LINUX BASIC COMMANDS

1. uname - The uname command is used to print system information such as the name of the operating system.

```
subbu@Ubuntu:~$ uname
Linux
subbu@Ubuntu:~$
```

2. man – It gives manual pages or help page regaridng particular command usage which you want to know.

```
NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...
```

3. whoami - It will give us which user is currently logged in.

```
subbu@Ubuntu:~$ whoami
subbu
subbu@Ubuntu:~$
```

4. pwd – It will print working directory.

```
subbu@Ubuntu:~$ pwd
/home/subbu
subbu@Ubuntu:~$
```

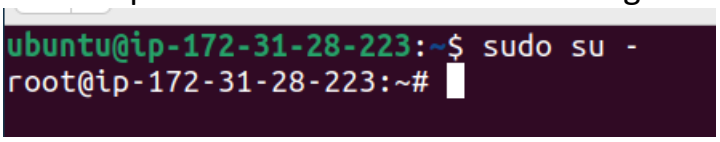
5. history - it will display the previous commands.

```
ubuntu@ip-172-31-28-223:~$ whoami
ubuntu
ubuntu@ip-172-31-28-223:~$ users
ubuntu
ubuntu@ip-172-31-28-223:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-28-223:~$ ls -l
total 0
ubuntu@ip-172-31-28-223:~$ uname
Linux
ubuntu@ip-172-31-28-223:~$ history
    1  clear
    2  whoami
    3  users
    4  pwd
    5  ls -l
    6  uname
    7  history
ubuntu@ip-172-31-28-223:~$
```

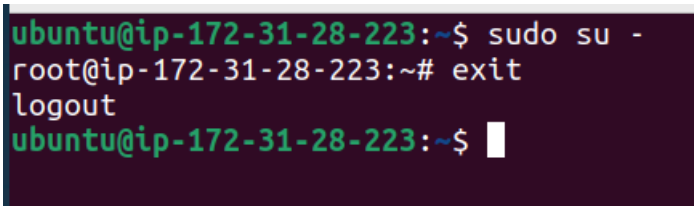
To delete particular history record

```
ubuntu@ip-172-31-28-223:~$ history
    1  clear
    2  whoami
    3  users
    4  pwd
    5  ls -l
    6  uname
    7  history
ubuntu@ip-172-31-28-223:~$ history -d 7
ubuntu@ip-172-31-28-223:~$ history
    1  clear
    2  whoami
    3  users
    4  pwd
    5  ls -l
    6  uname
    7  history -d 7
    8  history
ubuntu@ip-172-31-28-223:~$
```

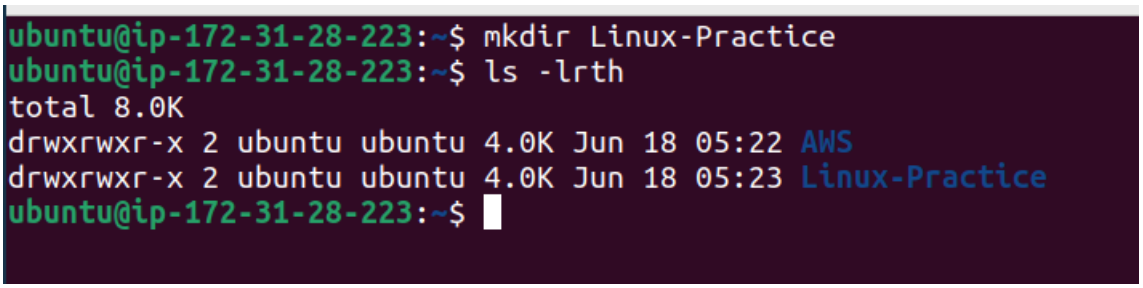
6. sudo su - -- Normal User to Root user. (sudo command is very important) # symbol and $ for normal.
sudo: Stands for "superuser do" and allows a permitted user to execute a command as the superuser.
su: Stands for "substitute user" and is used to switch the current user to another user.
-: This option is used with su to start a login shell as the target user.



7. exit - root user to normal user.

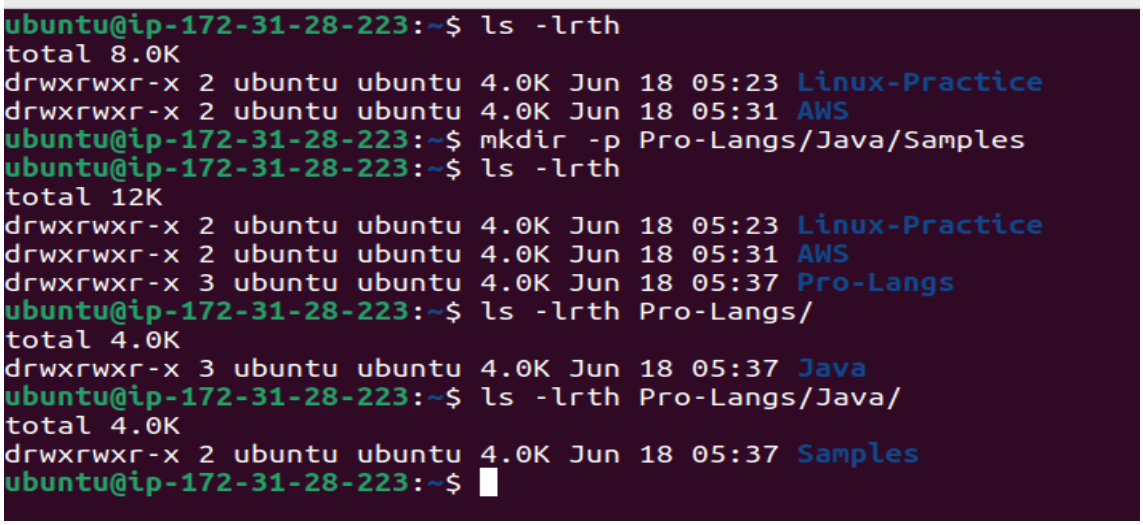


8. mkdir - To make directory.



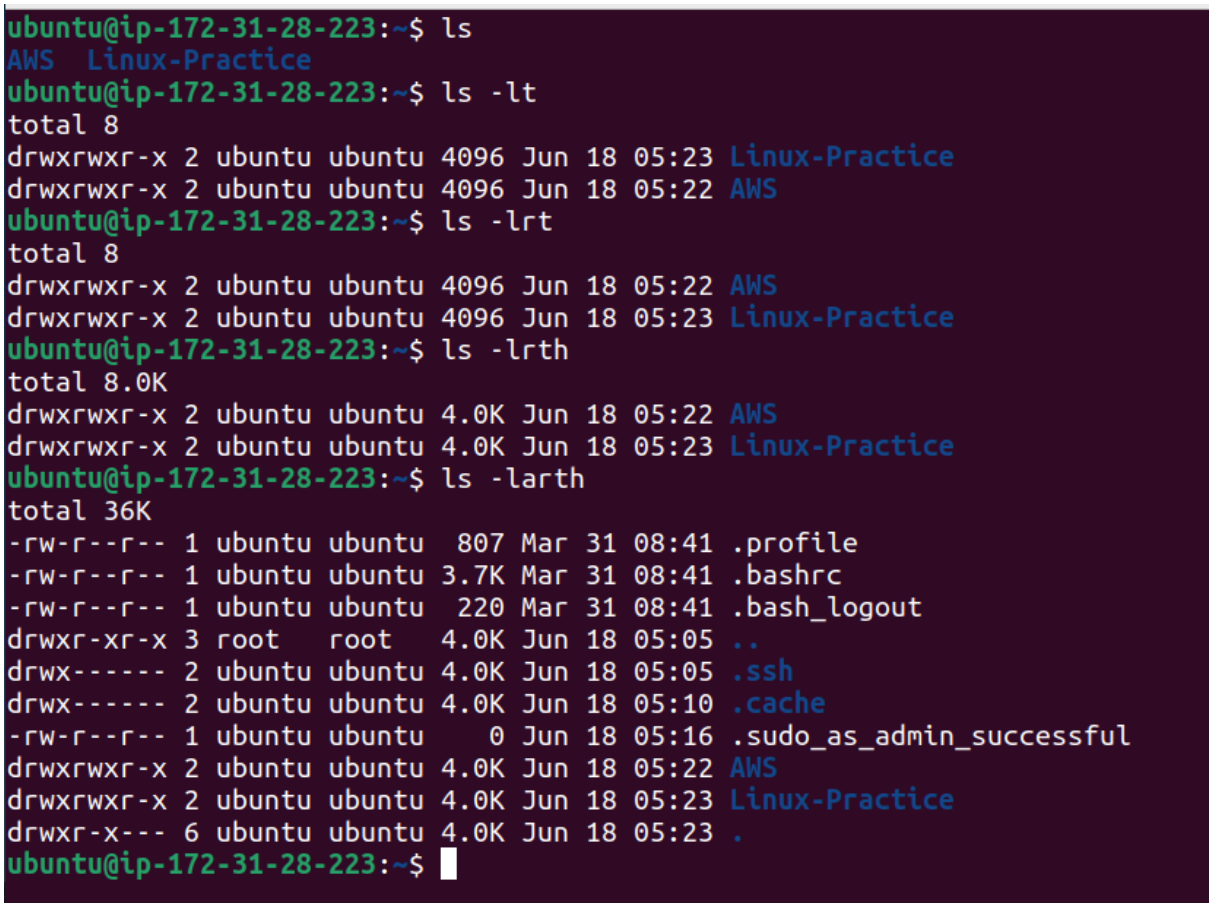
mkdir -p /home/user/newfolder/subfolder:

The mkdir -p command is used to create directories, including any necessary parent directories that do not already exist.



9. ls – To list the directories and files in the current working directory or other directories using complete path.

- ls -l: Lists files in long format, showing details such as permissions, number of links, owner, group, size, and modification date.
- ls -a: Lists all files, including hidden files (those starting with a dot .).
- ls -la or ls -al: Combines the options to list all files in long format.
- ls -h: With -l, it prints sizes in human-readable format (e.g., 1K, 234M, 2G).
- ls -t: Sorts files by modification time, most recent first.
- ls -r: Reverses the order of the sort.
- ls -S: Sorts files by size, largest first.



If you run ls in a specific directory, you can include the directory path
ls /path/to/directory - **Ex: ls -lrth AWS/**

```
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 8.0K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
ubuntu@ip-172-31-28-223:~$ ls -lrth AWS/
total 0
-rw-rw-r-- 1 ubuntu ubuntu 0 Jun 18 05:31 sample.txt
ubuntu@ip-172-31-28-223:~$
```

To get the inode number of each file and directory presented in the current working directory: ls -li

```
ubuntu@ip-172-31-28-223:~$ ls -lirth
total 36K
262392 drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
262391 drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
262394 drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
262432 drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
262438 -rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
262471 -rw-rw-r-- 1 ubuntu ubuntu  300 Jun 18 06:30 newfile.txt
262472 -rw-rw-r-- 1 ubuntu ubuntu   52 Jun 18 06:34 file.py
262473 -rw-rw-r-- 1 ubuntu ubuntu  344 Jun 18 06:48 samples.txt.save
262435 -rw-rw-r-- 1 ubuntu ubuntu 3.4K Jun 18 07:23 samples.txt
ubuntu@ip-172-31-28-223:~$
```

-v option with a command: It gives you the information about the action was done by that command.

The -v option stands for "verbose" and is used with many commands to provide detailed information about the actions being performed by the command.
When you use the -v option, the command typically outputs more information to the terminal, helping you understand exactly what is happening.

```
ubuntu@ip-172-31-28-223:~$ mkdir -v Programs
mkdir: created directory 'Programs'
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 16K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
ubuntu@ip-172-31-28-223:~$
```

10. cd:

The cd command is used to change the current working directory.

10.1. Change to a specific directory: cd /path/to/directory

10.2. Change to the home directory: cd # or cd ~

10.3. Change to the previous directory: cd -

10.4. Change to a subdirectory: cd subdirectory

10.5. Change to the parent directory: cd ..

```
ubuntu@ip-172-31-28-223:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 16K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
ubuntu@ip-172-31-28-223:~$ cd Pro-Langs/
ubuntu@ip-172-31-28-223:~/Pro-Langs$ cd Java/
ubuntu@ip-172-31-28-223:~/Pro-Langs/Java$ cd Samples
ubuntu@ip-172-31-28-223:~/Pro-Langs/Java/Samples$ cd ..
ubuntu@ip-172-31-28-223:~/Pro-Langs/Java$ cd -
/home/ubuntu/Pro-Langs/Java/Samples
ubuntu@ip-172-31-28-223:~/Pro-Langs/Java/Samples$ cd ../..
ubuntu@ip-172-31-28-223:~/Pro-Langs$ cd ~
ubuntu@ip-172-31-28-223:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-28-223:~$
```

11. touch : This command is help us to create empty file.

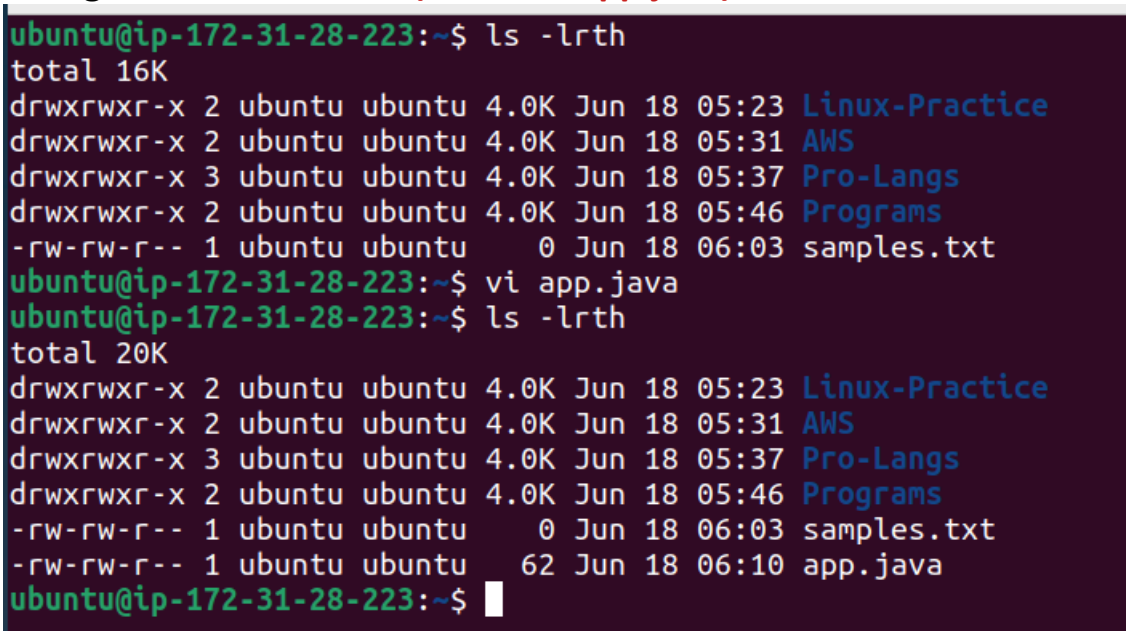
Ex: touch samples.txt

```
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 16K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
ubuntu@ip-172-31-28-223:~$ touch samples.txt
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 16K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 18 06:03 samples.txt
ubuntu@ip-172-31-28-223:~$
```

12. vi :

vi is a text editor available on Unix-like operating systems. It's a powerful tool used for creating and editing text files.

Opening a File: vi filename (Ex----> vi app.java)



Modes:

- **Normal Mode:** The default mode for navigation and issuing commands.
- **Insert Mode:** Used for inserting text. Enter this mode by pressing i.
- **Visual Mode:** Used for selecting text.
- **Command Mode:** Accessed by pressing : from Normal Mode, used for issuing commands such as saving or quitting.
- **Switching Between Modes:**
- **Normal to Insert Mode:** Press i (insert before cursor), a (append after cursor), o (open a new line below).
- **Insert to Normal Mode:** Press Esc.
- **Normal to Command Mode:** Press :.

Example Workflow

1. **Open a file:** vi example.txt
2. **Enter Insert Mode:** Press i and start typing.
3. **Return to Normal Mode:** Press Esc.

4. **Save the file and exit: Press : to enter Command Mode, then type wq and press Enter.**

In Command Mode:

- **Save and Quit:**
  - :w (write/save)
  - :q (quit)
  - :wq or ZZ (save and quit)
  - :q! (quit without saving)

In Normal Mode:

- **Move the Cursor:**
  - h (left), j (down), k (up), l (right)
  - w (next word), b (previous word)
  - 0 (beginning of line), $ (end of line)
  - G (end of file), gg (beginning of file)
- **Delete Text:**
  - x (delete character)
  - dd (delete line)
  - dw (delete word)
- **Undo/Redo:**
  - u (undo)
  - Ctrl-r (redo)
- **Copy and Paste:**
  - yy (yank/copy line)
  - p (paste after cursor)
  - P (paste before cursor)
- **Search:**
  - /pattern (search forward for pattern)
  - ?pattern (search backward for pattern)
  - n (repeat search in the same direction)
  - N (repeat search in the opposite direction)

13. cat :

The cat command is used to concatenate and display the content of files.

Basic Usage:

1. **Display the Content of a File: cat filename**

Ex-> cat app.java

```
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 20K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 18 06:03 samples.txt
-rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
ubuntu@ip-172-31-28-223:~$ cat app.java
THis is a file created by vi editor
It is a java file
ubuntu@ip-172-31-28-223:~$
```

2. Concatenate Multiple Files: cat file1 file2

```
ubuntu@ip-172-31-28-223:~$ cat app.java
THis is a file created by vi editor
It is a java file
ubuntu@ip-172-31-28-223:~$ cat samples.txt
Notes
Path Types: The path can be absolute (starting from the root directory, /) or relative (starting
Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
Previous Directory: cd - switches to the last directory you were in, which is useful for quickly
ubuntu@ip-172-31-28-223:~$ cat app.java samples.txt
THis is a file created by vi editor
It is a java file
Notes
Path Types: The path can be absolute (starting from the root directory, /) or relative (starting
Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
Previous Directory: cd - switches to the last directory you were in, which is useful for quickly
ubuntu@ip-172-31-28-223:~$
```

Ex: cat app.java samples.txt

3. Redirect Output to a New File: cat file1 file2 > newfile

Ex: cat app.java samples.txt > newfile.txt

```
ubuntu@ip-172-31-28-223:~$ cat app.java
THis is a file created by vi editor
It is a java file
ubuntu@ip-172-31-28-223:~$ cat samples.txt
Notes
Path Types: The path can be absolute (starting from the root directory.
Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
Previous Directory: cd - switches to the last directory you were in directories.
ubuntu@ip-172-31-28-223:~$ cat app.java samples.txt > newfile.txt
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 28K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
-rw-rw-r-- 1 ubuntu ubuntu  246 Jun 18 06:29 samples.txt
-rw-rw-r-- 1 ubuntu ubuntu  300 Jun 18 06:30 newfile.txt
ubuntu@ip-172-31-28-223:~$ cat newfile.txt
THis is a file created by vi editor
It is a java file
Notes
Path Types: The path can be absolute (starting from the root directory.
Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
Previous Directory: cd - switches to the last directory you were in directories.
ubuntu@ip-172-31-28-223:~$
```

4. Creating a file using cat command:

cat > filename
Enter the data....
Ctrl+d--->to save the file with data

```
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 28K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
-rw-rw-r-- 1 ubuntu ubuntu  246 Jun 18 06:29 samples.txt
-rw-rw-r-- 1 ubuntu ubuntu  300 Jun 18 06:30 newfile.txt
ubuntu@ip-172-31-28-223:~$ cat > file.py
This is a python program file.
print("hello world")
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 32K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
-rw-rw-r-- 1 ubuntu ubuntu  246 Jun 18 06:29 samples.txt
-rw-rw-r-- 1 ubuntu ubuntu  300 Jun 18 06:30 newfile.txt
-rw-rw-r-- 1 ubuntu ubuntu   52 Jun 18 06:34 file.py
ubuntu@ip-172-31-28-223:~$ cat file.py
This is a python program file.
print("hello world")
ubuntu@ip-172-31-28-223:~$
```

5.Append to a file from standard input:
cat >> existingfile
 # Type your content and press Ctrl+D to save and exit

```
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Notes
     2  Path Types: The path can be absolute (starting from the root directory.
     3  Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
     4  Previous Directory: cd - switches to the last directory you were in directories.
ubuntu@ip-172-31-28-223:~$ cat >> samples.txt
This is a 5th line i am adding to the existing data using cat >> command.
you can do more.......
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Notes
     2  Path Types: The path can be absolute (starting from the root directory.
     3  Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
     4  Previous Directory: cd - switches to the last directory you were in directories.
     5  This is a 5th line i am adding to the existing data using cat >> command.
     6  you can do more.......
ubuntu@ip-172-31-28-223:~$
```

6. Number All Output Lines: cat -n filename

Ex: cat -n samples.txt

```
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Notes
     2  Path Types: The path can be absolute (starting from the root directory.
     3  Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
     4  Previous Directory: cd - switches to the last directory you were in directories.
ubuntu@ip-172-31-28-223:~$
```

7. . Number non-empty output lines: cat -b filename

Ex: cat -b samples.txt

```
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Notes
     2
     3  Path Types: The path can be absolute (starting from the root directory.
     4
     5  Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.
     6
     7  Previous Directory: cd - switches to the last directory you were in directories.
     8
     9  This is a 5th line i am adding to the existing data using cat >> command.
    10
    11  you can do more.......
ubuntu@ip-172-31-28-223:~$ cat -b samples.txt
     1  Notes

     2  Path Types: The path can be absolute (starting from the root directory.

     3  Home Directory Shortcut: Typing cd alone or cd ~ will take you to your home directory.

     4  Previous Directory: cd - switches to the last directory you were in directories.

     5  This is a 5th line i am adding to the existing data using cat >> command.

     6  you can do more.......
ubuntu@ip-172-31-28-223:~$
```

Combining with Other Commands:

8. Using cat with grep to search within a file:

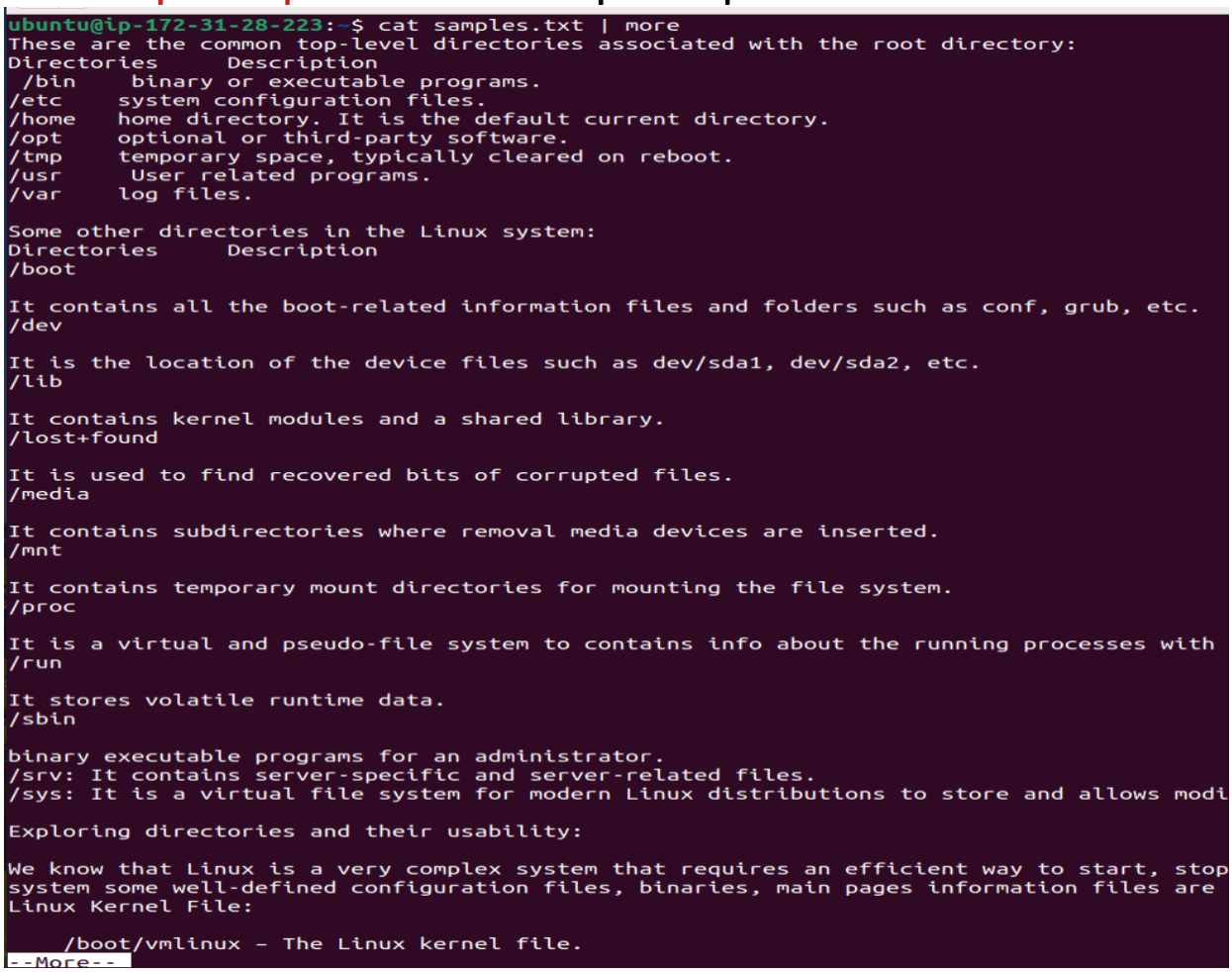
cat myfile.txt | grep "search_term"

