We will be walking thourgh basic linux terminal commands

1. Navigation Commands

`pwd`– Print Working Directory

Shows the current location in the filesystem.

```
pwd
```

Output example:

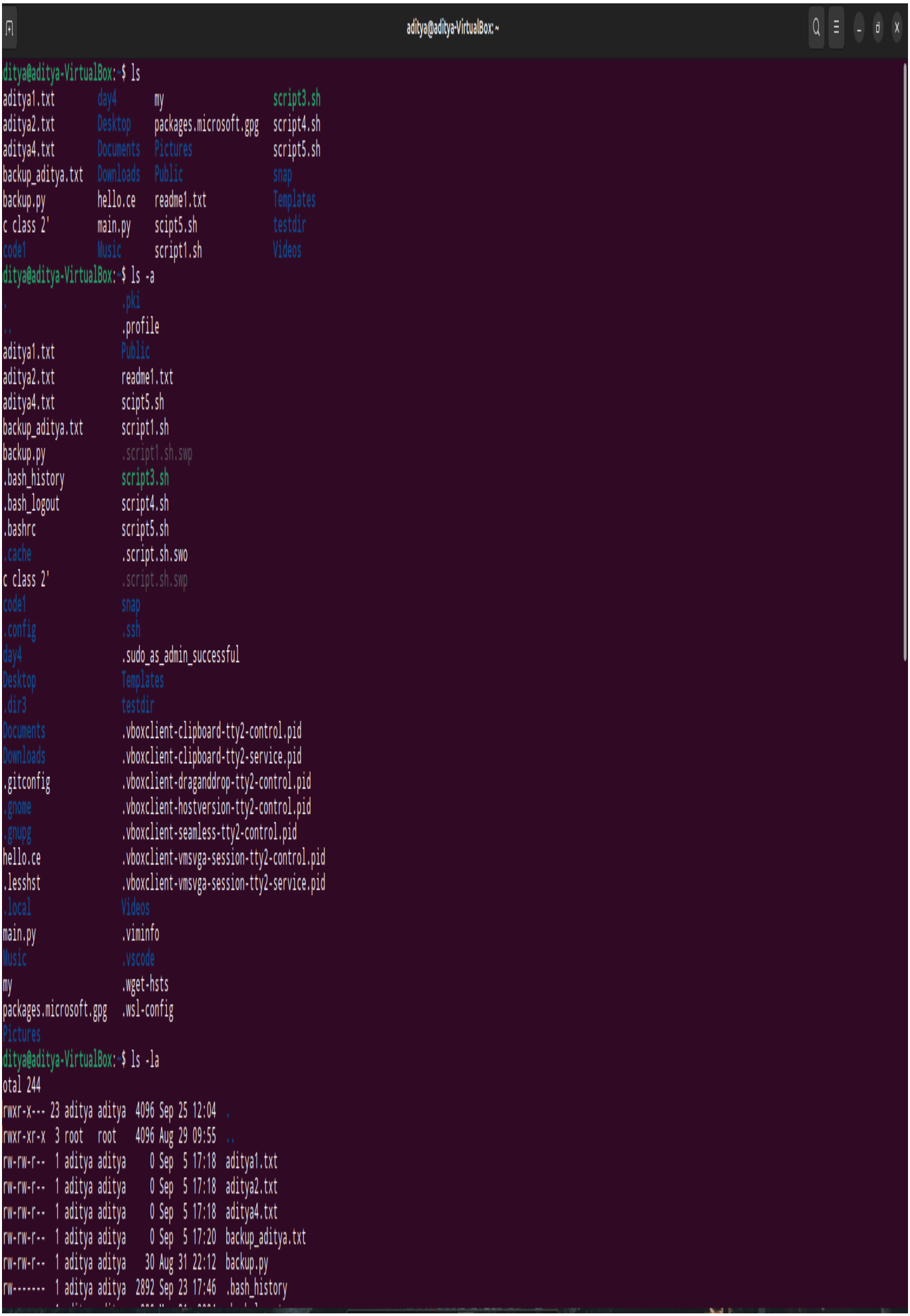
/home/aditya/Assignment

`ls` – List Directory Contents

The ls command is used to list files and directories in the current working directory. flag-a list down all the file and folder including the one which are hidden

```
ls -l → Detailed list (permissions, size, date)
ls -a → Shows hidden files (those starting with .)
ls -la → Combined
```

