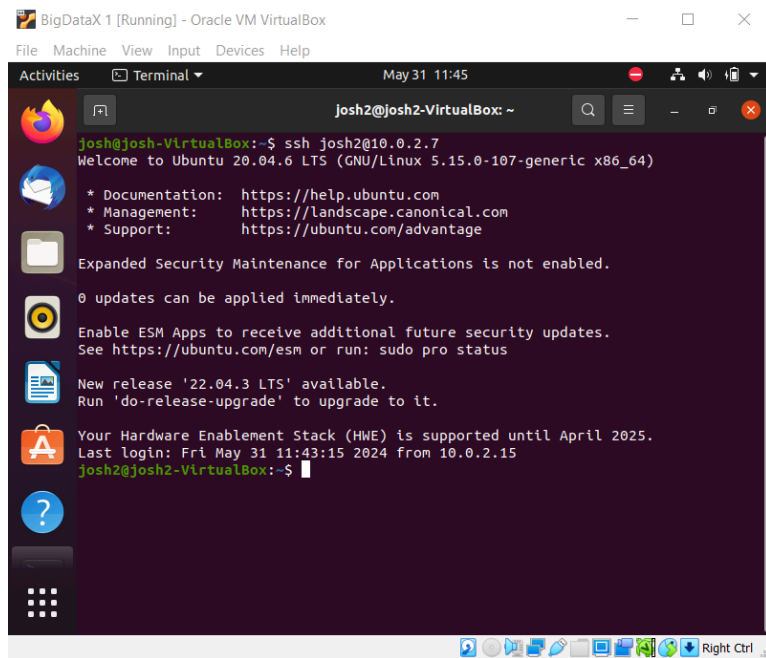


1. VM Setup



```
BigDataX 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal May 31 11:45
josh2@josh2-VirtualBox: ~
josh@josh-VirtualBox:~$ ssh josh2@10.0.2.7
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-107-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.

0 updates can be applied immediately.

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

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri May 31 11:43:15 2024 from 10.0.2.15
josh2@josh2-VirtualBox:~$
```



```
josh2@josh2-VirtualBox:~$ ssh josh@10.0.2.15
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-107-generic x86_64)

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

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

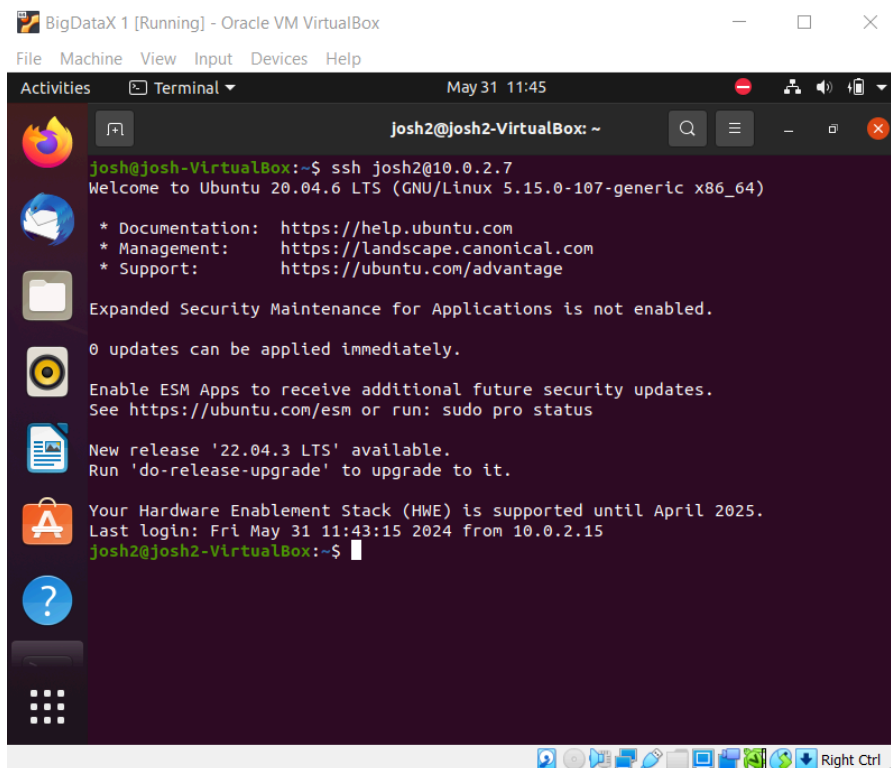
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri May 31 11:42:48 2024 from 10.0.2.7
josh@josh-VirtualBox:~$
```

2. Commands:

- ssh: This command logs in as a user into a remote host machine according to the username and IP address provided.



```
BigDataX 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal May 31 11:45
josh2@josh2-VirtualBox: ~
josh@josh-VirtualBox:~$ ssh josh2@10.0.2.7
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-107-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.

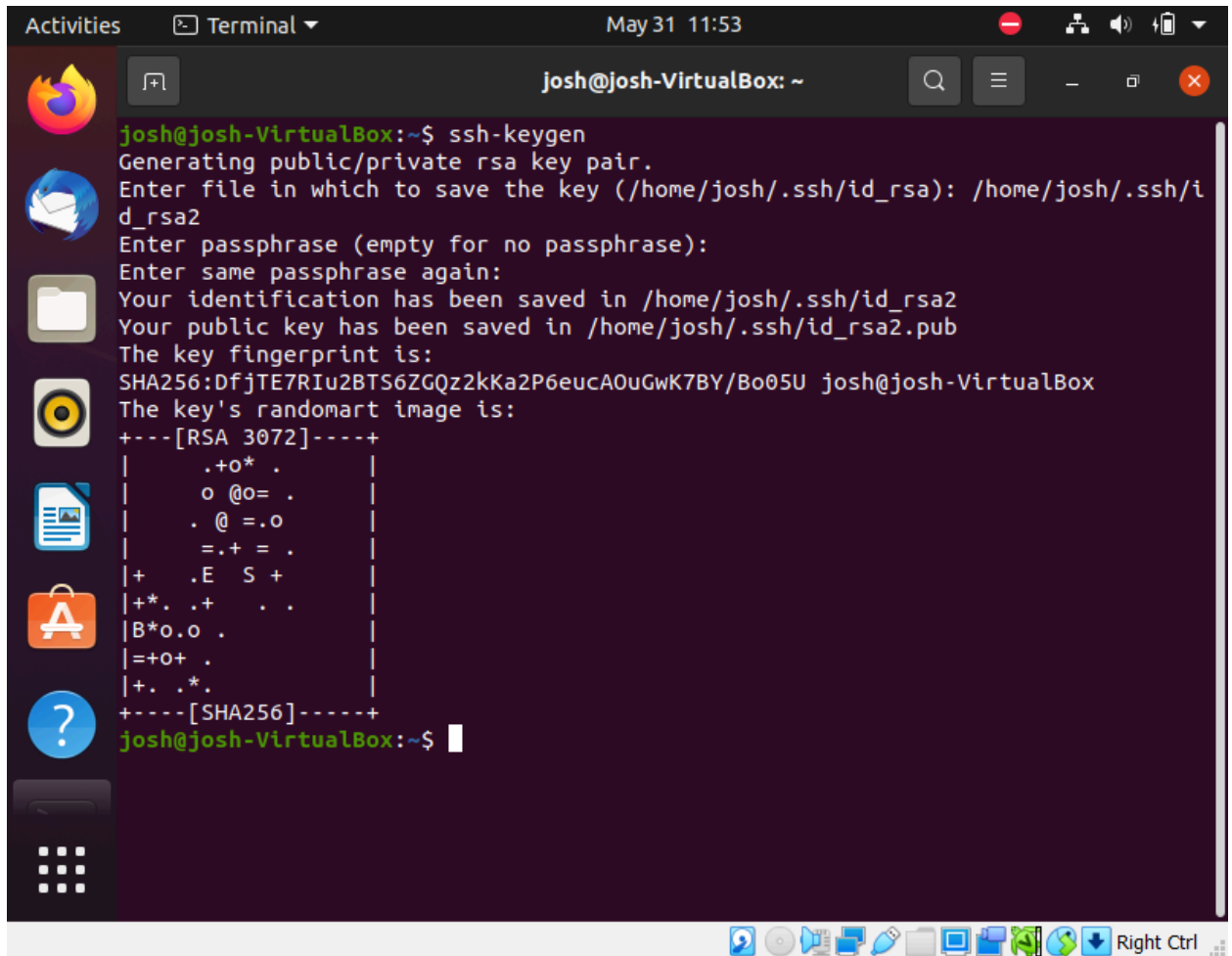
0 updates can be applied immediately.

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

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri May 31 11:43:15 2024 from 10.0.2.15
josh2@josh2-VirtualBox:~$
```

