Hanish Patel

CS 497-01

Assignment 2

CS 497 – Assignment 2

**Abstract**

Examine user and group permissions and controls through Ubuntu. Understand how to set and observe user and group permissions through the command line. Understand how to create folders and text files and how different permissions may deny and allow certain users and groups to interact with the folders and files. The user and groups can also be created through the command line.

**Introduction**

man will be used for understanding commands:

man command

setfacl will be used to set controls:

setfacl -m (modify) [group:] group\_name:[r][w][x] [File Name]

getfacl will be used to explain the controls:

getfacl [name]

group add and add user will add groups and user accounts:

sudo groupadd

sudo adduser

sudo will execute commands as authorized users

sudo -i

usermod with a and g will allow users to be added to a group:

usermod -a -G group\_name user

cd is used to navigate directories and ls is used to list files, while ls -ld lists directories ls la will show all permissions of the files without ignoring .

cd path

ls [-ld] [-la]

touch will be used for creating txt files.

touch filename

date will be used to show the current date, uname -a will be used to show information such as the kernel name, node name, kernel release, etc, ip addr will show the ip address, cat will access files, and >> will append at the end of the file.

date

uname -a

ip addr

cat filename

>>

mv will move a file to a path

mv file\_name path

chown will set file owner

sudo chown new\_owner file

wc will be used to count words in the file

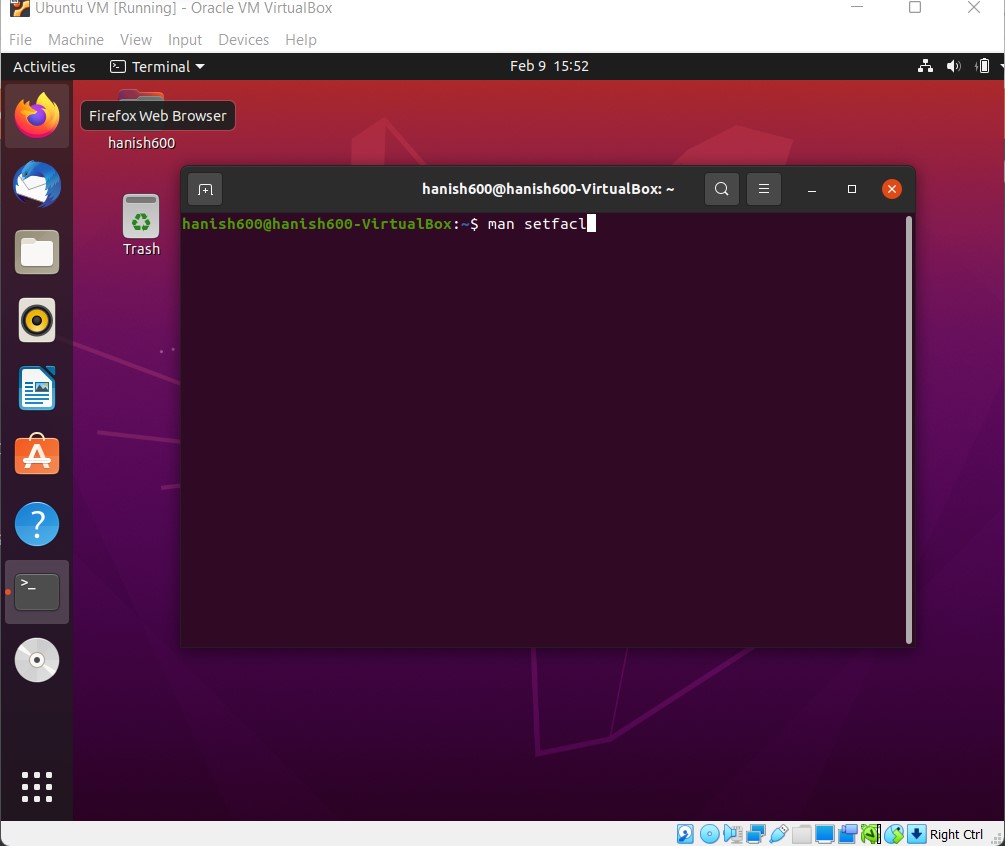
wc file\_name

This will be used to make a new directory

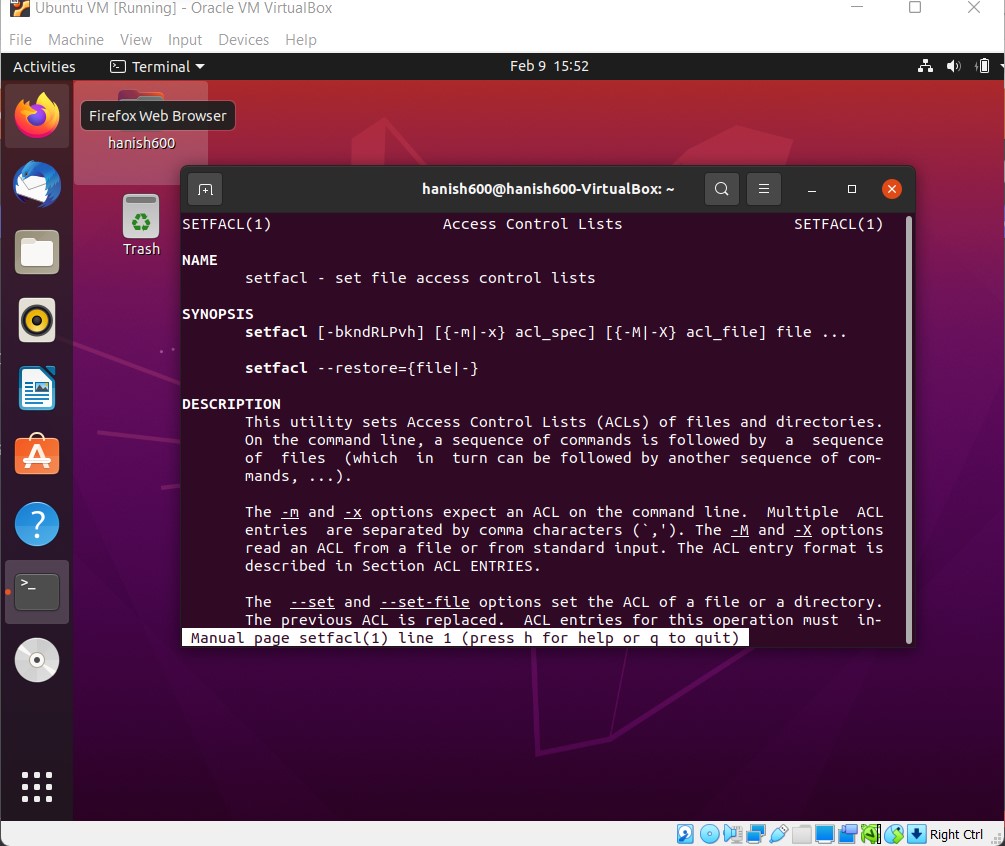
mkdir

**Summary of Results**

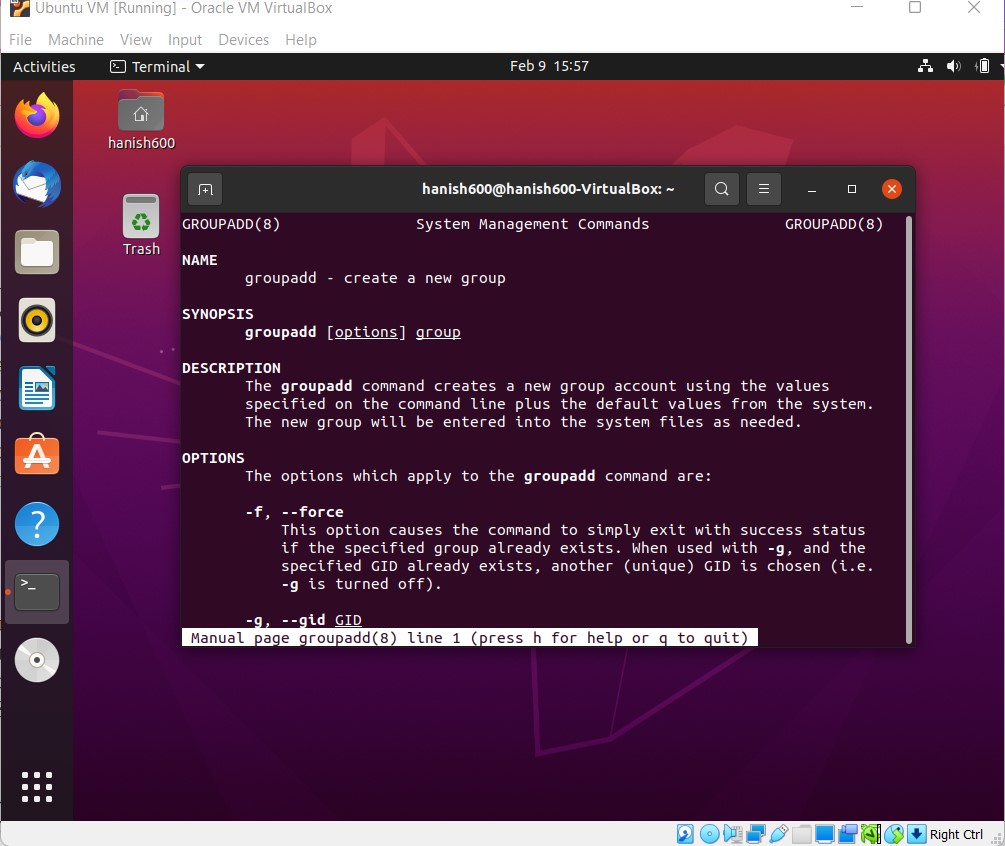
We begin with getting the manual for setfacl by using the man command.



