

CS 351 Lab #0 Part 1 - Linux & Tools

Alexis Edwards

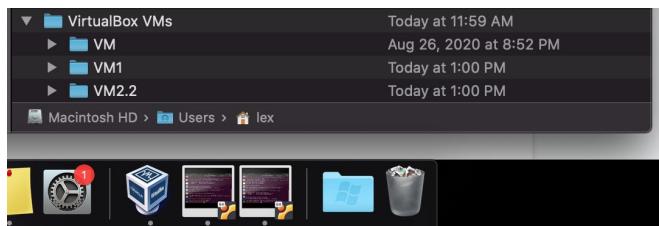
A20455965

1. Setup VM, Linux, and basic testing – must take screenshots at each step to receive points
 - a. Read Oracle VirtualBox White Paper
(<http://www.oracle.com/us/technologies/virtualization/oracle-vm-virtualbox-overview2981353.pdf>)



The overview states that the purpose of VirtualBox is to provide a platform for users to run multiple systems on their computers at one time. It is great for testing out software on different platforms/operating systems. It also bridges with cloud computing, as the VM may be uploaded and downloaded from the cloud in order to make or submit changes. VirtualBox is the only virtualization solution for all x86 platforms.

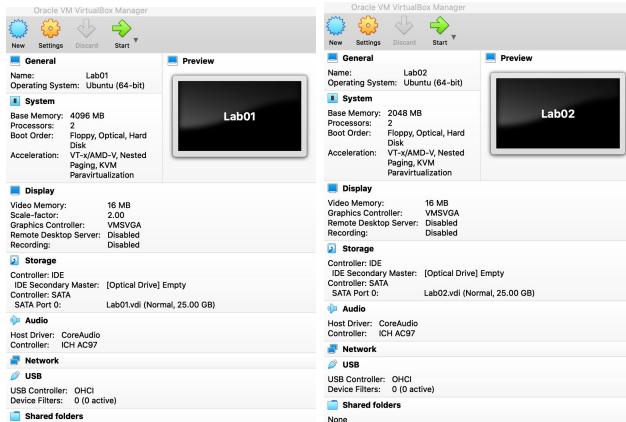
- b. Download Oracle VirtualBox 6.1.12 (<https://www.virtualbox.org/wiki/Downloads>)
 - c. Install VirtualBox (if VirtualBox is not supported on your platform, document it carefully why it does not work, and download/install Cygwin, <https://cygwin.com/install.html>)



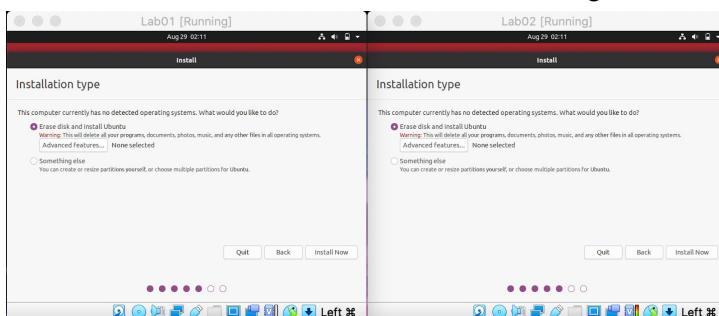
- d. Download Ubuntu Desktop 20.04.1 LTS Linux (<https://ubuntu.com/download/desktop>) ISO image



- e. Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with NAT support



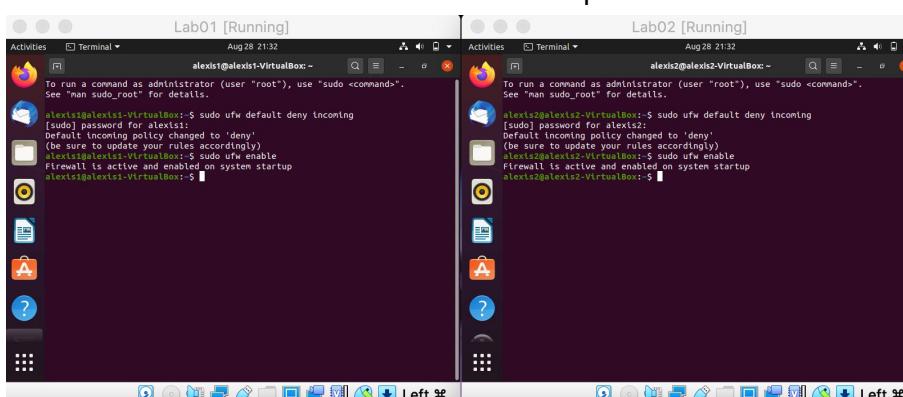
f. Install Linux from the ISO image



g. Create a user id and password



h. Turn on Firewall and block all ports



i. Enable SSH access to your new Linux installation; open SSH port in firewall

Lab01 [Running]

```
alex1@alex1-VirtualBox:~$ sudo apt update
Get:1 http://us.archive.ubuntu.com/ubuntu focal InRelease [111 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [109 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [111 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [98.3 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [204 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [184 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [194 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [52.1 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,976 B]
Fetched 786 kB in 1s (787 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
alex1@alex1-VirtualBox:~$
```

Lab02 [Running]

```
alex1@alex1-VirtualBox:~$ sudo apt update
Get:1 http://us.archive.ubuntu.com/ubuntu focal InRelease [111 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [109 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [98.3 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [204 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [184 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [194 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,976 B]
Fetched 602 kB in 1s (648 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
alex1@alex1-VirtualBox:~$
```

Lab01 [Running]

```
alex1@alex1-VirtualBox:~$ sudo apt update
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [184 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [24.3 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [52.1 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [24.3 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,976 B]
Fetched 786 kB in 1s (787 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  openssh-client openssh-sftp-server ssh-askpass
The Following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
0 to get 688 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Lab02 [Running]

```
alex1@alex1-VirtualBox:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [98.3 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [204 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [184 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,976 B]
Fetched 602 kB in 1s (648 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  openssh-client openssh-sftp-server ssh-askpass
The Following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
0 to get 688 kB of archives.
A Show Applications icon, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Lab01 [Running]

```
alex1@alex1-VirtualBox:~$ sudo systemctl status ssh
systemd/system/ssh.service
  rescue-ssh.target is a disabled or a static unit, not starting it.
  Attaching to unit rescue-ssh.target (pid 1).
  rescue-ssh.target (pid 1) exited, reason: Exit code 0
  rescue-ssh.target (pid 1) started.
  rescue-ssh.target (pid 1) active (running) since Fri 2020-08-28 21:36:31 CDT; 1min 17s ago
    Docs: man:sshd(8)
      Main PID: 3454 (sshd)
        Tasks: 1 (llimit: 2319)
       Memory: 1.2M
          CPU: 0.000 CPU(s) slice:/system.slice/sshd.service
            └─ 3454 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
Aug 28 21:36:31 alex1-VirtualBox systemd[1]: Starting OpenBSD Secure Shell...
Aug 28 21:36:31 alex1-VirtualBox sshd[3454]: Server listening on 0.0.0.0 port 22.
Aug 28 21:36:31 alex1-VirtualBox sshd[3454]: Server listening on :: port 22.
Aug 28 21:36:31 alex1-VirtualBox systemd[1]: Started OpenBSD Secure Shell service.
alex1@alex1-VirtualBox:~$ sudo ufw allow ssh
Rule added
Rule added (v)
alex1@alex1-VirtualBox:~$ 
```

Lab02 [Running]

```
alex1@alex1-VirtualBox:~$ sudo systemctl status ssh
systemd/system/ssh.service
  rescue-ssh.target is a disabled or a static unit, not starting it.
  Attaching to unit rescue-ssh.target (pid 1).
  rescue-ssh.target (pid 1) exited, reason: Exit code 0
  rescue-ssh.target (pid 1) started.
  rescue-ssh.target (pid 1) active (running) since Fri 2020-08-28 21:36:33 CDT; 1min 25s ago
    Docs: man:sshd(8)
      Main PID: 3456 (sshd)
        Tasks: 1 (llimit: 2319)
       Memory: 1.2M
          CPU: 0.000 CPU(s) slice:/system.slice/sshd.service
            └─ 3456 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
Aug 28 21:36:33 alex1-VirtualBox systemd[1]: Starting OpenBSD Secure Shell...
Aug 28 21:36:33 alex1-VirtualBox sshd[3456]: Server listening on 0.0.0.0 port 22.
Aug 28 21:36:33 alex1-VirtualBox sshd[3456]: Server listening on :: port 22.
Aug 28 21:36:33 alex1-VirtualBox systemd[1]: Started OpenBSD Secure Shell service.
alex1@alex1-VirtualBox:~$ sudo ufw allow ssh
Rule added
Rule added (v)
alex1@alex1-VirtualBox:~$ 
```

- j. Repeat steps 5 through 9, and create another VM with the same specifications as the first one (shown simultaneously with screenshots above)
- k. Create private/public keys and install them properly in both of your new VMs

Lab01 [Running]

```
alex1@alex1-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alex1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your public key has been saved in /home/alex1/.ssh/id_rsa.
The key fingerprint is:
SHA256:0pkywfpFL4NF9GonFvF+1BX22c alex1@alex1-VirtualBox
The key's randomart image is:
...[RSA 3072]... 
```

Lab02 [Running]

```
alex1@alex1-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alex1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your public key has been saved in /home/alex1/.ssh/id_rsa.
The key fingerprint is:
SHA256:0pkywfpFL4NF9GonFvF+1BX22c alex1@alex1-VirtualBox
The key's randomart image is:
...[RSA 3072]... 
```

```

Lab01 [Running]
Activities Terminal Aug 28 21:53
alex1s1@alex1s1-VirtualBox:~$ ssh-copy-id -l -i ./ssh/id_rsa.pub alex1s2@192.168.56.104
INFO: Source of key(s) to be installed: '/home/alex1s1/.ssh/id_rsa.pub'
INFO: attempting to log in with the new key(s), to filter
INFO: out any that are already installed
INFO: 1 key(s) remain to be installed -- if you are prompted now it's to install the new keys
alex1s2@192.168.56.104's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'alex1s2@192.168.56.104'" and check to make sure that only the key(s) you wanted were added.
alex1s2@alex1s2-VirtualBox:~$ 

Lab02 [Running]
Activities Terminal Aug 28 21:53
alex1s2@alex1s2-VirtualBox:~$ ssh-copy-id -l -i ./ssh/id_rsa.pub alex1s1@192.168.56.103
INFO: Source of key(s) to be installed: '/home/alex1s2/.ssh/id_rsa.pub'
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ECDSA key fingerprint is SHA256:S9A8QpcFFnG0D1mNxewBZc+fHfCvCxUF77392ba.
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
INFO: attempting to log in with the new key(s), to filter
INFO: out any that are already installed
INFO: 1 key(s) remain to be installed -- if you are prompted now it's to install the new keys
alex1s1@192.168.56.104's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'alex1s1@192.168.56.103'" and check to make sure that only the key(s) you wanted were added.
alex1s1@alex1s1-VirtualBox:~$ 