Ex: cat app.java | grep “java”

```
ubuntu@ip-172-31-28-223:~$ cat app.java
THis is a file created by vi editor
It is a java file
ubuntu@ip-172-31-28-223:~$ cat app.java  | grep "java"
It is a java file
ubuntu@ip-172-31-28-223:~$
```

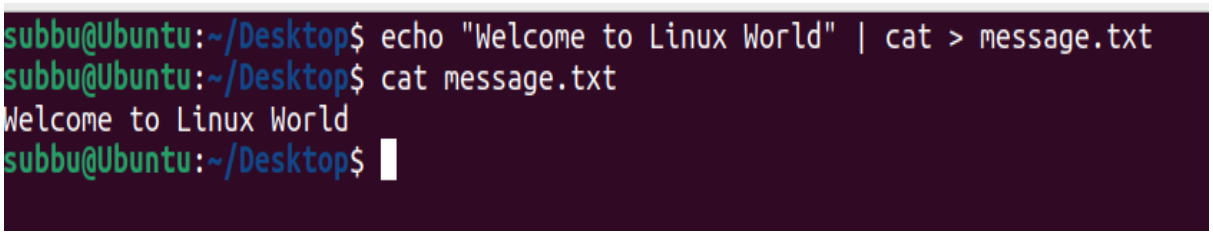
9. Using cat with more or less for paginated viewing:

cat samples.txt | more or cat samples.txt | less



10. Using cat to create a new file with redirected output:

echo "Some content" | cat > newfile.txt



11. To delete the content of a file without deleting the file:

> filename Ex: > app.java

```
ubuntu@ip-172-31-28-223:~$ ls -lrth
total 32K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:23 Linux-Practice
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:31 AWS
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 18 05:37 Pro-Langs
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 18 05:46 Programs
-rw-rw-r-- 1 ubuntu ubuntu   54 Jun 18 06:15 app.java
-rw-rw-r-- 1 ubuntu ubuntu  300 Jun 18 06:30 newfile.txt
-rw-rw-r-- 1 ubuntu ubuntu   52 Jun 18 06:34 file.py
-rw-rw-r-- 1 ubuntu ubuntu  344 Jun 18 06:48 samples.txt.save
ubuntu@ip-172-31-28-223:~$ cat app.java
THis is a file created by vi editor
It is a java file
ubuntu@ip-172-31-28-223:~$ > app.java
ubuntu@ip-172-31-28-223:~$ cat app.java
ubuntu@ip-172-31-28-223:~$
```

14. more , less commands:

The more and less commands are both used for viewing text files in Unix-like operating systems, but they have some differences in functionality and usability.

Baisc usage:

more filename

less filename

Difference between more and less commands in Linux:

Key Differences:

1. **Navigation:**
   - more only allows forward navigation, whereas less allows both forward and backward navigation.
   - In more, you can only move forward one page or one line at a time, while less allows more fine-grained navigation.
2. **Features:**
   - less has more features than more, including searching within the file, executing shell commands, and viewing compressed files.

- more is simpler and more straightforward, suitable for quick file viewing tasks.

3. Efficiency:

- less is more efficient for viewing large files because it doesn't load the entire file into memory at once, unlike more.

Ex: **more samples.txt**

Less command options:
Press ENTER-→Line by Line display
Press SPACEBAR→Page by page view
CTRL+B→ Backward View page by page
CTRL+F→Forward view page by page
CRTL+D→Forward half view
CTRL+U→Backward halfview

15. head:

The head command is used to display the beginning of a text file or the output of another command. By default, it displays the first 10 lines of the specified file or input.

Basic Usage

1. Display the first 10 lines of a file: **head filename**

```
ubuntu@ip-172-31-28-223:~$ wc -l samples.txt
35 samples.txt
ubuntu@ip-172-31-28-223:~$ head samples.txt
Linux DIrectory Structure:
    / -- The Root Directory
    /bin -- Essential User Binaries
    /boot -- Static Boot Files
    /cdrom -- Historical Mount Point for CD-ROMs
    /dev -- Device Files
    /etc -- Configuration Files
    /home -- Home Folders
    /lib -- Essential Shared Libraries
    /lost+found -- Recovered Files
ubuntu@ip-172-31-28-223:~$
```

2. Display a specific number of lines from the beginning of a file:
head -n N filename
Ex: head -n 5 samples.txt

It will display the first 5 lines from the samples.txt file.

```
ubuntu@ip-172-31-28-223:~$ wc -l samples.txt
35 samples.txt
ubuntu@ip-172-31-28-223:~$ head -n 5 samples.txt
Linux DIrectory Structure:
    / -- The Root Directory
    /bin -- Essential User Binaries
    /boot -- Static Boot Files
    /cdrom -- Historical Mount Point for CD-ROMs
ubuntu@ip-172-31-28-223:~$
```

3. Display the first part of the output of another command:

command | head

Ex: cat -n samples.txt | head -n 9

```
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Linux DIrectory Structure:
     2       / -- The Root Directory
     3       /bin -- Essential User Binaries
     4       /boot -- Static Boot Files
     5       /cdrom -- Historical Mount Point for CD-ROMs
     6       /dev -- Device Files
     7       /etc -- Configuration Files
     8       /home -- Home Folders
     9       /lib -- Essential Shared Libraries
    10       /lost+found -- Recovered Files
    11       /media -- Removable Media
    12       /mnt -- Temporary Mount Points
    13       /opt -- Optional Packages
    14       /proc -- Kernel & Process Files
    15       /root -- Root Home Directory
    16       /run -- Application State Files
    17       /sbin -- System Administration Binaries
    18       /srv -- Service Data
    19       /tmp -- Temporary Files
    20       /usr -- User Binaries & Read-Only Data
    21       /var -- Variable Data Files
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt | head -n 9
     1  Linux DIrectory Structure:
     2       / -- The Root Directory
     3       /bin -- Essential User Binaries
     4       /boot -- Static Boot Files
     5       /cdrom -- Historical Mount Point for CD-ROMs
     6       /dev -- Device Files
     7       /etc -- Configuration Files
     8       /home -- Home Folders
     9       /lib -- Essential Shared Libraries
ubuntu@ip-172-31-28-223:~$
```

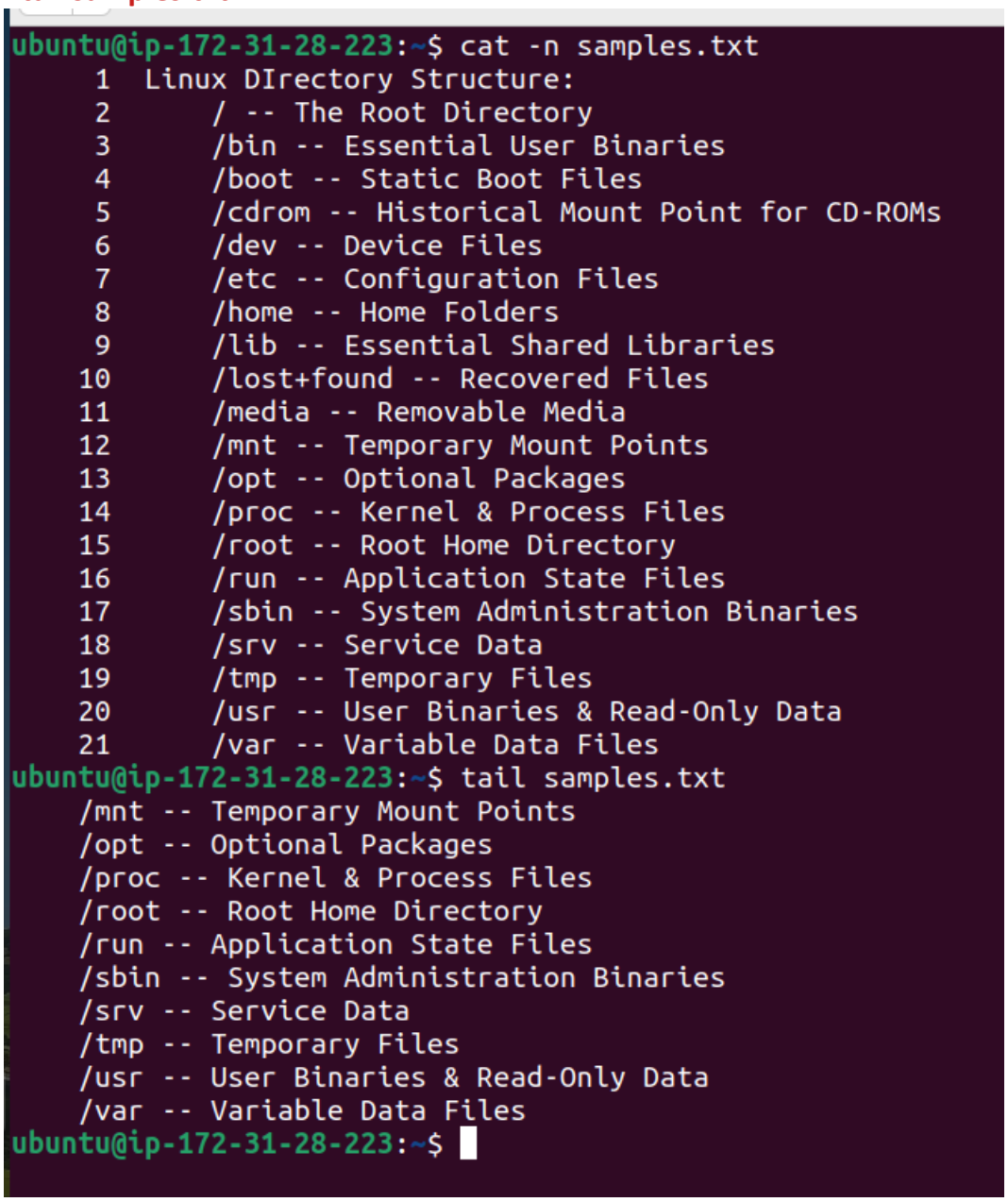
16. tail:

The tail command is used to display the last part of a text file or the output of another command. By default, it displays the last 10 lines of the specified file or input.

Basic Usage:

1. Display the last 10 lines of a file: tail filename

Ex: tail samples.txt

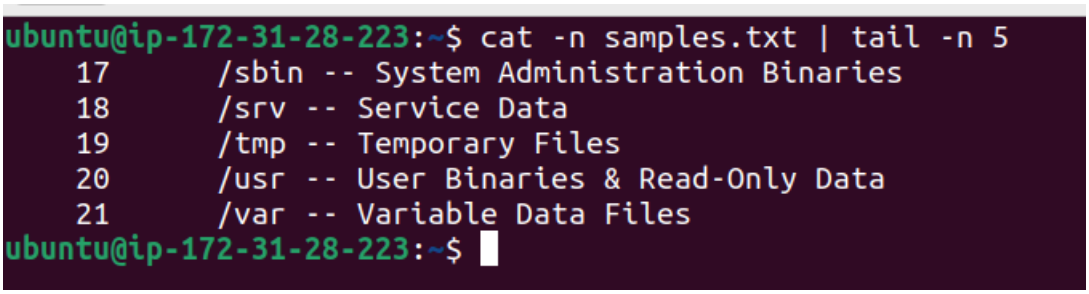


2. Display a specific number of lines from the end of a file:
tail -n N filename (Replace N with the number of lines you want to display.)

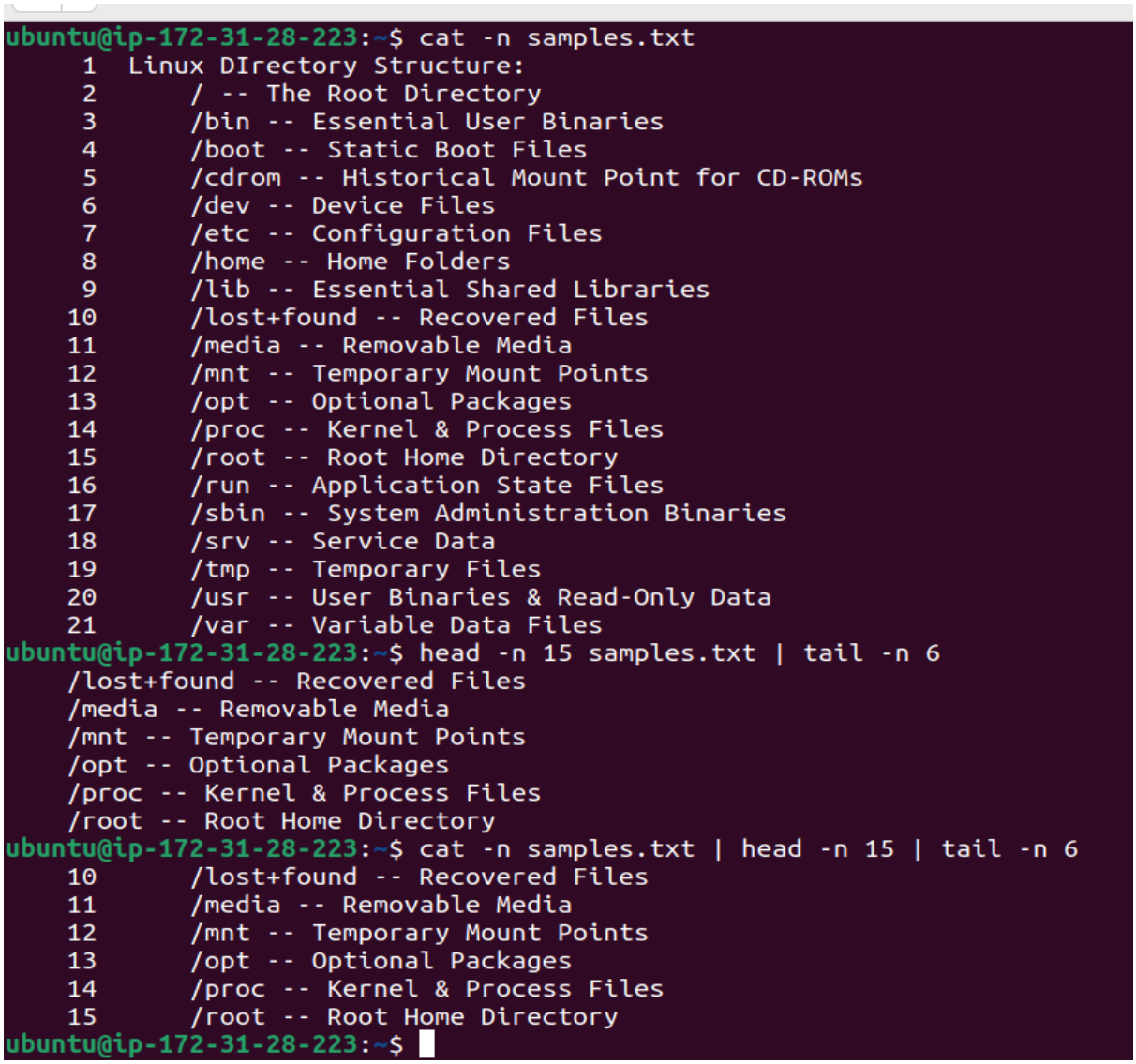
Ex: tail -n 5 samples.txt

```
ubuntu@ip-172-31-28-223:~$ cat -n samples.txt
     1  Linux DIrectory Structure:
     2       / -- The Root Directory
     3       /bin -- Essential User Binaries
     4       /boot -- Static Boot Files
     5       /cdrom -- Historical Mount Point for CD-ROMs
     6       /dev -- Device Files
     7       /etc -- Configuration Files
     8       /home -- Home Folders
     9       /lib -- Essential Shared Libraries
    10       /lost+found -- Recovered Files
    11       /media -- Removable Media
    12       /mnt -- Temporary Mount Points
    13       /opt -- Optional Packages
    14       /proc -- Kernel & Process Files
    15       /root -- Root Home Directory
    16       /run -- Application State Files
    17       /sbin -- System Administration Binaries
    18       /srv -- Service Data
    19       /tmp -- Temporary Files
    20       /usr -- User Binaries & Read-Only Data
    21       /var -- Variable Data Files
ubuntu@ip-172-31-28-223:~$ tail -n 5 samples.txt
    /sbin -- System Administration Binaries
    /srv -- Service Data
    /tmp -- Temporary Files
    /usr -- User Binaries & Read-Only Data
    /var -- Variable Data Files
ubuntu@ip-172-31-28-223:~$
```

3. Display the last part of the output of another command:
command | tail Ex: cat -n samples.txt | tail -n 5



4. Combining the head and tail commands can be useful for various purposes, such as extracting a specific range of lines from a file or excluding a certain number of lines from the beginning or end of a file.
Logic: If you need to display x to y of lines from a file then
head -n y | tail (y-x)+1
Ex: Extract lines 10 to 15 from a file:
x=10 y=15 then it is head -n 15 filename | tail (15-10)+1
head -n 15 filename | tail -n 6



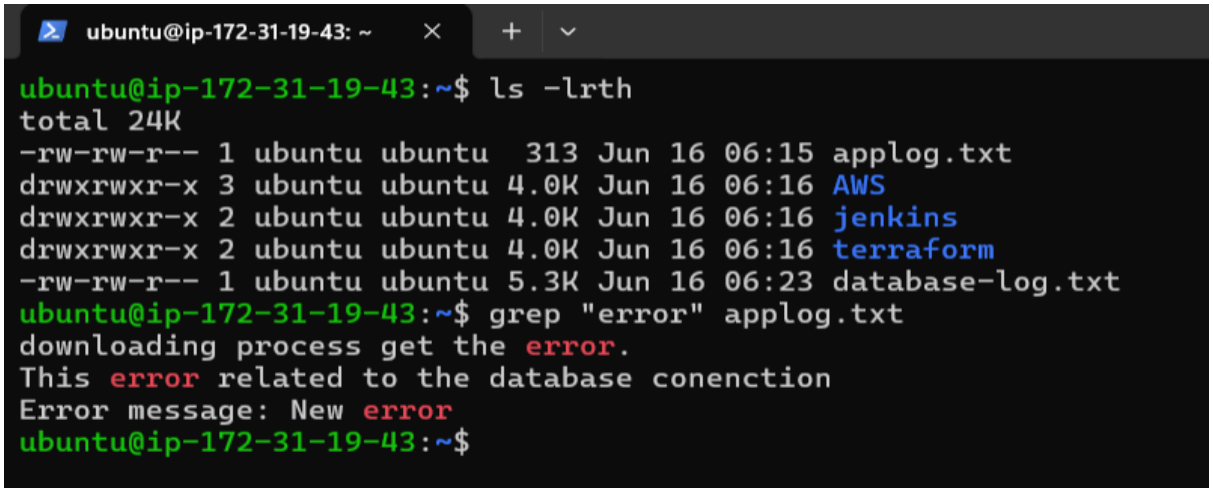
17. ‘grep’ Command:

The grep command in Linux is a powerful tool used for searching plain-text data sets for lines that match a regular expression.

1. **Basic String Search**:

   Ex: grep “error” applog.txt

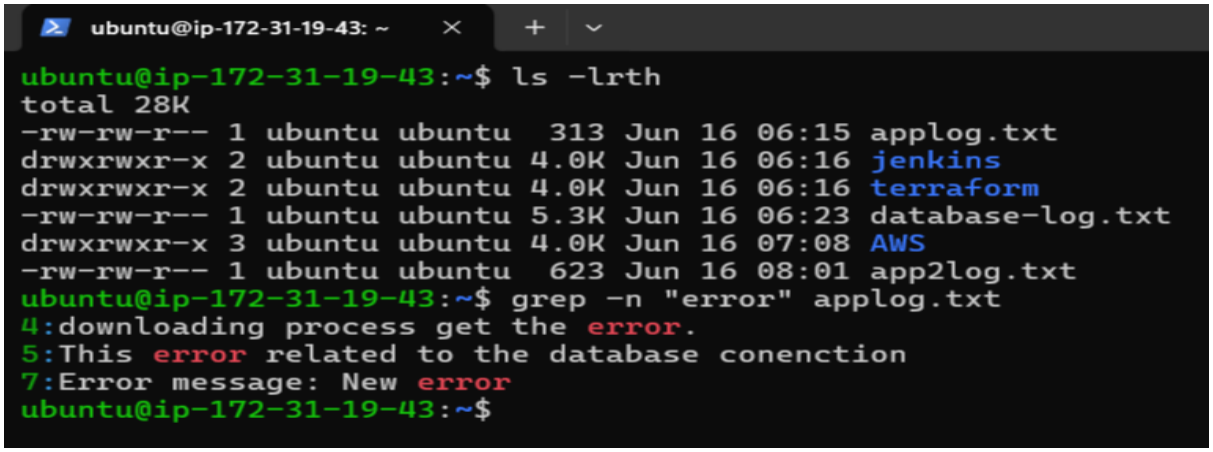
   This command searches for the string "error" in the file applog.txt.



2. **Display Line Numbers**:

   Ex: **grep -n “error” applog.txt**

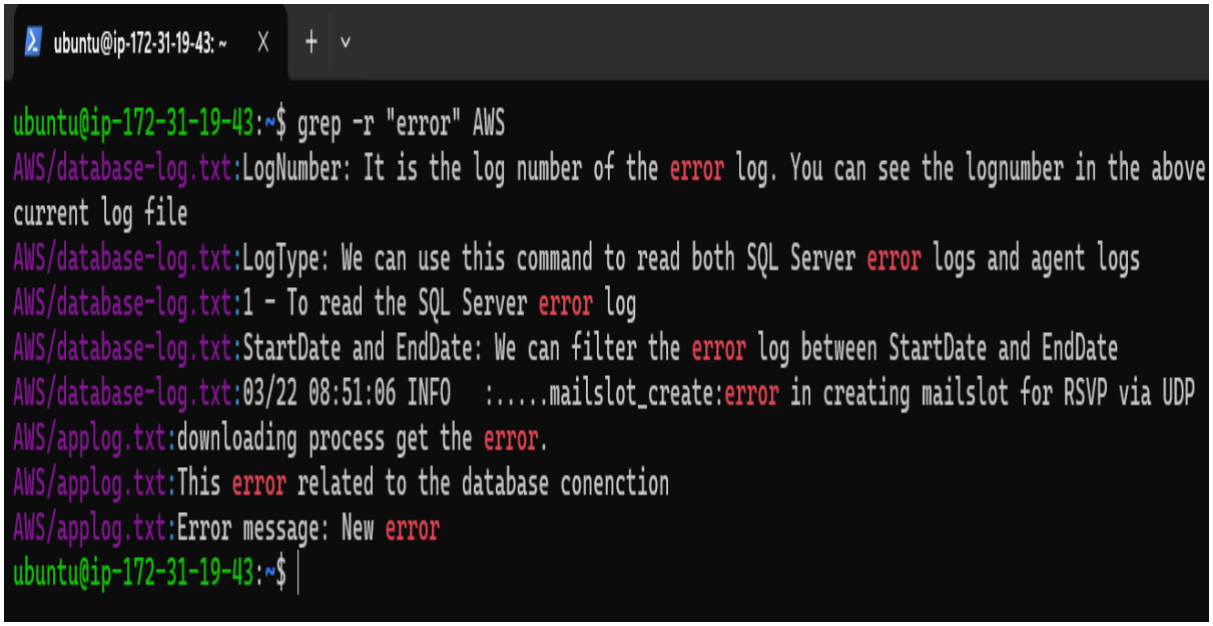
This command shows the line numbers along with matching lines in the file.



