

# Advanced Operating Systems

## Written Assignment 1

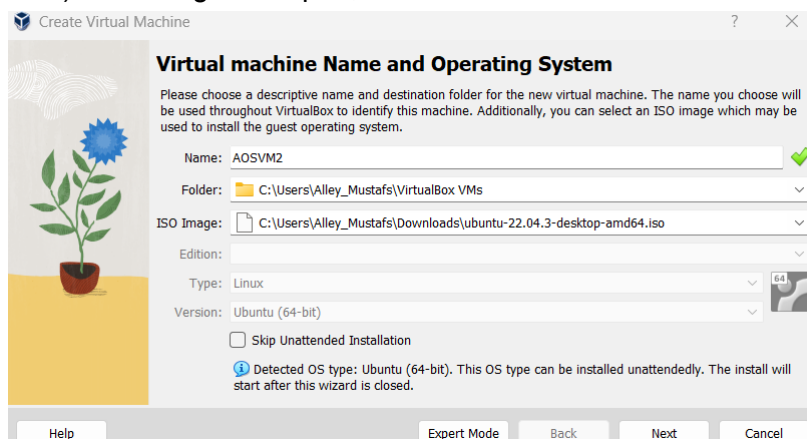
A20491471- Sashank Lakshmana Bommadevara

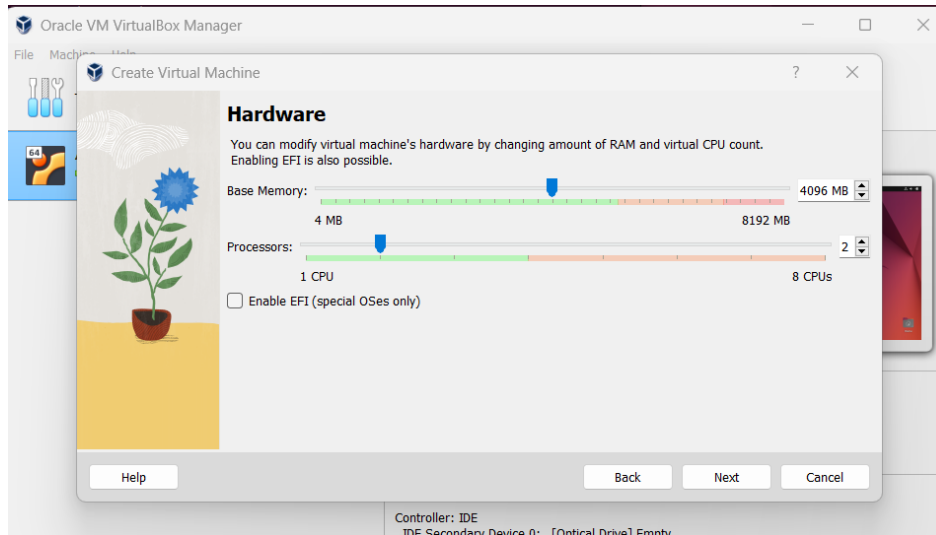
A20516702 - Abdurakhmon Urazboev

A20519664-Syed Alle Mustafa

### Question 1:

Firstly, I installed VMBox from the given link and then installed linux image 22.04. After installing both of them, I configured two virtual machines with following specifications: 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with Bridged Adapter, this can be seen in the screenshots below as well.





After installing the images, I installed openssh-server on both the machines.

```

allemustafa@AosVM:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:

```

Then I turned on the firewall to block all the incoming and outgoing traffic from the ports except port 22 which is used for ssh using ufw which is an abstract layer to use iptables (firewall for packet filtering in linux). This can also be seen from the screenshots below:

```

vboxuser@AOSVM2:~$ sudo ufw status
Status: inactive
vboxuser@AOSVM2:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
vboxuser@AOSVM2:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
vboxuser@AOSVM2:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
vboxuser@AOSVM2:~$ sudo ufw status
Status: inactive

```

```

vboxuser@AOSVM2:~$ sudo ufw enable
Firewall is active and enabled on system startup
vboxuser@AOSVM2:~$ sudo ufw status
Status: active

```

To	Action	From
--	----	----
22/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)

Then I got the ips of both the machine by using following command:

```
allemustafa@AosVM:~$ ip -brief addr show
lo                UNKNOWN          127.0.0.1/8 ::1/128
enp0s3            UP                192.168.68.68/22 fe80::db2a:8198:59c0:6402/64
```

```
vboxuser@AOSVM2:~$ ip -brief addr show
lo                UNKNOWN          127.0.0.1/8 ::1/128
enp0s3            UP                192.168.68.69/22 fe80::a348:ea59:b009:446a/64
```

After which i created public/private pair of keys by using ssh-keygen command:

```
vboxuser@AOSVM2:~$ ssh-keygen -t rsa -C "smustafa22@hawk.iit.edu"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vboxuser/.ssh/id_rsa):
Thunderbird Mail
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vboxuser/.ssh/id_rsa
Your public key has been saved in /home/vboxuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Yc+RkEDZQoaSfv0tCHpaHGFegrKsBPgpROCH3lt1L3M smustafa22@hawk.iit.edu
The key's randomart image is:
+---[RSA 3072]-----+
|o. . +=+..          |
|+.+ ..O O. .        |
|==+O.  oo.O         |
|*+=+O  ...+..       |
|o+=o +  .SooE        |
|oo..O  . .  +        |
|..oo  . .            |
|...  . .             |
|.O                   |
+----[SHA256]-----+
```

```
allemustafa@AosVM:~$ ssh-keygen -t rsa -C "smustafa2@hawk.iit.edu"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/allemustafa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/allemustafa/.ssh/id_rsa
Your public key has been saved in /home/allemustafa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:z9EhpSBHo61YZjxpsV6uwZZkJ5vkA8H9g1baKXMxIr8 smustafa2@hawk.iit.edu
The key's randomart image is:
+---[RSA 3072]-----+
|...O.= .            |
|oooO+o o            |
|.o/B++o .           |
|#B@*  o .           |
|..%=S...            |
|.E+  o .            |
|.  o                 |
|                     |
+----[SHA256]-----+
```

After generating the keys, I copied the keys from the ssh-copy-id command to the other system, which prompted success. After which i was able to remotely access another machine without using the password of that machine (using keypair).

```
vboxuser@AOSVM2:~$ ssh-copy-id allemustafa@192.168.68.68
The authenticity of host '192.168.68.68 (192.168.68.68)' can't be established.
ED25519 key fingerprint is SHA256:atCEh3WgSyiXGRn0hR+ezQ0P3bxNWvBae6Haa4rOSbw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
allemustafa@192.168.68.68's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'allemustafa@192.168.68.68'"
and check to make sure that only the key(s) you wanted were added.

vboxuser@AOSVM2:~$ ssh allemustafa@192.168.68.68
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep  9 06:47:55 2023 from 192.168.68.51
allemustafa@AosVM:~$ ^C
allemustafa@AosVM:~$ exit
```