output is :


aditya@aditya-VirtualBox: ~
aditya@aditya-VirtualBox:~$ ls
aditya1.txt day4 my script3.sh
aditya2.txt Desktop packages.microsoft.gpg script4.sh
aditya4.txt Documents Pictures script5.sh
backup_aditya.txt Downloads Public snap
backup.py hello.ce readme1.txt Templates
c class 2' main.py scipt5.sh testdir
code1 Music script1.sh Videos
aditya@aditya-VirtualBox:~$ ls -a
. .pki
.. .profile
aditya1.txt Public
aditya2.txt readme1.txt
aditya4.txt scipt5.sh
backup_aditya.txt script1.sh
backup.py .script1.sh.swp
.bash_history script3.sh
.bash_logout script4.sh
.bashrc script5.sh
.cache .script.sh.swo
c class 2' .script.sh.swp
code1 snap
.config .ssh
day4 .sudo_as_admin_successful
Desktop Templates
.dir3 testdir
Documents .vboxclient-clipboard-tty2-control.pid
Downloads .vboxclient-clipboard-tty2-service.pid
.gitconfig .vboxclient-draganddrop-tty2-control.pid
.gnome .vboxclient-hostversion-tty2-control.pid
.gnupg .vboxclient-seamless-tty2-control.pid
hello.ce .vboxclient-vmsvga-session-tty2-control.pid
.lesshst .vboxclient-vmsvga-session-tty2-service.pid
.local Videos
main.py .viminfo
Music .vscode
my .wget-hsts
packages.microsoft.gpg .wsl-config
Pictures
aditya@aditya-VirtualBox:~$ ls -la
otal 244
rwxr-x--- 23 aditya aditya 4096 Sep 25 12:04 .
rwxr-xr-x 3 root root 4096 Aug 29 09:55 ..
rw-rw-r-- 1 aditya aditya 0 Sep 5 17:18 aditya1.txt
rw-rw-r-- 1 aditya aditya 0 Sep 5 17:18 aditya2.txt
rw-rw-r-- 1 aditya aditya 0 Sep 5 17:18 aditya4.txt
rw-rw-r-- 1 aditya aditya 0 Sep 5 17:20 backup_aditya.txt
rw-rw-r-- 1 aditya aditya 30 Aug 31 22:12 backup.py
rw------- 1 aditya aditya 2892 Sep 23 17:46 .bash_history

