

CEIS 106 FINAL COURSE PROJECT ON OPERATING SYSTEMS



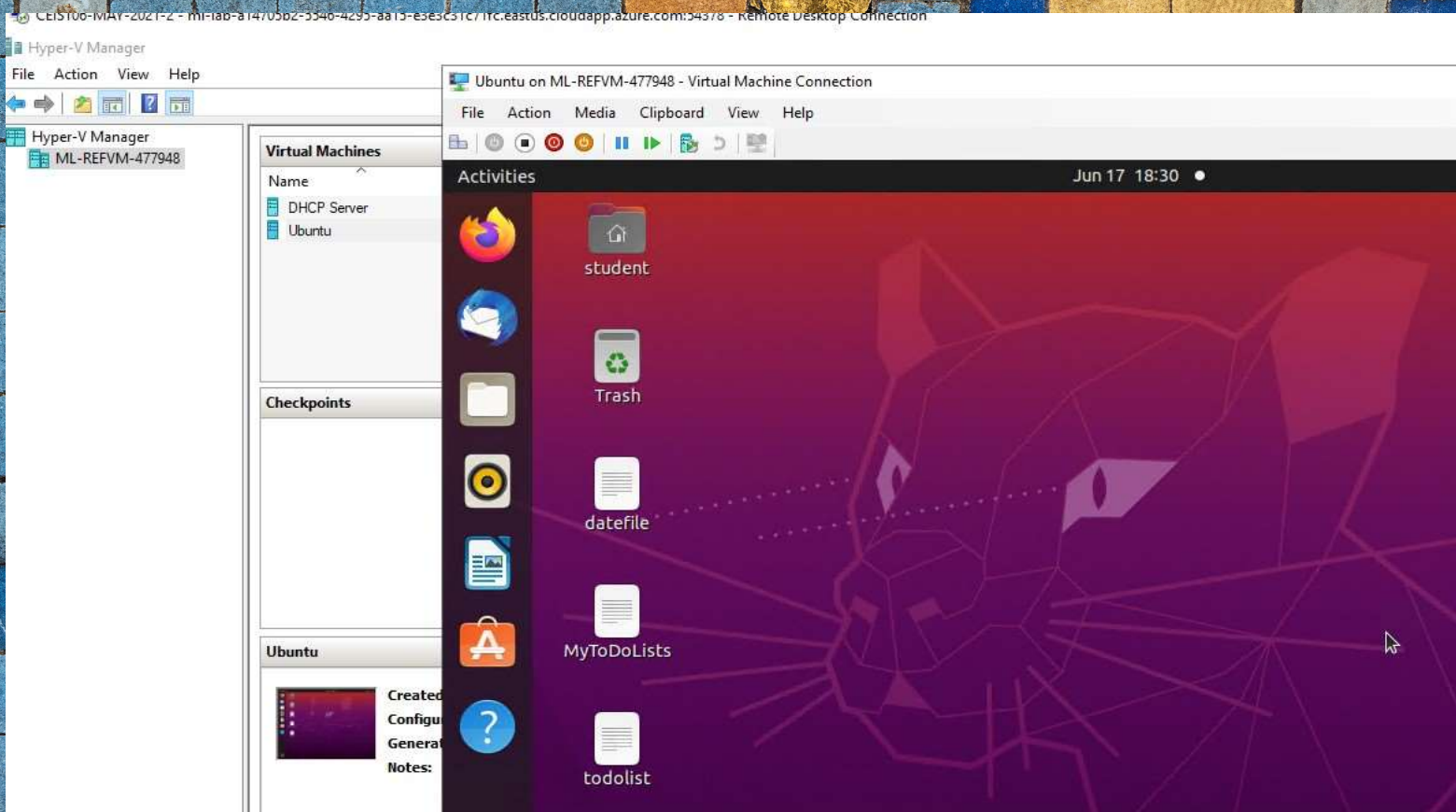
PRESENTED BY: MARYAM ZISHAN

The background of the slide is an abstract composition of flowing, translucent lines in shades of blue, teal, and green. These lines create a sense of movement and depth, resembling liquid or smoke captured in time. The colors are vibrant yet slightly muted, giving it a professional and artistic feel.

Introduction

This PowerPoint Demonstrates a detailed Summary of knowledge I gained over 8 weeks in Ceis106 Introduction to Operating System at DeVry University under the directions of Professor Simon Obeid.

Screenshot of Installed Linux's Ubuntu downloaded from software called VMware workstation through Azure Las services



LINUX FILE SYSTEM HIRARCHY

- It's a tree structure that organizes files into directories and folders. Starting from the root, directories can contain subdirectories and files within them. Files in a directory are physically stored in the filesystem of a disk partition. Hence, without a solid understanding of filesystem hierarchy it can be very difficult to store and retrieve data in Linux.
- Some of the useful commands are as following: `cd ~`, `pwd`, `tree -d -L 2.`, `cd Documents`, `cd ..`, `cd /`, `tree -d -L 3.` etc.

the LiNavigate The Linux File System Tree

1. What is the *pwd* command an acronym for? What about the *cd* command?

Answer here: The *pwd* command is an acronym for “Print Working Directory”. It’s used a lot in Linux system. The *cd* command stand for ‘change directory’ and just like the name suggest it helps change the location of the current directory to any other directory of your choice.

2. Explain the differences between a relative path and an absolute/full path in Linux.

Answer here: The absolute path or full-pathway starts from root and use / command and use absolute URL. Whereas. The relative path also known as non-absolute pathway or parted pathway contains a specified current path only and uses two dots “..” command along with relative URL.

References:

1 a. www.maketecheasier.com/pwd-command-linux-guide/

b. linuxhandbook.com/cd-command-examples/

2. [Difference Between Absolute and Relative Path \(With Table\) – Ask Any Difference](#)


```
e3c31c71fc.eastus.cloudapp.azure.com:54378 - Remote Desktop Connection

May 15 22:42 •

student@ubuntuvm: ~/JanFebSession/Course1

~/Desktop$ cd ~
~$ pwd
~/Desktop
~$ mkdir JanFebSession
~$ cd JanFebSession
~/JanFebSession$ mkdir Course1
~/JanFebSession$ mkdir Course2
~/JanFebSession$ mkdir Course3
~/JanFebSession$ cd Course1
~/JanFebSession/Course1$ touch file1 file2 file3
~/JanFebSession/Course1$ tree -d -L 2 ~

~/JanFebSession/Course1$ ls -l ~/JanFebSession/Course1

-rw-rw-r-- 1 student student 0 May 15 22:37 file1
-rw-rw-r-- 1 student student 0 May 15 22:37 file2
-rw-rw-r-- 1 student student 0 May 15 22:37 file3
~/JanFebSession/Course1$
```

Create Directories & Files

Here I created a subdirectory by entering `mkdir JanFebSession`, new directories were created by using `touch` command and `tree -d -L 2` command was used to view the entire home directory tree along with newly created subdirectories. Command `ls -l ~/JanFebSession/Course1` was entered to list the three new files as shown in the screenshot on the left.

