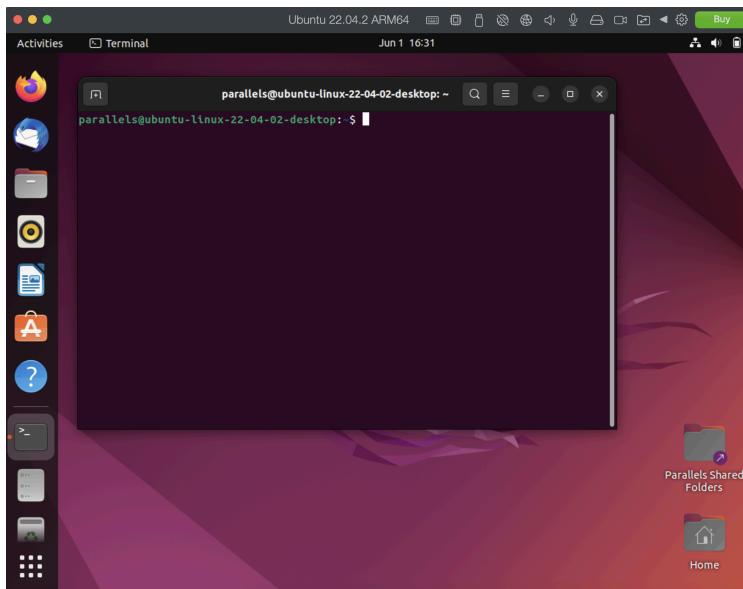


1)

*Note = VirtualBox is not supported on Apple M1 Chip so I used Parallels

Virtual Machine (Parallels, Ubuntu-linux-22.04.2) :



My Terminal:

```
Last login: Fri May 31 15:16:32 on ttys001
[(base) anagharam@Anaghlas-MacBook-Air ~ % cd
[(base) anagharam@Anaghlas-MacBook-Air ~ % ls
Applications           Downloads          PycharmProjects
AugmentedLangarian     Library           Zotero
DataGripProjects       Movies            anaconda3
Demo                  Music             hehe
Desktop                Pictures          vim
Documents              Public
(base) anagharam@Anaghlas-MacBook-Air ~ %
```

Public/Private rsa Key Pair:

```
parallels@ubuntu-linux-22-04-02-desktop:~$ mkdir -p $HOME/.ssh
parallels@ubuntu-linux-22-04-02-desktop:~$ chmod 0700 $HOME/.ssh
parallels@ubuntu-linux-22-04-02-desktop:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/parallels/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/parallels/.ssh/id_rsa
Your public key has been saved in /home/parallels/.ssh/id_rsa.pub
The key's fingerprint is:
SHA256:gKyLjTIRQUVY3W/tUjqDc9sTOuo/Z265ysQHRU+4K6k parallels@ubuntu-linux-22-04-02-desktop
The key's randomart image is:
+---[RSA 3072]---+
|ooo+ . . . .
| .o. o . .o .
| . o . . . . .
| . . . . o.+ |
| .. S.= .
| = o . o.+ |
|= o . +oO.o |
|.. Eo=.B |
| .oo+ooo |
+---[SHA256]---+
```

SSH connection from Virtual Machine to my Terminal:

```
parallels@ubuntu-linux-22-04-02-desktop:~/.ssh$ ssh-copy-id -i /home/parallels/.ssh/id_rsa.pub anagharam@Anaghas-MacBook-Air.local
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/parallels/.ssh/id_rsa.pub"
The authenticity of host 'anaghas-macbook-air.local (10.211.55.2)' can't be established.
ED25519 key fingerprint is SHA256:pGAAdwRq/sUNwfryRvKICjP2XXRo1BjuzuGpi/CuMTzc.
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
(anagharam@anaghas-macbook-air.local) Password:
(anagharam@anaghas-macbook-air.local) Password:
(anagharam@anaghas-macbook-air.local) Password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'anagharam@Anaghas-MacBook-Air.local'"
and check to make sure that only the key(s) you wanted were added.

parallels@ubuntu-linux-22-04-02-desktop:~/.ssh$ ssh 'anagharam@Anaghas-MacBook-Air.local'
Last login: Sat Jun  1 16:41:15 2024
(base) anagharam@Anaghas-MacBook-Air ~ %
```