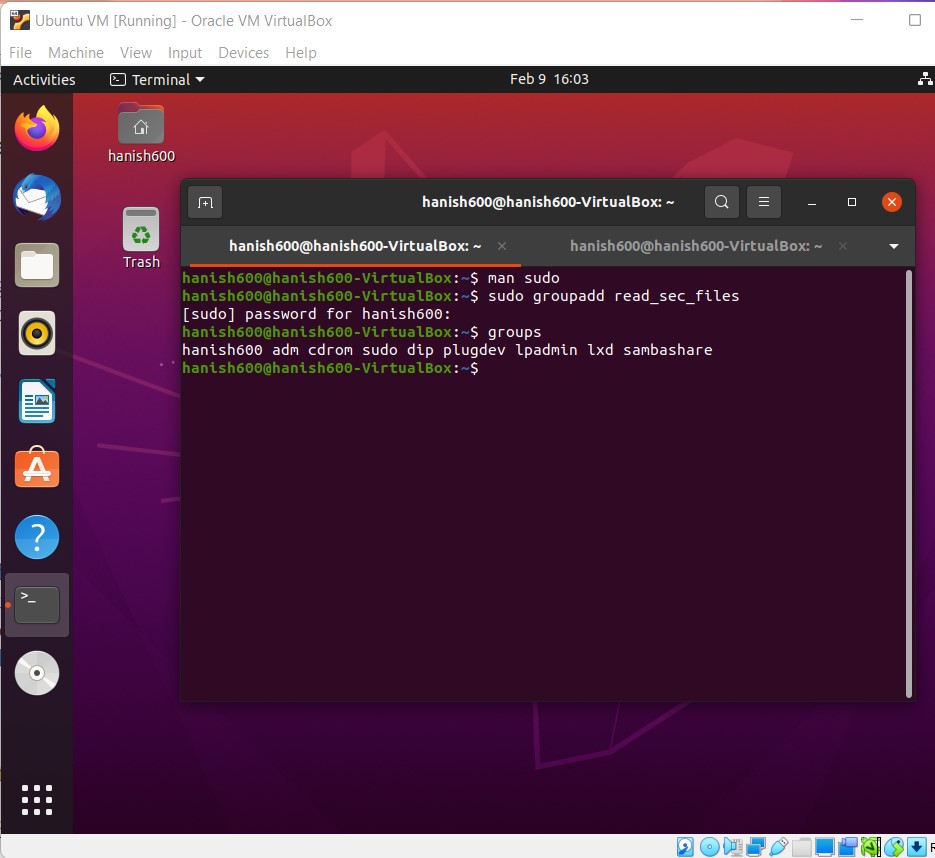
We are prompted here.



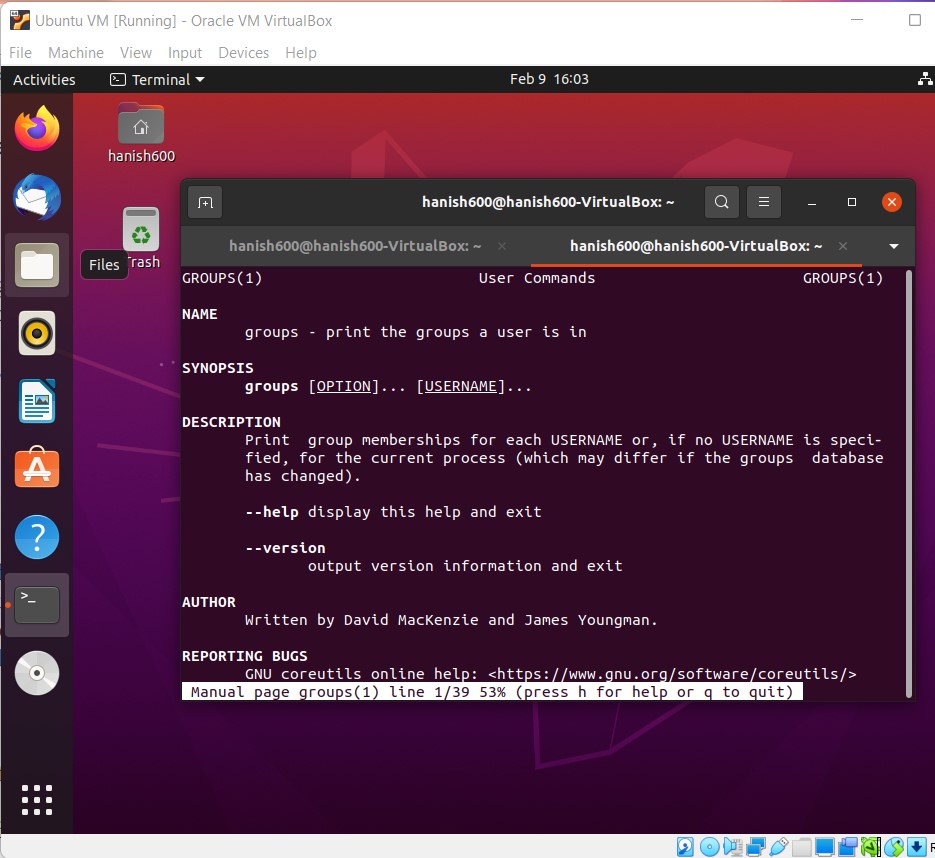
As we will see -m will modify controls and we will need to specify a group and file. Next, we take a look at the manual for adding groups.



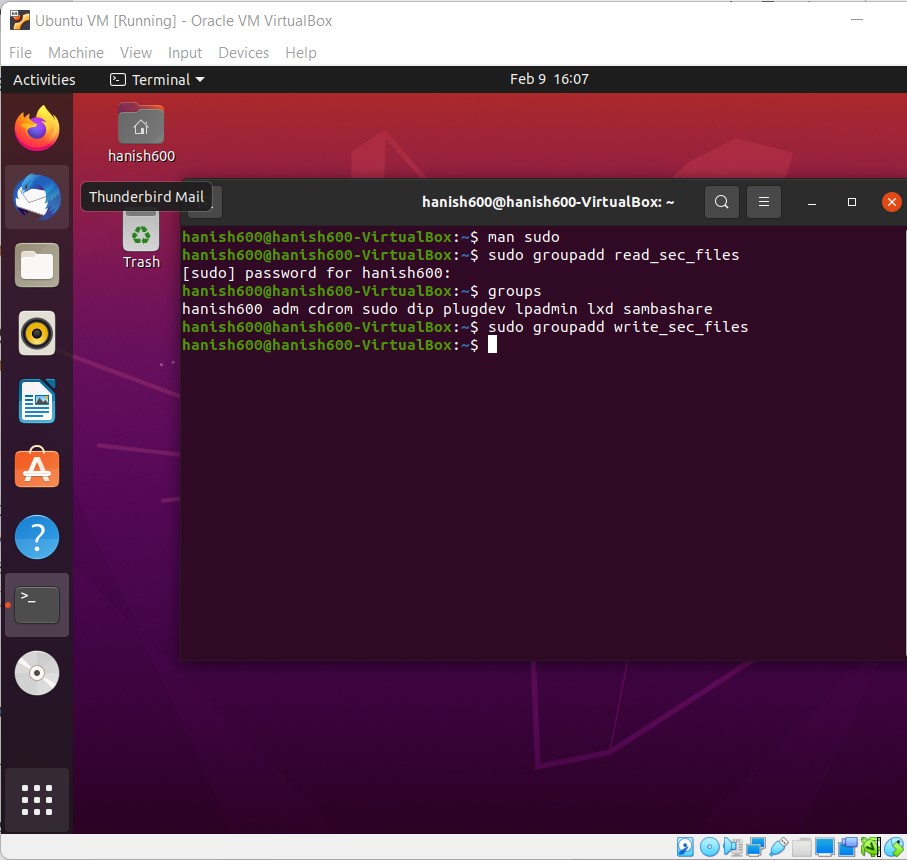
We can see that this will create a group for us. We can check the manual for some more additional information before we start. We will now try to create a group and show the groups with the groupadd command.



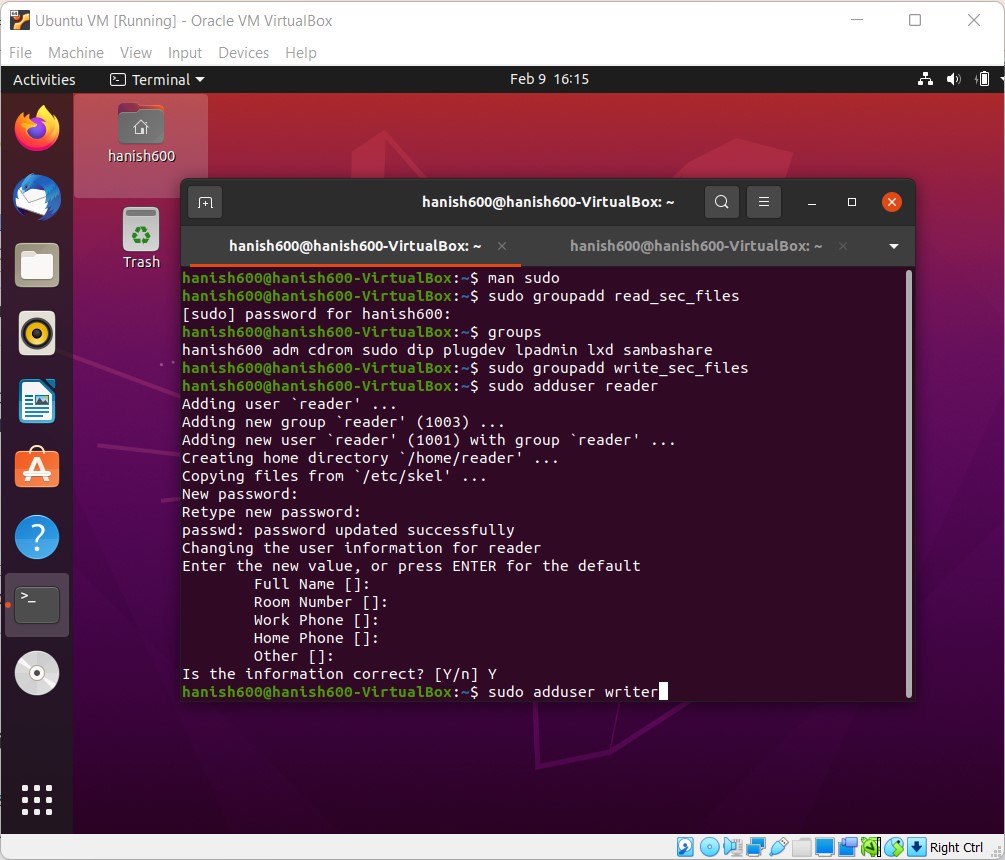
We cannot see the group when we type groups. To understand why read\_sec\_files isn’t in groups, we take a look at the manual again.

