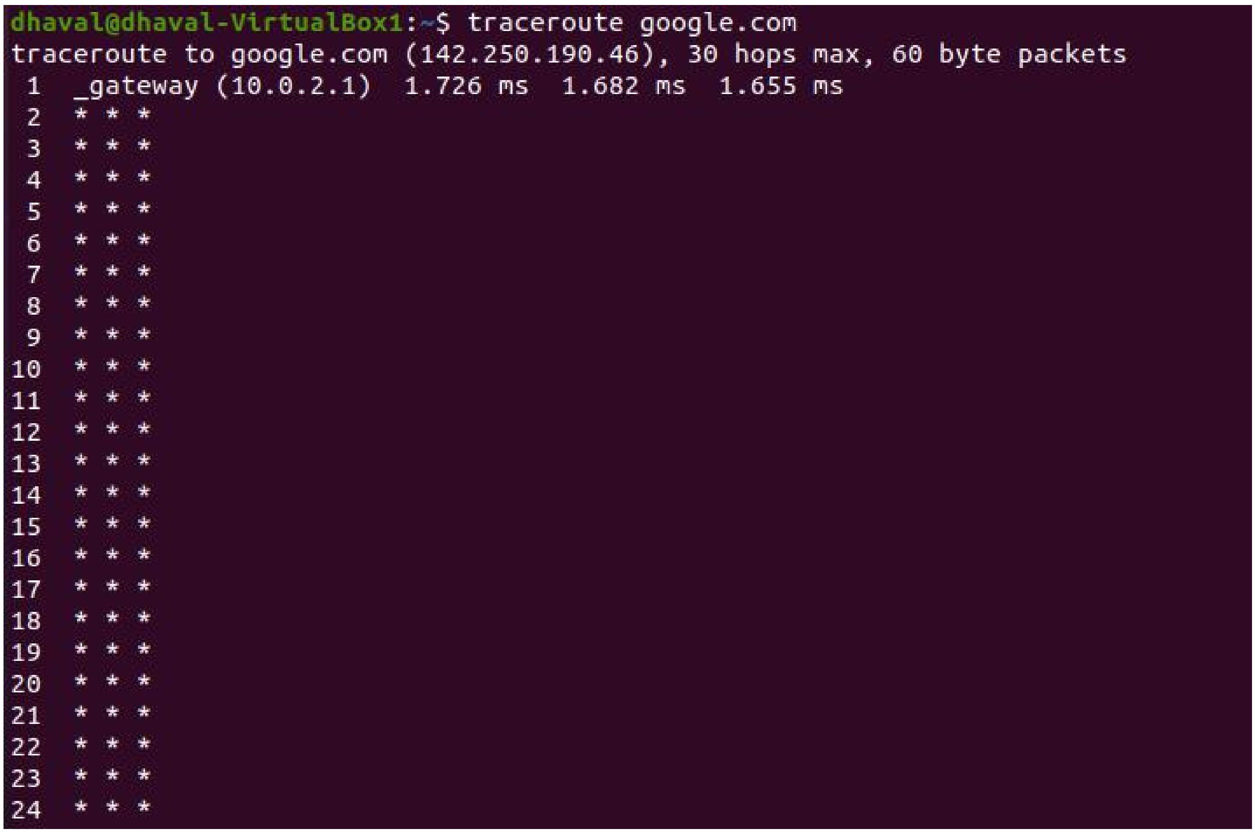
```
dhaval@dhaval-VirtualBox1:~$ su dhaval demo.sh
Password:
hello world!
```

AT. ping: it is used to ping a particular endpoint or a url.



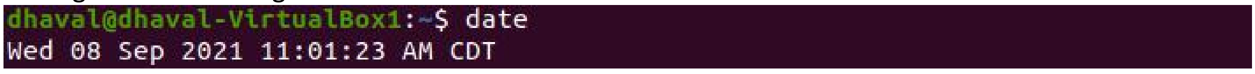
```
dhaval@dhaval-VirtualBox1:~$ ping google.com
PING google.com (142.250.190.46) 56(84) bytes of data.
64 bytes from ord37s33-in-f14.1e100.net (142.250.190.46): icmp_seq=1 ttl=110 ti
me=22.0 ms
64 bytes from ord37s33-in-f14.1e100.net (142.250.190.46): icmp_seq=2 ttl=110 ti
me=14.6 ms
64 bytes from ord37s33-in-f14.1e100.net (142.250.190.46): icmp_seq=3 ttl=110 ti
me=16.1 ms
```

AU. traceroute: this command is used to trace the route of the packet from source to destination.



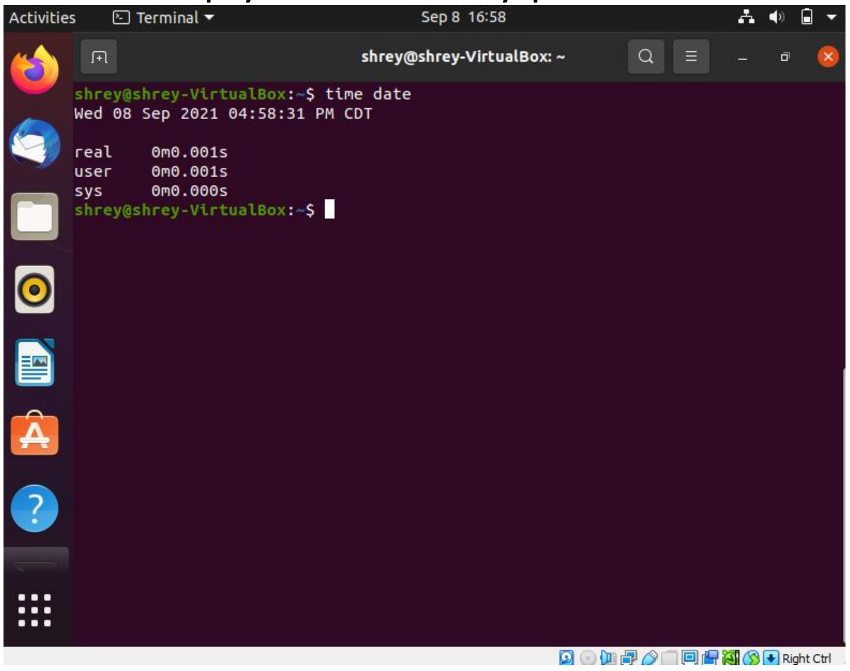
```
dhaval@dhaval-VirtualBox1:~$ traceroute google.com
traceroute to google.com (142.250.190.46), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.1)  1.726 ms  1.682 ms  1.655 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
```

AV.date: This command is used to display today's date, day and time. We can use different arguments to change the format as well.



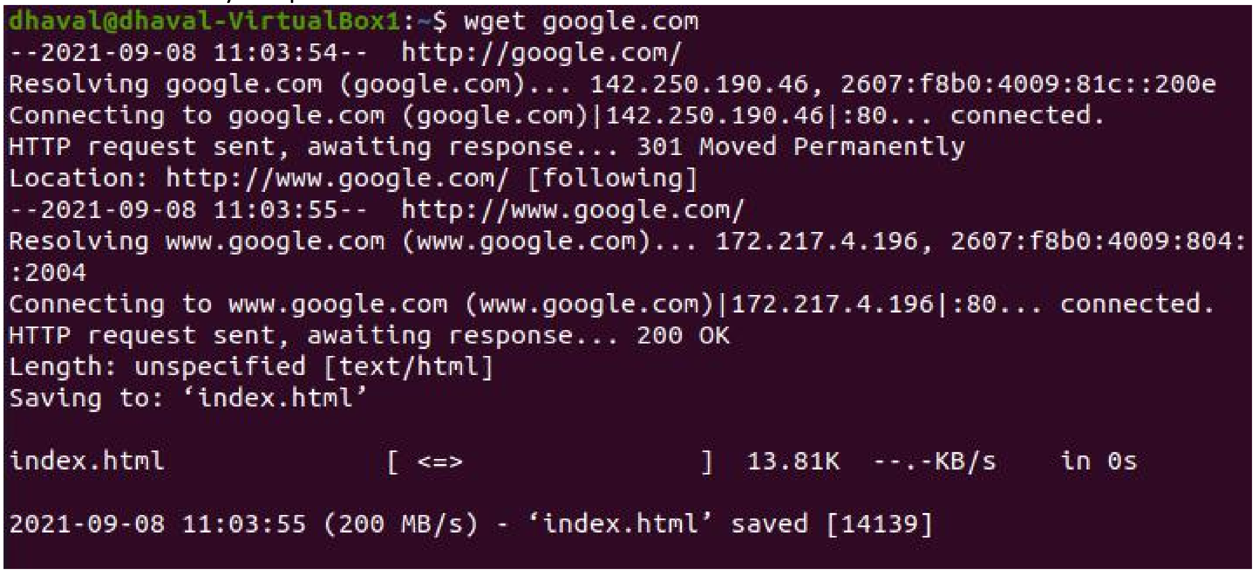
```
dhaval@dhaval-VirtualBox1:~$ date
Wed 08 Sep 2021 11:01:23 AM CDT
```

AW. time: used to display date and time in any specified format.