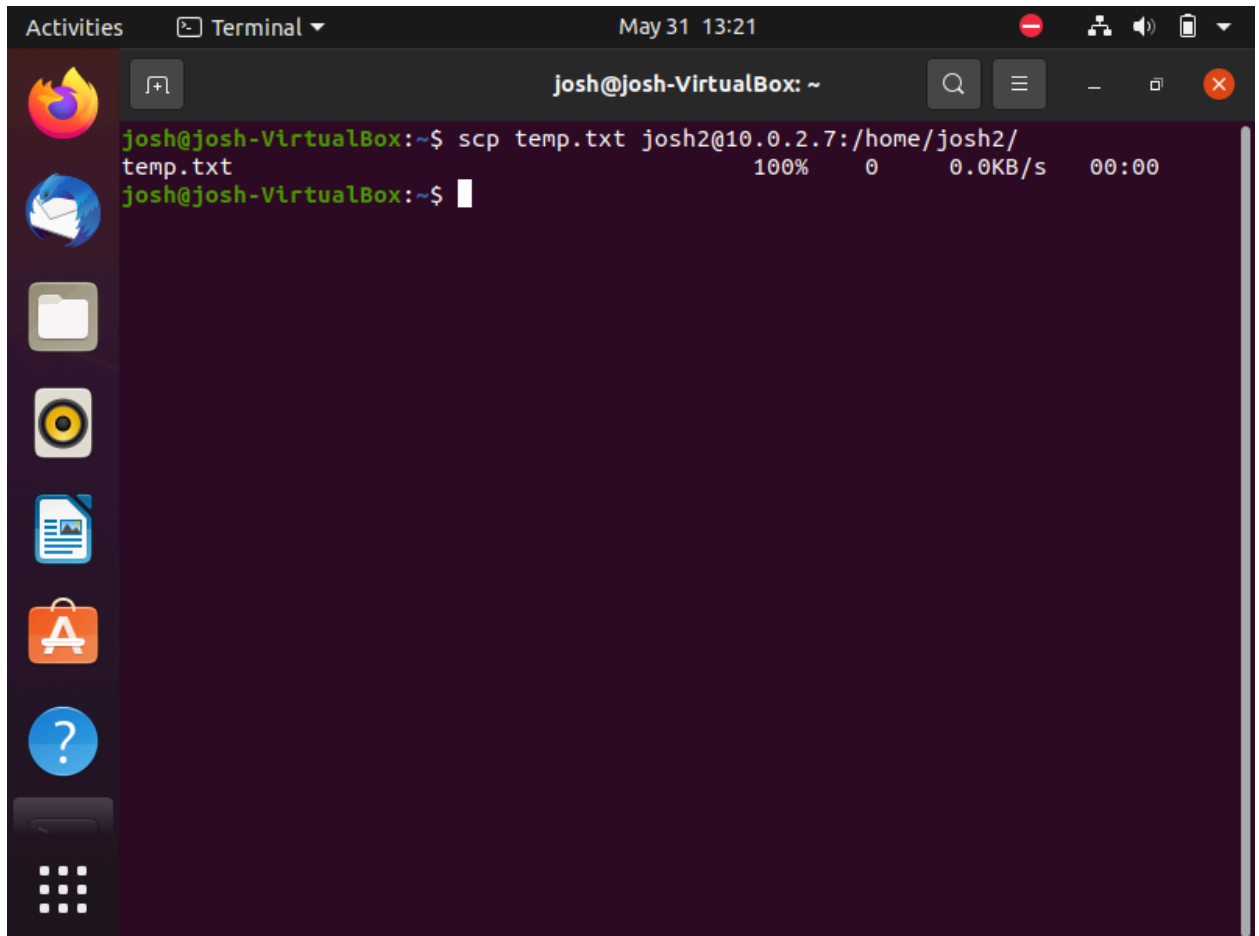
- b) ssh-keygen: This command generates a public and private key pair, by default using RSA.



The image shows a terminal window titled "josh@josh-VirtualBox: ~" with a dark purple background. The window has a top bar with "Activities", "Terminal", and the date/time "May 31 11:53". On the left side, there is a vertical dock with icons for Firefox, a mail client, a file manager, a CD/DVD drive, a document viewer, the Ubuntu Software Center, and a help icon. The terminal output shows the execution of the `ssh-keygen` command, which generates an RSA key pair. It prompts for a file name (defaulting to `/home/josh/.ssh/id_rsa`), a passphrase, and then displays the key fingerprint and a randomart image. The randomart image is a square composed of characters representing the key's fingerprint. The terminal ends with the prompt `josh@josh-VirtualBox:~$`.

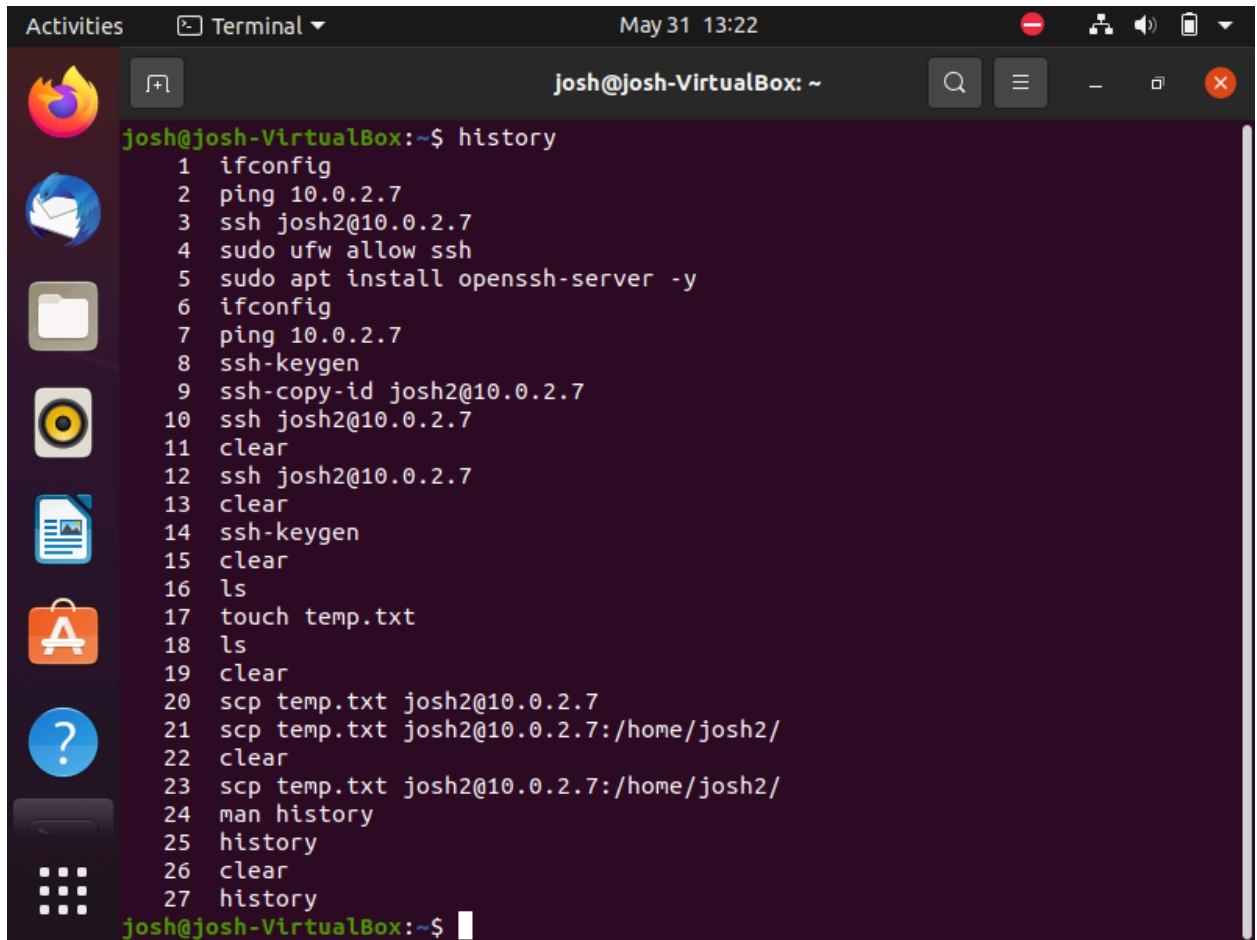
```
josh@josh-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/josh/.ssh/id_rsa): /home/josh/.ssh/i
d_rsa2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/josh/.ssh/id_rsa2
Your public key has been saved in /home/josh/.ssh/id_rsa2.pub
The key fingerprint is:
SHA256:DfjTE7RIu2BTS6ZGQz2kKa2P6eucA0uGwK7BY/Bo05U josh@josh-VirtualBox
The key's randomart image is:
+---[RSA 3072]---+
|      .+o*   .      |
|      o @o=   .      |
|      . @ =.o      |
|      =.+ =.   .      |
|+   .E  S +       +  |
|+*.  .+   .   .      |
|B*o.o .           |
|+=o+  .           |
|+.  .*.           |
+-----[SHA256]-----+
josh@josh-VirtualBox:~$
```

- c) scp: This command copies a file (or multiple) to the destination at a remote machine. It can also copy from a remote machine to the user's machine.