## Question 2 :

### Commands:

**ssh:** This command is used to connect to remote servers securely. This can be seen in the screenshot below:

```

allemustafa@AosVM:~$ ssh vboxuser@192.168.68.72
vboxuser@192.168.68.72's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

13 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Sat Sep  9 07:06:17 2023 from 192.168.68.51
vboxuser@AOSVM2:~$

```

**Ssh-keygen:** This command is used to generate SSH key Pairs for authentication.

```

asura@asura-VirtualBox:~$ ssh-keygen -t rsa -C "sbommadevara@hawk.iit.edu"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/asura/.ssh/id_rsa):

Created directory '/home/asura/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/asura/.ssh/id_rsa
Your public key has been saved in /home/asura/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+rVL6v8uddaaGe8VVMNxpsewSPjSkfJyAU+N/LDWio sbommadevara@hawk.iit.edu
The key's randomart image is:
+---[RSA 3072]---+
|      . . . oo=|
|      o + o O+|
|      * O *.o|
|      O +.o |
|      S . B  o|
|      . . *.+.o|
|      . E *. o *.|
|      . =.. + o|
|      .+.+=o  ..|
+-----[SHA256]-----+

```

**scp:** This command is used to copy files between local machine and remote servers.  
Copying to Remote machine:

```
allemustafa@AosVM: ~  
allemustafa@AosVM:~$ scp /home/allemustafa/Desktop/Test.txt vboxuser@192.168.68.72:/home/vboxuser/  
vboxuser@192.168.68.72's password:  
Test.txt 100% 7 0.5KB/s 00:00  
allemustafa@AosVM:~$
```

Pulling File From Remote Server:

```
allemustafa@AosVM: ~  
allemustafa@AosVM:~$ scp vboxuser@192.168.68.72:/home/vboxuser/Test.txt /home/allemustafa/Desktop/  
vboxuser@192.168.68.72's password:  
Test.txt 100% 7 3.0KB/s 00:00  
allemustafa@AosVM:~$
```

**history:** This command shows a list of previous commands in the current session.

```
asura@asura:~$ history
1  sudo apt update
2  sudo apt install openssh-server
3  sudo ufw status
4  sudo ufw default deny incoming
5  sudo ufw default deny outgoing
6  sudo ufw allow 22/tcp
7  sudo ufw status
8  sudo ufw enable
9  sudo ufw status
10 clear
11 ip -brief addr show
12 sudo ufw allow 22/tcp
13 clear
14 sudo ufw allow 22/tcp
15 clear
16 ip -brief addr show
17 clear
18 ip addr show
19 clear
20 ip -brief addr show
21 history
```

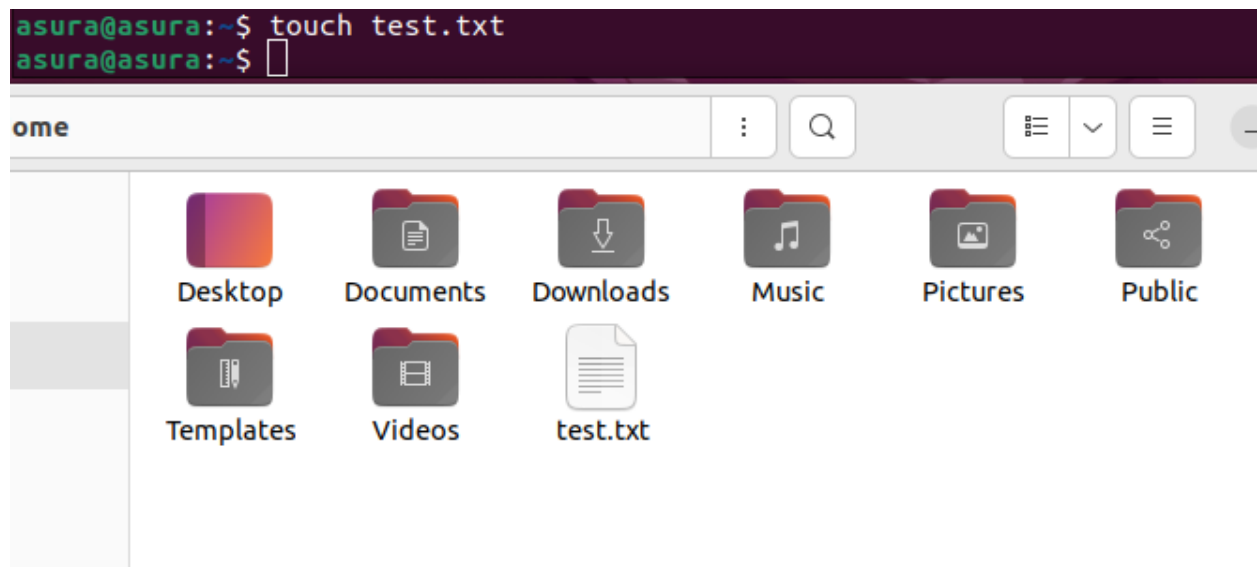
**sudo:** This command enables us to execute commands with admin privileges.

```
asura@asura:~$ sudo apt update
[sudo] password for asura:
Sorry, try again.
[sudo] password for asura:
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
54 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

**Ip:** This command is used to display or configure network-related information.

```
asura@asura:~$ ip -brief addr show
lo                UNKNOWN    127.0.0.1/8 ::1/128
enp0s3            UP         10.0.2.15/24 fe80::6d74:9bcd:6a89:860a/64
```

**Touch:** This command creates an empty file with the specified name.



**ls:** This command lists all files and directories in the current directory.

```
asura@asura-VirtualBox:~$ ls -l ~/.ssh
total 12
-rw----- 1 asura asura 2655 Sep 15 11:24 id_rsa
-rw-r--r-- 1 asura asura 579 Sep 15 11:24 id_rsa.pub
-rw-r--r-- 1 asura asura 142 Sep 15 11:14 known_hosts
```

**mkdir:** This command is used to create a new directory with the specified name.

```
asura@asura:~$ mkdir test
asura@asura:~$ ls
Desktop  Downloads  Pictures  snap      test      Videos
Documents Music      Public   Templates test.txt
```

**cd:** This command is used to change the current working directory.

```
asura@asura:~$ cd /home/asura/Desktop
asura@asura:~/Desktop$
```

**dd:** This command is used for low-level copying and conversion of data.



```
asura@asura:~$ dd if=test.txt of=text1.txt
0+1 records in
0+1 records out
7 bytes copied, 0.000417551 s, 16.8 kB/s
```

**fdisk:** It's a command-line partitioning tool to manage disk partitions.

