

**Batch: B-2 Roll No: 16010422234 Name: Chandana Galgali Date: 30/07/2024**

**Experiment No: 2**

**Aim:** Understanding Linux File system and executing basic commands in Linux

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resources needed:** Any Open Source OS/CoCalc Linux terminal online

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

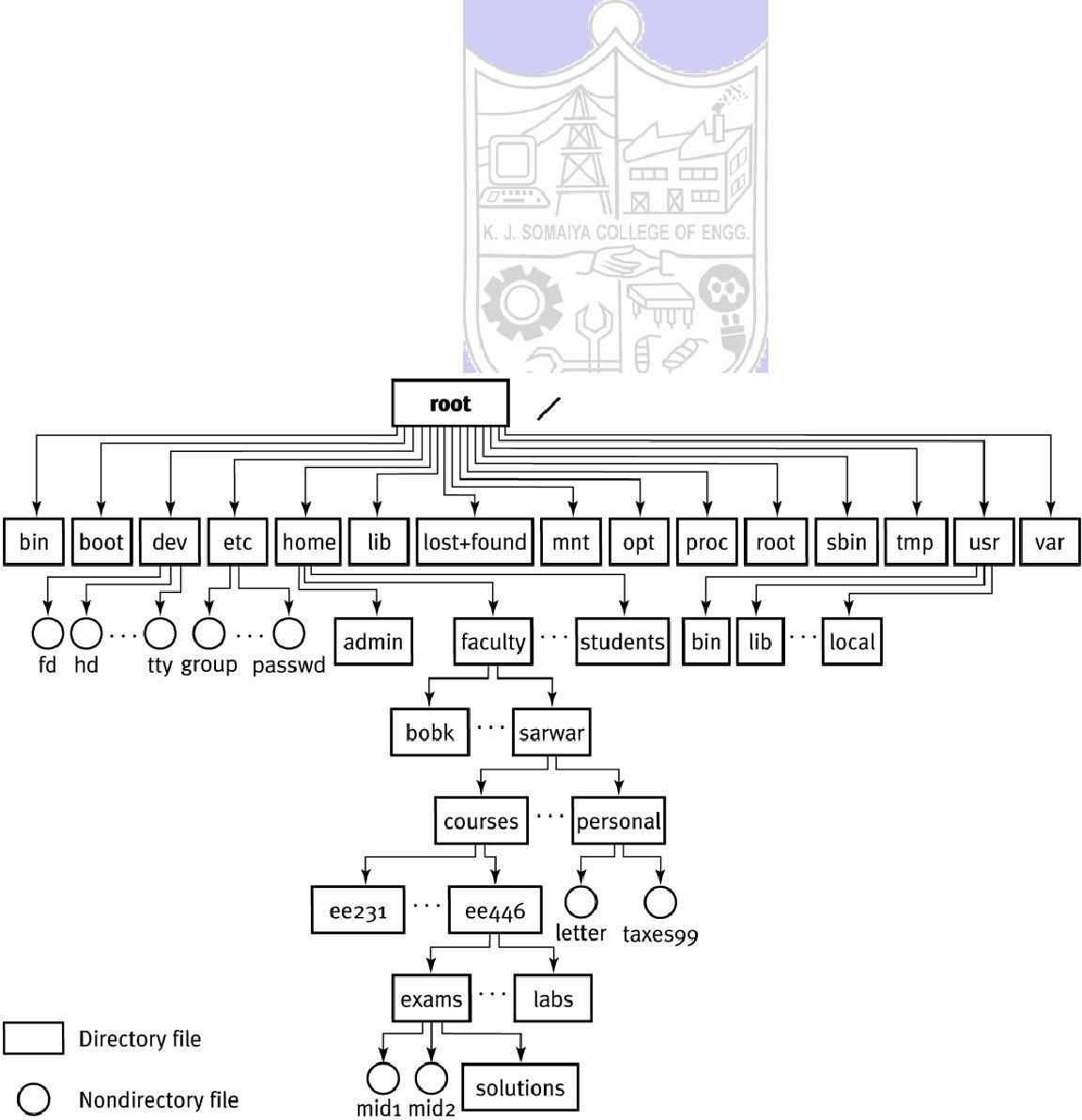
# Theory:

**Pre lab/Prior concepts:**

**File-system management:**

The file-system is the way an operating system manages all the files to be stored on the external storage - binaries, images, etc. Linux uses an advanced version of the Extended File-system from Unix, called ext2.

An important feature of the ext2 file-system is that it treats everything as files - directories are also represented as files containing pointers to other files. This does not stop only at directories, even the hardware can be addressed as files under the standard directory /dev.

One drawback in Linux has been the susceptibility of ext2 to damages due to power failures. But there has been significant development to rectify this shortcoming through the introduction of journaling file-systems like reiserfs, ext3. A journaling file-system has the inherent capability to be recovered to a stable state in case of a crash due to power failures.

# Directories

In Linux, different partitions need not be assigned special names to access them. This is because Linux uses a standard directory structure to take care of all partitions. This structure ensures that a particular file for a particular program will almost always be present at the same place on any machine running Linux.

The figure above shows the hierarchical organization of directories created on the file-system by default, and which are considered to be standard for all installations. Let us take a look at some of the more important ones.

4. /dev is the directory through which all the devices on the machine are accessible as files. These include the serial terminals (COM ports), modem, mouse, sound card

... everything!

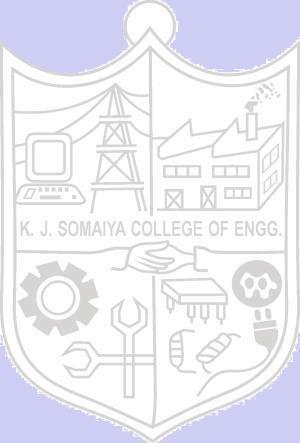
5. /var contains most of the "variable" data such as mails, log files, databases, etc.

6. /usr is where almost all the packages get installed. Discounting /mnt, it is the largest directory on the hard-disk. This directory itself has a pretty complicated sub-directory system used by the packages.

7. /etc contains all the configuration files, used by the operating system itself as well as various packages installed on the system.

8. /home contains the home directories for all the users created on the system.

9. /mnt is conventionally used as the base for all directories which are not part of the standard directory hierarchy. It is most commonly used to mount the cd-rom drives, floppy drives, non-Linux partitions, etc.

10. /root is the home directory of the privileged user or the system administrator, called "root".

In Linux there are different commands available to perform file operations like create, edit, copy, rename and move files.

We can assign read, write, and execute permissions to file. Every file in Linux has a list of permissions attached to it, that specifies the kind of access that different users have to that file. The file permissions are also called its access modes.

The very first column in the above output defines the access modes for the file. Their meaning will become apparent when we look at the way access modes are represented. These can be expressed as either octal numbers or mnemonics.

# Linux commands for file handling:

File handling commands include commands for creating files/directories, navigating through file system.

# Commands to create a file

**touch:** creates one or more empty file(s).

# syntax:

touch filename1 [filename2] [filename3]

**cat:** creates/displays file

# syntax:

cat [options] [filename]

🡪

cat > [filename]

🡪

cat [filename]

🡪

cat >> [filename]

To create a file

To view a file

To append text to a file

**mkdir :** Make directory (or) To create a directory

# Syntax :

mkdir [options] [dirname]

mkdir –p /home/abc creates directory with subdirectories

mkdir –m 777 ~/data

creates data directory in home and assigns read, write, execute

permissions to it

**rmdir:** Remove an empty directory. If you want to remove a directory with files in it type "rm -R directory"

# Syntax:

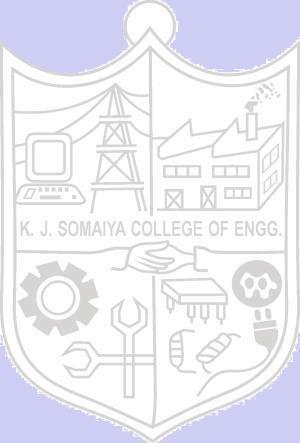
rmdir dirname

This will only remove the directory if it's empty otherwise it will exit with an error message.

**pwd:** Print working directory. Print the absolute (complete) path to the directory the user is currently in.

# Syntax:

pwd

This will tell you the full path to the directory you are in, for example it may output "/usr/local/bin" if you are currently in that directory.

**cd:** Change directory. Use " cd .." to go up one directory.

One dot '.' represents the current directory while two dots '..' represent the parent directory. " cd -" will return you to the previous directory (a bit like an "undo").You can also use cd absolute path or cd relative path

**ls :** List the Files and Directories within the current directory.

# Syntax:

ls [options] Options:

-l List file with permission.

-a List hidden file.

-i List files and inode number.

-R List directories , sub directories and their contents.

ls -options string This lists files using a certain string. The string can contain standard wildcards to list multiple files

You can use ls -d to show directories that match an exact string, or use standard wildcards.

**chmod:** change mode(permissions) for file/directory Chmod [rwxrwxrwx]/[421412421] filename

**rm:**Remove/delete a file(s) or directories(s). You can use standard wildcards with this command

# Syntax:

rm -options file\_or\_folder

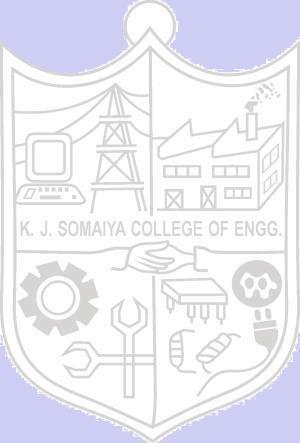
You can of course use standard wildcards to delete multiple files or multiple directories and files.

Use the -R or -r option to remove recursively, this removes everything within subdirectories. Also try the -f option to force removal (useful when you don't want to be prompted).

**mv:**Move a file or a directory to a new location or rename a file/directory.

Rename example:

mv filename1 filename2 Renames filename1 to filename2.

To move a file or directory, simply type: mv original\_file\_or\_folder new location

Note that this command can use standard wildcards to move files (not for renaming). Move and rename

Note that you can also move and rename a file in a single command. The difference is with the destination (right hand side) you change the filename to the new name of the file.

For example typing:

mv /etc/configuration.txt /home/joe/backupconfig

This would move the file "configuration.txt" to /home/joe/ and rename it "backupconfig"

**cp:**Copy a file. Has a number of useful options, such as -R (or -r ) which recursively copies directories and subdirectories.

**Syntax:**

cp -options file\_or\_files new\_location

Examples:

cp file1 file2

Simply copy file1 to file2 (in the same directory). cp /tmp/file1 ~/file2 /mnt/win\_c

**find:** The following examples illustrate typical uses of the command [find](http://linux.about.com/od/commands/l/blcmdl1_find.htm) for finding files on a computer.

find / -name game

Looks for a file named "game" starting at the root directory (searching all directories including mounted filesystems). The `-name' option makes the search case sensitive. You can use the `-iname' option to find something regardless of case.

find /home -user joe

Find every file under the directory /home owned by the user joe

There are other commands to locate files like locate, whereis, which, whatis etc.

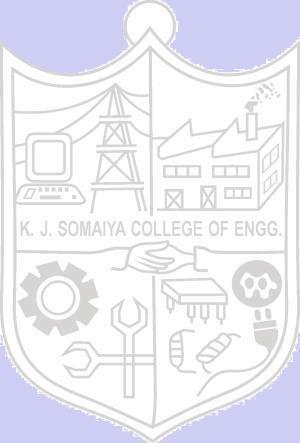
**User administration:**

Linux is a true multiuser environment. This means that the system can support different users with different privileges. Each user has access to a predefined set of system services and his/her own private data. The private data in turn can be shared with other users by granting access privileges to them. All the privilege checking is done with the help of user accounts maintained by the system. Linux provides a number of commands that can be used to create and manage user accounts. These are supported by a number of files and directories under /etc that are used to hold information about the users. The facilities provided can be broadly classified into two categories - user management and group management.

# User accounts

Traditionally, the information regarding the users is placed in a file called /etc/passwd. This contains the login name, full name, home directory and other info in a standard format. It may also contain the encrypted password used by the user, hence the name of the file. But nowadays, better ways for user authentication are used, which store the password elsewhere.

# Superuser Account

By default, every Linux installation has a specially privileged account called the root or superuser. This user has complete access to all the services and resources present on the system. The account is normally owned by the system administrator, and used to carry out special task that require special privileges not available to normal accounts. A person who logs in as root can modify any file on the entire system irrespective of the actual owner of the file and run any program anywhere on the system. As such, it is the most powerful account and has to be used with caution. Mistakes made while logged in as root can prove very dangerous to the system; hence its extremely important that the user should avoid using the root account unless absolutely necessary.

All other accounts are said to be non-privileged, since they have only access to a limited amount of services. Their privileges can be further controlled by use of [groups](http://db.glug-bom.org/Documentation/NGL/userinfo.html) which are used for collective management of user accounts.

# The concept of “groups” in Linux

In Linux, users are divided into logical collections called groups. These are used to confer various kinds of privileges to system objects to a group of users together. One user may belong to a number of groups, but he/she will always have a default group, along with other groups which are said to be supplementary. When a user is added to a particular group, all the privileges that are conferred on the group are also conferred on the user.

# File group

In order to implement access privileges, the first thing to do is define an owner and a group for the file. This information establishes who can claim control of the file. A file is given to an owner and a group as soon as it is created. Usually the owner is the current user and the group is the group of the directory within which the file is created; but this is system dependent.

# Linux commands for user administration:

**Adding new users**

The standard command useradd can be used to create a new user on the system. It is one of a family of commands for user management that can only be invoked by a user with special privileges, ie, the [root.](http://db.glug-bom.org/Documentation/NGL/userinfo.html) The most common arguments provided to the command are as follows:

# useradd :

To create a new user account and login directory for that new account. Useradd will create new entries in system files.

# Syntax

useradd [*options*] [*user*]

# Options

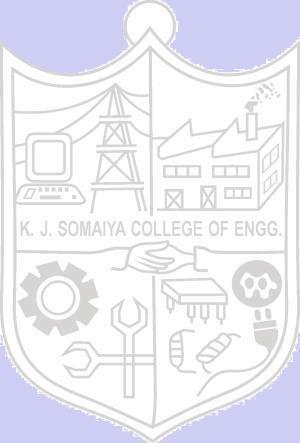
-c Comment field.

-d Home directory

-e Account expiration date.

To assign password to this user use following command passwd [password]

When a new user account is created, its entries update the following system files.

1. /etc/passwd
2. /etc/group
3. /etc/shadow

# Removing users

The command userdel is used to remove an existing user from a system. This can be invoked as follows:

userdel -r <username>

This command modifies all the system files, deleting all information about the specified username. The -r option will cause the user's home directory to be deleted along with any files or sub-directories it may contain.

# Modifying user information

The command usermod is used to modify information about an existing user. The arguments accepted by this are almost the same as those accepted by [useradd](http://db.glug-bom.org/Documentation/NGL/userinfo.html) command. The only difference is that it modifies existing entries rather than creating new one's.

usermod –G newgroupname username

# Managing groups

The superuser can use a set of commands analogous to the once used for user management.

New group is created by hand-editing the file /etc/group or by using groupadd command.

# Syntax:

groupadd groupname

groupdel command is used to remove the group.

# Syntax:

groupdel groupname

Groupmod is used to rename the existing groupname.

# Syntax:

groupmod –n newname oldname

These commands have functions similar to the corresponding one's used for user accounts. Again, every group is assigned a unique number called GID, ie, the Group ID.

**chgrp:** change group of file

# Syntax:

chgrp newgroupname filename

**chown:** change owner of file

# Syntax:

chown user:group filename

**Changing Ownership and Group**

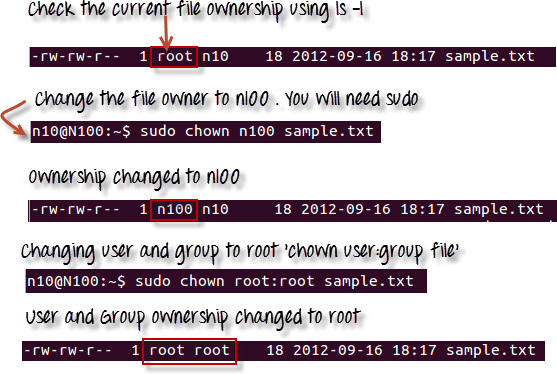
For changing the ownership of a file/directory, you can use the following command:

***chown user***

In case you want to change the user as well as group for a file or directory use the command

***chown user:group filename***

Let's see this in action



In case you want to change group-owner only, use the command

***chgrp group\_name filename***

'**chgrp'** stands for change group.

**Note:**

* The file /etc/group contains all the groups defined in the system
* You can use the command "groups" to find all the groups you are a member of
* You can use the command newgrp to work as a member a group other than your default group
* You cannot have 2 groups owning the same file.
* You do not have nested groups in Linux. One group cannot be sub-group of other
* x- eXecuting a directory means Being allowed to "enter" a dir and gain possible access to sub-dirs

**chage**: command let you specify an expiration limit for a user's account and password .

# Syntax:

chage [option] username Option

-l lists the current password expiration.

-m set the min. days to change the password. - M set the max. days to change the password. - E specific expiration date for user account.

-I set inactive period (in days)

-W warning period, number of days before expiration.

**Process management:**

Process is running an instance of a program. Process management starts with the init command which is present in a file named /etc/initta. Init is the first process that runs on the system and can start other processes depending upon the run level. Every process has a process identification (PID) number. PID of init is 1.

# Different types of process:

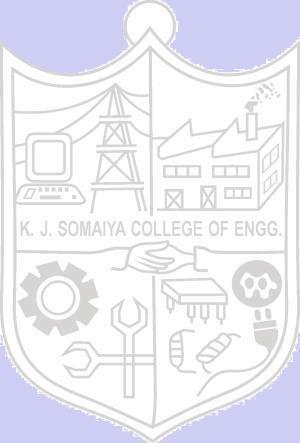
Daemon process User process

To display currently running processes on linux systems the following **ps** command is used. Output of the ps command has following fields

Column HeaderContents

%CPU How much of the CPU the process is using

%MEM How much memory the process is using

ADDR Memory address of the process

C or CP CPU usage and scheduling information

NI nice value

PID Process ID number

PPID ID number of the process's parent process

PRI Priority of the process

RSS Real memory usage

S or STAT Process status code

START or STIME Time when the process started SZ Virtual memory usage

TIME Total CPU usage

TT or TTY Terminal associated with the process

UID or USER Username of the process's owner

WCHAN Memory address of the event the process is waiting for

To kill a process using PID command **kill** is used. And to kill process by name **killall**

command is used Job management:

Job management uses the concept of foreground and background processes. Foreground processes always have access to the standard input stream to receive commands from the user and the standard output stream to print information to the user. Where the programs which run in the background do not have access to these streams. So they can’t interact with the user .

Compiling and building software related programs can run in the background. Through command we can make any program executing in background to run in foreground and vice versa.

# Linux commands for Process Management:

**ps:** used to list the process.To display information about processes specific to the active terminal.

# Syntax:

ps [option] Options available:

-a : To display all process, excluding process not controlled by a terminal

-x : To include all processes not controlled by a terminal, such as the daemon process.

-l : To display a long list including more information such as the process owner's user id.

-u : To display the user name of the process owner.

-e : To view all the process ids

**kill:** used to send a particular process to get clear.

# Syntax :

kill -9 <process id>

ps - aux is the command to view all process name and id

Example: kill -9 3743 (This example shows the kill command, option and the process id to kill.)

**Renice:** used to set the priority of a process.The priority value can range from -20 (highest priority) to 19 (Lowest priority)

# Syntax :

renice - +(minus symbol or plus symbol) <priority number> -p <process id>

Example : renice -15 -p 1970



**jobs** command helps you to view the background processes.

**Fg** command is used to run the background process in the foreground (Front End) jobs <enter> to view the process running and job numbers

fg %2 (job number)

**bg** command is used to resume the suspended command in the back ground.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Activities:**

1. Try out different file management commands. Browse manual pages of the commands and try-out different options of the commands

2. User Administration

Create a two users and assign passwords

Check for creation of the login directory of the user account.

See and note down userid, groupid of this user from /etc/passwd file and group form /etc/group file

Delete any one of the two users created above Create a new group1 and add the user to this group. Create three users and add them to a new group2.