A screenshot of a Linux terminal window titled "josh@josh-VirtualBox: ~". The window shows the execution of the command "scp temp.txt josh2@10.0.2.7:/home/josh2/temp.txt". The output indicates a successful copy: "100% 0 0.0KB/s 00:00". The terminal has a dark purple background and a light-colored prompt. The window's title bar includes "Activities", "Terminal", and the date/time "May 31 13:21". A sidebar on the left contains icons for various applications like Firefox, Mail, Files, and the App Store.

```
josh@josh-VirtualBox: ~  
josh@josh-VirtualBox:~$ scp temp.txt josh2@10.0.2.7:/home/josh2/  
temp.txt                                100%  0  0.0KB/s  00:00  
josh@josh-VirtualBox:~$
```

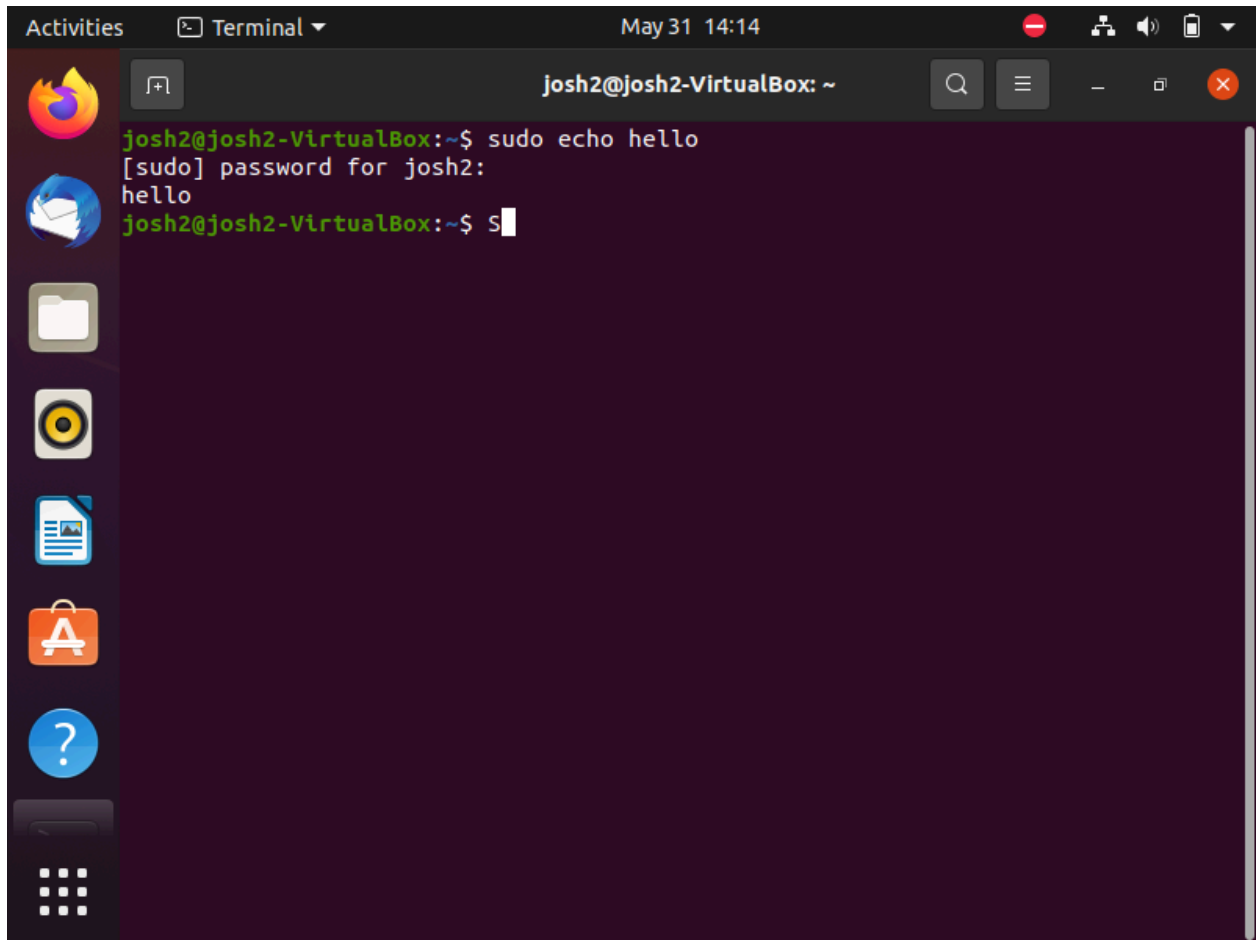
- d) history: This command outputs a history of previous commands input into the shell.



The screenshot shows a terminal window titled "josh@josh-VirtualBox: ~" with a search icon, a menu icon, and window control buttons. The terminal displays the output of the 'history' command, which lists 27 previous commands. The commands are numbered 1 through 27. The terminal window has a dark background and a light-colored text. The left sidebar of the desktop environment shows various application icons including Firefox, a mail client, a file manager, a media player, a document viewer, an application store, a help icon, and a dock icon.

```
josh@josh-VirtualBox:~$ history
1  ifconfig
2  ping 10.0.2.7
3  ssh josh2@10.0.2.7
4  sudo ufw allow ssh
5  sudo apt install openssh-server -y
6  ifconfig
7  ping 10.0.2.7
8  ssh-keygen
9  ssh-copy-id josh2@10.0.2.7
10 ssh josh2@10.0.2.7
11 clear
12 ssh josh2@10.0.2.7
13 clear
14 ssh-keygen
15 clear
16 ls
17 touch temp.txt
18 ls
19 clear
20 scp temp.txt josh2@10.0.2.7
21 scp temp.txt josh2@10.0.2.7:/home/josh2/
22 clear
23 scp temp.txt josh2@10.0.2.7:/home/josh2/
24 man history
25 history
26 clear
27 history
josh@josh-VirtualBox:~$
```

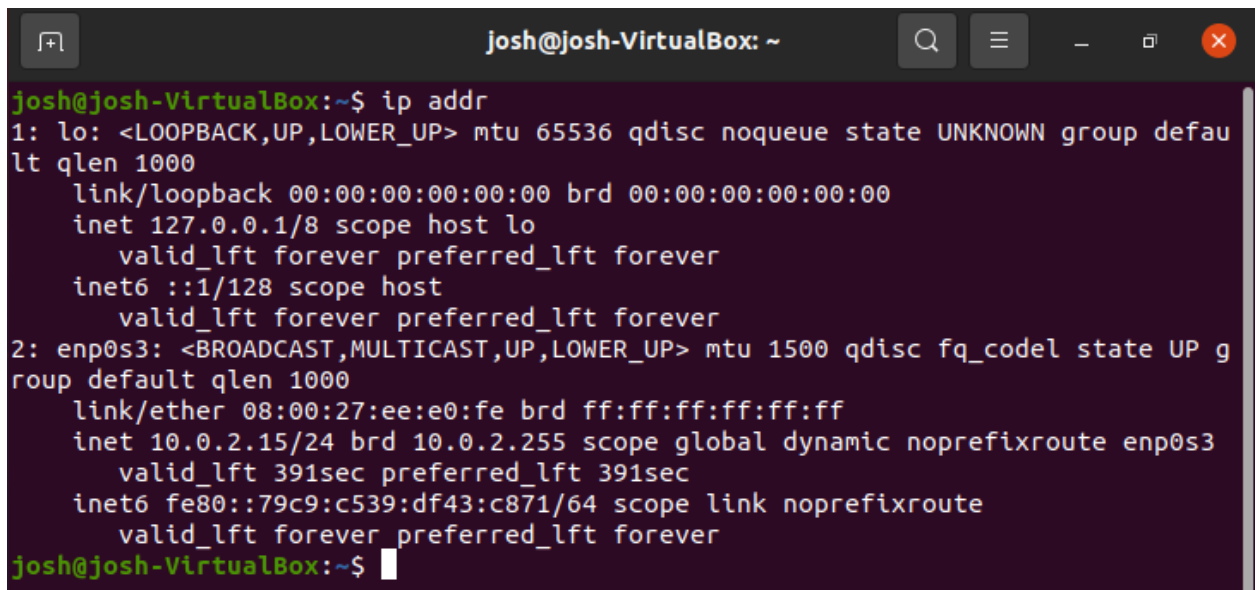
- e) `sudo`: This command allows you to execute a command as a “superuser” (with administrator privileges)



A terminal window titled "josh2@josh2-VirtualBox: ~" with a search bar and menu icons. The terminal shows the following commands and output:

```
josh2@josh2-VirtualBox:~$ sudo echo hello
[sudo] password for josh2:
hello
josh2@josh2-VirtualBox:~$ S
```

- f) `ip`: This command has multiple subcommands relating to devices/routing/networks.