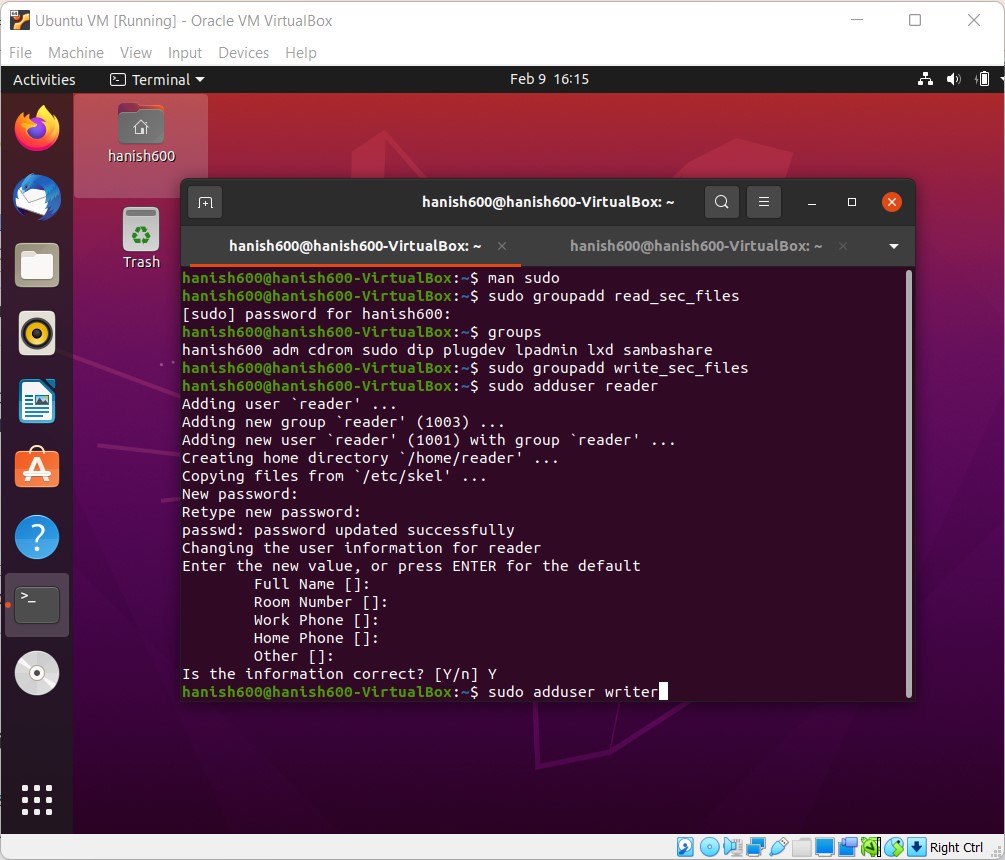


We see that it only prints the groups the user is in, so read\_sec\_files isn’t in the users group. Next, we add another group called write\_sec\_files.

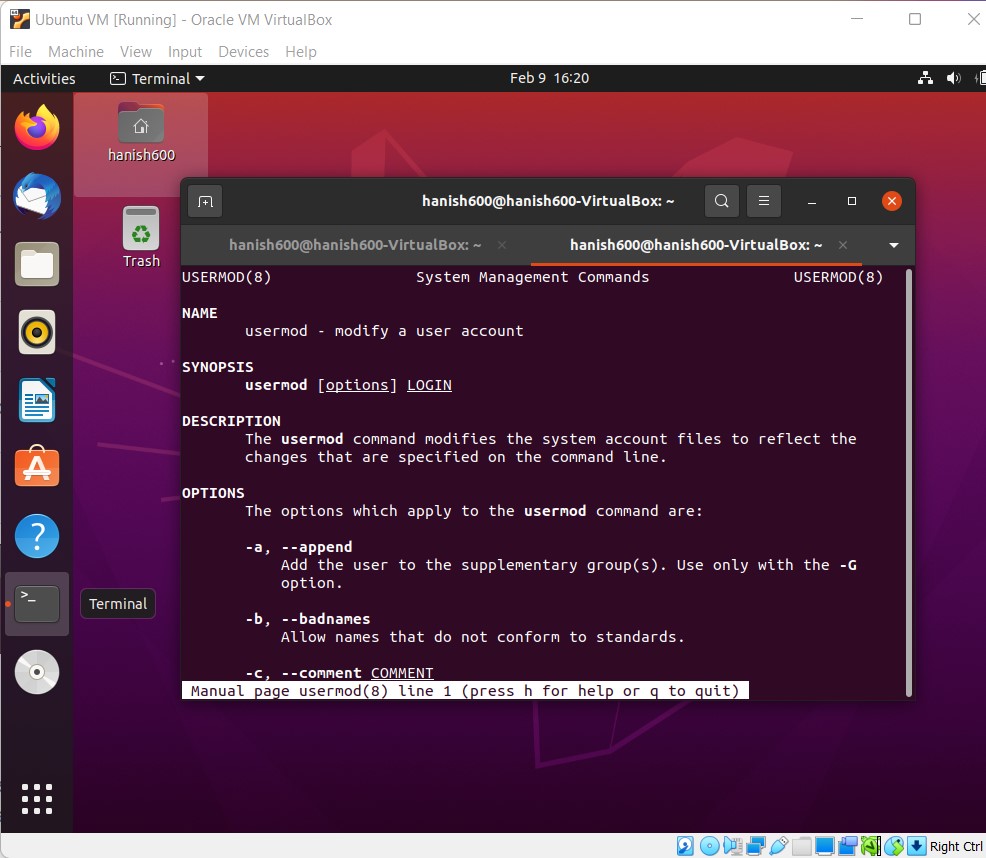


We now use the adduser command to add in two new users which we call reader and writer.

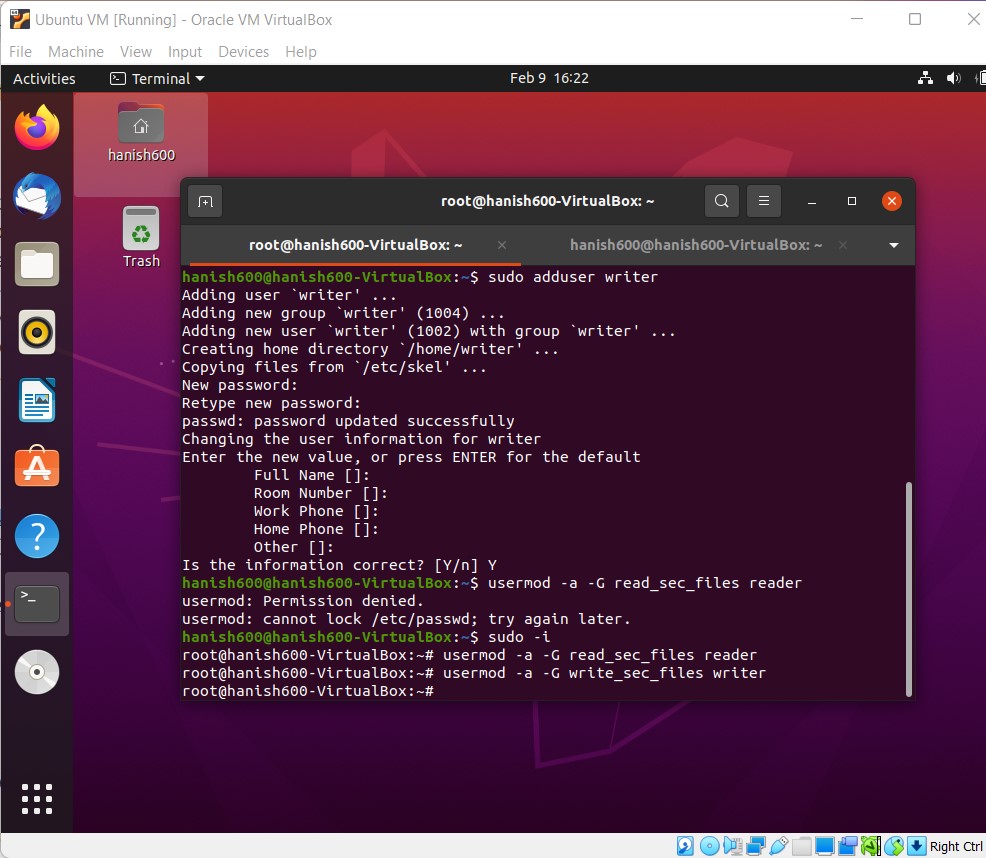




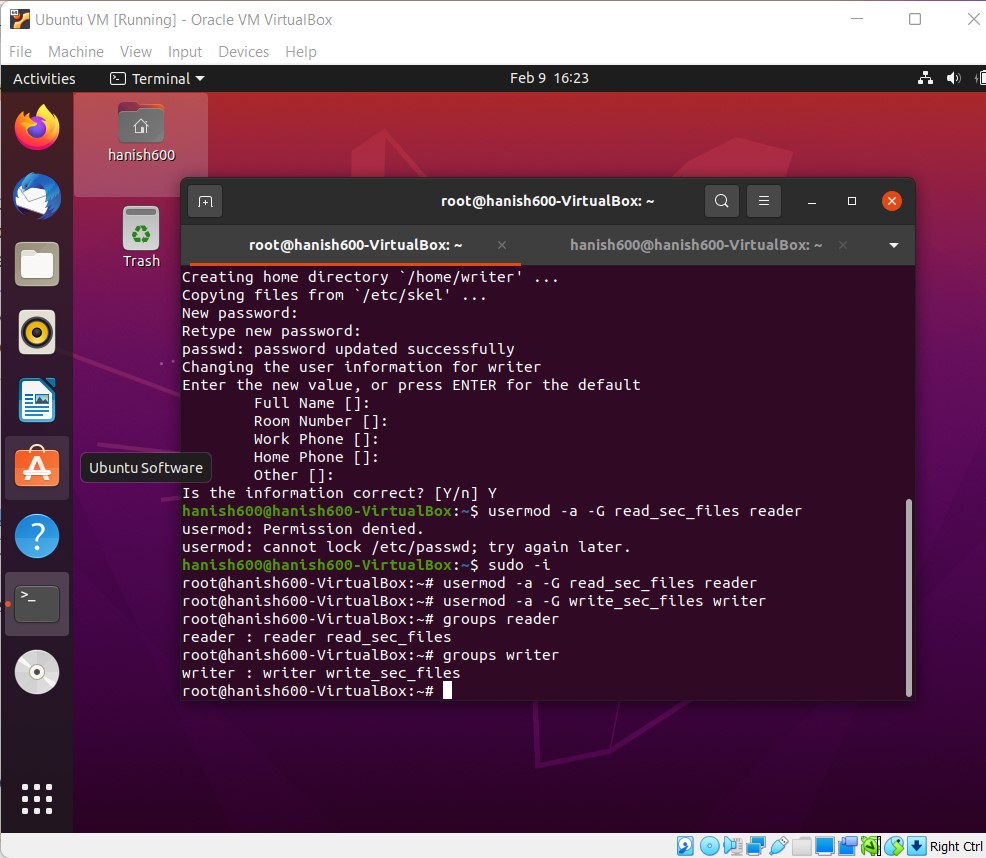
Now we want to add the users to some of the newly created groups. We check the manual for usermod.