Remove directories and Files

Here rmdir was used to remove the subdirectories and rm to remove file from the subdirectories as shown in the image

The tree command along with the name of subdirectory can be used to verify the successful removal of the file and subdirectory.

705b2-5546-4295-aa15-e3e3c31c71fc.eastus.cloudapp.azure.com:54378 - Remote Desktop Connection



Terminal

May 15 22:51



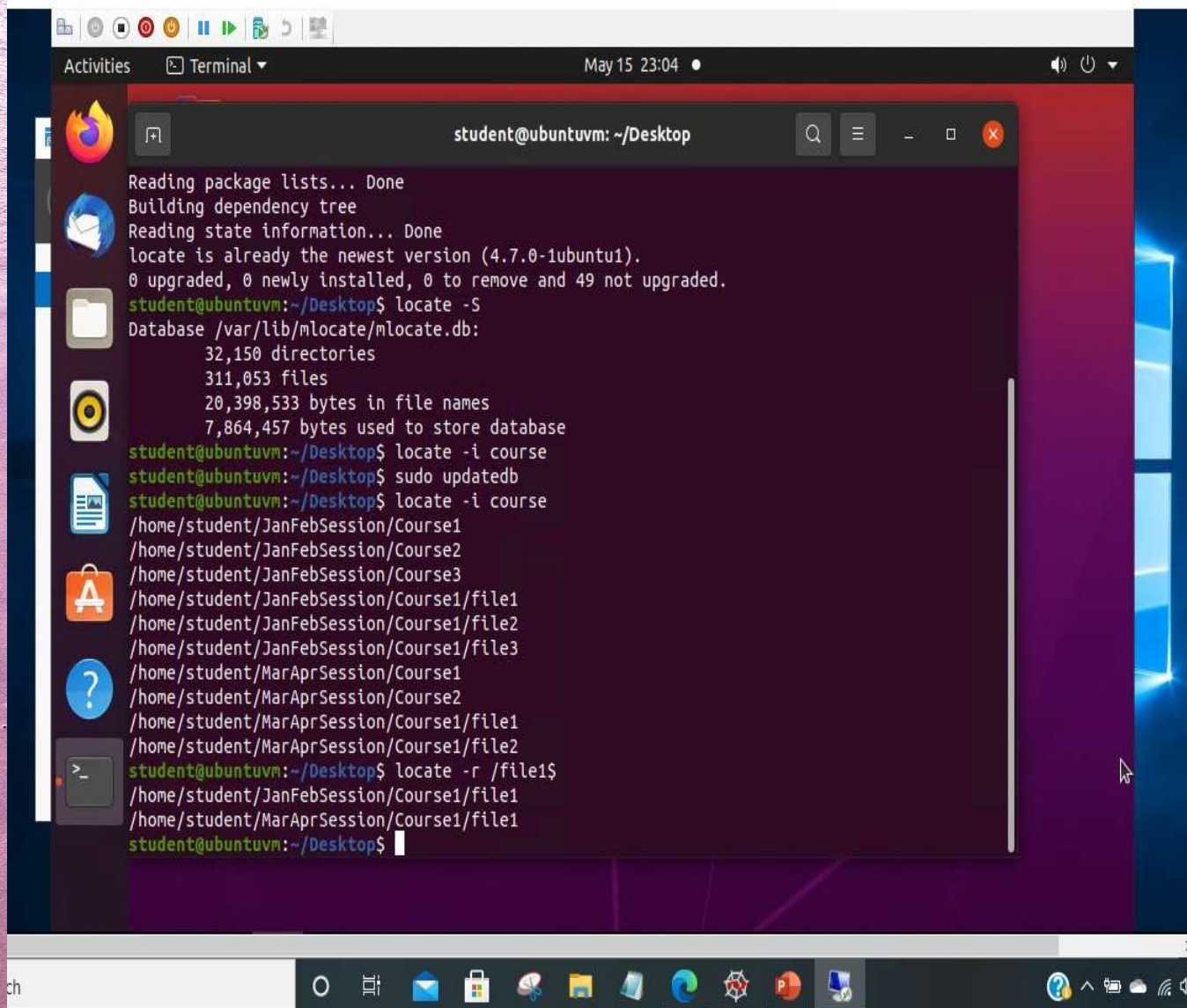
student@ubuntuvm: ~



```
student@ubuntuvm:~$ pwd
/home/student
student@ubuntuvm:~$ cp -r JanFebSession MarAprSession
student@ubuntuvm:~$ tree JanFebSession MarAprSession
JanFebSession
├── Course1
│   ├── file1
│   ├── file2
│   └── file3
├── Course2
└── Course3
MarAprSession
├── Course1
│   ├── file1
│   ├── file2
│   └── file3
├── Course2
└── Course3

6 directories, 6 files
student@ubuntuvm:~$ rmdir MarAprSession/Course3
student@ubuntuvm:~$ rm MarAprSession/Course1/file3
student@ubuntuvm:~$ tree JanFebSession MarAprSession
JanFebSession
├── Course1
│   ├── file1
│   ├── file2
│   └── file3
├── Course2
└── Course3
MarAprSession
├── Course1
```