A terminal window titled "josh@josh-VirtualBox: ~" with a search bar and menu icons. The terminal shows the following command and output:

```
josh@josh-VirtualBox:~$ 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:ee:e0:fe brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 391sec preferred_lft 391sec
    inet6 fe80::79c9:c539:df43:c871/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
josh@josh-VirtualBox:~$
```

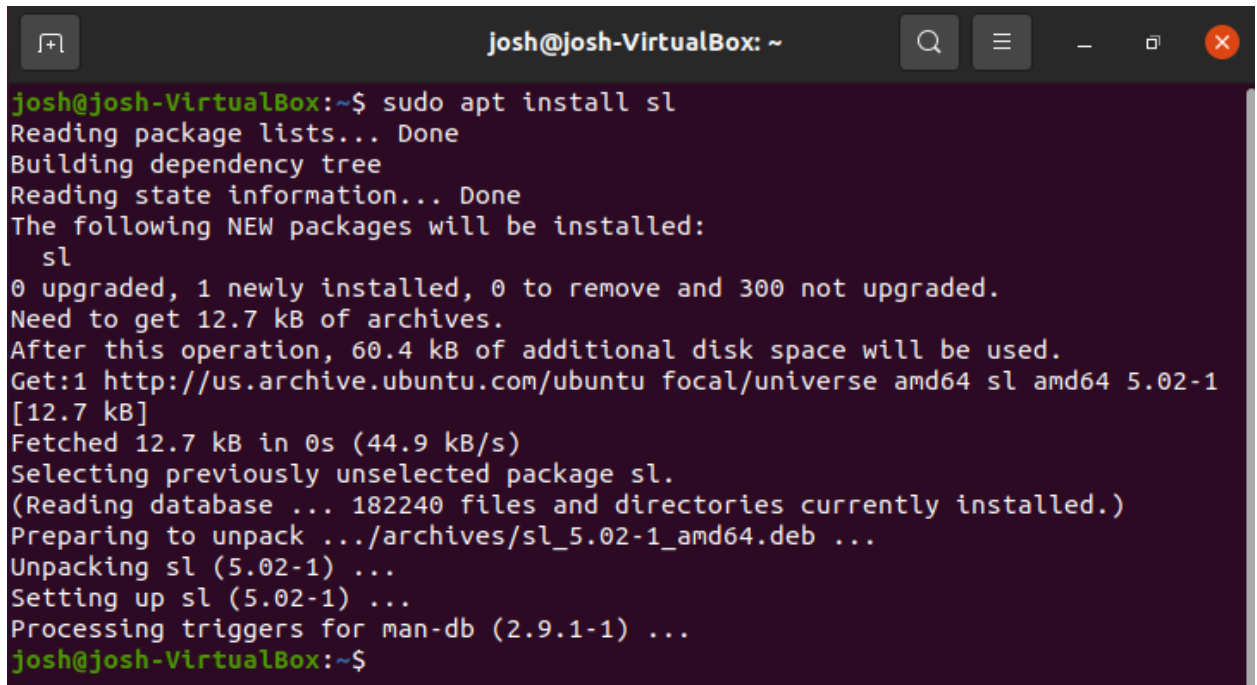
- g) dd: This command copies data from one location to another, using stdin by default.

```
josh@josh-VirtualBox: ~  
josh@josh-VirtualBox:~$ dd if=temp.txt of=new.txt  
0+0 records in  
0+0 records out  
0 bytes copied, 0.000116212 s, 0.0 kB/s  
josh@josh-VirtualBox:~$
```

- h) fdisk: This command allows you to inspect/add/modify/delete partitions on hard disks.

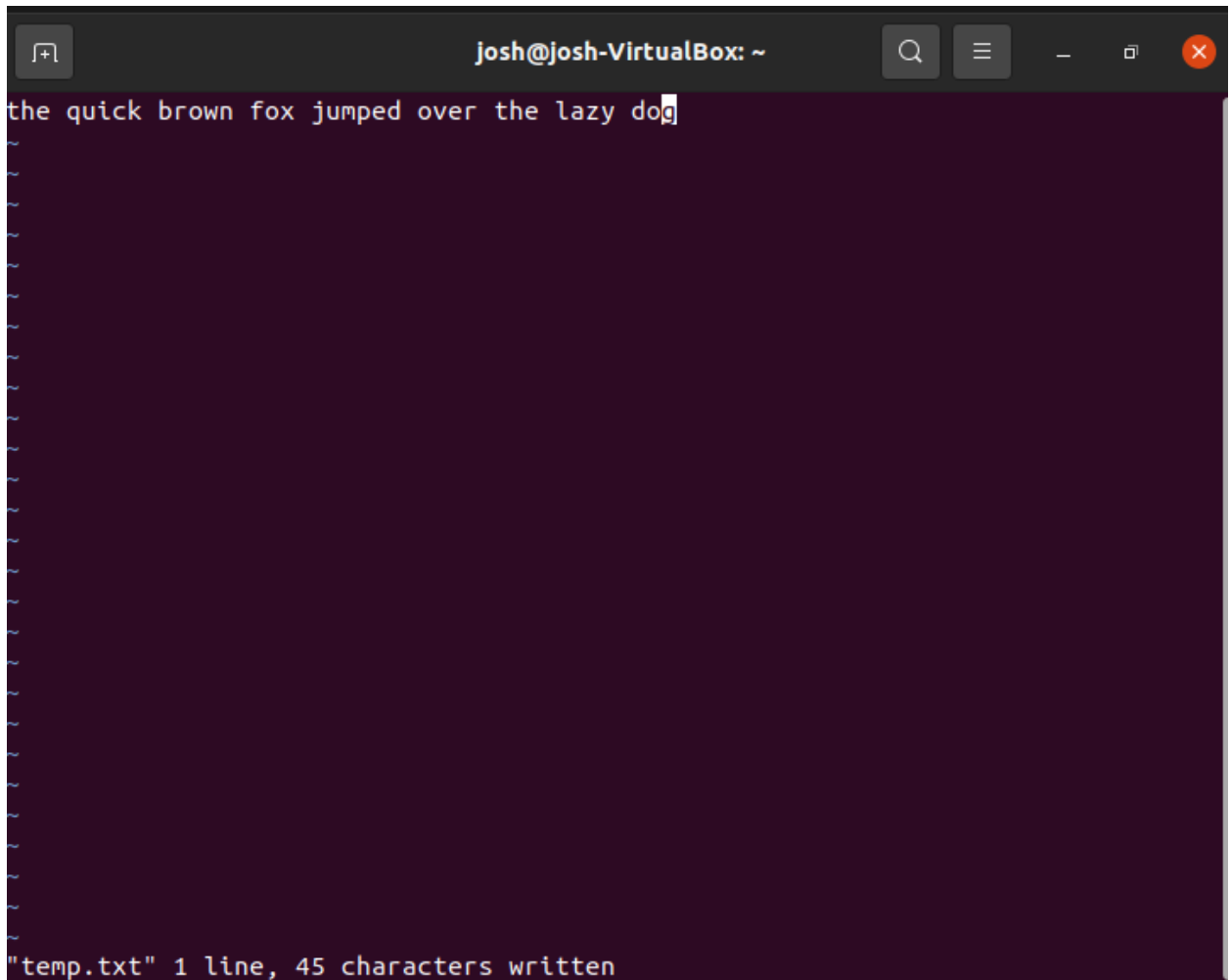
```
josh@josh-VirtualBox:~$ sudo fdisk /dev/sda  
  
Welcome to fdisk (util-linux 2.34).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Command (m for help): p  
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors  
Disk model: VBOX HARDDISK  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disklabel type: dos  
Disk identifier: 0x3d67674f  
  
Device      Boot  Start      End  Sectors  Size Id Type  
/dev/sda1   *      2048   1050623   1048576   512M  b W95 FAT32  
/dev/sda2             1052670  52426751  51374082  24.5G  5 Extended  
/dev/sda5             1052672  52426751  51374080  24.5G  83 Linux  
  
Command (m for help):
```

- i) apt: apt is the CLI package manager for Linux. In combination with sudo, you can install and update packages.