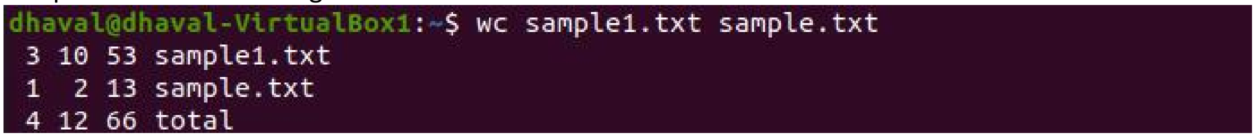
```
shrey@shrey-VirtualBox:~$ time date
Wed 08 Sep 2021 04:58:31 PM CDT

real    0m0.001s
user    0m0.001s
sys     0m0.000s
shrey@shrey-VirtualBox:~$
```

AX. wget: this command it used to perform a non-interactive download of a file from a url. Also we can hit any endpoint url to browse it.



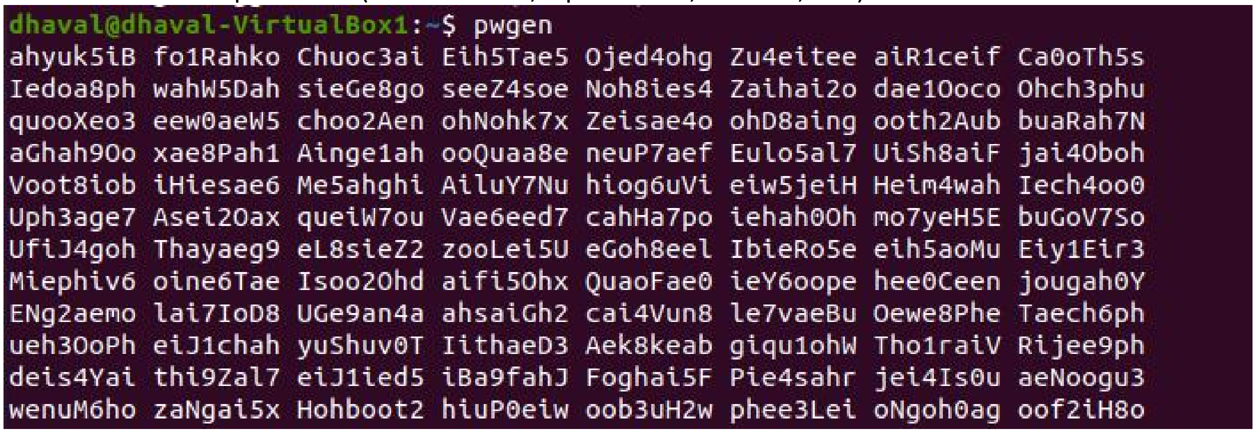
```
dhaval@dhaval-VirtualBox1:~$ wget google.com
--2021-09-08 11:03:54--  http://google.com/
Resolving google.com (google.com)... 142.250.190.46, 2607:f8b0:4009:81c::200e
Connecting to google.com (google.com)|142.250.190.46|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2021-09-08 11:03:55--  http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.4.196, 2607:f8b0:4009:804:
:2004
Connecting to www.google.com (www.google.com)|172.217.4.196|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘index.html’

index.html              [ <=>                ]  13.81K  --.-KB/s    in 0s

2021-09-08 11:03:55 (200 MB/s) - ‘index.html’ saved [14139]
```

AY. wc: It is used to find the number of lines, word count, character and byte count of all the specified files in the argument.



```
dhaval@dhaval-VirtualBox1:~$ wc sample1.txt sample.txt
 3 10 53 sample1.txt
 1  2 13 sample.txt
 4 12 66 total
```

AZ. pwgen: It is used to generate random passwords. We can use different arguments to customize the passwords(i.e. numerical, alphabetical, secured, etc.)

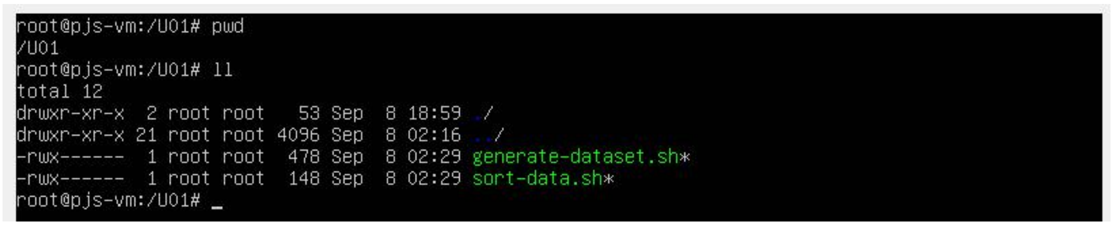


```
dhaval@dhaval-VirtualBox1:~$ pwgen
ahyuk5iB fo1Rahko Chuoc3ai Eih5Tae5 Ojed4ohg Zu4eitee aiR1ceif Ca0oTh5s
Iedoa8ph wahW5Dah sieGe8go seeZ4soe Noh8ies4 Zaihai2o dae1Ooco Ohch3phu
quooXeo3 eew0aeW5 choo2Aen ohNohk7x Zeisae4o ohD8aing ooth2Aub buaRah7N
aGhah9Oo xae8Pah1 Ainge1ah ooQuaa8e neuP7aef Eulo5al7 UiSh8aiF jai4Oboh
Voot8iob iHiesae6 Me5ahghi AiluY7Nu hiog6uVi eiw5jeiH Heim4wah Iech4oo0
Uph3age7 Asei2Oax queiW7ou Vae6eed7 cahHa7po iehah0Oh mo7yeH5E buGoV7So
UfiJ4goh Thayaeg9 eL8sieZ2 zooLei5U eGoh8eel IbieRo5e eih5aoMu Eiy1Eir3
Miephiv6 oine6Tae Isoo2Ohd aifi5Ohx QuaoFae0 ieY6oope hee0Ceen jougah0Y
ENg2aemo lai7IoD8 UGe9an4a ahsaiGh2 cai4Vun8 le7vaeBu Oewe8Phe Taech6ph
ueh3OoPh eiJ1chah yuShuv0T IithaeD3 Aek8keab giqu1ohW Tho1raiV Rijee9ph
deis4Yai thi9Zal7 eiJ1ied5 iBa9fahJ Foghai5F Pie4sahr jei4Is0u aeNoogu3
wenuM6ho zaNgai5x Hohboot2 hiuP0eiw oob3uH2w phee3Lei oNgoh0ag oof2iH8o
```

Question 3:

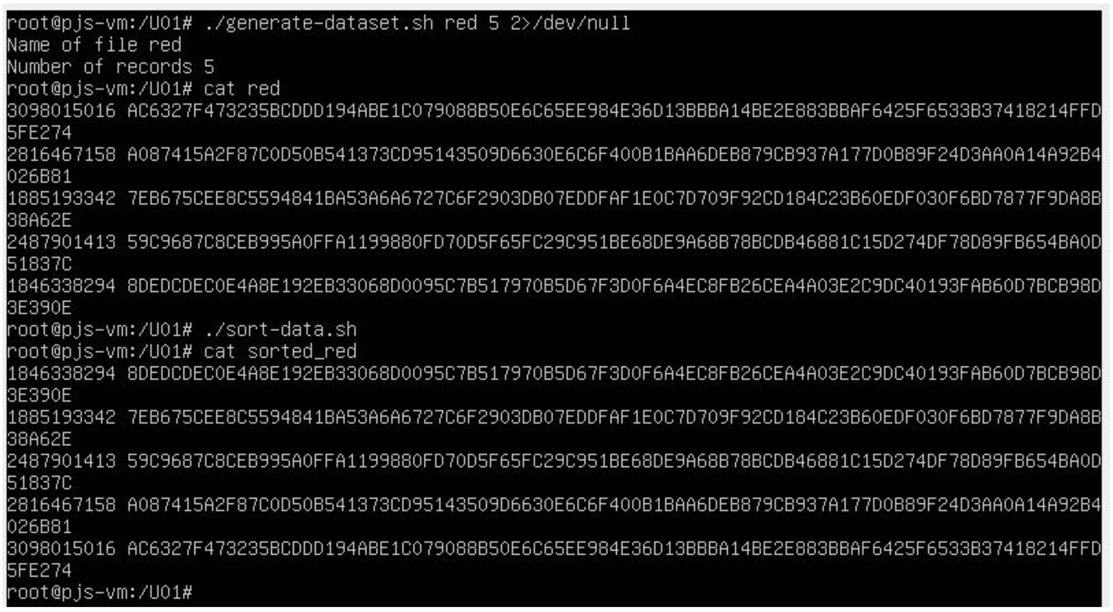
The first 2 screenshots are just the working of the bash scripts and does not answer any of Q.3 a-e.

Screenshot 1-



```
root@pjs-vm:/U01# pwd
/U01
root@pjs-vm:/U01# ll
total 12
drwxr-xr-x  2 root root   53 Sep  8 18:59 ./
drwxr-xr-x 21 root root 4096 Sep  8 02:16 ../
-rwx------  1 root root  478 Sep  8 02:29 generate-dataset.sh*
-rwx------  1 root root  148 Sep  8 02:29 sort-data.sh*
root@pjs-vm:/U01# _
```

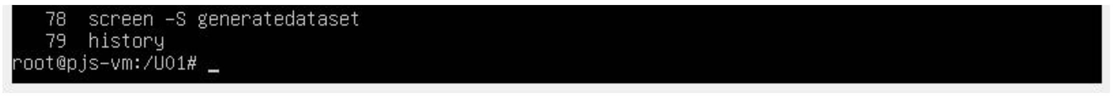
Screenshot 2-
Here, “red” is a sample dataset generated to show the output of sorting algorithm.



```
root@pjs-vm:/U01# ./generate-dataset.sh red 5 2>/dev/null
Name of file red
Number of records 5
root@pjs-vm:/U01# cat red
3098015016 AC6327F473235BCDDD194ABE1C079088B50E6C65EE984E36D13BBBA14BE2E883BBAF6425F6533B37418214FFD
5FE274
2816467158 A087415A2F87C0D50B541373CD95143509D6630E6C6F400B1BAA6DEB879CB937A177D0B89F24D3AA0A14A92B4
026B81
1885193342 7EB675CEE8C5594841BA53A6A6727C6F2903DB07EDDFAF1E0C7D709F92CD184C23B60EDF030F6BD7877F9DA8B
38A62E
2487901413 59C9687C8CEB995A0FFA1199880FD70D5F65FC29C951BE68DE9A68B78BCDB46881C15D274DF78D89FB654BA0D
51837C
1846338294 8DEDCDEC0E4A8E192EB33068D0095C7B517970B5D67F3D0F6A4EC8FB26CEA4A03E2C9DC40193FAB60D7BCB98D
3E390E
root@pjs-vm:/U01# ./sort-data.sh
root@pjs-vm:/U01# cat sorted_red
1846338294 8DEDCDEC0E4A8E192EB33068D0095C7B517970B5D67F3D0F6A4EC8FB26CEA4A03E2C9DC40193FAB60D7BCB98D
3E390E
1885193342 7EB675CEE8C5594841BA53A6A6727C6F2903DB07EDDFAF1E0C7D709F92CD184C23B60EDF030F6BD7877F9DA8B
38A62E
2487901413 59C9687C8CEB995A0FFA1199880FD70D5F65FC29C951BE68DE9A68B78BCDB46881C15D274DF78D89FB654BA0D
51837C
2816467158 A087415A2F87C0D50B541373CD95143509D6630E6C6F400B1BAA6DEB879CB937A177D0B89F24D3AA0A14A92B4
026B81
3098015016 AC6327F473235BCDDD194ABE1C079088B50E6C65EE984E36D13BBBA14BE2E883BBAF6425F6533B37418214FFD
5FE274
root@pjs-vm:/U01#
```

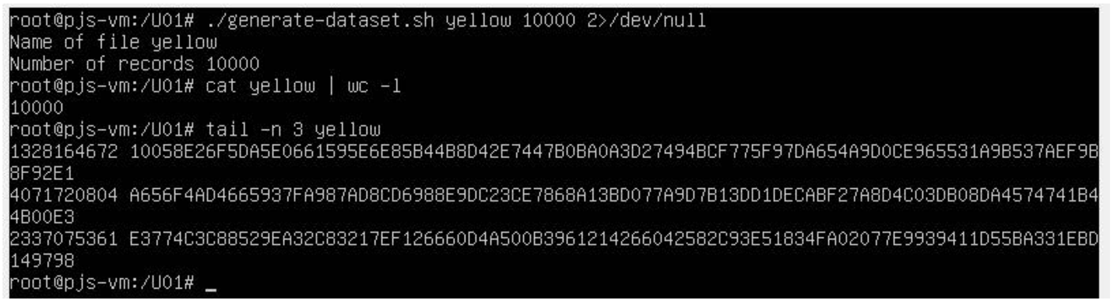
Question 3.a:

Creating a dataset named “yellow” with 10,000 records. The script would be run using the “screen” command in linux so that the script would complete even if the ssh/bash session is abruptly terminated.



```
   78  screen -S generatedataset
   79  history
root@pjs-vm:/U01# _
```

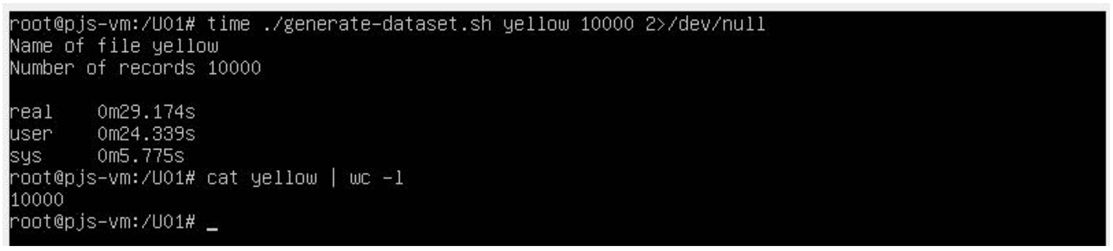
1st Column is a collection of random number of 32-bit integer.
2nd Column is a collection of ASCII strings of 95-bytes long.
The script takes 2 arguments (filename and number of records) as inputs during the start of execution.



```
root@pjs-vm:/U01# ./generate-dataset.sh yellow 10000 2>/dev/null
Name of file yellow
Number of records 10000
root@pjs-vm:/U01# cat yellow | wc -l
10000
root@pjs-vm:/U01# tail -n 3 yellow
1328164672 10058E26F5DA5E0661595E6E85B44B8D42E7447B0BA0A3D27494BCF775F97DA654A9D0CE965531A9B537AEF9B
8F92E1
4071720804 A656F4AD4665937FA987AD8CD6988E9DC23CE7868A13BD077A9D7B13DD1DECABF27A8D4C03DB08DA4574741B4
4B00E3
2337075361 E3774C3C88529EA32C83217EF126660D4A500B3961214266042582C93E51834FA02077E9939411D55BA331EBD
149798
root@pjs-vm:/U01# _
```