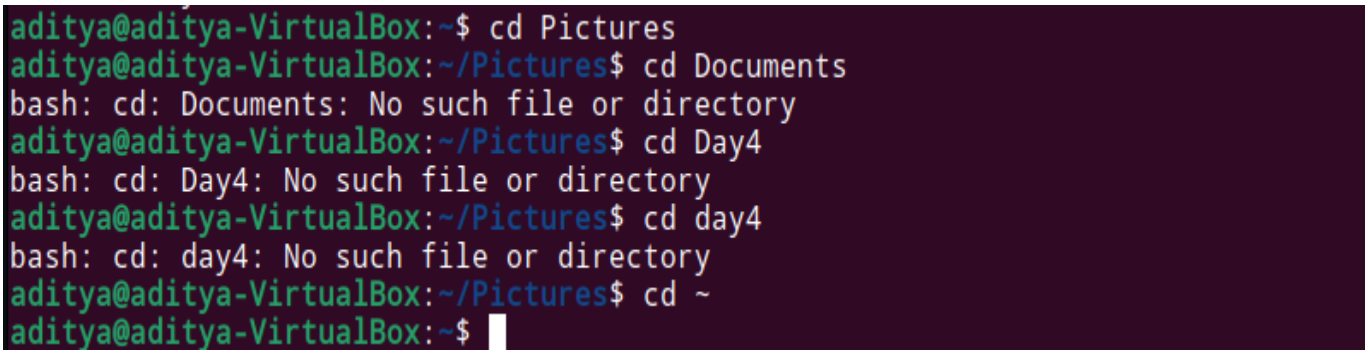
`cd` – Change Directory

Moves into a directory.

```
cd
```

Examples:

```
   cd Documents        # Go to Documents
   cd ..               # Go up one level
   cd /                # Go to root
   cd ~                # Go to home directory
```



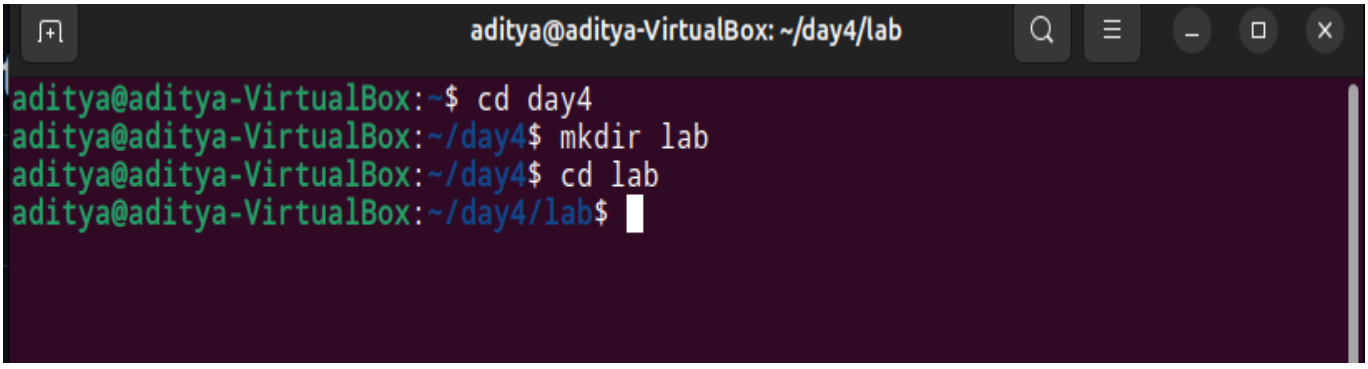
2. File and Directory Management

`mkdir` – Make Directory

Creates a new folder.

```
mkdir new_folder
```

output :

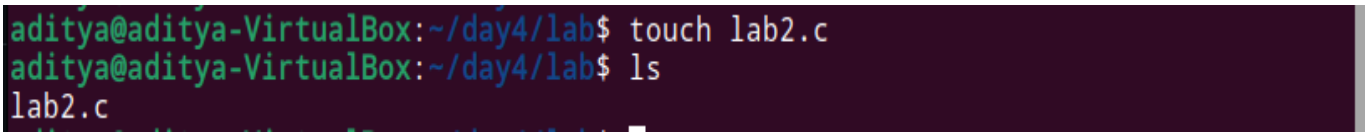


`touch`– Create File

Creates an empty file.

```
touch file.txt
```

output:



`cp` – Copy Files or Directories

```
cp source.txt destination.txt
```

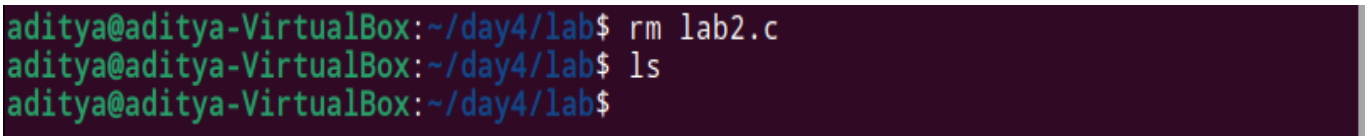
`mv` – Move or Rename Files

```
mv oldname.txt newname.txt
```

`rm` – Remove Files

```
rm file.txt          # Delete file
rm -r folder_name    # Delete folder (recursively)
```

Be careful! There is no undo.