```

- I. Test that you can connect remotely to your VMs with your keys, from one VM to the other VM

```

alex1s1@alex1s1-VirtualBox:~$ ssh alex1s2@192.168.56.104
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

94 updates can be installed immediately.
37 of these updates are security updates.
To see these additional updates run: apt list --upgradable

>Your Hardware Enablement Stack (HWE) is supported until April 2025.
alex1s1@alex1s1-VirtualBox:~$ 

alex1s2@alex1s2-VirtualBox:~$ ssh alex1s1@192.168.56.103
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

94 updates can be installed immediately.
37 of these updates are security updates.
To see these additional updates run: apt list --upgradable

>Your Hardware Enablement Stack (HWE) is supported until April 2025.
alex1s2@alex1s2-VirtualBox:~$ 

```

2. Show an example of using the following commands (hint: you can use man to find more information about each one); take screen shots of your commands; make sure to clear the screen between each command; explain in your own words what these commands do:

- a. ssh: Provides connection between two hosts.

```

Lab01 [Running]
Activities Terminal Aug 28 22:03
alex1s1@alex1s1-VirtualBox:~$ ssh alex1s2@192.168.56.104
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

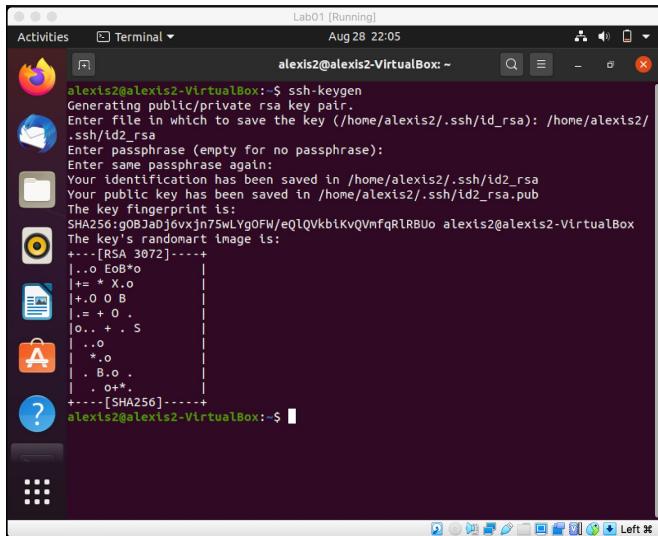
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

94 updates can be installed immediately.
37 of these updates are security updates.
To see these additional updates run: apt list --upgradable

>Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Aug 28 21:53:44 2020 from 192.168.56.103
alex1s1@alex1s1-VirtualBox:~$ 

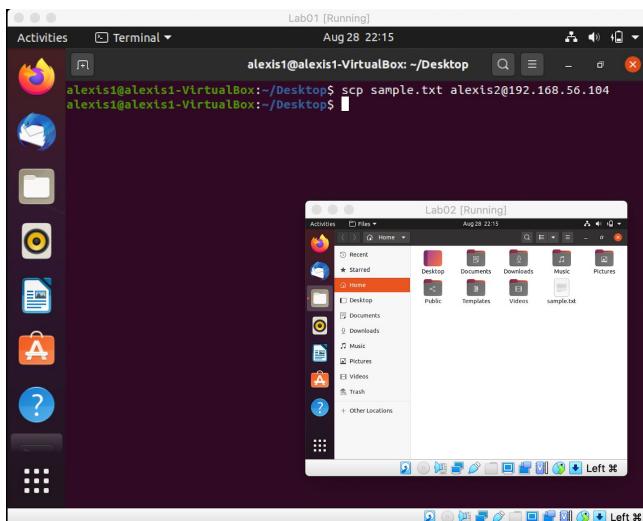
```

- b. ssh-keygen: Creates new private/public keys for authentication.



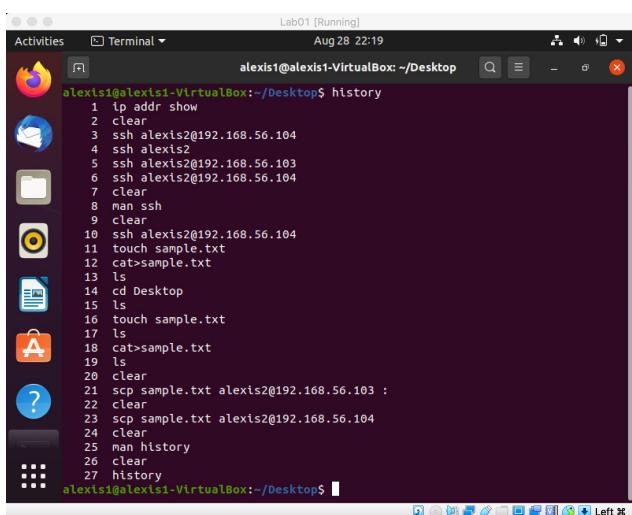
```
alexis2@alexis2-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alexis2/.ssh/id_rsa): /home/alexis2/.ssh/id2_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alexis2/.ssh/id2_rsa
Your public key has been saved in /home/alexis2/.ssh/id2_rsa.pub
The key's randomart image is:
+---[RSA 3072]----+
|...o EoB*o|
|+= * X.o|
|+.0 O B|
|= + 0 .|
|o.. + . S|
|..o|
| * o|
| . B.o|
| . 0*.|
+---[SHA256]-----+
alexis2@alexis2-VirtualBox:~$
```

c. scp: Copies files over a SSH connection. Below example is copying sample.txt.



```
alexis1@alexis1-VirtualBox:~/Desktop$ scp sample.txt alexis2@192.168.56.104
alexis1@alexis1-VirtualBox:~/Desktop$
```

d. history: Shows a list of the commands entered since starting the terminal session.



```
alexis1@alexis1-VirtualBox:~/Desktop$ history
1 ip addr show
2 clear
3 ssh alexis2@192.168.56.104
4 ssh alexis2
5 ssh alexis2@192.168.56.103
6 ssh alexis2@192.168.56.104
7 clear
8 man ssh
9 clear
10 ssh alexis2@192.168.56.104
11 touch sample.txt
12 cat>sample.txt
13 ls
14 cd Desktop
15 ls
16 touch sample.txt
17 ls
18 cat>sample.txt
19 ls
20 clear
21 scp sample.txt alexis2@192.168.56.103 :
22 clear
23 scp sample.txt alexis2@192.168.56.104
24 clear
25 man history
26 clear
27 history
alexis1@alexis1-VirtualBox:~/Desktop$
```

- e. sudo: Allows you to run programs as another user. Typically the root user. Below command is showing the version number.

```
alexls1@alexls1-VirtualBox: ~/Desktop$ sudo -V
Sudo version 1.8.31
Sudoers policy plugin version 1.8.31
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.31
alexls1@alexls1-VirtualBox: ~/Desktop$
```

- f. ip: Used for configuring network interfaces. Below example shows the IP address.

```
alexls1@alexls1-VirtualBox: ~/Desktop$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.0.0.1 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 brd :: scope host ::1
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3e:d5:e9 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 84425sec preferred_lft 84425sec
    inet6 fe80::ecce:adoc:es55:65f5/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a6:2d:e8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
        valid_lft 425sec preferred_lft 425sec
    inet6 fe80::dcad:4622:a68:dc10/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute
        valid_lft forever preferred_lft forever
alexls1@alexls1-VirtualBox: ~/Desktop$
```

- g. dd: Copies, converts, and formats a file based on the arguments provided. Below example converted lowercase text from file1 to uppercase text in file2

```
alexls1@alexls1-VirtualBox: ~/Desktop$ dd if=file1.txt of=file2.txt conv=ucase
0+1 records in
0+1 records out
0 bytes copied, 0.00143266 s, 4.2 kB/s
alexls1@alexls1-VirtualBox: ~/Desktop$ cat file2
catfile2: command not found
alexls1@alexls1-VirtualBox: ~/Desktop$ cat file2
cat: file2: No such file or directory
alexls1@alexls1-VirtualBox: ~/Desktop$ cat file2.txt
HELLO
alexls1@alexls1-VirtualBox: ~/Desktop$
```

- h. fdisk: Used for creating and manipulating partition tables (a data structure that provides info on an operating system's hard drive). Below example shows all disk partitions.

```

alexis1@alexis1-VirtualBox:~$ sudo fdisk -l
[sudo] password for alexis1:
Disk /dev/loop0: 54.98 MB, 57626624 bytes, 112552 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: 255.58 MB, 267980800 bytes, 523400 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: 62.9 MB, 65105920 bytes, 127160 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: 49.8 MB, 52203520 bytes, 101960 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/loop4: 29.9 MB, 31334000 bytes, 61200 sectors
U Show Applications f 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

```

- i. apt: A command line tool for adding, removing, or updating packages.

```

alexis1@alexis1-VirtualBox:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [204 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [88.4 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [184 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,976 B]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [52.3 kB]
Fetched 1,037 kB in 1s (818 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
alexis1@alexis1-VirtualBox:~$ 

```

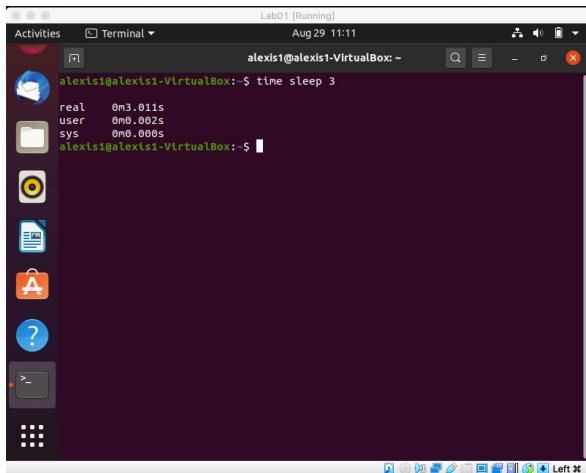
- j. vi: Allows you to edit files. VI stands for “visual editor.”

```

alexis1@alexis1-VirtualBox:~$ 
alexis1@alexis1-VirtualBox:~$ touch file1.txt
alexis1@alexis1-VirtualBox:~$ 

```

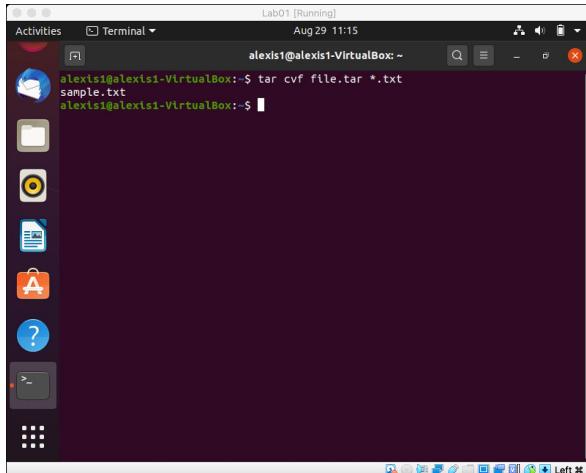
- k. time: Returns the amount of time a given command takes to load. Below example creates a program that essentially stalls for 3 seconds.