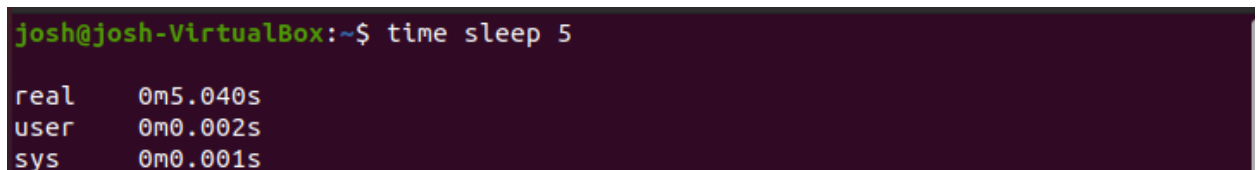


```
josh@josh-VirtualBox: ~  
josh@josh-VirtualBox:~$ sudo apt install sl  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  sl  
0 upgraded, 1 newly installed, 0 to remove and 300 not upgraded.  
Need to get 12.7 kB of archives.  
After this operation, 60.4 kB of additional disk space will be used.  
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 sl amd64 5.02-1  
[12.7 kB]  
Fetched 12.7 kB in 0s (44.9 kB/s)  
Selecting previously unselected package sl.  
(Reading database ... 182240 files and directories currently installed.)  
Preparing to unpack .../archives/sl_5.02-1_amd64.deb ...  
Unpacking sl (5.02-1) ...  
Setting up sl (5.02-1) ...  
Processing triggers for man-db (2.9.1-1) ...  
josh@josh-VirtualBox:~$
```

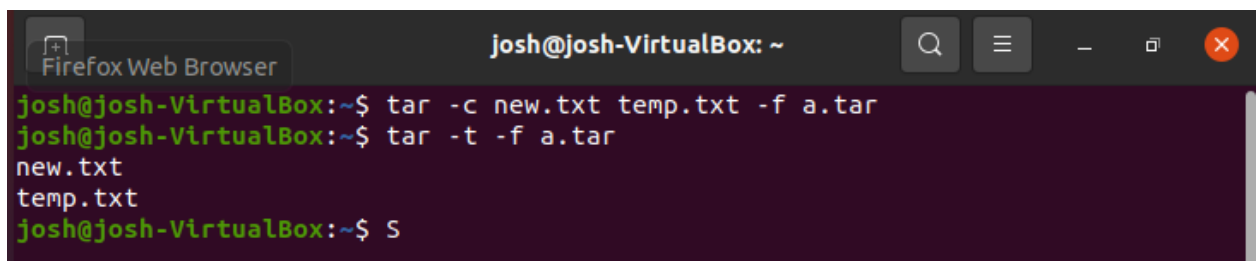
- j) vi: This command opens a visual editor for a file, with multiple commands to navigate around the file and make changes.



- k) time: This command times the resource usage of a command when executed.



- l) tar: This command provides access to the tarball archiving utility, with multiple subcommands to create or extract from archives.



- m) cat: This command outputs the contents of a file to stdout.

```
josh@josh-VirtualBox:~$ cat temp.txt
the quick brown fox jumped over the lazy dog
josh@josh-VirtualBox:~$
```

- n) watch: This command opens a new screen and periodically executes the command given as argument, displaying the output. By default, it does this every 2 seconds.

```
Every 2.0s: date                                josh-VirtualBox: Fri May 31 15:02:45 2024
Fri 31 May 2024 03:02:45 PM CDT
```

- o) ps: This displays a list of processes, along with their PIDs.

```
josh@josh-VirtualBox:~$ ps
  PID TTY          TIME CMD
 6380 pts/0    00:00:00 bash
 7280 pts/0    00:00:00 sleep
 7282 pts/0    00:00:00 sleep
 7283 pts/0    00:00:00 sleep
 7285 pts/0    00:00:00 ps
josh@josh-VirtualBox:~$
```

- p) top: This command outputs a list of running processes and their system information in real time, by default sorted by CPU consumption.

```
top - 15:06:04 up 3:31, 1 user, load average: 0.04, 0.04, 0.04
Tasks: 257 total, 1 running, 253 sleeping, 3 stopped, 0 zombie
%Cpu(s): 1.3 us, 0.8 sy, 0.0 ni, 97.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3912.1 total, 1996.4 free, 1083.0 used, 832.7 buff/cache
MiB Swap: 1162.4 total, 1162.4 free, 0.0 used, 2596.7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5989	josh	20	0	4163472	323644	122708	S	2.6	8.1	0:19.85	gnome-++
5820	josh	20	0	251016	68576	43700	S	1.6	1.7	0:10.53	Xorg
6294	josh	20	0	818044	52388	39356	S	0.6	1.3	0:04.58	gnome-++
6704	root	20	0	0	0	0	I	0.3	0.0	0:00.33	kworker+
7310	josh	20	0	11992	4008	3236	R	0.3	0.1	0:00.01	top
1	root	20	0	169480	12984	8460	S	0.0	0.3	0:03.02	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthrea+
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_pa+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_f+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
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_per+
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_ta+
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_ta+
13	root	20	0	0	0	0	S	0.0	0.0	0:00.36	ksofti+
14	root	20	0	0	0	0	I	0.0	0.0	0:00.65	rcu_sc+
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.03	migrat+
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_i+
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
20	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_i+

- q) htop: This command is similar to top but allows the user to scroll horizontally and vertically.

```

1 [ 0.0%] Tasks: 182, 413 thr; 1 running
2 [ 0.5%] Load average: 0.05 0.06 0.04
Mem[||||||| 1.09G/3.82G] Uptime: 03:34:54
Swp[ 0K/1.14G]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
 5820 josh        20   0  245M 68576 43700 S   1.0  1.7   0:11.41 /usr/lib/xorg/
8071 josh        20   0 10756  3996  3308 R   0.5  0.1   0:00.26 htop
6294 josh        20   0   798M 52388 39356 S   0.5  1.3   0:05.01 /usr/libexec/g
5847 josh        20   0   245M 68576 43700 S   0.5  1.7   0:01.19 /usr/lib/xorg/
6097 josh        20   0   557M 30900 20648 S   0.5  0.8   0:00.29 /usr/libexec/g
5989 josh        20   0 4073M  316M  119M S   0.0  8.1   0:21.31 /usr/bin/gnome
6004 josh        20   0 4073M  316M  119M S   0.0  8.1   0:03.11 /usr/bin/gnome
  643 syslog     20   0   219M  5044  3748 S   0.0  0.1   0:00.39 /usr/sbin/rsys
5861 josh       -6   0 1631M 19888 15376 S   0.0  0.5   0:03.90 /usr/bin/pulse
5765 josh        9  -11 1631M 19888 15376 S   0.0  0.5   0:04.17 /usr/bin/pulse
6005 josh        20   0 4073M  316M  119M S   0.0  8.1   0:03.47 /usr/bin/gnome
    1 root        20   0   165M 12984  8460 S   0.0  0.3   0:03.03 /sbin/init spl
  235 root        19  -1 52828 23388 21268 S   0.0  0.6   0:01.43 /lib/systemd/s
  260 root        20   0   2496   508   440 S   0.0  0.0   0:00.00 bpfilter_umh
  584 systemd-t   20   0 90912  6116  5328 S   0.0  0.2   0:00.00 /lib/systemd/s
  576 systemd-t   20   0 90912  6116  5328 S   0.0  0.2   0:00.10 /lib/systemd/s
  618 root        20   0   236M  9408  8408 S   0.0  0.2   0:00.14 /usr/lib/accou
  651 root        20   0   236M  9408  8408 S   0.0  0.2   0:00.03 /usr/lib/accou
  613 root        20   0   236M  9408  8408 S   0.0  0.2   0:00.25 /usr/lib/accou
  614 root        20   0   2548   768   704 S   0.0  0.0   0:00.24 /usr/sbin/acpi
  617 avahi       20   0   8528  3584  3260 S   0.0  0.1   0:00.66 avahi-daemon:
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

```

- r) gcc: This command is used to compile, assemble, and link files. Different options can be used to run specific parts of this process.

```

josh@josh-VirtualBox:~$ gcc temp.c
josh@josh-VirtualBox:~$ ls
a.out Desktop Downloads new.txt Public Templates Videos
a.tar Documents Music Pictures temp.c temp.txt
josh@josh-VirtualBox:~$ ./a.out
hello world
josh@josh-VirtualBox:~$ gcc -c temp.c
josh@josh-VirtualBox:~$ ls
a.out Desktop Downloads new.txt Public Templates temp.txt
a.tar Documents Music Pictures temp.c temp.o Videos
josh@josh-VirtualBox:~$

```

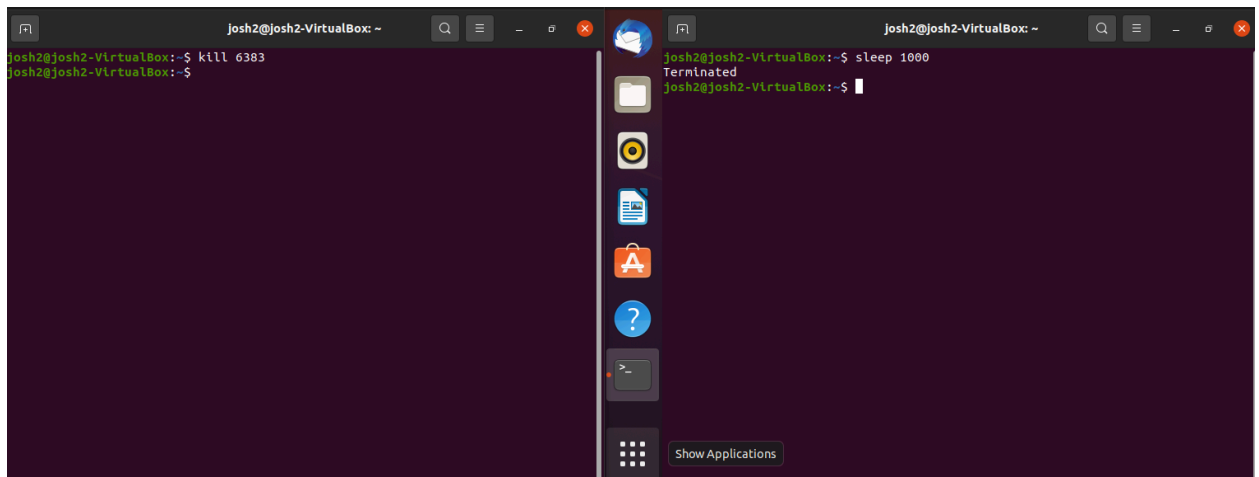
- s) tail: This command outputs the last n lines (10 by default) of a given file.