3. File Viewing & Editing

`cat` – View File Contents

Displays content in terminal.

```
cat file.txt
```

Output:

```
aditya@aditya-VirtualBox:~/Pictures$ cat readme1.txt
hello i am aditya
```

nano – Edit Files in Terminal

A basic terminal-based text editor.

```
nano file.txt
```

```
Use arrows to move
CTRL + O to save
CTRL + X to exit
```

output :

```
aditya@aditya-VirtualBox:~/Pictures$ nano readme1.txt
```

clear – Clears the Terminal

```
clear
```

Shortcut: CTRL + L

4. **System Commands**

echo – Print Text

Useful for debugging or scripting.

```
echo "Hello, World!"
```

---

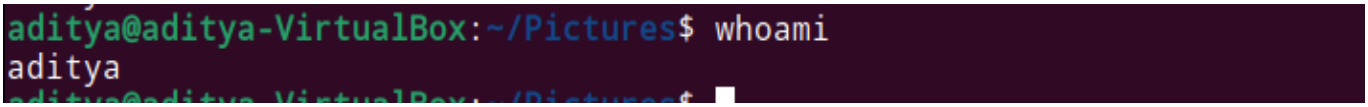
output :

```
aditya@aditya-VirtualBox:~/Pictures$ echo aditya
aditya
```

whoami – Show Current User

```
whoami
```

output :



`man` – Manual for Any Command

```
man ls
```

Use `q` to quit the manual.

---

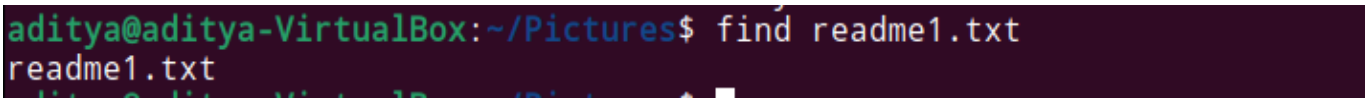
5. **Searching and Finding**

`find` – Locate Files

```
find . -name "*.txt"
```

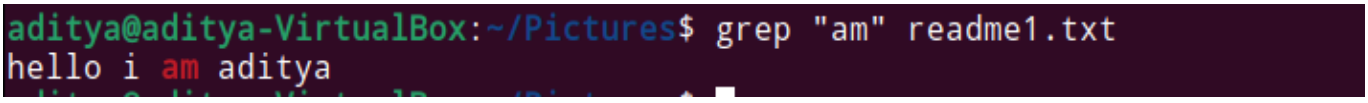
🔍 Finds all `.txt` files in current folder and subfolders.

output :



`grep` – Search Inside Files

```
grep "grep hello i mam aditya" file.txt
```



🔍 Searches for the word `hello` inside `file.txt`.

---

6. **Helpful Shortcuts**

Shortcut	Action
`Tab`	Auto-complete files/folders
`↑ / ↓`	Browse command history

Shortcut	Action
`CTRL + C`	Stop a running command
`CTRL + L`	Clear screen

---

7. **Bonus: Chaining Commands**

- **Run multiple commands**:

```
mkdir test && cd test && touch hello.txt
```

- **Run only if previous command succeeds**: `&&`
- **Run regardless of success**: `;`

Shell Tutorial – File Permissions with `chmod` and `chown`

🔹 1. Understanding File Permissions in Linux

Each file/directory in Linux has:

- **Owner** → The user who created the file.
- **Group** → A group of users who may share access.
- **Others** → Everyone else.