```
asura@asura:~$ sudo fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 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: 63.45 MiB, 66531328 bytes, 129944 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/loop2: 73.88 MiB, 77463552 bytes, 151296 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/loop3: 237.21 MiB, 248729600 bytes, 485800 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
```

**apt:** This command is a Package manager command useful for install, update, remove Packages.

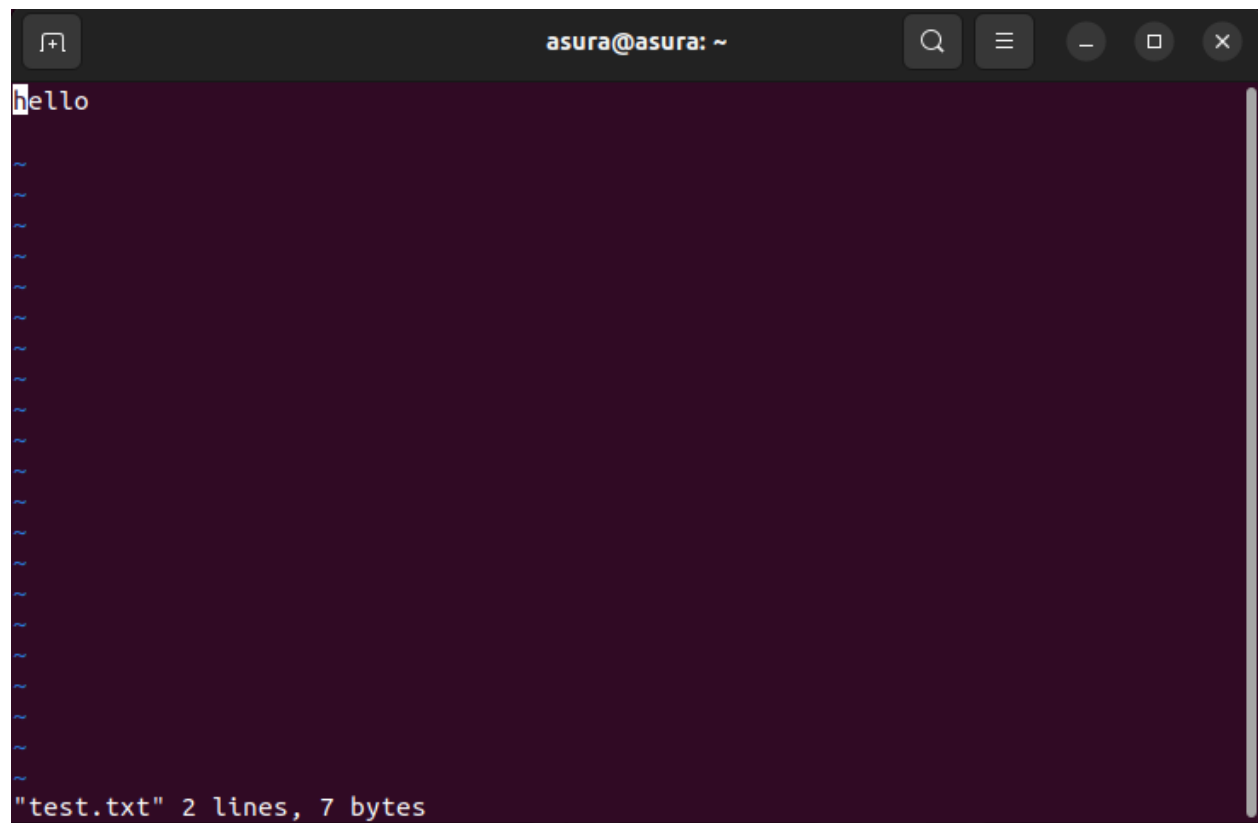
```

asura@asura:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 53 not upgraded.
Need to get 1,655 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-client amd64 1:8.9p1-3ubuntu0.4 [905 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.4 [38.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.4 [434 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
Fetched 1,655 kB in 0s (6,781 kB/s)
Preconfiguring packages ...
(Reading database ... 198892 files and directories currently installed.)
Preparing to unpack .../openssh-client_1%3a8.9p1-3ubuntu0.4_amd64.deb ...
openssh-client (1:8.9p1-3ubuntu0.4) over (1:8.9p1-3ubuntu0.3) ...
previously unselected package openssh-sftp-server.
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3ubuntu0.4_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.4) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.9p1-3ubuntu0.4_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3ubuntu0.4) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.3-2ubuntu0.1_all.deb ...
Unpacking ncurses-term (6.3-2ubuntu0.1) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../ssh-import-id_5.11-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.11-0ubuntu1) ...
Setting up openssh-client (1:8.9p1-3ubuntu0.4) ...
Setting up ssh-import-id (5.11-0ubuntu1) ...

```

vi: This command is the Text editor used to create or edit files.

```
asura@asura:~$ vi test.txt
```



```
hello
```

```
"test.txt" 2 lines, 7 bytes
```

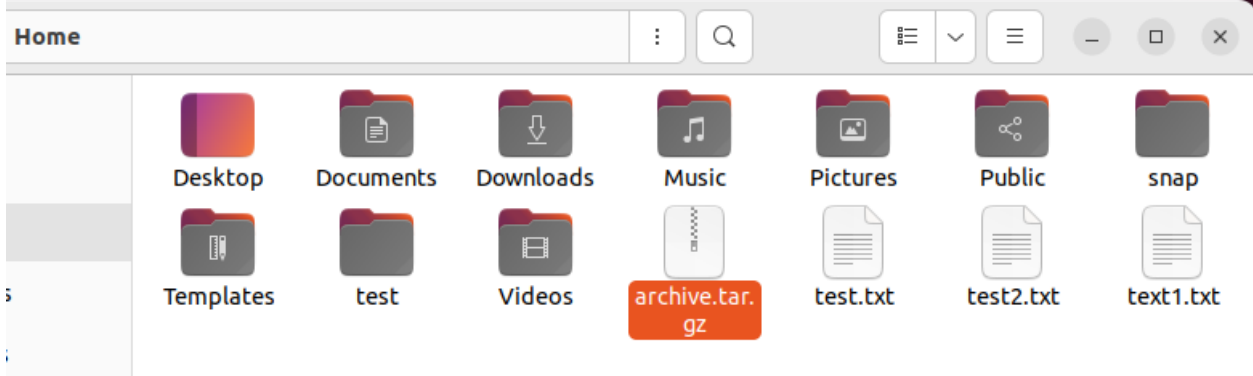
**time:** This command is used to measure the time taken for a command to execute.