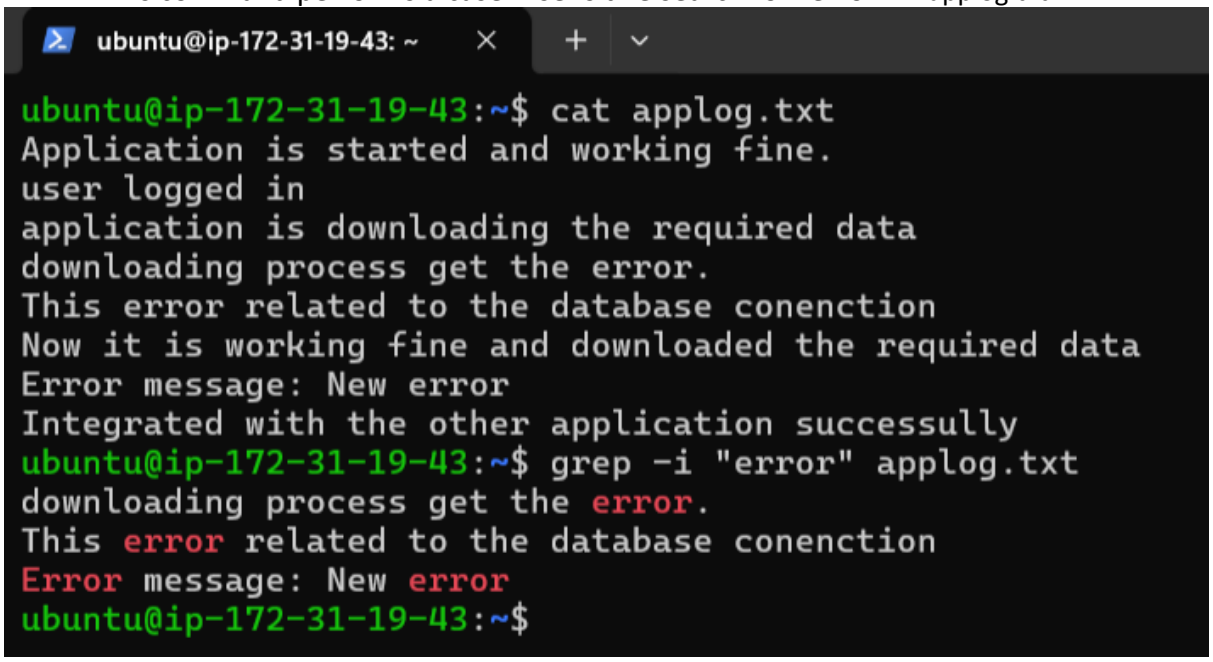
3. **Search Recursively**:

   Ex: **grep -r “error” AWS**

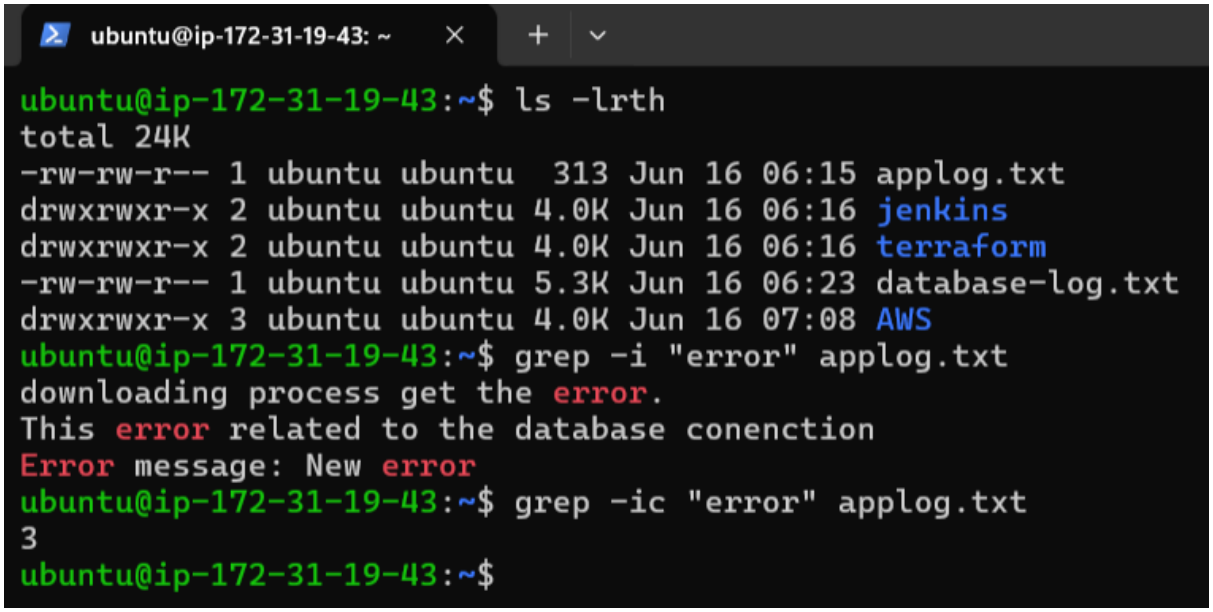
   This command searches for "error" in all files within the specified “AWS” directory and its subdirectories.



4. **Case-Insensitive Search**:
   Ex: **grep -i "error" applog.txt**
   This command performs a case-insensitive search for "error" in applog.txt.



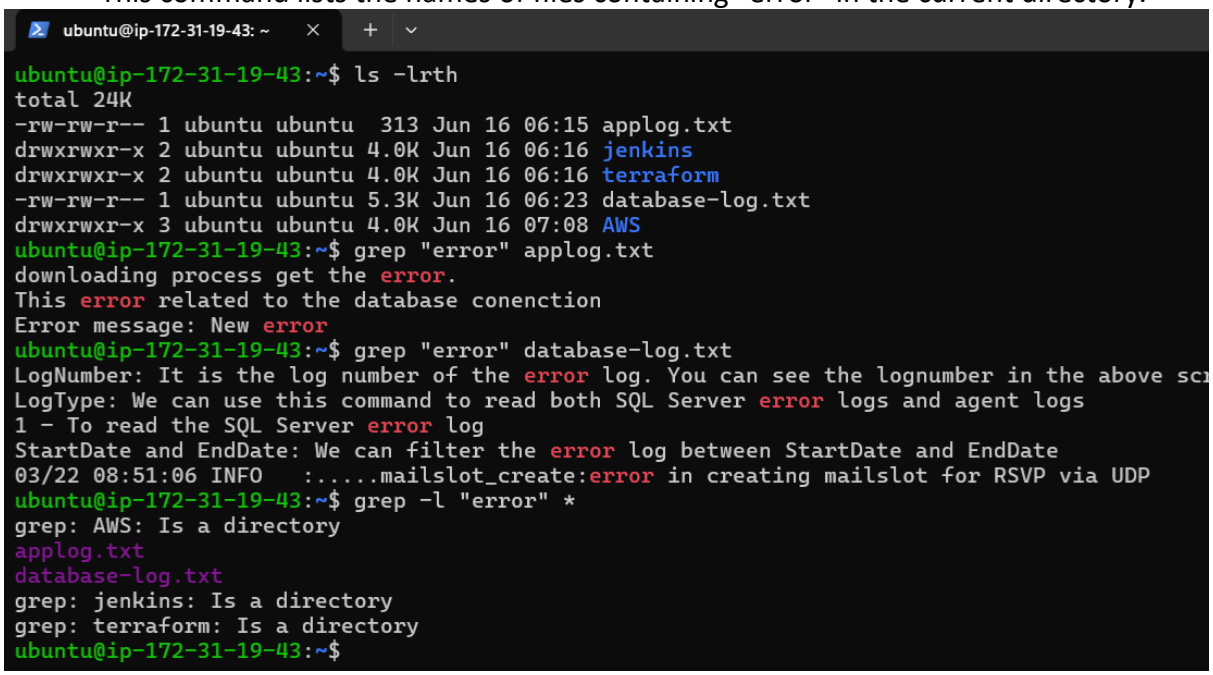
5. **If we want to get how many lines are matched (count) with our search key word or regular expression using grep command**
   Ex: grep -ic "error" applog.txt



6. **Print File Names with Matches**:

   Ex: grep -l “error” *

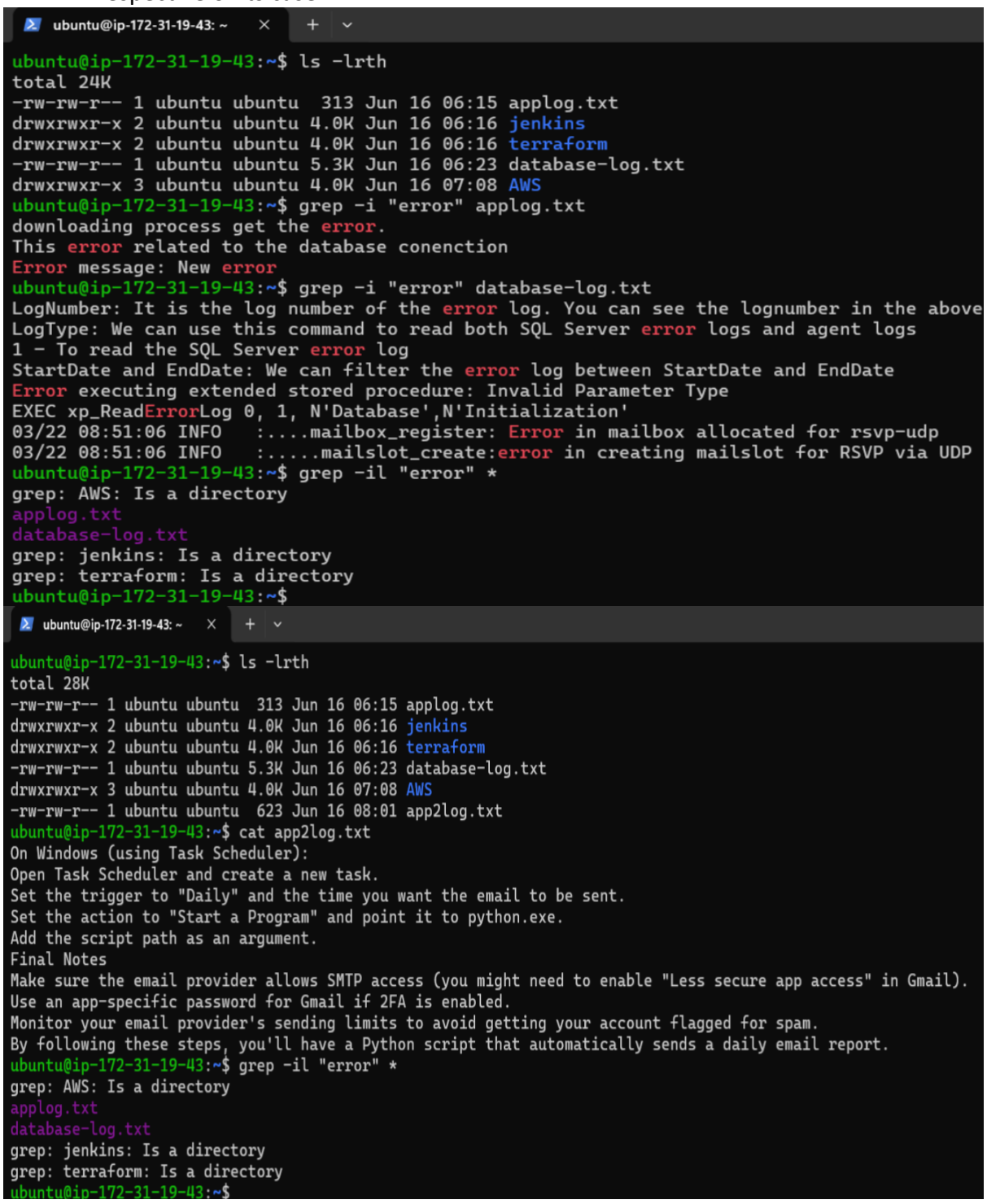
   This command lists the names of files containing "error" in the current directory.



7. **Print File Names with Matches**:

   Ex: grep -il “error” *

This command lists the names of files containing "error" in the current directory irrespective of its case.

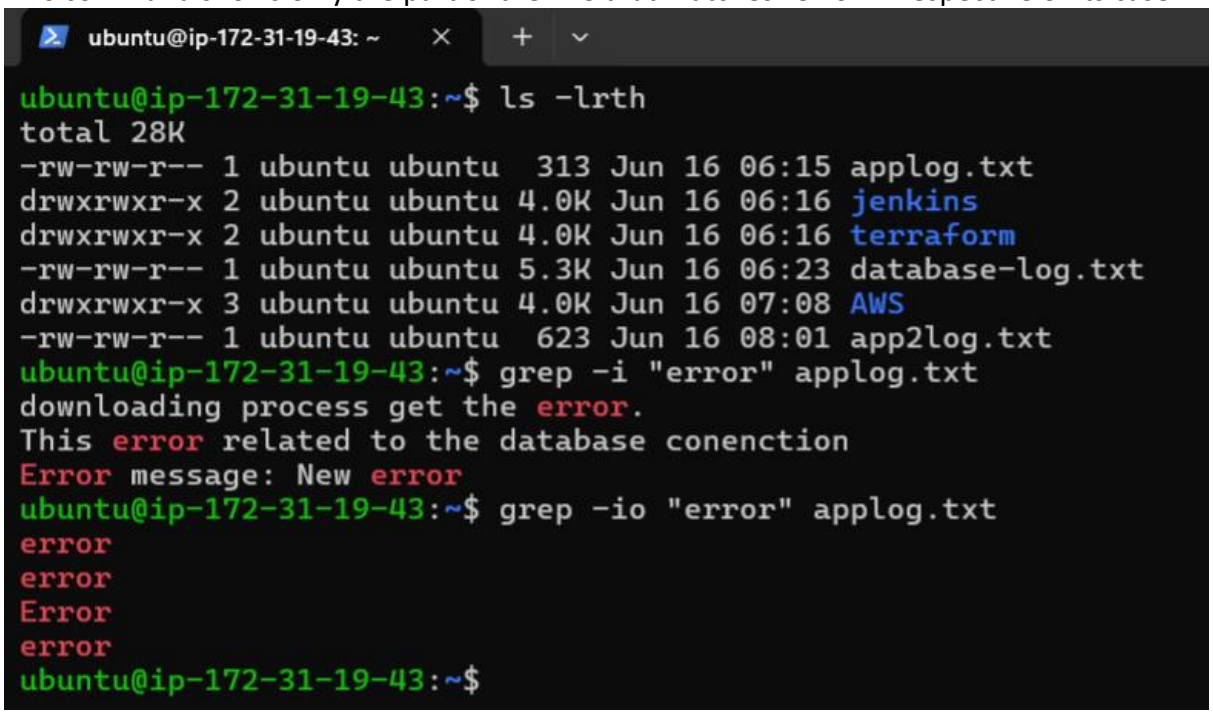


Note: “Hi guys app2log.txt do not have any "error" that's why we did not get that filename along with applog.txt and database-log.txt in the above comm and result."

8. **Show Only Matching Part of the Line**:

Ex: grep -io “error” applog.txt

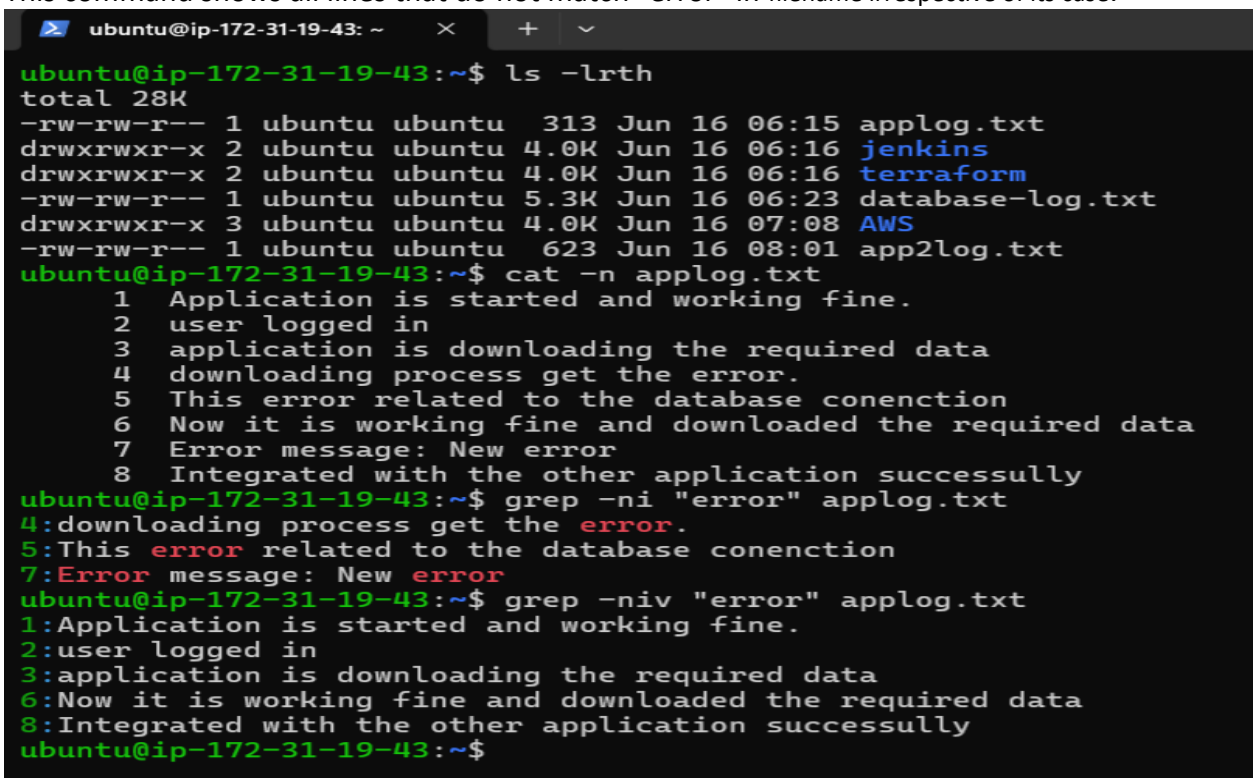
This command shows only the part of the line that matches "error" irrespective of its case.



9. Invert Match (Show Non-Matching Lines):

Ex: grep -iv “error” filelog.txt

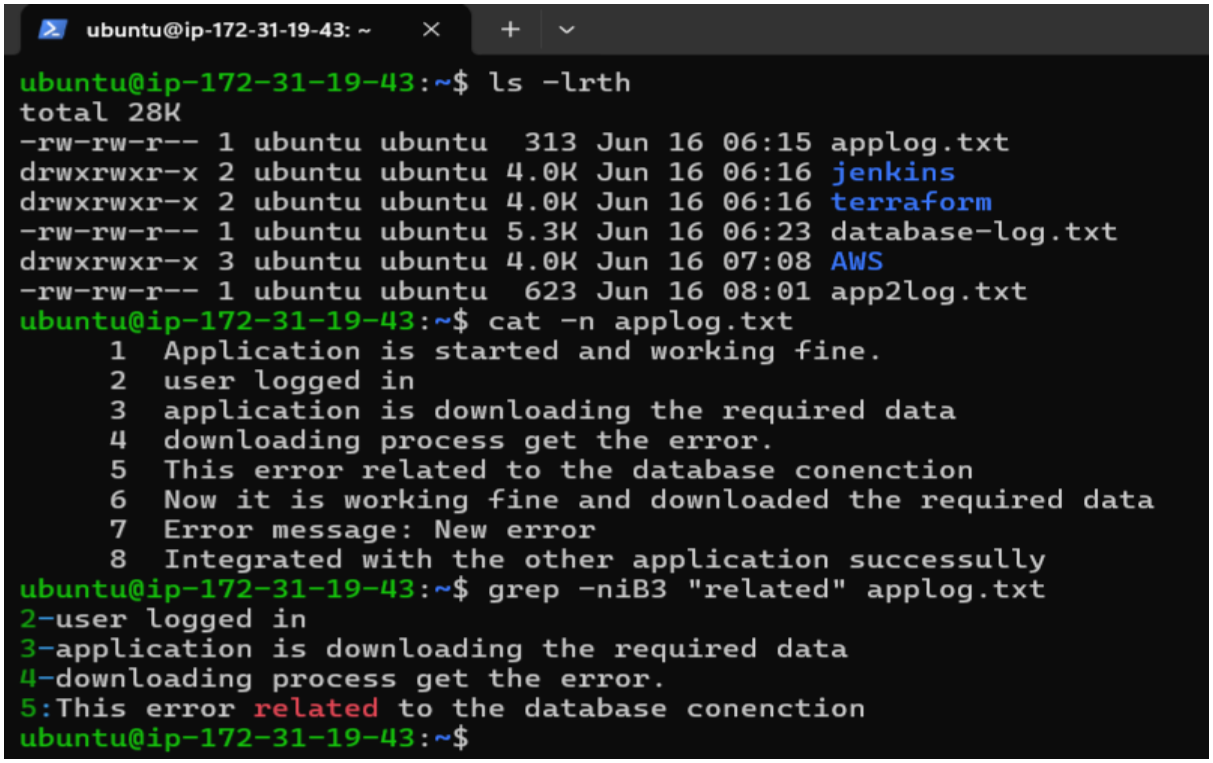
This command shows all lines that do not match "error" in filename irrespective of its case.



10. Show Lines Before Matching Lines:

Ex: grep -niB3 “related” applog.txt

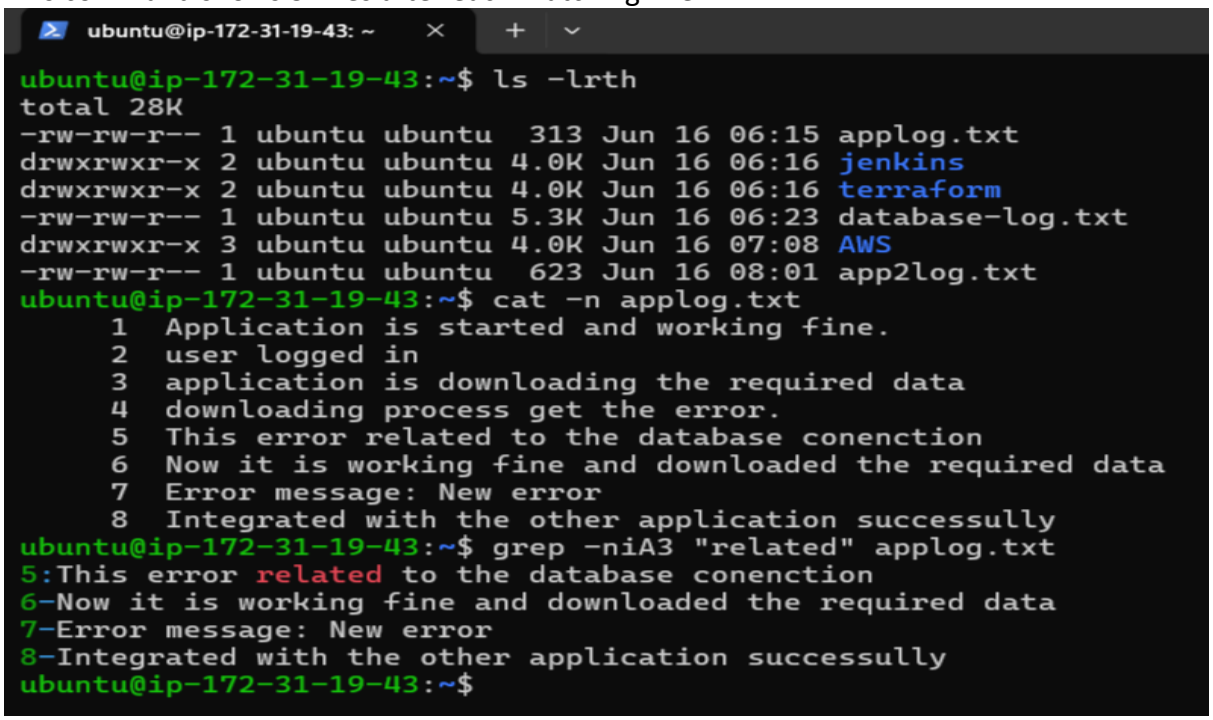
This command shows 3 lines before each matching line.