Create a directory and use chgrp to change group of this directory to group2

Now login using first user of group 1 and try accessing this directory of group2 note down output

Check group and owner of the file and then change both group and owner of that file using chown

Explore chage command

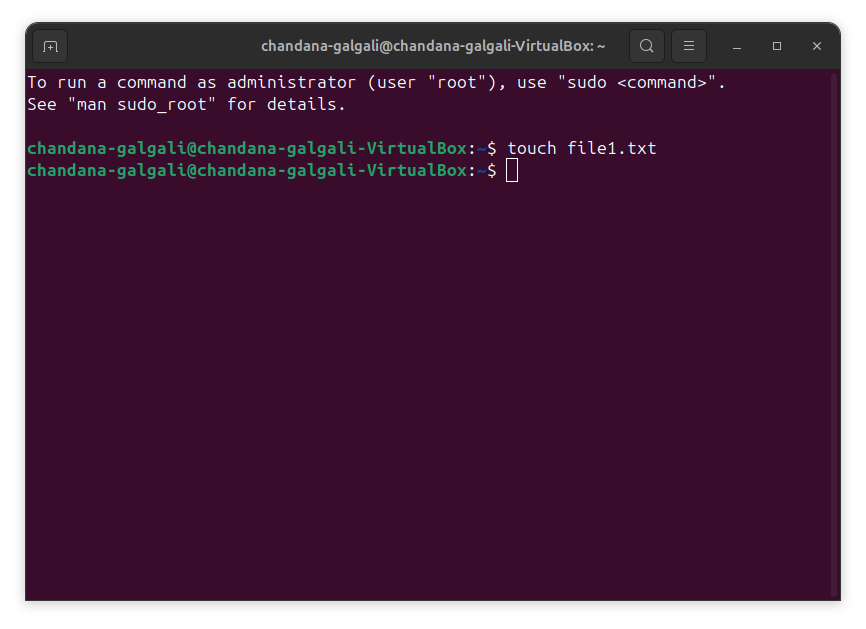
3. Process management

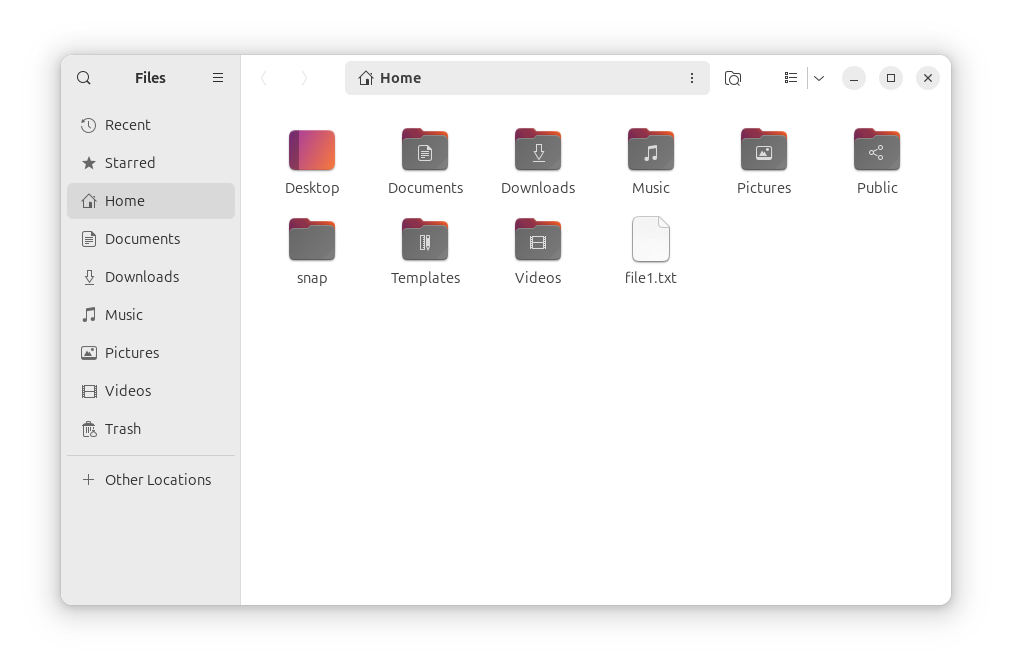
* explore ps command
* list all jobs
* start any new process in background
* start its execution in foreground
* suspend execution of this process
* resume its execution in background

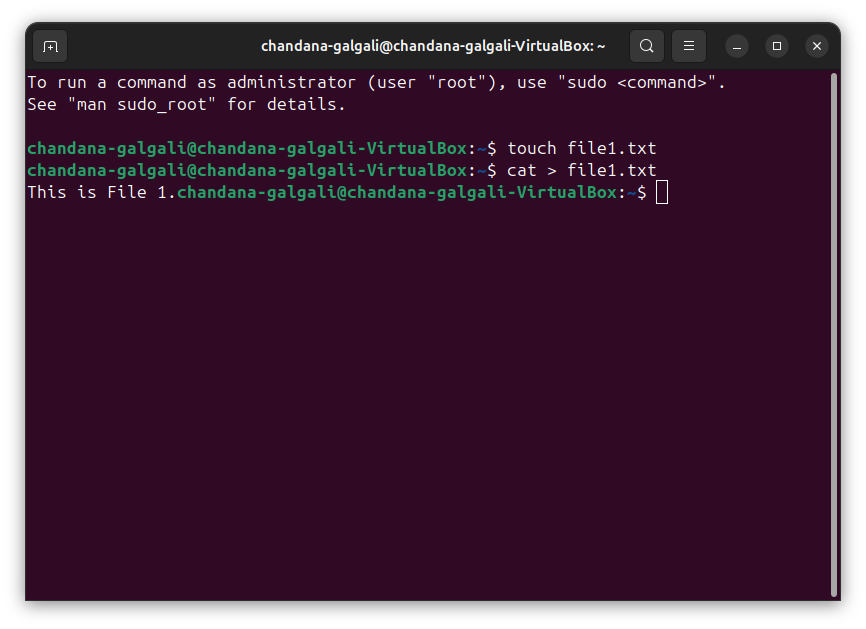
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Results:**