Permission Types

- **r** → Read (4 in numeric)
- **w** → Write (2 in numeric)
- **x** → Execute (1 in numeric)

Permission Layout

Example from `ls -l`:

```
-rwxr-xr--
```

Breakdown:

- `-` → Regular file (`d` = directory, `l` = symlink, etc.)
- `rwx` → Owner has read, write, execute
- `r-x` → Group has read, execute

- `r--` → Others have read only

output :

```
aditya@aditya-VirtualBox:~/Pictures$ ls -la
total 20
drwxr-xr-x  4 aditya aditya 4096 Sep 25 16:15 .
drwxr-x--- 24 aditya aditya 4096 Sep 25 12:56 ..
drwxrwxr-x  2 aditya aditya 4096 Sep 25 12:54 lab1
-rw-rw-r--  1 aditya aditya   18 Sep 25 16:15 readme1.txt
drwxrwxr-x  2 aditya aditya 4096 Sep 25 16:30 Screenshots
```

🔹 2. `chmod` – Change File Permissions

Syntax

```
chmod [options] mode filename
```

Modes can be set in **numeric (octal)** or **symbolic** form.

(A) Numeric (Octal) Method

Each permission is represented as a number:

- Read = 4
- Write = 2
- Execute = 1

Add them up:

- `7 = rwx`
- `6 = rw-`
- `5 = r-x`
- `4 = r--`
- `0 = ---`

Example:

```
chmod 777 script.sh
```

Meaning:

- Owner: 7 → `rwx`
- Group: 7 → `r-w-x`
- Others: 7 → `r-w-r`

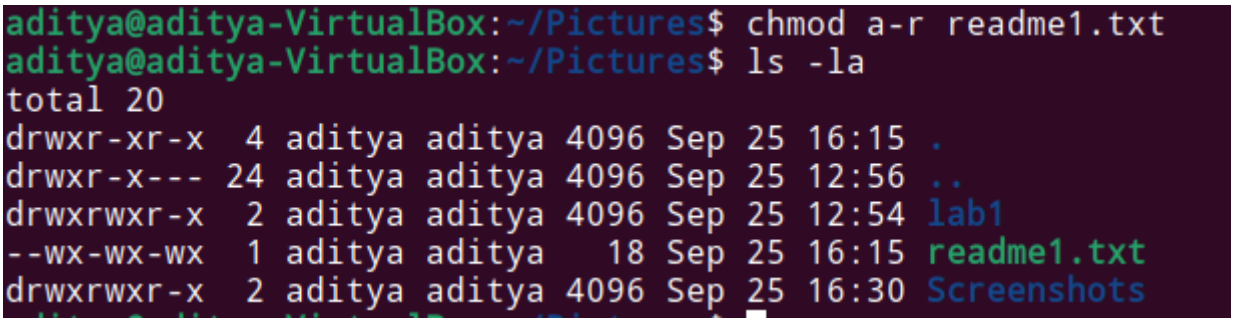
(B) Symbolic Method

Use u (user/owner), g (group), o (others), a (all). Operators:

- + → Add permission
- - → Remove permission
- = → Assign exact permission

Modes can be set in **numeric (octal)** or **symbolic** form.

output:



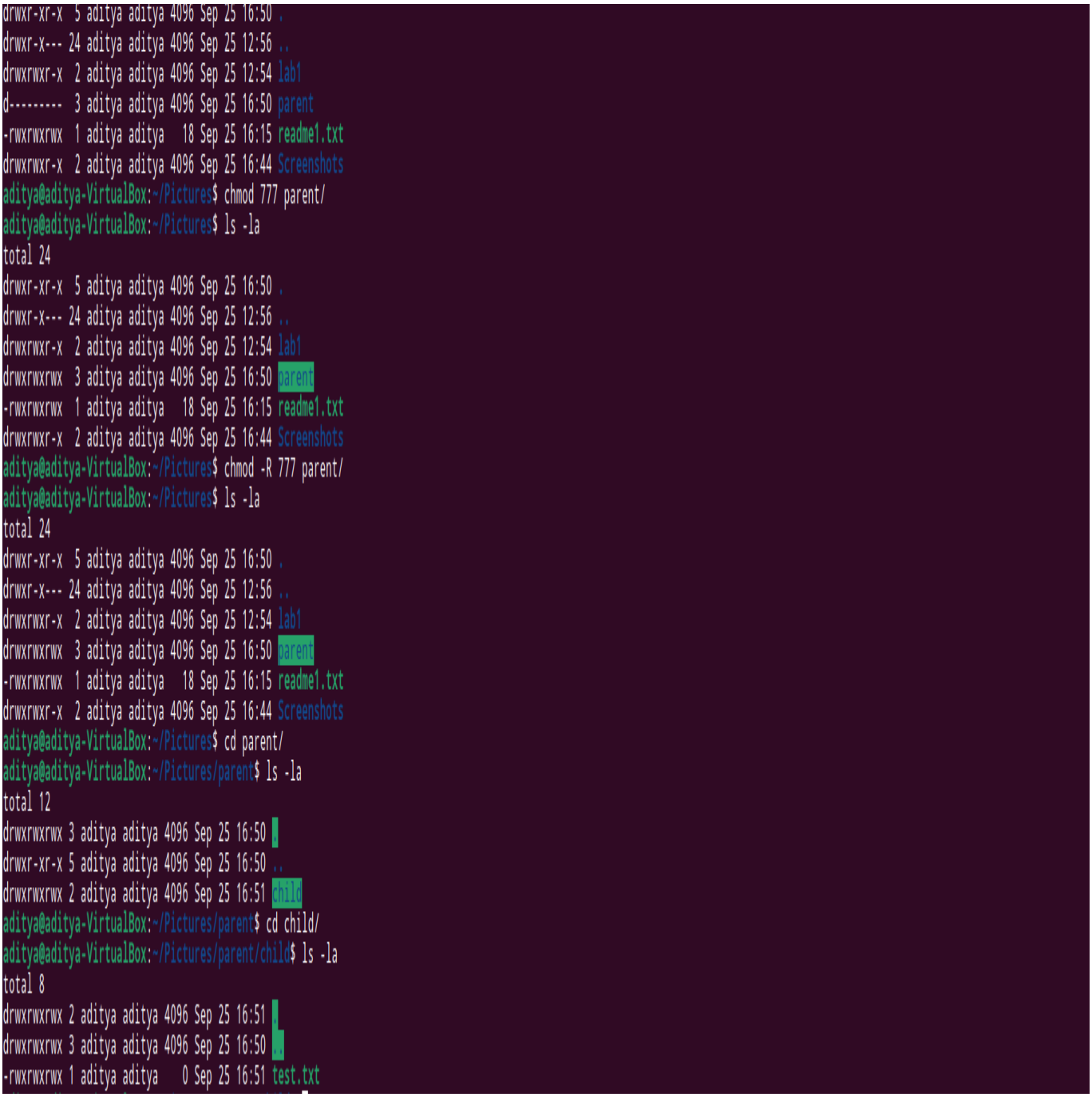
(C) Recursive Changes

```
chmod -R 755 /mydir
```