```
asura@asura:~$ time touch test2.txt
```

real	0m0.011s
user	0m0.000s
sys	0m0.001s

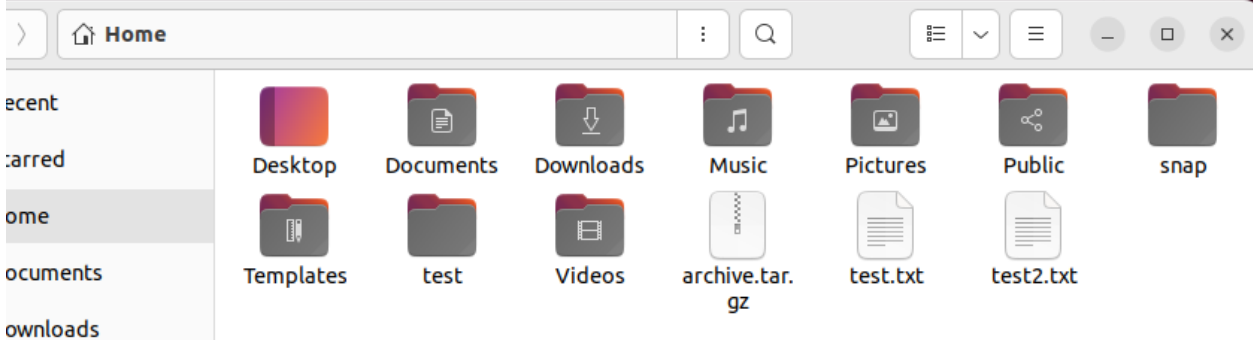
**tar:** Used for creating and extracting archive files.

```
asura@asura:~$ tar -czvf archive.tar.gz test
test/
asura@asura:~$
```



**rm:** This command is used to delete files or directories.

```
asura@asura:~$ rm text1.txt
asura@asura:~$
```



**cat:** This command is used to display the contents of a text file.

```
asura@asura:~$ cat test.txt
hello
```

**bash:** Launches a new bash shell.

```
asura@asura:~$ name() { echo "asura"; }
asura@asura:~$ export -f name
asura@asura:~$ bash
asura@asura:~$ name
asura
```

**more:** This command is used to display the contents of a text file one page at a time.

```
asura@asura:~$ more test1.txt
hello world
hello USA
hello Canada
hi world
hi USA
hi Canada
2
5
3
4
1
```

**watch:** Repeatedly runs a command and displays the output.

```
asura@asura:~$ watch -e cat test.txt
```

```
Every 2.0s: cat test.txt                               asura: Sat Sep 16 13:32:21 2023
hello
```

**ps:** Lists running processes.

```
asura@asura:~$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.2	166672	11768	?	Ss	13:22	0:00	/sbin/init sp
root	2	0.0	0.0	0	0	?	S	13:22	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	13:22	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	13:22	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	13:22	0:00	[slub_flushwq
root	6	0.0	0.0	0	0	?	I<	13:22	0:00	[netns]
root	7	0.0	0.0	0	0	?	I	13:22	0:00	[kworker/0:0-
root	8	0.0	0.0	0	0	?	I<	13:22	0:00	[kworker/0:0H
root	10	0.0	0.0	0	0	?	I<	13:22	0:00	[mm_percpu_wq
root	11	0.0	0.0	0	0	?	I	13:22	0:00	[rcu_tasks_kt
root	12	0.0	0.0	0	0	?	I	13:22	0:00	[rcu_tasks_ru
root	13	0.0	0.0	0	0	?	I	13:22	0:00	[rcu_tasks_tr
root	14	0.0	0.0	0	0	?	S	13:22	0:00	[ksoftirqd/0]
root	15	0.0	0.0	0	0	?	I	13:22	0:00	[rcu_preempt]
root	16	0.0	0.0	0	0	?	S	13:22	0:00	[migration/0]
root	17	0.0	0.0	0	0	?	S	13:22	0:00	[idle_inject/
root	18	0.0	0.0	0	0	?	I	13:22	0:00	[kworker/0:1-
root	19	0.0	0.0	0	0	?	S	13:22	0:00	[cpuhp/0]

**top:** gives real-time data on system activity and resource utilization.

```
top - 13:33:45 up 11 min, 1 user, load average: 0.12, 0.08, 0.05
Tasks: 178 total, 1 running, 177 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.7 sy, 0.0 ni, 99.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3907.7 total, 2321.7 free, 694.1 used, 891.9 buff/cache
MiB Swap: 2680.0 total, 2680.0 free, 0.0 used. 2965.7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1459	asura	20	0	4226980	334264	134028	S	3.3	8.4	0:18.37	gnome-s+
1973	asura	20	0	552868	51312	39084	S	1.0	1.3	0:01.60	gnome-t+
514	root	20	0	0	0	0	I	0.3	0.0	0:00.43	kworker+
2254	root	20	0	0	0	0	I	0.3	0.0	0:00.09	kworker+
1	root	20	0	166672	11768	8312	S	0.0	0.3	0:00.75	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
7	root	20	0	0	0	0	I	0.0	0.0	0:00.07	kworker+
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+

**htop:** Similar to 'top' but with more interactive and user friendly features.

```
0[||| 1.3%] Tasks: 111, 251 thr; 1 running
1[||| 2.0%] Load average: 0.25 0.31 0.15
Mem[|||||||814M/3.82G] Uptime: 00:04:15
Swp[ 0K/2.62G]
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1828	asura	20	0	4166M	344M	136M	S	2.7	8.8	0:08.53	/usr/bin/gnome-
2591	asura	20	0	549M	52220	39792	S	0.7	1.3	0:00.76	/usr/libexec/gn
1584	asura	20	0	235M	7576	6528	S	0.0	0.2	0:00.02	/usr/bin/gnome-
1865	asura	20	0	4166M	344M	136M	S	0.0	8.8	0:00.38	/usr/bin/gnome-
1	root	20	0	98M	11904	8448	S	0.0	0.3	0:00.90	/lib/systemd/sy
185	root	19	-1	40100	17536	16000	S	0.0	0.4	0:00.15	/lib/systemd/sy
224	root	20	0	27048	7168	4608	S	0.0	0.2	0:00.18	/lib/systemd/sy
450	systemd-r	20	0	25528	13768	9600	S	0.0	0.3	0:00.05	/lib/systemd/sy
451	systemd-t	20	0	89376	7424	6656	S	0.0	0.2	0:00.03	/lib/systemd/sy
455	systemd-t	20	0	89376	7424	6656	S	0.0	0.2	0:00.00	/lib/systemd/sy
500	root	20	0	234M	7772	6876	S	0.0	0.2	0:00.05	/usr/libexec/ac
501	root	20	0	2812	1920	1792	S	0.0	0.0	0:00.01	/usr/sbin/acpid
502	root	20	0	8372	2432	2304	S	0.0	0.1	0:00.00	/usr/sbin/anacr
504	avahi	20	0	7628	4096	3712	S	0.0	0.1	0:00.04	avahi-daemon: r
505	root	20	0	9492	2816	2688	S	0.0	0.1	0:00.00	/usr/sbin/cron
506	messagebu	20	0	11084	6656	4096	S	0.0	0.2	0:00.48	@dbus-daemon --