2)

- a) ssh: protocol for establishing communication and sharing of data between two machines
*screenshot of command already shown in 1)
- b) ssh- keygen: generates public/private key pair; use to access remote machine
*screenshot of command already shown in 1)
- c) scp: copies a file from remote to local given a source & destination

```
anagharam@Anaghas-MacBook-Air Desktop % scp parallels@ubuntu-linux-22-04-02-desktop:/home/parallels/.ssh/parallels.txt anagharam@Anaghas-MacBook-Air.local/Users/anagharam/Desktop
```

- d) history: list of linux commands used before

```
(base) anagharam@Anaghas-MacBook-Air Desktop % history
175 vi id_rsa.pub
176 vi known_host
177 ssh ubuntu-linux-22-04-02-desktop\n
178
179 mv id_rsa.pub id_rsa.pub.bkp
180 hostname
181 ping Anaghas-MacBook-Air.local
182 pwd
183 ls
184 cd ..
185 ls
186 cd Desktop
187 pwd
188 scp anagharam@Anaghas-MacBook-Air.local@parallels@ubuntu-linux-22-04-02-desktop:/home/parallels/.ssh/parallels.txt /Users/anagharam/Desktop\n
189 scp parallels@ubuntu-linux-22-04-02-desktop:/home/parallels/.ssh/parallels.txt anagharam@Anaghas-MacBook-Air.local@Users/anagharam/Desktop
190 scp parallels@ubuntu-linux-22-04-02-desktop:/home/parallels/.ssh/parallels.txt anagharam@Anaghas-MacBook-Air.local@Users/anagharam/Desktop\n
```

- e) sudo: allows user to execute commands which have security privileges of a different user to execute.

```
anagharam@Anaghas-MacBook-Air Desktop % sudo apt install vim
```

- f) ip: ntwrk tool for monitoring & altering IP addresses, ntwrk interfaces, & other routing procedures. (Below command lists ntwrk interfaces).

```
(base) anagharam@Anaghas-MacBook-Air Desktop % ip addr show
```

- g) dd: copying/converting data

```
dd if=/dev/ones of=lottaones.bin bs=1M count=10
```

- h) fdisk: divides your hard-drive into multiple diff. Sections. Following tests if disk is partitioned

```
(base) anagharam@Anaghas-MacBook-Air desktop % Command (m for help): w
```

```
[$sudo fdisk -t /dev/disk3s1
fdisk: /dev/disk3s1: Resource busy
$
```

- i) apt: helps monitor/update packages (below command updates all existing packages)

```
(base) anagharam@Anaghas-MacBook-Air Desktop % sudo apt upgrade
[Password:
```

- j) vi: creating/altering file

```
(base) anagharam@Anaghas-MacBook-Air Desktop % vi newfile.txt
zsh: suspended vi newfile.txt
```

- k) time: measures length of execution time of a program

```
(base) anagharam@Anaghas-MacBook-Air Desktop % time ls
ls 0.00s user 0.01s system 14% cpu 0.077 total
```

- l) tar: used to modify/alter archived files

```
(base) anagharam@Anaghas-MacBook-Air ~ % tar -cf archive.tar ~/Desktop
tar: Removing leading '/' from member names
```

- m) cat: Inspects, concatenates, generates, copies, or alters content of file provided

```
(base) anagharam@Anaghlas-MacBook-Air Desktop % cat > social.txt
```

- n) watch: executes command and displays on S.O. ever 4 seconds in this case

```
watch -n 4 df -h .
```

- o) ps: tool which gives info on curr. running processes

```
(base) anagharam@Anaghlas-MacBook-Air ~ % ps
  PID TTY          TIME CMD
 1655 ttys000    0:00.03 -zsh
 4025 ttys001    0:00.54 -zsh
 10128 ttys001   0:00.05 vi newfile.txt
```