```
josh@josh-VirtualBox:~$ tail alphabet.txt
q
r
s
t
u
v
w
x
y
z
josh@josh-VirtualBox:~$
```

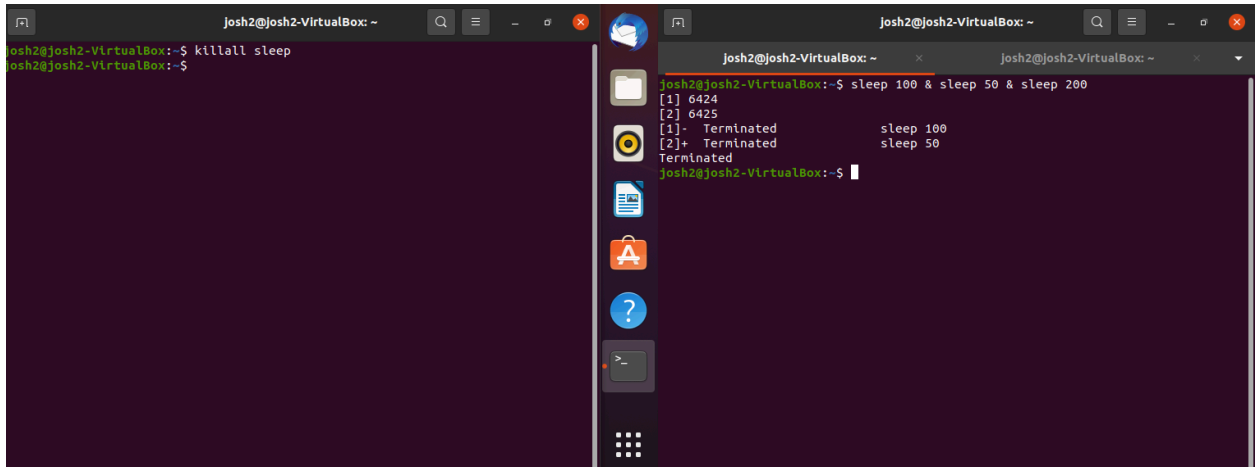
- t) grep: This command outputs all strings that match a given pattern or exact match in a target file or set of files.

```
josh@josh-VirtualBox:~$ grep "quick" temp.txt
the quick brown fox jumped over the lazy dog
josh@josh-VirtualBox:~$
```

- u) kill: This command sends the SIGTERM signal to a target process at the input PID.



- v) killall: This command sends SIGTERM to all processes running the input command.



The image shows two terminal windows from a virtual machine named 'josh2-VirtualBox'. The left window shows the command 'killall sleep' being executed. The right window shows a sequence of commands: 'sleep 100 & sleep 50 & sleep 200', which runs three background processes. It then shows the process IDs [1] 6424 and [2] 6425, followed by 'killall sleep', which terminates all three processes, as indicated by the 'Terminated' status for each.

```
josh2@josh2-VirtualBox: ~  
josh2@josh2-VirtualBox:~$ killall sleep  
josh2@josh2-VirtualBox:~$  
  
josh2@josh2-VirtualBox:~$ sleep 100 & sleep 50 & sleep 200  
[1] 6424  
[2] 6425  
[1]- Terminated sleep 100  
[2]+ Terminated sleep 50  
Terminated  
josh2@josh2-VirtualBox:~$
```

- w) du: This command outputs the amount of space each file is taking up in bytes. The example shown adds the max-depth argument, which limits the depth of files that get output.



The image shows a terminal window with the command 'du --max-depth=1' being executed. The output lists the size in bytes for various directories in the home directory, limited to a depth of 1. The total size for all listed directories is 9784 bytes.

```
josh2@josh2-VirtualBox:~$ du --max-depth=1  
172      ./config  
8820     ./cache  
20       ./ssh  
4        ./Documents  
4        ./Music  
704      ./local  
4        ./Downloads  
16       ./gnupg  
4        ./Templates  
4        ./Public  
4        ./Videos  
4        ./Pictures  
4        ./Desktop  
9784     .  
josh2@josh2-VirtualBox:~$
```

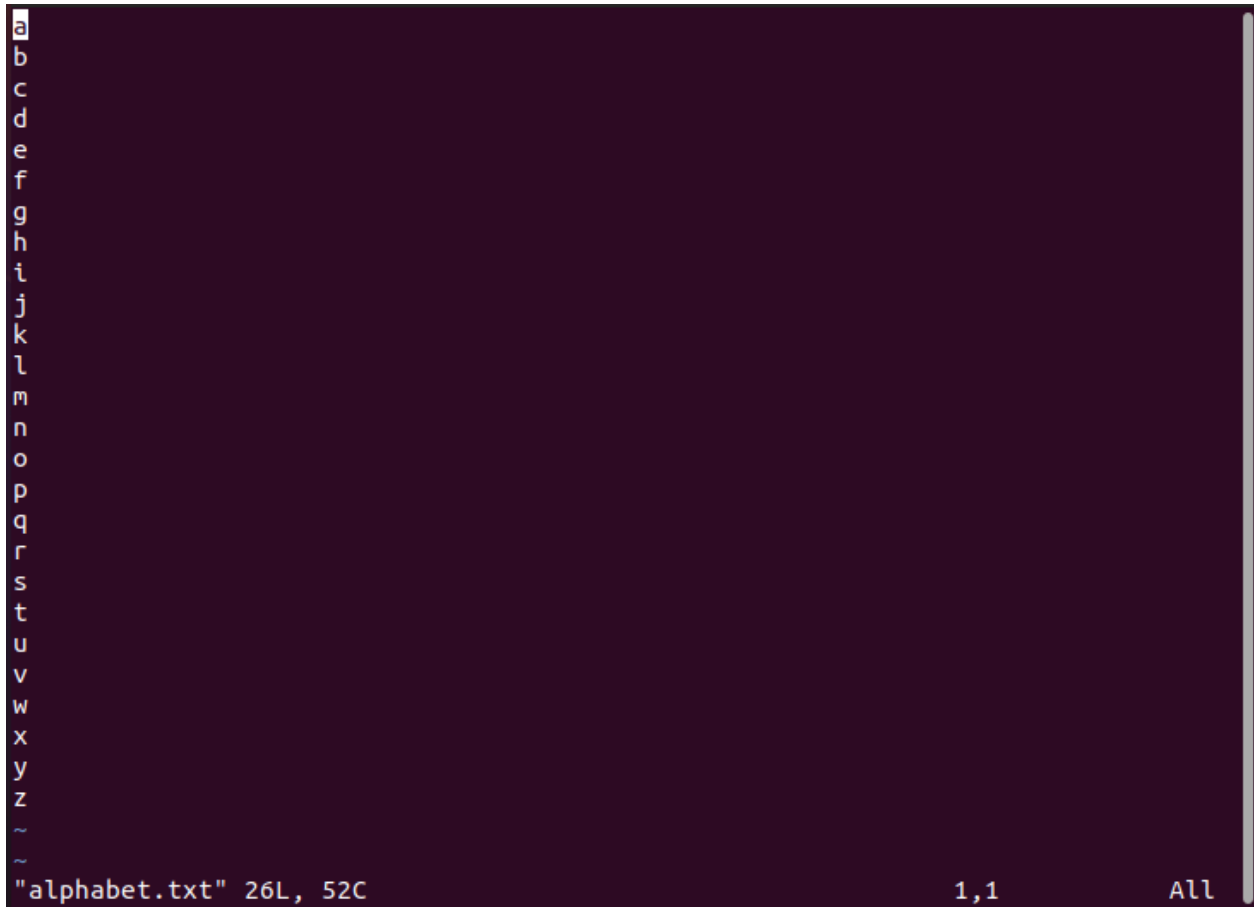
- x) df: This command outputs the amount of disk space used/available on each file system.