**gcc:** Compiler for the C programming language.

```
asura@asura:~/Desktop$ gcc hello.c -Wall -o opt
asura@asura:~/Desktop$ ls
hello.c  opt
```

**tail:** This command is used to display the last few lines of a text file.

```
asura@asura:~$ tail test.txt
hello
```

**grep:** This command is used to search for text patterns in files or output.

```
asura@asura:~$ grep "hello" test.txt
hello
hello
hello
```

**kill:** This command is used to terminate processes by their PID.

```
asura@asura:~$ ps
  PID TTY          TIME CMD
  3160 pts/0    00:00:00 bash
  3195 pts/0    00:00:00 ps
asura@asura:~$ kill 3195
bash: kill: (3195) - No such process
```

**killall:** This command is used to terminate processes by their name.

**du:** Displays disk usage information.

```

asura@asura:~$ du -h
4.0K    ./Pictures
4.0K    ./Downloads
4.0K    ./Videos
4.0K    ./Templates
4.0K    ./test
4.0K    ./Public
4.0K    ./Music
4.0K    ./Documents
4.0K    ./snap/firefox/2987/.config/gtk-3.0
4.0K    ./snap/firefox/2987/.config/ibus
8.0K    ./snap/firefox/2987/.config/pulse
4.0K    ./snap/firefox/2987/.config/dconf
8.0K    ./snap/firefox/2987/.config/fontconfig
4.0K    ./snap/firefox/2987/.config/gtk-2.0
52K     ./snap/firefox/2987/.config
4.0K    ./snap/firefox/2987/.local/share/icons
4.0K    ./snap/firefox/2987/.local/share/glib-2.0/schemas
8.0K    ./snap/firefox/2987/.local/share/glib-2.0
16K     ./snap/firefox/2987/.local/share
20K     ./snap/firefox/2987/.local

```

**df:** Displays disk space usage.

```

asura@asura:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           391M  1.6M  390M   1% /run
/dev/sda3       24G   13G   11G  55% /
tmpfs           2.0G    0  2.0G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  6.1M  506M   2% /boot/efi
tmpfs           391M  188K  391M   1% /run/user/1000

```

**screen:** A terminal multiplexer that allows you to run multiple terminal sessions within one.



Copyright (c) 2018-2020 Alexander Naumov, Amadeusz Slawinski  
Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski  
Copyright (c) 2010-2014 Juergen Weigert, Sadrul Habib Chowdhury  
Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan,  
Sadrul Habib Chowdhury  
Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder  
Copyright (c) 1987 Oliver Laumann

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

[Press Space for next page; Return to end.]

[illegible]

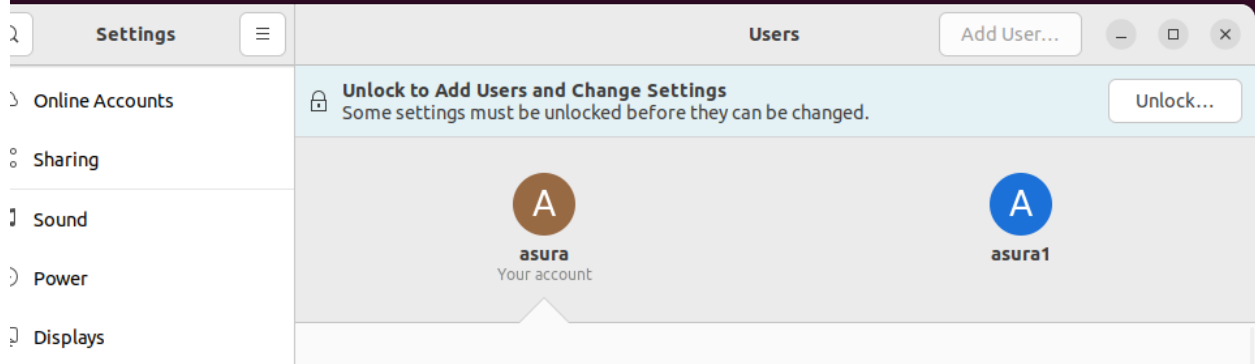
```
asura@asura:~$ chmod go-w test.txt
```

**chown:** This command is used to change file ownership.

```
asura@asura:~$ sudo chown -hR root /home/asura/test  
[sudo] password for asura:
```

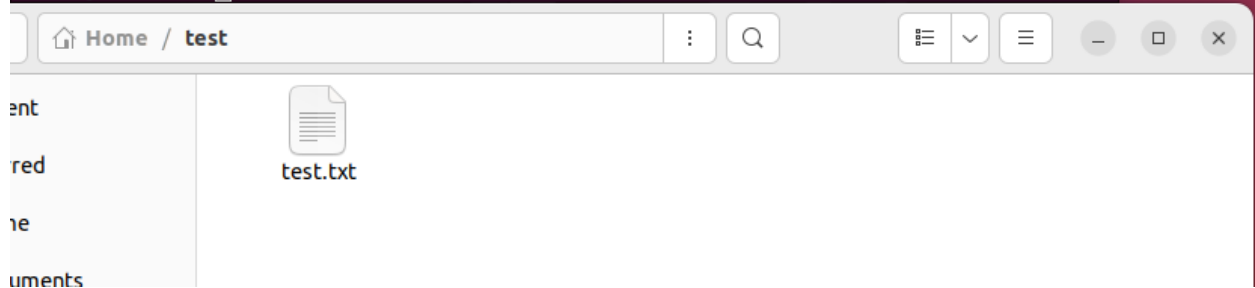
**useradd:** This command is used to add new users to the system.

```
asura@asura:~$ sudo useradd asura1  
asura@asura:~$
```



**mv:** This command is used to move or rename files or directories.

```
asura@asura:~$ mv test.txt -t test  
asura@asura:~$
```



**man:** Displays manual page for a command.

```
asura@asura:~$ man sudo
```

```
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 [-ABEHnPS] [-C num] [-D directory] [-g group] [-h host] [-p prompt]
        [-R directory] [-r role] [-t type] [-T timeout] [-u user]
        [VAR=value] [-i | -s] [command]
    sudoedit [-ABkNS] [-C num] [-D directory] [-g group] [-h host]
        [-p prompt] [-R directory] [-r role] [-t type] [-T timeout]
        [-u user] file ...

DESCRIPTION
    sudo allows a permitted user to execute a command as the superuser or an-
    other 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.
```