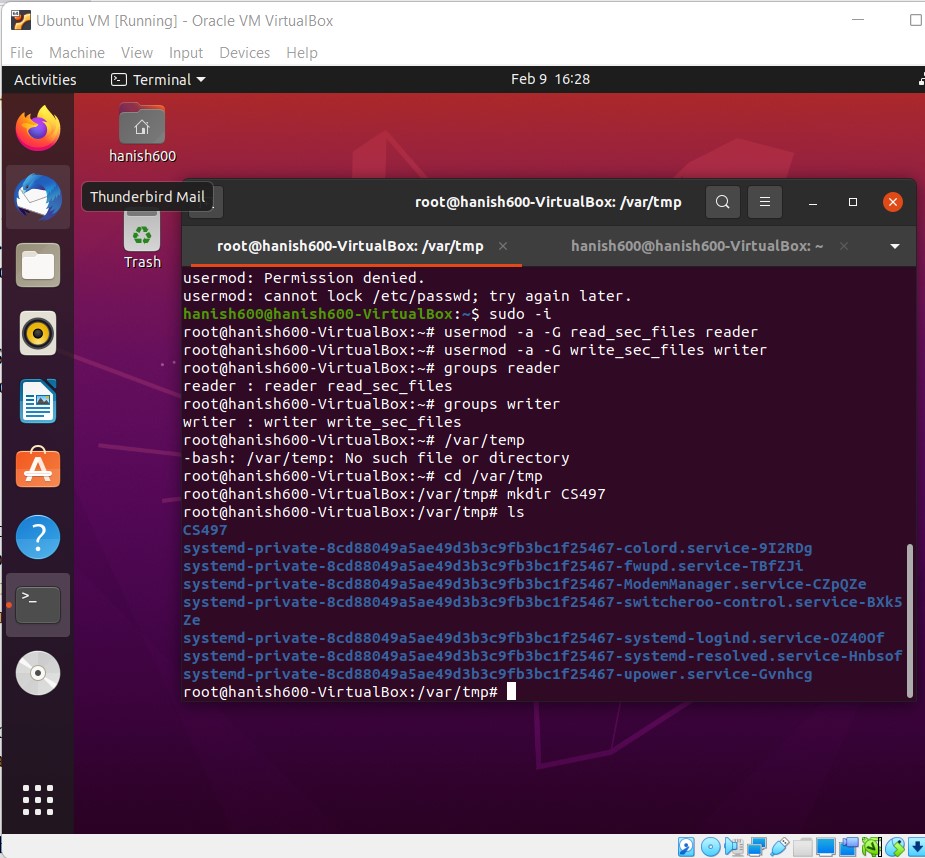
We can now use usermod -a to append the user to a group and -G for the group. We add in both the reader to the read\_sec\_files and the writer to write\_sec\_files. We notice we first must use sudo -I before we can change any control.



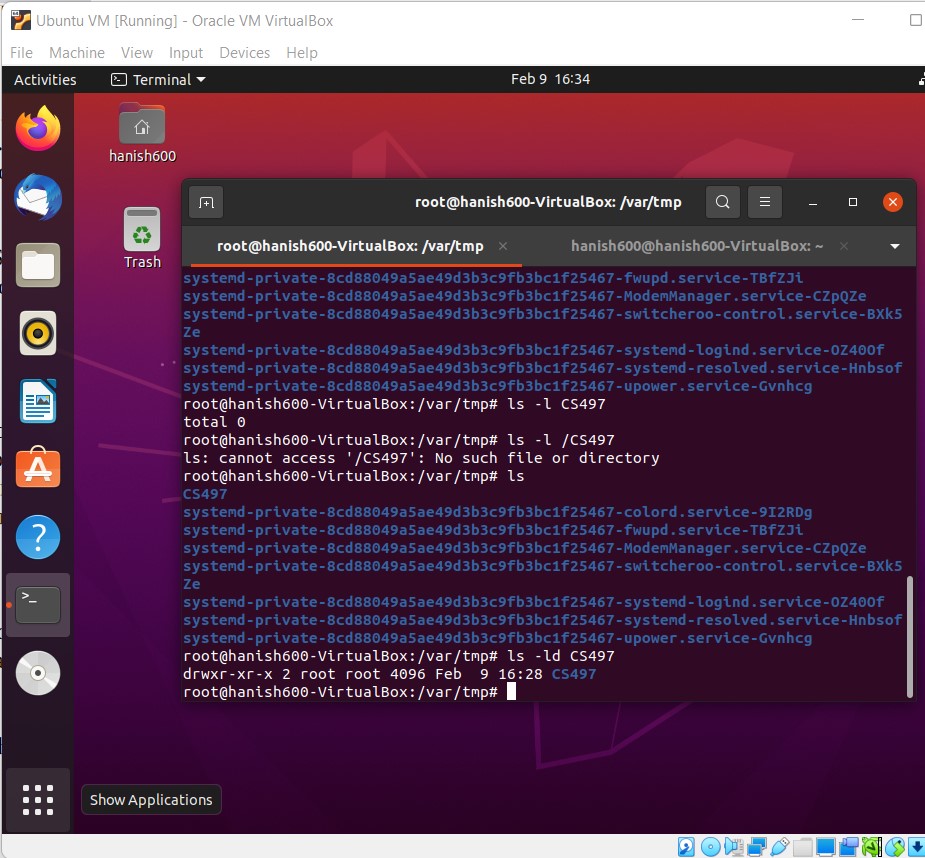
Now we are able to use our groups command to check since we just added the user to the groups.



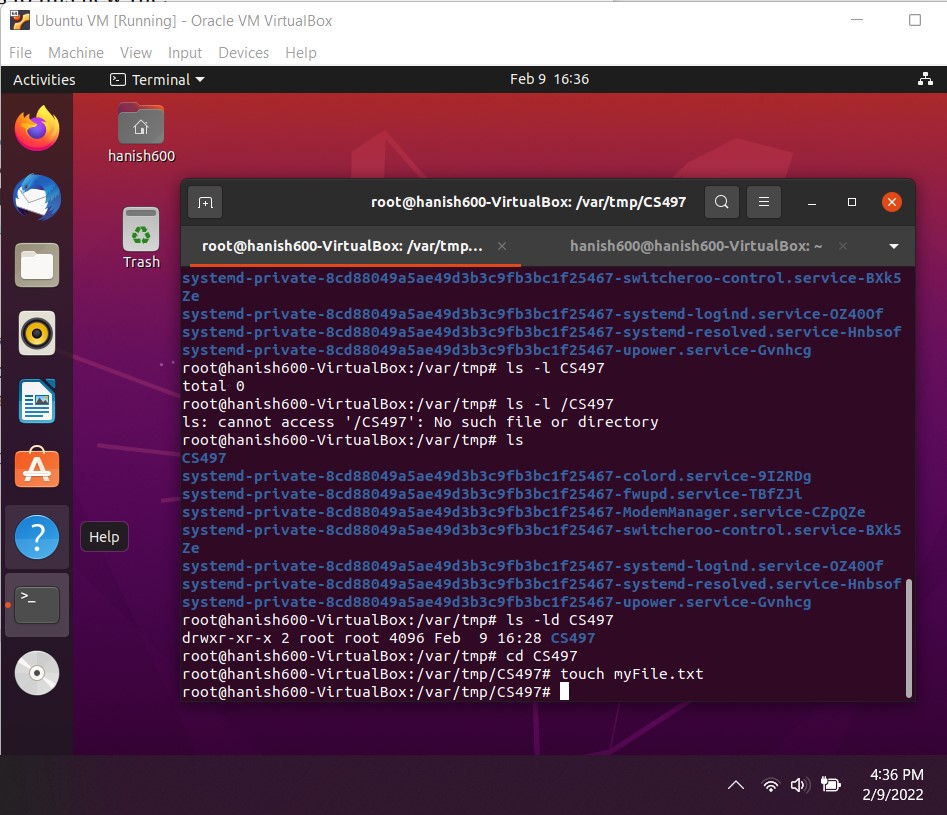
We then use cd to navigate to /var/tmp, and we use mkdir to make a directory with a name called CS497. We use ls to confirm our directory is inside the path.