A screenshot of a Linux desktop environment showing a terminal window titled "Lab01 [Running]". The terminal window has a dark background and contains the following text:

```
alex1@alex1-VirtualBox:~$ time sleep 3
real 0m3.011s
user 0m0.002s
sys 0m0.000s
alex1@alex1-VirtualBox:~$
```

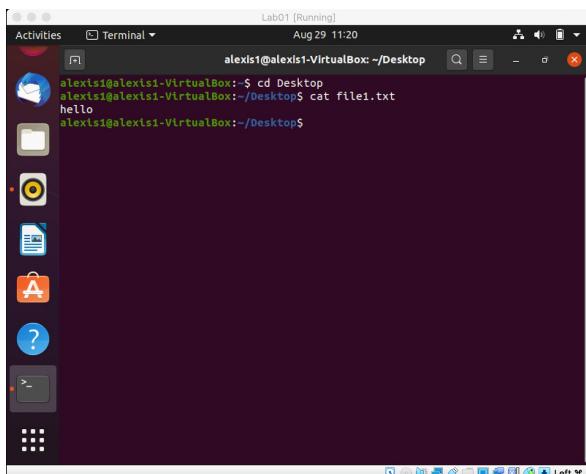
I. tar: Creates Archive and extracts the files. TAR stands for “tape archive.”



A screenshot of a Linux desktop environment showing a terminal window titled "Lab01 [Running]". The terminal window has a dark background and contains the following text:

```
alex1@alex1-VirtualBox:~$ tar cvf file.tar *.txt
sample.txt
alex1@alex1-VirtualBox:~$
```

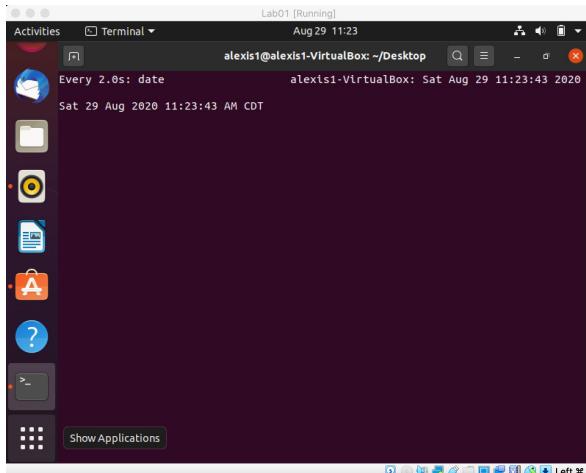
m. cat: Reads data from files and outputs their contents. Also allows concatenation.



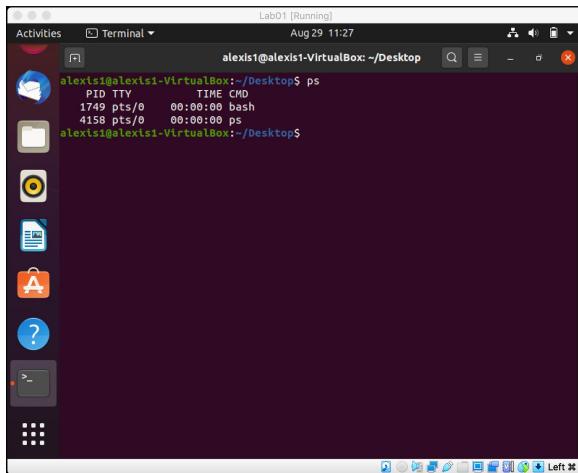
A screenshot of a Linux desktop environment showing a terminal window titled "Lab01 [Running]". The terminal window has a dark background and contains the following text:

```
alex1@alex1-VirtualBox:~$ cd Desktop
alex1@alex1-VirtualBox:/Desktop$ cat file1.txt
Hello
alex1@alex1-VirtualBox:/Desktop$
```

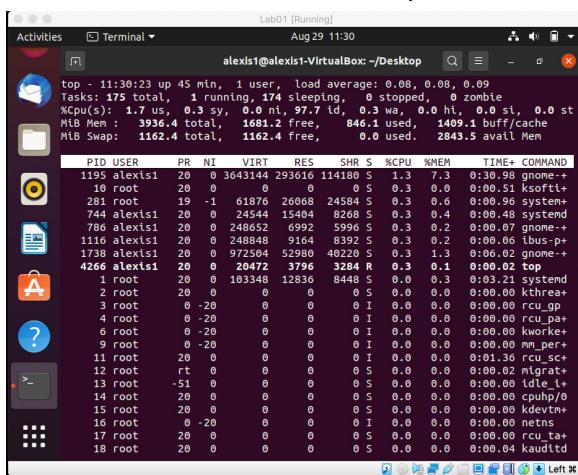
n. watch: Runs commands at regular intervals and outputs the results so you can see the changes. Below example outputs the date and time every 2 seconds (the default).



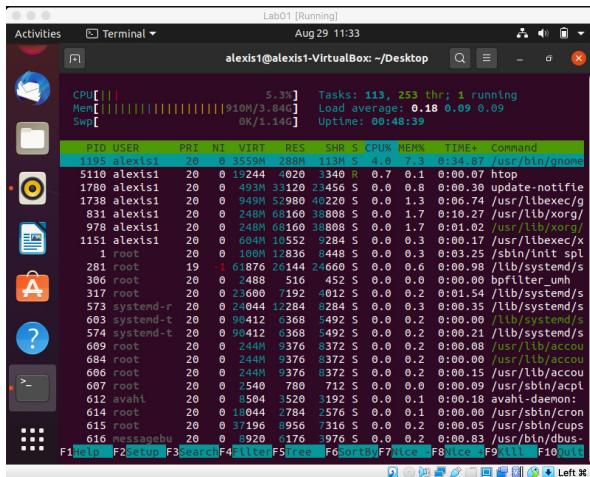
o. ps: Provides information on the running processes.



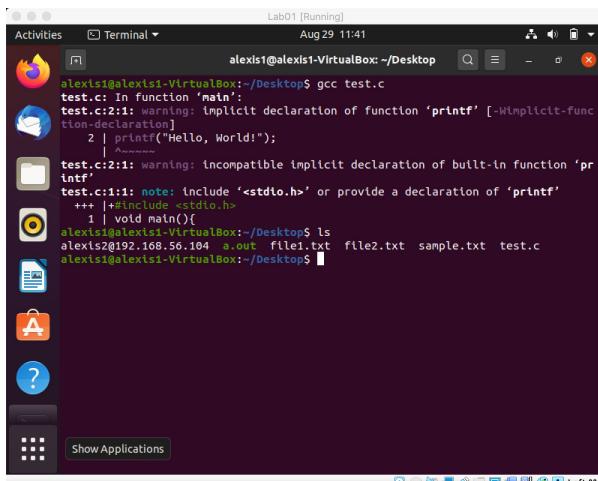
p. top: Outputs the Linux processes. This is different from ps because it constantly refreshes, while ps is static.



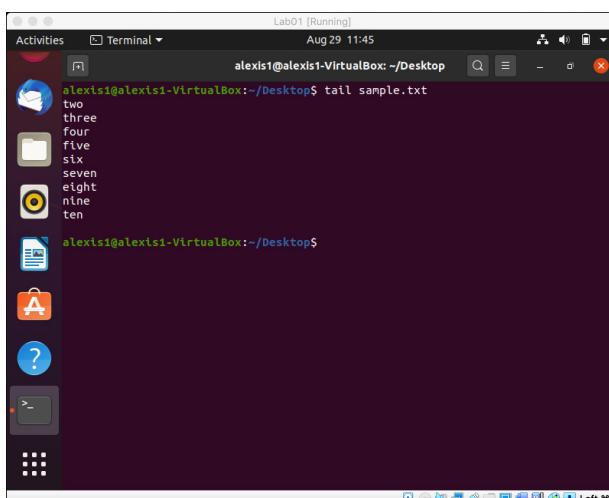
q. htop: Similar to top, but upgraded. Allows you to interact with and monitor the Linux processes in real time.



r. gcc: Stands for GNU Compiler Connections. Compiles C and C++ languages.



s. tail: Returns the last 10 lines of the given file.



t. grep: Searches lines of text for the given String.

```

alexisi@alexisi1-VirtualBox:~/Desktop$ grep 'one' sample.txt
one
alexisi@alexisi1-VirtualBox:~/Desktop$ 

```

- u. kill: Manually terminate a process with its PID. Below example just shows processes available to terminate. I don't know what they do so I didn't want to try it on any of them.

```

alexisi@alexisi1-VirtualBox:~/Desktop$ kill -l
 1) SIGHUP  2) SIGINT  3) SIGQUIT   4) SIGILL    5) SIGTRAP
 6) SIGABRT  7) SIGBUS   8) SIGPPE    9) SIGKILL   10) SIGUSR1
11) SIGSEGV  12) SIGUSR2  13) SIGPIPE   14) SIGALRM   15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD  18) SIGCONT   19) SIGSTOP   20) SIGTSTP
21) SIGTTIN  22) SIGTTOU  23) SIGURG    24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF  28) SIGWINCH  29) SIGTO    30) SIGPWR
31) SIGSYS  34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
alexisi@alexisi1-VirtualBox:~/Desktop$ 