- -R → applies changes recursively to all files/subdirectories.

```
aditya@aditya-VirtualBox:~/Pictures$ chmod 777 script1.sh
chmod: cannot access 'script1.sh': No such file or directory
aditya@aditya-VirtualBox:~/Pictures$ chmod a-r file.txt
chmod: cannot access 'file.txt': No such file or directory
aditya@aditya-VirtualBox:~/Pictures$ chmod a-r readme1.txt
aditya@aditya-VirtualBox:~/Pictures$ ls -la
total 20
drwxr-xr-x  4 aditya aditya 4096 Sep 25 16:15 .
drwxr-x--- 24 aditya aditya 4096 Sep 25 12:56 ..
drwxrwxr-x  2 aditya aditya 4096 Sep 25 12:54 lab1
--wx-wx-wx  1 aditya aditya   18 Sep 25 16:15 readme1.txt
drwxrwxr-x  2 aditya aditya 4096 Sep 25 16:30 Screenshots
aditya@aditya-VirtualBox:~/Pictures$ chmod a-r readme1.txt
aditya@aditya-VirtualBox:~/Pictures$ ls-la
ls-la: command not found
aditya@aditya-VirtualBox:~/Pictures$ chmod 755 readme1,txt
chmod: cannot access 'readme1,txt': No such file or directory
aditya@aditya-VirtualBox:~/Pictures$ chmod 777 readme1.txt
aditya@aditya-VirtualBox:~/Pictures$ ls -la
total 20
drwxr-xr-x  4 aditya aditya 4096 Sep 25 16:15 .
drwxr-x--- 24 aditya aditya 4096 Sep 25 12:56 ..
drwxrwxr-x  2 aditya aditya 4096 Sep 25 12:54 lab1
-rwxrwxrwx  1 aditya aditya   18 Sep 25 16:15 readme1.txt
drwxrwxr-x  2 aditya aditya 4096 Sep 25 16:44 Screenshots
aditya@aditya-VirtualBox:~/Pictures$ mkdir parent
aditya@aditya-VirtualBox:~/Pictures$ cd parent
aditya@aditya-VirtualBox:~/Pictures/parent$ mkdir child
aditya@aditya-VirtualBox:~/Pictures/parent$ cd child/
aditya@aditya-VirtualBox:~/Pictures/parent/child$ touch test.txt
aditya@aditya-VirtualBox:~/Pictures/parent/child$ ls -la
total 8
drwxrwxr-x 2 aditya aditya 4096 Sep 25 16:51 .
drwxrwxr-x 3 aditya aditya 4096 Sep 25 16:50 ..
-rw-rw-r-- 1 aditya aditya    0 Sep 25 16:51 test.txt
aditya@aditya-VirtualBox:~/Pictures/parent/child$ chmod 000 test.txt
aditya@aditya-VirtualBox:~/Pictures/parent/child$ ls -la
total 8
drwxrwxr-x 2 aditya aditya 4096 Sep 25 16:51 .
drwxrwxr-x 3 aditya aditya 4096 Sep 25 16:50 ..
---------- 1 aditya aditya    0 Sep 25 16:51 test.txt
```

```
aditya@aditya-VirtualBox:~/Pictures/parent/child$ cd .
aditya@aditya-VirtualBox:~/Pictures/parent/child$ cd ..
aditya@aditya-VirtualBox:~/Pictures/parent$ chmod 000 child/
aditya@aditya-VirtualBox:~/Pictures/parent$ ls -la
total 12
drwxrwxr-x 3 aditya aditya 4096 Sep 25 16:50 .
drwxr-xr-x 5 aditya aditya 4096 Sep 25 16:50 ..
d--------- 2 aditya aditya 4096 Sep 25 16:51 child
aditya@aditya-VirtualBox:~/Pictures/parent$ cd ..
aditya@aditya-VirtualBox:~/Pictures$ chmod 000 parent/
aditya@aditya-VirtualBox:~/Pictures$ ls -la
total 24
```



---

3. chown – Change File Ownership

Syntax

```
chown [options] new_owner:new_group filename
```

Examples:

```
chown ashish.txt          # Change owner to user 'sameer'
chown ashish:dev.txt        # Change owner to 'sameer' and group to 'dev'
chown ashish:dev file.txt             # Change only group to 'dev'
chown -R ashish:dev /project     # Recursive ownership change
```

Practice Experiment on chown

- 1. Create a new user

```
sudo useradd -m newuser
```

  - `-m` → creates a home directory `/home/newuser`.

- 2. Create a new group

```
sudo groupadd newgroup
```

- 3. Add the user to the group

```
sudo usermod -aG newgroup newuser
```

  - `-aG` → append user to the supplementary group (doesn't remove existing groups).