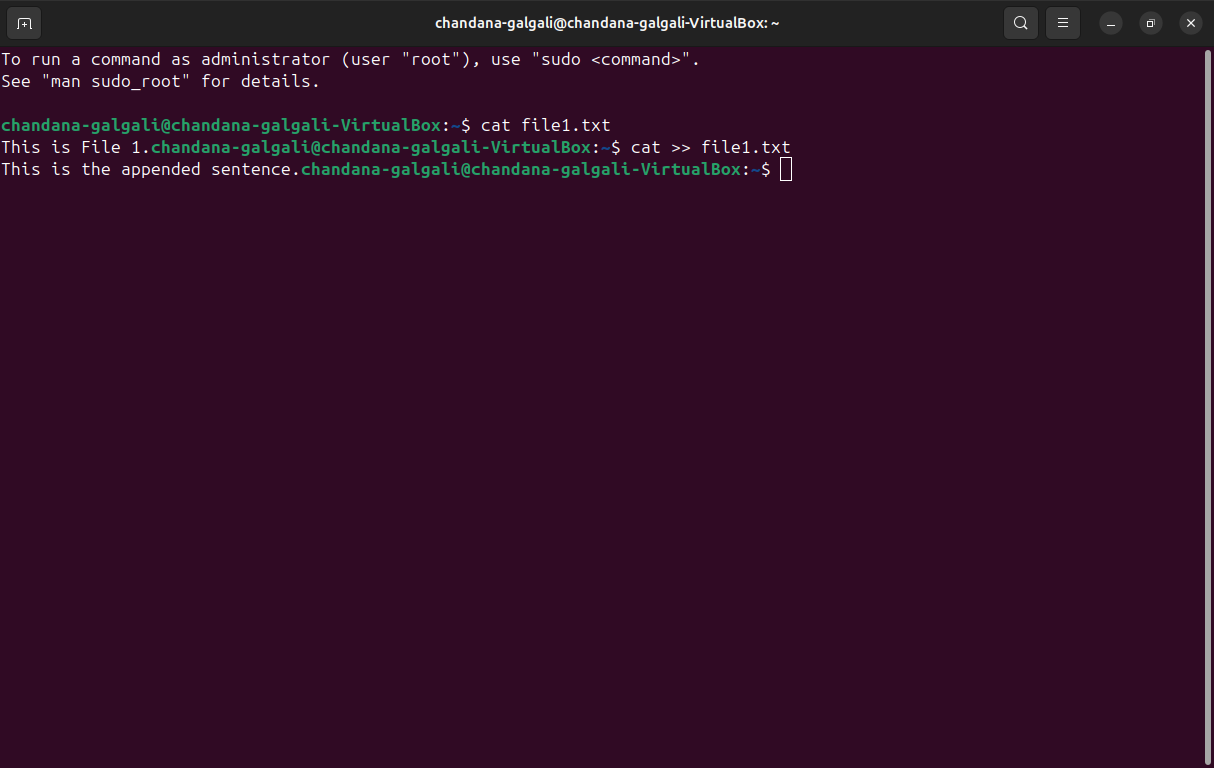
**File-system management:**

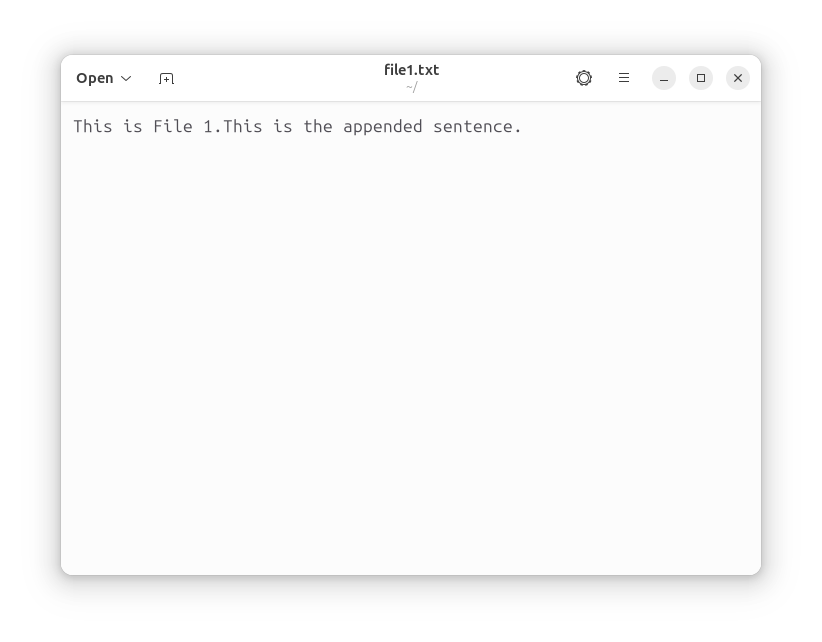
****

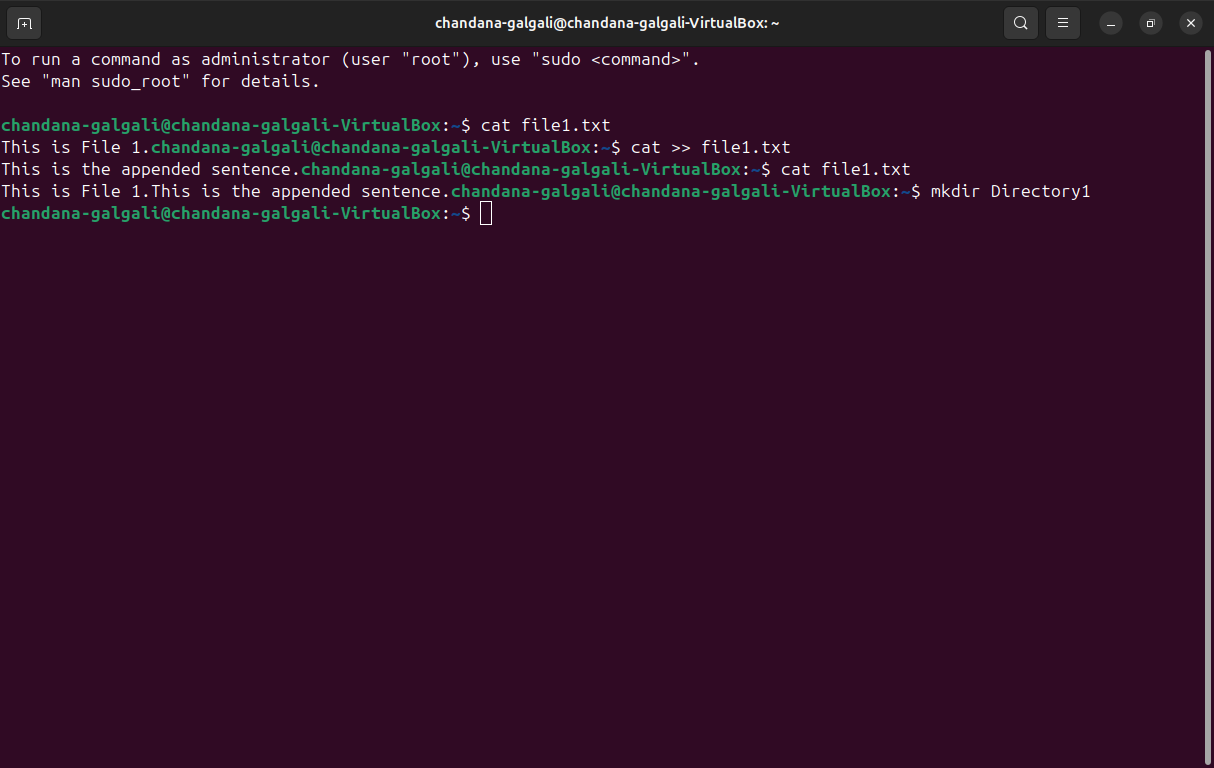
****

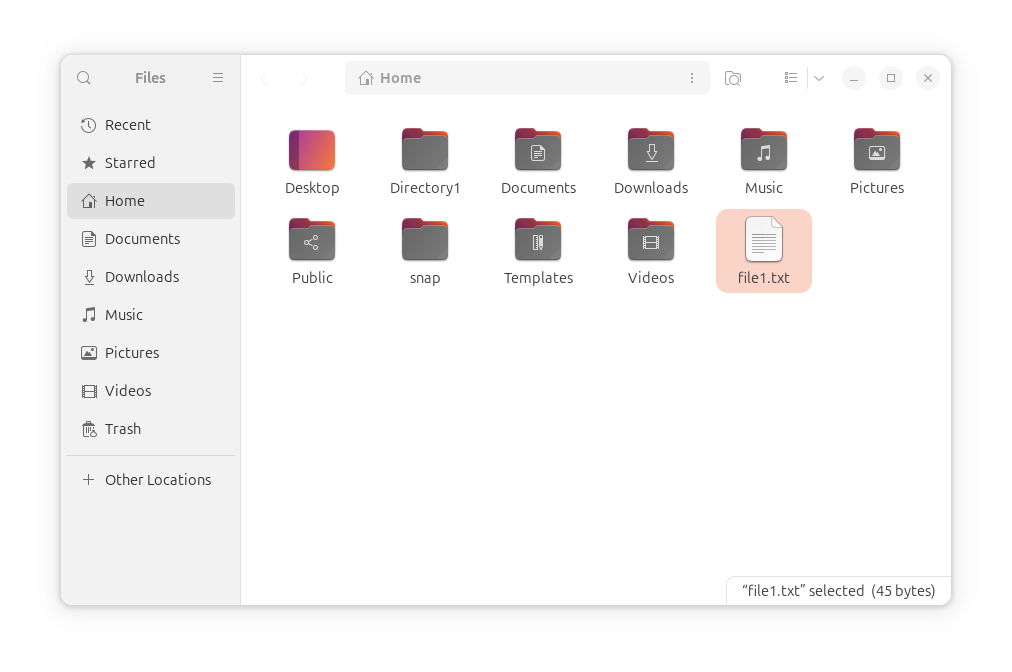
****

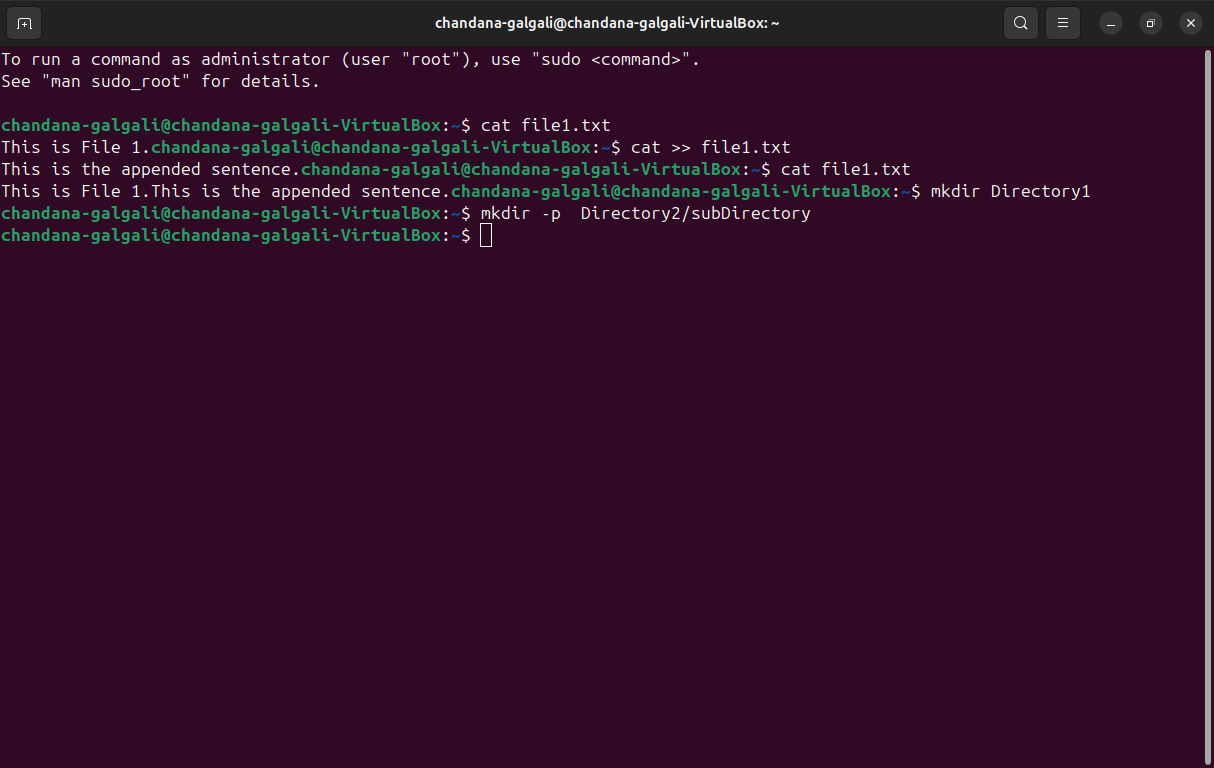
****

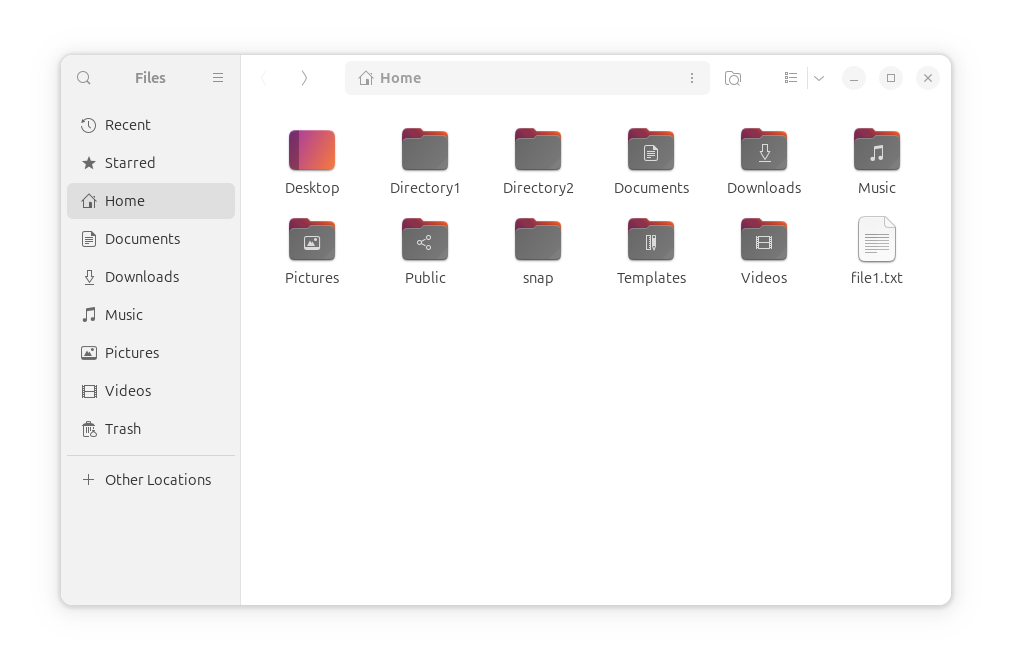
****

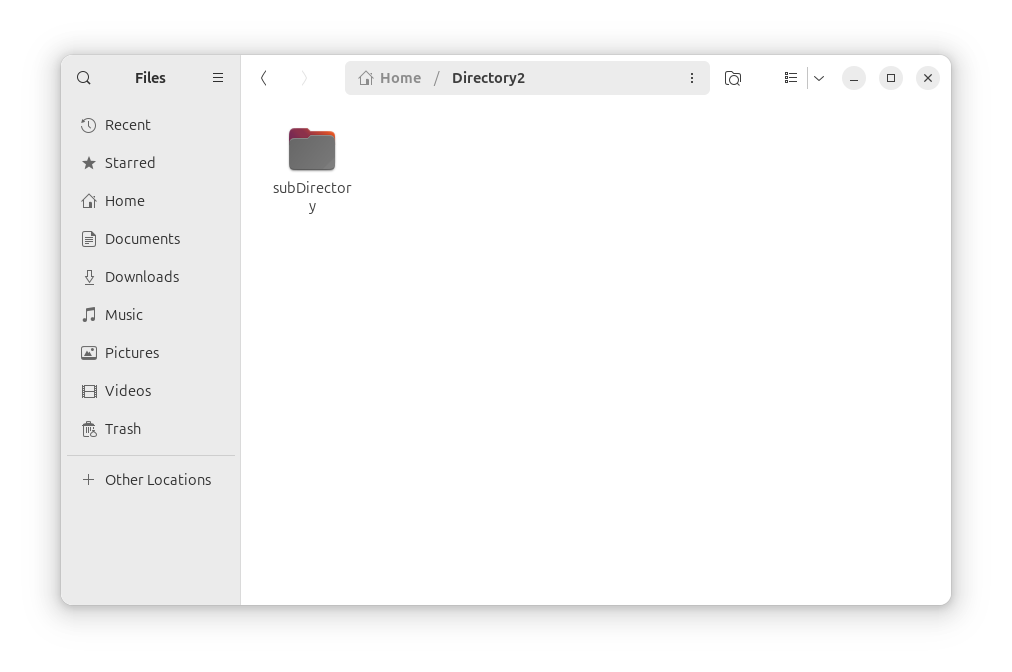
****

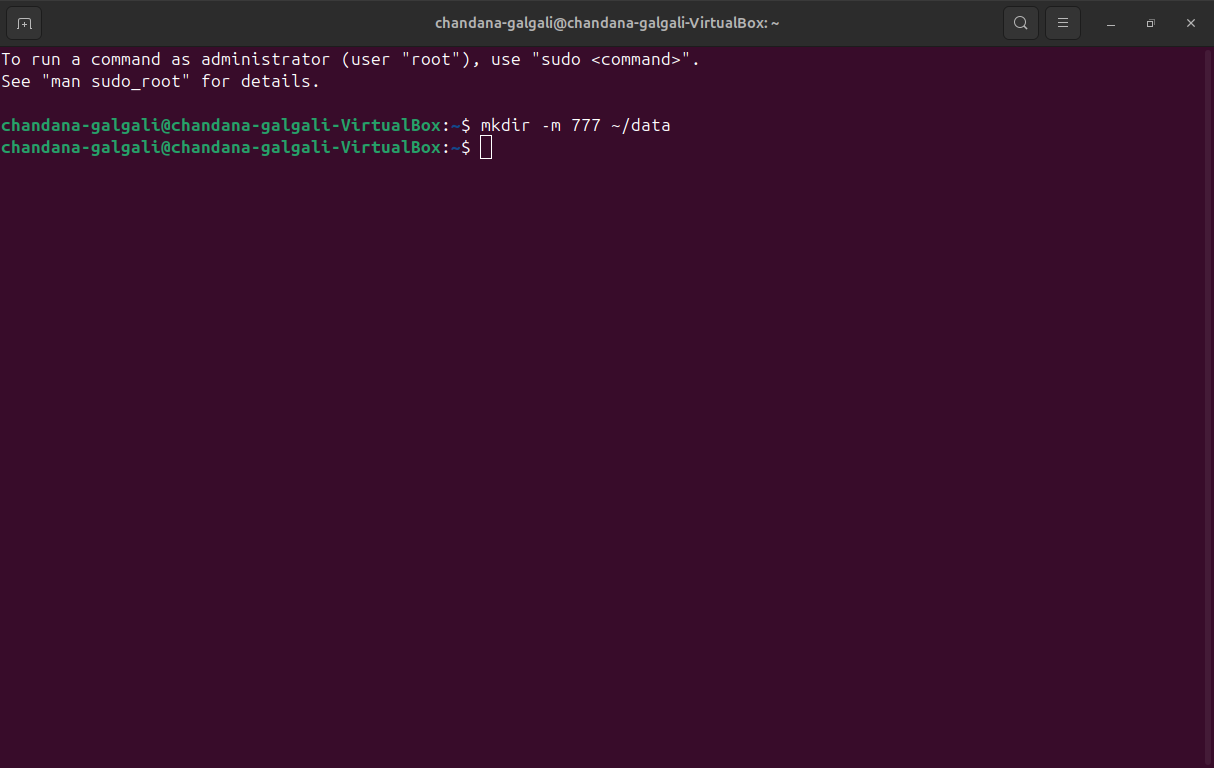
****

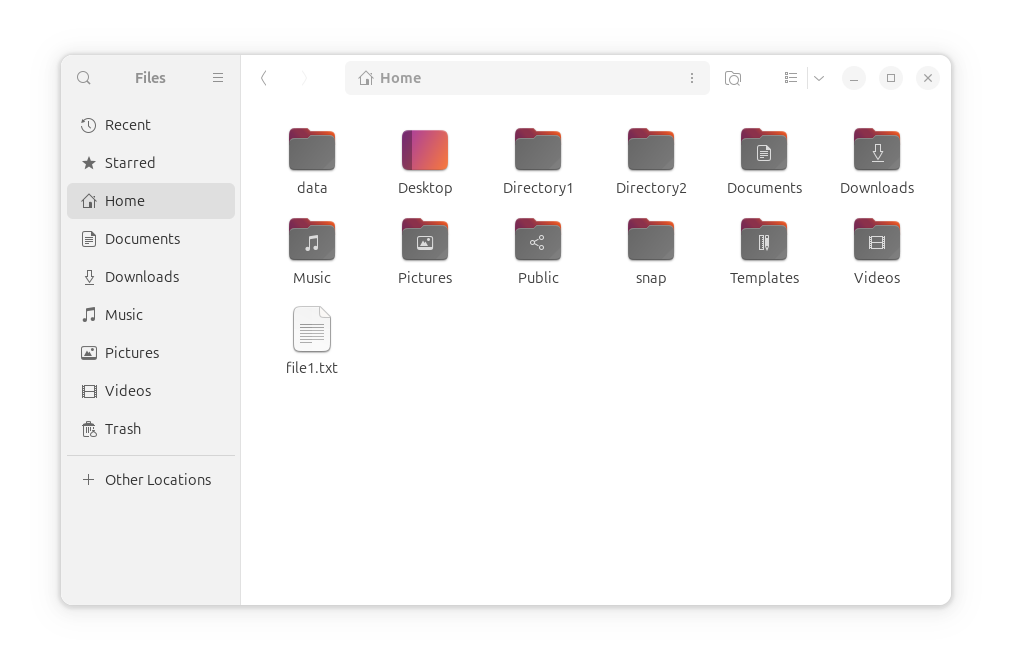
****

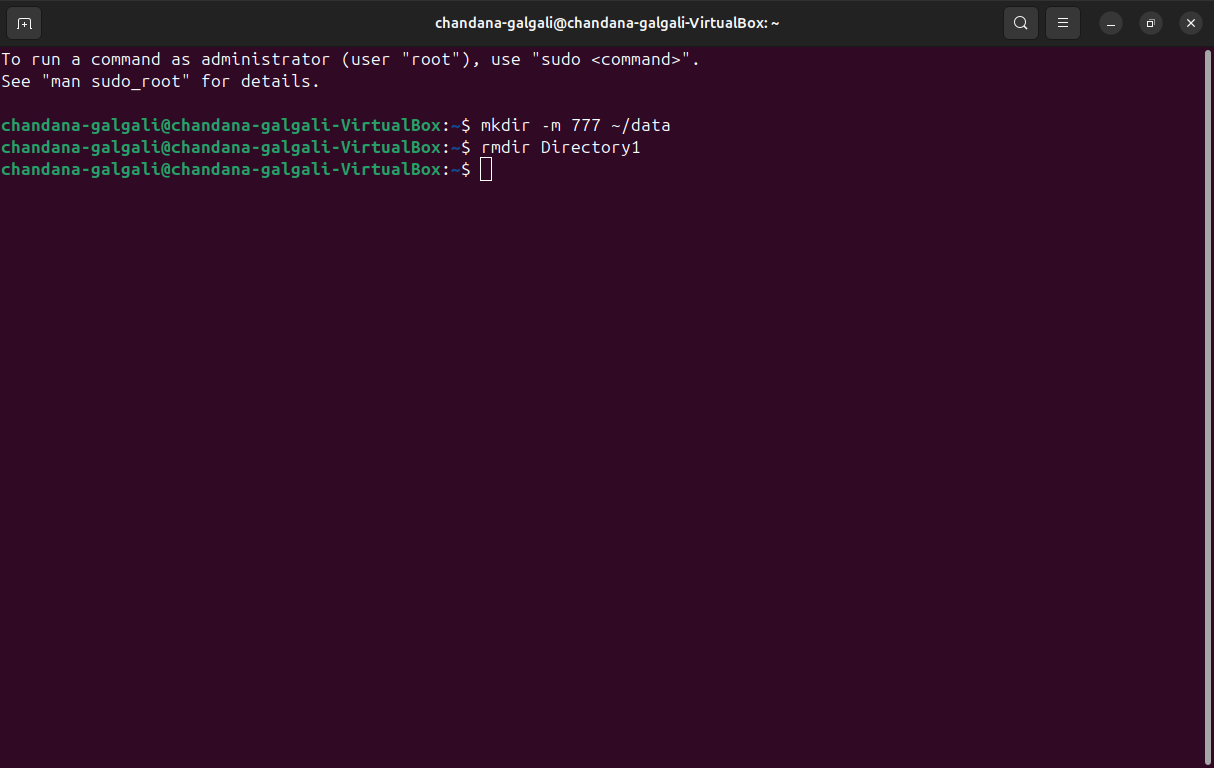
****

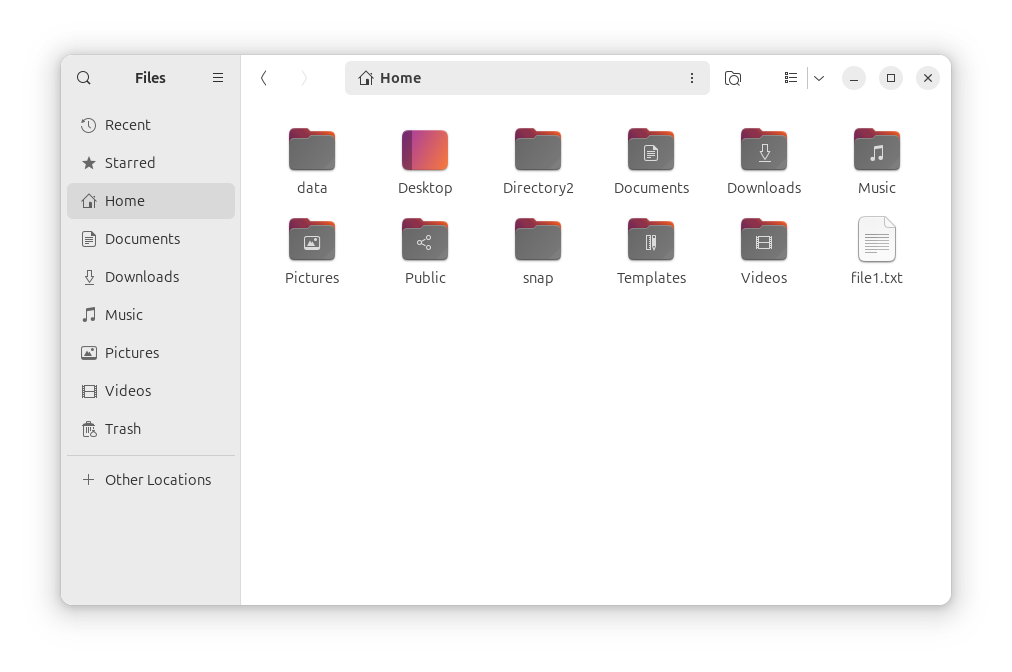
****

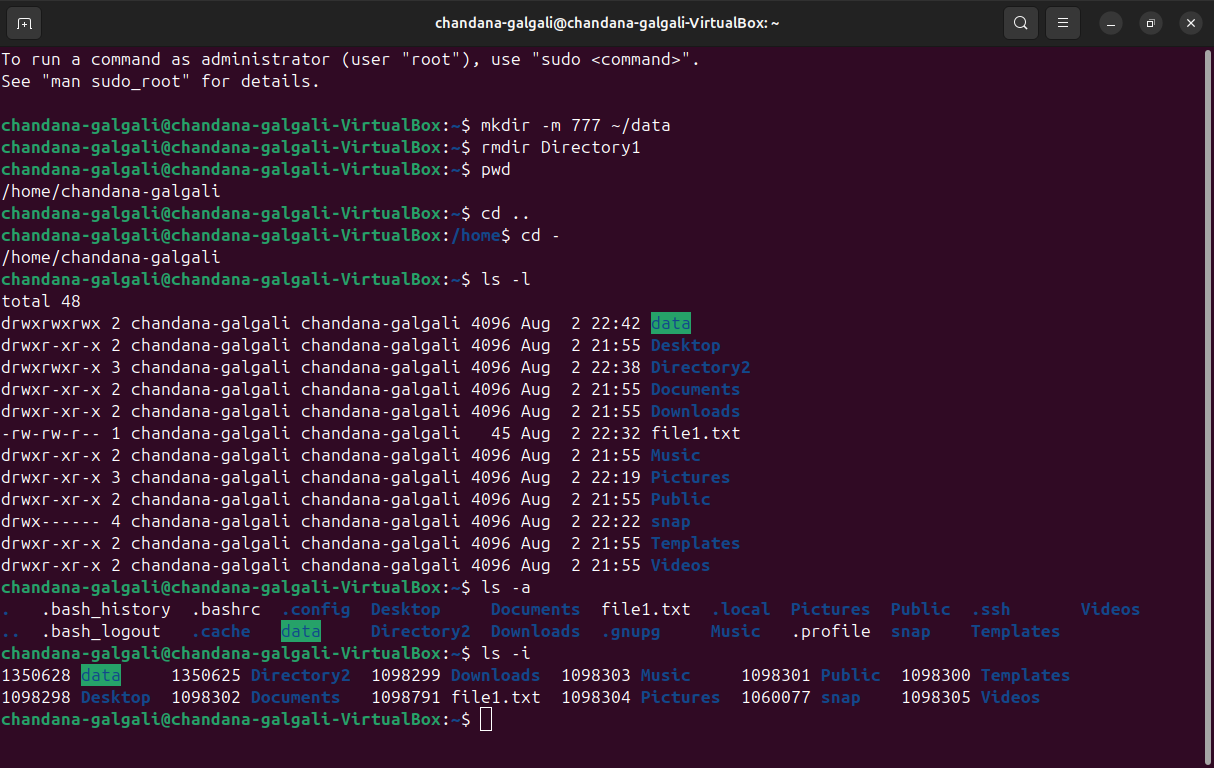