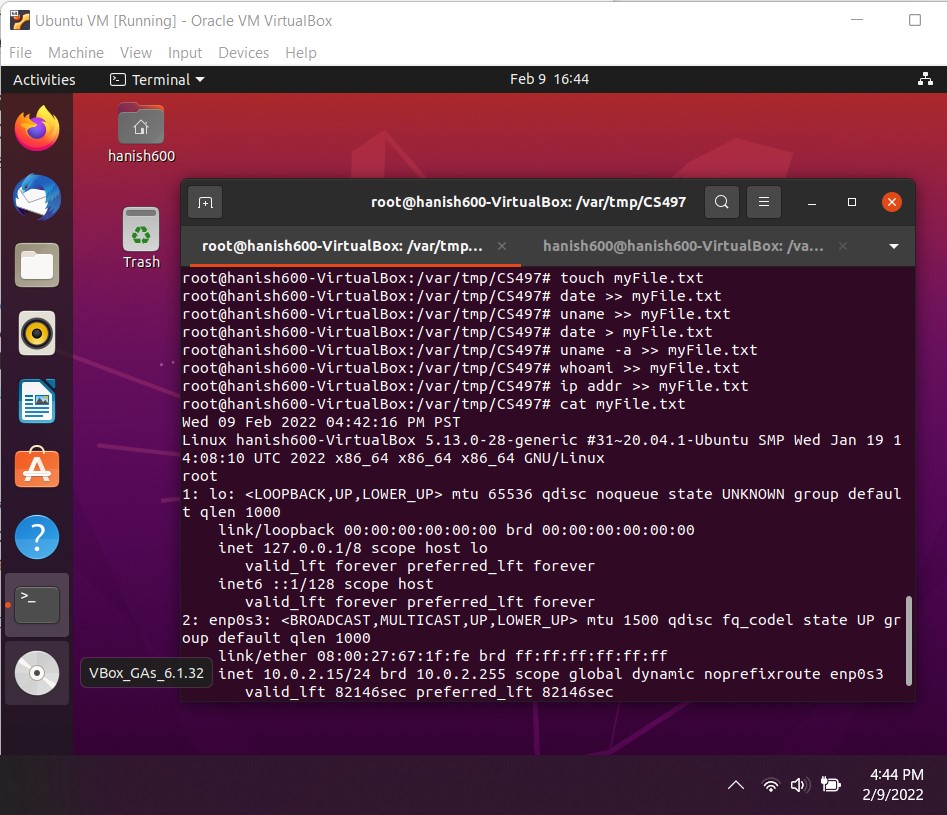
We use ls -ld to check our permissions.

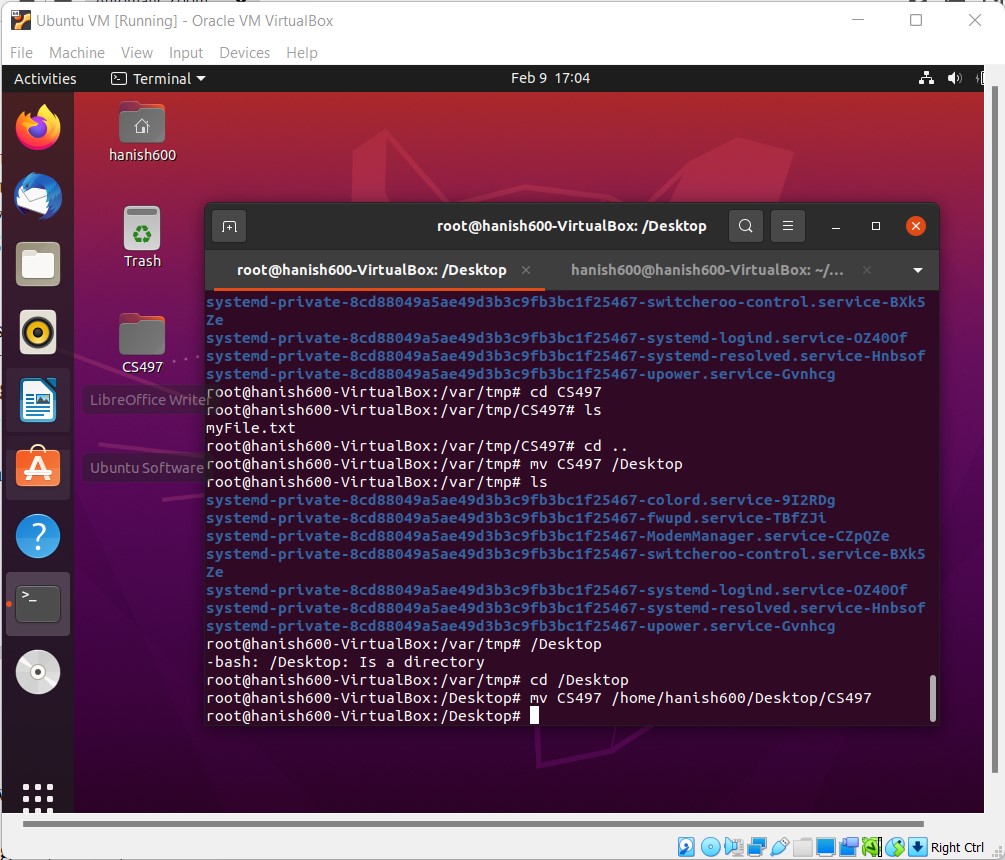


We navigate to our new directory folder and we create a new txt file using touch called myFile.txt.

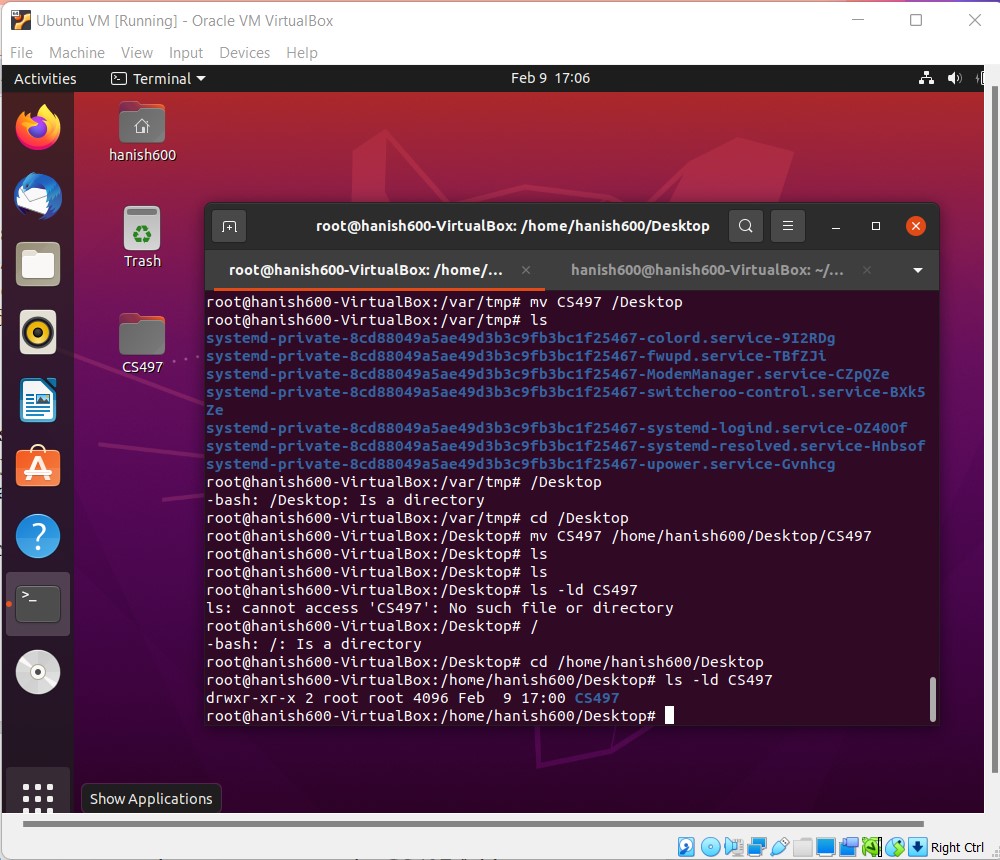


We then write some information into our txt file, then overwrite and append it after making a mistake. We do this by typing date,uname -a, whoami, and ip addr to get the current date, some information such as the kernel name and version, the user accessing it, and the ip address of the system.

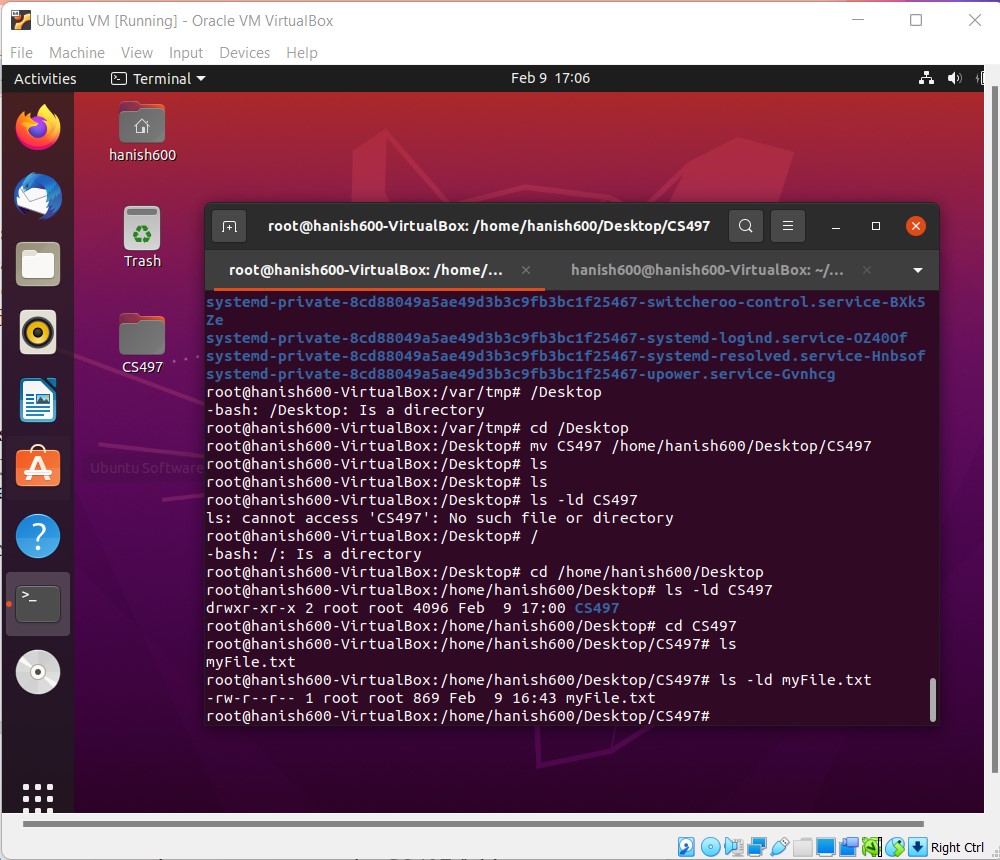
We check this information was created by using cat to open the file. We append it with >> and overwrite it with >.