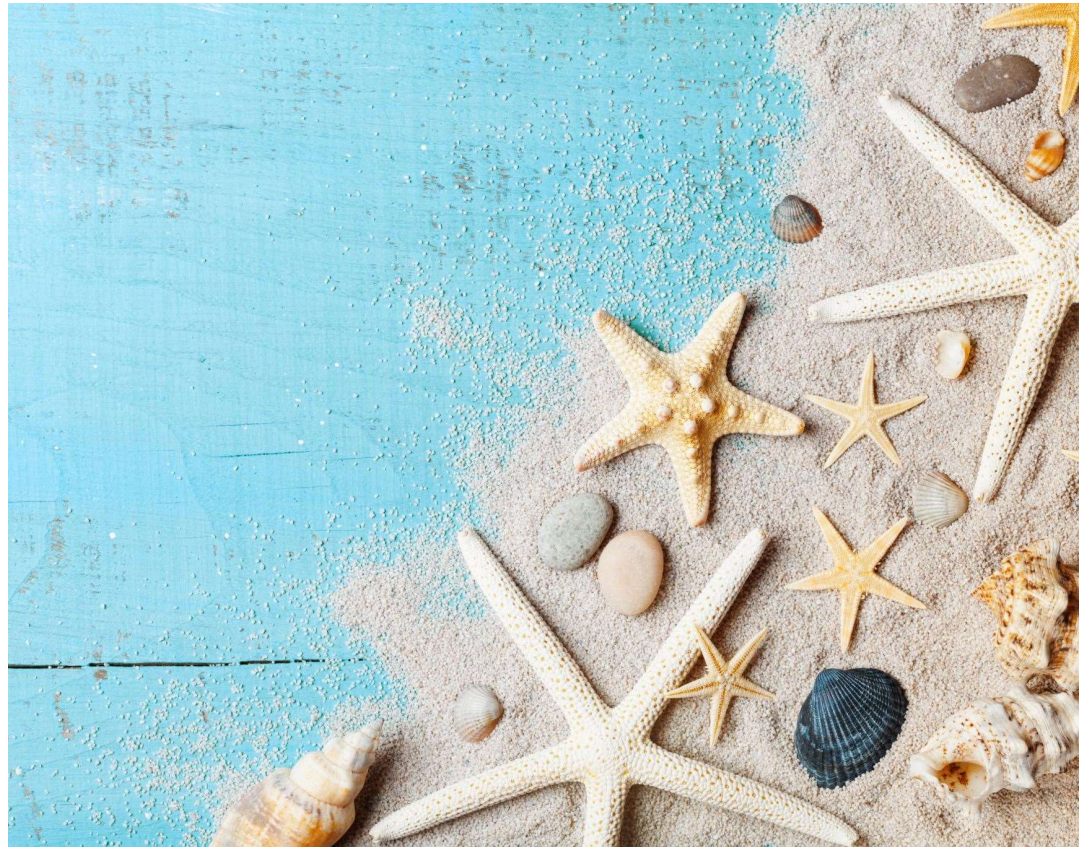
```
student@ubuntuvm: ~/Desktop
Reading package lists... Done
Building dependency tree
Reading state information... Done
locate is already the newest version (4.7.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
student@ubuntuvm:~/Desktop$ locate -S
Database /var/lib/mlocate/mlocate.db:
 32,150 directories
311,053 files
20,398,533 bytes in file names
7,864,457 bytes used to store database
student@ubuntuvm:~/Desktop$ locate -i course
student@ubuntuvm:~/Desktop$ sudo updatedb
student@ubuntuvm:~/Desktop$ locate -i course
/home/student/JanFebSession/Course1
/home/student/JanFebSession/Course2
/home/student/JanFebSession/Course3
/home/student/JanFebSession/Course1/file1
/home/student/JanFebSession/Course1/file2
/home/student/JanFebSession/Course1/file3
/home/student/MarAprSession/Course1
/home/student/MarAprSession/Course2
/home/student/MarAprSession/Course1/file1
/home/student/MarAprSession/Course1/file2
student@ubuntuvm:~/Desktop$ locate -r /file1$
/home/student/JanFebSession/Course1/file1
/home/student/MarAprSession/Course1/file1
student@ubuntuvm:~/Desktop$
```

Locate Directories & Files

I can locate directories and files by using the command `locate`—and to search for the exact directory or file name, use `-e` command and the `$` character at the end of the each term.

LINUX SHELL SCRIPTS

A shell script is a file that contains Linux commands and special constructs. It is used to perform and automate administrative tasks, by combining a lengthy sequence of commands into one script. A script is interpreted and executed by the Linux shell. Hence, any command that can be entered on the Linux command line can be part of a shell script. A shell script is often created as an ASCII text file by using a text editor program. I will create and execute a shell script by experimenting with the standard input, user defines



Create a shell script

- 1. What are the file permissions of the script?
- Answer here:

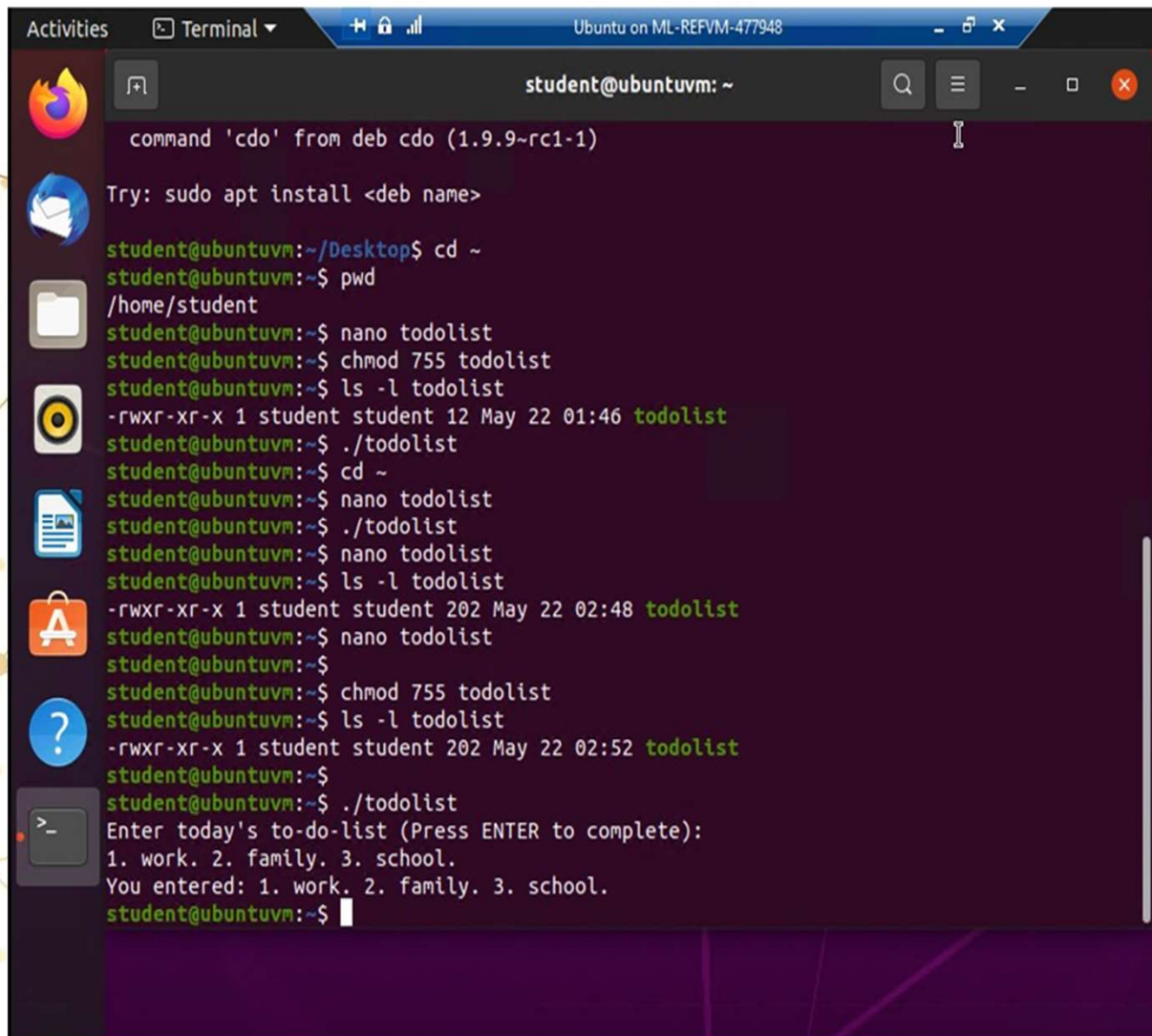
```
student@ubuntuvm:~/Desktop$ nano todolist
student@ubuntuvm:~/Desktop$ nano todolist
student@ubuntuvm:~/Desktop$ ls -l todolist
-rw-rw-r-- 1 student student 189 May 22 01:31 todolist
student@ubuntuvm:~/Desktop$
```

- 2. What's the name of the user-defined variable in the script?
- Answer here: Text is the name of the user-defined variable in the script.