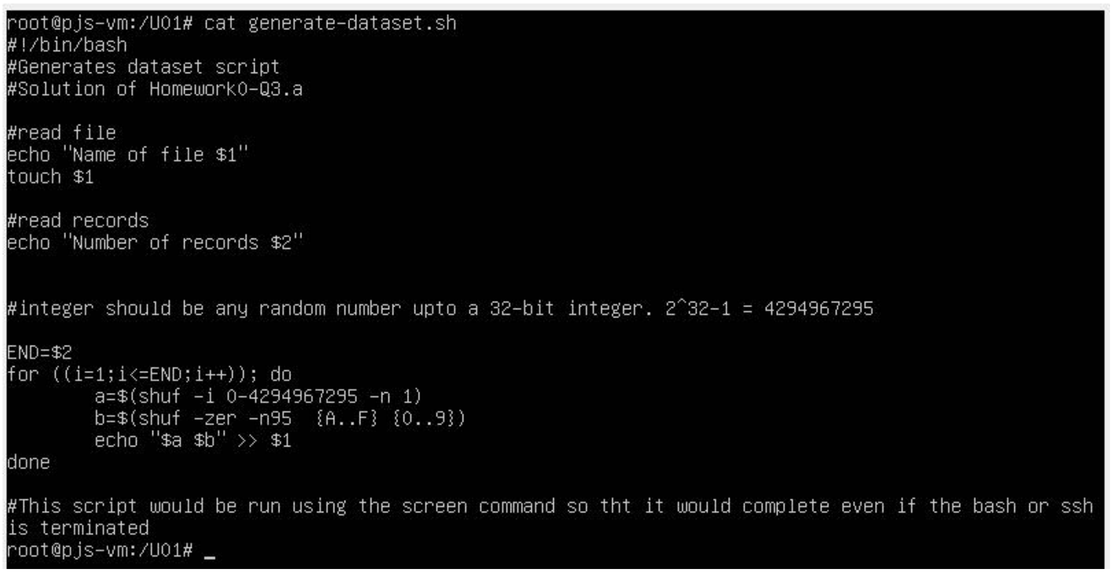
The number of records is 10,000 as indicated by “wc” command in the below screenshot.

The output of “time” command below shows the exact time taken for execution.
Real time - 29.174s
User time - 24.339s
System time - 5.775s



```
root@pjs-vm:/U01# time ./generate-dataset.sh yellow 10000 2>/dev/null
Name of file yellow
Number of records 10000

real    0m29.174s
user    0m24.339s
sys     0m5.775s
root@pjs-vm:/U01# cat yellow | wc -l
10000
root@pjs-vm:/U01# _
```

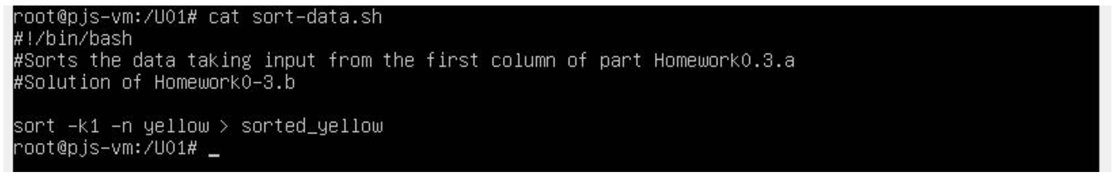
The script of generate-dataset.sh :



```
root@pjs-vm:/U01# cat generate-dataset.sh
#!/bin/bash
#Generates dataset script
#Solution of Homework0-Q3.a

#read file
echo "Name of file $1"
touch $1

#read records
echo "Number of records $2"


#integer should be any random number upto a 32-bit integer. 2^32-1 = 4294967295

END=$2
for ((i=1;i<=END;i++)); do
        a=$(shuf -i 0-4294967295 -n 1)
        b=$(shuf -zer -n95  {A..F} {0..9})
        echo "$a $b" >> $1
done

#This script would be run using the screen command so tht it would complete even if the bash or ssh
is terminated
root@pjs-vm:/U01# _
```

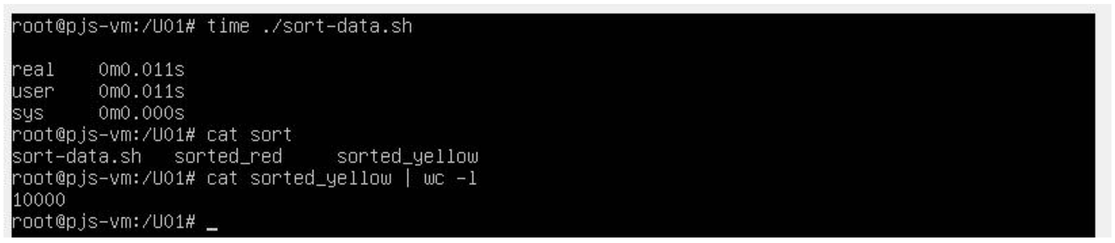
Question 3.b:

Script for sort-data.sh :



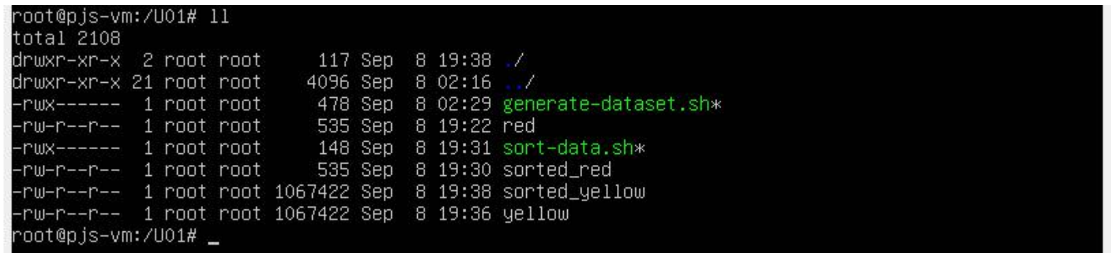
```
root@pjs-vm:/U01# cat sort-data.sh
#!/bin/bash
#Sorts the data taking input from the first column of part Homework0.3.a
#Solution of Homework0-3.b

sort -k1 -n yellow > sorted_yellow
root@pjs-vm:/U01# _
```

The output of “time” command below shows the exact time taken for execution.
Real time - 0.011s
User time - 0.0119s
System time - 0.000s