- p) top: displays list of processes currently running

```
(base) anagharam@Anaghlas-MacBook-Air ~ % top
(base) anagharam@Anaghlas-MacBook-Air ~ %
```

```
Processes: 341 total, 2 running, 339 sleeping, 2476 threads
Load Avg: 6.17, 3.87, 2.72 CPU usage: 6.22% user, 5.4% sys, 88.73% idle SharedLibs: 335M resident, 70M data, 14M linkedit. MemRegions: 452093 total, 1997M resident, 101M private, 1074M shared.
PhysMem: 7451M used (1331M wired, 2445M compressor), 178M unused. VM: 167T vsiz
e, 4283M framework vsiz
e, 2967256(0) swaps
ins, 3386909(0) swapouts. Networks: packets: 23708100/28G in, 3375263/1679M out.
Disks: 11923198/286G read, 3982619/127G written.
```

- q) htop: similar to top but with more visuals such as trees (also more color)

```
(base) anagharam@Anaghlas-MacBook-Air ~ % sudo apt-get install htop
(base) anagharam@Anaghlas-MacBook-Air ~ % htop
```

- r) gcc: compiles C programs

```
(base) anagharam@Anaghlas-MacBook-Air ~ % gcc -o output input.c
```

- s) tail: last specified amount of file (10 def.)

```
(base) anagharam@Anaghlas-MacBook-Air desktop % tail ~/Desktop/test.txt
hi
```

- t) grep: searches within text for specific words or strings

```
(base) anagharam@Anaghlas-MacBook-Air desktop % grep 'hi' test.txt
hi
```

- u) kill: terminates all specified programs via signal

```
(base) anagharam@Anaghlas-MacBook-Air ~ % ps
  PID TTY          TIME CMD
 1655 ttys000    0:00.03 -zsh
 4025 ttys001    0:00.54 -zsh
 10128 ttys001   0:00.05 vi newfile.txt
```

PID: 10128

```
((base) anagharam@Anaghas-MacBook-Air ~ % kill 10128  
(base) anagharam@Anaghas-MacBook-Air ~ % █
```

- v) Killall: kills all programs with the parameter name

```
(base) anagharam@Anaghas-MacBook-Air Desktop % killall Safari  
(base) anagharam@Anaghas-MacBook-Air Desktop % █
```

- w) Du: estimates file space usage

```
[[ec2-user@ip-172-31-31-15 ~]$ du ./uchicago  
3155032 ./uchicago  
[ec2-user@ip-172-31-31-15 ~]$ █
```

- x) df: shows used and available space in the file system

```
[[ec2-user@ip-172-31-31-15 ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        483M   60K  483M   1% /dev  
tmpfs           493M     0  493M   0% /dev/shm  
/dev/xvda1       30G   25G  4.4G  86% /  
[ec2-user@ip-172-31-31-15 ~]$ █
```

- y) screen: Manage multiple shell sessions from a single ssh session.

```
[ec2-user@ip-172-31-31-15 uchicago]$ screen -r  
There are several suitable screens on:  
 24095.pts-0.ip-172-31-31-15      (Detached)  
 24182.pts-0.ip-172-31-31-15      (Attached)  
 24129.pts-0.ip-172-31-31-15      (Detached)  
 23969.pts-0.ip-172-31-31-15      (Detached)  
Type "screen [-d] -r [pid.ltty.host]" to resume one of them.  
[ec2-user@ip-172-31-31-15 uchicago]$ █
```

- z) Vim: Text editor

```
[ec2-user@ip-172-31-31-15 uchicago]$ vim sort-data.sh █
```

```
#!/bin/bash
filename=$1
sort -n -k1,1 "$filename" -o "$filename.sortedoutput"
```

```
echo "Output file is $filename.sortedoutput"
```