- 3. Which redirection meta-character is used in the script? What does it do?
- Answer here: The meta-character used in script is ">>". It redirects output from date to the file MyToDoLists.

Change Script File Permissions

The scripts can be made executable by using `chmod 755` command and the command `ls -l` can be used to verify the file permissions. Command `./todolist` can be used to execute the script by calling its name. Note that the `.(dot)` character represents the current directory, and `./todolist` is the relative path name of the script.



```
command 'cdo' from deb cdo (1.9.9-rc1-1)

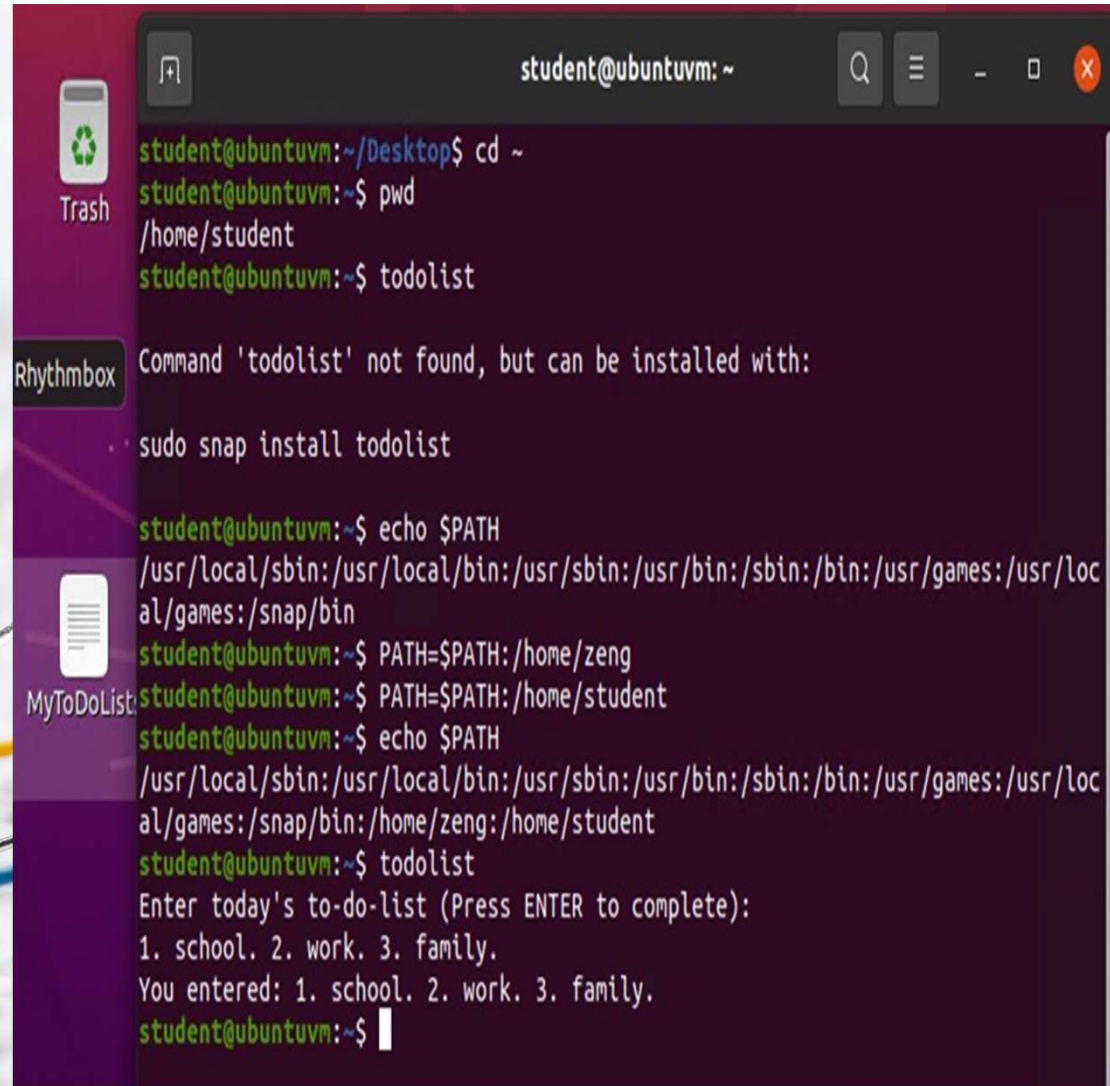
Try: sudo apt install <deb name>

student@ubuntuvm:~/Desktop$ cd ~
student@ubuntuvm:~$ pwd
/home/student
student@ubuntuvm:~$ nano todolist
student@ubuntuvm:~$ chmod 755 todolist
student@ubuntuvm:~$ ls -l todolist
-rwxr-xr-x 1 student student 12 May 22 01:46 todolist
student@ubuntuvm:~$ ./todolist
student@ubuntuvm:~$ cd ~
student@ubuntuvm:~$ nano todolist
student@ubuntuvm:~$ ./todolist
student@ubuntuvm:~$ nano todolist
student@ubuntuvm:~$ ls -l todolist
-rwxr-xr-x 1 student student 202 May 22 02:48 todolist
student@ubuntuvm:~$ nano todolist
student@ubuntuvm:~$
student@ubuntuvm:~$ chmod 755 todolist
student@ubuntuvm:~$ ls -l todolist
-rwxr-xr-x 1 student student 202 May 22 02:52 todolist
student@ubuntuvm:~$
student@ubuntuvm:~$ ./todolist
Enter today's to-do-list (Press ENTER to complete):
1. work. 2. family. 3. school.
You entered: 1. work. 2. family. 3. school.
student@ubuntuvm:~$
```

Set The Path Variable

Path creates a list of directories and can be set by entering the command `echo $PATH`

6.



```
student@ubuntuvm: ~  
student@ubuntuvm:~/Desktop$ cd ~  
student@ubuntuvm:~$ pwd  
/home/student  
student@ubuntuvm:~$ todolist  
Command 'todolist' not found, but can be installed with:  
sudo snap install todolist  
student@ubuntuvm:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
student@ubuntuvm:~$ PATH=$PATH:/home/zeng  
student@ubuntuvm:~$ PATH=$PATH:/home/student  
student@ubuntuvm:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/zeng:/home/student  
student@ubuntuvm:~$ todolist  
Enter today's to-do-list (Press ENTER to complete):  
1. school. 2. work. 3. family.  
You entered: 1. school. 2. work. 3. family.  
student@ubuntuvm:~$
```



```
student@ubuntuvm:~$ ls -a .bash*
.bash_history .bash_logout .bashrc
student@ubuntuvm:~$ cp .bashrc .bashrc.old
student@ubuntuvm:~$ nano .bashrc
student@ubuntuvm:~$ todolist

Command 'todolist' not found, but can be installed with:

sudo snap install todolist

student@ubuntuvm:~$ source .bashrc
student@ubuntuvm:~$ todolist
Enter today's to-do-list (Press ENTER to complete):
```