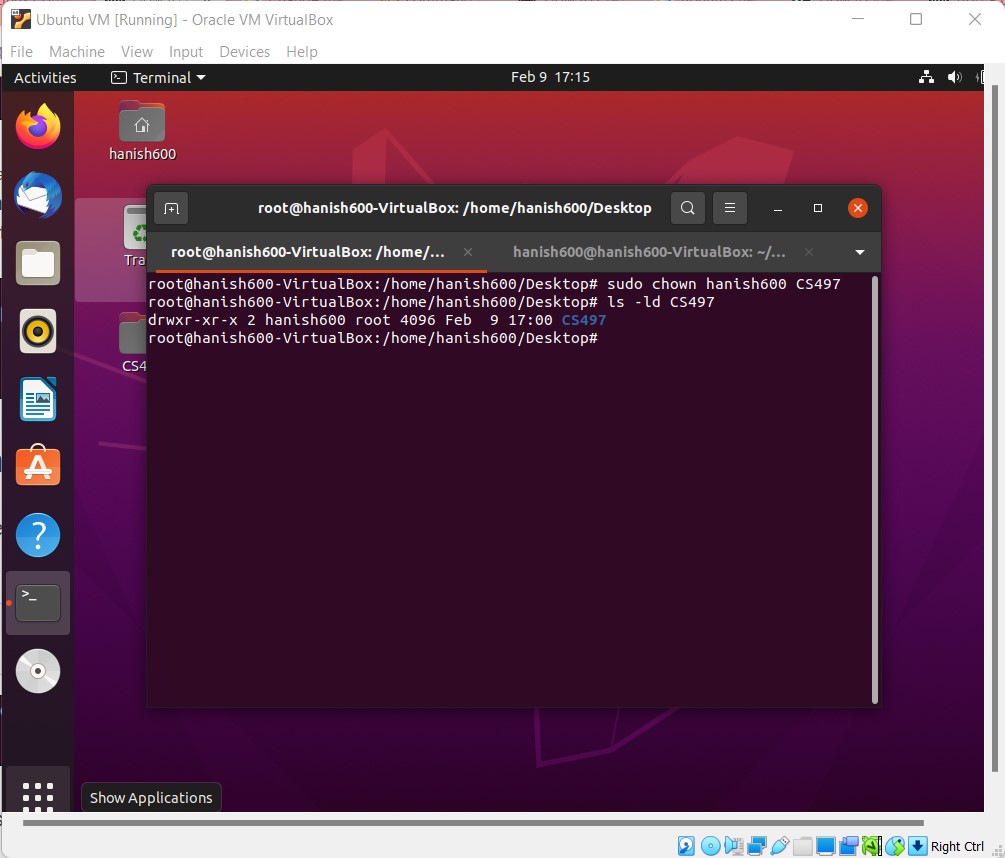
We move this file a few times with mv.

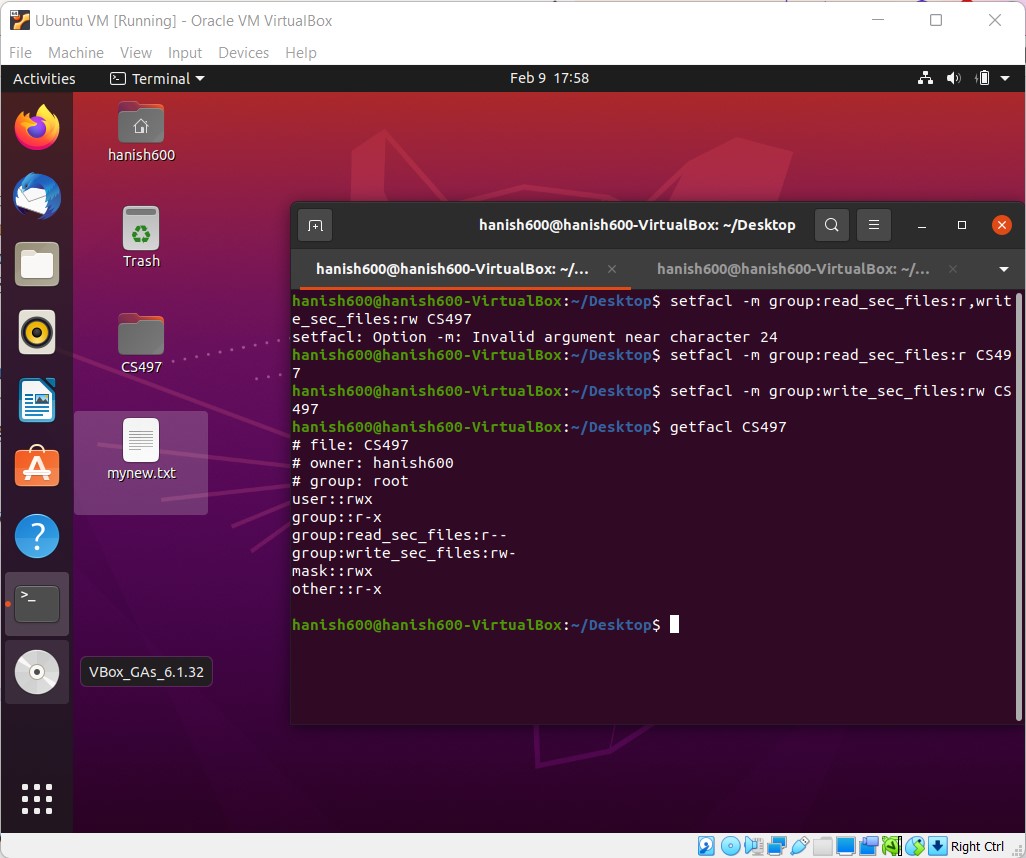
We do this until we get to our home users desktop. We then print the permissions again by using la -ld to list directories.



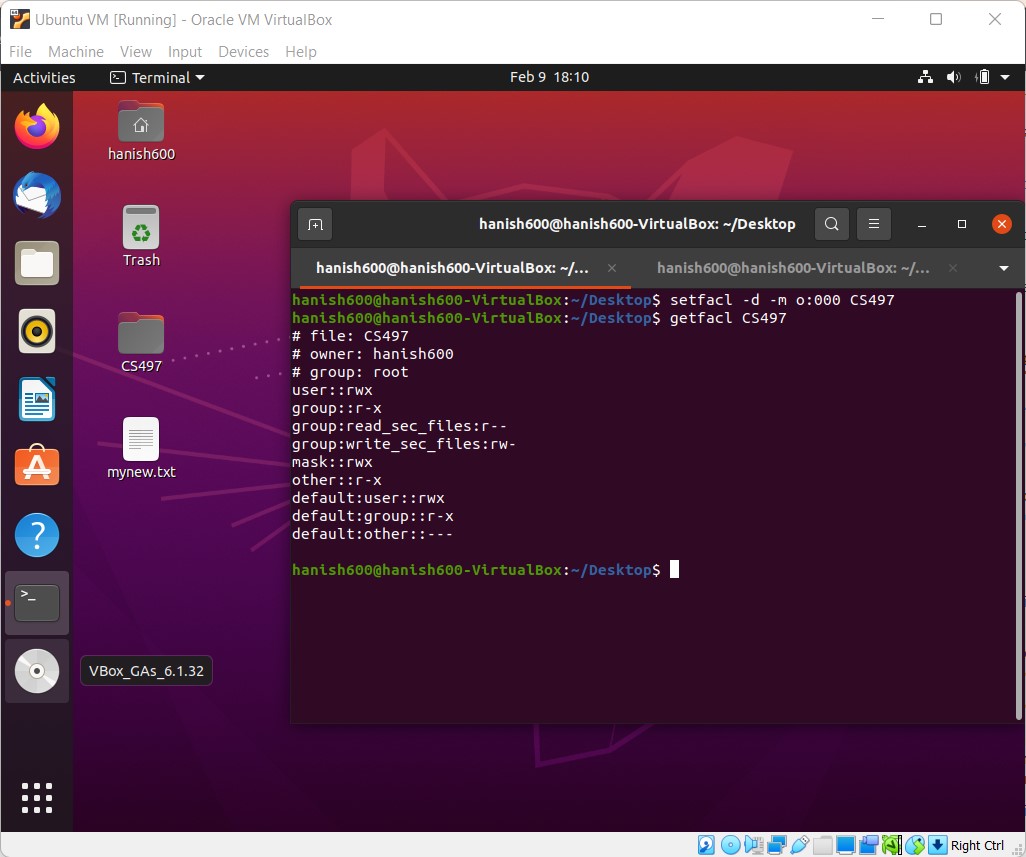
We also show the txt permissions with this command.