~

~

aa) Chmod: change permissions of files

```
[ec2-user@ip-172-31-31-15 uchicago]$ chmod u+x sort-data.sh
[ec2-user@ip-172-31-31-15 uchicago]$ █
```

bb) Chown: changes ownership of files

```
[ec2-user@ip-172-31-31-15 uchicago]$ chown ec2-user sort-data.sh
[ec2-user@ip-172-31-31-15 uchicago]$ █
```

cc) Useradd: adds new user

```
[ec2-user@ip-172-31-31-15 uchicago]$ sudo useradd newuser
[ec2-user@ip-172-31-31-15 uchicago]$ █
```

dd) Man: shows manual for command operation

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 [-AknS] [-g group_name | #gid] [-p prompt] [-u user_name | #uid]
sudo -l[l] [-AknS] [-g group_name | #gid] [-p prompt] [-U user_name]
[-u user_name | #uid] [command]
sudo [-AbEHnPS] [-C fd] [-g group_name | #gid] [-p prompt] [-r role] [-t type]
[-u user_name | #uid] [VAR=value] -i | -s [command]
sudoedit [-Ans] [-C fd] [-g group_name | #gid] [-p prompt] [-u user_name | #uid] file ...
```

DESCRIPTION

sudo allows a permitted user to execute a **command** as the superuser or another user, as

```
[ec2-user@ip-172-31-31-15 uchicago]$ man sudo
```

ee) Locate: finds the file on the file system

```
$ locate elastic.pem
```

- ff) Find: tool to find files based on different characteristics of the file. E.g. find all files ending in .h

```
[ec2-user@ip-172-31-31-15 ~]$ find *.h  
my_config.h  
[ec2-user@ip-172-31-31-15 ~]$
```

- gg) Sed: Is the steam editor for altering file content. Following command changes 'one' to 'two' in example.txt

```
[ec2-user@ip-172-31-31-15 ~]$ sed 's/one/two/g' example.txt  
two  
[ec2-user@ip-172-31-31-15 ~]$
```

- hh) awk : language for pattern matching and processing. Following prints column two in example.txt

```
[ec2-user@ip-172-31-31-15 ~]$ awk '{print $2}' example.txt  
two1  
two2  
[ec2-user@ip-172-31-31-15 ~]$
```

- ii) Diff: compares files line by line

```
[ec2-user@ip-172-31-31-15 ~]$ diff example.txt example2.txt  
1,2c1,2  
< one two1  
< one two2  
---  
> one two3  
> one two4  
[ec2-user@ip-172-31-31-15 ~]$
```

- jj) Sort: sorts lines of files

```
[ec2-user@ip-172-31-31-15 ~]$ sort example.txt
one two1
one two2
[ec2-user@ip-172-31-31-15 ~]$ █
```

kk) Export: used to set environment variables

```
[ec2-user@ip-172-31-31-15 ~]$ set | fgrep PATH
AWS_PATH=/opt/aws
PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/aws/bin:/home/ec2-user/.local/bin:/home/ec2-user/bin:/opt/aws/bin:/opt/aws/bin
[ec2-user@ip-172-31-31-15 ~]$ export PATH=./:$PATH
[ec2-user@ip-172-31-31-15 ~]$ set | fgrep PATH
AWS_PATH=/opt/aws
PATH=./:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/aws/bin:/home/ec2-user/.local/bin:/home/ec2-user/bin:/opt/aws/bin:/opt/aws/bin
_=PATH
[ec2-user@ip-172-31-31-15 ~]$ █
```

ll) Pwd: displays current working directory

```
[ec2-user@ip-172-31-31-15 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-31-15 ~]$ █
```

mm) Crontab: command to schedule job to run at specified times. Following runs export.sh every day at 1:00 AM

```
0 1 * * * /home/ec2-user/dbbackup/export.sh
```

nn) Mount: can be used to mount a file system. Following shows the current mounts

```
[ec2-user@ip-172-31-31-15 ~]$ mount
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
devtmpfs on /dev type devtmpfs (rw,relatime,size=494140k,nr_inodes=123535,mode=755)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,relatime)
/dev/xvda1 on / type ext4 (rw,noatime,data=ordered)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)
```

oo) Passwd: changes user passwords

```
[ec2-user@ip-172-31-31-15 ~]$ passwd
Changing password for user ec2-user.
Changing password for ec2-user.
(current) UNIX password: █
```

pp) Uname: displays system information