****

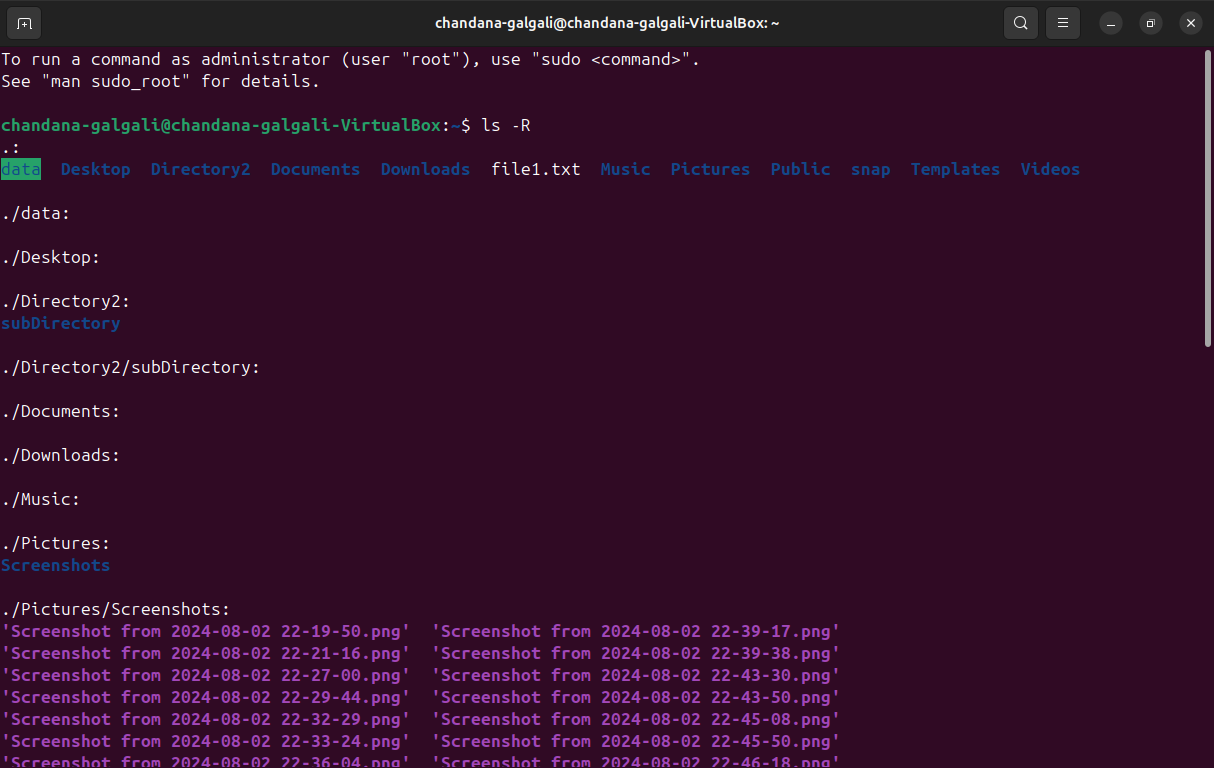
****

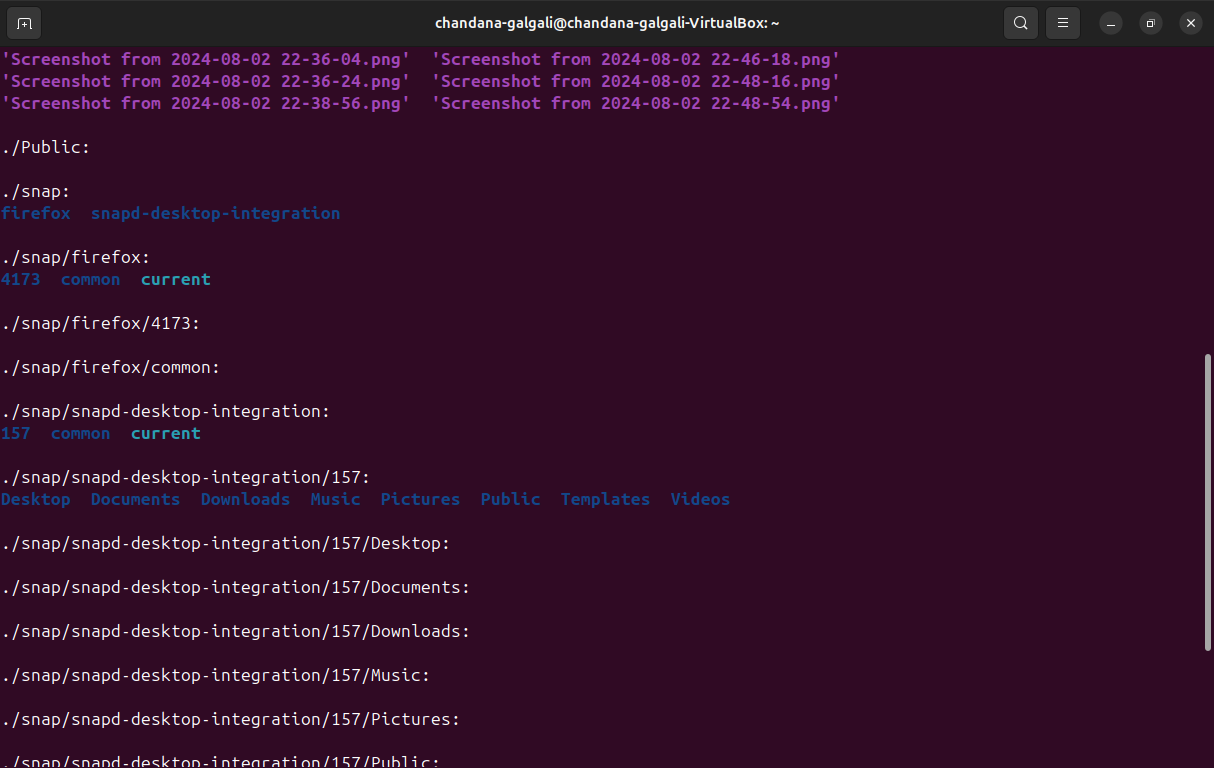
****

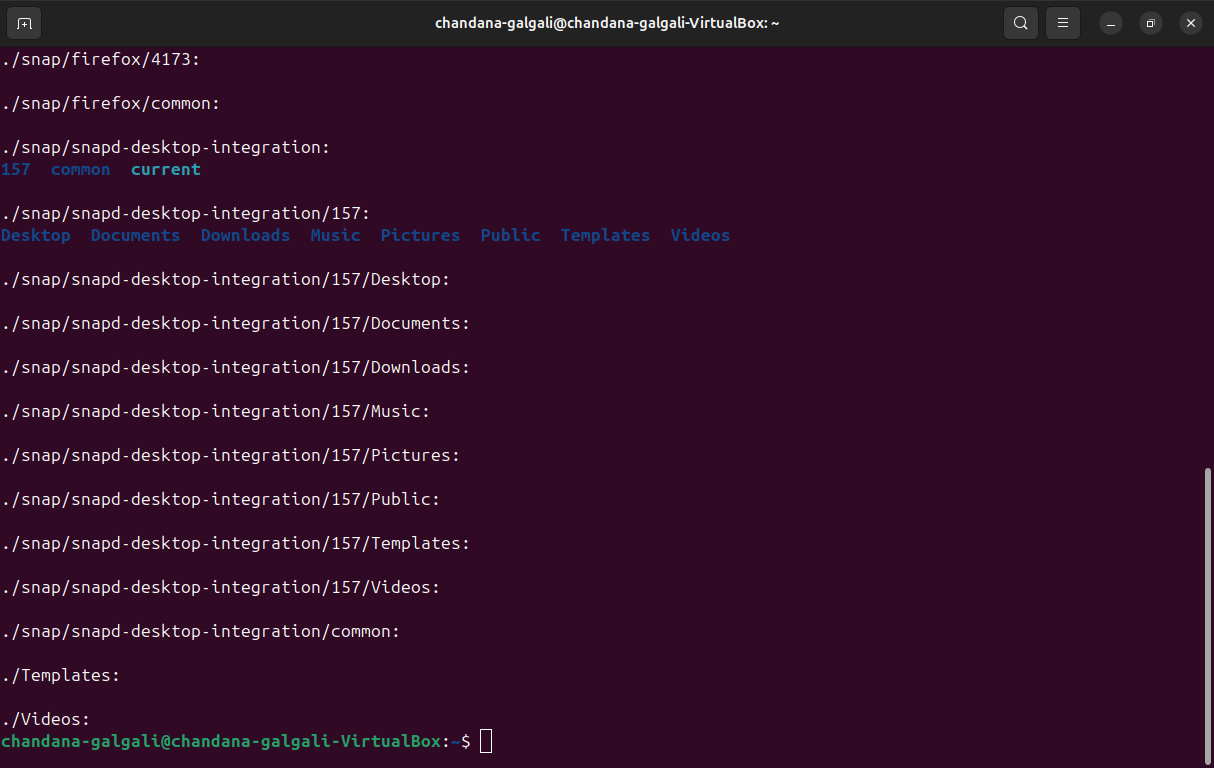
****

****

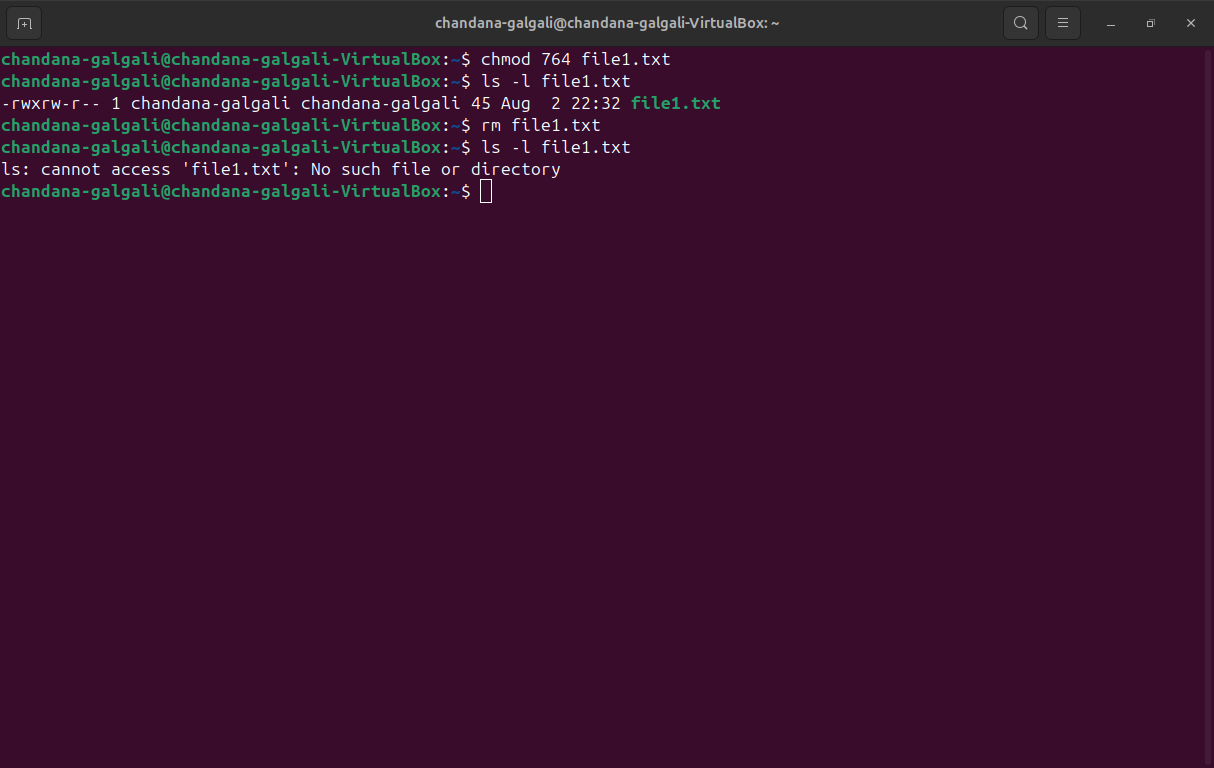
****

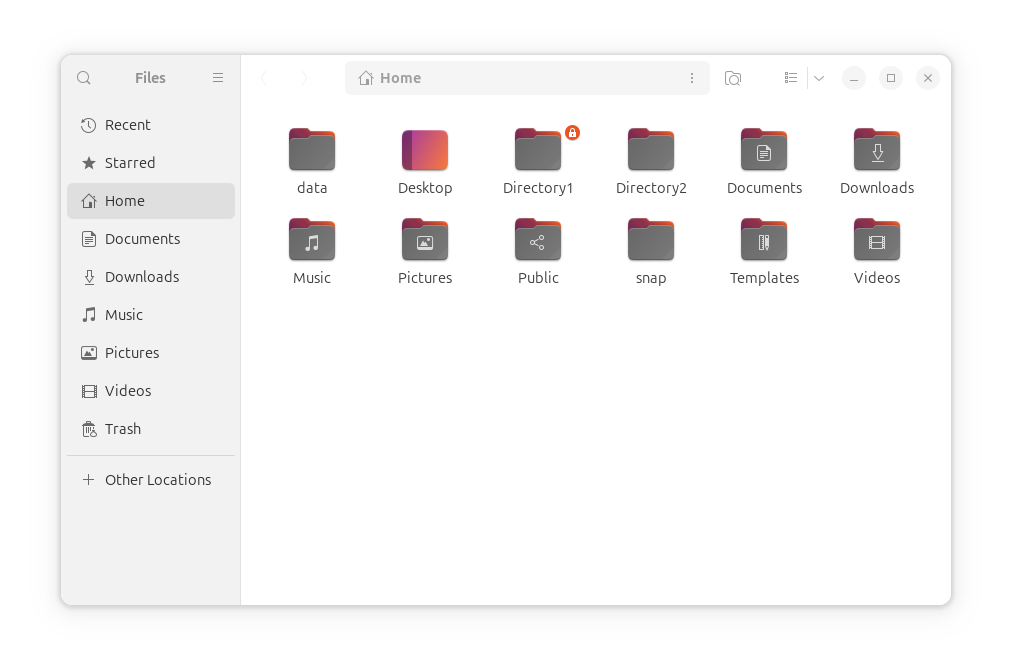
****

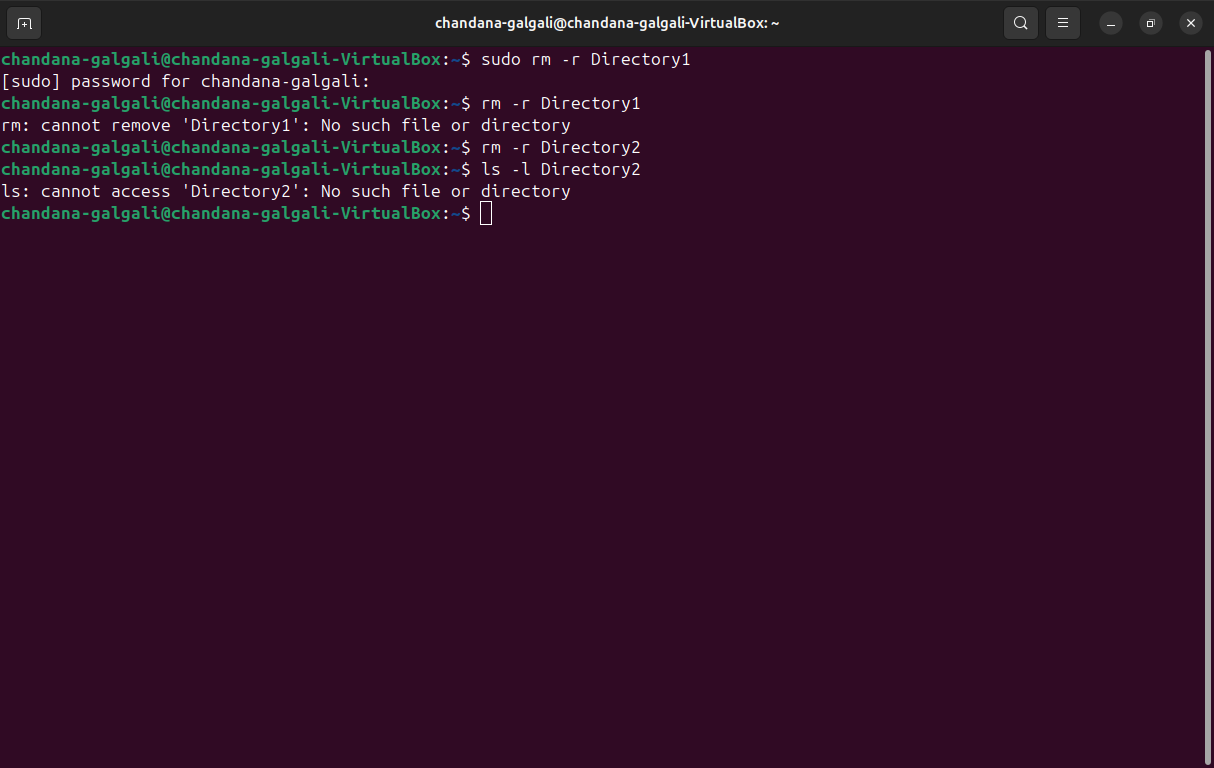
****

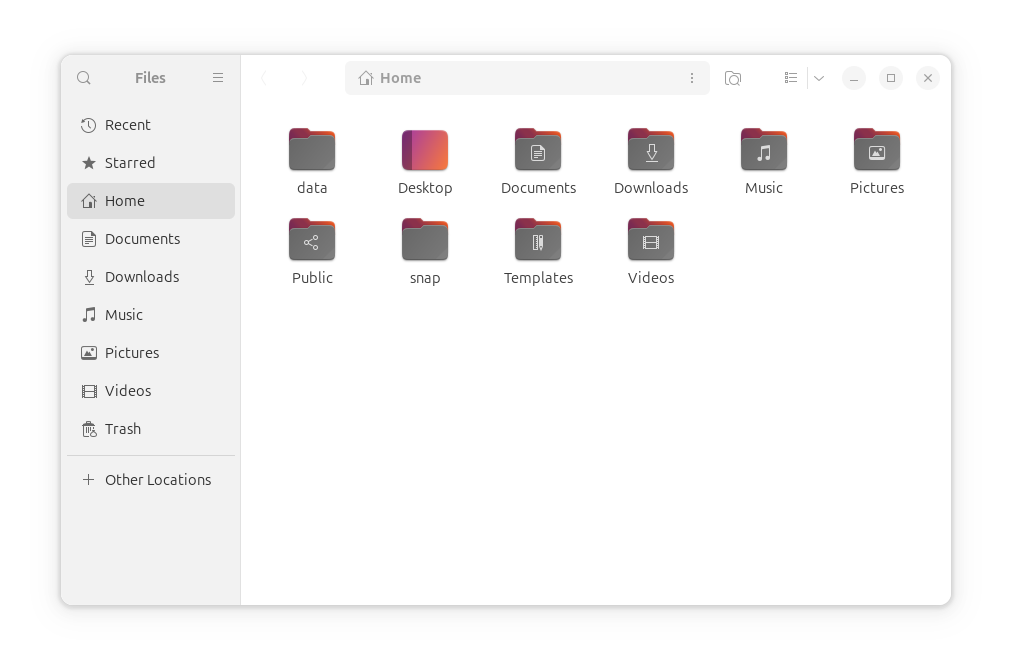
****

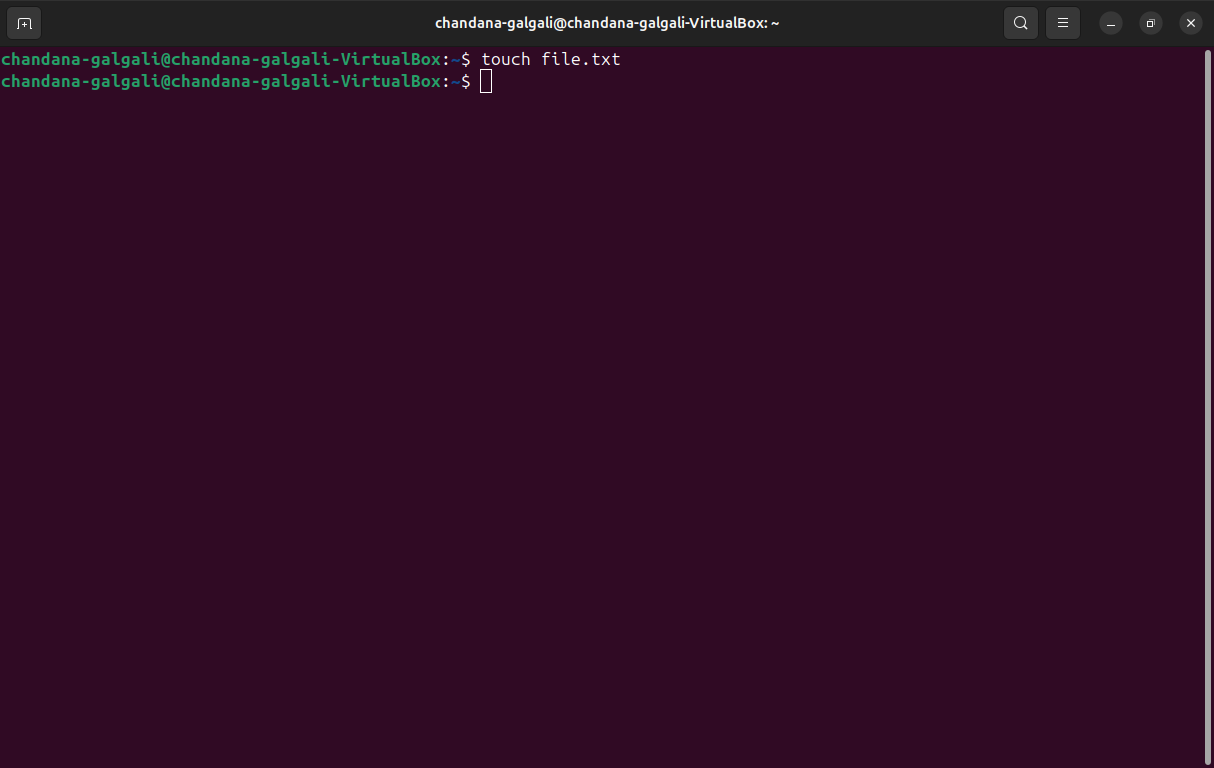
****

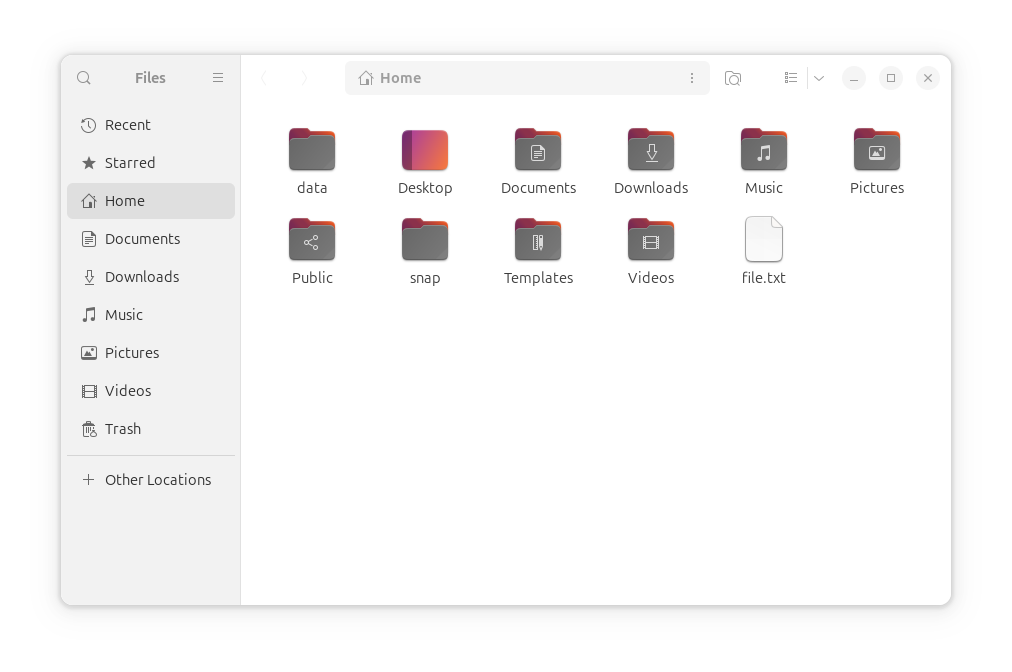
****

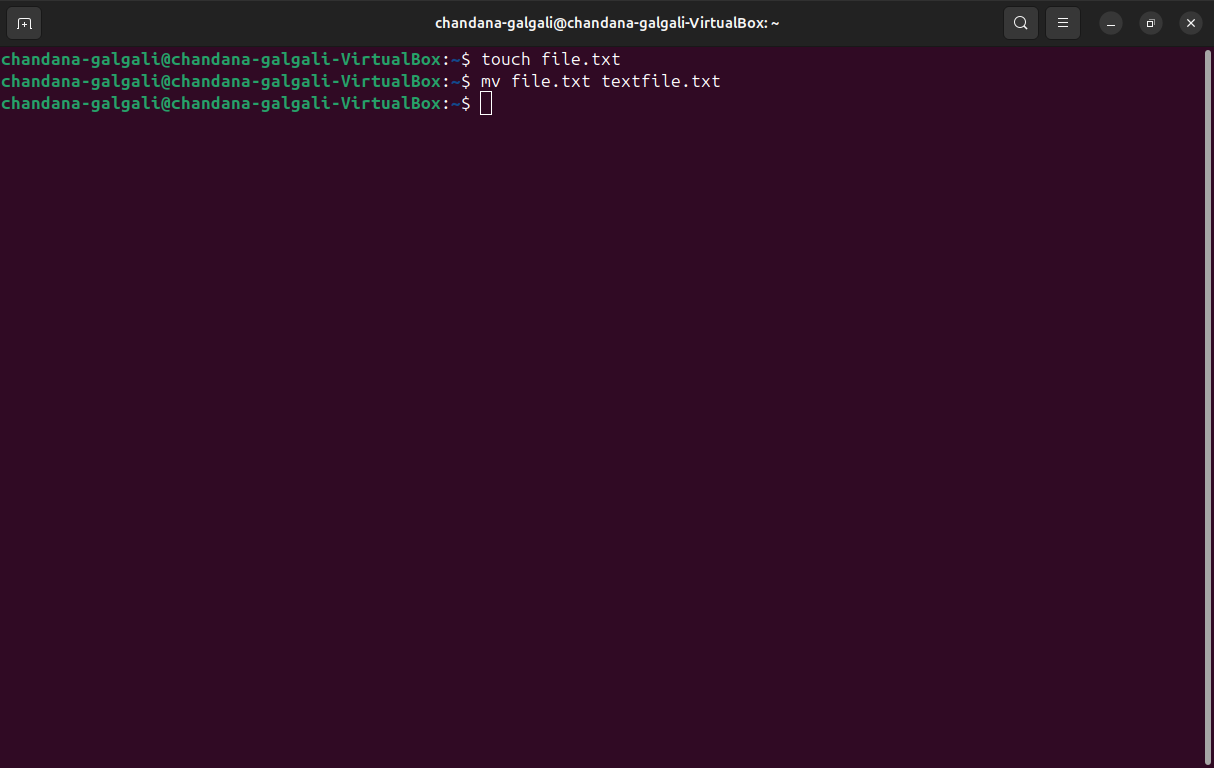
****

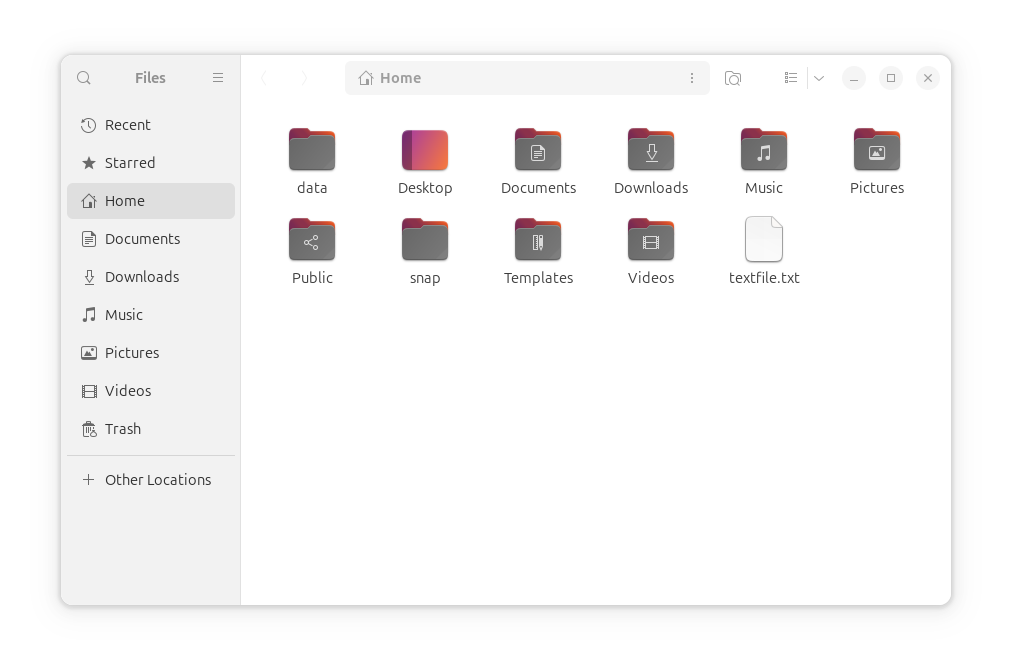
****

****

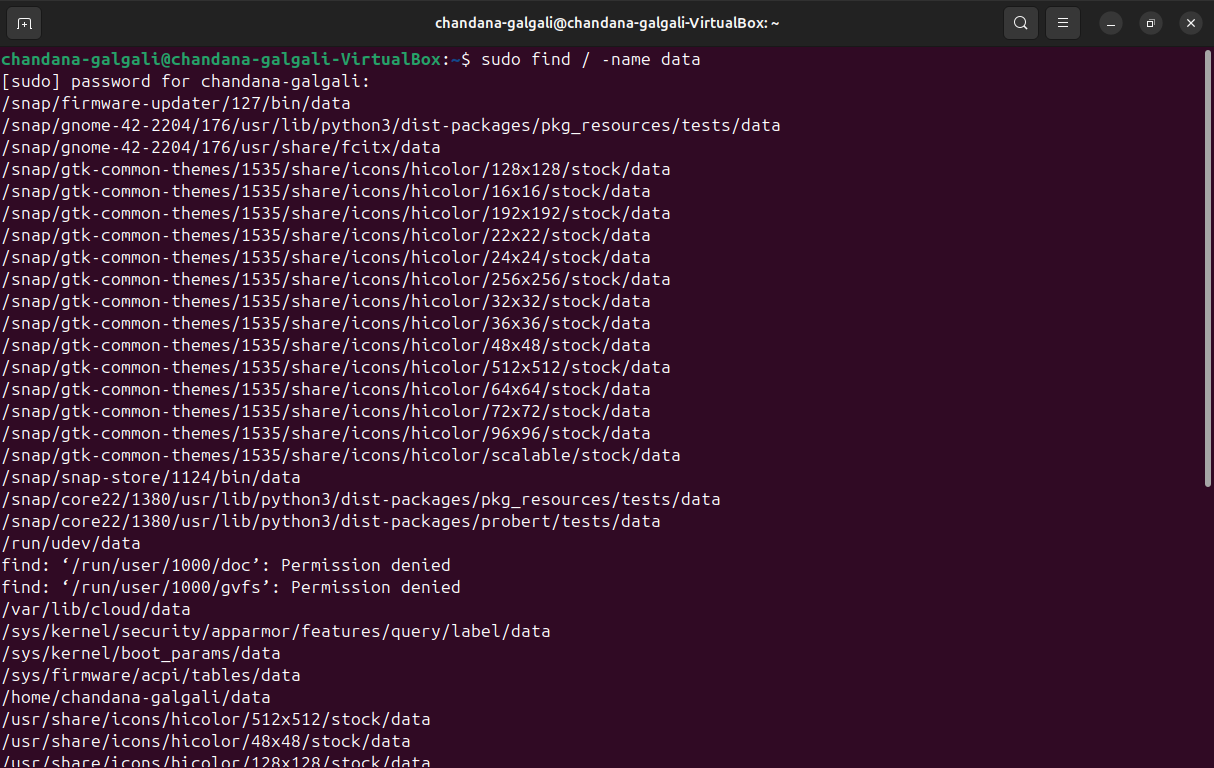
****

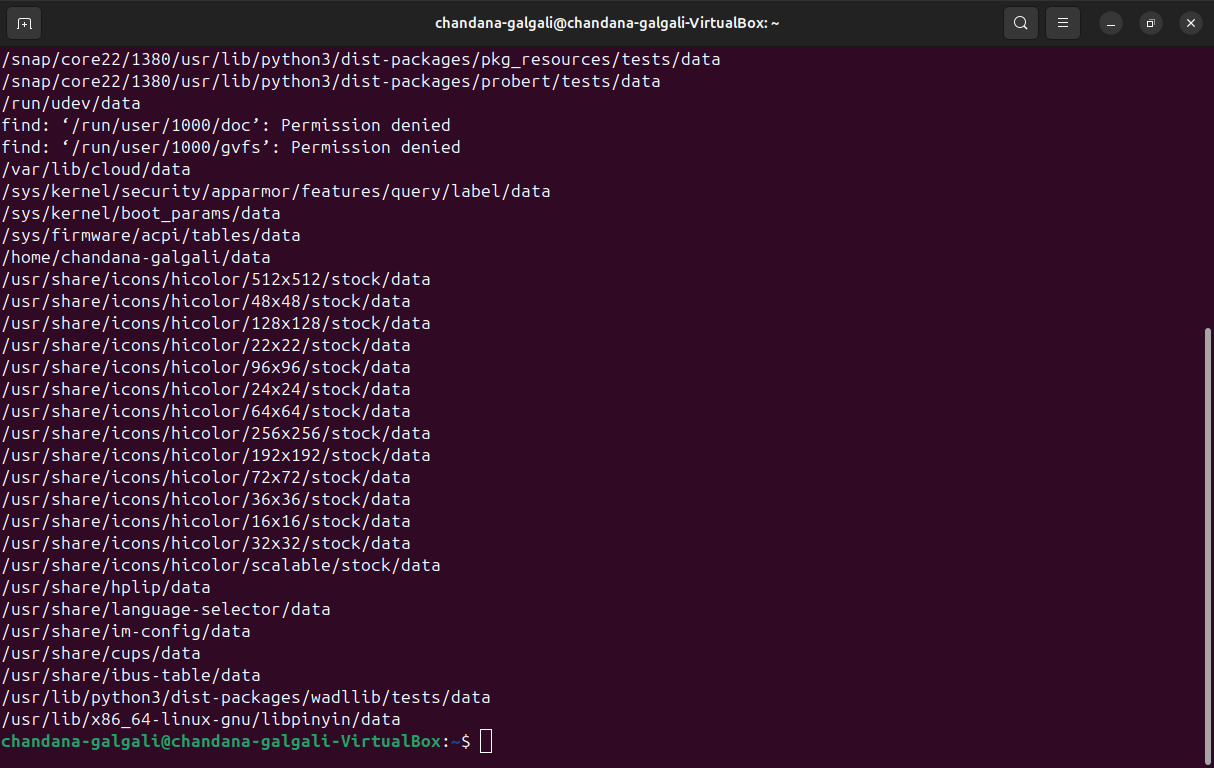