11. Show Lines after Matching Lines:

Ex: grep -niA3 “related” applog.txt

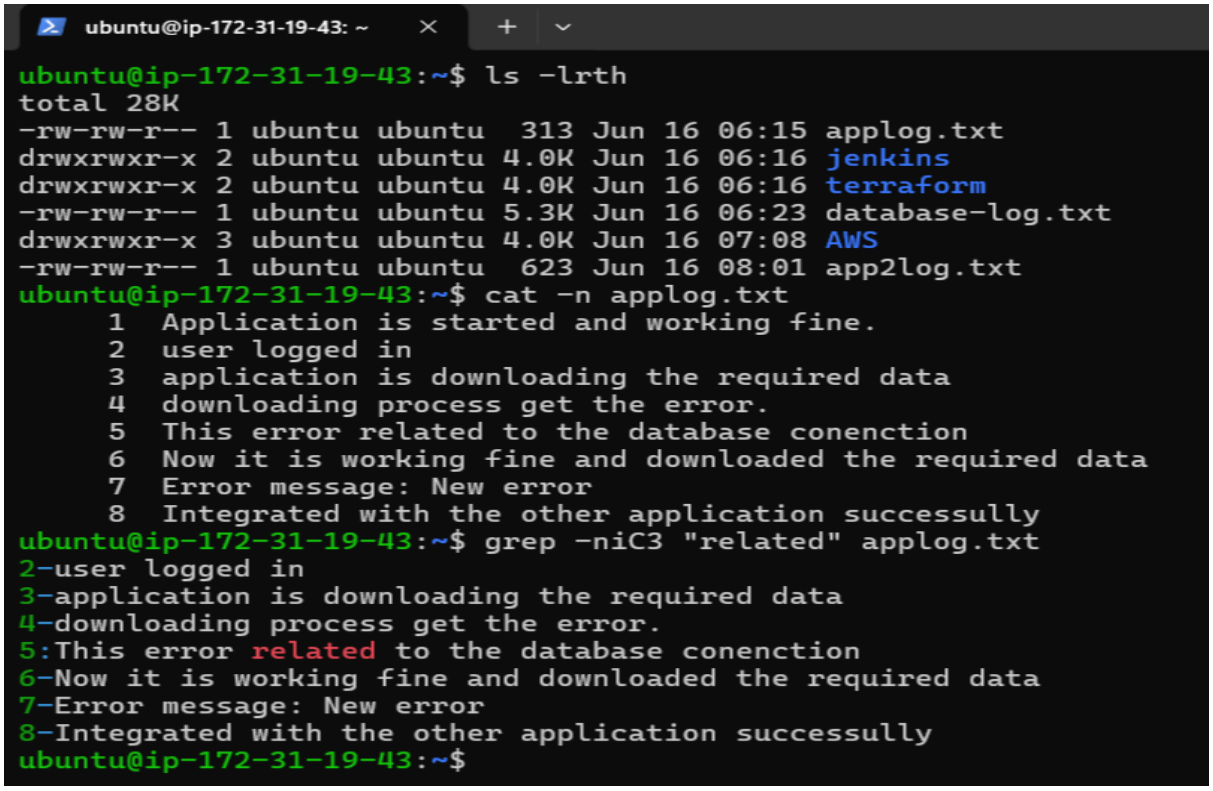
This command shows 3 lines after each matching line.



12. Show Lines before and after Matching Lines:

Ex: grep -niC3 “related” applog.txt

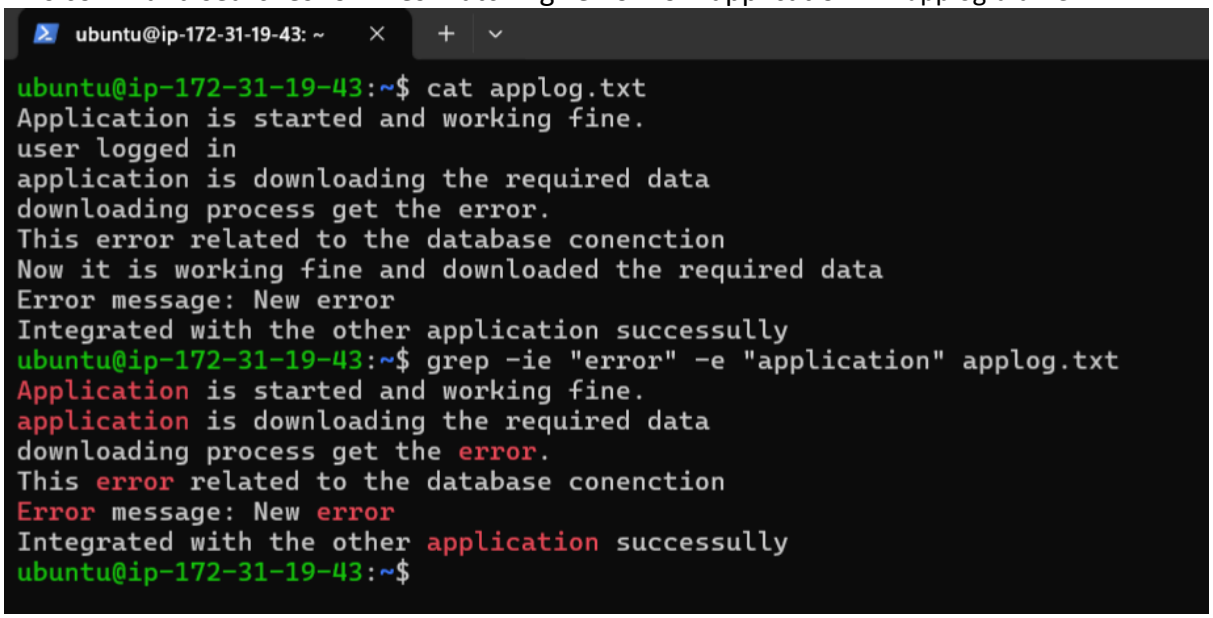
This command shows 3 lines both before and after each matching line.



13. Search Multiple Patterns:

Ex: grep -ie “error” -e “application” applog.txt

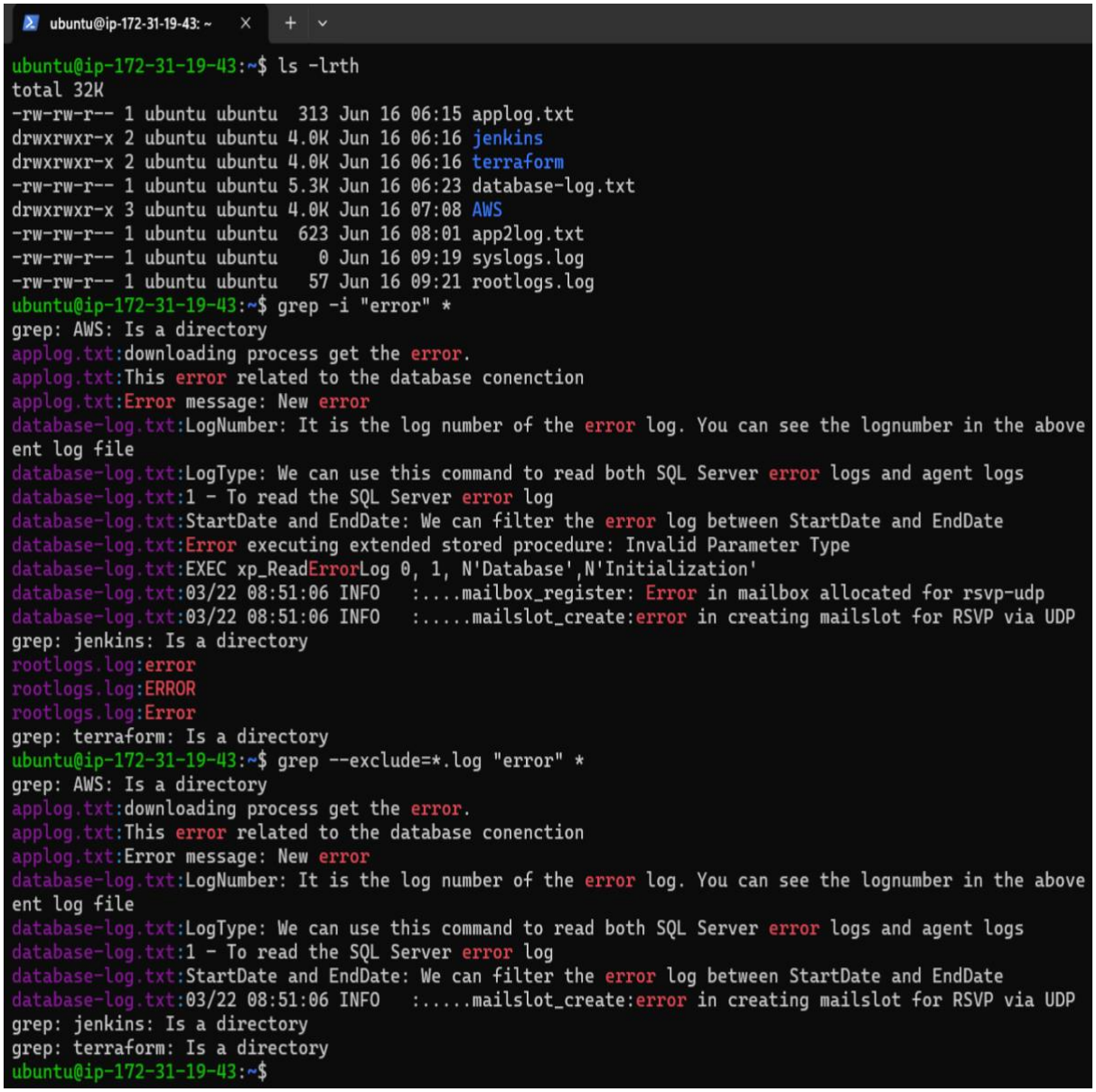
This command searches for lines matching "error" or "application" in applog.txt file



14. Exclude Certain File Types:

Ex: grep --exclude=*.log “error” *

This command searches for "error" in all files except those with the .log extension.



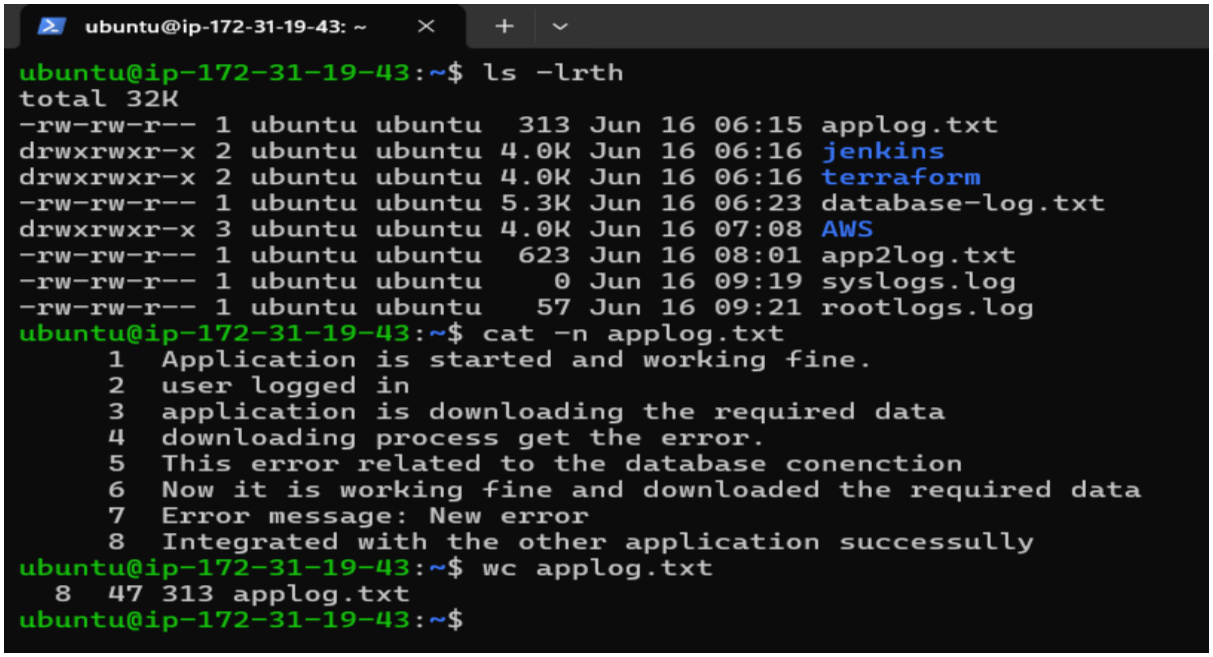
18. ‘wc’ command:

The wc (word count) command in Linux is used to count the number of lines, words, characters, and bytes in files.

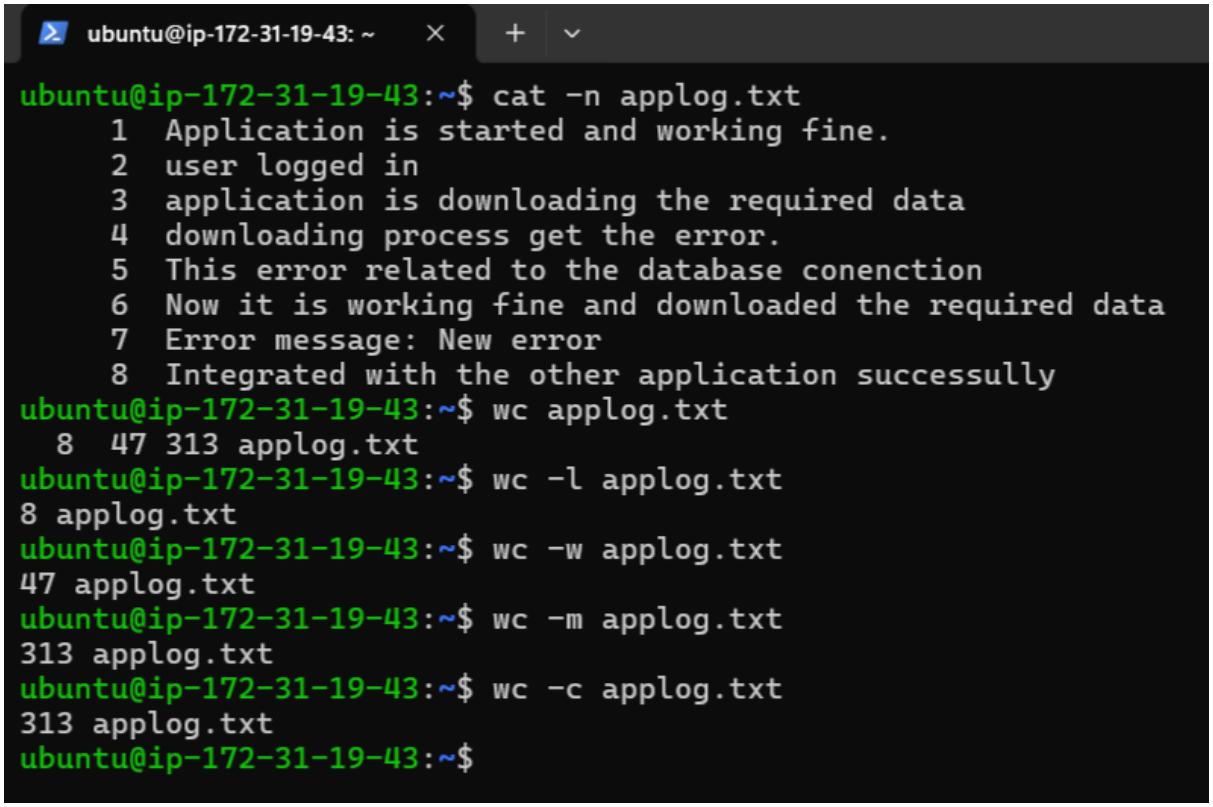
Here are some common use cases for the wc command:

1. **Count Lines, Words, and Characters**:
   Ex. wc applog.txt

   This command displays the number of lines, words, and characters in the file applog.txt.



2. **Count Words Only, Count Words Only, Count Characters Only, & Count Bytes Only:**
   Ex: wc -l applog.txt
   This command displays the number of lines in the file applog.txt
   wc -w applog.txt
   This command displays the number of words in the file applog.txt
   wc -m applog.txt
   This command displays the number of characters in the file applog.txt
   wc -c applog.txt
   This command displays the number of bytes in the file applog.txt



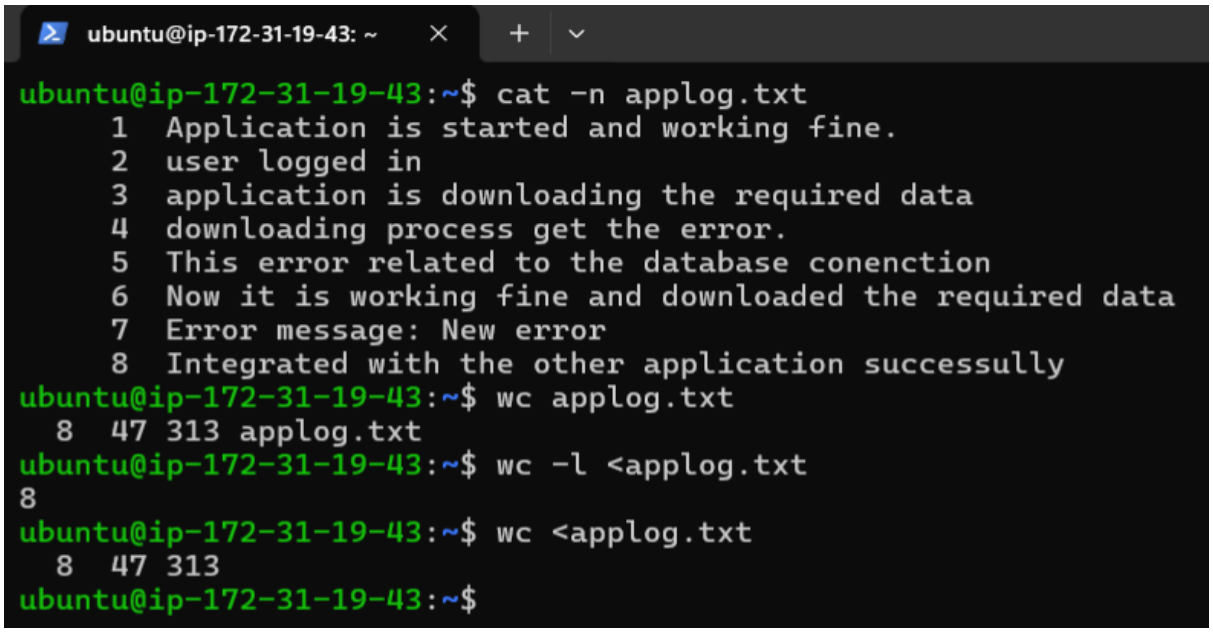
3. **Suppress Filename in Output**:

Ex1: **wc -l < applog.txt**

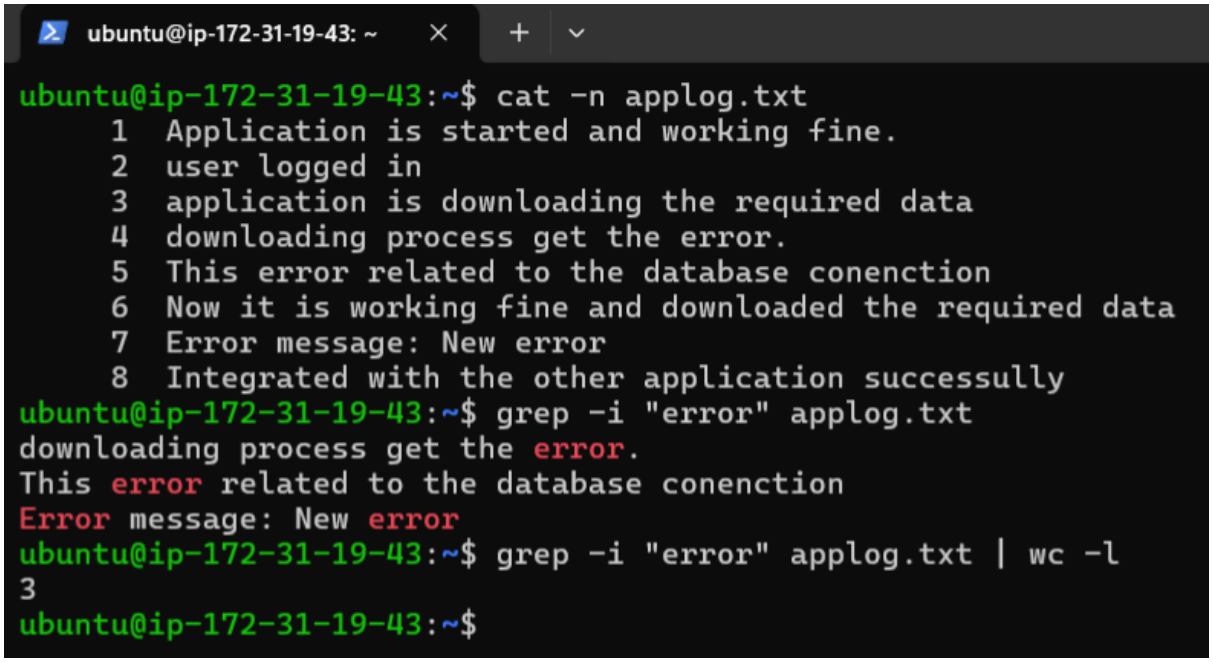
This command displays the line count without showing the filename, useful when used in scripts.

Ex2. **wc < applog.txt**

This command displays the line count, word count, and character count without showing the filename.