```
[ec2-user@ip-172-31-31-15 ~]$ uname -a
Linux ip-172-31-31-15 4.14.67-66.56.amzn1.x86_64 #1 SMP Tue Sep 4 22:03:21 UTC 2018 x86_64 x86_64
x86_64 GNU/Linux
[ec2-user@ip-172-31-31-15 ~]$ █
```

qq) Whereis:

```
[ec2-user@ip-172-31-31-15 ~]$ whereis pwd
pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz /usr/share/man/man1p/pwd.1p.gz
[ec2-user@ip-172-31-31-15 ~]$ █
```

rr) Whatis: describes command briefly

```
[ec2-user@ip-172-31-31-15 ~]$ whatis pwd
pwd (1)           - print name of current/working directory
pwd (1p)          - return working directory name
[ec2-user@ip-172-31-31-15 ~]$ █
```

ss) Su: switches to another user account

```
[ec2-user@ip-172-31-31-15 ~]$ su newuser
Password: █
```

tt) Ping: checks if you can connect to a computer

```
[ec2-user@ip-172-31-31-15 ~]$ ping `hostname`  
PING ip-172-31-31-15.us-west-1.compute.internal (172.31.31.15) 56(84) bytes of data.  
64 bytes from ip-172-31-31-15.us-west-1.compute.internal (172.31.31.15): icmp_seq=1 ttl=255 time=  
0.012 ms  
64 bytes from ip-172-31-31-15.us-west-1.compute.internal (172.31.31.15): icmp_seq=2 ttl=255 time=  
0.026 ms  
64 bytes from ip-172-31-31-15.us-west-1.compute.internal (172.31.31.15): icmp_seq=3 ttl=255 time=  
0.027 ms  
64 bytes from ip-172-31-31-15.us-west-1.compute.internal (172.31.31.15): icmp_seq=4 ttl=255 time=  
0.026 ms  
^C  
--- ip-172-31-31-15.us-west-1.compute.internal ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3061ms  
rtt min/avg/max/mdev = 0.012/0.022/0.027/0.008 ms  
[ec2-user@ip-172-31-31-15 ~]$
```

uu) Traceroute: traces the route taken by packets to a host

```
[ec2-user@ip-172-31-31-15 ~]$ traceroute apple.com  
traceroute to apple.com (17.253.144.10), 30 hops max, 60 byte packets  
 1  240.0.168.14 (240.0.168.14)  0.975 ms 240.0.168.12 (240.0.168.12)  1.060 ms 1.206 ms  
 2  242.2.26.65 (242.2.26.65)  1.414 ms 242.2.27.193 (242.2.27.193)  1.398 ms 242.2.26.197 (242.2  
.26.197)  11.076 ms  
 3  15.230.28.86 (15.230.28.86)  11.290 ms 15.230.28.72 (15.230.28.72)  11.299 ms 15.230.28.84 (1  
5.230.28.84)  11.310 ms  
 4  17.1.128.76 (17.1.128.76)  1.077 ms 99.83.92.229 (99.83.92.229)  1.070 ms 17.1.128.74 (17.1.1  
28.74)  1.089 ms  
 5  www.brkcls.com (17.253.144.10)  1.957 ms !X  0.995 ms !X  1.382 ms !X  
[ec2-user@ip-172-31-31-15 ~]$
```

vv) Date: displays or changes system date

```
[ec2-user@ip-172-31-31-15 ~]$ date  
Mon Jun 3 08:39:54 UTC 2024  
[ec2-user@ip-172-31-31-15 ~]$
```

ww) Time: measures how long a command takes to execute

```
[ec2-user@ip-172-31-31-15 ~]$ time date  
Mon Jun 3 08:40:31 UTC 2024  
  
real      0m0.001s  
user      0m0.001s  
sys       0m0.000s  
[ec2-user@ip-172-31-31-15 ~]$
```

xx) Wget: downloads a file from the url location

```
[ec2-user@ip-172-31-31-15 ~]$ wget https://wordpress.org/latest.zip
--2024-06-03 08:41:59-- https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26193148 (25M) [application/zip]
Saving to: 'latest.zip'

latest.zip          100%[=====] 24.98M 19.7MB/s   in 1.3s

2024-06-03 08:42:01 (19.7 MB/s) - 'latest.zip' saved [26193148/26193148]