```

- v. killall: Manually terminate a process based on its name.

```

alexisi@alexisi1-VirtualBox:~/Desktop$ killall -l
HUP INT QUIT ILL TRAP ABRT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STKFLT
CHLD CONT STOP TSTP TTIN TTOUT URG XCPU XFSZ VTALRM PROF WINCH POLL PWR SYS
alexisi@alexisi1-VirtualBox:~/Desktop$ 

```

- w. du: Stands for disk usage. Outputs estimated file space usage.

```

alexis1@alexis1-VirtualBox: ~
.
4 ./Pictures
4 ./local/share/sounds
4 ./local/share/ibus-table
60 ./local/share/xorg
8 ./local/share/evolution/tasks/system
4 ./local/share/evolution/tasks/trash
16 ./local/share/evolution/tasks
8 ./local/share/evolution/calendar/system
4 ./local/share/evolution/calendar/crash
16 ./local/share/evolution/calendar
4 ./local/share/evolution/mail/trash
8 ./local/share/evolution/mail
4 ./local/share/evolution/memos/trash
8 ./local/share/evolution/memos
4 ./local/share/evolution/addressbook/system/photos
92 ./local/share/evolution/addressbook/system
4 ./local/share/evolution/addressbook/trash
100 ./local/share/evolution/addressbook
152 ./local/share/evolution
4 ./local/share/webkitgtk/deviceldhashsalts/1
8 ./local/share/webkitgtk/deviceldhashsalts
4 ./local/share/webkitgtk/databases/indexedb/v1
8 ./local/share/webkitgtk/databases/indexedb
12 ./local/share/webkitgtk/localstorage
4 ./local/share/webkitgtk
40 ./local/share/webkitgtk
4 ./local/share/flatpak/db
8 ./local/share/flatpak

```

x. df: Stands for disk free. Outputs information about available disk space.

Fs	1K-blocks	Used	Available	Use%	Mounted on
udev	1987280	0	1987280	0%	/dev
tmpfs	403088	1352	401736	1%	/run
/dev/sda5	25152772	6712824	17139212	29%	/
tmpfs	2015440	0	2015440	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	2015440	0	2015440	0%	/sys/fs/cgroup
/dev/loop0	56320	56320	0	100%	/snap/core18/1880
/dev/loop1	261760	261760	0	100%	/snap/gnome-3-34-1804/36
/dev/loop2	63616	63616	0	100%	/snap/gtk-common-themes/1506
/dev/loop3	51072	51072	0	100%	/snap/snap-store/467
/dev/loop4	30720	30720	0	100%	/snap/snapd/8542
/dev/sda1	523248	4	523244	1%	/boot/efi
tmpfs	403088	36	403052	1%	/run/user/1000
/dev/loop5	30720	30720	0	100%	/snap/snapd/8790
/dev/loop6	56704	56704	0	100%	/snap/core18/1885

y. screen: Allows you to run multiple shell sessions from a single SSH.

```

GNU Screen version 4.08.00 (GNU) 05-Feb-20
Copyright (c) 2018-2020 Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2010-2014 Juergen Weigert, Sadruл Habib Chowdhury
Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan,
Sadruл Habib Chowdhury
Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 3, or (at your option) any later version.

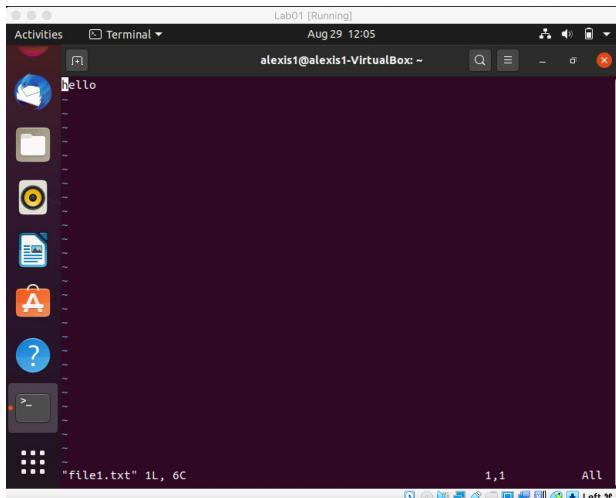
This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
details.

You should have received a copy of the GNU General Public License along with
this program (see the file COPYING); if not, see
https://www.gnu.org/licenses/, or contact Free Software Foundation, Inc., 51
Franklin Street, Fifth Floor, Boston, MA 02111-1301 USA.

Send bugreports, fixes, enhancements, t-shirts, money, beer & pizza to
Show Applications [Press Space for next page; Return to end.]

```

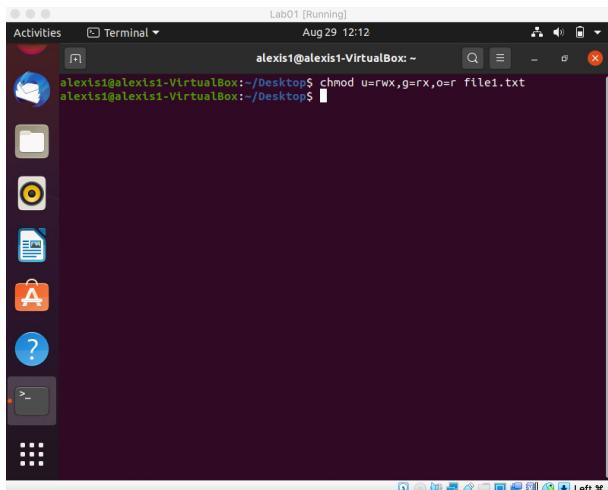
z. vim: A command line text editor.



```
alexis1@alexis1-VirtualBox: ~
```

"file1.txt" 1L, 6C

aa. chmod: Changes the permissions of a given file. Determines who can access files and directories. Below example states that I can read, write, and execute, the group may read and execute, and others may only read.



```
alexis1@alexis1-VirtualBox: ~/Desktop$ chmod u=rwx,g=rx,o=r file1.txt
```

bb. Allows you to change the ownership of a given file. Below example would change ownership to alexis3 if I had that user on my computer (I only have my 1 user - me).

```
alex1@alex1-VirtualBox:~/Desktop$ chown alexis3 file2.txt
chown: invalid user: 'alexis3'
alex1@alex1-VirtualBox:~/Desktop$
```

cc. useradd: Adds a new user account to the system.

```
alex1@alex1-VirtualBox:~/Desktop$ sudo useradd alexis3
alex1@alex1-VirtualBox:~/Desktop$
```

dd. man: Provides the user manual of a given command

```
VIM(1) General Commands Manual VIM(1)

NAME
    vim - Vi IMproved, a programmer's text editor

SYNOPSIS
    vim [options] [file ...]
    vim [options] -
    vim [options] +t tag
    vim [options] -q [errorfile]

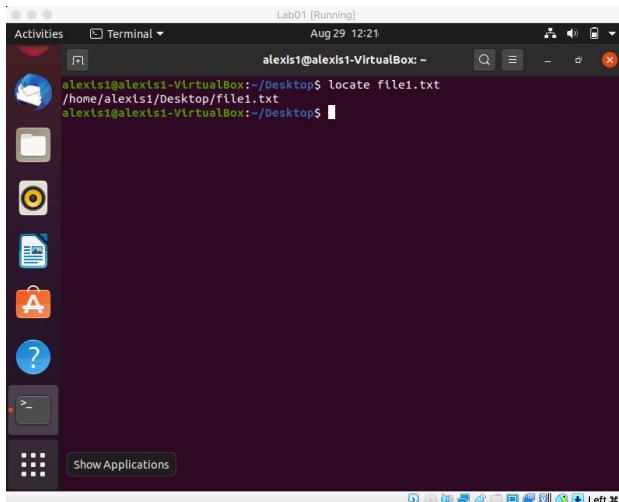
    ex
    view
    gvim gview evim eview
    rgvim rview rgview

DESCRIPTION
    Vim is a text editor that is upwards compatible to Vi. It can be used
    to edit all kinds of plain text. It is especially useful for editing
    programs.

    There are a lot of enhancements above Vi: multi level undo, multi windows
    and buffers, syntax highlighting, command line editing, filename
    completion, on-line help, visual selection, etc.. See ":help
    vi_diff.txt" for a summary of the differences between Vim and Vi.

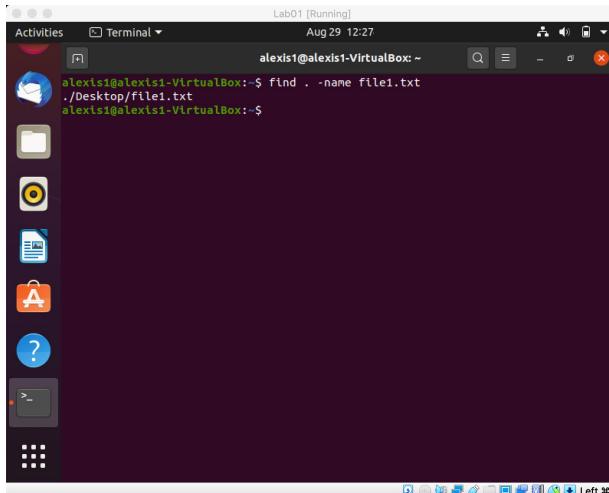
    While running Vim a lot of help can be obtained from the on-line help
    Show Applications with the ":help" command. See the ON-LINE HELP section below.
    Manual page Vim(1) line 1 (press h for help or q to quit)
```

ee. locate: Provides the location of a given file within your system.



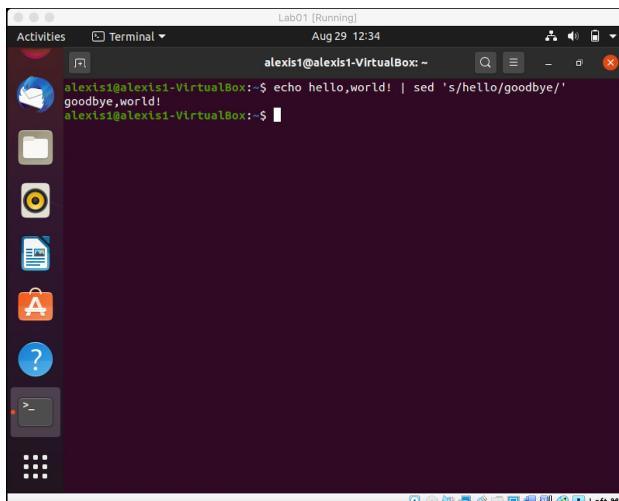
```
alex1@alexis1-VirtualBox:~/Desktop$ locate file1.txt
/home/alexis1/Desktop/file1.txt
alex1@alexis1-VirtualBox:~/Desktop$
```

- ff. **find**: Similar to locate. Can find files or directories based on patterns, such as extensions or permissions. Below is an example of finding a file with a specific name.