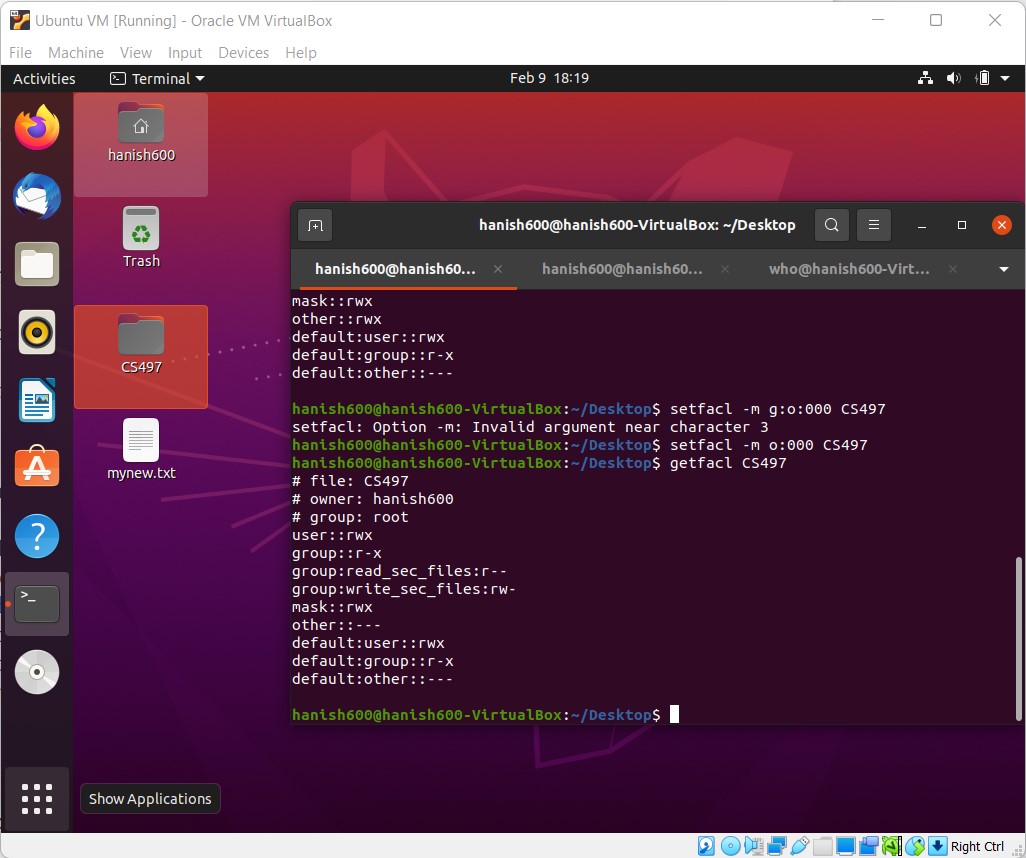


We use sudo with chown to get our user ownership of the folder. We then print this ownership. 

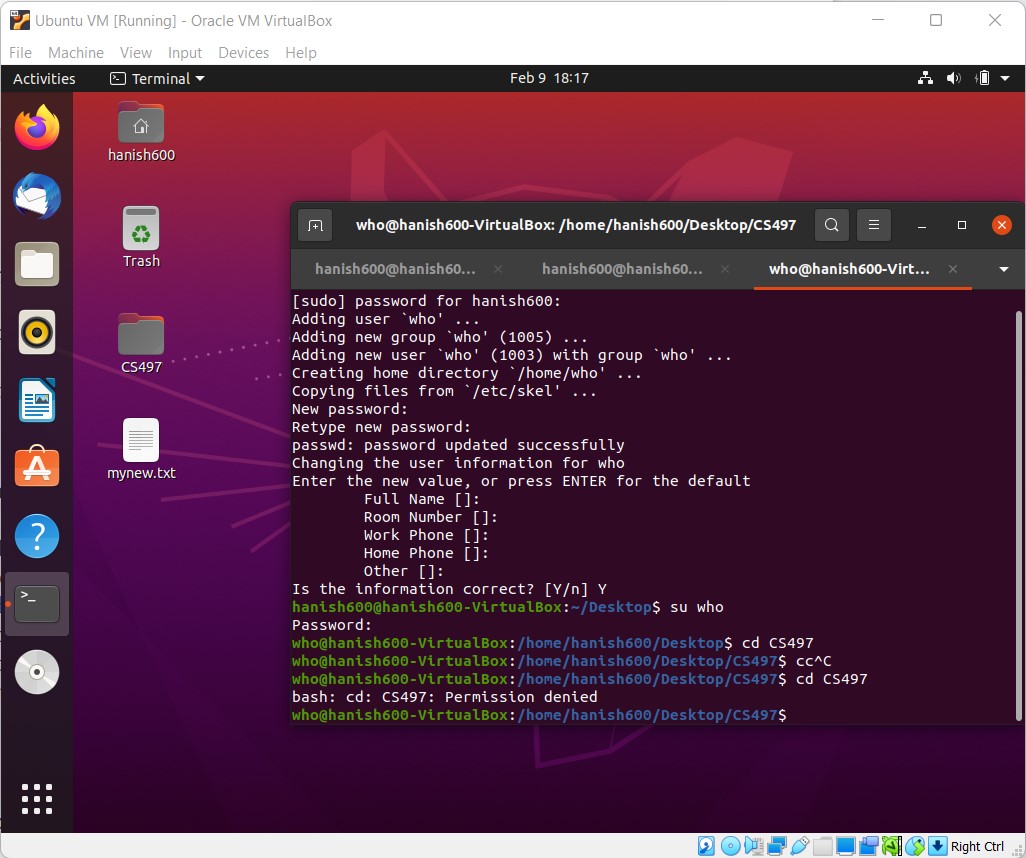
Now we start to set the permissions. We use setfacl with -m to modify, group along with our read and write groups, and our permissions followed by the folder name. We check the permissions with getfacl CS497. 

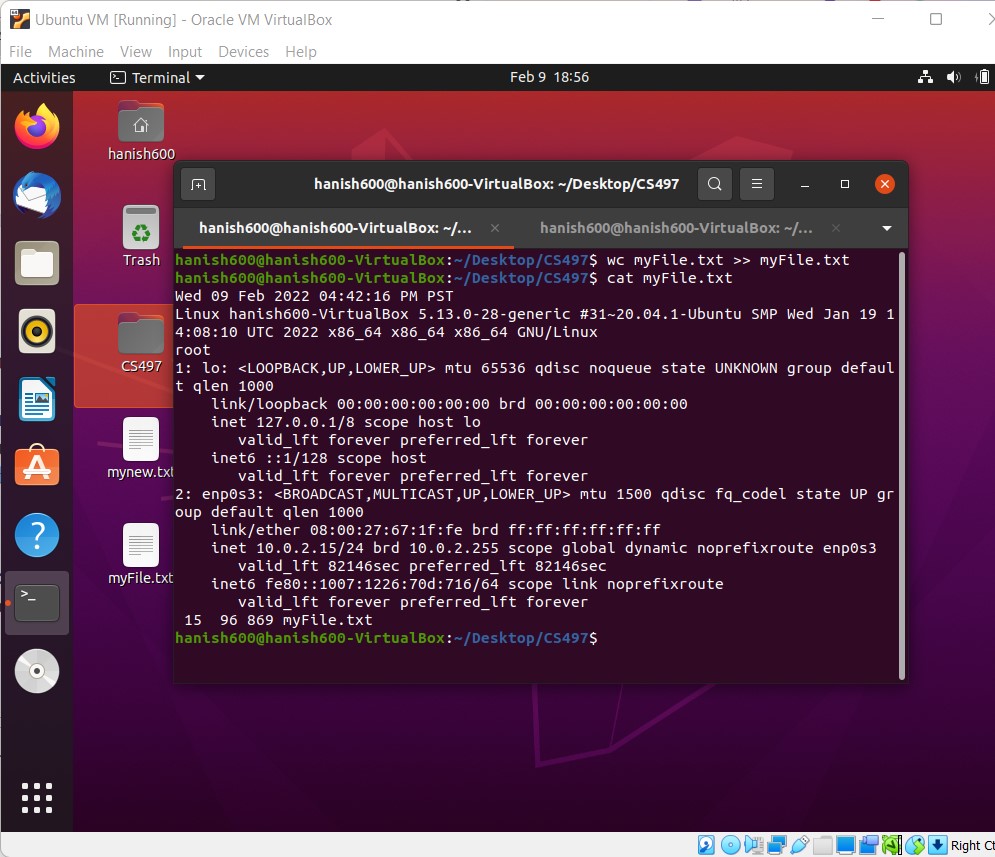
We also set a default other user for no permissions. We do not want the default so we also go back and change all others without default to no permissions.