Make the PATH variable permanent

You can set a path variable permanently by modifying the hidden .bashrc file which runs every time a new terminal window is opened. To do this enter command `ls -a .bash*` to list the .bashrc file. You can also make a copy of it by entering `cp .bashrc .bashrc.old` command. You can use `nano .bashrc` to open the .bashrc file. You can add a new line command by `export PATH=$PATH:/home/username`. Press `^x`(control+x) and answer `y` to the question "Save modified buffer?" and press enter to keep the file name same. To active the change enter `source .bashrc`. Now you can easily open new terminal window and run the todolist script in both windows.

```
student@ubuntuvm:~$ cd ~
student@ubuntuvm:~$ pwd
/home/student
student@ubuntuvm:~$ todolist
```

```
Command 'todolist' not found, but can be installed with:

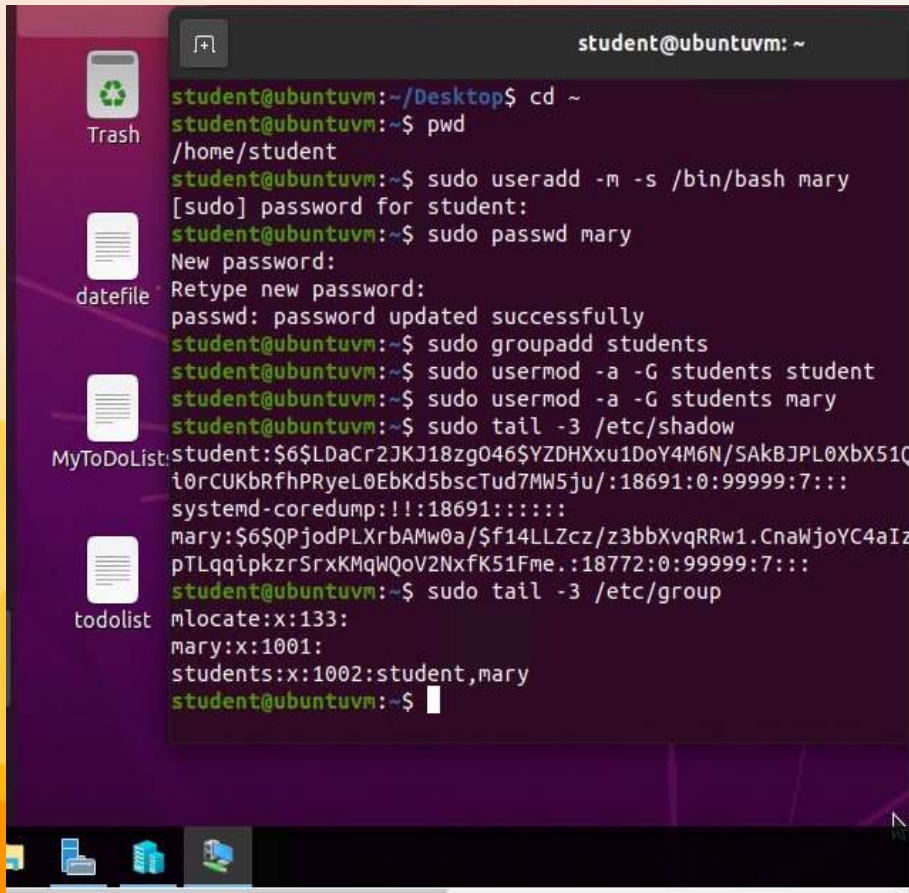
sudo snap install todolist
```

User & Group Management

User and group management is a common Linux administration task. Linux system requires a valid username and password like any other system for a user to log in. Only an authenticated user is granted access to system resources (e.g., files and directories) based on the user account. Groups are created so that administrative rights and file access are assigned to a group of users rather than one user at a time. To manage Linux users and groups, both Command Line Interface (CLI) and Graphical User Interface (GUI) can be utilized



Add users and groups in CLI



A terminal window titled 'student@ubuntuvm: ~' showing a series of commands and their outputs. The user 'student' is at the prompt. The commands and outputs are as follows:

```
student@ubuntuvm:~/Desktop$ cd ~
student@ubuntuvm:~$ pwd
/home/student
student@ubuntuvm:~$ sudo useradd -m -s /bin/bash mary
[sudo] password for student:
student@ubuntuvm:~$ sudo passwd mary
New password:
Retype new password:
passwd: password updated successfully
student@ubuntuvm:~$ sudo groupadd students
student@ubuntuvm:~$ sudo usermod -a -G students student
student@ubuntuvm:~$ sudo usermod -a -G students mary
student@ubuntuvm:~$ sudo tail -3 /etc/shadow
student:$6$LDaCr2JKJ18zg046$YZDHXxu1DoY4M6N/SAkBjPL0XbX510
i0rCUKbRfhPRyeL0EbKd5bscTud7MW5ju/:18691:0:99999:7:::
systemd-coredump:!!:18691:0:99999:7:::
mary:$6$QPjodPLXrbAMw0a/$f14LLZcz/z3bbXvqRRw1.CnaWjoYC4aIz
pTLqqipkzrSrxKMqWQoV2NxfK51Fme.:18772:0:99999:7:::
student@ubuntuvm:~$ sudo tail -3 /etc/group
mlocate:x:133:
mary:x:1001:
students:x:1002:student,mary
student@ubuntuvm:~$
```

1. What does the `-m` option in the `useradd` command do?

Answer here: Its used to add a new user and only root user can do that!

2. What does the `-3` option in the `tail` command do?

Answer here: It's used to print last K number of lines.