**locate:** This command is used to find files in a database

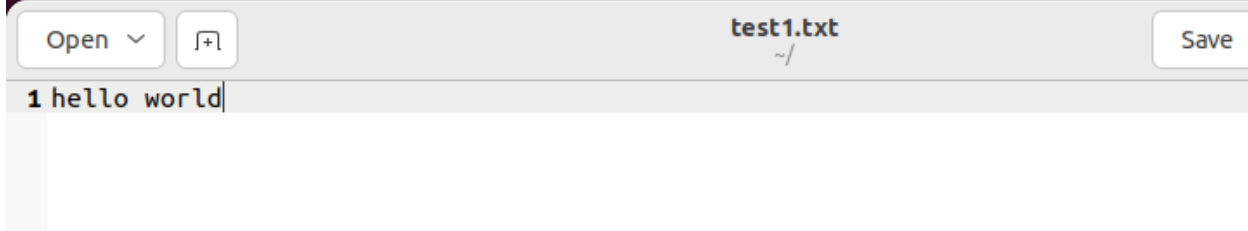
```
asura@asura:~$ locate test.txt
/home/asura/test/test.txt
/usr/share/doc/screen/terminfo/test.txt.gz
```

**find:** Searches for files and directories based on various conditions given.

```
asura@asura:~$ find /home/asura/test -name test.txt
/home/asura/test/test.txt
```

**sed:** It's a stream editor for text manipulation.

```
asura@asura:~$ vi test1.txt
asura@asura:~$
```



The screenshot shows a text editor window titled 'test1.txt'. The window has a menu bar with 'Open' and 'Save' buttons, and a toolbar with a file icon and a plus sign. The main text area contains the text '1 hello world' on the first line.

```
asura@asura:~$ sed 's/hello/hi/' test1.txt
hi world
```

**awk:** It's a text processing tool for data extraction and reporting.

```
asura@asura:~$ cat test1.txt
hello world
hello USA
hello Canada
hi world
hi USA
hi Canada
asura@asura:~$ awk '/hello/ {print}' test1.txt
hello world
hello USA
hello Canada
```

**diff:** This command is used to compare the contents of two text files.

```
asura@asura:~$ diff test.txt test1.txt
3a4,6
> hi world
> hi USA
> hi Canada
```

**sort:** Sorts lines in text files.

```
asura@asura:~$ cat test1.txt
hello world
hello USA
hello Canada
hi world
hi USA
hi Canada
2
5
3
4
1
asura@asura:~$ sort test1.txt
1
2
3
4
5
hello Canada
hello USA
hello world
hi Canada
hi USA
hi world
```

**export:** Sets environment variables.

```
asura@asura:~$ name() { echo "asura"; }
asura@asura:~$ export -f name
asura@asura:~$ bash
asura@asura:~$ name
asura
```

**pwd:** Prints the current working directory.

```
asura@asura:~$ pwd
/home/asura
```

**crontab:** Manages scheduled tasks for users.

```
GNU nano 6.2 /tmp/crontab.C1o3pa/crontab
## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
```

```
asura@asura:~$ crontab -e
no crontab for asura - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 1
No modification made
```

**mount:** Mount devices and file systems.

```
asura@asura:~$ sudo mount -o loop /home/asura/win11.iso /mnt/iso
mount: /mnt/iso: WARNING: source write-protected, mounted read-only.
```

**passwd:** This command is used to change passwords of users.

```
asura@asura:~$ passwd
Changing password for asura.
Current password:
New password:
BAD PASSWORD: The password is shorter than 8 characters
New password:
Retype new password:
passwd: password updated successfully
```

**uname:** Displays system information.

```
asura@asura:~$ uname -a
Linux asura 6.2.0-32-generic #32~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 18 1
0:40:13 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
```

**whereis:** Locates the binary, source, and manual page files for a command.

```
asura@asura:~$ whereis sudo
sudo: /usr/bin/sudo /usr/libexec/sudo /usr/share/man/man8/sudo.8.gz
```

**whatis:** This command gives description of a command.

```
asura@asura:~$ whatis sudo
sudo (8) - execute a command as another user
```

**less:** A text pager that allows to view files one screen at a time.

```
hello world
hello USA
hello Canada
hi world
hi USA
hi Canada
2
5
3
4
1
test1.txt (END)
```

```
asura@asura:~$ less test1.txt

[1]+  Stopped                  less test1.txt
```

**su:** Switches to another user account.

```
asura@asura:~$ su asura1
Password:
$ uname
Linux
$ su asura
Password:
asura@asura:~$
```

**ping:** This command is used to test network connectivity.

```
asura@asura:~$ ping google.com
PING google.com (142.250.191.174) 56(84) bytes of data.
64 bytes from ord38s30-in-f14.1e100.net (142.250.191.174): icmp_seq=1 ttl=118 time=8.93 ms
64 bytes from ord38s30-in-f14.1e100.net (142.250.191.174): icmp_seq=2 ttl=118 time=2.47 ms
64 bytes from ord38s30-in-f14.1e100.net (142.250.191.174): icmp_seq=3 ttl=118 time=2.38 ms
64 bytes from ord38s30-in-f14.1e100.net (142.250.191.174): icmp_seq=4 ttl=118 time=2.78 ms
```

**Traceroute:** This command is used to trace the route taken by the network packets to the destination.

```
asura@asura:~$ traceroute google.com
traceroute to google.com (142.250.191.174), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.614 ms  0.565 ms  0.540 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
```

**date:** Displays the current date and time.

```
asura@asura:~$ date
Sat Sep 16 02:22:52 PM CDT 2023
```

**time:** Measures the time taken by a command.

```
asura@asura:~$ time touch test2.txt

real    0m0.011s
user    0m0.000s
sys     0m0.001s
```

**wget:** Downloads files from the internet.

```
asura@asura:~$ wget https://www.microsoft.com/software-download/windows11
--2023-09-16 15:03:00-- https://www.microsoft.com/software-download/windows11
Resolving www.microsoft.com (www.microsoft.com)... 23.11.197.159, 2600:1407:e800:686::356e, 2600:1407:e800:684::356e, ...
Connecting to www.microsoft.com (www.microsoft.com)|23.11.197.159|:443... connected.
HTTP request sent, awaiting response...
```

**wc:** Counts the number of words, lines and characters in a text file.

```
asura@asura:~$ wc test1.txt
11 17 72 test1.txt
```

**clear:** Clears the terminal screen.

```
Setting up libcc1-0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up liblsan0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libitm1:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libc-devtools (2.35-0ubuntu3.3) ...
Setting up libtsan0:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up libctf0:amd64 (2.38-4ubuntu2.3) ...
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up libc6-dev:amd64 (2.35-0ubuntu3.3) ...
Setting up binutils-x86-64-linux-gnu (2.38-4ubuntu2.3) ...
Setting up binutils (2.38-4ubuntu2.3) ...
Setting up gcc-11 (11.4.0-1ubuntu1~22.04) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
asura@asura:~$
```

```
asura@asura:~$
```

**exit:** closes the current terminal session.



```
asura@asura:~$ exit
exit
$
```

## Question 3:

a)

```
1 #!/bin/bash
2
3 time (
4     echo "DD Benchmark"
5     echo "Write benchmark running..."
6     dd if=/dev/zero of=testfile bs=15M count=1024 oflag=dsync
7     echo "Write benchmark completed."
8
9     echo "Read benchmark running..."
10    dd if=testfile of=/dev/null bs=15M
11    echo "Read benchmark completed."
12
13    rm testfile
14 ) &> disk-benchmark-background-log.txt &
15
16 disown
```

Step by step explanation of the script: We are using time command to measure the time it takes for a code inside parentheses to execute. Inside the parentheses, We have used multiple echo commands to give some information about the execution of benchmark.