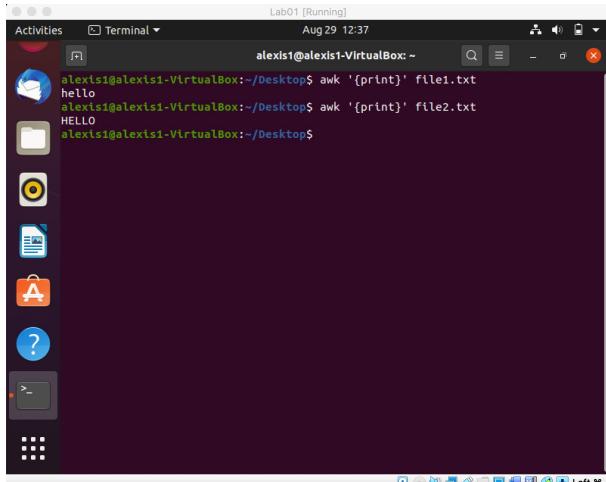
```
alex1@alexis1-VirtualBox:~$ find . -name file1.txt
./Desktop/file1.txt
alex1@alexis1-VirtualBox:~$
```

- gg. **sed**: Stands for stream editor. Filters and transforms text based on given rules (substituting, adding, deleting, etc.).



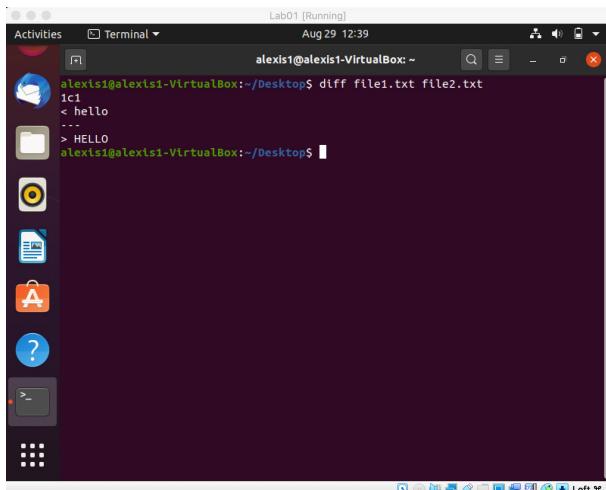
```
alex1@alexis1-VirtualBox:~$ echo hello,world! | sed 's/hello/goodbye/'
goodbye,worl!
alex1@alexis1-VirtualBox:~$
```

hh. awk: Used for pattern scanning and processing. Searches files for a given pattern and performs the provided commands. Below example simply prints the contents of the files.



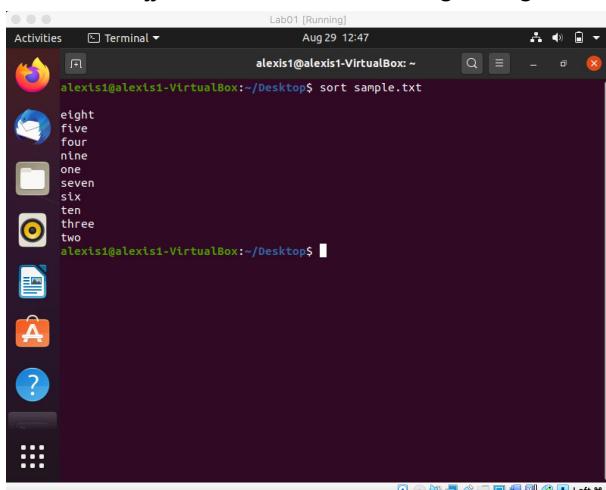
```
alex1@alex1-VirtualBox:~/Desktop$ awk '{print}' file1.txt
hello
alex1@alex1-VirtualBox:~/Desktop$ awk '{print}' file2.txt
HELLO
alex1@alex1-VirtualBox:~/Desktop$
```

ii. diff: Compares each line of given files and outputs the difference.



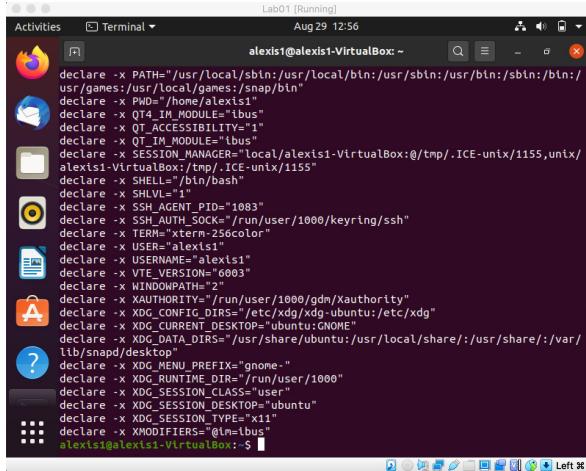
```
alex1@alex1-VirtualBox:~/Desktop$ diff file1.txt file2.txt
ici
< hello
...
> HELLO
alex1@alex1-VirtualBox:~/Desktop$
```

jj. sort: Sorts or arranges a given file in a particular order.



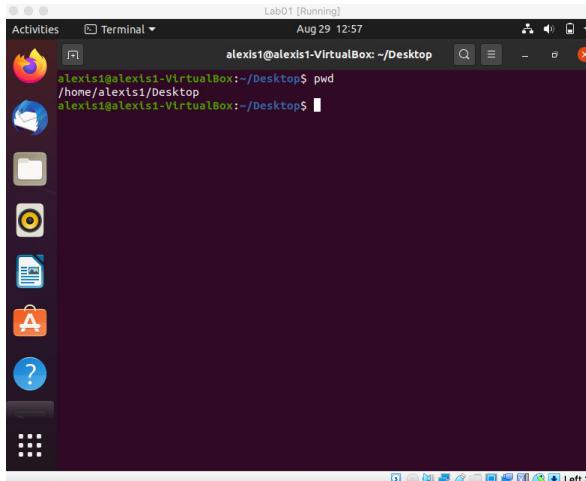
```
alex1@alex1-VirtualBox:~/Desktop$ sort sample.txt
eight
five
four
nine
one
seven
six
ten
three
two
alex1@alex1-VirtualBox:~/Desktop$
```

kk. **export**: Exports or passes on an environment variable to child processes. Below example shows all exported variables.



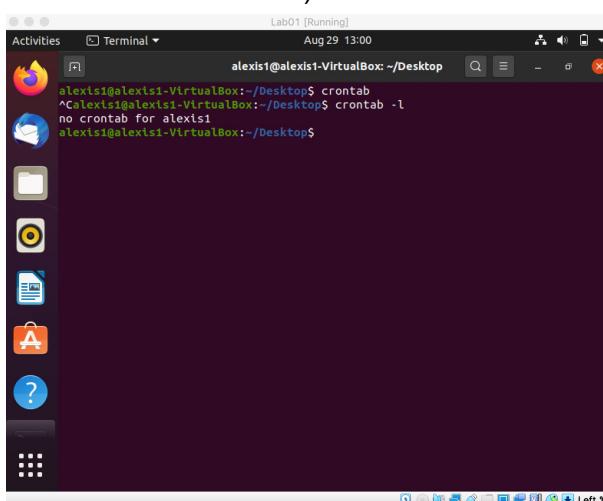
```
declare -x PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
declare -x PWMS="/home/alexis1"
declare -x QT4_IM_MODULE="ibus"
declare -x QT_ACCESSIBILITY="1"
declare -x QT_IM_MODULE="ibus"
declare -x SESSION_MANAGER="local/alexis1-VirtualBox:@tmp/.ICE-unix/1155,unix/alexis1-VirtualBox:/tmp/.ICE-unix/1155"
declare -x SHHELL="/bin/bash"
declare -x SHLVL="1"
declare -x XAUTHORITY=":1083"
declare -x SSH_AUTH_SOCK="/run/user/1000/keyring/ssh"
declare -x TERM="stern-256color"
declare -x USER="alexis1"
declare -x USERNAME="alexis1"
declare -x VTE_VERSION="6003"
declare -x WINDOWPATH="2"
declare -x XAUTORITY="/run/user/1000/gdm/Xauthority"
declare -x XDG_CONFIG_DIRS="/etc/xdg:xdg-ubuntu:/etc/xdg"
declare -x XDG_CURRENT_DESKTOP="ubuntu:GNOME"
declare -x XDG_DATA_DIRS="/usr/share/ubuntu:/usr/local/share:/usr/share/:/var/lib/snapd/desktop"
declare -x XDG_MENU_PREFIX="gnome-"
declare -x XDG_RUNTIME_DIR="/run/user/1000"
declare -x XDG_SESSION_CLASS="user"
declare -x XDG_SESSION_DESKTOP="ubuntu"
declare -x XDG_SESSION_TYPE="x11"
declare -x XMODIFIERS="@im=ibus"
alexis1@alexis1-VirtualBox:~$
```

ll. **pwd**: Stands for print working directory. Outputs the path of the working directory



```
alexis1@alexis1-VirtualBox:~/Desktop$ pwd
/home/alexis1/Desktop
alexis1@alexis1-VirtualBox:~/Desktop$
```

mm. **crontab**: Allows you to edit the cron table (list of tasks set to run at regular intervals).



```
alexis1@alexis1-VirtualBox:~/Desktop$ crontab
^Calexis1@alexis1-VirtualBox:~/Desktop$ crontab -l
no crontab for alexis1
alexis1@alexis1-VirtualBox:~/Desktop$
```

nn. **mount**: Attaches a filesystem to the big tree structure starting at "./". Below example just lists all the file systems mounted yet.

```
alex1@alex1-VirtualBox:~$ ls -l /sys/fs/fuse/connections
total 0
alex1@alex1-VirtualBox:~$
```

The terminal window shows the output of the command 'ls -l /sys/fs/fuse/connections'. It lists several entries, each representing a file or directory under the /sys/fs/fuse/connections path. The entries include 'fusectl', 'configfs', 'squashfs', 'tmpfs', and various 'gvfsd-fuse' and 'fuse' entries.

oo. passwd: Changes the user's password.

```
alex1@alex1-VirtualBox:~$ passwd
Changing password for alex1.
Current password:
New password:
Retype new password:
passwd: password updated successfully
alex1@alex1-VirtualBox:~$
```

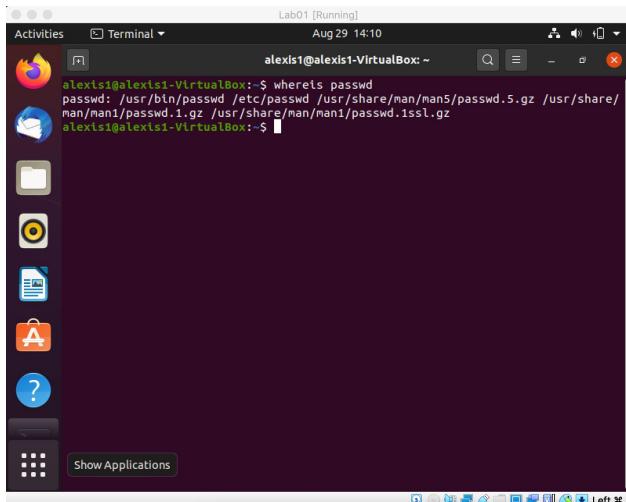
The terminal window shows the output of the 'passwd' command. It prompts for the current password, then asks for a new password, re-prompts for it, and finally confirms that the password has been updated successfully.

pp. uname: Outputs information about the operating system and hardware.

```
alex1@alex1-VirtualBox:~$ uname
Linux
alex1@alex1-VirtualBox:~$
```

The terminal window shows the output of the 'uname' command. It prints 'Linux' as the system name.