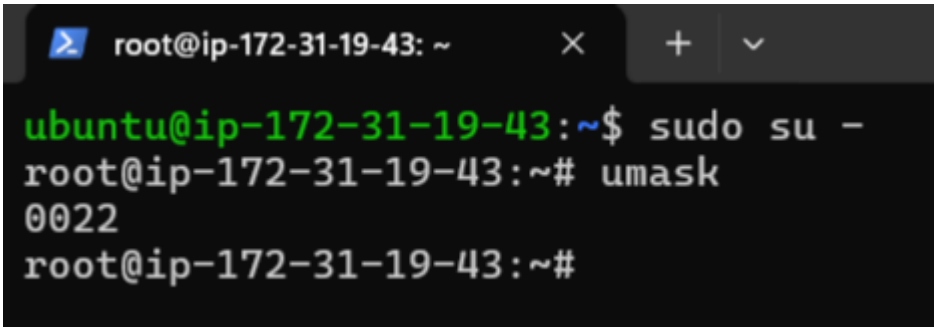
4. **Use in Combination with Other Commands**:
   Ex: grep -i “error” applog.txt | wc -l



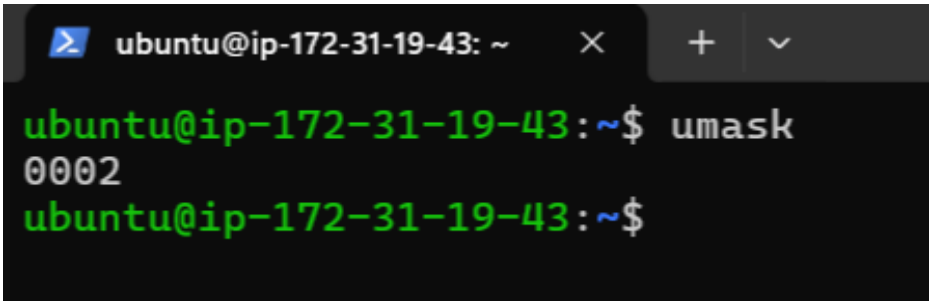
19. ‘umask’ command:

The **umask** command in Unix/Linux is used to set or display the default file or directory creation permissions (also known as the file mode creation mask) for newly created files and directories. The **umask** value determines the permissions that will be masked (i.e., turned off) when new files and directories are created.
Here are some key points and use cases for the **umask** command:

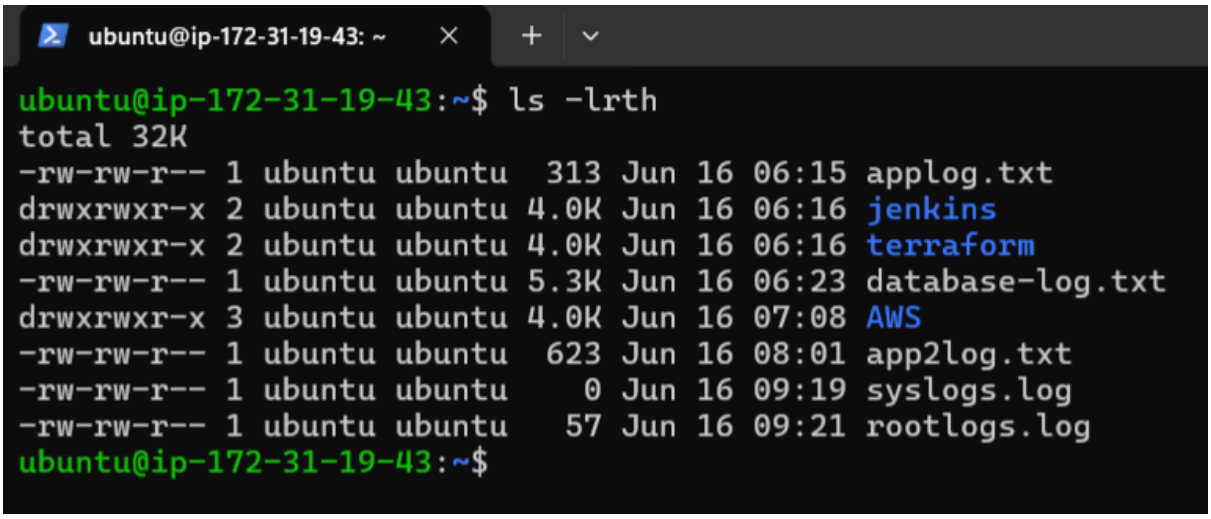
1. **Display Current umask Value:**
   Ex. umask
   Root User default umask value is :



This normal User default umask value is :



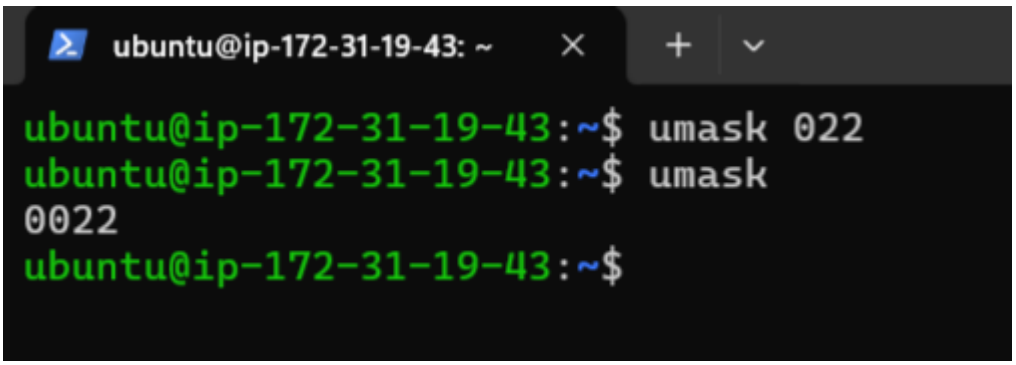
In the following example the default file permission are (666-002=664) and default directory permissions are (777-002=775)



2. Set umask Value:

Ex. umask 022

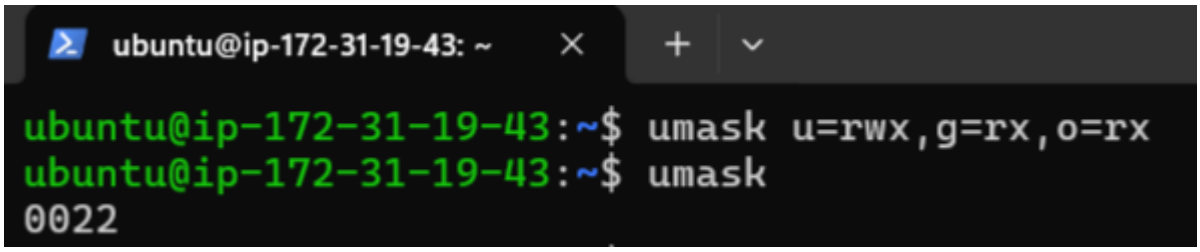
This command sets the umask value to 022, which means newly created files will have permissions 755 (rwxr-xr-x) and directories will have permissions 755 (rwxr-xr-x).



3. **Set umask Value in Symbolic Format**:

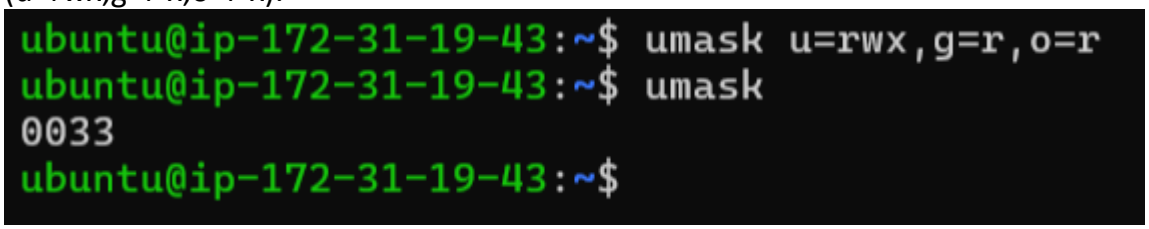
Ex1: umask u=rwx,g=rx,o=rx

This command sets the umask value to 022, which means newly created files will have permissions 755 (u=rwx,g=r-x,o=r-x) and directories will have permissions 755 (u=rwx,g=r-x,o=r-x).

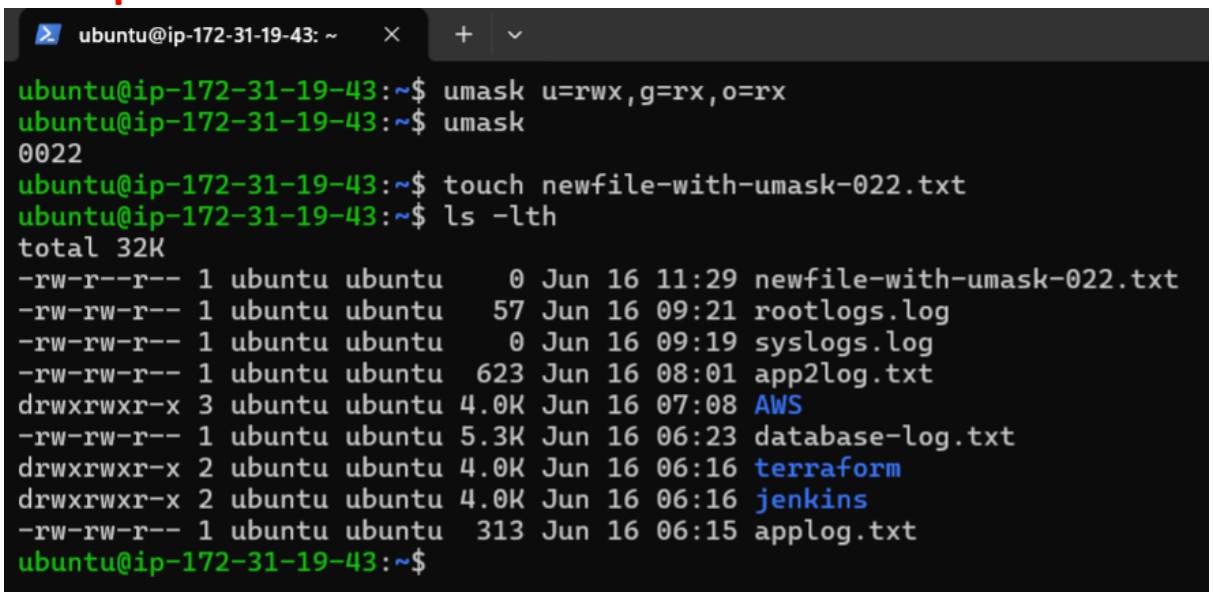


Ex2: umask u=rwx,g=r,o=r

This command sets the umask value to 033, which means newly created files will have permissions 755 (u=rwx,g=r-x,o=r-x) and directories will have permissions 755 (u=rwx,g=r-x,o=r-x).

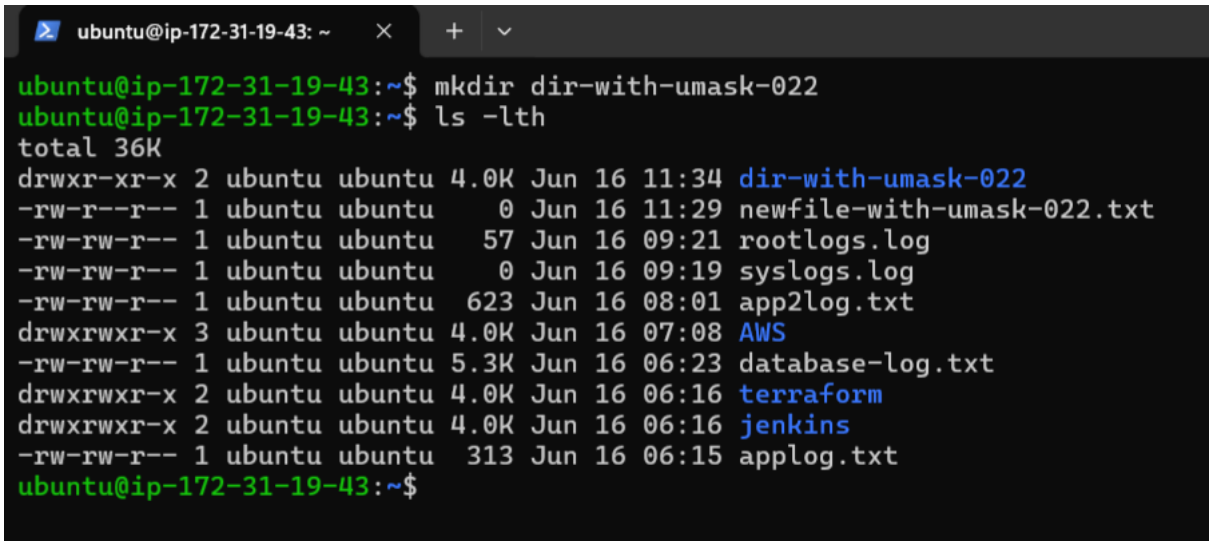


Example for Files with umask 022:



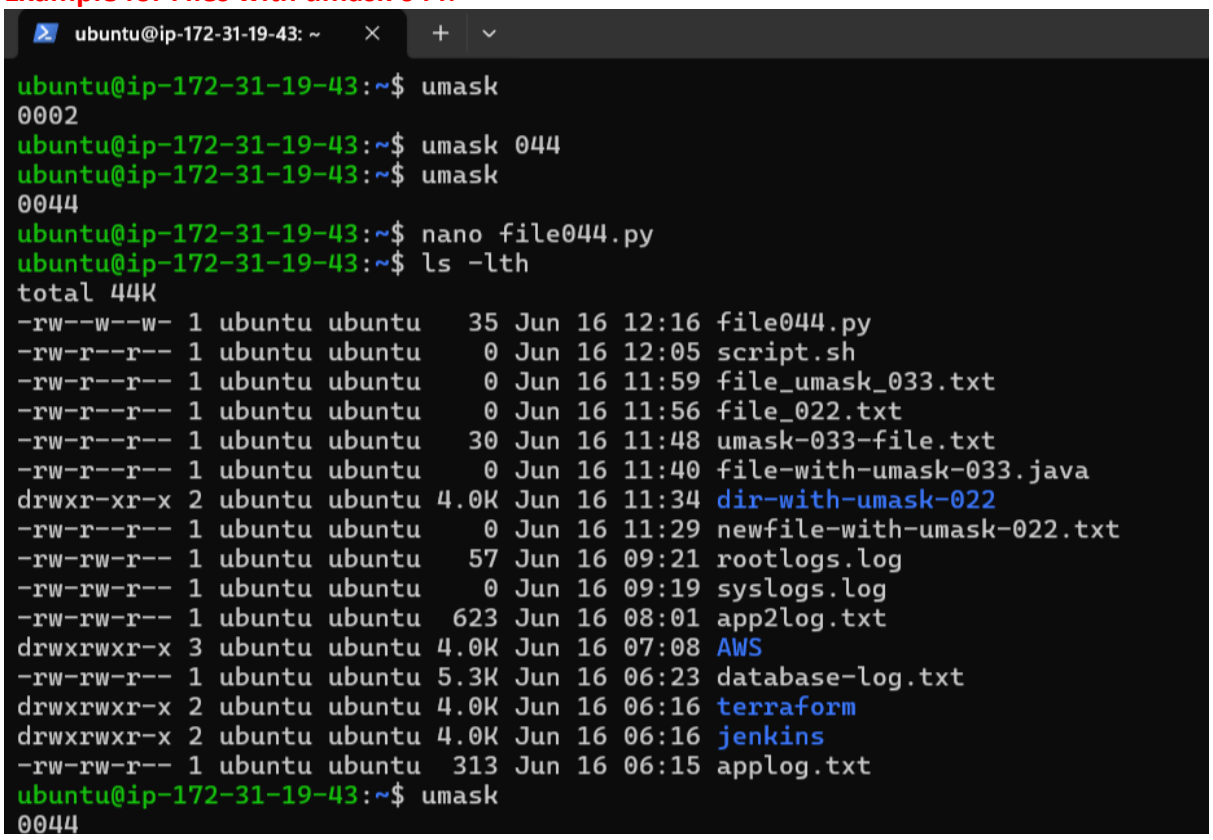
In the above files example I have set the umask with 022 and created a new file called “newfile-with-umask-022.txt”. If you check the file permissions, they are (666-022=644: rw-,r--,r--)

Example for Directories with umask 022:



In the above directories example I have set the umask with 022 and created a new directory called “dir-with-umask-022”. If you check the file permissions, they are (777-022=755: rwx,r-x,r-x)

Example for Files with umask 044:



In the above Files example I have set the umask with 044 and created a new file called “file044.py”. If you check the file permissions, they are :
(666-044=622: rw-,-w-,-w-)

Example for Directories with umask 044:

```
ubuntu@ip-172-31-19-43:~$ umask
0044
ubuntu@ip-172-31-19-43:~$ mkdir dir044
ubuntu@ip-172-31-19-43:~$ ls -lth
total 48K
drwx-wx-wx 2 ubuntu ubuntu 4.0K Jun 16 12:17 dir044
-rw--w--w- 1 ubuntu ubuntu   35 Jun 16 12:16 file044.py
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 12:05 script.sh
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 11:59 file_umask_033.txt
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 11:56 file_022.txt
-rw-r--r-- 1 ubuntu ubuntu   30 Jun 16 11:48 umask-033-file.txt
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 11:40 file-with-umask-033.java
drwxr-xr-x 2 ubuntu ubuntu 4.0K Jun 16 11:34 dir-with-umask-022
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 11:29 newfile-with-umask-022.txt
-rw-rw-r-- 1 ubuntu ubuntu   57 Jun 16 09:21 rootlogs.log
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 09:19 syslogs.log
-rw-rw-r-- 1 ubuntu ubuntu  623 Jun 16 08:01 app2log.txt
drwxrwxr-x 3 ubuntu ubuntu 4.0K Jun 16 07:08 AWS
-rw-rw-r-- 1 ubuntu ubuntu 5.3K Jun 16 06:23 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 06:16 terraform
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 06:16 jenkins
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 06:15 applog.txt
ubuntu@ip-172-31-19-43:~$
```

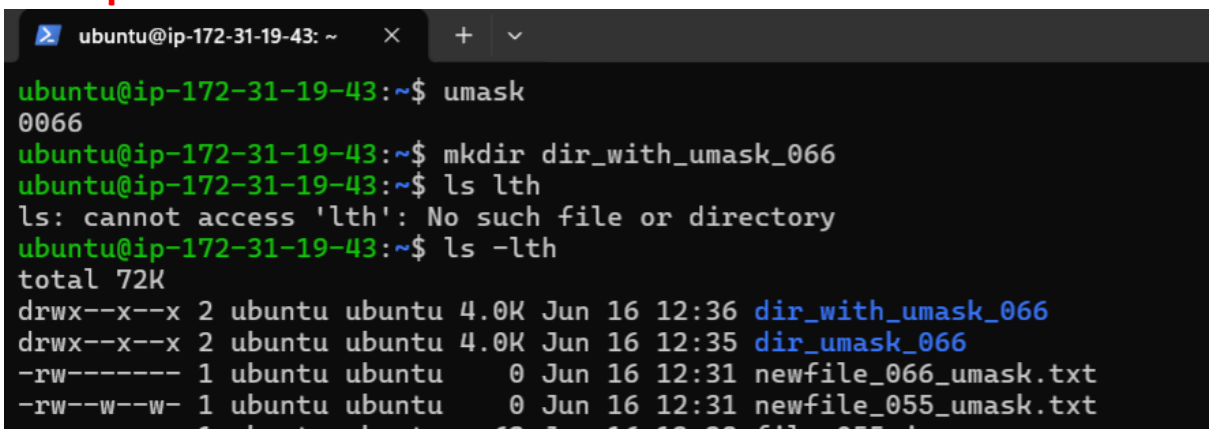
In the above directories example I have set the umask with 044 and created a new directory called “dir044”. If you check the file permissions, they are:
(777-044=733: rwx,-wx,-wx)

Example for Files with umask 066:

```
ubuntu@ip-172-31-19-43:~$ umask 066
ubuntu@ip-172-31-19-43:~$ umask
0066
ubuntu@ip-172-31-19-43:~$ touch newfile_066_umask.txt
ubuntu@ip-172-31-19-43:~$ ls -lth
total 64K
-rw------- 1 ubuntu ubuntu    0 Jun 16 12:31 newfile_066_umask.txt
-rw--w--w- 1 ubuntu ubuntu    0 Jun 16 12:31 newfile_055_umask.txt
-rw--w--w- 1 ubuntu ubuntu   63 Jun 16 12:28 file_055.java
-rw-r--r-- 1 ubuntu ubuntu   31 Jun 16 12:27 file_033.py
-rw-r--r-- 1 ubuntu ubuntu   37 Jun 16 12:25 ex_file_033.java
-rw-r--r-- 1 ubuntu ubuntu   29 Jun 16 12:23 file033.txt
```

In the above Files example I have set the umask with 066 and created a new file called “newfile_066_umask.txt”. If you check the file permissions, they are :
(666-066=600: rw-, ---, ---)

Example for Directories with umask 066:



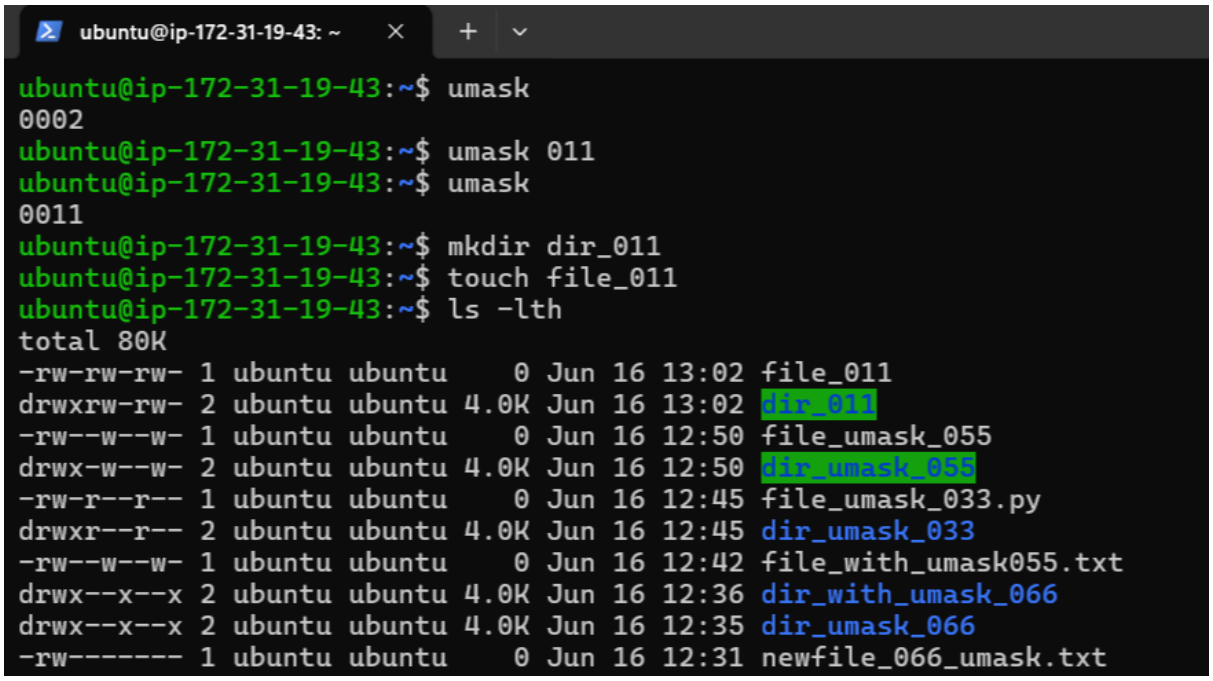
In the above Directories example I have set the umask with 066 and created a new directory called "dir_with_umask_066". If you check the file permissions, they are : (777-066=711: rwx, --x, --x)

Umask with 011:

In the below example I have set the umask with 011 and created a new file called "file_011" and directory called "dir_011".