```
josh2@josh2-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1962564         0   1962564  0% /dev
tmpfs           400628      1544   399084  1% /run
/dev/sda5       25107716 9303616 14503364 40% /
tmpfs           2003124         0   2003124  0% /dev/shm
tmpfs            5120          4     5116  1% /run/lock
tmpfs           2003124         0   2003124  0% /sys/fs/cgroup
/dev/loop0        128        128         0 100% /snap/bare/5
/dev/loop1       64896     64896         0 100% /snap/core20/1828
/dev/loop2      354688   354688         0 100% /snap/gnome-3-38-2004/119
/dev/loop3       93952     93952         0 100% /snap/gtk-common-themes/1535
/dev/loop4       51072     51072         0 100% /snap/snapd/18357
/dev/loop5       47104     47104         0 100% /snap/snap-store/638
/dev/sda1        523248          4   523244  1% /boot/efi
tmpfs           400624         24   400600  1% /run/user/126
tmpfs           400624         28   400596  1% /run/user/1000
josh2@josh2-VirtualBox:~$
```

- y) screen: This command creates a new screen instance, in which you can modify the current window into subinstances, etc.

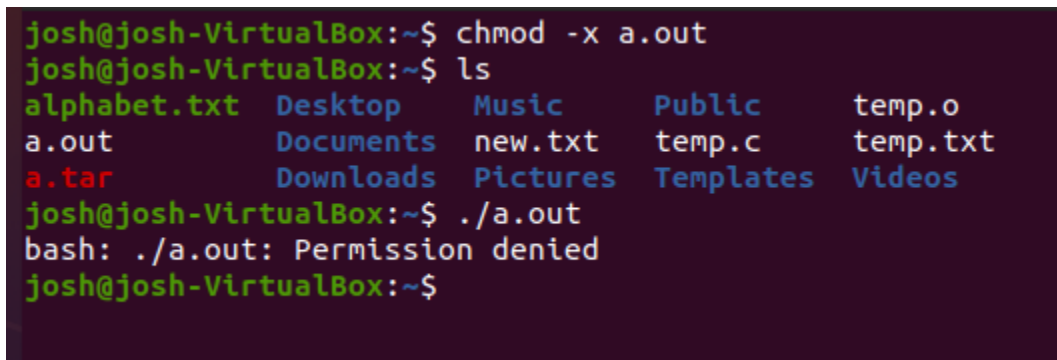


- z) vim: This command launches a new instance of vim, which seems to just be vi but a little bit better.



The screenshot shows a vim editor window with a dark purple background. On the left side, the lowercase alphabet 'a' through 'z' is listed vertically. At the bottom left, the status bar shows '"alphabet.txt" 26L, 52C'. At the bottom right, it shows '1,1' and 'All'.

- aa) chmod: This command lets you change file permissions. You can make things executable, change read/write access, etc. The below command removed executable privileges from a.out



The screenshot shows a terminal window with a dark purple background. The commands and output are as follows:

```
josh@josh-VirtualBox:~$ chmod -x a.out
josh@josh-VirtualBox:~$ ls
alphabet.txt  Desktop      Music        Public        temp.o
a.out         Documents   new.txt      temp.c        temp.txt
a.tar         Downloads   Pictures     Templates    Videos
josh@josh-VirtualBox:~$ ./a.out
bash: ./a.out: Permission denied
josh@josh-VirtualBox:~$
```


bb) chown: This command lets you modify the owner of a given file.

```
josh@josh-VirtualBox:~$ sudo chown josh a.out
josh@josh-VirtualBox:~$ ls -l a.out
-rwxrwxr-x 1 josh josh 16696 May 31 15:15 a.out
josh@josh-VirtualBox:~$
```

cc) useradd: This command creates a new user with the given username.

```
josh@josh-VirtualBox:~$ sudo useradd josh3
josh@josh-VirtualBox:~$
```

dd) man: This command opens the manual pages for a given command, displaying information about it.

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

    The table below shows the section numbers of the manual followed by the types of pages they contain.

    1    Executable programs or shell commands
    2    System calls (functions provided by the kernel)
    Manual page man(1) line 1 (press h for help or q to quit)
```

ee) locate: This command locates files matching a given pattern in the database of files.

```
josh@josh-VirtualBox:~$ locate *hello*
/boot/grub/i386-pc/hello.mod
/home/josh/Desktop/hello
/snap/core20/1828/usr/lib/python3.8/__phello__.foo.py
/snap/core20/1828/usr/lib/python3.8/__pycache__/__phello__.foo.cpython-38.pyc
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/helloworld
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/pythonhell
o
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/helloworld
/helloworld.plugin
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/helloworld
/libhelloworld.so
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/pythonhell
o/pythonhello.plugin
/snap/gnome-3-38-2004/119/usr/lib/x86_64-linux-gnu/peas-demo/plugins/pythonhell
o/pythonhello.py
/usr/lib/grub/i386-pc/hello.mod
/usr/lib/python3.8/__phello__.foo.py
/usr/lib/python3.8/__pycache__/__phello__.foo.cpython-38.pyc
/usr/share/doc/python3-uno/demo/hello_world_comp.py
/usr/share/doc/syslinux-common/asciidoc/hello.txt
/usr/share/locale-langpack/en@boldquot/LC_MESSAGES/hello.mo
/usr/share/locale-langpack/en@quot/LC_MESSAGES/hello.mo
/usr/share/locale-langpack/en_AU/LC_MESSAGES/hello.mo
/usr/share/locale-langpack/en_CA/LC_MESSAGES/hello.mo
/usr/share/locale-langpack/en_GB/LC_MESSAGES/hello.mo
josh@josh-VirtualBox:~$
```

ff) find: This command finds files matching a given pattern in a series of directories/paths given.

```
josh@josh-VirtualBox:~$ find . -name *hello*
./Desktop/hello
josh@josh-VirtualBox:~$
```

gg) sed: This command functions as a stream editor, allowing you to efficiently find and replace matching text in a file.

```
josh@josh-VirtualBox:~$ sudo vim alphabet.txt
josh@josh-VirtualBox:~$ sudo sed 's/a/b/' alphabet.txt
b
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
josh@josh-VirtualBox:~$
```

hh) awk: This command runs an awk program on a given file input. The below program prints all lines that contain the pattern 'b'.

```
josh@josh-VirtualBox:~$ awk '/b/ {print}' alphabet.txt
b
josh@josh-VirtualBox:~$
```

ii) diff: This command outputs the difference between two files.

```
josh@josh-VirtualBox:~$ diff alphabet.txt newalpha.txt
6d5
< f
7a7
> f
10,11c10,11
< j
< k
---
> i
> n
josh@josh-VirtualBox:~$ s
```

jj) sort: This command sorts lines of text in a file alphabetically.

```
josh@josh-VirtualBox:~$ sort newalpha.txt
a
b
c
d
e
f
g
h
i
i
l
m
n
n
o
p
q
r
s
t
u
v
w
x
y
z
josh@josh-VirtualBox:~$
```

kk) export: This command outputs all environment variables and their values.

```
josh@josh-VirtualBox:~$ 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=":1"
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"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/ecb82bfb_c114_44a6_91f0_1e7391b399be"
declare -x GNOME_TERMINAL_SERVICE=":1.144"
declare -x GPG_AGENT_INFO="/run/user/1000/gnupg/S.gpg-agent:0:1"
declare -x GTK_MODULES="gail:atk-bridge"
declare -x HOME="/home/josh"
declare -x IM_CONFIG_PHASE="1"
declare -x INVOCATION_ID="23f31922433d43acb44705002ae7fd72"
declare -x JOURNAL_STREAM="8:76099"
declare -x LANG="en_US.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="josh"
declare -x LS_COLORS="rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01
```

ll) pwd: This command prints the full current working directory.

```
josh@josh-VirtualBox:~$ pwd
/home/josh
josh@josh-VirtualBox:~$
```

mm) crontab: This command has multiple subcommands related to crontab scheduling. The below shows an example of setting up a command to run every week at 5am.

```
GNU nano 4.8 /tmp/crontab.G604gy/crontab Modified
0 5 * * 1 touch newfile
# 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)
#
# m h dom mon dow  command
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit        ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell
```

nn) mount: This command has multiple subcommands relating to mounting drives. The below commands mount all available drives and list the current mounted drives.

```
josh@josh-VirtualBox:~$ sudo mount -a
josh@josh-VirtualBox:~$ mount -l
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=1962444k,nr_inodes=490611,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=400604k,mode=755,inode64)
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755,inode64)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
```

oo) passwd: This command allows you to change the password for the current user.