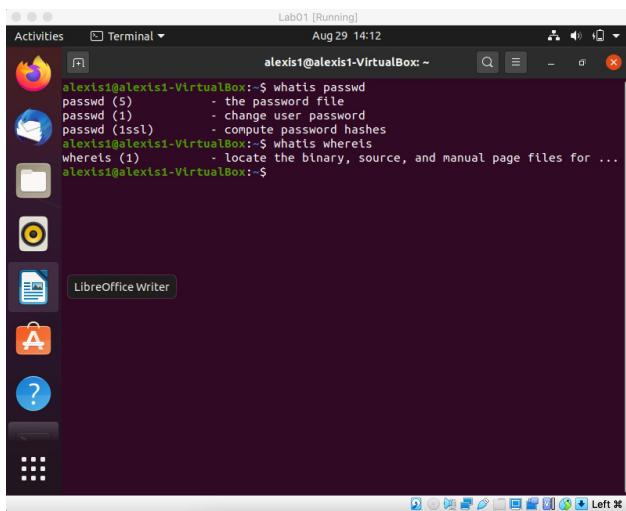
qq. whereis: Similar to the find command, but is more accurate and consumes less time. Locates a program within the directories. Below example shows location of passwd command.



A screenshot of an Ubuntu desktop environment. A terminal window titled "Lab01 [Running]" is open, showing the command "wherels passwd" and its output. The terminal window is located in the top panel of the desktop environment.

```
alex1@alex1-VirtualBox:~$ wherels passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man5/passwd.5.gz /usr/share/
man/man1/passwd.1.gz /usr/share/man/man1/passwd.1ssl.gz
alex1@alex1-VirtualBox:~$
```

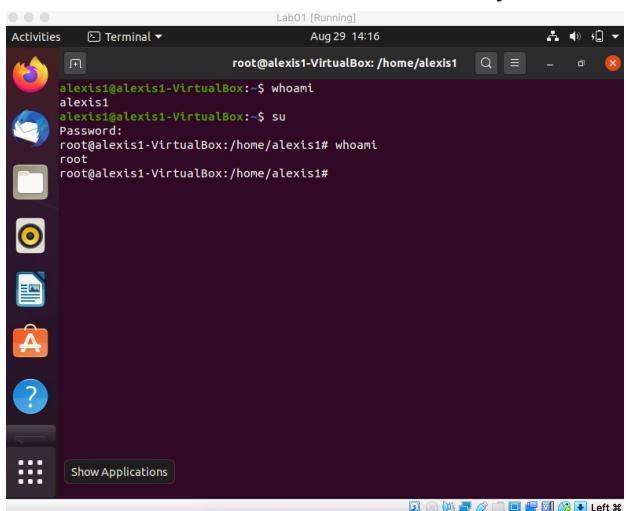
rr. whatis: Obtains a short description of the given command.



A screenshot of an Ubuntu desktop environment. A terminal window titled "Lab01 [Running]" is open, showing the command "whatis passwd" and its output. The terminal window is located in the top panel of the desktop environment.

```
alex1@alex1-VirtualBox:~$ whatis passwd
passwd (5)           - the password file
passwd (1)           - change user password
passwd (ssl)         - compute password hashes
alex1@alex1-VirtualBox:~$ whatis wherels
wherels (1)          - Locate the binary, source, and manual page files for ...
alex1@alex1-VirtualBox:~$
```

ss. su: Stands for switch user. Allows you to run commands as a substitute user, which is the root user by default.



A screenshot of an Ubuntu desktop environment. A terminal window titled "Lab01 [Running]" is open, showing the command "su" being used to switch to the root user. The terminal window is located in the top panel of the desktop environment.

```
alex1@alex1-VirtualBox:~$ whoami
alex1
alex1@alex1-VirtualBox:~$ su
Password:
root@alex1-VirtualBox:/home/alex1# whoami
root
root@alex1-VirtualBox:/home/alex1#
```

tt. ping: Checks the internet connectivity between the host and the server.

```
alex1@alex1-VirtualBox:~$ ping
ping: usage error: Destination address required
alex1@alex1-VirtualBox:~$ ping www.google.com
PING www.google.com (172.217.4.196) 56(84) bytes of data.
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=1 ttl=63 time=
4.35 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=2 ttl=63 time=
3.96 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=3 ttl=63 time=
9.26 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=4 ttl=63 time=
7.26 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 3.963/6.287/9.263/2.176 ms
alex1@alex1-VirtualBox:~$
```

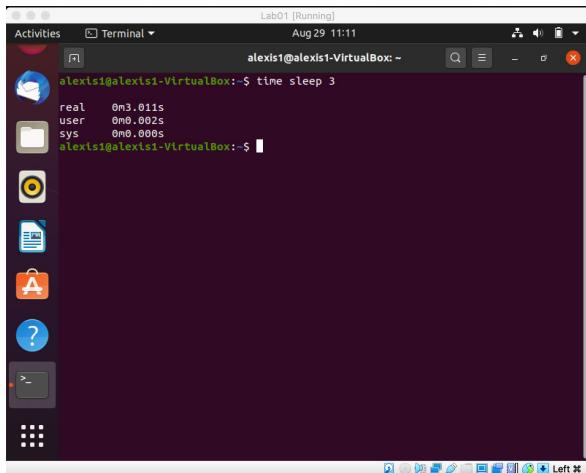
uu. traceroute: Outputs the route of a packet to the host.

```
alex1@alex1-VirtualBox:~$ traceroute www.google.com
traceroute to www.google.com (172.217.5.4), 30 hops max, 60 byte packets
1 _gateway (10.0.2.2) 0.241 ms 0.201 ms 0.182 ms
2 testwifi.here (192.168.86.1) 2.049 ms 4.469 ms 4.435 ms
3 1.208.30.136.in-addr.arpa (136.30.208.1) 3.942 ms 4.798 ms 4.531 ms
4 10.224.16.25 (10.224.16.25) 4.363 ms 4.372 ms 5.649 ms
5 10.224.12.13 (10.224.12.13) 4.039 ms 4.086 ms 5.265 ms
6 10.3.4.41 (10.3.4.41) 2.740 ms 2.617 ms 3.639 ms
7 100.xe-1-0-18.mcs1.ord2.us.zip.zayo.com (64.125.80.77) 3.656 ms 3.087 ms
5.130 ms
8 ae1.cs4.ord2.us.eth.zayo.com (64.125.24.82) 5.123 ms *
9 * * *
10 74.125.50.194 (74.125.50.194) 4.275 ms 5.034 ms 4.702 ms
11 108.170.243.174 (108.170.243.174) 4.074 ms 5.542 ms 108.170.243.193 (108.
170.243.193) 4.897 ms
12 209.85.255.145 (209.85.255.145) 4.920 ms 6.438 ms 209.85.255.173 (209.85.
255.173) 5.021 ms
13 ord38s19-in-f4.1e100.net (172.217.5.4) 4.338 ms 4.283 ms 4.251 ms
alex1@alex1-VirtualBox:~$
```

vv. date: Displays the date and time.

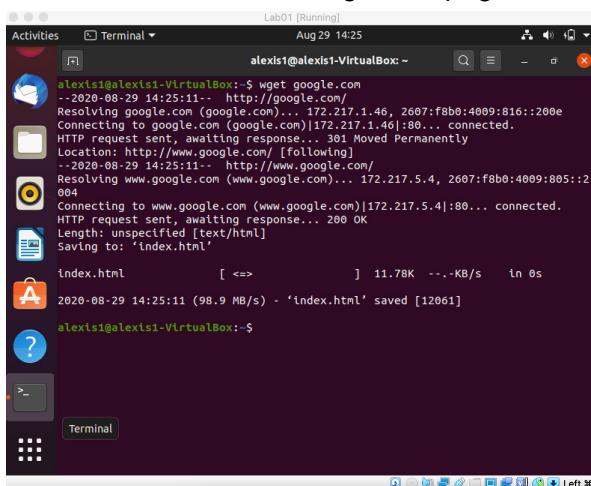
```
alex1@alex1-VirtualBox:~$ date
Sat 29 Aug 2020 02:22:37 PM CDT
alex1@alex1-VirtualBox:~$
```

ww. time: Returns the amount of time a given command takes to load. Below example creates a program that essentially stalls for 3 seconds.



```
alex1si@alex1si-VirtualBox:~$ time sleep 3
real 0m3.011s
user 0m0.002s
sys 0m0.000s
alex1si@alex1si-VirtualBox:~$
```