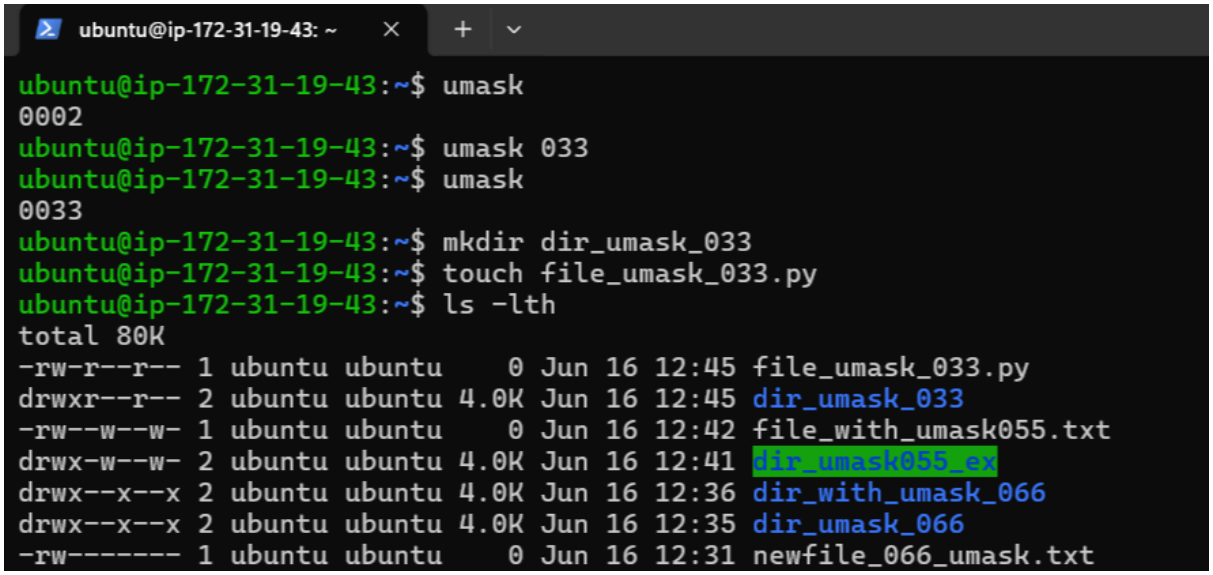
If you check the file permissions, they are : (666-011=655 but it is 666: rw-, rw-, rw-)

If you check the directory permissions, they are : (777-011=766: rwx, rw-, rw-)



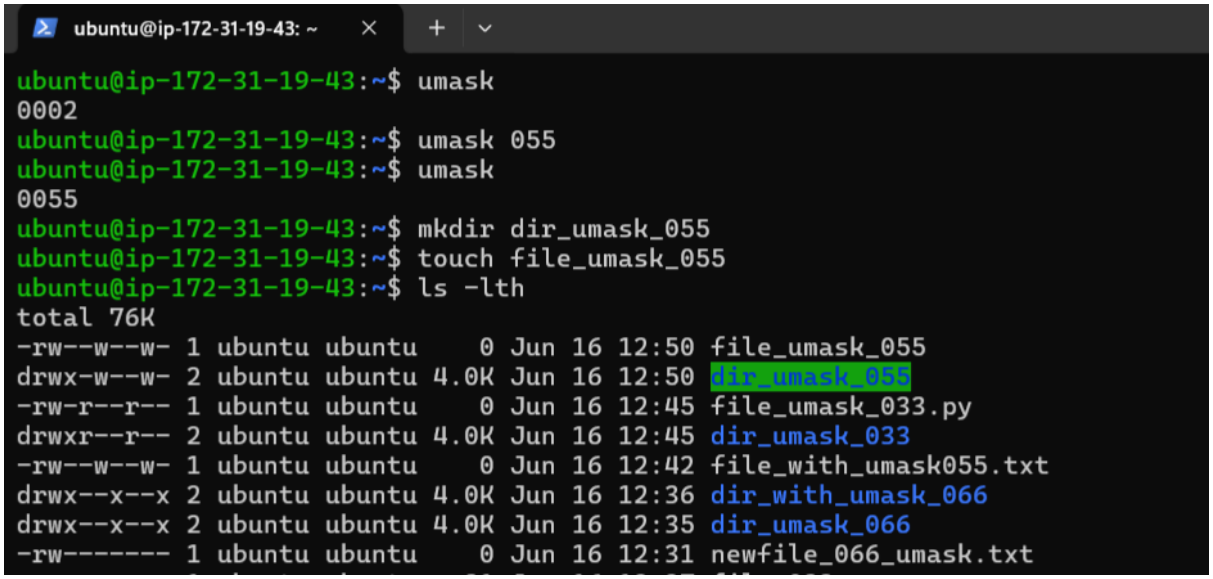
Umask with 033:

In the below example I have set the umask with 033 and created a new file called "file_umask_033.py" and directory called "dir_umask_033".
If you check the file permissions, they are : (666-033=633 but it is 644: rw-, r--, r--)
If you check the directory permissions, they are : (777-033=744: rwx, r--, r--)



Umask with 055:

In the below example I have set the umask with 055 and created a new file called "newfile_066_umask.txt" and directory called "dir_umask_055".
If you check the file permissions, they are : (666-055=611 but it is 622: rw-, -w-, -w-)
If you check the directory permissions, they are : (777-055=722: rwx, -w-, -w-)



In this way whenever default file permissions 666-umask value gives the execute permission (For example if umask value is 011 then 666-011=655 but it gives 666 only(rw-, rw-, rw-), it will always gives the default directory permissions regarding Group, Others categories

(Ex. umask value is 011 then 777-011=766 which means
User: READ WRITE EXECUTE (rwx)
Group: READ WRITE – (rw-)
Others: READ WRITE – (rw-)

Default Permissions and umask:

The default permissions for newly created files and directories are determined by the umask value.

The umask command sets the default permission mask, which subtracts from the default full permissions (666 for files and 777 for directories).

Types of Permissions

There are three types of permissions in Linux:

1. **Read (r)**: Allows reading the contents of a file or listing the contents of a directory.
2. **Write (w)**: Allows modifying the contents of a file or making changes within a directory (such as creating or deleting files).
3. **Execute (x)**: Allows executing a file (running it as a program) or entering a directory and accessing its contents.

Permission Categories

Permissions are set for three categories of users:

1. **Owner (u)**: The user who owns the file or directory.
2. **Group (g)**: The group that owns the file or directory.
3. **Others (o)**: All other users.

Viewing Permissions

Permissions can be viewed using the ls -l command:

ls -l filename

-rwxr-xr—

This breaks down as follows:

- -: Indicates a regular file (a d indicates a directory).
- rwx: Permissions for the owner (read, write, execute).
- r-x: Permissions for the group (read, execute).
- r--: Permissions for others (read).

Setting Permissions

20. chmod:

Permissions can be set using the chmod command. There are two ways to specify permissions: symbolic mode and numeric mode.

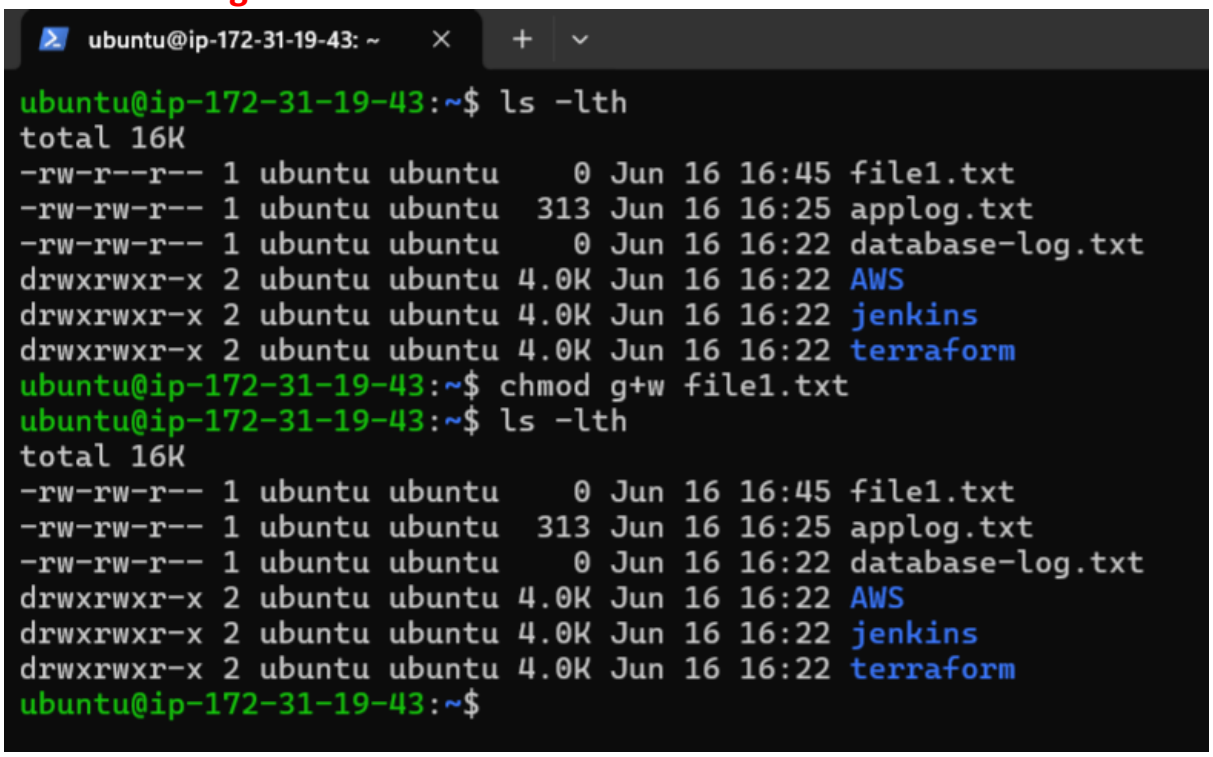
Symbolic Mode:

In symbolic mode, you use letters to specify permissions:

- **u: User (owner)**
- **g: Group**
- **o: Others**
- **a: All (user, group, and others)**
- **+: Add permission**
- **-: Remove permission**
- **=: Set exact permission**

1. **Add write permission for the group**:

   chmod g+w file1.txt



2. **Remove execute permission for others**:

chmod o-x file2.py

```
ubuntu@ip-172-31-19-43:~$ ls -lth
total 20K
-rw-r--r-x 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$ chmod o-x file2.py
ubuntu@ip-172-31-19-43:~$ ls -lth
total 20K
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$
```

3. **Set read and write permissions for the owner, and read-only for others**:

chmod u=rw,o=r prog.java

```
ubuntu@ip-172-31-19-43: ~
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
----r----- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$ chmod u=rw,o=r prog.java
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$
```

4. **Set read and write permissions for the user, and write permission for group and others**:

chmod u+rw,go+r app.go

```
ubuntu@ip-172-31-19-43: ~
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
---------- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$ chmod u+rw,go+r app.go
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$
```

5. **Set read and write permissions for user, group and others:**

chmod ugo+rw ecomapp.py

```
ubuntu@ip-172-31-19-43: ~
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
---------- 1 ubuntu ubuntu    0 Jun 16 17:16 ecomapp.py
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$ chmod ugo+rw ecomapp.py
ubuntu@ip-172-31-19-43:~$ ls -lth
total 24K
-rw-rw-rw- 1 ubuntu ubuntu    0 Jun 16 17:16 ecomapp.py
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$
```

6. Remove read permission for user, group and others for a Directory:

chmod ugo-r terraform/

```
ubuntu@ip-172-31-19-43: ~
ubuntu@ip-172-31-19-43:~$ ls -lrth
total 24K
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
--w--w--w- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-rw-rw- 1 ubuntu ubuntu    0 Jun 16 17:16 ecomapp.py
ubuntu@ip-172-31-19-43:~$ chmod ugo-r terraform/
ubuntu@ip-172-31-19-43:~$ ls -lrth
total 24K
d-wx-wx--x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
--w--w--w- 1 ubuntu ubuntu    0 Jun 16 17:10 app.go
-rw-rw-rw- 1 ubuntu ubuntu    0 Jun 16 17:16 ecomapp.py
ubuntu@ip-172-31-19-43:~$
```

7. General Scenario: **If you remove the read permission for a file then it's then there is no use with only write permission even though you can write into it but you can not read it.**

```
ubuntu@ip-172-31-19-43: ~
ubuntu@ip-172-31-19-43:~$ ls -lth
total 28K
--w--w--w- 1 ubuntu ubuntu   31 Jun 16 17:37 app.go
-rw-rw-rw- 1 ubuntu ubuntu    0 Jun 16 17:16 ecomapp.py
-rw-r--r-- 1 ubuntu ubuntu   52 Jun 16 16:55 prog.java
-rw-r--r-- 1 ubuntu ubuntu   44 Jun 16 16:49 file2.py
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:45 file1.txt
-rw-rw-r-- 1 ubuntu ubuntu  313 Jun 16 16:25 applog.txt
-rw-rw-r-- 1 ubuntu ubuntu    0 Jun 16 16:22 database-log.txt
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 AWS
drwxrwxr-x 2 ubuntu ubuntu 4.0K Jun 16 16:22 jenkins
d-wx-wx--x 2 ubuntu ubuntu 4.0K Jun 16 16:22 terraform
ubuntu@ip-172-31-19-43:~$ cat > app.go
asdadadad
qeeqqeq
bxbxbbb
ubuntu@ip-172-31-19-43:~$ cat app.go
cat: app.go: Permission denied
ubuntu@ip-172-31-19-43:~$
```

Numeric Mode:

In numeric mode, you use a three-digit number to specify permissions, with each digit representing the permissions for the user, group, and others. Each permission is represented by a specific value:

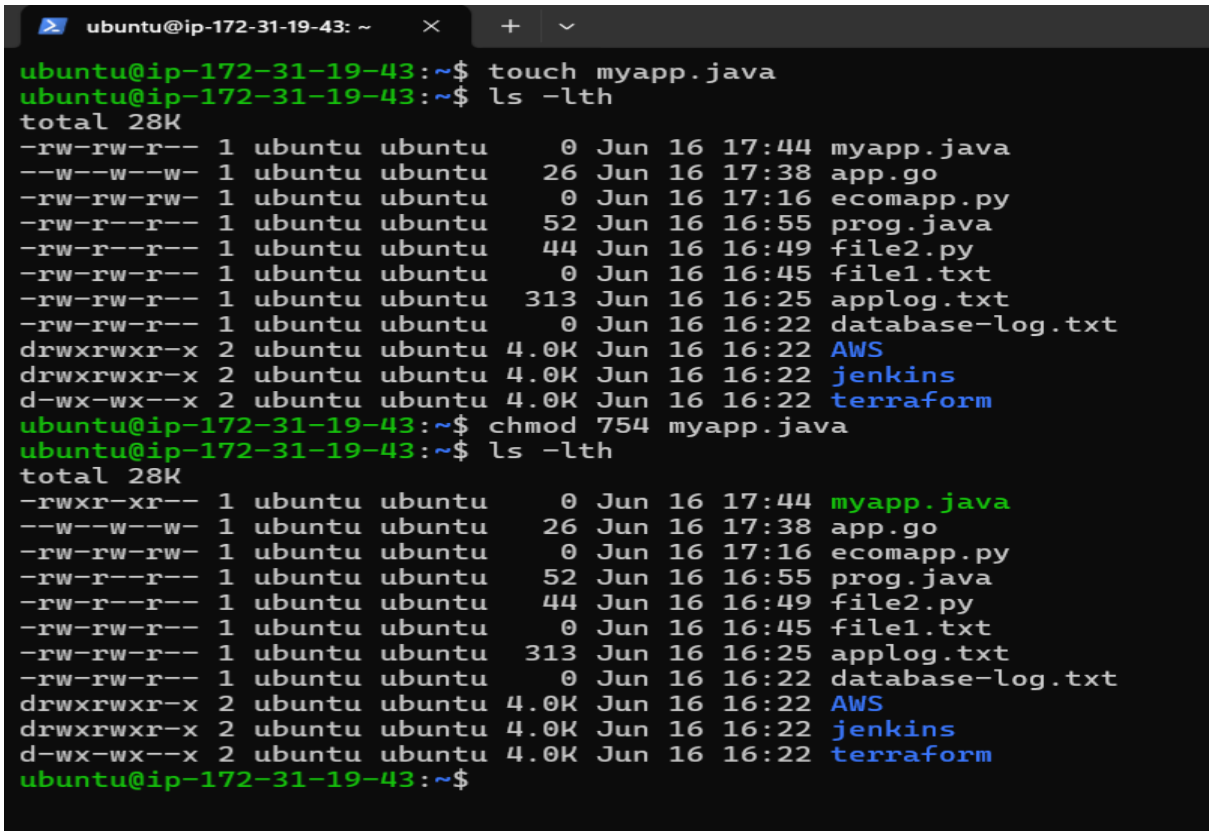
- r (read) = 4
- w (write) = 2
- x (execute) = 1

These values are summed to get the desired permissions. For example:

- 7 (rwx) = 4 (read) + 2 (write) + 1 (execute)
- 6 (rw-) = 4 (read) + 2 (write)
- 5 (r-x) = 4 (read) + 1 (execute)
- 4 (r--) = 4 (read)

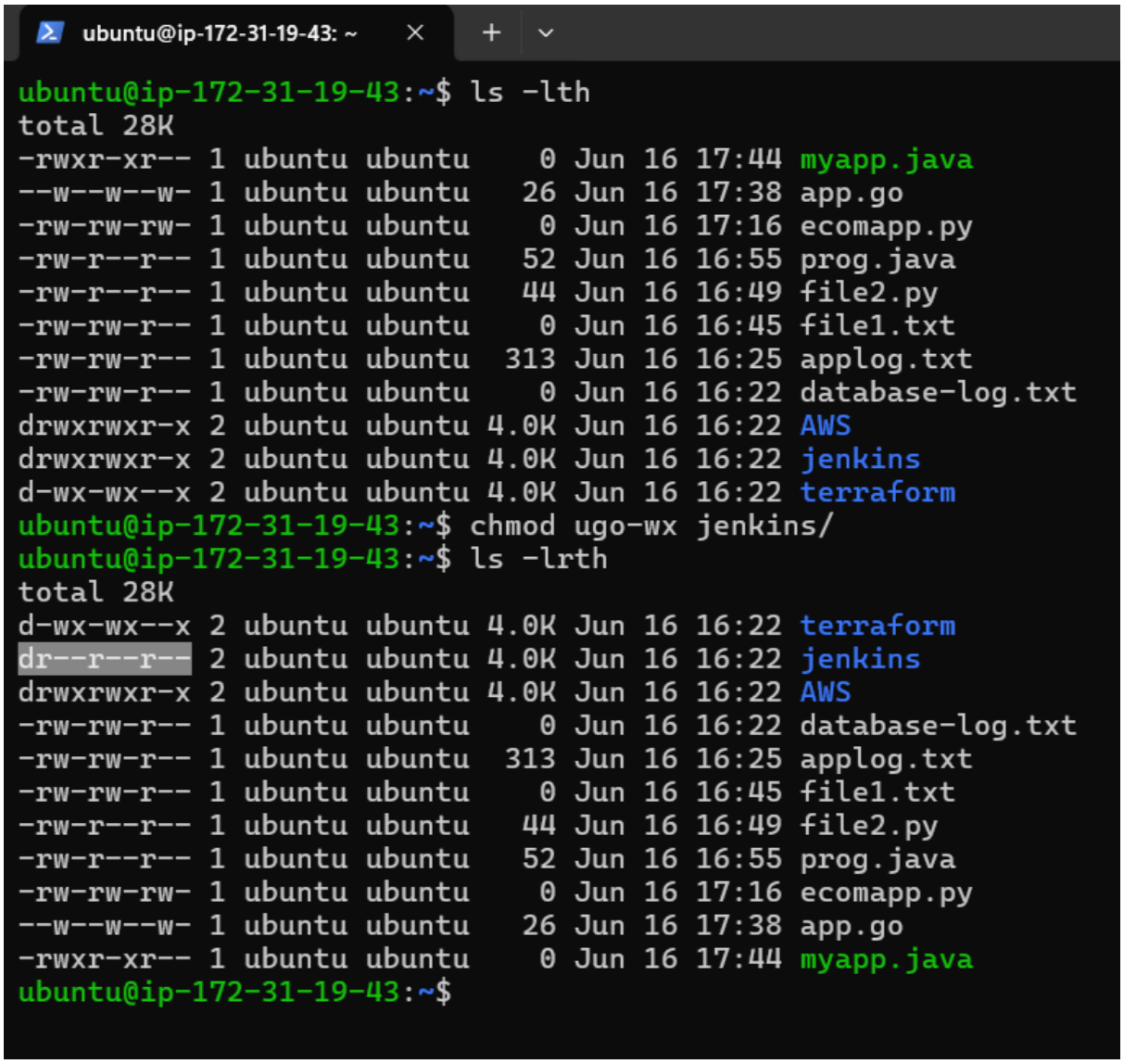
1. **Set read, write, and execute for the owner; read and execute for the group; and read-only for others**:

   chmod 754 myapp.java



2. **Remove write and execute permission for user, group and others regarding a Directory (Ex: jenkins):**

   chmod ugo-wx jenkins



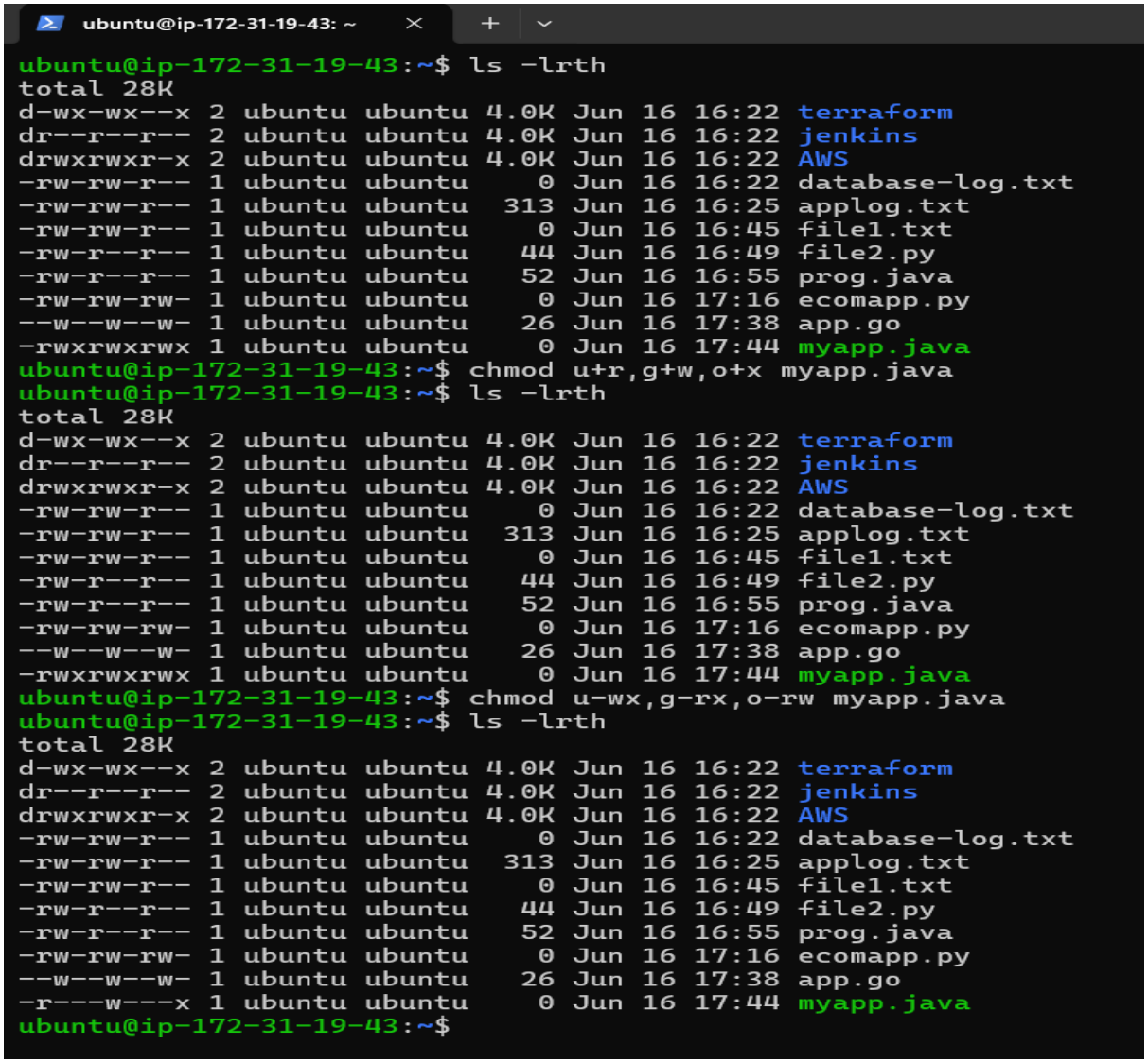
For example, if a file has (rwx,rwx,rwx) permission for it. Now you want to change it to (r--,-w-,--x)

If you change permissions in numeric mode, it will override the current permissions and add new permissions as per the given values.
(Ex: “chmod 421 myapp.java” will give you as you required)

If you do the same in symbolic mode it will check the only that particular permission and change and remaining permissions are will keep same as previous.

(Ex: “chmod u+r,g+w,o+x myapp.java” will not change as per your requirement)
For this you have to give as “chmod u-wx,g-rx,o-rw myapp.java”

So we have to be very careful while using chmod command in numeric mode



21. chown:

The chown command in Linux operating systems is used to change the owner and group ownership of a file or directory.
Syntax:

chown [OPTIONS] USER[:GROUP] FILE...

- **USER: The new owner of the file.**
- **GROUP**: The new group of the file. This is optional.
- **FILE**: The file or directory whose ownership you want to change.