We are using dd command to write into a testfile null characters from /dev/zero which does not need memory allocation so we will not be spending time for reading from the disk measuring only write speed. /dev/zero can provide as much null characters as needed when read. We are specifying that block size that needs to be read along with number of blocks to read as bs and count. Oflag option is used to set dsync flag which makes sure that we are actually writing into a real disk not buffering anything to measure disk speed.

To test the read speed, we are using the testfile we just created and to avoid writing anything to a disk, we are giving /dev/null as write destination. In the linux systems all writes to /dev/null are silently dropped so we will not waste time for write only measuring read time.

After that we are channeling both stdout and stderr to disk-benchmark-background-log.txt using linux's redirection operation. We know that > and >> operators are used to redirect the stream into a file. To redirect stdout and stderr we should use their file descriptors 1 and 2 with redirect operator but to simplify things we can use &> operator to redirect both stdout and stderr of the commands executed before that command to disk-benchmark-background-log.txt.

The & at the end makes the shell to run this program on background.

Disown command is used to remove jobs from job table and ensures that this script is terminated after execution.

b)

```
1 #!/bin/bash
2
3 if [ ! -f network-test-machinelist.txt ];
4 then
5     echo "File network-test-machinelist.txt not found!"
6     exit 1
7 fi
8
9 : > network-test-latency.txt
10
11 while IFS= read -r address;
12 do
13     if [ -z "$address" ];
14     then
15         continue
16     fi
17
18     rtt=$(ping -c 3 "$address" | tail -1 | awk '{print $4}' | cut -d '/' -f 2)
19
20     if [ -n "$rtt" ];
21     then
22         echo "$address $rtt" >> network-test-latency.txt
23     else
24         echo "$address -" >> network-test-latency.txt
25     fi
26 done < network-test-machinelist.txt
```

Step by step explanation of the script: We are checking in if statement if the file named network-test-machinelist.txt exists in the folder and if not we are terminating the program immediately.

If file exists, we continue the execution of the program. Line 9 creates by redirecting no-op command to a file network-test-latency.txt which essentially truncates the content of the file.

In the while loop, we are reading lines as addresses IFS= just means we are not removing any whitespaces in reading. -r ensures backslashes are not treated as escape characters.

In the following if statement we are skipping the line if it is empty. If not, we execute a ping command to the address and format the output as we need.

```

aurazboev@aurazboev-VM:~/CS550$ ping -c 3 google.com
PING google.com (142.250.190.78) 56(84) bytes of data.
64 bytes from ord37s34-in-f14.1e100.net (142.250.190.78): icmp_seq=1 ttl=115 time=21.1 ms
64 bytes from ord37s34-in-f14.1e100.net (142.250.190.78): icmp_seq=2 ttl=115 time=19.9 ms
64 bytes from ord37s34-in-f14.1e100.net (142.250.190.78): icmp_seq=3 ttl=115 time=18.5 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 18.548/19.845/21.060/1.027 ms

```

As we can see ping command returns above results when executed. All needed information about rtt is at the last line. So, we take the last line of the response with `tail -1` command using pipe and then we are piping it to `awk` command which allows us to perform text processing. 'print \$4' results in 4th portion of the file split by whitespace. So, the whole text is \$0 and each portion split by white space is numbered starting with 1. rtt is \$1, min/avg/max/mdev is \$2 and so on so we take 4th part of the last line which has numbers that we need. After that we cut numbers by "/" character and take second part of the text which is average of latencies. After that we append the address and latency to a file we truncated. If ping command does not work and we have nothing as rtt we just leave - next to an address.

At the end we close the while loop by specifying the file that is read line-by-line.

c)

```

1 import matplotlib.pyplot as plt
2
3 dns_entries = []
4 rtt = []
5
6 with open('network-test-latency.txt', 'r') as records:
7     for line in records:
8         part = line.strip().split(' ')
9         dns_entries.append(part[0])
10        rtt.append(float(part[1]) if part[1] != '-' else None)
11
12 plt.title('Average RTT for Various DNS entries')
13 plt.ylabel('Average RTT (ms)')
14 plt.xlabel('DNS Entries')
15
16 plt.bar(dns_entries, rtt, color='red')
17 plt.xticks(rotation=45)
18
19 plt.tight_layout()
20 plt.show()
21

```

Step by step explanation of the script: We are starting off with importing the Pyplot library to draw the graph. We are opening the file with rtt records in reading mode and starting to iterate through lines using for loop. Then we are splitting the line into parts with blankspace delimiter and adding those parts into respective arrays to display later. After that we are giving a title for the graph along with axis names. Then we are giving respective data as parameters to the bar graph. In order to fit the names of domains we are rotating the x axis labels by 45 degrees. Then using `tight_plot()` function we are adjusting plot parameters to fit into the figure area. Then we are displaying the resulting graph.

## Question 4:

**a. In the system configuration of the VM, explain how changing the number of processors**

**changes the behavior 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. Changing the number of processors changes the available computing power of the VM. We can set it to minimum when we only want to test the environment, running lightweight applications or running multi VM's on the same host. We use maximum settings when we want to perform heavy intense tasks like video rendering or gaming etc.

However setting to maximum can be a bad idea when it causes resource congestion and overallocates resources, slowing down other VMs or the host system.

**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. ?**

A. **None:** This disables the hardware virtualization acceleration. It is appropriate for virtual machines (VMs) using outdated hardware or in situations where virtualization capabilities are not available or required.

**Legacy:** This option is used when running old operating systems which do not support hardware virtualization like windows 98 etc.

**Minimal:** The support for paravirtualization in this option is rather limited. It is a fundamental level of acceleration and is appropriate for the majority of general-purpose virtual machines, including Ubuntu Linux.

**Hyper V:** This is virtualization technology developed by Windows and works best with windows VM's

**KVM:** For Linux-based systems, Kernel-based Virtual Machine (KVM) is a popular and effective virtualization solution. The "KVM" option will offer the optimum performance and compatibility if you are running Ubuntu Linux as a guest VM on a KVM-based hypervisor.