3. Which line of the `/etc/group` file lists members of the "students" group? Copy it here.

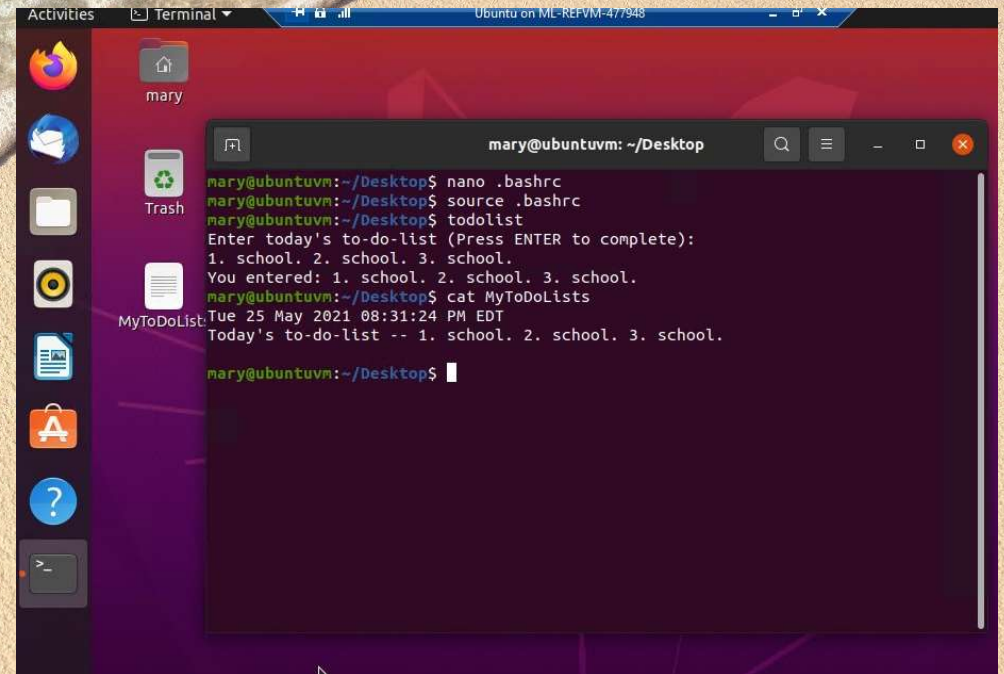
References:

1. Google

Test User & Group Settings

This can be done by using command `sudo chgrp students todolist` to change ownership of the todolist scripts to the "students" group. These changes can be verified by using `ls -l` command. `Sudo chmod 750 todolist` to modify the file permissions of the todolist script so others can use the file as well. Again this can be verified by using the command `ls -l`, then log out and log in as new user from the drop down menu. Then use `nano .bashrc` and open a new file in a new terminal window. In order to activate the change add a new line `export`

`PATH=$PATH:/home/student` and enter the command `source .bashrc`. Finally, enter the todolist to execute the script and to view this new entry use `cat MyToDoLists`



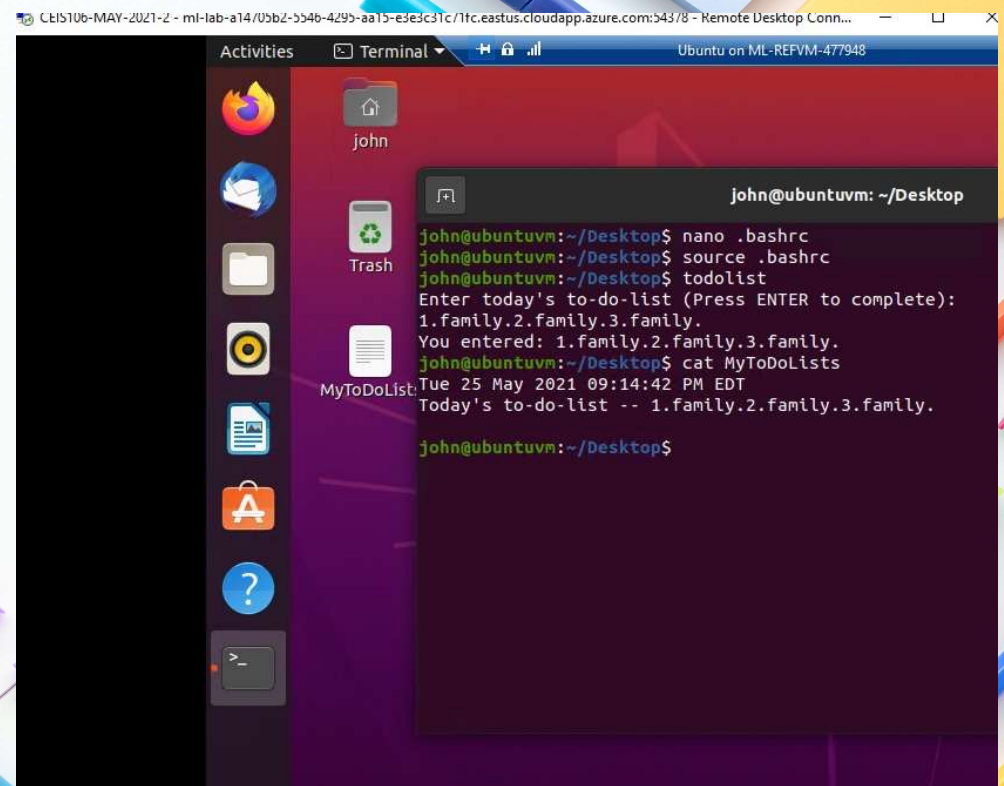
The screenshot shows an Ubuntu desktop with a purple background. A terminal window is open, displaying the following commands and output:

```
mary@ubuntuvm: ~/Desktop
mary@ubuntuvm:~/Desktop$ nano .bashrc
mary@ubuntuvm:~/Desktop$ source .bashrc
mary@ubuntuvm:~/Desktop$ todolist
Enter today's to-do-list (Press ENTER to complete):
1. school. 2. school. 3. school.
You entered: 1. school. 2. school. 3. school.
mary@ubuntuvm:~/Desktop$ cat MyToDoLists
Tue 25 May 2021 08:31:24 PM EDT
Today's to-do-list -- 1. school. 2. school. 3. school.
mary@ubuntuvm:~/Desktop$
```

Take a screenshot of the output in Step 6.

Add Users in GUI

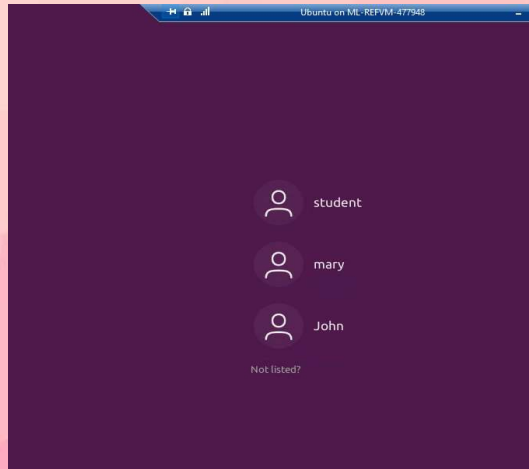
Click nine dots on the bottom left corner in Ubuntu and launch GUI by clicking Users and Groups. Then click on the Add button and create a new user with name “John” and assign a password. Next, highlight John on the user list and click Manage Groups and select “students” in the groups available on the system: list and click the “Properties” button. In the Group “students” Properties window check the name “John” on the “Group Member” list and click “ok” to close window. Click “close” twice and logout of the previous user and log in with the new user created. (ex. John)