- 4. Create a file (as current user, e.g. root or your login user)

```
touch testfile.txt
```

Check ownership:

```
ls -l testfile.txt
```

Example:

```
-rw-rw-r--1 sameerchoudhary sameerchoudhary 0 Aug 20 18:52 testfile.txt
```

- 5. Assign ownership of the file to `newuser` and `newgroup`

```
sudo chown newuser:newgroup testfile.txt
```

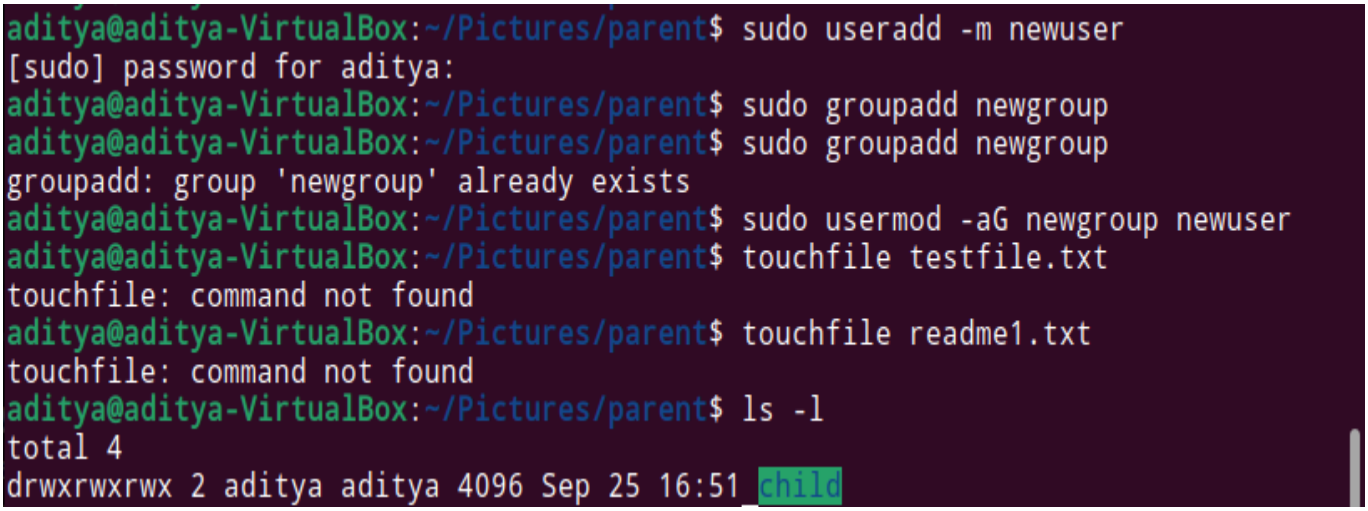
- 6. Verify ownership

```
ls -l testfile.txt
```

Output:

```
-rw-rw-r--1 newuser newgroup 0 Aug 20 18:52 testfile.txt
```

✅ **Key Tip**: Use **numeric for quick settings** (e.g., 755, 644) and **symbolic for fine adjustments** (`u+x`, `g-w`).



- 4. Putting It All Together

Example Scenario

```
touch project.sh
ls -l project.sh
```

Output:

```
-rw-r--r-- 1 sameer dev 0 Aug 19 12:00 project.sh
```

Now:

```
chmod 700 project.sh       # Only owner has rwx
chmod u+x,g-w project.sh   # Add execute for user, remove write for group
chown root:admin project.sh # Change owner to root and group to admin
```

🔹 5. Quick Reference Table

Numeric	Permission	Meaning
0	---	No access
1	--x	Execute only
2	-w-	Write only
3	-wx	Write + Exec
4	r--	Read only
5	r-x	Read + Exec
6	rw-	Read + Write
7	rwx	Full access

Q1 what is the difference between chmod and chown?

```
ANS=chown-change ownership
     change the owner and group of a file or directory

    chmod-change permissions
     changes the permissions for the owner,group, and others
```

📌 Q2 how do you check current directory and user?

```
  ANS = by using the pwd command to check current directory and by using
the whoami to check current user.
```