When running Ubuntu Linux on a KVM-based hypervisor, such as QEMU or virt-manager, the "KVM" option is often the best option. In the Ubuntu ecosystem, KVM is highly supported and delivers good performance and compatibility for Linux guests. But the "Minimal" option should work well in most cases if you're using a different hypervisor or virtualization platform.

**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 ?**

**A. IDE (Integrated Drive Electronics) Controller:** In modern VM systems, IDE controllers are less used and older. In earlier PCs, they were frequently used to link optical drives and hard drives.

When you need to simulate outdated hardware or use older operating systems that don't support more contemporary storage controllers, you might want to use an IDE controller in legacy virtual machine settings.

**SATA (Serial Advanced Technology Attachment) Controller:** Hard disks and SSDs in virtual machines are frequently connected using SATA controllers. They perform effectively and have a large popularity.

SATA controllers work well with the majority of general-purpose virtual machines (VMs), including file servers, database servers, and web servers. They are affordable and offer a nice mix between compatibility and performance. A SATA controller is frequently a trustworthy option if the workload for your virtual machine doesn't require exceedingly high I/O performance.

**NVMe (Non-Volatile Memory Express) Controller:** NVMe controllers are made for high-performance storage systems like SSDs. In comparison to SATA or IDE controllers, they use the PCIe interface for quicker data transfer and lower latency.

For VMs that require outstanding storage performance, such as data analytics, machine learning, or virtual desktop infrastructure (VDI), NVMe controllers are the best choice. Using NVMe storage controllers can help maximize performance if your virtual machine work involves managing huge datasets, real-time data processing, or high-speed I/O activities.

**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.**

**A. NAT (Network Address Translation):** The VM shares the network connection and IP address of the host computer when in NAT mode. Due to the host's role as a router, the VM can connect to other networks while only having one IP address shown.

When you require internet connection for your virtual machine but not direct external access, NAT is the right choice. It is frequently employed in testing and development settings where virtual machines don't require public IP addresses but do require internet access for software upgrades.

**Bridged Adapter:** A bridged adapter connects the virtual machine (VM) to the physical network directly, giving it access to the host's network for its IP address. On the network, it functions as a different physical machine.

When you need the VM to be accessible from other devices on the same physical network, use a bridged adapter. It's perfect for situations when you require the virtual machine to have its own IP address and be connected to the local network, like hosting a web server or executing network services.

**Internal Network:** However, it separates them from external networks and allows communication between VMs running on the same host. However, VMs are unable to connect to the host's network or the internet. They can only communicate with one another.

When you want many VMs to communicate with one another but not with outside resources, internal networks are ideal for creating closed lab settings or testing scenarios.

**Host-only Network:** Host-only Network connects VMs to the host system over a network, but isolates them from other networks. VMs are capable of communicating with the host and one another, but not with the internet.

It works well in situations where communication between VMs and the host is necessary for development, debugging, or isolated network testing.

**e. For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers.**

A. **USB 1.1:** it is the earliest USB standard. It supports a maximum data transfer speed of 12Mbps. It is also known as Full speed USB

USB 2.0: it is the next generation of USB 1.1. It supports a maximum data transfer rate of 480Mbps. It is also known as High speed USB

USB3.0: also known as Super speed USB, it is a more recent standard of USB. It supports a maximum data transfer rate of 5 Gbps for Gen 1 and 10 Gbps for Gen 2.

In Vm's we can choose the USB configuration based on the devices we plan to connect. If we are unsure of their supported versions then we can choose USB 3.0 as it is backward compatible.

## Question 5:

**i. Today's commodity processors have 1 to 96 cores, and specialized GPUs having 10000+ CUDA-cores. About how many cores/threads are expected to be in future commodity processors in the next five years?**

- A. As the rate of advancement, we can forecast commodity processors having 256 cores (since 128 cores cpu is already introduced by AMD called BERGAMO. However, because of how fast technology can improve there is a chance for this number to be higher but it cannot be too much from here as the heat dissipation in a single socket will be a bottleneck.

**ii. Describe what a core and hardware thread is on a modern processor, and the difference between them?**

- A. Core is responsible for execution of a process or program, it processes all the instructions. It performs 4 basic tasks to do so: fetch, decode, execute, write back. It processes the information using ALU and fetches it from the storage.  
Where a hardware thread holds essential information about a software thread that is needed for the core to run that software thread. The rest of work is handled by the core.

**iii. Compare GPU and CPU chips in terms of their strength and weakness. In particular, discuss the tradeoffs between power efficiency, programmability and performance?**

- A. GPU's have numerous small cores, which make them extremely efficient for performing multithreaded tasks, also they consume less power and can have better throughput on less power. However CPU's contain less threads comparatively, but are more programmable, and can run huge programs with complex sequential logic.

**iv. Why do we not have processors running at 100GHz today (as might have been predicted in the year 2000)?**

- A. The power used by circuits increases with the square of the voltage and clock frequency, keeping everything else constant. The cpu will use millions times more power than 1GHz processor, which is not practical. Similarly the memory access speed will be so much slower than the CPU which won't have much gain.

## Question 6:

**i . Why is threading useful on a single-core processor?**

- A. Undoubtedly threading in a single-core processor is beneficial, since it gives an image of parallelism to the user. It gives time to every process at the same time, which makes sure that more than one process is working concurrently (unless a process needs full resource utilization).

**ii. Do more threads always mean better performance?**

- A. Not always, More thread count may increase the performance of a system depending on the type of process (i.e data intensive or video editing/processing tasks). But in general if

there are too many threads, it may lead to too much context switching, memory utilization and will affect the performance of the system.

**iii. Is super-linear speedup possible? Explain why or why not.**

- A. Yes, although it is rare for a speed up to be more than the number of processors working, it might be possible but not sustainable. For example, if the workload depends upon the memory caching and localizing, it might be possible (one process fetches it from memory while others from cache, so it may appear that the speed up is superlinear). Also if the workload depends upon backtracking (dynamic problems), this way the workload might get less in every step, hence each process will work on the information (calculations) of other processes, hence speedup could be superlinear/

**iv. Why are locks needed in a multi-threaded program?**

- A. Locks are very important for a multi-threaded program, since it solves the problem of two or more processes simultaneously accessing the same resource at the same time, which might cause race conditions, also dirty reads and lack of concurrency. Using locks will provide data consistency, proper sharing/usage of resources and prevention from race conditions. Although it can lead to deadlocks, this issue can be solved by taking safety measures (such as avoiding mutual exclusion).

**v. Would it make sense to limit the number of threads in a server process?**

- A. Yes, it might make sense to limit the number of threads, since using too many threads might exhaust the memory, since each of them will require its own stack and it will be very difficult to handle concurrency between threads.

The End