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

pp) uname: This command prints information about the current system, including the OS, time, and name.

```
josh@josh-VirtualBox:~$ uname -a
Linux josh-VirtualBox 5.15.0-107-generic #117~20.04.1-Ubuntu SMP Tue Apr 30 10:35:57 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
josh@josh-VirtualBox:~$
```

qq) whereis: This command shows the location of the binary, source, and man pages for a given command.

```
josh@josh-VirtualBox:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
josh@josh-VirtualBox:~$
```

rr) whatis: This command displays one-line information about a given command.

```
josh@josh-VirtualBox:~$ whatis ls
ls (1)          - list directory contents
LS (6)          - display animations aimed to correct users who acciden...
```

ss) su: This command lets you switch to a target user.

```
josh@josh-VirtualBox:~$ su josh3
Password:
$ whoami
josh3
$
```

tt) ping: This command pings another machine at a target address.

```
josh@josh-VirtualBox:~$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=1.36 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=0.886 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=1.33 ms

--- 10.0.2.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2109ms
rtt min/avg/max/mdev = 0.886/1.189/1.355/0.214 ms
^Cjosh@josh-VirtualBox:~$
```

uu) traceroute: This command outputs the paths taken by packets from a source to destination.

```
josh@josh-VirtualBox:~$ traceroute google.com
traceroute to google.com (142.250.191.206), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.1)  0.378 ms  0.389 ms  0.583 ms
 2 * * *
```

vv) date: This command outputs the current date.

```
josh@josh-VirtualBox:~$ date
Fri 31 May 2024 04:47:44 PM CDT
josh@josh-VirtualBox:~$
```


ww) time: This command times the resource usage of a command when executed.

```
josh@josh-VirtualBox:~$ time sleep 5

real    0m5.040s
user    0m0.002s
sys     0m0.001s
```

xx) wget: This command allows the user to download files from a target address on the web.

```
josh@josh-VirtualBox:~$ wget -m google.com
--2024-05-31 16:50:12-- http://google.com/
Resolving google.com (google.com)... 142.250.191.206, 2607:f8b0:4009:81a::200e
Connecting to google.com (google.com)|142.250.191.206|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2024-05-31 16:50:12-- http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.191.164, 2607:f8b0:4009:819::2004
Connecting to www.google.com (www.google.com)|142.250.191.164|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'google.com/index.html'

google.com/index.ht  [ <=>          ] 19.45K  --.-KB/s   in 0s

Last-modified header missing -- time-stamps turned off.
2024-05-31 16:50:12 (265 MB/s) - 'google.com/index.html' saved [19920]

FINISHED --2024-05-31 16:50:12--
Total wall clock time: 0.5s
Downloaded: 1 files, 19K in 0s (265 MB/s)
```

yy) wc: This command counts the number of words in a file and outputs to stdout.

```
josh@josh-VirtualBox:~$ wc alphabet.txt
26 26 52 alphabet.txt
josh@josh-VirtualBox:~$
```


zz) pwgen: This command generates multiple random passwords of a given length (default 8)

```
josh@josh-VirtualBox:~$ pwgen
thi00oto ohs7ohSu Ooph9chi aiBii4oo eK4EeHae AidohC1u Die1pha3 iNg2shoo
be8gooFa soW4geer Lah2vahP Ieph6eiw Tae0shiF xeed5Ezu Equi5zei ahphaeT6
voqua4Lo uChahHi9 AiShei4o ocheit9A jaSoo9ax LoC4ji8n Yie3Uree zei2aiTu
eiShe7Fo nu0ahGhe Ach7ivoo Aitee7Uc ok3Xiena ahT5ahTh audo6Pha aeTh3AiY
eirau0Ee iim0ohD3 Coad4ohx vo3Ongog De8ahM4c choh8iiH ieb9wiej Aehaing7
fahZ4aen iBoo0cha maiThoo6 ahg8teJo Sahquae0 toh4Phie yah8ReiG neip8Ii9
xoh5Eipo Atav3too ya1ieV1u ukuo1Eix eiGh8Aef bukeen5E Ji3vaeHo zei9oWu2
EiYoo7Pi ooLu6rou Feeghei5 Ahy1ohth ReeF7koo feb6Aequ aiNg4Poo Pohwee2o
XeiGh9ie fech4Ue5 oCahl5sh eiCheix8 Oli6Jie cho3Toos oy50iM6h Aik0goh0
phei7ohB eiQu4AhN eSie0ohf shae1Aiw agh7aiVu oz20suka Aefai2Ae ahVeis40
uoNg0dee Xu8oodah Vo0pheem quohta6L ohSei8Wi keKi0eeC oonih9Ei ith7Ahti
pe6ieYoo Ou9hooy6 Ieth1Tee Eehee6ah Gomoo9fa eH5ceec7 ita2ooNg Ashi0uko
Eike8kee eiPh0eaM pee6uMoo Thei1eir WawooX4k ahl6uGh2 foh7Saec ahp1fi7U
Yeep6Que bee7Ei2i ThaiM8Ie dae4ohS6 AiRai1du bohziQu7 Av6aegho kohphi7I
ahPomah1 vieKee3H Eik7ahyi kieM6Eew OochioM8 uch8Mejo ieCh3tho cahH7thu
eiHei6ph OhPhoo5a muapah3Z ainie5Ri bohP8Eip Aenae5Ae uaCo8ho1 ieh8ueCh
aeTiey6a nee0ufec thaing3M ooVohNg9 axo0Ighe Ti8iFein thaeV2yi ieluo5Th
aeG3dae8 Eeja5cho raMu4che Ahth8rah aeWe2Hei Moo4etao bahbo8Eg queeCh8h
Ea7doo1e ohn3Aeph aiS6zooD ie8oBaew eeCh3tae Noo6ce4e ohcuTh4Y AigePh5y
ot9Ielei eiBah4im xu3ahZai Oobai1uu ziem9Dep ahk5Ne4u DooGhee7 reph6ciM
```

3. Bash Scripts

```
1000 records created.

real    0m0.016s
user    0m0.009s
sys     0m0.007s
100000 records created.

real    0m1.315s
user    0m1.256s
sys     0m0.032s
10000000 records created.

real    2m5.702s
user    2m0.180s
sys     0m4.946s
```

```
josh@josh-VirtualBox:~/linux-homework$ bash sort-data.sh onethousand.txt sort1000.txt
real    0m0.038s
user    0m0.000s
sys     0m0.003s
josh@josh-VirtualBox:~/linux-homework$ bash sort-data.sh hundredthousand.txt sort100000.txt
real    0m0.104s
user    0m0.070s
sys     0m0.023s
josh@josh-VirtualBox:~/linux-homework$ bash sort-data.sh tenmillion.txt sort10000000.txt
real    0m17.720s
user    0m13.104s
sys     0m8.960s
```

4. Answers to Questions

- a) Setting the processor count to a high number will likely increase performance, while a lower number will decrease performance. However, this performance comes at an energy cost, as we see with Bitcoin mining consuming massive amounts of energy today. In addition, the host machine will likely suffer performance impact as well, since if most of the cores are dedicated to the VM, then the host has less to work with, making this potentially a bad idea. You'd likely want to set this to the minimum for lightweight, non-intensive tasks, like a basic script, and you'd want to set this to the maximum for intensive tasks that require a lot of compute power, like machine learning model training.
- b) Paravirtualization options allow the user to change the hypervisor interface.
 - i) None - paravirtualization is disabled
 - ii) Legacy - paravirtualization is minimally supported using legacy interfaces (lacks modern OS support)

- iii) Minimal - paravirtualization is minimally supported using modern interfaces
 - iv) HyperV - paravirtualization through Microsoft Hyper-V hypervisor interface
 - v) KVM - paravirtualization through Linux KVM hypervisor interface
 - vi) Out of these, the best is likely KVM because Ubuntu is a Linux distro
- c) Storage Devices:
- i) IDE - A user would likely want to use IDE for a legacy system that only runs on older hardware
 - ii) SATA - A user would want to use SATA in most cases or for general use, like running an application at normal load.
 - iii) NVMe - A user would want to use NVMe for high performance things like ML models or databases
- d) Network Types:
- i) NAT - This allows users to access the internet on the VM with an internal IP without connecting to other machines. This could be used for developers who want to install dependencies but not reach other devices.
 - ii) Bridged Server - This is for when a user wants to fully access something on the host machine's local network, like if someone were running a web server on the VM.
 - iii) Internal Network - This could be used if someone were testing out a network application with users and servers, since outside connection is not necessary.
 - iv) Host-Only Network - This could be used if someone were developing software for communication between a host machine and the VM, since it only opens the connection between the two of them.
- e) USB:
- i) 1.1 - This is a legacy hardware option with slower data transfer, designed for low bandwidth devices. This is universally supported by every machine.
 - ii) 2.0 - This is backwards compatible with 1.1 and has improvements on all aspects of 1.1. It's typically used for moderate bandwidth devices
 - iii) 3.0 - This provides backwards compatibility with both 1.1 and 2.0, and has even higher power and data transfer speeds.