The screenshot shows a remote desktop connection to an Ubuntu system. The desktop has a dark theme with a sidebar on the left containing icons for Firefox, Mail, Files, Music, Videos, and Applications. The main area shows a desktop with icons for 'john', 'Trash', and 'MyToDoList'. A terminal window is open, showing the following commands and output:

```
john@ubuntuvm: ~/Desktop
john@ubuntuvm:~/Desktop$ nano .bashrc
john@ubuntuvm:~/Desktop$ source .bashrc
john@ubuntuvm:~/Desktop$ todolist
Enter today's to-do-list (Press ENTER to complete):
1.family.2.family.3.family.
You entered: 1.family.2.family.3.family.
john@ubuntuvm:~/Desktop$ cat MyToDoLists
Tue 25 May 2021 09:14:42 PM EDT
Today's to-do-list -- 1.family.2.family.3.family.

john@ubuntuvm:~/Desktop$
```

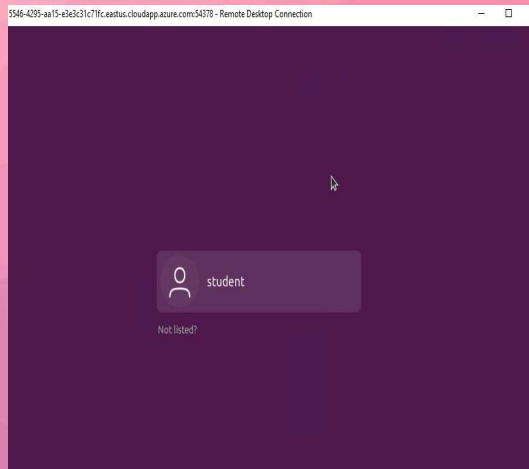


Remove users and groups

The screenshot of the log on page with three user accounts.

These users can be removed with command `sudo userdel -r xxxx` (username). And the group can be removed by using command `sudo groupdel xxxx`(group name). Then log out and you can always log in to verify only your name is listed as the user account.

The screenshot of the log on page with only your user account (i.e., student).



NETWORK CONFIGURATION

A Linux machine requires an IP configurations including an IPv4/IPv6 address, default gateway, DNS server, etc. to be able to function on a network. A unique IP consists of a network portion and a host portion, is assigned to a computer to identify itself on a network. Networks with same IP addresses can communicate with each other without the use of a routing device. Therefore, computers on a local area network (LAN) segment have IP addresses with the same network portion but different host portions. Communications beyond a LAN segment, however require a routing device (i.e., default gateway) that forwards packets to and from a different network or network segment. In the following slides you will discover the IP configurations of your Linux machine and explore different network utility programs



Discover host IP configurations

1. What is the IP address of your Ubuntu machine?

Answer here: inet 192.168.1.105/24

2. What is the IP address of its default gateway?

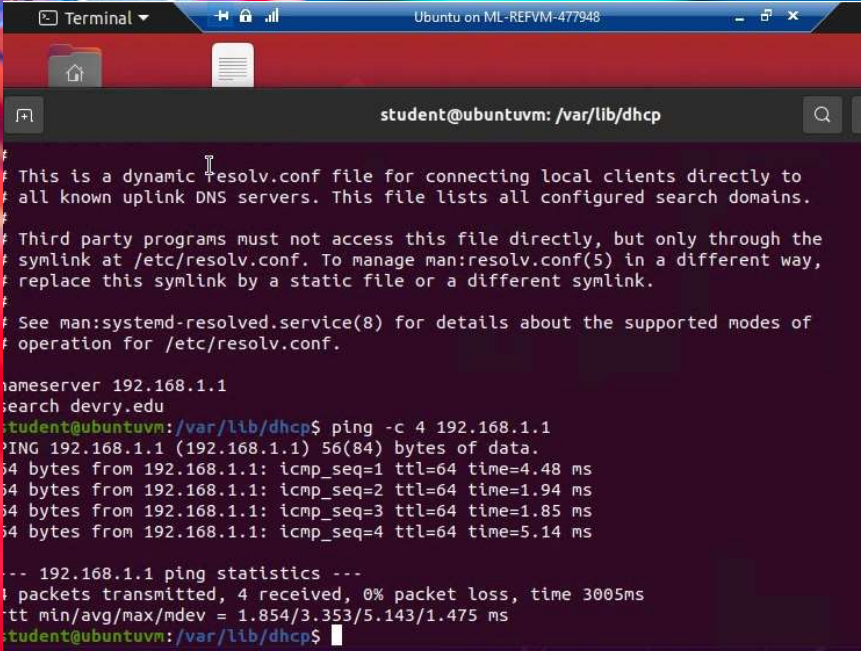
Answer here: 192.168.1.1

3. What is the IP address of its DHCP server?

Answer here: 192.168.1.1

4. What is the IP address of its DNS server?

Answer here: 192.168.1.1

A terminal window titled 'Terminal' with the subtitle 'Ubuntu on ML-REFVM-477948'. The prompt is 'student@ubuntuvm: /var/lib/dhcp'. The terminal shows the contents of a dynamic resolv.conf file, followed by a ping command: 'ping -c 4 192.168.1.1'. The output shows four successful ping requests with varying times. Finally, it displays the ping statistics: '--- 192.168.1.1 ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3005ms rtt min/avg/max/mdev = 1.854/3.353/5.143/1.475 ms'.

```
Terminal
Ubuntu on ML-REFVM-477948
student@ubuntuvm: /var/lib/dhcp

# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.

# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.

# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.1.1
search devry.edu
student@ubuntuvm: /var/lib/dhcp$ ping -c 4 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=4.48 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.94 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.85 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=5.14 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.854/3.353/5.143/1.475 ms
student@ubuntuvm: /var/lib/dhcp$
```


Manage network interfaces

1. Which DHCP message is shown in the output of the `sudo dhclient -v -r eth0` command? [hint: the message name is in uppercase.]

```
student@ubuntuvm:~$ sudo dhclient -v -r eth0
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:15:5d:00:04:01
Sending on   LPF/eth0/00:15:5d:00:04:01
Sending on   Socket/fallback
DHCPRELEASE of 192.168.1.105 on eth0 to 192.168.1.1 port 67 (xid=0x53deb5f)
```

2. Which four DHCP messages are shown in the output of the `sudo dhclient -v eth0` command? [hint: the message names are in uppercase.]