****

****

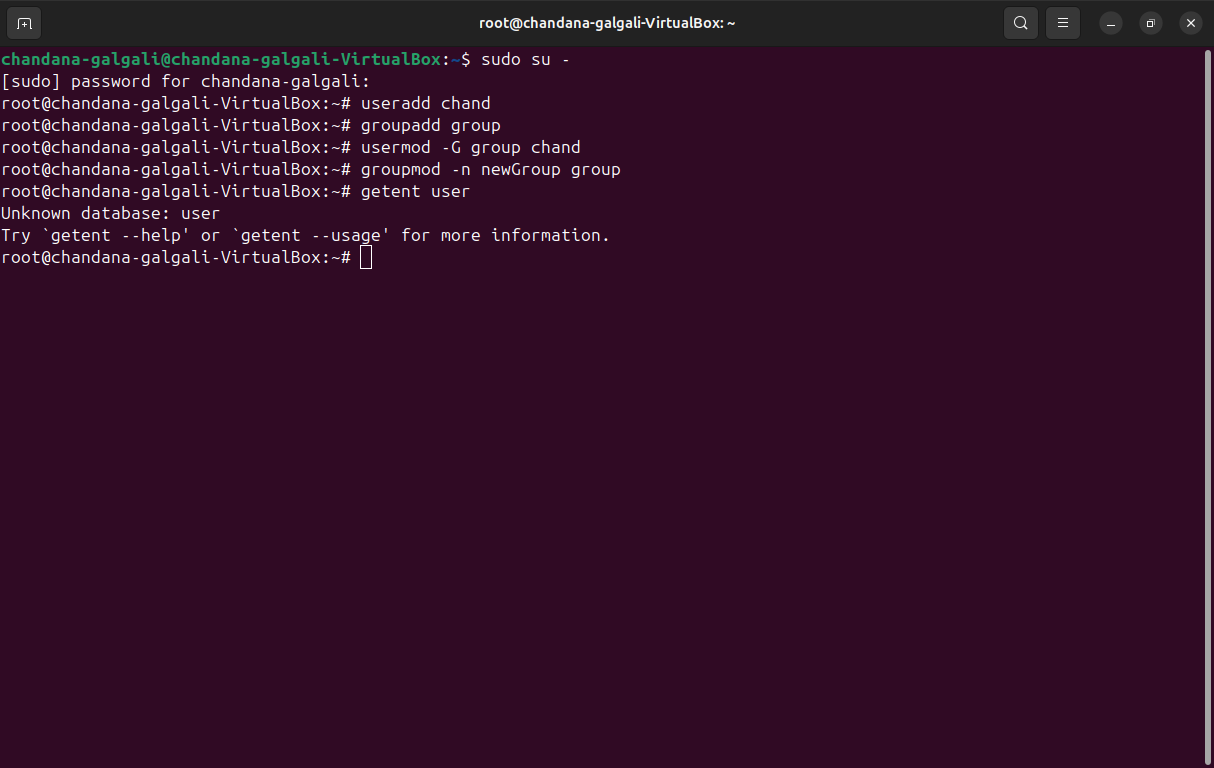
****

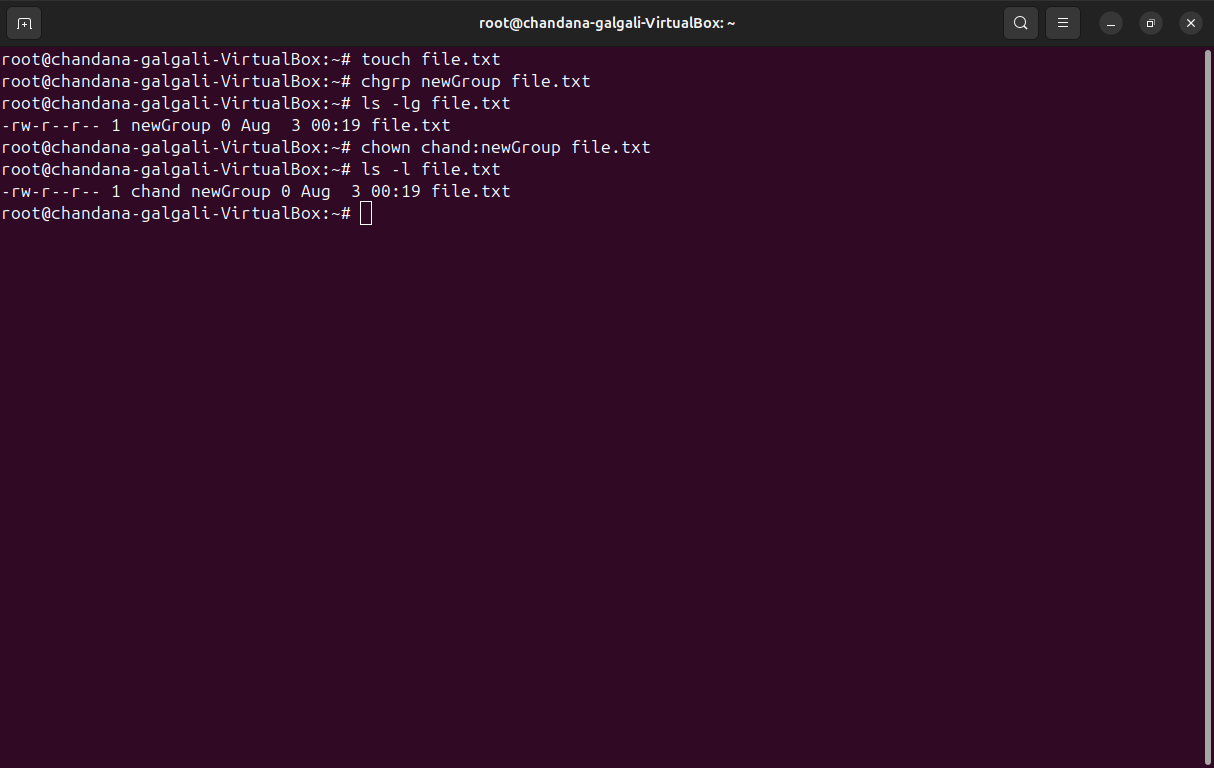
****

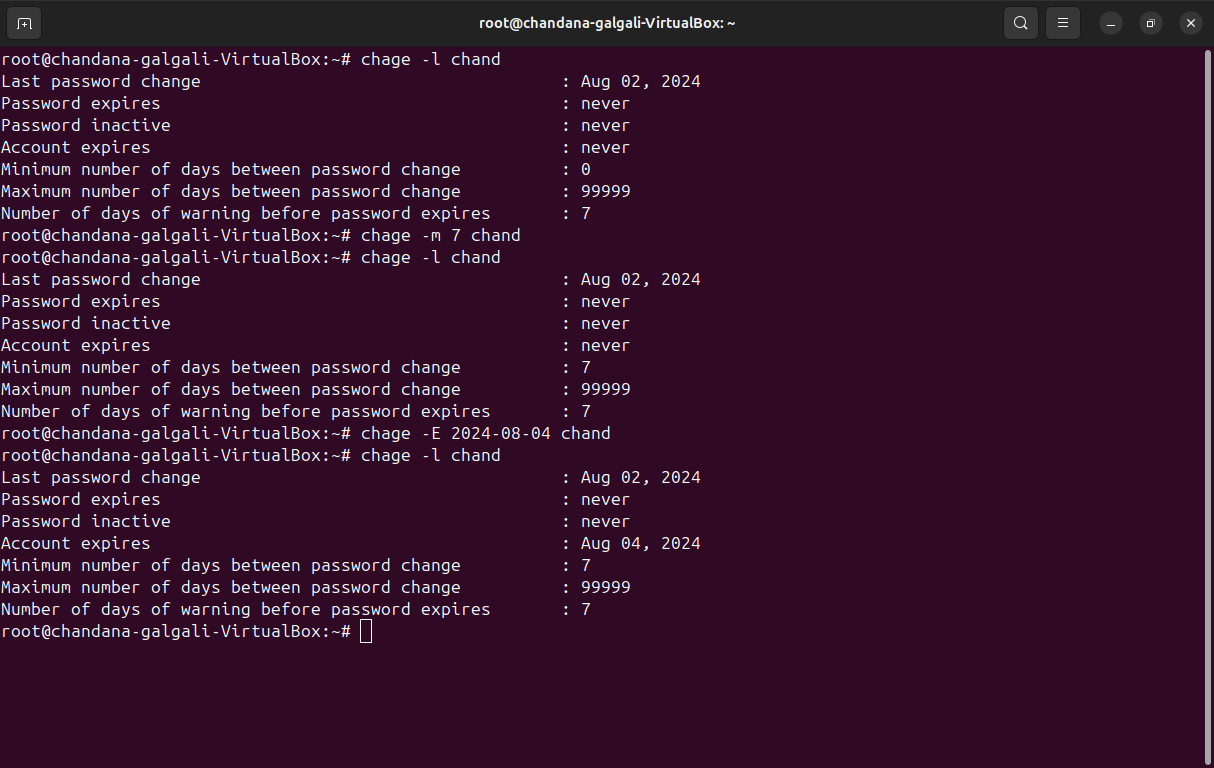
****

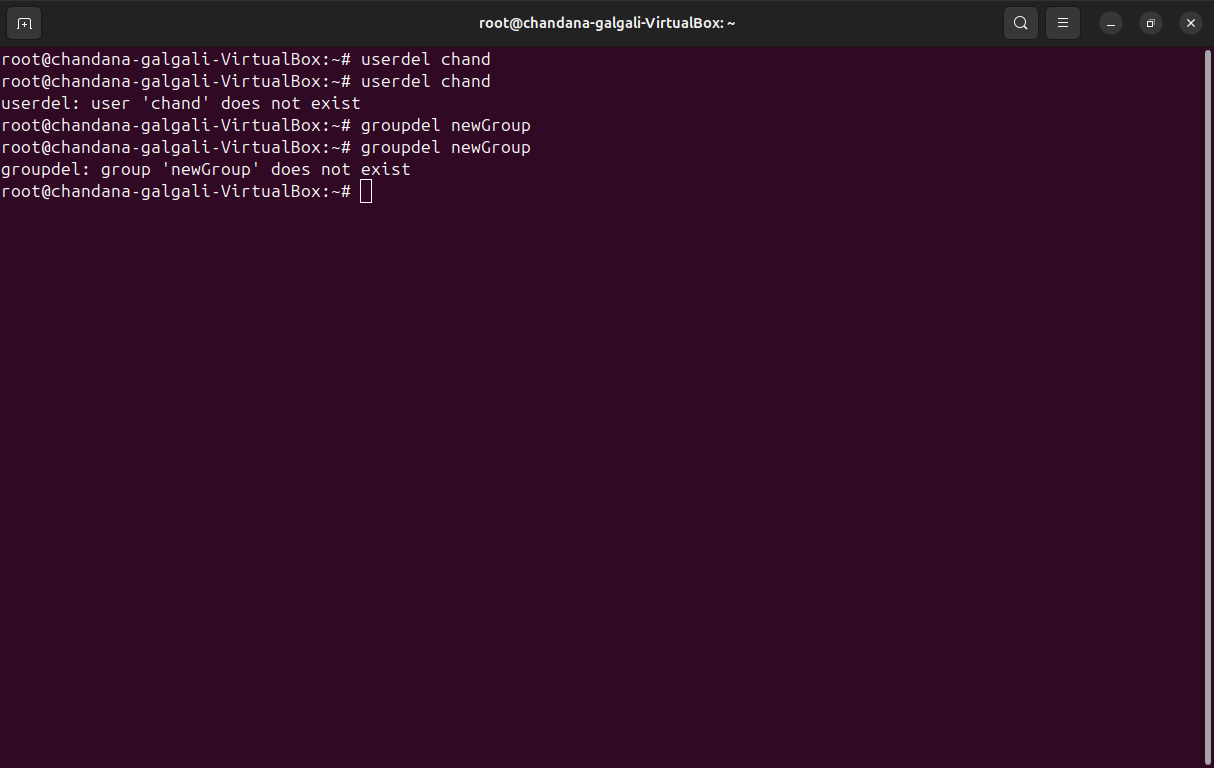
****

**User administration:**

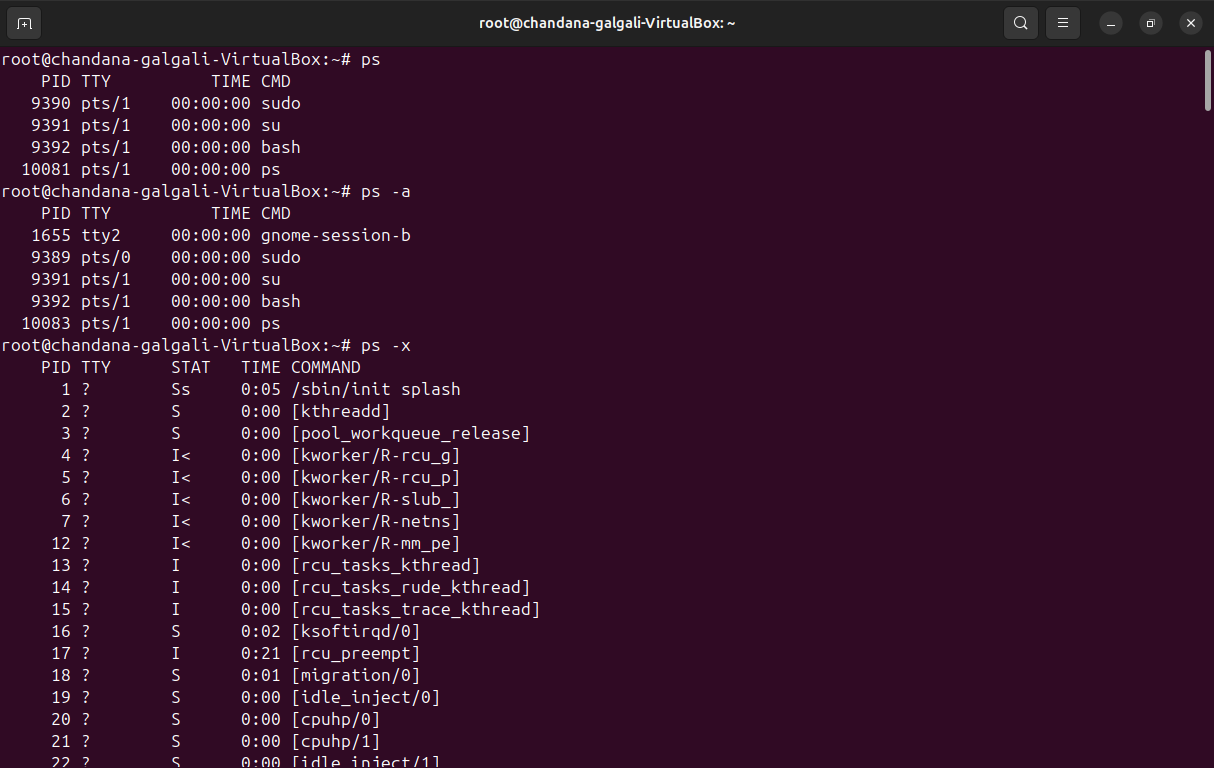
****

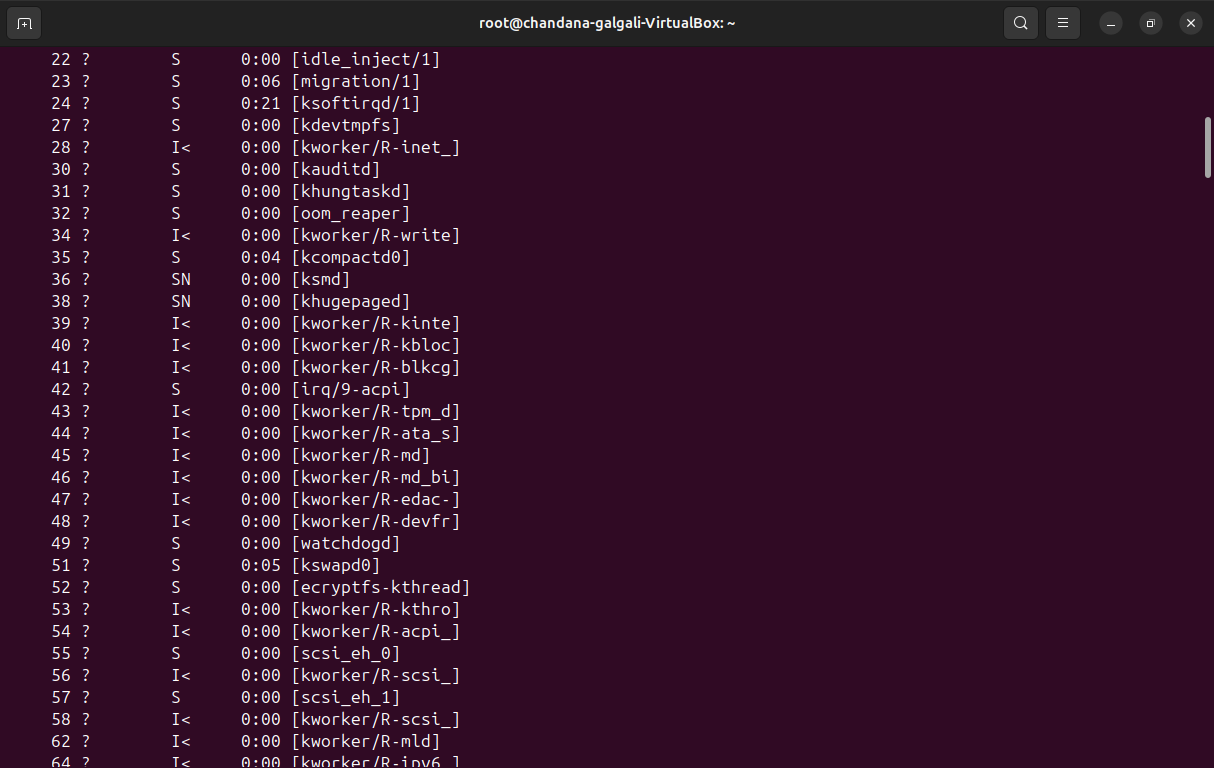
****

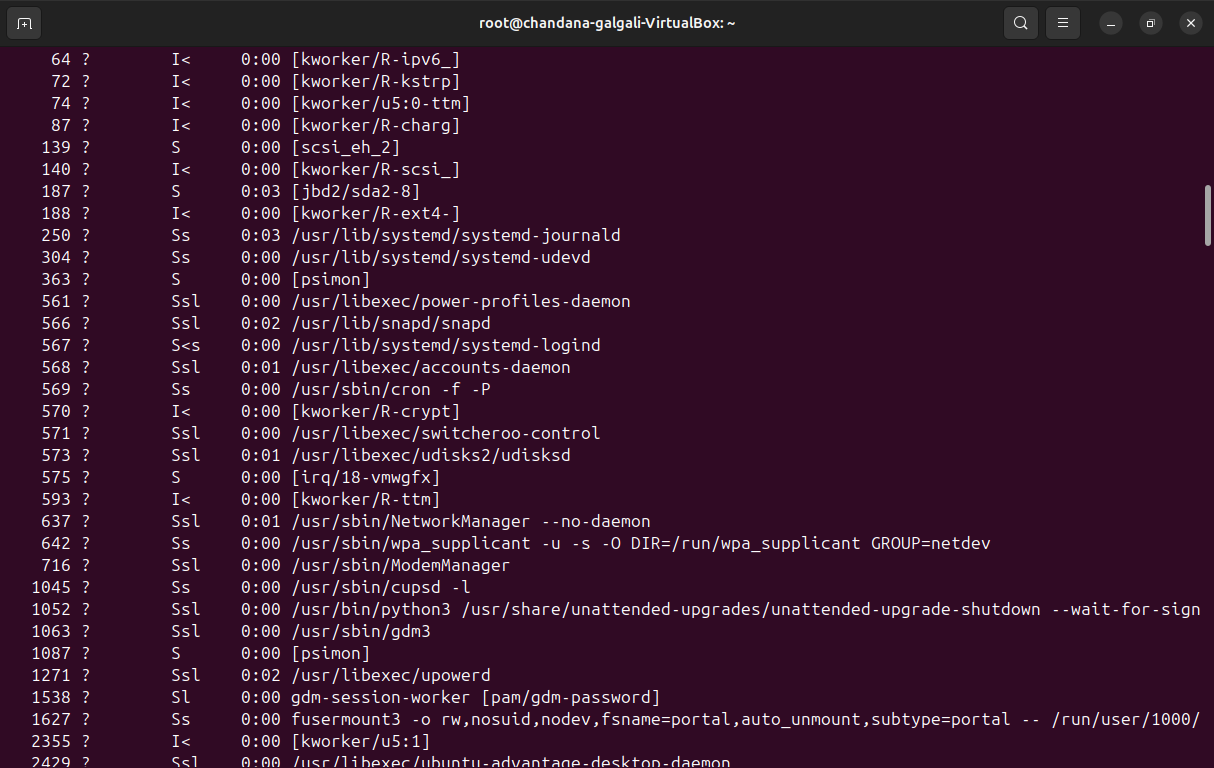
****

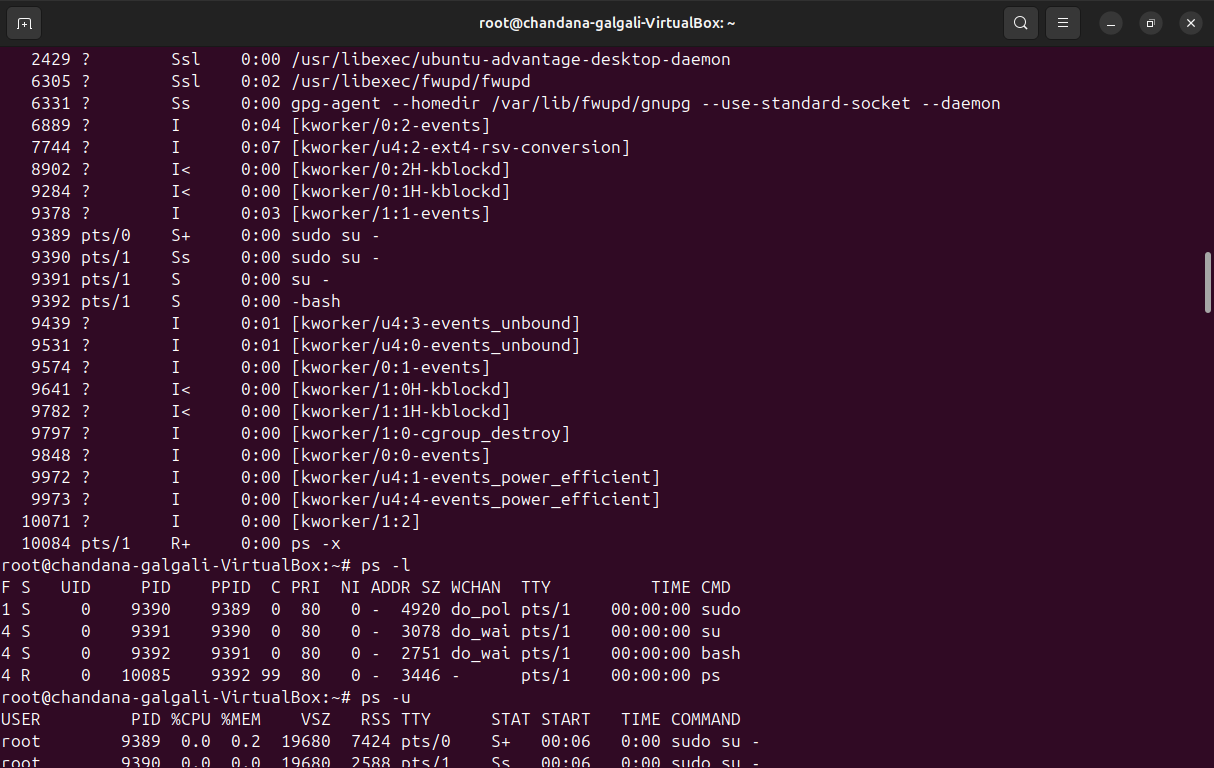
****

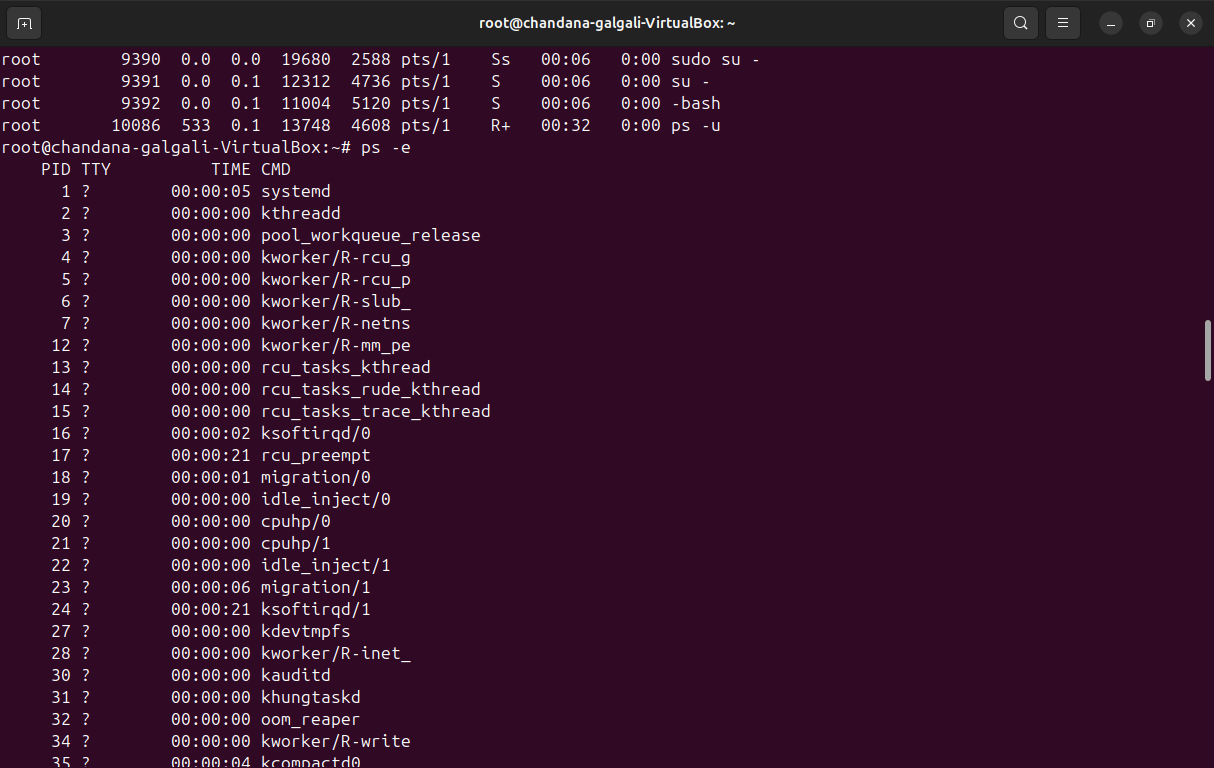
**Process Management:**

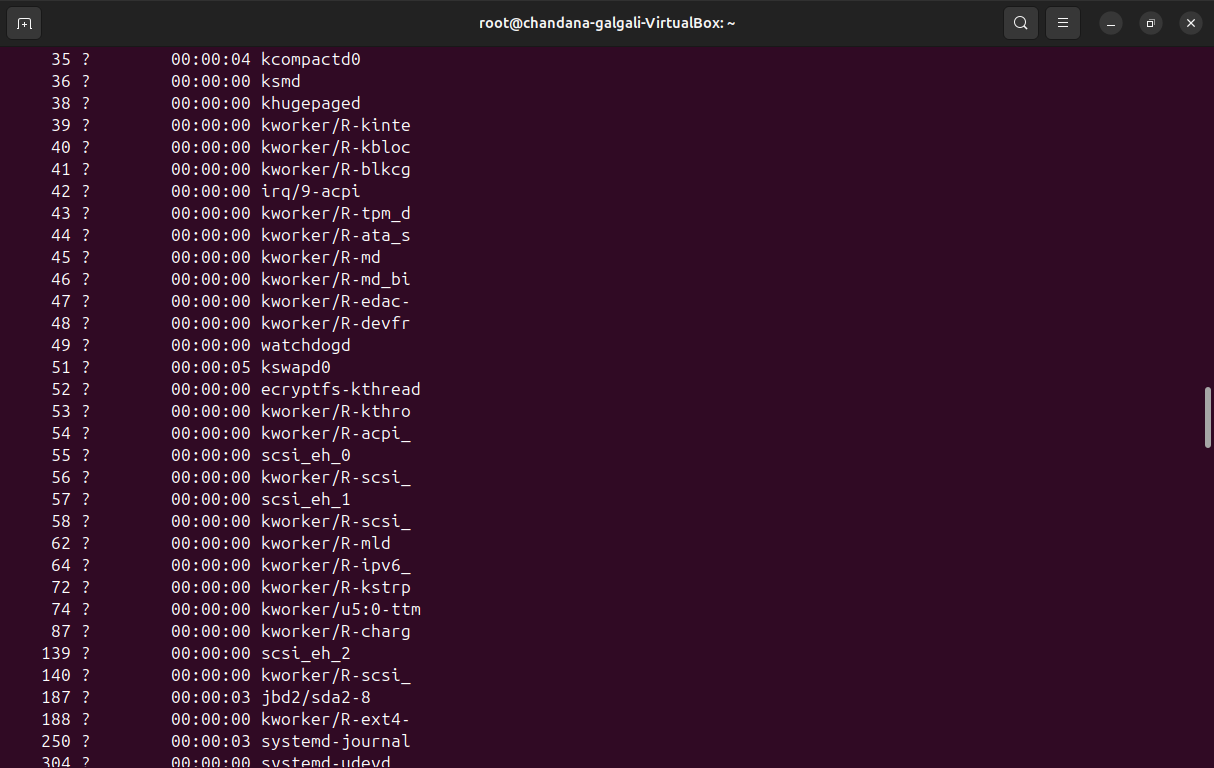
****

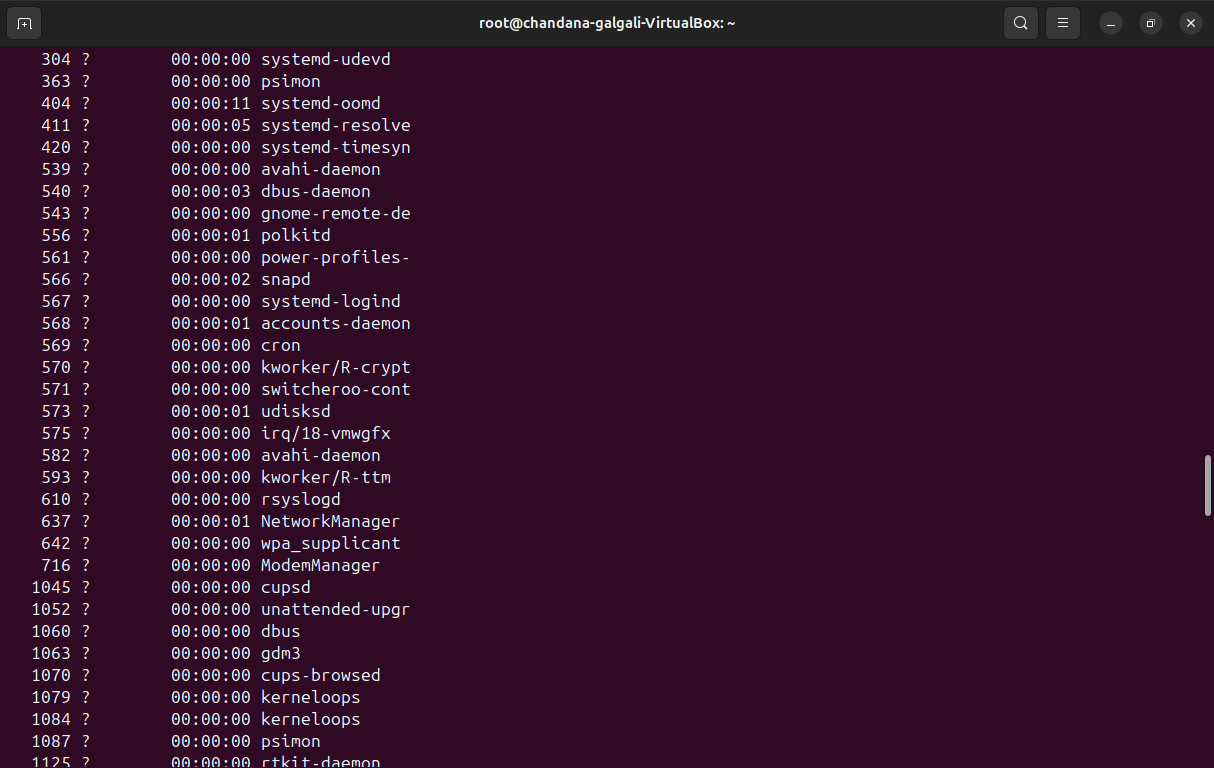