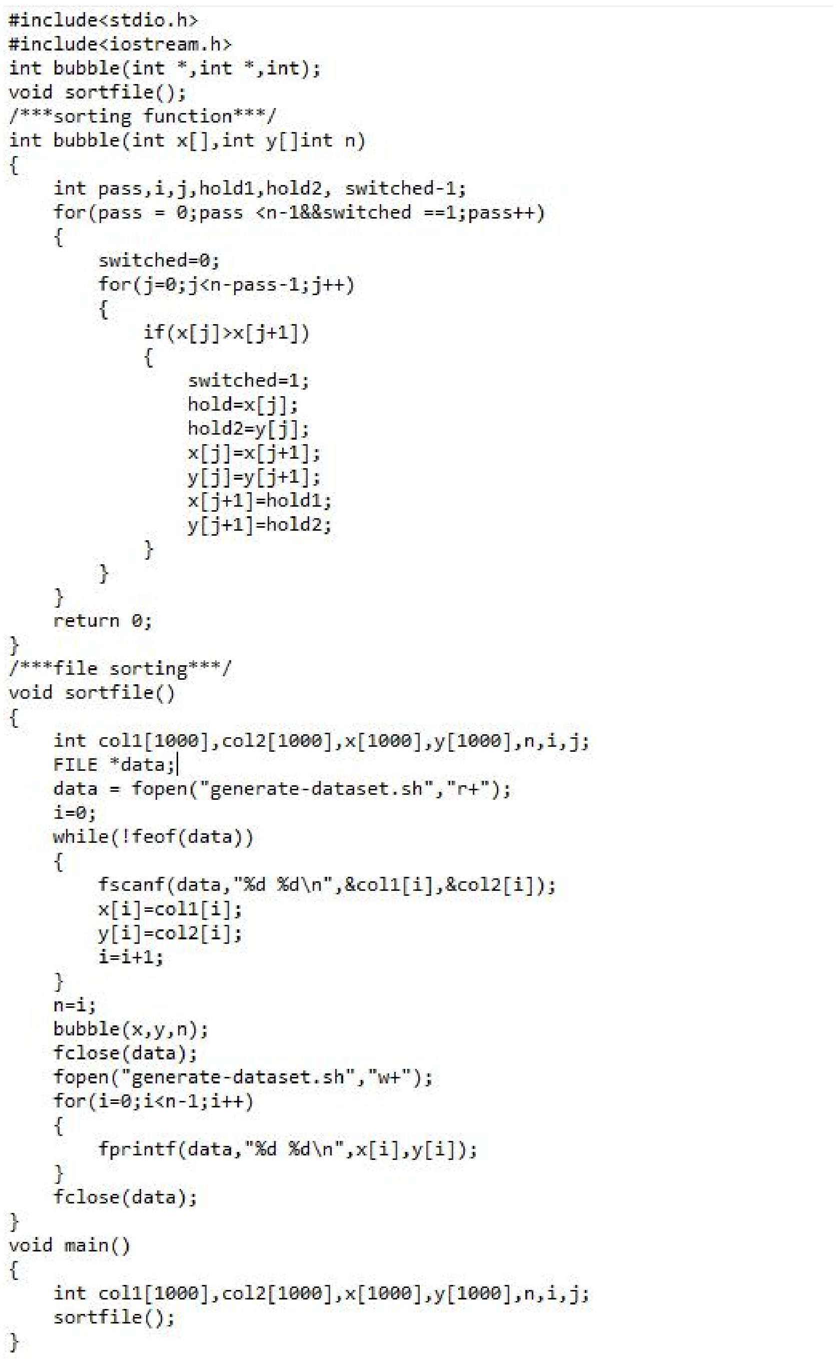
```
root@pjs-vm:/U01# time ./sort-data.sh

real    0m0.011s
user    0m0.011s
sys     0m0.000s
root@pjs-vm:/U01# cat sort
sort-data.sh   sorted_red     sorted_yellow
root@pjs-vm:/U01# cat sorted_yellow | wc -l
10000
root@pjs-vm:/U01# _
```

Files generated after Question 3.a and 3.b:



```
root@pjs-vm:/U01# ll
total 2108
drwxr-xr-x  2 root root     117 Sep  8 19:38 ./
drwxr-xr-x 21 root root    4096 Sep  8 02:16 ../
-rwx------  1 root root     478 Sep  8 02:29 generate-dataset.sh*
-rw-r--r--  1 root root     535 Sep  8 19:22 red
-rwx------  1 root root     148 Sep  8 19:31 sort-data.sh*
-rw-r--r--  1 root root     535 Sep  8 19:30 sorted_red
-rw-r--r--  1 root root 1067422 Sep  8 19:38 sorted_yellow
-rw-r--r--  1 root root 1067422 Sep  8 19:36 yellow
root@pjs-vm:/U01# _
```

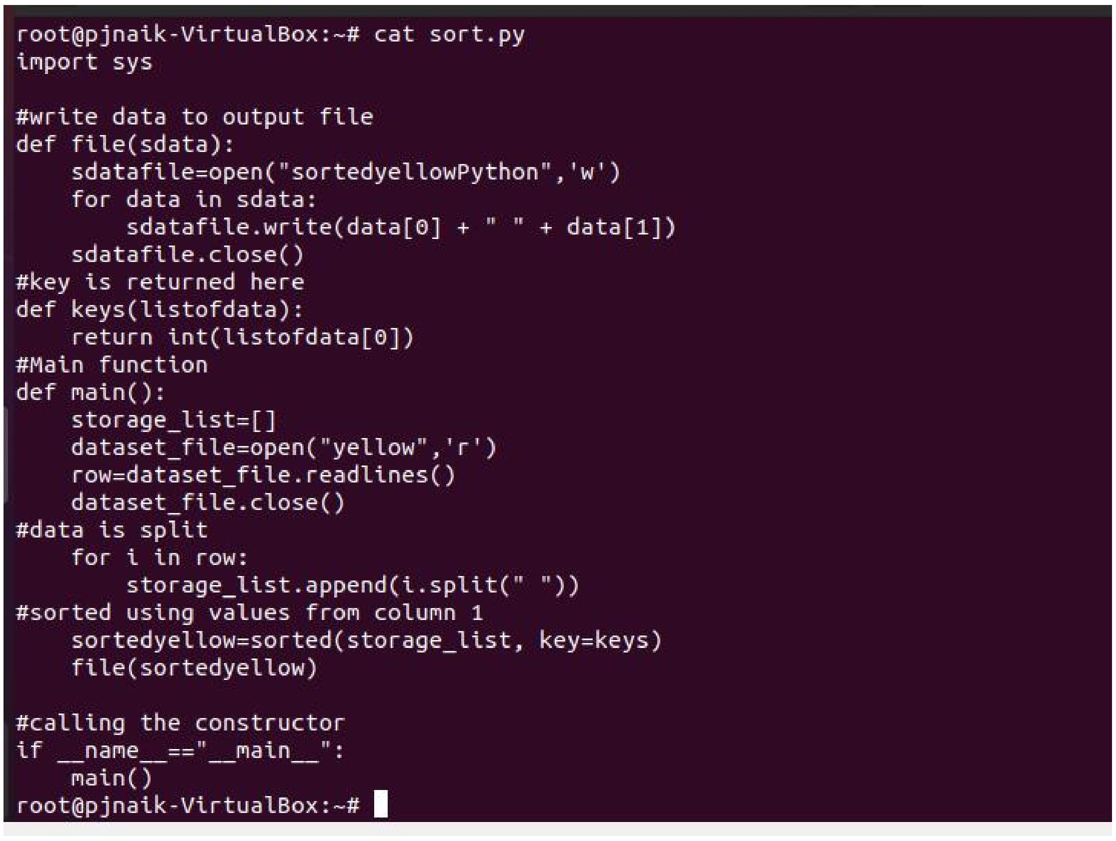
Question 3.c:



```
#include<stdio.h>
#include<iostream.h>
int bubble(int *,int *,int);
void sortfile();
/***sorting function***/
int bubble(int x[],int y[]int n)
{
    int pass,i,j,hold1,hold2, switched-1;
    for(pass = 0;pass <n-1&&switched ==1;pass++)
    {
        switched=0;
        for(j=0;j<n-pass-1;j++)
        {
            if(x[j]>x[j+1])
            {
                switched=1;
                hold=x[j];
                hold2=y[j];
                x[j]=x[j+1];
                y[j]=y[j+1];
                x[j+1]=hold1;
                y[j+1]=hold2;
            }
        }
    }
    return 0;
}
/***file sorting***/
void sortfile()
{
    int col1[1000],col2[1000],x[1000],y[1000],n,i,j;
    FILE *data;
    data = fopen("generate-dataset.sh","r+");
    i=0;
    while(!feof(data))
    {
        fscanf(data,"%d %d\n",&col1[i],&col2[i]);
        x[i]=col1[i];
        y[i]=col2[i];
        i=i+1;
    }
    n=i;
    bubble(x,y,n);
    fclose(data);
    fopen("generate-dataset.sh","w+");
    for(i=0;i<n-1;i++)
    {
        fprintf(data,"%d %d\n",x[i],y[i]);
    }
    fclose(data);
}
void main()
{
    int col1[1000],col2[1000],x[1000],y[1000],n,i,j;
    sortfile();
}
```

Question 3.d:

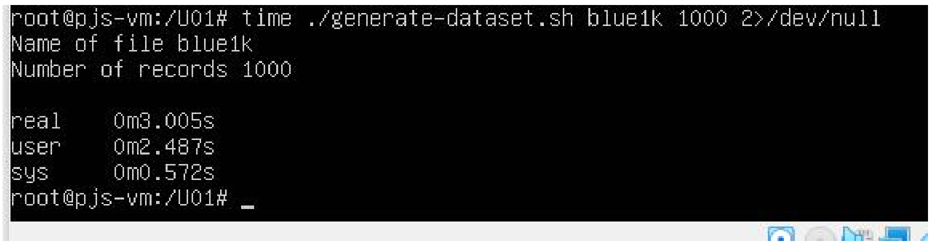
Script for sorting data using Python Programming language