Answer here:

```
student@ubuntuvm:~$ sudo dhclient -v eth0
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:15:5d:00:04:01
Sending on   LPF/eth0/00:15:5d:00:04:01
Sending on   Socket/fallback
DHCPCDISCOVER on eth0 to 255.255.255.255 port 67 interval 3 (xid=0x24aa371c)
DHCPOFFER of 192.168.1.105 from 192.168.1.1
DHCPREQUEST for 192.168.1.105 on eth0 to 255.255.255.255 port 67 (xid=0x1c37aa24)
DHCPACK of 192.168.1.105 from 192.168.1.1 (xid=0x24aa371c)
bound to 192.168.1.105 -- renewal in 269 seconds.
```

Use network utilities

IP addresses assigned to all network interfaces can be displayed by using command `ifconfig`. Command `sudo ifconfig eth0 down` can be used to disable/deactivate the `eth0` interface. Command `ifconfig eth0` can be used to verify that the interface no longer has an IP address. Finally command `sudo ifconfig eth0 up` can be used to enable/activate the `eth0` interface. Enter `ifconfig eth0` to verify it's status as `RUNNING`. (screenshot on the left shows that)

```
5106-MAY-2021-2 - ml-lab-a14705b2-5546-4295-aa15-e3e3c31c71fc.eastus.cloudapp.azure.com:54378 - Remote Desktop Connection

student@ubuntuvm:~$ ifconfig
eth0: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 7608 bytes 638630 (638.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7608 bytes 638630 (638.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

student@ubuntuvm:~$ sudo ifconfig eth0 down
student@ubuntuvm:~$ ifconfig eth0
eth0: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 00:15:5d:00:04:01 txqueuelen 1000 (Ethernet)
    RX packets 7974 bytes 565758 (565.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4939 bytes 412596 (412.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

student@ubuntuvm:~$ sudo ifconfig eth0 up
student@ubuntuvm:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.104 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::656c:1648:75d4:44d0 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:00:04:01 txqueuelen 1000 (Ethernet)
    RX packets 7976 bytes 566433 (566.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4968 bytes 416588 (416.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

student@ubuntuvm:~$
```


SYSTEM PERFORMANCE MONITORING



Linux System Performance could be affected by factors such as the amount of RAM, CPU utilization, storage device speed, and process load. A constant investigation is essential in order to keep the system up and running. Various CLI and GUI monitoring tools can be used to monitor system performance detect root causes of any problems.

Monitoring Linux Processes

- 1. What is the default action of the *15 SIGTERM* kill signal?
- Answer here: It terminates/kill the task or application right away. For example, if an application such as calculator is running, we can immediately terminate it by pressing letter K and then pressing enter after 15 SIGTERM appears.
- 2. In the System Monitor window, click on % CPU to sort the processes by CPU load. Which process shows the highest percentage of CPU usage?
- Answer here: gnome shell show the highest % of CPU used.
- References:
- 1. Google

Monitor user activities



- Issue the **sudo accton on** command to turn on GNC accounting. Run the **sudo updatedb** command. Enter **lastcomm updatedb** to check if the **updatedb** command was executed before. Remember to turn off GNC accounting (**sudo accton off**) after answering the questions.

1. What flag value is displayed in the output?

Answer here:

```
student@ubuntuvm:~$ sudo accton on
[sudo] password for student:
Turning on process accounting, file set to the default '/var/log/account/pacct'.
student@ubuntuvm:~$ sudo updatedb
sudo: updatedb: command not found
student@ubuntuvm:~$ sudo updatedb
student@ubuntuvm:~$ sudo updatedb
student@ubuntuvm:~$ lastcomm updatedb
updatedb      S      root      pts/2      0.15 secs Sat Jun 12 19:39
updatedb      S      root      pts/2      0.17 secs Sat Jun 12 19:39
student@ubuntuvm:~$
```

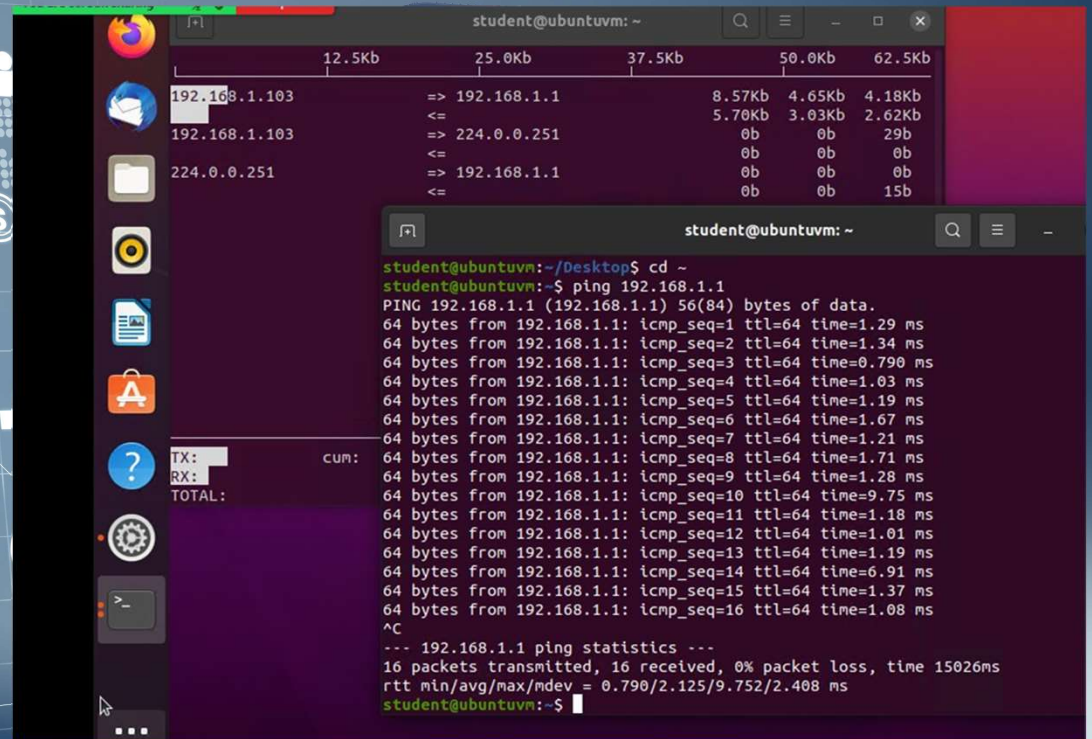
2. Why is the name of the user who ran the processes shown as root, not student?

Answer here: Since my user is the default account on Linux system it has root privileges, I used the sudo command to execute the updated process. Hence, it was run with root privileges that's why it's shown as root and not as student.

Monitoring Network Bandwidth usage

To display bandwidth usage you can use the command `sudo iftop` and in another window type in `ping 192.168.1.1`. The display can be paused by typing in upper-case P and q to close the iftop utility. Letter n can be used to toggle DNS reverse lookup on/off.

4.



CHALLENGES



Below is the list of some of the challenges I had to face as a beginner during this course project:

The commands were difficult to memorize.
Sometimes I would forget to create space between a command.
Initially setting up Ubuntu through VM was difficult.
Learning commands and their functions can be time consuming.

CAREER SKILLS

Basic career skills that I acquired during this project are:

Thinking with patience before proceeding.

Good understanding of basic Linux commands and their functions.

Networking.

Troubleshooting.

Security.

creating users and groups.

Managing IP addresses.

Analyzing and Planning.

Basic understanding of Operating Systems and their functions.

Characteristics of Linux and importance in the world today.



CONCLUSION

With some basic understanding of Operating Systems through Linux I feel more confident about my future in IT. It's been interesting experience but definitely required time and practice to excel!