Examples

1. **Change the owner of a file**:

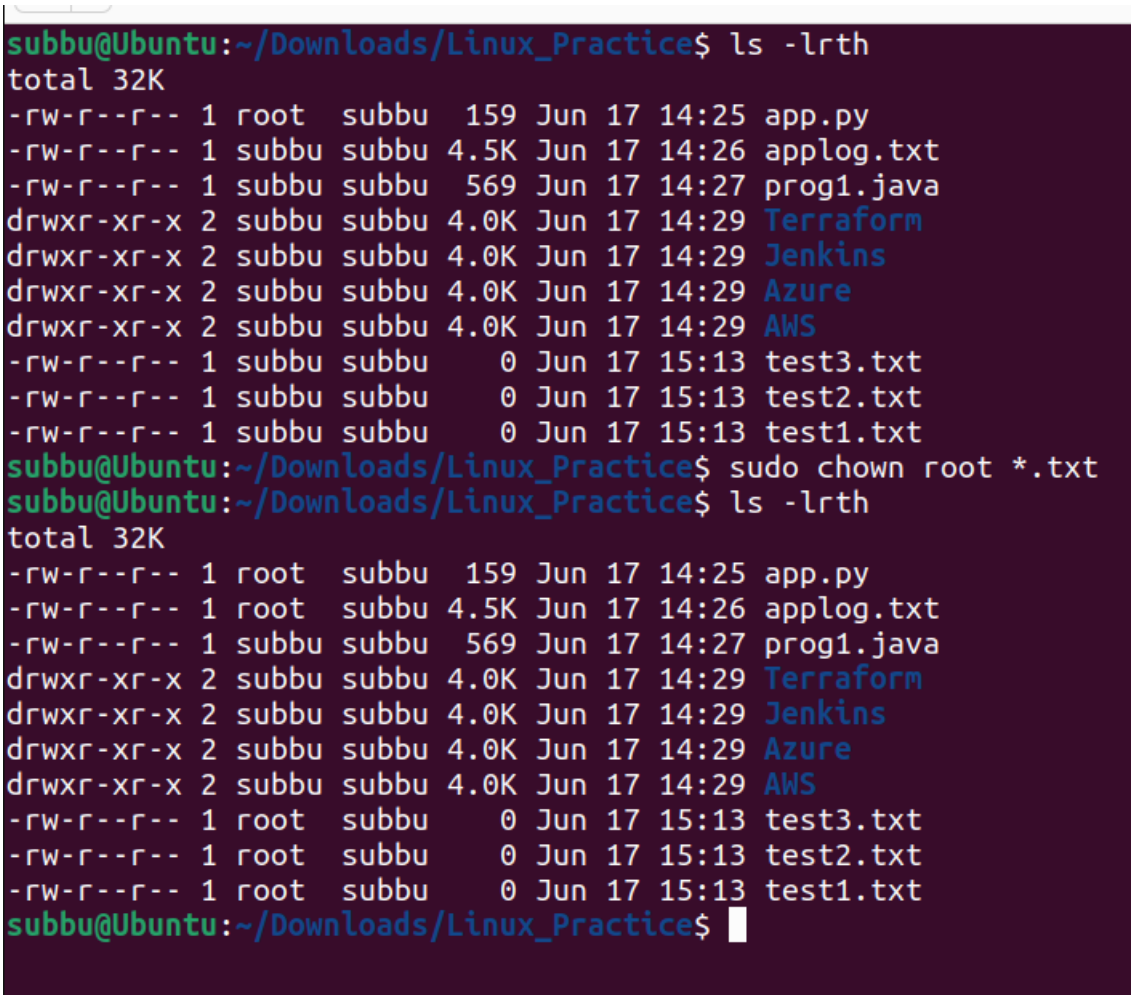
Ex: sudo chown root app.py

```
subbu@Ubuntu:~/Downloads/Linux_Practice$ ls -lrth
total 32K
-rw-r--r-- 1 subbu subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Terraform
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Jenkins
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Downloads/Linux_Practice$ chown root app.py
chown: changing ownership of 'app.py': Operation not permitted
subbu@Ubuntu:~/Downloads/Linux_Practice$ sudo chown root app.py
[sudo] password for subbu:
subbu@Ubuntu:~/Downloads/Linux_Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Terraform
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Jenkins
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Downloads/Linux_Practice$
```

2. Change the owner of a particular type of files at a time:

Ex. sudo chown root *.txt

This command is used to change the ownership of all .txt files in the current directory to the user root.



3. Change the group of a file

Ex: **sudo chown :root applog.txt**

This command is used to change the group ownership of the file applog.txt to the group root, while keeping the current file owner unchanged.

```
subbu@Ubuntu:~/Downloads/Linux_Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Terraform
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Jenkins
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 AWS
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test3.txt
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test2.txt
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test1.txt
subbu@Ubuntu:~/Downloads/Linux_Practice$ sudo chown :root prog1.java
subbu@Ubuntu:~/Downloads/Linux_Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Terraform
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Jenkins
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 AWS
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test3.txt
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test2.txt
-rw-r--r-- 1 root  subbu    0 Jun 17 15:13 test1.txt
subbu@Ubuntu:~/Downloads/Linux_Practice$
```

4. **Change the owner and group of a Directory**:

Ex: sudo chown -R root:root AWS/

This command is used to change the owner and group of the directory AWS/ and all its contents (files and subdirectories) to root.

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root   569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$ sudo chown -R root:root AWS/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root   569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$
```

5. **Change the owner and group of a file**:

Ex: sudo chown root:root app.py

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 subbu subbu  159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root   569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$ sudo chown root:root app.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root   569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$
```

22. chgrp:

The chgrp command in Linux operating systems is used to change the group ownership of a file or directory.

Basic Syntax:

chgrp [OPTIONS] GROUP FILE...

- GROUP: The new group of the file.
- FILE: The file or directory whose group ownership you want to change.

Examples:

1. Change the group of a file:

Ex: **sudo chgrp subbu prog1.java**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu root   569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$ sudo chgrp subbu prog1.java
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$
```

1. Change the group of a directory:

Ex: **sudo chgrp root Azure/**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu subbu 4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$ sudo chgrp root Azure/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxr-xr-x 2 subbu root  4.0K Jun 17 14:29 Azure
drwxr-xr-x 2 root  root  4.0K Jun 17 14:29 AWS
subbu@Ubuntu:~/Linux-Practice$
```

23. diff:

The diff command in Linux is used to compare the contents of two files line by line. It outputs the differences between the files, which can be useful for identifying changes or discrepancies.

Basic Syntax:

diff [OPTIONS] FILE1 FILE2

- **FILE1**: The first file to compare.
- **FILE2**: The second file to compare.

Ex: **diff app.py prod.py**

```
subbu@Ubuntu:~/Linux-Practice$ cat app.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$ cat prod.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num3

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$ diff app.py prod.py
6c6
< sum = num1 + num2
---
> sum = num1 + num3
subbu@Ubuntu:~/Linux-Practice$
```

24. file:

The file command in Linux operating systems is used to determine the type of a file.

Basic Syntax:

file [OPTIONS] FILE...

FILE: The file or files you want to check.

Ex: **file app.py**

file applog.txt

file prog1.java

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
-rw-rw-r-- 1 subbu subbu  159 Jun 17 16:01 prod.py
subbu@Ubuntu:~/Linux-Practice$ file app.py
app.py: ASCII text
subbu@Ubuntu:~/Linux-Practice$ file applog.txt
applog.txt: Unicode text, UTF-8 text, with very long lines (440)
subbu@Ubuntu:~/Linux-Practice$ file prog1.java
prog1.java: Java source, ASCII text
subbu@Ubuntu:~/Linux-Practice$
```

Multiple Files:

Ex: **file prog1.java prod.py**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
-rw-rw-r-- 1 subbu subbu  159 Jun 17 16:01 prod.py
subbu@Ubuntu:~/Linux-Practice$ file prog1.java prod.py
prog1.java: Java source, ASCII text
prod.py:    ASCII text
subbu@Ubuntu:~/Linux-Practice$
```

25. mv:

The mv command in Linux operating system is used to move or rename files and directories
Basic Syntax:

mv [OPTIONS] SOURCE DESTINATION

- **SOURCE**: The file or directory you want to move or rename.
- **DESTINATION**: The new location or new name for the file or directory.

1. **Move Multiple Files to a Directory**:

Ex: mv app.py prod.py PYTHON/

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  root   159 Jun 17 14:25 app.py
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
-rw-rw-r-- 1 subbu subbu  159 Jun 17 16:01 prod.py
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 PYTHON
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
subbu@Ubuntu:~/Linux-Practice$ mv app.py prod.py PYTHON/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth PYTHON/
total 8.0K
-rw-r--r-- 1 root  root  159 Jun 17 14:25 app.py
-rw-rw-r-- 1 subbu subbu 159 Jun 17 16:01 prod.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
subbu@Ubuntu:~/Linux-Practice$
```

2.**Rename a File**:

Ex: **mv applog.txt errorlog.txt**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 applog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
subbu@Ubuntu:~/Linux-Practice$ mv applog.txt errorlog.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
subbu@Ubuntu:~/Linux-Practice$
```

3. Move a Directory:

Ex: **mv PYTHON/ TERRAFORM/**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
subbu@Ubuntu:~/Linux-Practice$ mv PYTHON/ TERRAFORM/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth TERRAFORM/
total 4.0K
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 14:27 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:18 JAVA
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:35 TERRAFORM
subbu@Ubuntu:~/Linux-Practice$
```

26. cp:

The cp command in Linux operating system is used to copy files and directories.
Basic Syntax:

cp [OPTIONS] SOURCE DESTINATION

- **SOURCE**: The file or directory you want to copy.
- **DESTINATION**: The destination where the file or directory should be copied.

1. Copy a File:

Ex: cp file1.txt JAVA/

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:35 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:39 JAVA
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
subbu@Ubuntu:~/Linux-Practice$ cp file1.txt JAVA/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth JAVA/
total 4.0K
-rw-r--r-- 1 subbu subbu 569 Jun 17 16:38 prog1.java
-rw-rw-r-- 1 subbu subbu   0 Jun 17 16:47 file1.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:35 TERRAFORM
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
subbu@Ubuntu:~/Linux-Practice$
```

2. **Copy and Rename a File**:

Ex: **cp errorlog.txt TERRAFORM/applog.txt**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:35 TERRAFORM
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
subbu@Ubuntu:~/Linux-Practice$ cp errorlog.txt TERRAFORM/applog.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth TERRAFORM/
total 12K
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 16:50 applog.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 20K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:50 TERRAFORM
subbu@Ubuntu:~/Linux-Practice$
```

3. **Copy Multiple Files to a Directory**:

Ex: **cp file2.txt prog1.java AWS/**

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:50 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
subbu@Ubuntu:~/Linux-Practice$ cp file2.txt prog1.java AWS/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth AWS/
total 4.0K
-rw-r--r-- 1 subbu subbu 569 Jun 17 16:52 prog1.java
-rw-rw-r-- 1 subbu subbu   0 Jun 17 16:52 file2.txt
subbu@Ubuntu:~/Linux-Practice$
```

4. **Copy a Directory and Its Contents**:

Ex: **cp -r JAVA/ TERRAFORM/**

The command cp -r JAVA/ TERRAFORM/ is used to copy the contents of the JAVA/ directory into the TERRAFORM/ directory.

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file2.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 17 16:46 file1.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 3 subbu subbu 4.0K Jun 17 16:50 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
subbu@Ubuntu:~/Linux-Practice$ ls -lrth JAVA/
total 4.0K
-rw-r--r-- 1 subbu subbu 569 Jun 17 16:38 prog1.java
-rw-rw-r-- 1 subbu subbu   0 Jun 17 16:47 file1.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth TERRAFORM/
total 12K
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 16:50 applog.txt
subbu@Ubuntu:~/Linux-Practice$ cp -r JAVA/ TERRAFORM/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth TERRAFORM/
total 16K
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 16:50 applog.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:55 JAVA
subbu@Ubuntu:~/Linux-Practice$
```

Difference Between cp and mv commands:

cp: copy & paste

mv: cut & paste

Key Differences:

1. **Functionality**:
    - cp copies files or directories from one location to another, leaving the source files unchanged.
    - mv moves files or directories from one location to another, or renames them, removing the source files.
2. **Use Cases**:
    - Use cp when you want to duplicate files or directories.
    - Use mv when you want to relocate files or directories or rename them.
3. **Impact on Source**:
    - cp retains the original files/directories at the source location.
    - mv removes the original files/directories from the source location.

27. ln: ‘ln’ command

The ln command in Linux operating systems is used to create links between files. It can create both hard links and symbolic links.

Basic Syntax:
ln [OPTIONS] SOURCE [LINK_NAME]

- **SOURCE**: The existing file or directory you want to link to.
- **LINK_NAME**: The name of the link you want to create. If not specified, the link will be created in the current directory with the same name as the source file.

Hardlink:
ln original.txt link.txt

1. **Data synchronization will happen**
2. **Inode number of both files are same**
3. **Even original file is deleted hardlink file will remains**
4. **Hardlink will not useful for directories**

Softlink:
ln -s original.txt link.txt

5. **Data synchronization will happen**
6. **Inode number of both files are different**
7. **If original file is deleted softlink file will not available**
8. **Softlink are useful for both files and directories**

Examples:

1. **Create a Hard Link**:

    Ex: **ln app.py dev-data/applink.py**

```
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
-rw-r--r-- 1 subbu subbu  159 Jun 17 17:15 app.py
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:17 dev-data
subbu@Ubuntu:~/Linux-Practice$ ln app.py dev-data/applink.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
-rw-r--r-- 2 subbu subbu  159 Jun 17 17:15 app.py
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:18 dev-data
subbu@Ubuntu:~/Linux-Practice$ ls -lrth dev-data/
total 4.0K
-rw-r--r-- 2 subbu subbu 159 Jun 17 17:15 applink.py
subbu@Ubuntu:~/Linux-Practice$ cat app.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$ cat dev-data/applink.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$
```

Hardlinks: 1. Data synchronization will happen

```
subbu@Ubuntu:~/Linux-Practice$ nano app.py
subbu@Ubuntu:~/Linux-Practice$ cat app.py
# Python3 program to add three numbers
num1 = 15
num2 = 12
num3 = 25
# Adding three nos
sum = num1 + num2 + num3

# printing values
print("Sum of", num1, "and", num2 ,"and", num3, "is", sum)
subbu@Ubuntu:~/Linux-Practice$ cat dev-data/applink.py
# Python3 program to add three numbers
num1 = 15
num2 = 12
num3 = 25
# Adding three nos
sum = num1 + num2 + num3

# printing values
print("Sum of", num1, "and", num2 ,"and", num3, "is", sum)
subbu@Ubuntu:~/Linux-Practice$
```

Hardlinks: 2. Inode number of both files are same

```
subbu@Ubuntu:~/Linux-Practice$ ls -lirth
total 32K
923242 -rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
923238 -rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
922775 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
923355 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
923340 drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
923342 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:18 dev-data
923341 -rw-r--r-- 2 subbu subbu  191 Jun 17 17:22 app.py
subbu@Ubuntu:~/Linux-Practice$ ls -lirth dev-data/applink.py
923341 -rw-r--r-- 2 subbu subbu 191 Jun 17 17:22 dev-data/applink.py
subbu@Ubuntu:~/Linux-Practice$
```

Hardlinks: 3. Even original file is deleted hardlink file will remains

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:18 dev-data
-rw-r--r-- 2 subbu subbu  191 Jun 17 17:22 app.py
subbu@Ubuntu:~/Linux-Practice$ rm app.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth dev-data/applink.py
-rw-r--r-- 1 subbu subbu 191 Jun 17 17:22 dev-data/applink.py
subbu@Ubuntu:~/Linux-Practice$ cat dev-data/applink.py
# Python3 program to add three numbers
num1 = 15
num2 = 12
num3 = 25
# Adding three nos
sum = num1 + num2 + num3

# printing values
print("Sum of", num1, "and", num2 ,"and", num3, "is", sum)
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:18 dev-data
subbu@Ubuntu:~/Linux-Practice$
```

Hardlinks: 4.Hardlink will not useful for directories:

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:18 dev-data
subbu@Ubuntu:~/Linux-Practice$ ln AWS/ dev-data/
ln: AWS/: hard link not allowed for directory
subbu@Ubuntu:~/Linux-Practice$
```

2. Create a Soft Link:

Ex: ln -s app.py softlink.py

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
-rw-r--r-- 1 subbu subbu  159 Jun 17 17:59 app.py
subbu@Ubuntu:~/Linux-Practice$ ln -s app.py softlink.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 32K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
-rw-r--r-- 1 subbu subbu  159 Jun 17 17:59 app.py
lrwxrwxrwx 1 subbu subbu    6 Jun 17 17:59 softlink.py -> app.py
subbu@Ubuntu:~/Linux-Practice$
```

```
subbu@Ubuntu:~/Linux-Practice$ cat app.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$ cat softlink.py
# Python3 program to add two numbers
num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of", num1, "and", num2 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$
```

Softlinks: 1. Data synchronization will happen

```
subbu@Ubuntu:~/Linux-Practice$ nano app.py
subbu@Ubuntu:~/Linux-Practice$ cat app.py
# Python3 program to add two numbers
num2 = 15
num3 = 12

# Adding two nos
sum = num2 + num3

# printing values
print("Sum of", num2, "and", num3 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$ cat softlink.py
# Python3 program to add two numbers
num2 = 15
num3 = 12

# Adding two nos
sum = num2 + num3

# printing values
print("Sum of", num2, "and", num3 , "is", sum)
subbu@Ubuntu:~/Linux-Practice$
```

Softlinks: 2. Inode number of both files are different

```
subbu@Ubuntu:~/Linux-Practice$ ls -lirth
total 32K
923242 -rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
923238 -rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
922775 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
923355 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
923340 drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
923342 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
923241 lrwxrwxrwx 1 subbu subbu    6 Jun 17 17:59 softlink.py -> app.py
923356 -rw-r--r-- 1 subbu subbu  159 Jun 17 18:03 app.py
subbu@Ubuntu:~/Linux-Practice$
```

Softlinks: 3. If original file is deleted softlink file will not available

```
subbu@Ubuntu:~/Linux-Practice$ ls -lirth
total 32K
923242 -rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
923238 -rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
922775 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
923355 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
923340 drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
923342 drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
923241 lrwxrwxrwx 1 subbu subbu    6 Jun 17 17:59 softlink.py -> app.py
923356 -rw-r--r-- 1 subbu subbu  159 Jun 17 18:03 app.py
subbu@Ubuntu:~/Linux-Practice$ rm app.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
lrwxrwxrwx 1 subbu subbu    6 Jun 17 17:59 softlink.py -> app.py
subbu@Ubuntu:~/Linux-Practice$ cat softlink.py
cat: softlink.py: No such file or directory
subbu@Ubuntu:~/Linux-Practice$
```

Softlinks: 4. Softlink are useful for both files and directories

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 17:58 dev-data
subbu@Ubuntu:~/Linux-Practice$ ln -s AWS/ dev-data/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth dev-data/
total 4.0K
-rw-r--r-- 1 subbu subbu 191 Jun 17 17:22 applink.py
lrwxrwxrwx 1 subbu subbu   4 Jun 17 18:09 AWS -> AWS/
subbu@Ubuntu:~/Linux-Practice$ ls -lrth dev-data/AWS
lrwxrwxrwx 1 subbu subbu 4 Jun 17 18:09 dev-data/AWS -> AWS/
```

28. echo :

The echo command in Linux operating systems is used to display text or variables on the terminal. It is commonly used in shell scripts and interactive terminal sessions to output information.

Basic Syntax:

echo [OPTIONS] [STRING] [...]

STRING: The text you want to display. If no STRING is specified, echo will simply output a newline character.

Practical Use Cases

- **Script Output**: Use echo to provide feedback or output in shell scripts.
- **Debugging**: Display variable values or intermediate results to help with debugging scripts.
- **User Interaction**: Display prompts or messages for user interaction in scripts or interactive sessions.

1. Display Text:

Ex: echo "Hello, World!"

```
subbu@Ubuntu:~/Linux-Practice$ echo "Hello World"
Hello World
subbu@Ubuntu:~/Linux-Practice$
```

2. Display Multiple Strings:

Ex: echo "Today is" $(date)

```
subbu@Ubuntu:~/Linux-Practice$ echo "Today is" $(date)
Today is Monday 17 June 2024 06:16:47 PM IST
subbu@Ubuntu:~/Linux-Practice$
```

3. Display Variables:

Ex: name="John"

echo "Hello, $name!"

```
subbu@Ubuntu:~/Linux-Practice$ name="John"
echo "Hello, $name!"
Hello, John!
subbu@Ubuntu:~/Linux-Practice$
```

29. find:

The find command in Unix-based operating systems is a powerful tool used to search for files and directories within a directory hierarchy. It is highly versatile, allowing users to locate files based on a variety of criteria such as name, type, size, modification time, and permissions.

Basic Syntax:
find [path] [expression]

Examples:
1. Find files by name:
Ex: find . -type f -name app.py
This command is used to search for a file named app.py in the current directory and all its subdirectories.

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 18:09 dev-data
subbu@Ubuntu:~/Linux-Practice$ find . -type f -name app.py
./TERRAFORM/PYTHON/app.py
subbu@Ubuntu:~/Linux-Practice$
```

2. Find files by extension (To search for particular type of files):

Ex: **find . -type f -name "*.py"**
This command is used to search for all .py files in the current directory and its subdirectories.

```
subbu@Ubuntu:~/Linux-Practice$ find . -type f -name "*.py"
./TERRAFORM/PYTHON/prod.py
./TERRAFORM/PYTHON/app.py
./dev-data/applink.py
subbu@Ubuntu:~/Linux-Practice$
```

3. Find directories by name:

Ex: find . -type d -name "PYTHON"

```
subbu@Ubuntu:~/Linux-Practice$ ls -ltrh
total 28K
-rw-r--r-- 1 root  subbu 4.5K Jun 17 14:26 errorlog.txt
-rw-r--r-- 1 subbu subbu  569 Jun 17 16:40 prog1.java
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:47 JAVA
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:52 AWS
drwxrwxr-x 4 subbu subbu 4.0K Jun 17 16:55 TERRAFORM
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 18:09 dev-data
subbu@Ubuntu:~/Linux-Practice$ ls -ltrh TERRAFORM/
total 16K
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:19 PYTHON
-rw-r--r-- 1 subbu subbu 4.5K Jun 17 16:50 applog.txt
drwxrwxr-x 2 subbu subbu 4.0K Jun 17 16:55 JAVA
subbu@Ubuntu:~/Linux-Practice$ find . -type d -name "PYTHON"
./TERRAFORM/PYTHON
subbu@Ubuntu:~/Linux-Practice$
```

4. Find files by size:

Ex: find . -size +100M

```
subbu@Ubuntu:~/Linux-Practice$ find /home/subbu/ -size +500M
/home/subbu/.docker/desktop/vms/0/data/Docker.raw
subbu@Ubuntu:~/Linux-Practice$
```

5. Find files by modification time:

Ex: find . -mtime -7

Find files modified in the last 7 days.

```
subbu@Ubuntu:~/Linux-Practice$ find . -mtime -7
.
./errorlog.txt
./prog1.java
./TERRAFORM
./TERRAFORM/JAVA
./TERRAFORM/JAVA/prog1.java
./TERRAFORM/JAVA/file1.txt
./TERRAFORM/PYTHON
./TERRAFORM/PYTHON/prod.py
./TERRAFORM/PYTHON/app.py
./TERRAFORM/applog.txt
./JAVA
./JAVA/prog1.java
./JAVA/file1.txt
./AWS
./AWS/file2.txt
./AWS/prog1.java
./dev-data
./dev-data/applink.py
./dev-data/AWS
subbu@Ubuntu:~/Linux-Practice$
```

6. Find files by permission:

Ex: find . -perm 644

```
subbu@Ubuntu:~/Linux-Practice$ find . -perm 755
subbu@Ubuntu:~/Linux-Practice$ find . -perm 777
./dev-data/AWS
subbu@Ubuntu:~/Linux-Practice$ find . -perm 644
./errorlog.txt
./prog1.java
./TERRAFORM/JAVA/prog1.java
./TERRAFORM/PYTHON/app.py
./TERRAFORM/applog.txt
./JAVA/prog1.java
./AWS/prog1.java
./dev-data/applink.py
subbu@Ubuntu:~/Linux-Practice$
```

7. Combining Multiple Criteria:

Example: Find .txt files larger than 1MB that were modified in the last 30 days.

find . -name "*.txt" -size +1M -mtime -60

```
subbu@Ubuntu:~$ find . -name "*.txt" -size +1M -mtime -60
./Desktop/DOCKER/DOCKER DESKTOP/VS CODE/usr/share/code/resources/app/ThirdPartyNotices.txt
./.vscode/extensions/ms-python.vscode-pylance-2024.6.1/NOTICE.txt
subbu@Ubuntu:~$ find . -name "*.txt" -size +1M -mtime -30
./.vscode/extensions/ms-python.vscode-pylance-2024.6.1/NOTICE.txt
subbu@Ubuntu:~$
```

30. sed:

The sed command, short for "stream editor," is a powerful tool for text manipulation in Unix-based systems. It is commonly used for finding and replacing text, but it can also perform a variety of other text processing tasks such as inserting, deleting, and transforming text in a file or stream.