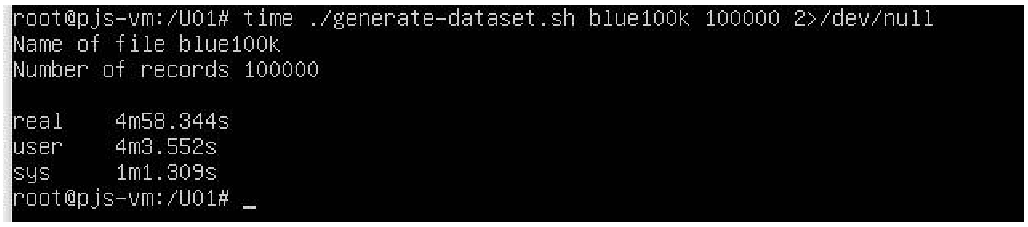
```
root@pjnaik-VirtualBox:~# cat sort.py
import sys

#write data to output file
def file(sdata):
    sdatafile=open("sortedyellowPython",'w')
    for data in sdata:
        sdatafile.write(data[0] + " " + data[1])
    sdatafile.close()
#key is returned here
def keys(listofdata):
    return int(listofdata[0])
#Main function
def main():
    storage_list=[]
    dataset_file=open("yellow",'r')
    row=dataset_file.readlines()
    dataset_file.close()
#data is split
    for i in row:
        storage_list.append(i.split(" "))
#sorted using values from column 1
    sortedyellow=sorted(storage_list, key=keys)
    file(sortedyellow)

#calling the constructor
if __name__=="__main__":
    main()
root@pjnaik-VirtualBox:~# 
```

Question 3.e:

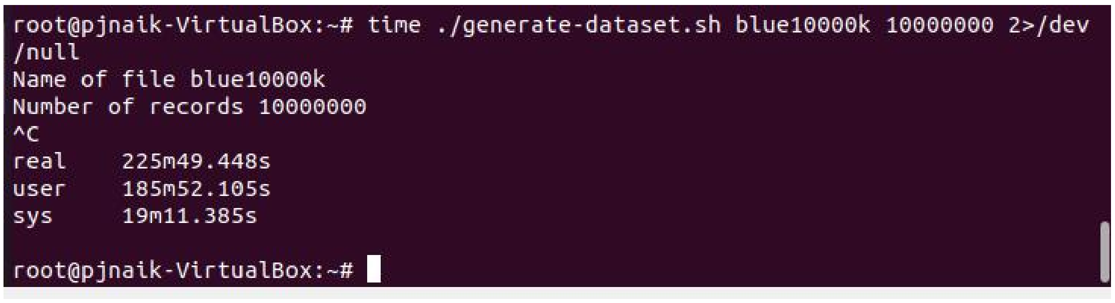
Time taken to generate 1000, 100000 and 10000000 records using bash script -



```
root@pjs-vm:/U01# time ./generate-dataset.sh blue1k 1000 2>/dev/null
Name of file blue1k
Number of records 1000

real    0m3.005s
user    0m2.487s
sys     0m0.572s
root@pjs-vm:/U01# _
```



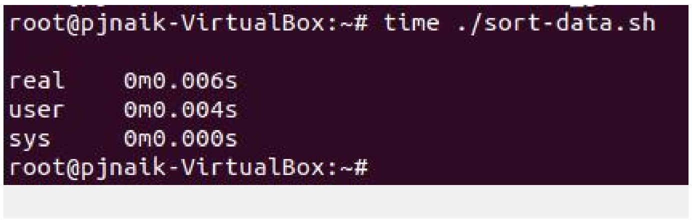
```
root@pjs-vm:/U01# time ./generate-dataset.sh blue100k 100000 2>/dev/null
Name of file blue100k
Number of records 100000

real    4m58.344s
user    4m3.552s
sys     1m1.309s
root@pjs-vm:/U01# _
```



```
root@pjnaik-VirtualBox:~# time ./generate-dataset.sh blue10000k 10000000 2>/dev
/null
Name of file blue10000k
Number of records 10000000
^C
real    225m49.448s
user    185m52.105s
sys     19m11.385s

root@pjnaik-VirtualBox:~#
```

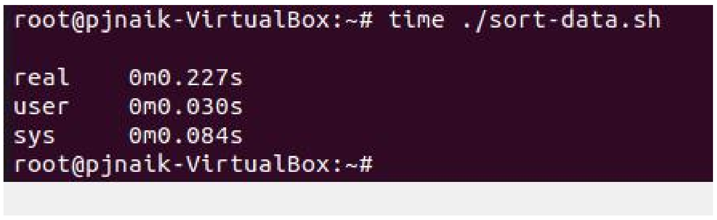
Time taken to sort the data using shell script

For 1000 rows -



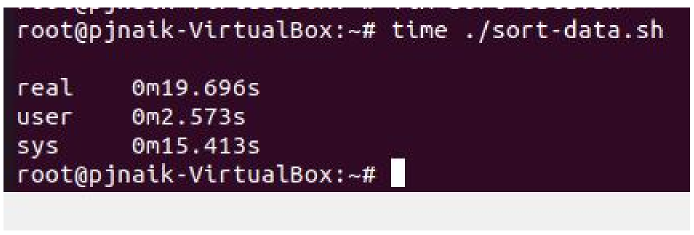
```
root@pjnaik-VirtualBox:~# time ./sort-data.sh

real    0m0.006s
user    0m0.004s
sys     0m0.000s
root@pjnaik-VirtualBox:~#
```

For 100000 rows -



```
root@pjnaik-VirtualBox:~# time ./sort-data.sh

real    0m0.227s
user    0m0.030s
sys     0m0.084s
root@pjnaik-VirtualBox:~#
```

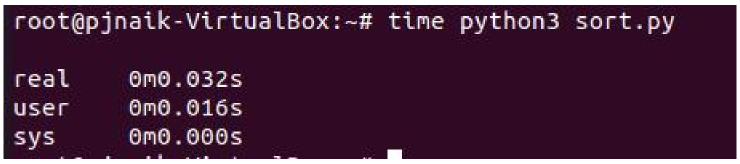
Time taken for 10000000 rows -



```
root@pjnaik-VirtualBox:~# time ./sort-data.sh

real	0m19.696s
user	0m2.573s
sys	0m15.413s
root@pjnaik-VirtualBox:~# 
```

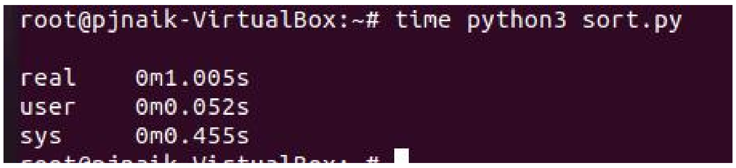
Time taken to sort the data using python script-

For 1000 rows-



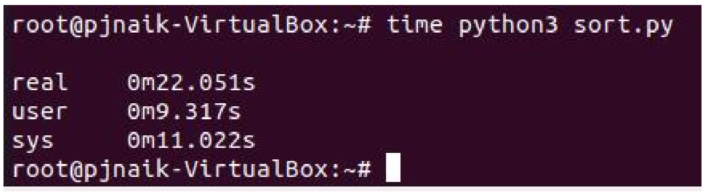
```
root@pjnaik-VirtualBox:~# time python3 sort.py

real	0m0.032s
user	0m0.016s
sys	0m0.000s
```