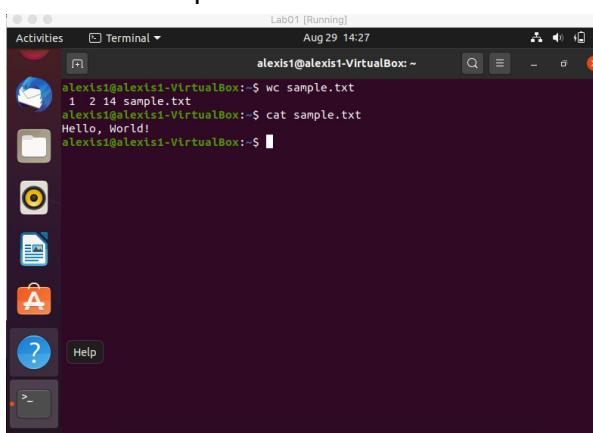
xx. wget: Allows you to download files from the internet. Below is an example of downloading a webpage.



```
alex1si@alex1si-VirtualBox:~$ wget google.com
--2020-08-29 14:25:11- http://google.com/
Resolving google.com (google.com)... 172.217.1.46, 2607:f8b0:4009:816::200e
Connecting to google.com (google.com)|172.217.1.46|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2020-08-29 14:25:11- http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.5.4, 2607:f8b0:4009:805::2
004
Connecting to www.google.com (www.google.com)|172.217.5.4|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html [=> ] 11.78K --.-KB/s in 0s
2020-08-29 14:25:11 (98.9 MB/s) - 'index.html' saved [12061]
alex1si@alex1si-VirtualBox:~$
```

yy. wc: Stands for word count. Outputs the number of lines, words, and characters in provided files.



```
alex1si@alex1si-VirtualBox:~$ wc sample.txt
1 2 14 sample.txt
alex1si@alex1si-VirtualBox:~$ cat sample.txt
Hello, World!
alex1si@alex1si-VirtualBox:~$
```

zz. pwgen: Generates random passwords. By default, the output will be 160-8 character passwords.

```

Activities Terminal Aug 29 14:29 alexis1@alexis1-VirtualBox: ~
alexis1@alexis1-VirtualBox: ~ S pwgen
b Thunderbird Mail :VirtualBox:- S pwgen
b gaQee8 vnkFaju ahd3Loop Ox3alt0o tahFee8d lez1Raep
oCha1j0l tQuu41a Ez8Ajeju Heeku6o Yeld1Vae AhYoh7ah leh7elku ojo9Efe
o16odite eCh01dee Hulth4oh phaloH9 dig9We5s a1T8oop We8xooWo Naljo2kl
ThuSpohg thalt4al lohH8gee k18mooPh a1lo0V9u oogh400w Oo0ghhi chipe13l
uuwaeh3G aquae2di Eephai9g aequ4taG AeR1As4w C1e4lt2 Quo4igh uoboaWn
eeFoo6n sheet7Eiw hotqu11b in8Twab ah13alk6 beixooF5 Feacah9j a1Co8die
ohbooJ8 k1Wahlah phleSAYa Daif3aal koCupho5 ahnae9Jo mat9been IekSeeG9
a1h3ohF5 EPat6Gus ootEU9th rofOh3o Zae2ph1 ulc0A1sa nle7fees Eengle3e
aey0YaEM ur0UGahh thaet5he leegu5lo Ahae2al Ieh30oh2 eld2Mex1 sheC7ale
Fuchoo3u vaep4ahX L5aek2l chohy3Al uz8eek1 xlebohX2 lleTe3oo ahQu3thu
Jaeng4nu Eyeit9p Eudshae5 ve1cohG2 elYee2ad uu7Atae ete2oPee yoodPho
iaTe14oh toSa1Soo phalV6ni iChal4x shaakee7 Saij3kee lque9Ve Elvel2ad
yle4Ac5E ooke1De6 leSh7Aat Sohgeedes Ieh7ahV4 mugh1037 leSho2ee ve7gaae
Jaeh0phe pusuaAbh oacseeG4
Jal1lall Eeyubaod luT7o10l evu9ahCoo Ohbel5al elv0Pah0 X18kooole ahGh10ge
epheeBa7 ag9a19o yuQu6Aev Alp2s1ek oe8Gepleh aenuY8wu Lo5aegel Wabla9ch
aeYsooCu alX0rroog Kte2ohtoh ochah9o ooVh8yl Abhabh8ka eep4a1x7 alPh6tac
OoQu4eog inSNohte leDie6ev Elboob6n osoY3Nux ooma3Buo Oothoh0x Ophpix10
guPe1f3l aephod4Pl hoSiUp14 OuN6cae7 Noh5xah6 elD2wail aiklesch ush3iet
alexis1@alexis1-VirtualBox:~$
```

3. Write bash scripts to do the following:

a. Generate-dataset.sh

```

Activities Terminal Aug 29 15:24 alexis1@alexis1-VirtualBox: ~/Desktop
alexis1@alexis1-VirtualBox: $ cd Desktop
alexis1@alexis1-VirtualBox: ~/Desktop$ touch generate-dataset.sh
alexis1@alexis1-VirtualBox: ~/Desktop$ chmod u+x generate-dataset.sh
alexis1@alexis1-VirtualBox: ~/Desktop$ ls -l
total 4
-rwxrwx--- 1 alexis1 alexis1 153 Aug 29 15:11 generate-dataset.sh
```

```

1 #!/bin/bash
2
3 #Exception handling - making sure correct num of args provided
4 if [[ $# -lt 2 || $# -gt 2 ]]; then
5   echo "You have provided the incorrect number of arguments. You need 2."
6   exit 1
7 fi
8
9 #Assigning both args to variables
10 fileName=$1
11 numRecords=$2
12
13 #Creating temporary output file for later use
14 touch ints.txt
15
16 #Inserting 2 random integers per line, then outputting to temp file
17 for i in $(seq 1 $numRecords); do
18   echo $RANDOM $RANDOM
19 done > ints.txt
20
21 #Generating proper number of random 100-byte ASCII strings, one per line
22 #and outputting them to another temporary file
23 base64 -w 100 /dev/urandom | head -n $numRecords > str.txt
24
25 #Combining the two temporary files by columns,
26 #Resulting in one file with 2 integers and 1 string per line
27 paste ints.txt str.txt > $fileName.txt
28
29 #Deleting the temporary files
30 rm ints.txt
31 rm str.txt
32
33 #Exiting the program
34 exit 0
```

```
alexisi@alexis1-VirtualBox:~/Desktop$ time ./generate-dataset.sh test01 1000000
&
[1] 2455
alexisi@alexis1-VirtualBox:~/Desktop$ disown %1
alexisi@alexis1-VirtualBox:~/Desktop$ jobs
alexisi@alexis1-VirtualBox:~/Desktop$ 
real    0m12.211s
user    0m6.775s
sys     0m5.039s
```

The ‘&’ allows it to run in the background, and the ‘disown’ command will allow it to continue running even if I disconnect from the ssh.

A screenshot of a Linux desktop environment showing a text editor window. The window title is 'test.txt'. The file content is a large text dump of generated data, starting with:

```
1|0358 2337
2|25169 29108
3|1259 2754
4|30371 19381
5|25477 19213
ktujo<449e9g3BXZHT45MFUkK1h2ggqqyTVJlfbnZYlsx4LPvjhRDQxsqxoyjeiWa2aVChZL
6|18322 20244
KR6KOVR1iy5uuV1n9CbvmE1MRSMOasGMJwgdbk7aq0rfWjltOtc2z5p3HRFv7JjdRGVjMoNc
7|12352 16876
CwAlumF0D7qqo05ef8id3pAGWP7MG550aUQohCwp01VmC7PmnTPQksnAshMJKidSqZewtYw
8|3673 2211
nFN9d9uNcOMYnV3LT81nderq0vpkjKwyoKxy4gzbLoNsK2ybhrsGVSzZkjkRlnba3jhTJE
9|4303 18108
4A18deJL2sBa3PC0f5heUzVGshTFn7WllCaPC3lndnzuEcssekal4fZGUjPEyHzTp3Svhjz
10|31495 9864
yn8jfKnmHtLkkfJ9AWh9edVlxXXFw76EkFRz1uAnYa6wsJVjbgn02KKj0AaB5M9nJVMtU
11|1770 12822
8CGas2YlrJ5upvhla0Qg7s29ncmCz3ToeRbhhdjqZ Nzj9u3E05XMytnalSFa2H7eaFL7cSff
12|17598 14250
6dau2vp31Pksv1nyu0xeNn27ZQIqbGn9WYDt21GyWtrbyst80j0VfFj7zLucTxX0gBfwXtvVs
13|5078 30570
```

File contains 1,000,000 records after running.

b. sort-data.sh

Creating the sort-data script

A screenshot of a Linux desktop environment. At the top, there's a dock with icons for various applications like a file manager, terminal, and system settings. Below the dock is a terminal window titled "Lab01 [Running]" with the command "alexiss1@alexiss1-VirtualBox:~/Desktop\$ touch sort-data.sh" entered. Below the terminal is a code editor window showing the content of "sort-data.sh".

```

#!/bin/bash
#Exception handling - making sure correct num of args provided
if [[ $# -lt 1 || $# -gt 1 ]]; then
    echo "You have provided the incorrect number of arguments. You need 1."
    exit 1
#Assigning argument to a variable
fileName=$1
#Exception handling - Checking the file exists
if [[ ! -e "$fileName" ]]; then
    echo "No such file exists."
    exit 2
fi
#Sorting the file, then replacing the original data with the sorted
#data in the same file
sort -k1 -n -o $fileName $fileName
#Successful exit
exit 0

```

Testing the sort-data script

A screenshot of a macOS terminal window. It shows three commands being run with the "time" command to measure execution time:

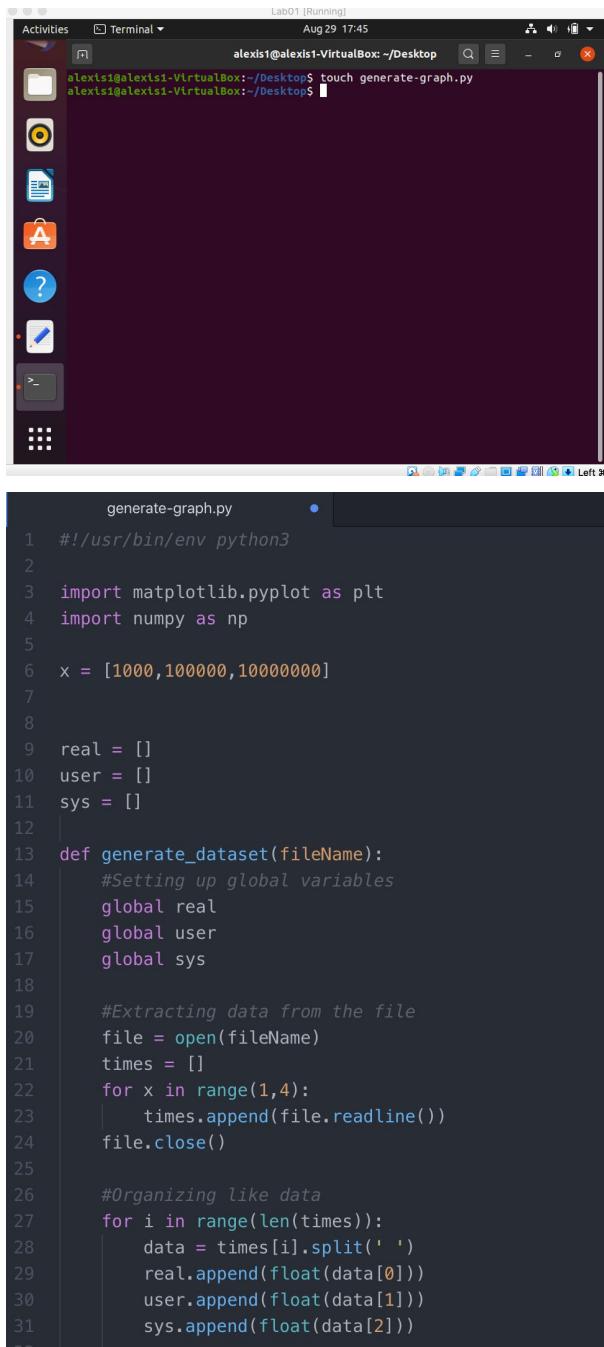
```

alexiss-mbp:Desktop lex$ time ./sort-data.sh thousand.txt
real    0m0.209s
user    0m0.018s
sys     0m0.017s
alexiss-mbp:Desktop lex$ time ./sort-data.sh hunthous.txt
real    0m0.745s
user    0m0.757s
sys     0m0.114s
alexiss-mbp:Desktop lex$ time ./sort-data.sh 10mil.txt
real   11m39.849s
user   4m46.863s
sys    5m24.351s
alexiss-mbp:Desktop lex$ 

```

c. generate-graph.py

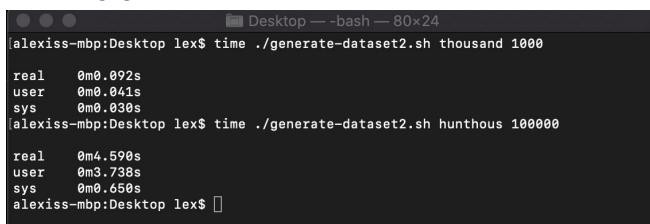
Creating the python script



A screenshot of a Linux desktop environment showing a terminal window titled "Terminal". The terminal window has a dark background and contains the following Python code:

```
generate-graph.py
1  #!/usr/bin/env python3
2
3  import matplotlib.pyplot as plt
4  import numpy as np
5
6  x = [1000, 10000, 1000000]
7
8
9  real = []
10 user = []
11 sys = []
12
13 def generate_dataset(fileName):
14     #Setting up global variables
15     global real
16     global user
17     global sys
18
19     #Extracting data from the file
20     file = open(fileName)
21     times = []
22     for x in range(1,4):
23         times.append(file.readline())
24     file.close()
25
26     #Organizing like data
27     for i in range(len(times)):
28         data = times[i].split(' ')
29         real.append(float(data[0]))
30         user.append(float(data[1]))
31         sys.append(float(data[2]))
32
```

Running generate-dataset for the 1,000 , 100,000 , and 10,000,000 records



A screenshot of a Mac OS X terminal window titled "Desktop — bash — 80x24". The terminal window contains the following output:

```
alexiss-mbp:Desktop lex$ time ./generate-dataset2.sh thousand 1000
real    0m0.092s
user    0m0.041s
sys     0m0.030s
alexiss-mbp:Desktop lex$ time ./generate-dataset2.sh hunthous 100000
real    0m4.590s
user    0m3.738s
sys     0m0.650s
alexiss-mbp:Desktop lex$ []
```

```
alexiss-mbp:Desktop lex$ time ./generate-dataset2.sh int1 10000000  
  
real    7m44.885s  
user    5m44.824s  
sys     0m58.117s
```

Above is example output from my script

Sorting the data:

```
alexiss-mbp:Desktop lex$ time ./sort-data.sh thousand.txt
real    0m0.209s
user    0m0.018s
sys     0m0.017s
alexiss-mbp:Desktop lex$ time ./sort-data.sh hunthous.txt
real    0m0.745s
user    0m0.757s
sys     0m0.114s
alexiss-mbp:Desktop lex$ time ./sort-data.sh 10mil.txt
real    11m30.040s
user    4m46.863s
sys     5m24.351s
alexiss-mbp:Desktop lex$ 
```

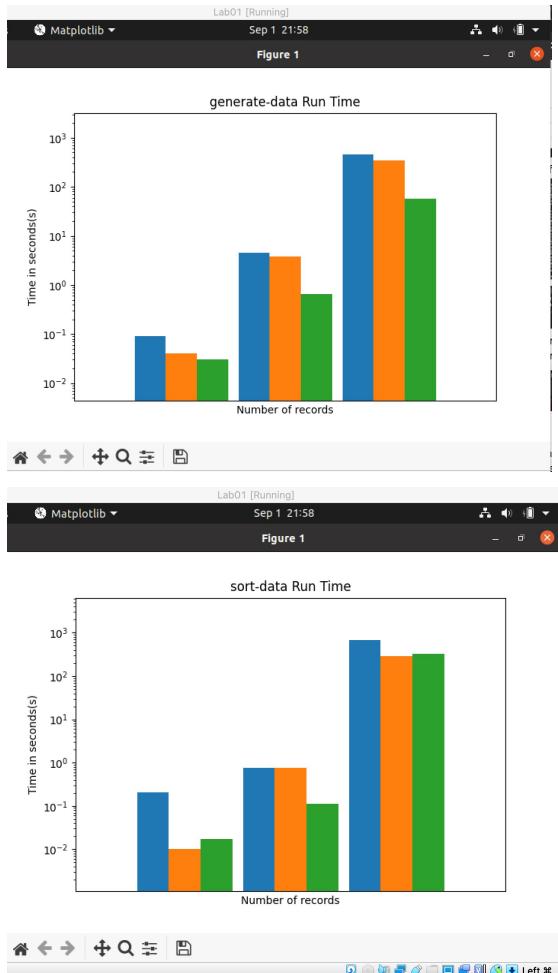
Before and After example:

Runtime .txt files used to generate the graphs:

gen-data-ti...	sort-data-ti...
0.092 0.041 0.030	0.209 0.010 0.017
4.590 3.738 0.650	0.745 0.757 0.114
464.885 344.824 58.117	690.040 286.863 324.351

Running generate-graph.py:

```
alex1@alex1-VirtualBox:~/Desktop$ python3 ./generate-graph.py
```



4. Answer the following questions about VMs:

- The number of processors essentially affects the speed and efficiency of the virtual machine. Fewer processors will result in a faster, more efficient virtual machine. More processors will result in a slower, less efficient virtual machine. Processors can be set to a minimum when only desktop applications, such as Microsoft Office, and web applications, such as Google Docs, are being used. Processors may be set to a maximum to increase the performance of more complicated, arduous tasks such as video encoding or data-intensive applications. However, this can be problematic for the host because it may prevent it from running necessary, background applications.
- Paravirtualization is allowing several operating systems to run on one set of hardware. This is achieved by efficiently allocating resources such as processors and memory.

None: This denies the exposure of any paravirtualization interface

Legacy: This option is for virtual machines that were created with older versions

of VirtualBox. It is my understanding that once these virtual machines are launched with the newer versions of VirtualBox, a different paravirtualization option will be chosen.

Minimal: This reports the presence of the virtual machine. It also reports the TSC (Time Stamp Counter) and APIC (Advanced Programmable Interrupt Controller) frequencies.

Hyper-V: This option provides a Microsoft Hyper-V hypervisor interface.

KVM: KVM stands for “Kernel-based Virtual Machine.” This option provides a Linux KVM hypervisor interface.

The main difference between all of these options is the interface they provide.

The KVM is the best choice for Ubuntu Linux users because its interface is recognized by Linux kernels.

- c. IDE: IDE stands for “Integrated Drive Electronics.” It is an interface that connects the computer’s motherboard to its disk storage devices. This option would typically only be used when it is the only option the guest supports since it is an older standard.

SATA: SATA stands for “Serial Advanced Technology Attachment.” SATA is a newer standard than IDE. It is faster and supports more devices per controller. This option could be chosen if the computer will be used for simple things like web browsing, gaming, and using desktop applications.

NVMe: NVMe stands for “Non-Volatile Memory Express.” This is even newer and faster than SATA. This option can accept more commands with less latency. This option is best suited for more complicated tasks such as compression and decompression, large file transfers, or transcoding.

- d. NAT: NAT stands for “Network Address Translation.” NAT is the simplest option in which the VM acts as a computer set up to a router. This option is ideal for simple web-browsing or checking emails within the virtual machine.

Bridged Adapter: The bridged adapter communicates with the host to exchange network packets. This is ideal for more advanced needs, such as running servers.

Internal Network: Internal networking is similar to bridged networking, but with increased privacy. An internal network may communicate with and have access to the host. However, the host does not have that same access to the VM. Information may only be shared between VMs on the same host connected to the

same network. This option is best used in scenarios when you want to hide data from the host system.

Host-Only Network: This can be thought of as a combination of bridged and internal networking. A loopback interface is created on the host. The data and communication between virtual machines will not be able to be seen. This option is commonly used for virtual appliances, which means multiple virtual machines working together

- e. USB 1.1: This is also referred to as Full-Bandwidth USB. It has a maximum bandwidth of 12 Mbps. It would be ideal if the only thing you are connecting is something like a USB mouse or keyboard.

USB 2.0: Also referred to as “Hi-Speed USB,” 2.0 has a maximum bandwidth of 480 Mbps. It is backwards compatible for the USB 1.1. This option would be chosen if you are planning to use something requiring the extra bandwidth like a storage device.

USB 3.0: This USB may also be called “SuperSpeed,” and offers an increased bandwidth of 4.8 Gbps. It is also backwards compatible for both USB 2.0 and 1.1. This is best when using devices such as larger storage devices or video adapters.

Information provided by

<https://www.cablestogo.com/learning/library/standards-specs-certs/when-to-use-usb>

- f. **Most information was derived from the VirtualBox manual found at
<https://www.virtualbox.org/manual/>