[ec2-user@ip-172-31-31-15 ~]$ yy) Wc: counts the number of lines, characters and words in files
```

```
[ec2-user@ip-172-31-31-15 ~]$ wc *txt
 2 4 18 example2.txt
 2 4 18 example.txt
 4 8 36 total
[ec2-user@ip-172-31-31-15 ~]$
```

zz) pwgen: generates random password

```
[$pwgen
laiR7uMa quoh9Eig dueP8fie jahFee2g Ohg0ieng baiG0iba Xaizoh1o du3iey5E
oYubieF4 oor0Quae aeNeK9El 0e2uqu1m EiCoh1Ph Ooy9toih Iu4ooTh9 sa4Vieko
Baithi4l aegh0aiH Deivem6y eeYiwi5k Ibuqu7em fae8baiL euW0Tohk Eich6aX6
thioC8ai ohg0Chah shae4aiP HohPeig7 Sieti8un cho5Aida iefaiT1d quoh0Isu
Cee7Uo3N Va1ohtoh eif4oR7a El8cayiu eez8Jeij eK2ooqua eiRo9eeC av1ieceS
0hX2ahr6 Yuogiph8 hang2Vee ooHaid2m Ana9ij8I eeW4ooje xanu2aeJ AN0chees
eib0Doh4 SooY2shi ohCe8Aiv aeT8Shei ooDeiqu5 HahDoo7Y oghiGh8a ein3heCh
roo1Eeri aeH5Equa ohY5dah0 Thu9xai9 Thae4eic eec1Ak8i waiX2yok aiC4foh7
MooseiG4 Fe5zee9a Ipoh9eoy Ohch1yoh Kachoh4o ui9cae7A to1Eewoh epe7Jieh
cufa1Bai looFoo2h Mei0thae eequaeY7 phah6No5 phaeJ4Ei Toz3caek ieChoh5a
um2too2Z ahb2Phah koPh40hr uHee7Ala Aiqu2qui sahcai0X aiboo5Fe Ou2VuiY0
aex4IBuD chahKoh9 ub3uka80 co3ohChu Ahhoog7i eeM7eeh2 bohw7aeS dooj9Cui
AijeNai0 xa3Aijoh aeV3Xeif oiT9cahd muo5UiCh Aetegie5 chioRee5 jah1Quua
Phoo6lee ee0reXei eigaiX0U eli10hxo eeT3shoo ooThae9e aeh7tohn Ei2weube
AFaidea3 Ao4gah2l ohGeejo6 Eik1phoh ohch1duB Sohvae0D aiw4Haif Eecubah0
eav1Ahc5 Fahm4fie xoChoo1t aF6or2ch aCi1tau4 juad1Ba9 Eep7aith Eech0AhP
aethee60 Ooghu2pu Il4duan6 ipaiC2ci Mizah4oh heeyaaD3 Xoy2Moh5 lahHai3y
eaSh0nei Oor0zaet Ra4ku5gi iix5Ek6t eiyeey0A chee5Xux gae5Iequ eif4Re0i
leiph1Ie ahmai7Ro Zooth2fi TaeShai0 ahs1Piej ahChohj0 ooyoX3to Phoh5the
mohThoh5 beeSh9ru kiezooV3 oe0ohg3Z bei4Fah9 Ush2ia4m bid7eB7h Iu3Pei1X
$
```

Question 3

- a) Script “generate-dataset.sh <filename> <num_records>” was generated with two command line arguments specifying the file name to output and the number of records, where each record is separated by new line character, and each has the following format: <integer> <integer> <ASCII_string>. The integers were stored in binary format (4 bytes) and the ASCII_string as a string.

Main script run-generate-dataset.sh runs generate-dataset.sh using the “time” command to show how long the benchmark took to complete. The benchmark runs for at least 10 seconds by the parent process sleeping. It uses a nohup command to ignore all hangup (SIGHUP) signals and to keep running program in the background after logging off.

As an example here is the output of running the process for 1000 records:

```
./run-generate-dataset.sh out1.txt 1000
Number of records generated: 1000

more nohup.out
0.92user 0.14system 0:01.09elapsed 98%CPU (0avgtext+0avgdata
3060maxresident)k
0inputs+216outputs (0major+294710minor)pagefaults 0swaps
1
```

- b) Script “sort-data.sh” was generated that takes input a file from part (a) above and sorts the file based on solely the first column data and not the entire line of data. The linux sort command is used to make sure the data in column 1 is treated as numbers and not text. It uses the “time” command to show how long the sort script took to complete.

As an example here is the output of running the process for 1000 records:

```
time ./sort-data.sh out1.txt
Output file is out1.txt.sortedoutput

real 0m0.045s
user 0m0.029s
```

```
sys 0m0.003s
```

- c) Script “script run-generate-dataset.sh” was used to generate 3 data files with different number of records: 1000, 100000, 10000000. Using the time command it measures time taken to generate these records.

```
./run-generate-dataset.sh out1.txt 1000
Number of records generated: 1000
```

```
more nohup.out
0.92user 0.14system 0:01.09elapsed 98%CPU (0avgtext+0avgdata
3060maxresident)k
0inputs+216outputs (0major+294710minor)pagefaults 0swaps
1
```

```
user+system time = 1.06 secs
```

```
./run-generate-dataset.sh out2.txt 100000
Number of records generated: 100000
more nohup.out
91.31user 13.85system 1:47.36elapsed 97%CPU (0avgtext+0avgdata
3068maxresident)k
0inputs+20920outputs (0major+28973695minor)pagefaults 0swaps
```

```
user+system time = 105.16 secs
```

```
./run-generate-dataset.sh out3.txt 10000000
Number of records generated: 10000000
more nohup.out
22866.49user 4565.22system 8:15:10elapsed 92%CPU (0avgtext+0avgdata
3096maxresident)k
2248inputs+2096976outputs (14major+2927733408minor)pagefaults 0swaps
```

```
user+system time = 27431.71 secs
```

Script “sort-data.sh” was used to sort the data files from part a) and measure the time.