For 100000 rows -



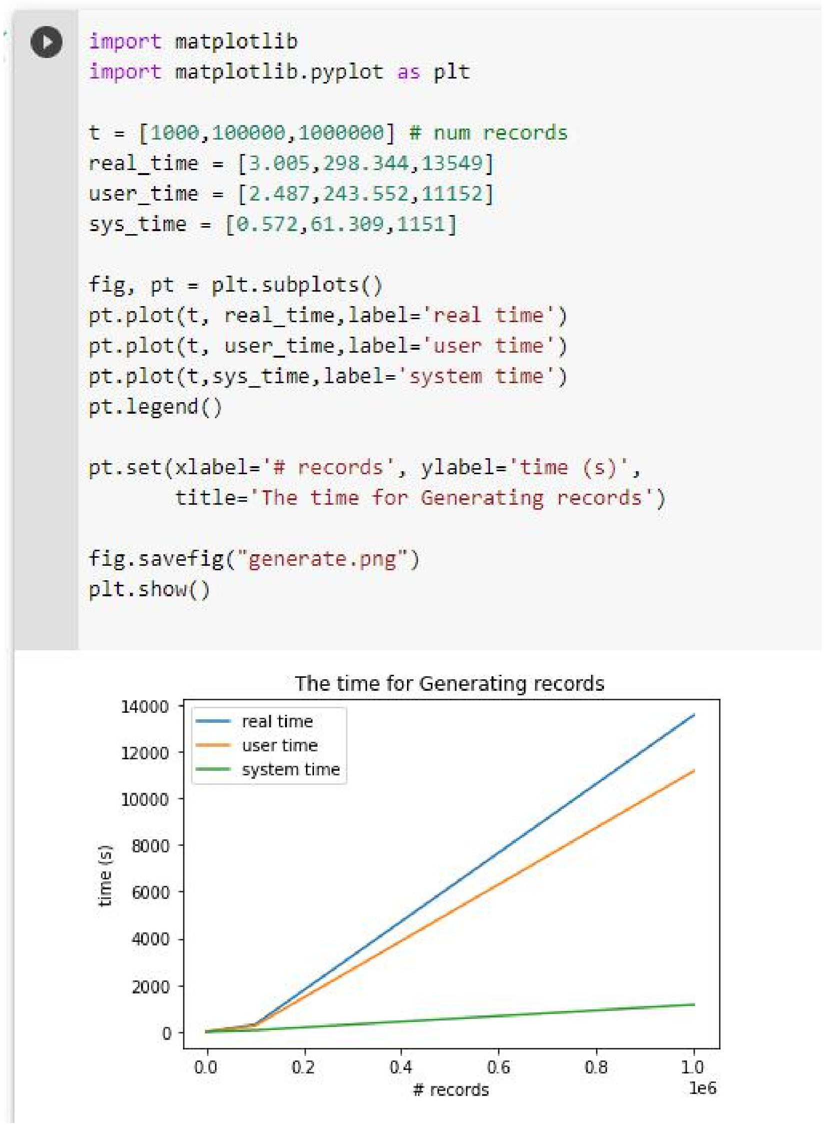
```
root@pjnaik-VirtualBox:~# time python3 sort.py

real	0m1.005s
user	0m0.052s
sys	0m0.455s
```

For 10000000 rows-



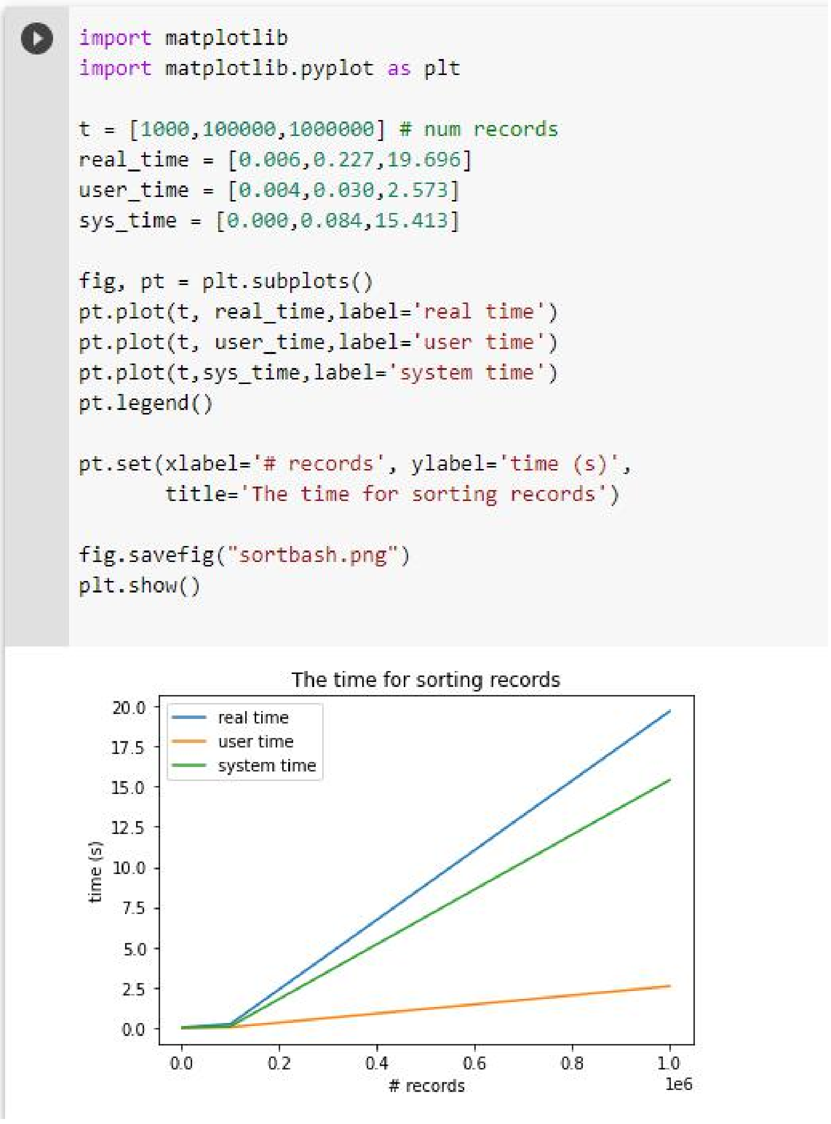
```
root@pjnaik-VirtualBox:~# time python3 sort.py

real	0m22.051s
user	0m9.317s
sys	0m11.022s
root@pjnaik-VirtualBox:~# 
```

Graph of generating records vs Time



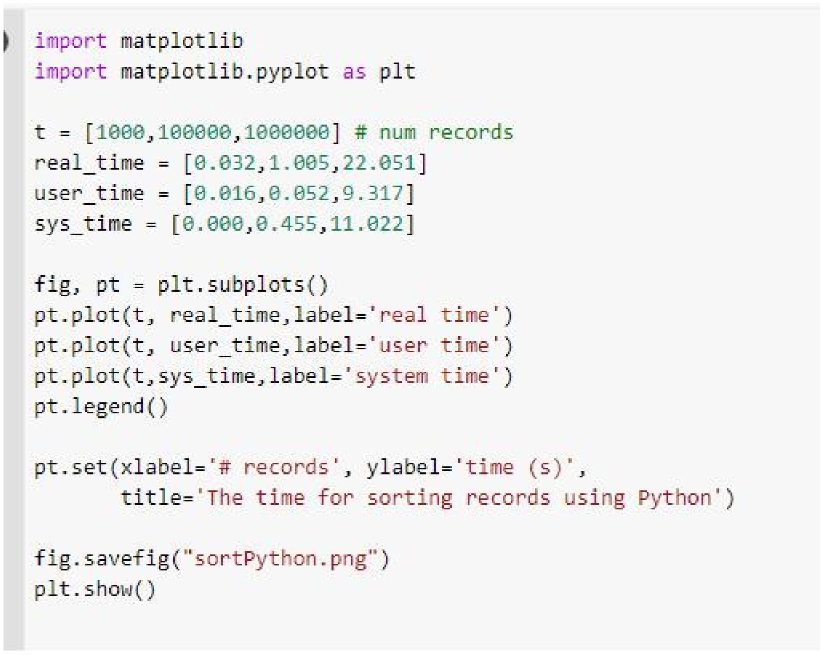
```
import matplotlib
import matplotlib.pyplot as plt

t = [1000,100000,1000000] # num records
real_time = [3.005,298.344,13549]
user_time = [2.487,243.552,11152]
sys_time = [0.572,61.309,1151]

fig, pt = plt.subplots()
pt.plot(t, real_time,label='real time')
pt.plot(t, user_time,label='user time')
pt.plot(t,sys_time,label='system time')
pt.legend()

pt.set(xlabel='# records', ylabel='time (s)',
       title='The time for Generating records')

fig.savefig("generate.png")
plt.show()
```