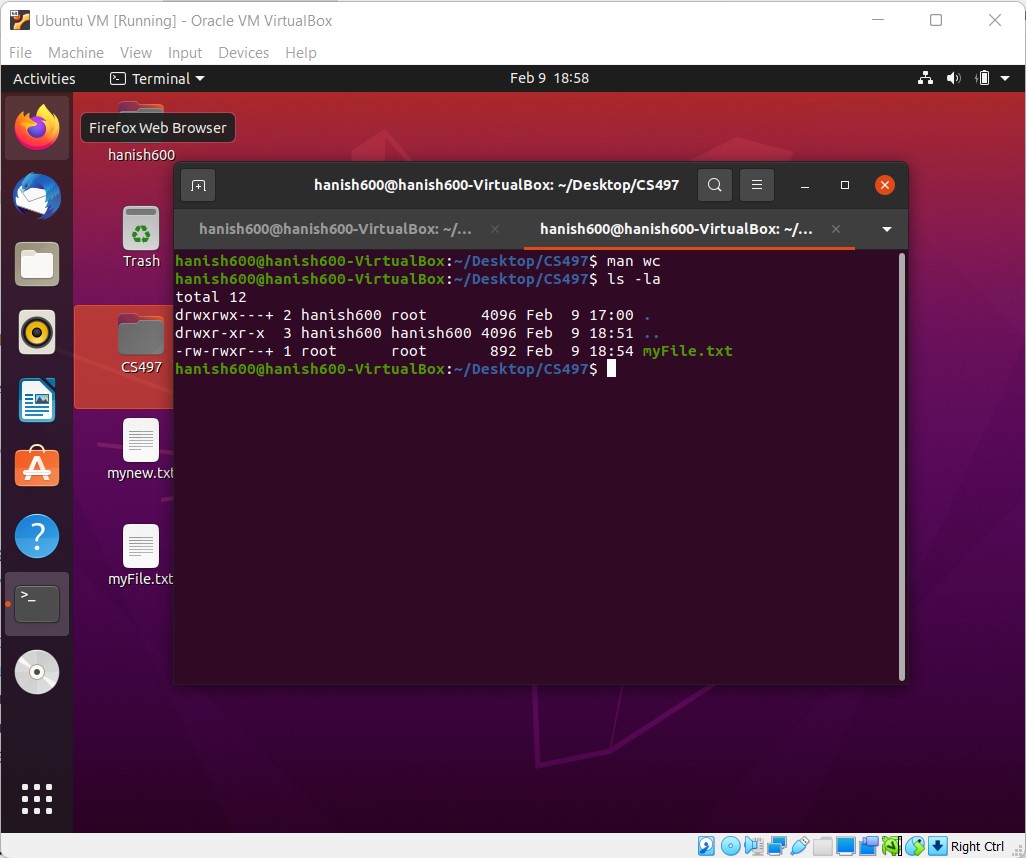




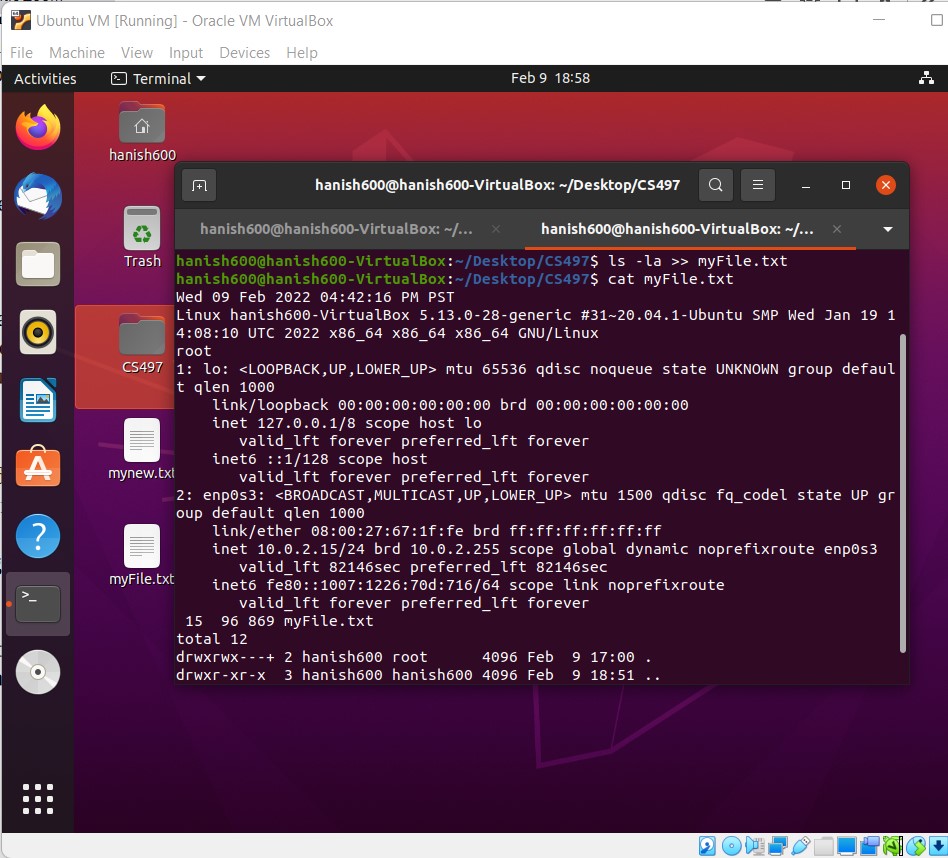
We create our third user “who”. We switch to su “who” and try to open the folder. The first time we had only the default others set to no permissions and “who” could access it. The second time we had all others to no permissions and who couldn’t access it.

There was some small mistake with the way we navigated to the folder, but regardless, who wouldn’t be able to access it.

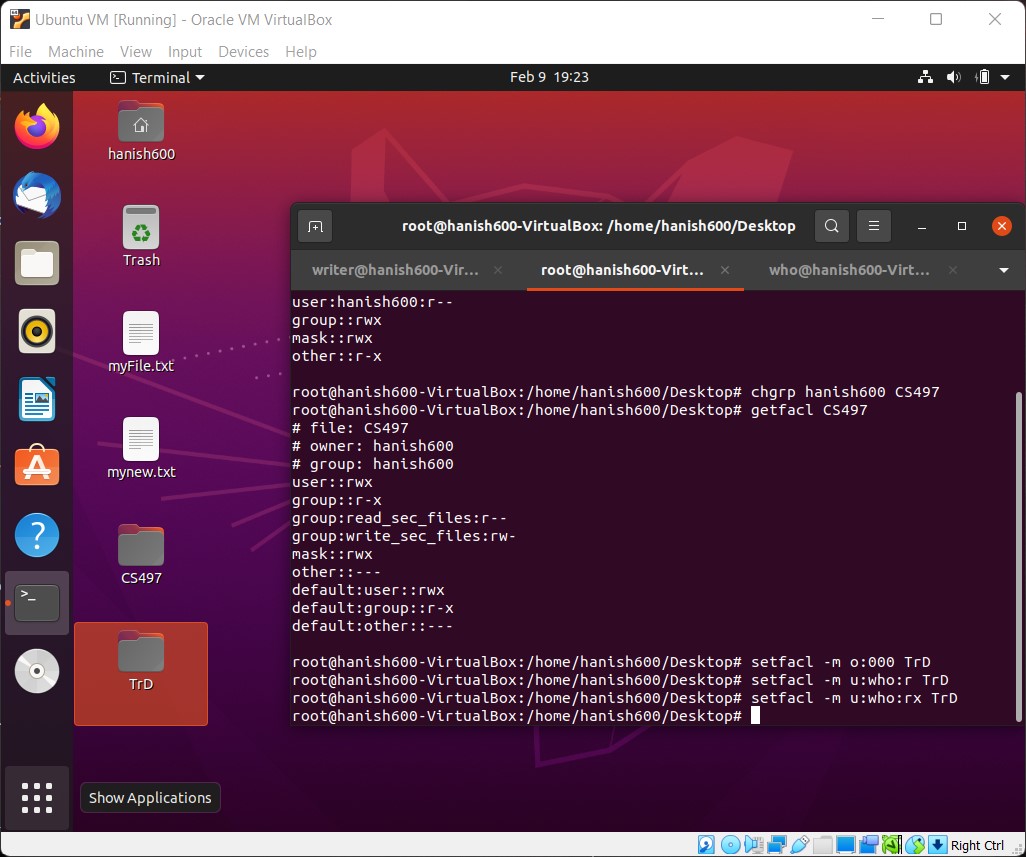
We started getting permission to write in this file from root using some previous commands. We started adding word count from our txt file and appended it to our txt file and check. 

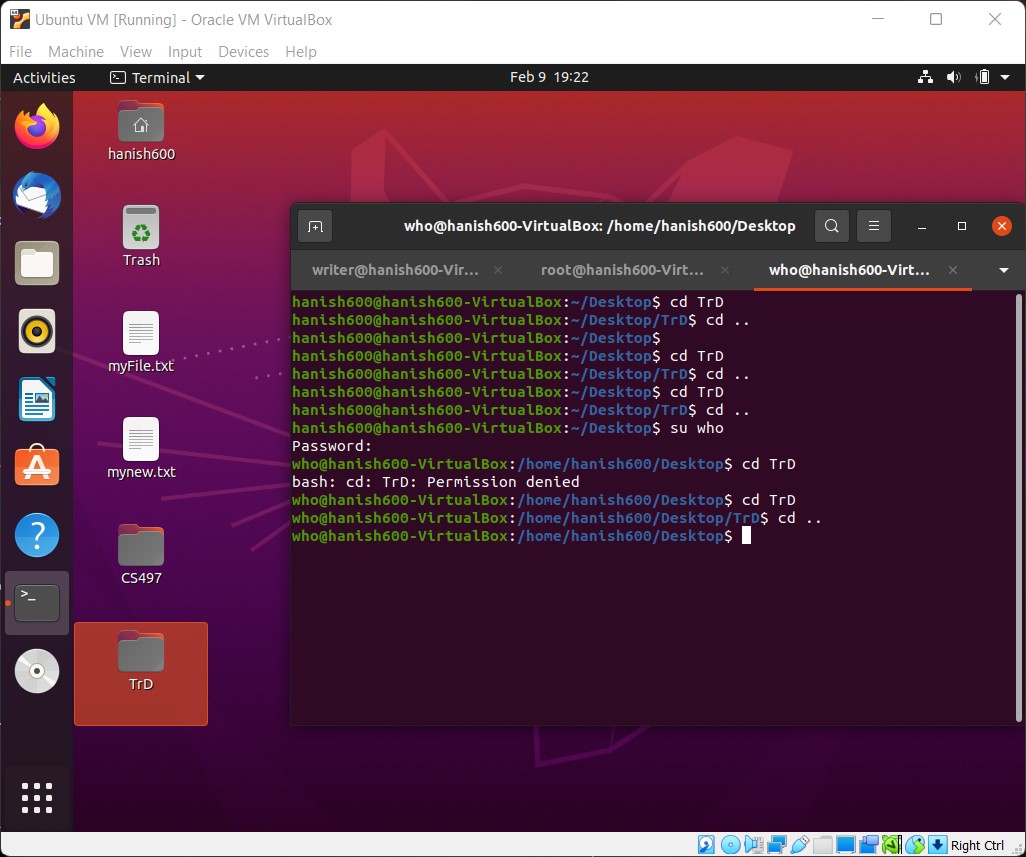
We now use our ls command again with la for listing all the files contained in the folder to check permissions. 

We check this in cat.



We change ownership and experiment with execution bit.





We find that only execution bits allow the file and folder to be accessed. Even when they are in read and write, execution bit is important.

**Conclusion**

Execution bits and read and write bits are the mechanisms to control permissions in Linux. When we say chmod 644 filename, we are saying the user can read and write (4+2 bits), the group and others can only read (4 bits). The first number represents the user and the bits for read and write. The second number represents the group and bits for read. The last number represents others and the bits for read. Other numbers would modify the permissions where 0 is none, 1 is execute, 2 is write, 3 is execute and write, 4 is read, 5 is read and execute, 6 is read and write, 7 is read write and execute. To make is readable only by one user, we can have the last two digits not be a 4,5,6, or 7. To make the files executable, we can have 1,3,5, or 7 bits. The owner permission is the permission of a single user that is also the creator of the folder or file. The group permission is every user that is added to the group. The owner permission consists of one individual, while the group permission consists of several people. This can be important if the user does not wish to share some files to the group, or if the group wishes to use those files and folders without the owner.