```
time ./sort-data.sh out1.txt
Output file is out1.txt.sortedoutput

real 0m0.045s
user 0m0.029s
sys 0m0.003s

user+system time = 0.032 secs
```

```
time ./sort-data.sh out2.txt
Output file is out2.txt.sortedoutput
```

```
real 0m9.400s
user 0m7.166s
sys 0m0.016s

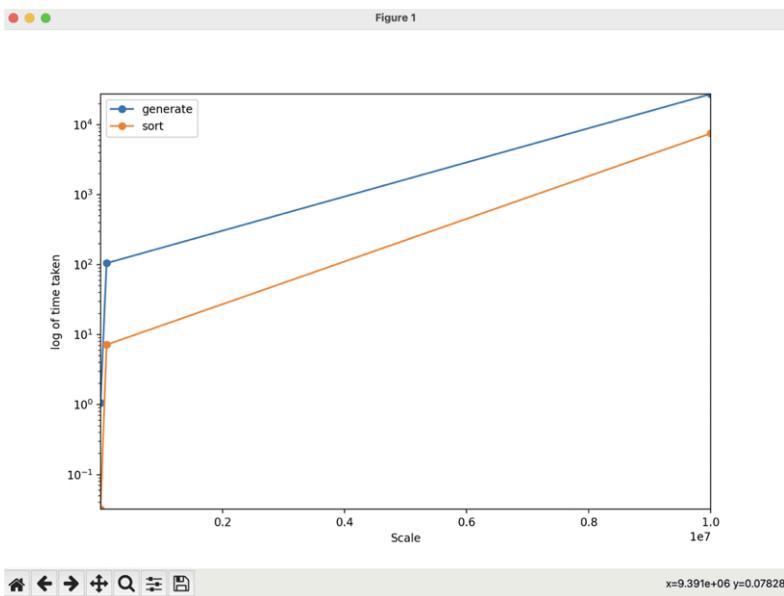
user+system time = 7.182 secs
```

```
time ./sort-data.sh out3.txt
Output file is out3.txt.sortedoutput
```

```
real 135m9.231s
user 124m58.416s
sys 0m12.636s

user+system time = 7511.052 secs
```

Python script `plot.py` was generated that uses `matplotlib` script to generate a graph for the time taken to generate the data and the time taken to sort the data at the 3 different scales. I used a log plot. The graph automatically adjust to the number of entries, and the scale of the data by using the `plt.axis((xmin,xmax,ymin,ymax))` command.
The resulting plot is below:



4)

- 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 do the following: 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?

When the number of processors is increased on a VM, tasks that can be parallelized or concurrent tasks run faster. On the flip side, more host resources are taken up by the particular VM.

If the scenario is that the VM needs to run smaller or lighter jobs like creating and hosting a simple internet website, then you can set the number of processors to a minimum.

If the scenario is that the VM needs to run heavier jobs like big data analytics, then you can set the number of processors to a maximum to increase performance.

Setting it to the maximum number of processors on a VM can potentially be a bad idea if there are multiple VMs running on a host. One particular VM will hog the host resources and the other VMs will starve.

- 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.

- **None:** disables the exposure of any paravirtualization interface.
- **Legacy:** is selected for VMs that were created with older versions of VirtualBox, and it will choose a paravirtualization interface when starting the VM with VirtualBox 5.0 or later.
- **Minimal:** Indicates the presence of a virtualized environment and also reports the TSC and APIC frequencies to the guest operating system. This provider is required for running any Mac OS X guests.
- **Hyper-V:** Provides a Microsoft Hyper-V hypervisor interface that is recognized by Windows 7 and later operating systems. VirtualBox's implementation currently includes support for paravirtualized clocks, APIC

- frequency reporting, guest debugging, guest crash reporting, and relaxed timer checks. Recommended for Windows guests.
- **KVM:** Offers a Linux KVM hypervisor interface recognized by Linux kernels starting from version 2.6.25. VirtualBox's implementation currently supports paravirtualized clocks and SMP spinlocks. This provider is recommended for Linux guests as it is optimized for linux
-
- 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.
 - IDE: is an asynchronous parallel interface that connects a host microprocessor system and a standard IDE device. It has slower transfer speeds. It is an older standard. If the guest OS does not support newer standards you can use IDE.
 - SATA: Is the interface that is usually used for Hard disk drives. They are faster than IDEs. Use it in scenarios where you have medium weight application that does not need blazing speed.
 - NVMe: Has the highest transfer rates, designed to work with SSDs. So if you have high disk access applications like large databases, transactional and analytical applications use NVMe for higher performance.
-
- 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: Similar to how your home network operates with a wireless router, the virtual machine will be allocated to a distinct subnet. For instance, if your host computer is assigned 192.168.6.1 and the VM is assigned 192.168.6.3, the VM can access external networks like your host can. However, direct external access to your VM is restricted, ensuring its protection. Can be used if scenarios like VM is set up for web browsing.
 - Bridged Adapter: Your VM will reside within the identical network as your host. For instance, if your host's IP is 172.16.120.45, then your VM's IP will resemble 172.16.120.50. It will be accessible to all computers within your host network. If you want to setup a VM as a database server and be accessible by other services.

- Internal Network: Allows only VMs to communicate. Channel cannot be used by host or other computers. Used for dev or testing purposes.
 - Host-only Network: Each virtual machine will receive a single IP address, it's exclusively reachable from the host machine; other computers are unable to connect to it. VMs and hosts can communicate with each other in a secluded manner. Can be used for development and testing environments where there is no need for internet access.
-
- For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers.
 - USB 1.1: It's an external bus standard with the ability to transfer data at a rate of 12 Megabits per second and has the capacity to accommodate up to 127 peripheral devices. Suitable for mice and keyboards.
 - USB 2.0: It's also referred to as high-speed USB, boasting a maximum transfer rate of 480 Megabits per second. However, in practice, it typically achieves around 280 Mbps. Suitable for HDDs
 - USB 3.0: This technology, also dubbed SuperSpeed USB, debuted in November 2009 as a significant advancement over USB 2.0. Offering a data transfer rate of 5 Gigabits per second, it far surpasses the speeds of its predecessor. Suitable for SSDs.