Graph of sorting records (using bash) vs Time

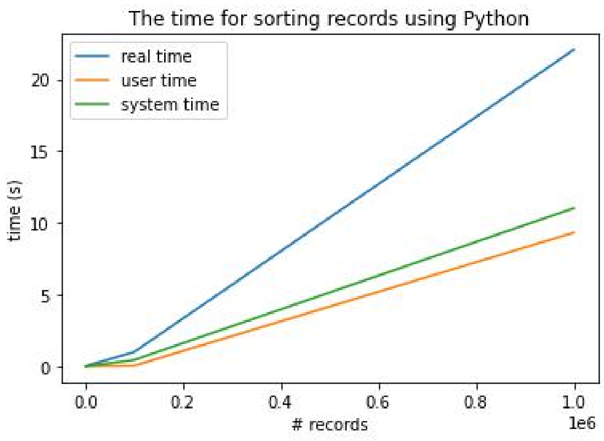


```
import matplotlib
import matplotlib.pyplot as plt

t = [1000,100000,1000000] # num records
real_time = [0.006,0.227,19.696]
user_time = [0.004,0.030,2.573]
sys_time = [0.000,0.084,15.413]

fig, pt = plt.subplots()
pt.plot(t, real_time,label='real time')
pt.plot(t, user_time,label='user time')
pt.plot(t,sys_time,label='system time')
pt.legend()

pt.set(xlabel='# records', ylabel='time (s)',
       title='The time for sorting records')

fig.savefig("sortbash.png")
plt.show()
```

Graph of sorting records (using Python) vs Time



```
import matplotlib
import matplotlib.pyplot as plt

t = [1000,100000,1000000] # num records
real_time = [0.032,1.005,22.051]
user_time = [0.016,0.052,9.317]
sys_time = [0.000,0.455,11.022]

fig, pt = plt.subplots()
pt.plot(t, real_time,label='real time')
pt.plot(t, user_time,label='user time')
pt.plot(t,sys_time,label='system time')
pt.legend()

pt.set(xlabel='# records', ylabel='time (s)',
       title='The time for sorting records using Python')

fig.savefig("sortPython.png")
plt.show()
```



Question 4:

A.In the system configuration of the VM, explain how changing the number of processors changes the behaviour of your VM. Explain a scenario where you want to set this to the minimum, and a scenario where you want to set it to the maximum. Why is setting it to the maximum potentially a bad idea?

A Virtual Machine is a software that allows you to emulate another computer without having a physical second computer, like an emulator. Example, you can use a virtual machine on your Microsoft Windows based computer to emulate an Ubuntu Linux based machine on your computer which gives you the workspace for a Linux based system. Changing the number of processors affects the VM drastically. It is best to use the minimum number of processors for basic functionalities such as using the terminal for some commands, browsing the internet, etc. Single processors work faster for such tasks as you do not have to wait for the task to be completed before the other processors can execute their jobs. Maximum number of processors can be useful for high intensity or high workload tasks such as a benchmarking tool or compression or encryption.
It is potentially a bad idea to allocate the maximum number of processors due to the wastage of CPU processing power, its is likely to degrade the virtual machine, it may cause the host system to become unresponsive in some cases.

B. In the system configuration of the VM, under the Acceleration Tab, explain the difference between the paravirtualization options: None, Legacy, Minimal, Hyper-V, and KVM. Explain which one would be best to use with Ubuntu Linux, and why.

Paravirtualization is a virtualization technique that give a software interface to virtual machines which is similar to the underlying hardware-software interface. The guest OS has to be explicitly ported before installing a VM otherwise the OS will not be able to run on top of a virtual machine monitor.
Difference between paravirtualization options is as follows,

- None: Specifying 'none' explicitly turns off exposing any paravirtualization interface.
- Legacy: Virtual Machines created on older versions of Virtual Box and use paravirtualization in any version after VirtualBox 5.0.
- Minimal: It announces the presence of a virtual environment. It is mandatory for running MAC OS X guests. Additionally, it reports the frequency of the time stamp counter and the advanced programmable interrupt controller to the guest OS.
- Hyper-V: It is recommended for windows guests. It presents a Microsoft Hyper-V hypervisor interface which is recognized by Windows 7 and newer operating systems. VirtualBox's implementation currently supports paravirtualized clocks, APIC frequency reporting, guest debugging, guest crash reporting and relaxed timer checks.
- KVM: It presents a Linux Kernel based Virtual Machine which is recognized by Linux Kernels after version 2.6.25. VirtualBox's implementation currently supports paravirtualized clocks and SMP clocks. It is recommended for Linux guests.

In conclusion, KVM is the best option of Ubuntu Linux due to the presence of The Linux KVM hypervisor, which is recognized by the Linux Kernel.

C. In storage devices when configuring the VM, there are multiple types of storage controllers: explain the difference between the IDE, SATA, and NVMe controller. Give an example for each type of storage controller of a scenario where you may want to use this type of controller

It is the job of a storage controller to connect the virtual machine to some kind of disk. It can either be a passthrough or a virtual hard disk. In a VM, the IDE controller gives up to four storage devices that can be attached to the machine, with the first one always used for the virtual CD/DVD drive.
A SATA controller is a more recent standard than IDE that supports both higher speeds as well as more devices per controller and consumes fewer CPU resources than the IDE controller. This enables

the connection of up to 30 virtual hard disks instead of just three. For this reason, it is the default controller used by VirtualBox.
NVMe is used to connect non-volatile memory directly over the PCIe to obtain a better bandwidth than SATA controllers for SSDs. Virtual NVMe devices have reduced guest I/O processing overhead, which allows more VDI VMs per host and more transactions per minute.
Examples:
IDE is always available to the VM even if there is no support for SATA devices in the guest OS.
SATA would be used for faster speeds and to save on CPU resources, it is usually the default storage controller.
NVMe would be used for tasks that require very quick response times like real world trading.

D.In the network configuration of the VM, there are multiple types of network adapters: explain the difference between NAT, Bridged Adapter, Internal Network, and Host-only Network. Give an example for each type of network of a scenario where you may want to use this type of network.

NAT is the default networking mode for VirtualBox since it is the simplest way of accessing an external network from a virtual machine, it does not require any configuration on the host network or the guest system.
Bridged mode uses a device driver on the host system to filter data from the physical network adapter the VM is able to intercept data from the physical network and inject data into it, making it seem like a new network interface in the software.
Internal network lets the VM be connected to other VMs on the same host which connect to the same internal network.
Host-only network is a mix of bridged and internal networking. The VMs can talk to each other as if they were connected via a physical Ethernet switch. Just like internal networking, the physical network interface doesn’t have to be present and the VMs cannot talk to the external world.
Examples:
A NAT network would be used to conned to a web server on the internet by translating the internal address to the target’s public address.
Host-only would be useful for preconfigured virtual environments where one VM may be a server, another VM may be a database and they are interconnected to talk to each other.
Internal network would be used when we want to use the capabilities of a bridged network without the risk of packet sniffers that can intercept data sent/received. It should therefore be used when we want to communicate between 2 or more VMs on the same network on the same host.
Bridged network can be used to let our VM connect to the rest of the network using the host adapter.

E. Difference between USB 1.0, 2.0 and 3.0

Usb 1.0 transfers data at the speed of 12Mb/sec whereas Usb 2.0 provides speed of 480Mb/sec and Usb 3.0 can transfer data upto 5Gb/sec.

USB 1.0 is used in computer devices to connect with various input devices like mouse and keyboards. It has the color code of white.
USB 2.0 is used in mobile phones cables and Hard Drive cables as it provides speedy transfer to data. And has the color code of black.
USB 3.0 is used in some of the latest hard drive cables to provide high transfer rate. USB 3.0 is reverse compatible(i.e. it supports older version of USB as well.). It uses the color code of blue or red in case of some devices.

USB 1.0 and 2.0 both uses Unicode type of encoding method while USB 3.0 used 8b/10b type of encoding method.
USB 1.0 and 2.0 can transfer data towards a single direction at a time. Whereas USB 3.0 can transfer data towards both the direction at a given point of time.

USB 2.0 and 3.0 requires power of 5V, 1.8A whereas USB 1.0 requires power of 5V, 1.5A.