****

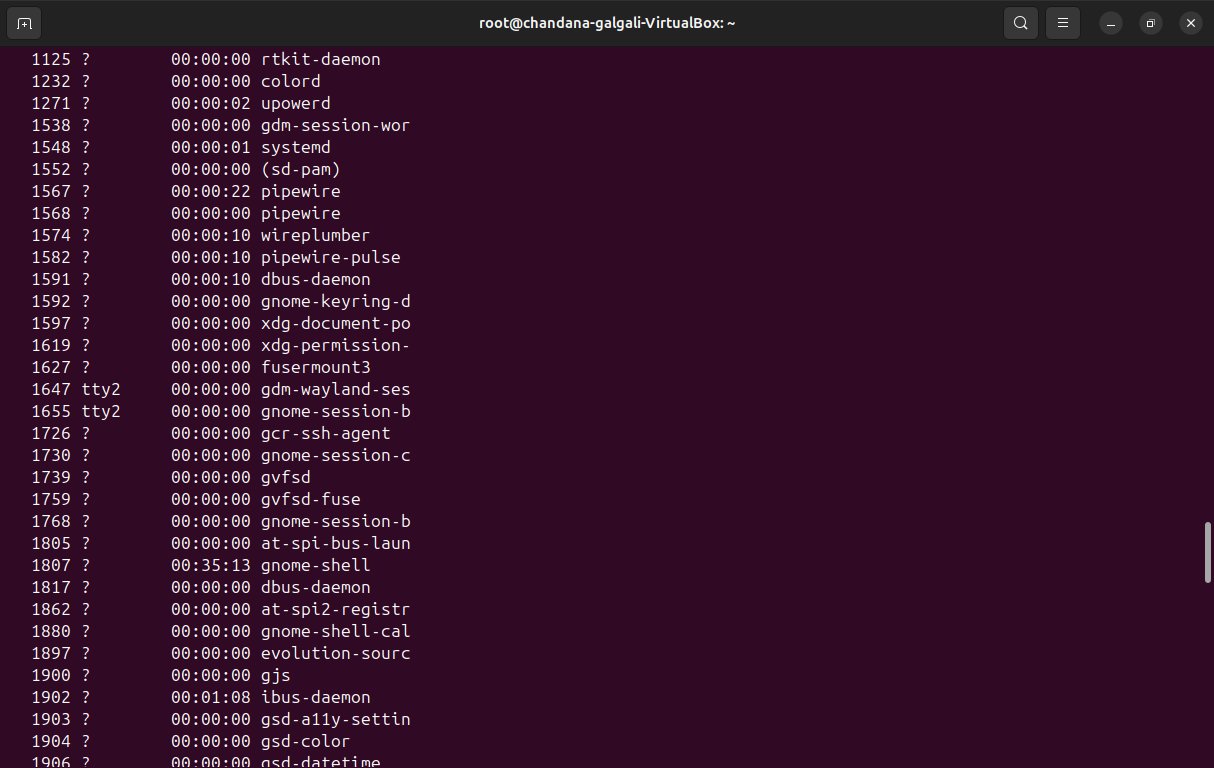
****

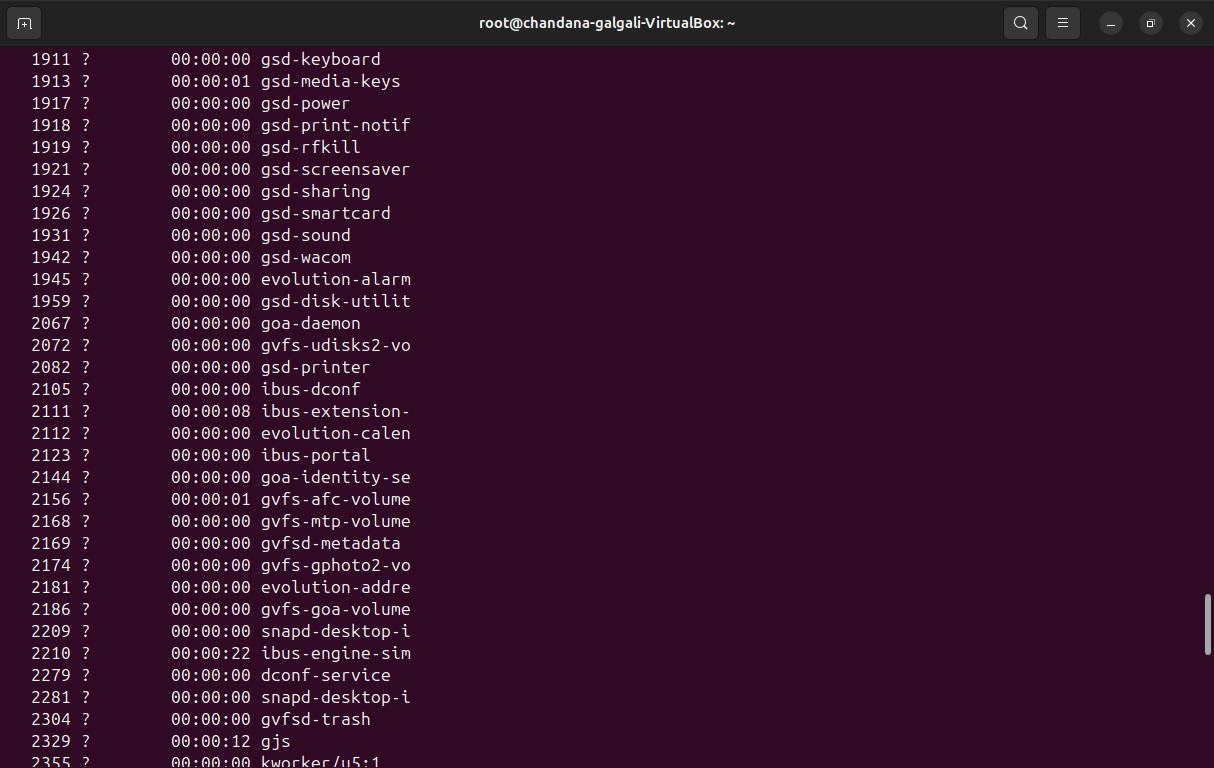
****

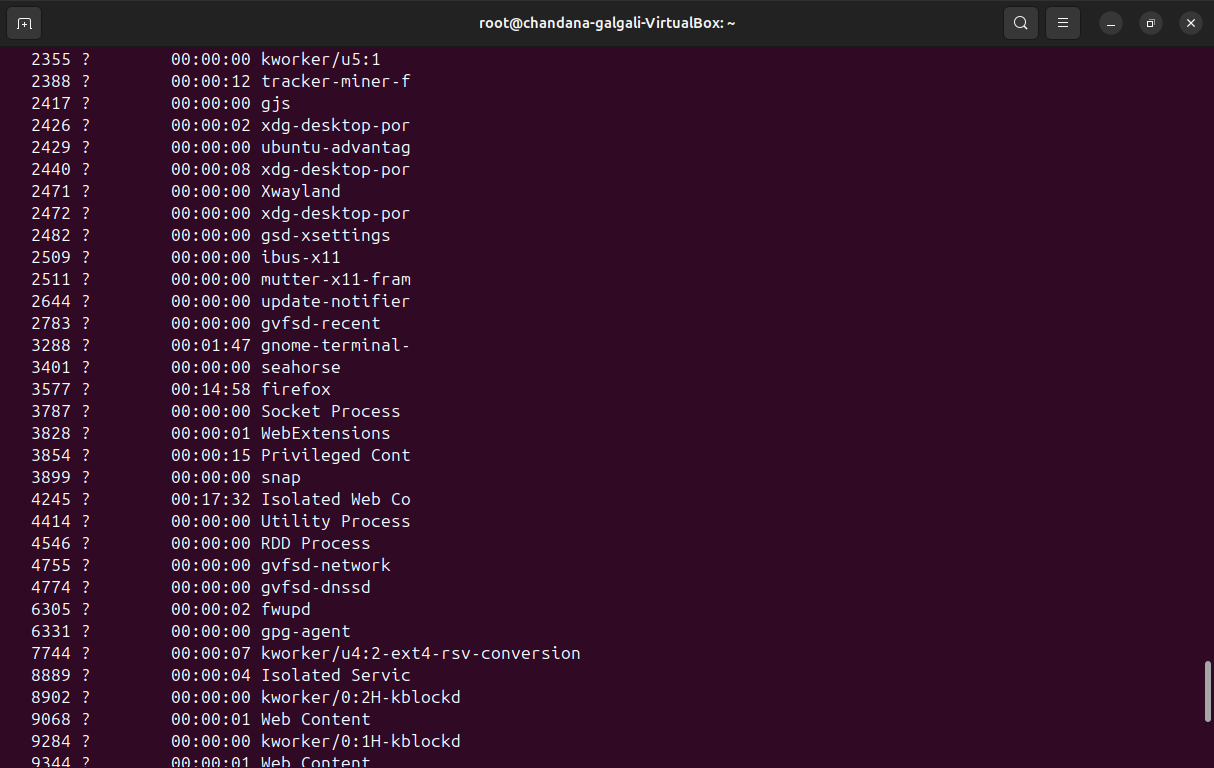
****

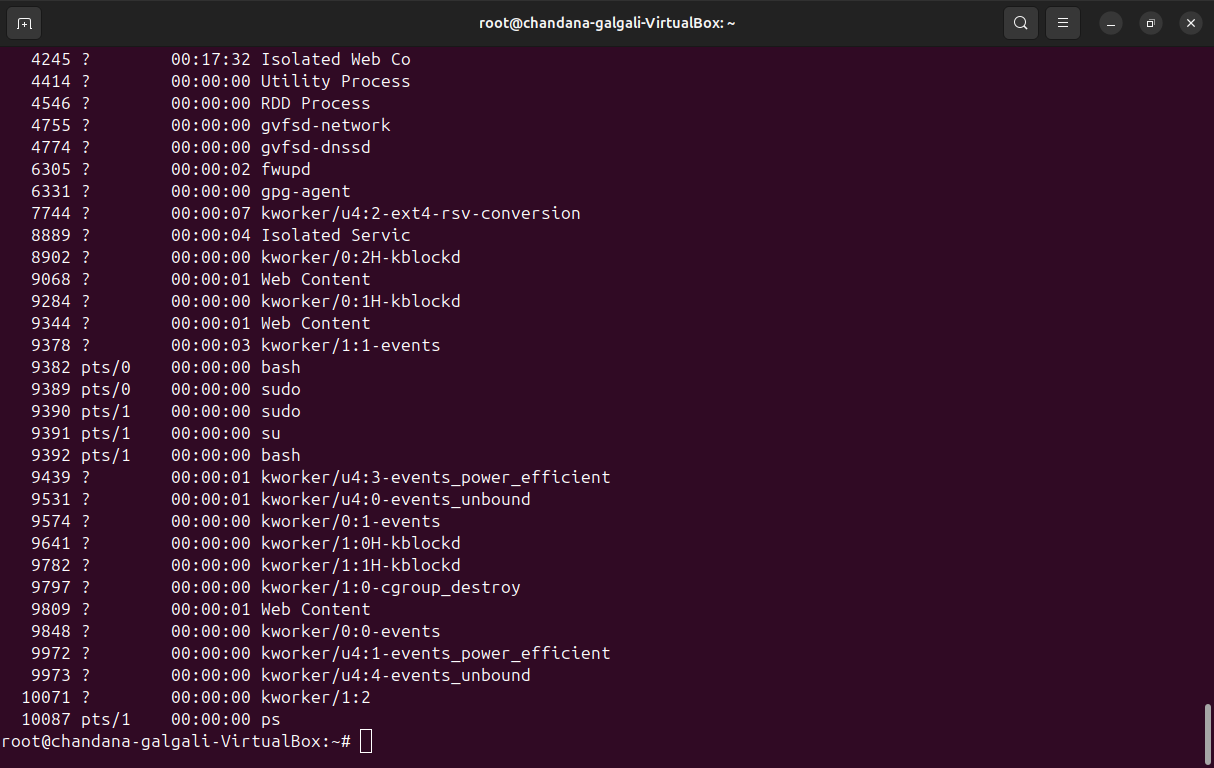
****

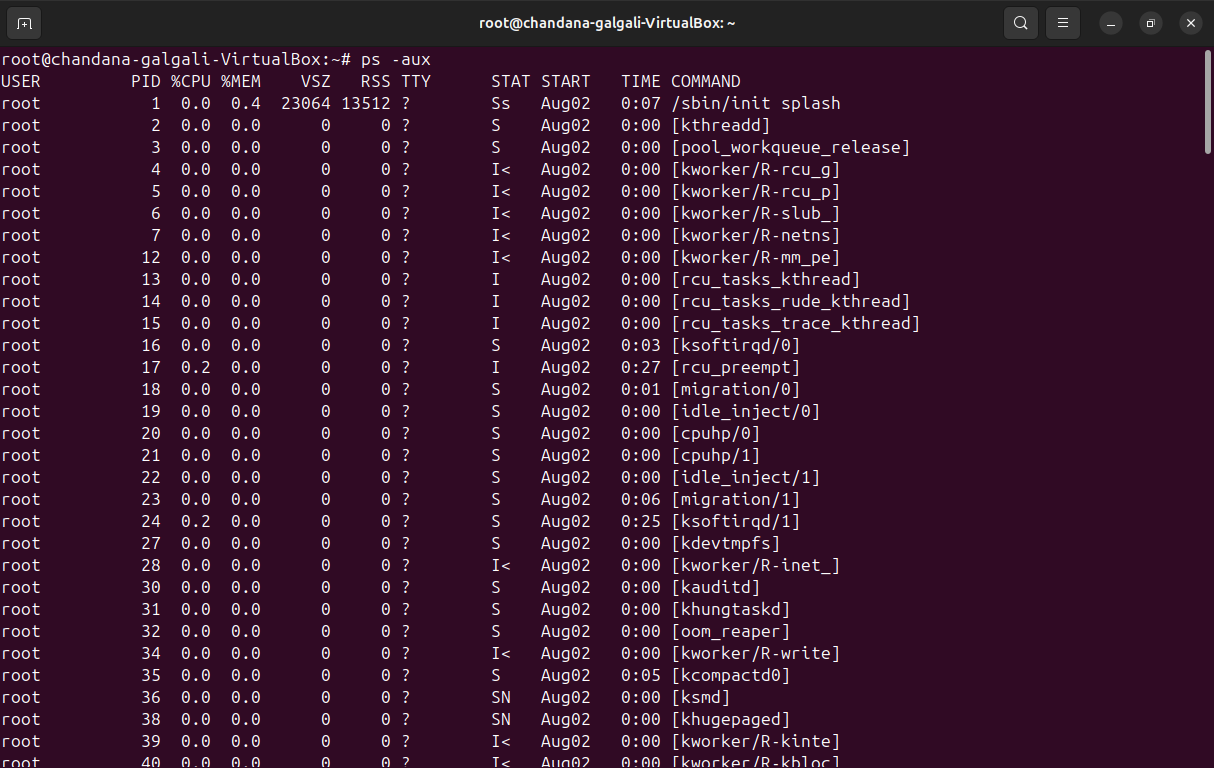
****

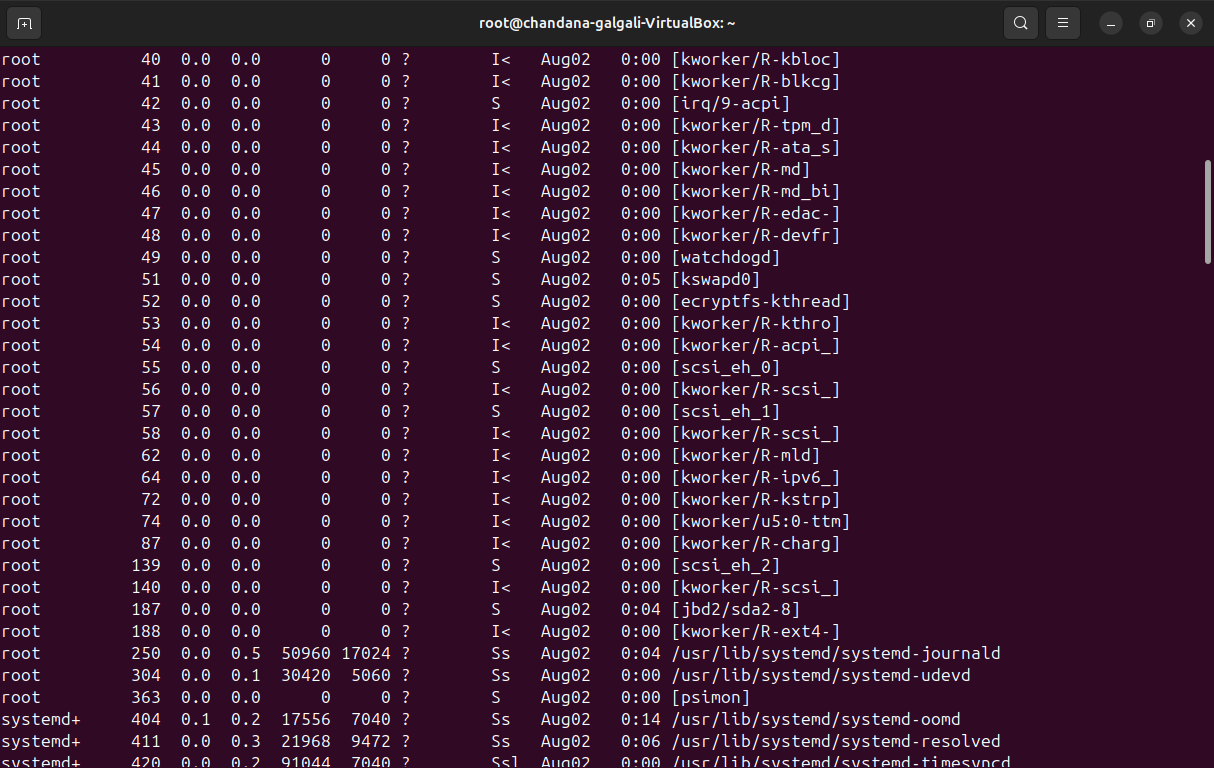
****

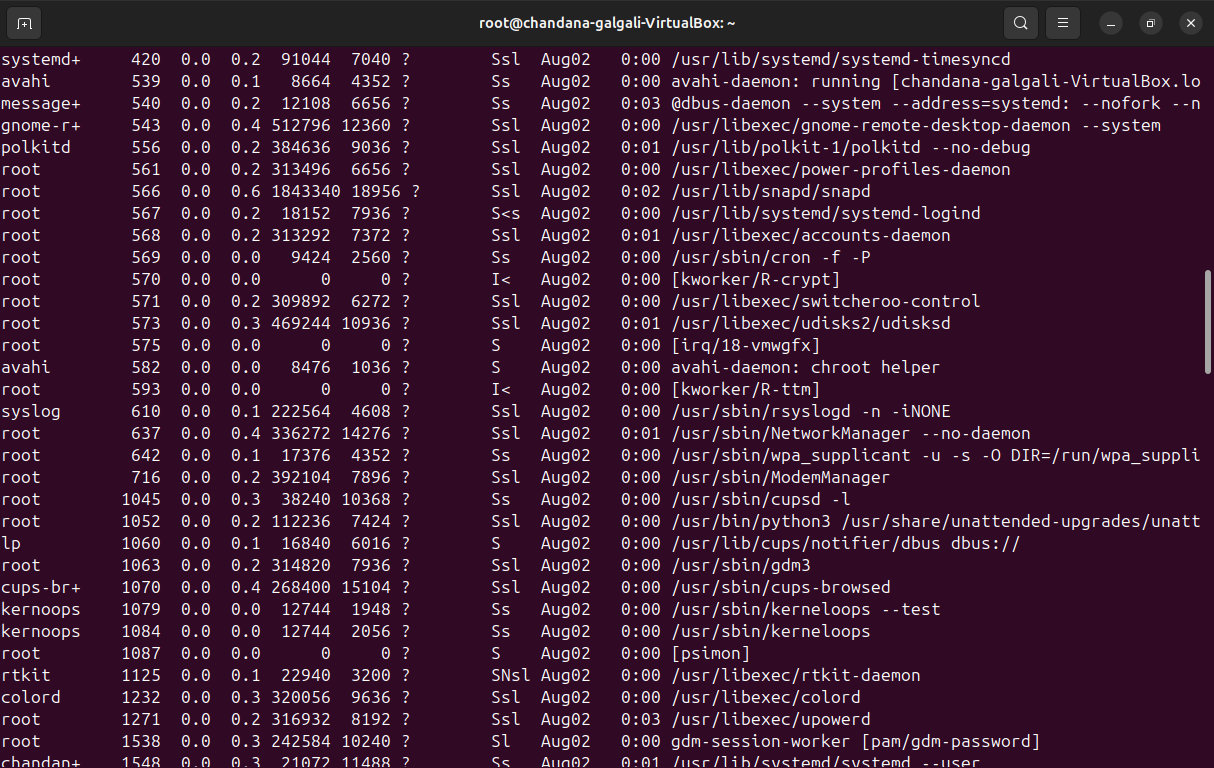
****

****

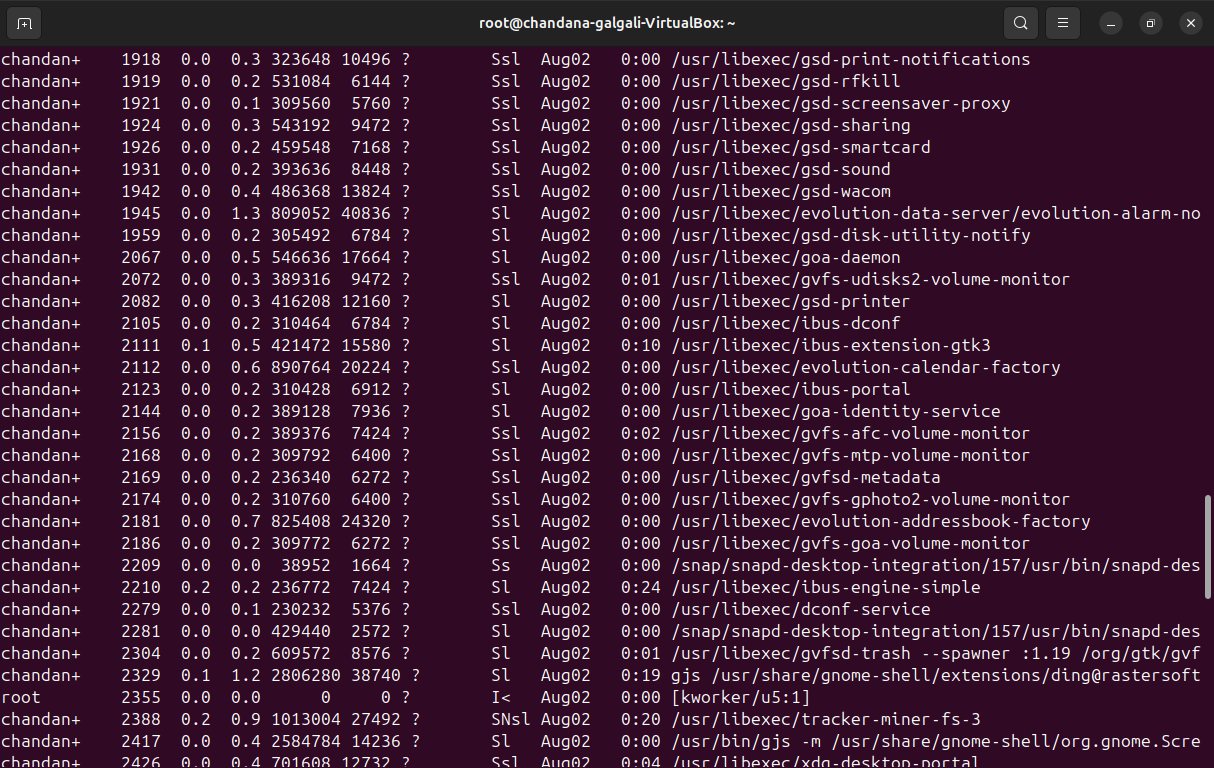
****

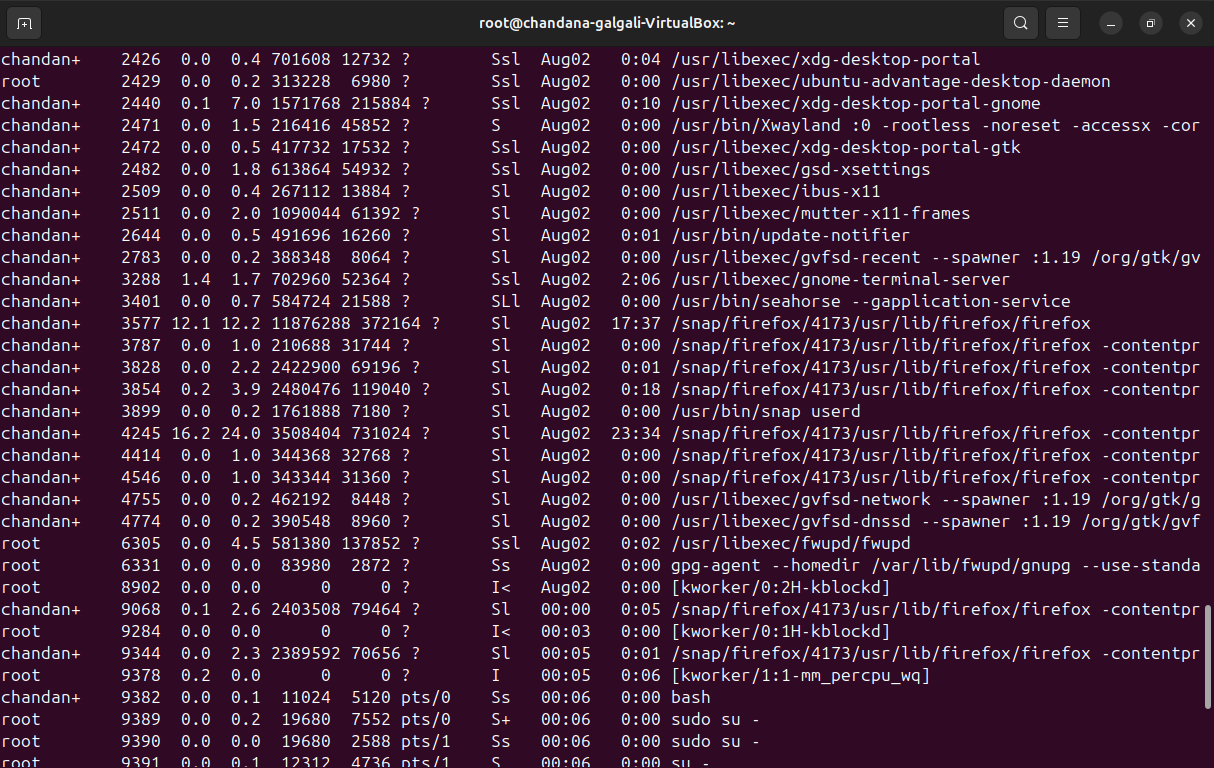
****

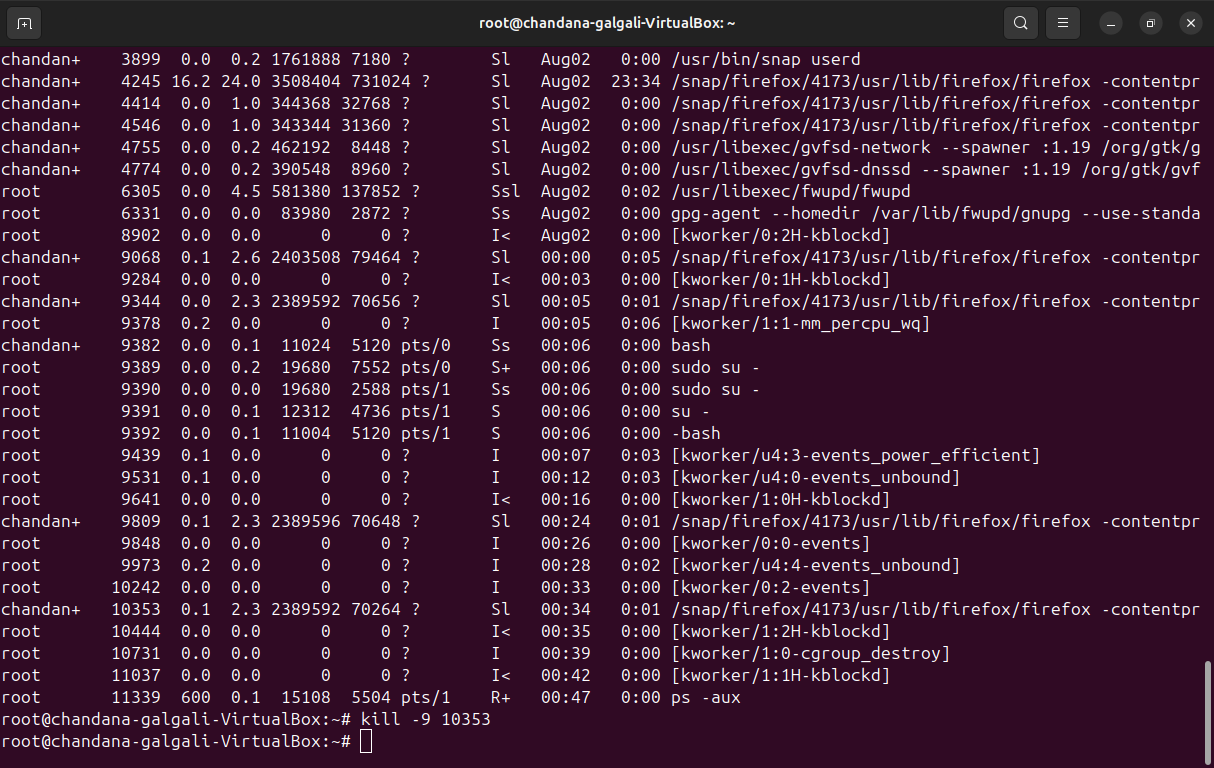
****

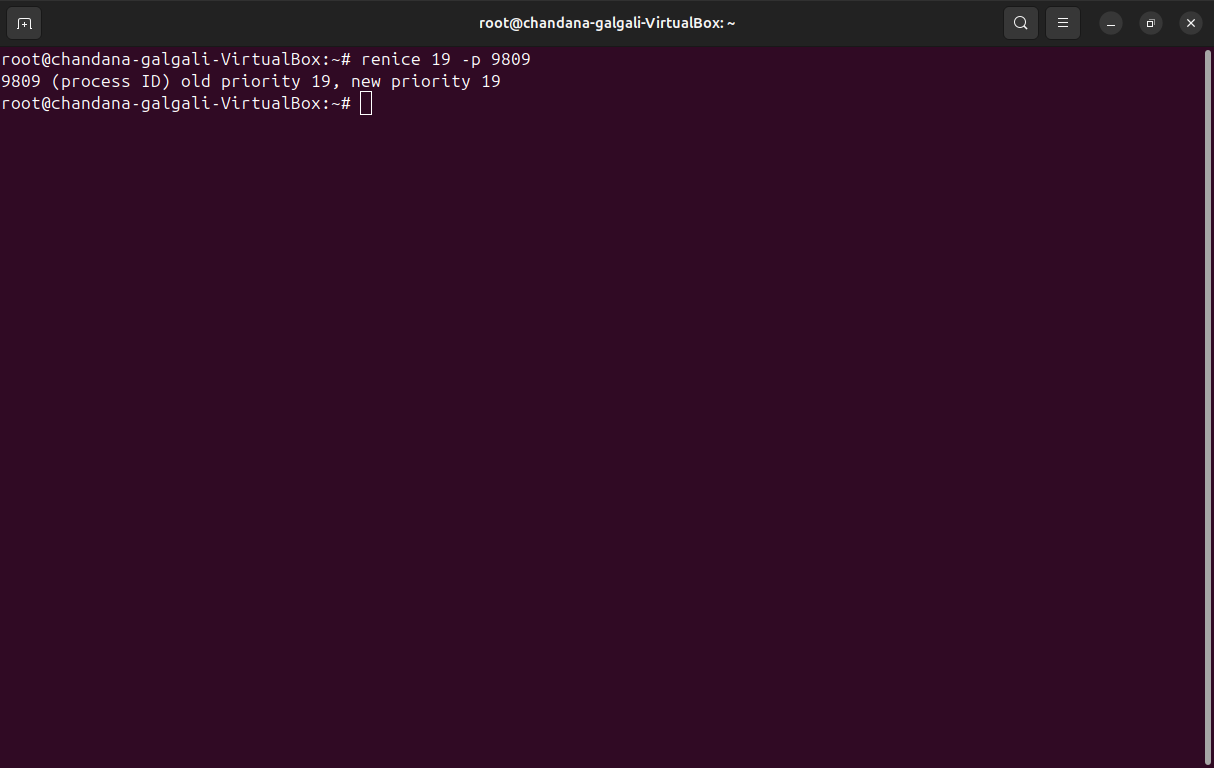
****

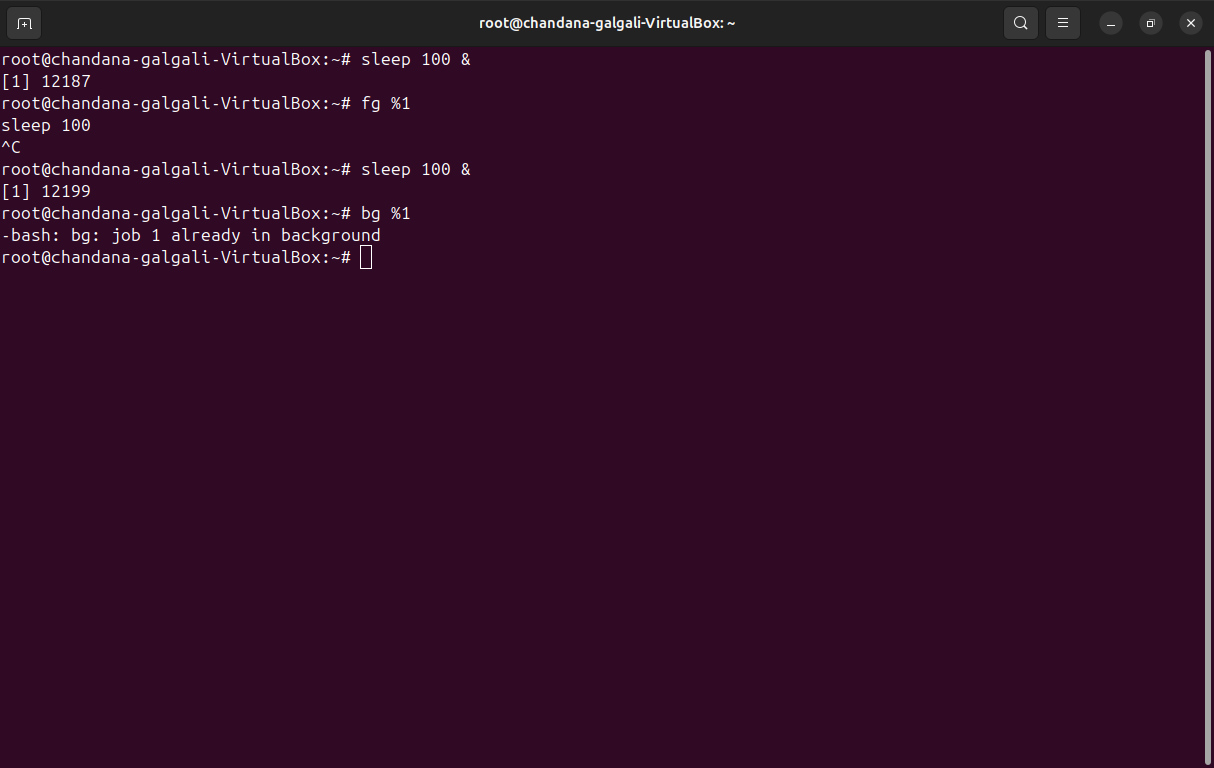
****

****

****

****

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:**

**CO1:** Understand basic structure of modern operating system

**CO4:** Demonstrate open source standards usage

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Conclusion:

The writeup provides an overview of the Linux file system, basic file handling commands, user administration, and process management. It emphasizes the hierarchical organization of directories and treating everything as a file. The practical exercises offer hands-on experience in managing files, directories, user accounts, and processes, providing foundational knowledge for effectively operating and administering Linux systems.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## 

## Grade: AA / AB / BB / BC / CC / CD /DD

## 

## 

## Signature of faculty in-charge with date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## References:

**Books/ Journals/ Websites:**

1. RichardBlumandChristineBresnahan, “LinuxCommandLine&ShellScripting”, IInd Edition, Wiley, 2012.
2. Guru99. (11 August 2020). File Permissions in Linux/Unix with Example. Retrieved from https://[www.guru99.com/file-Permissions.html](http://www.guru99.com/file-Permissions.html)