Basic Syntax:

sed [options] 'script' [input_file]

Common Uses and Examples:

1. Substitute (find and replace) text:

sed 's/old_text/new_text/' filename

Ex: sed 's/log/error/' errorlog.txt

Example: Replace the first occurrence of "log" with "error" in each line of errorlog.txt.

```
subbu@Ubuntu:~/Linux-Practice$ cat errorlog.txt  |grep "log"
computer software application log when an event occurs.
web server database instance log successful backup completion,
IT teams typically use application log data to track
log for more than troubleshooting.
They are using logged events to investigate security incidents,
we'll look at what an application log is, the different
application log, and how a centralized log management tool
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed 's/log/error/' errorlog.txt | grep -e "error" -e "log"
computer software application error when an event occurs.
web server database instance error successful backup completion,
IT teams typically use application error data to track
error for more than troubleshooting.
They are using errorged events to investigate security incidents,
we'll look at what an application error is, the different
application error, and how a centralized log management tool
What is an Application error? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$
```

2. Replace text globally (all occurrences in each line):

sed 's/old_text/new_text/g' filename

Ex: sed 's/log/error/g' errorlog.txt

Example: Replace all occurrence of "log" with "error" in each line of errorlog.txt.

```
subbu@Ubuntu:~/Linux-Practice$ cat errorlog.txt | grep "log"
computer software application log when an event occurs.
web server database instance log successful backup completion,
IT teams typically use application log data to track
log for more than troubleshooting.
They are using logged events to investigate security incidents,
we'll look at what an application log is, the different
application log, and how a centralized log management tool
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed 's/log/error/g' errorlog.txt  | grep "error"
computer software application error when an event occurs.
web server database instance error successful backup completion,
IT teams typically use application error data to track
error for more than troubleshooting.
They are using errorged events to investigate security incidents,
we'll look at what an application error is, the different
application error, and how a centralized error management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$
```

3. Copying result into a new file:

sed ‘s/old_text/new_text/g’ filename > newfile

Ex: sed 's/log/error/g' errorlog.txt > dumpfile.txt

```
subbu@Ubuntu:~/Linux-Practice$ cat errorlog.txt | grep "log"
computer software application log when an event occurs.
web server database instance log successful backup completion,
IT teams typically use application log data to track
log for more than troubleshooting.
They are using logged events to investigate security incidents,
we'll look at what an application log is, the different
application log, and how a centralized log management tool
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed 's/log/log-file/g' errorlog.txt >dumpfile.txt
subbu@Ubuntu:~/Linux-Practice$ cat dumpfile.txt | grep "log-file"
computer software application log-file when an event occurs.
web server database instance log-file successful backup completion,
IT teams typically use application log-file data to track
log-file for more than troubleshooting.
They are using log-fileged events to investigate security incidents,
we'll look at what an application log-file is, the different
application log-file, and how a centralized log-file management tool
What is an Application log-file? Software applications generate log-file when
subbu@Ubuntu:~/Linux-Practice$
```

4. Adding duplicates according to the changes are happend to get how many lines are effected with the changes:

sed ‘s/old_text/new_text/p’ filename

Ex: cat applog.txt | wc -l

sed 's/log/error/p' applog.txt | wc -l

```
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application log when an event occurs.
web server database instance successful backup completion,
IT teams application log data to track
more than troubleshooting.
They are using log events to investigate security incidents,
track customer behavior,
we'll look at what an application log is, the different
application how a centralized management tool
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed 's/log/error/p' applog.txt
computer software application error when an event occurs.
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application error data to track
IT teams application error data to track
more than troubleshooting.
They are using error events to investigate security incidents,
They are using error events to investigate security incidents,
track customer behavior,
we'll look at what an application error is, the different
we'll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate log when
What is an Application error? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt  | wc-l
wc-l: command not found
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt  | wc -l
9
subbu@Ubuntu:~/Linux-Practice$ sed 's/log/error/p' applog.txt  |wc -l
14
subbu@Ubuntu:~/Linux-Practice$ 14-9=5 in 5 lines changes are happend
```

4. Print only lines that match a pattern:

sed -n '/pattern/p' filename

Ex: sed -n '/log/p' applog.txt

```
subbu@Ubuntu:~/Linux-Practice$ cat -n applog.txt
     1  computer software application log when an event occurs.
     2  web server database instance successful backup completion,
     3  IT teams application log data to track
     4  more than troubleshooting.
     5  They are using log events to investigate security incidents,
     6  track customer behavior,
     7  we'll look at what an application log is, the different
     8  application how a centralized management tool
     9  What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed -n '/log/p' applog.txt
computer software application log when an event occurs.
IT teams application log data to track
They are using log events to investigate security incidents,
we'll look at what an application log is, the different
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$
```

5. Edit a file in-place (modify the original file):

sed -i 's/old_text/new_text/g' filename

Ex: sed -i 's/log/error/g' applog.txt

```
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application log when an event occurs.
web server database instance successful backup completion,
IT teams application log data to track
more than troubleshooting.
They are using log events to investigate security incidents,
track customer behavior,
we'll look at what an application log is, the different
application how a centralized management tool
What is an Application log? Software applications generate log when
subbu@Ubuntu:~/Linux-Practice$ sed -i 's/log/error/g' applog.txt
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application error data to track
more than troubleshooting.
They are using error events to investigate security incidents,
track customer behavior,
we'll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$ using "sed -i" we can Edit a file in-place (modify the original file)
```

6. Delete lines that match a pattern:

sed '/pattern/d' filename

Ex: sed '/error/d' applog.txt

It delete lines that contain "error" in applog.txt.

```
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application error data to track
more than troubleshooting.
They are using error events to investigate security incidents,
track customer behavior,
we'll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$ sed '/error/d' applog.txt
web server database instance successful backup completion,
more than troubleshooting.
track customer behavior,
application how a centralized management tool
subbu@Ubuntu:~/Linux-Practice$
```

7. Replace text only on specific lines:

sed 'line_number s/old_text/new_text/' filename

Ex: sed '3 s/error/log/' applog.txt

It will replace the first occurrence of "error with "log" on line 3 of applog.txt

```
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application error data to track
more than troubleshooting.
They are using error events to investigate security incidents,
track customer behavior,
we'll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$ sed '3 s/error/log/' applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application log data to track
more than troubleshooting.
They are using error events to investigate security incidents,
track customer behavior,
we'll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$
```

8. Replace text only on specified no of lines:

sed 'From_line_number, To_line_number s/old_text/new_text/' filename

Ex: sed ‘3,6 s/error/log/’ applog.txt

It will replace the first occurrence of "error with "log" from the 3rd line to 6th line of applog.txt

```
subbu@Ubuntu:~/Linux-Practice$ cat applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application error data to track
more than troubleshooting.
They are using error events to investigate security incidents,
track customer behavior,
we’ll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$ sed '3,6 s/error/log/' applog.txt
computer software application error when an event occurs.
web server database instance successful backup completion,
IT teams application log data to track
more than troubleshooting.
They are using log events to investigate security incidents,
track customer behavior,
we’ll look at what an application error is, the different
application how a centralized management tool
What is an Application error? Software applications generate error when
subbu@Ubuntu:~/Linux-Practice$
```

31. who:

The who command in Unix-based operating systems is used to display information about users who are currently logged into the system. It provides details such as the username, terminal line, login time, and the host from which the user is logged in.

Basic Use:

who

```
subbu@Ubuntu:~/Linux-Practice$ who
subbu    :1           2024-06-17 14:56 (:1)
subbu@Ubuntu:~/Linux-Practice$
```

32. w:

The w command in Unix-based operating systems is used to display information about the users currently logged into the system and their processes. It provides a more detailed view than the who command, including the uptime of the system, the number of users, the load averages, and the activity of each logged-in user.

```
subbu@Ubuntu:~/Linux-Practice$ w
 20:29:52 up  5:50,  1 user,  load average: 1.34, 1.14, 1.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
subbu    :1       :1               14:56    ?xdm?   1:03m  0.00s /usr/libexec/gdm-x-session --run-script env GNOME_SHELL_SESSION_MO
subbu@Ubuntu:~/Linux-Practice$
```

Load Average? : Ex: load average: 1.34, 1.14, 1.06

33. uptime:

The uptime command in Unix-based operating systems provides information about how long the system has been running, the number of users currently logged in, and the system load averages for the past 1, 5, and 15 minutes.

```
subbu@Ubuntu:~/Linux-Practice$ uptime
 20:33:56 up  5:54,  1 user,  load average: 1.03, 1.15, 1.08
subbu@Ubuntu:~/Linux-Practice$
```

34. users:

The users command in Unix-based operating systems is used to display the usernames of users currently logged into the system. It provides a simple list of logged-in users without additional details.

```
subbu@Ubuntu:~/Linux-Practice$ users
subbu
subbu@Ubuntu:~/Linux-Practice$
```

35. whereis:

The whereis command in Unix-based operating systems is used to locate the binary, source, and manual page files for a specified command. It provides information about the locations of these files in the system's file hierarchy.

```
subbu@Ubuntu:~/Linux-Practice$ whereis man
man: /usr/bin/man /usr/local/man /usr/share/man /usr/share/man/man1/man.1.gz /usr/share/man/man7/man.7.gz
subbu@Ubuntu:~/Linux-Practice$ whereis grep
grep: /usr/bin/grep /usr/share/man/man1/grep.1.gz /usr/share/info/grep.info.gz
subbu@Ubuntu:~/Linux-Practice$ whereis sed
sed: /usr/bin/sed /usr/share/man/man1/sed.1.gz /usr/share/info/sed.info.gz
subbu@Ubuntu:~/Linux-Practice$
```

36. date:

The date command in Unix-based operating systems is used to display or set the current system date and time. It can also be used to format the date and time in various ways.

```
subbu@Ubuntu:~/Linux-Practice$ date
Monday 17 June 2024 08:53:39 PM IST
subbu@Ubuntu:~/Linux-Practice$
```

37. timedatectl:

The timedatectl command in Unix-based operating systems, particularly those using systemd, is used to control and query the system clock and its settings. It allows users to view and modify the system's time and date settings, including the timezone, the current time, and whether the system clock is set to UTC or local time.

```
subbu@Ubuntu:~/Linux-Practice$ timedatectl
               Local time: Mon 2024-06-17 20:51:20 IST
           Universal time: Mon 2024-06-17 15:21:20 UTC
                 RTC time: Mon 2024-06-17 15:21:20
                Time zone: Asia/Kolkata (IST, +0530)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
subbu@Ubuntu:~/Linux-Practice$
```

38. df:

The df command in Unix-based operating systems is used to display information about the disk space usage on file systems. It shows the amount of disk space used and available on mounted file systems, including hard drives, partitions, and virtual filesystems like /proc and /sys.

```
subbu@Ubuntu:~/Linux-Practice$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs            1608388     3432   1604956   1% /run
/dev/nvme0n1p6  81121736 20178840  56776140  27% /
tmpfs            8041928    22768   8019160   1% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
efivarfs             268      163       101  62% /sys/firmware/efi/efivars
tmpfs            8041928        0   8041928   0% /run/qemu
/dev/nvme0n1p7  47745772 19561808  25726160  44% /home
/dev/nvme0n1p1    262144    45236    216908  18% /boot/efi
tmpfs            1608384      152   1608232   1% /run/user/1000
/dev/nvme0n1p4  80055292 78020220   2035072  98% /media/subbu/New Volume
subbu@Ubuntu:~/Linux-Practice$
```

Ex: df -h

Human-Readable Format

```
subbu@Ubuntu:~/Linux-Practice$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1.6G  3.4M  1.6G   1% /run
/dev/nvme0n1p6   78G   20G   55G  27% /
tmpfs           7.7G   23M  7.7G   1% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
efivarfs        268K  163K  101K  62% /sys/firmware/efi/efivars
tmpfs           7.7G     0  7.7G   0% /run/qemu
/dev/nvme0n1p7   46G   19G   25G  44% /home
/dev/nvme0n1p1  256M   45M  212M  18% /boot/efi
tmpfs           1.6G  152K  1.6G   1% /run/user/1000
/dev/nvme0n1p4   77G   75G  2.0G  98% /media/subbu/New Volume
subbu@Ubuntu:~/Linux-Practice$
```

Common Options:

- -h: Display sizes in a human-readable format (e.g., kilobytes, megabytes).
- -T: Display the filesystem type (e.g., ext4, tmpfs).
- -i: Display inode information (number of used and available inodes).
- -a: Display information about all filesystems, including those with a size of 0 blocks.
- --total: Display a summary of total disk space usage at the end of the output.
- -t: Limit the output to filesystems of a specific type.

39. du:

The du command in Unix-based operating systems is used to estimate file and directory space usage. It reports the disk space used by the specified files and directories, including their subdirectories, in kilobytes by default, but it can be configured to display in other units as well.

Ex: du :

```
subbu@Ubuntu:~/Linux-Practice$ du
8       ./TERRAFORM/JAVA
12      ./TERRAFORM/PYTHON
32      ./TERRAFORM
8       ./JAVA
8       ./AWS
8       ./dev-data
72      .
subbu@Ubuntu:~/Linux-Practice$
```

du -h :

```
subbu@Ubuntu:~/Linux-Practice$ du -h
8.0K    ./TERRAFORM/JAVA
12K     ./TERRAFORM/PYTHON
32K     ./TERRAFORM
8.0K    ./JAVA
8.0K    ./AWS
8.0K    ./dev-data
72K     .
subbu@Ubuntu:~/Linux-Practice$
```

Common Options

- -h: Display sizes in a human-readable format (e.g., kilobytes, megabytes).
- -s: Display only the total for each specified directory.
- -c: Display a grand total for all specified files or directories.
- -a: Include files and directories that begin with a dot (hidden files).
- --max-depth=N: Limit the output to a specified depth of subdirectories.
- -k, -m, -g: Display sizes in kilobytes, megabytes, or gigabytes, respectively.

40. tree:

In Linux, the tree command is used to display the directory structure in a tree-like format. If tree is not installed on your system, you can install it using your package manager.

Installing tree:

Ubuntu/Debian: sudo apt-get install tree

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 8.0K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
subbu@Ubuntu:~/Linux-Practice$ tree
.
├── app.py
├── BASH
│   └── Scripts
├── dev-app.java
└── testfile.txt

2 directories, 3 files
subbu@Ubuntu:~/Linux-Practice$
```

41. hostname:

The hostname command in Linux is used to display or set the system's hostname.

1. Display the Current Hostname

To simply display the current hostname of your system, you can run:

```
subbu@Ubuntu:~/Linux-Practice$ hostname
Ubuntu
subbu@Ubuntu:~/Linux-Practice$
```

2. Display the IP Address(es)

```
subbu@Ubuntu:~/Linux-Practice$ hostname
Ubuntu
subbu@Ubuntu:~/Linux-Practice$ hostname -I
10.0.2.15
subbu@Ubuntu:~/Linux-Practice$
```

42. ifconfig:

The ifconfig command in Linux is used to configure network interfaces. It can be used to display the current network configuration, configure IP addresses, enable or disable interfaces, and more.

Ubuntu: sudo apt-get install net-tools

RHEL: sudo yum install net-tools

```
subbu@SUBBU:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 176 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 4s (49.4 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 206972 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
subbu@SUBBU:~$
```

1. **Display All Network Interfaces**:

```
subbu@Ubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::4dac:1f0:1cdc:db5c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:69:91:3f  txqueuelen 1000  (Ethernet)
        RX packets 115601  bytes 160749583 (160.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35180  bytes 3171069 (3.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: 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 1553  bytes 189194 (189.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1553  bytes 189194 (189.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subbu@Ubuntu:~$
```

2. **Display a Specific Interface**:

ifconfig eth0

Replace eth0 with the name of your network interface to display information for that specific interface.

Ex: **ifconfig enp0s3 :Here "enp0s3" is my system network interface**

```
subbu@Ubuntu:~/Linux-Practice$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::4dac:1f0:1cdc:db5c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:69:91:3f  txqueuelen 1000  (Ethernet)
        RX packets 291440  bytes 417904390 (417.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 70613  bytes 5319512 (5.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subbu@Ubuntu:~/Linux-Practice$
```

3. Enable an Interface:

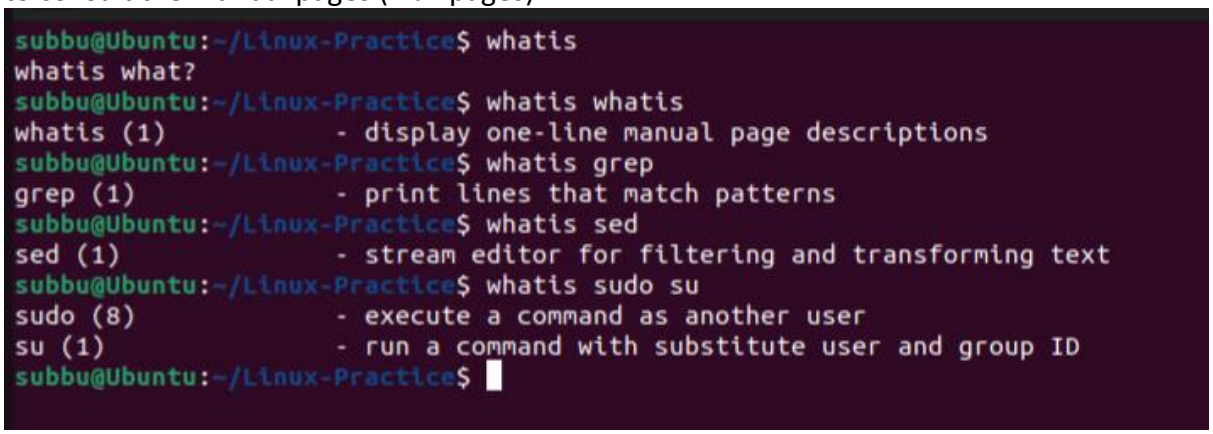
sudo ifconfig enp0s3 up : This command enables the network interface enp0s3.

sudo ifconfig enp0s3 down : This command disables the network interface enp0s3.

```
subbu@Ubuntu:~/Linux-Practice$ sudo ifconfig enp0s3 up
[sudo] password for subbu:
subbu@Ubuntu:~/Linux-Practice$ sudo ifconfig enp0s3 down
subbu@Ubuntu:~/Linux-Practice$ ifconfig
lo: 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 1714  bytes 201375 (201.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1714  bytes 201375 (201.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subbu@Ubuntu:~/Linux-Practice$ sudo ifconfig enp0s3 up
subbu@Ubuntu:~/Linux-Practice$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::4dac:1f0:1cdc:db5c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:69:91:3f  txqueuelen 1000  (Ethernet)
        RX packets 324701  bytes 466127362 (466.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 79269  bytes 5847908 (5.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: 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 2250  bytes 243231 (243.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2250  bytes 243231 (243.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subbu@Ubuntu:~/Linux-Practice$
```

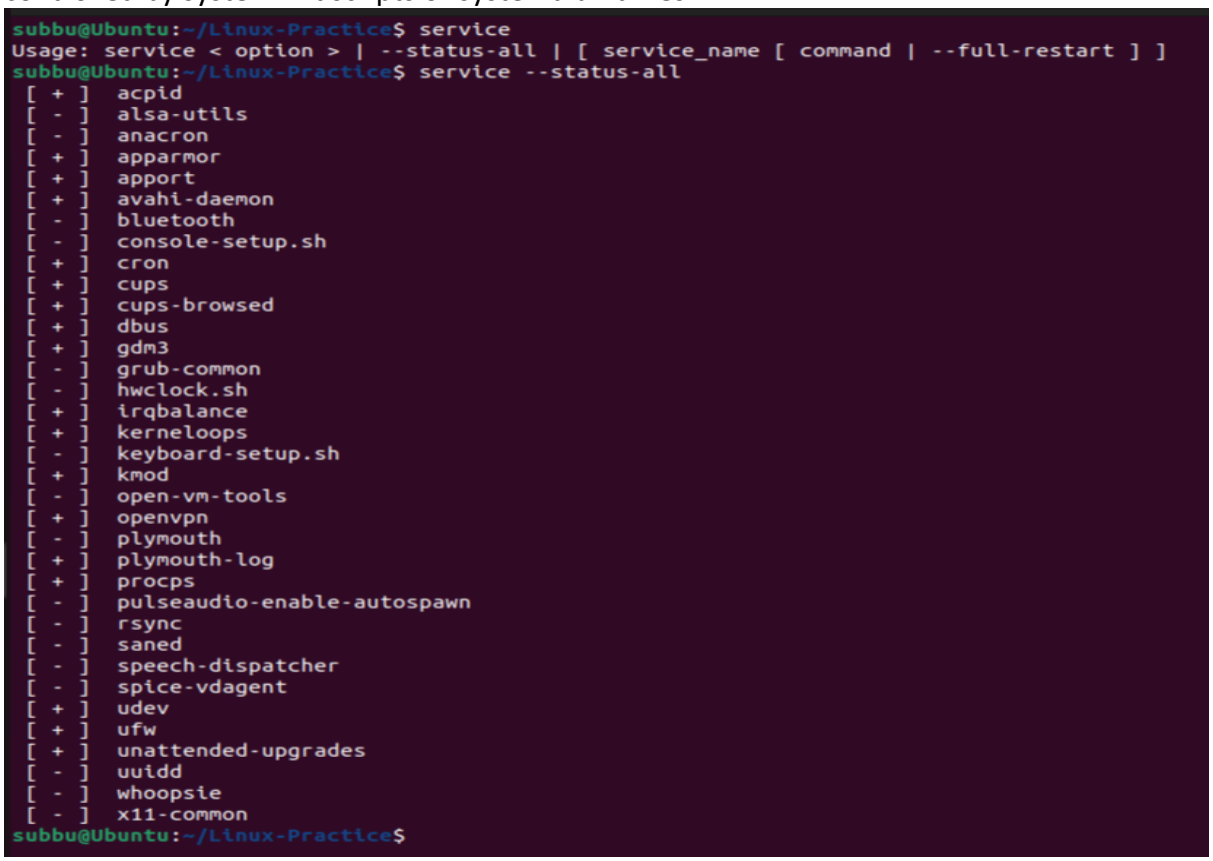
43. whatis:

The whatis command in Linux is used to display a short description of a command. It provides a brief summary of the purpose and functionality of the specified command. The whatis command is helpful for quickly finding information about a command without having to consult the manual pages (man pages).



44. service:

The service command in Linux is used to manage services (daemons) running on a Linux system. It allows you to start, stop, restart, enable, disable, and check the status of services controlled by system init scripts or systemd unit files.



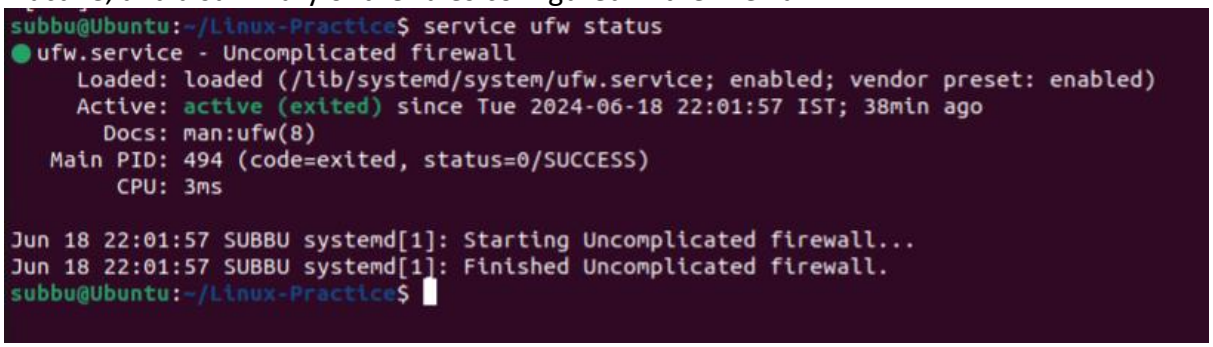
The basic syntax of the service command is:

service service_name action

Replace service_name with the name of the service you want to manage and action with the operation you want to perform (e.g., start, stop, restart, status, etc.).

Ex: **service ufw status**

This command will display the current status of the firewall, including whether it's active or inactive, and a summary of the rules configured in the firewall.



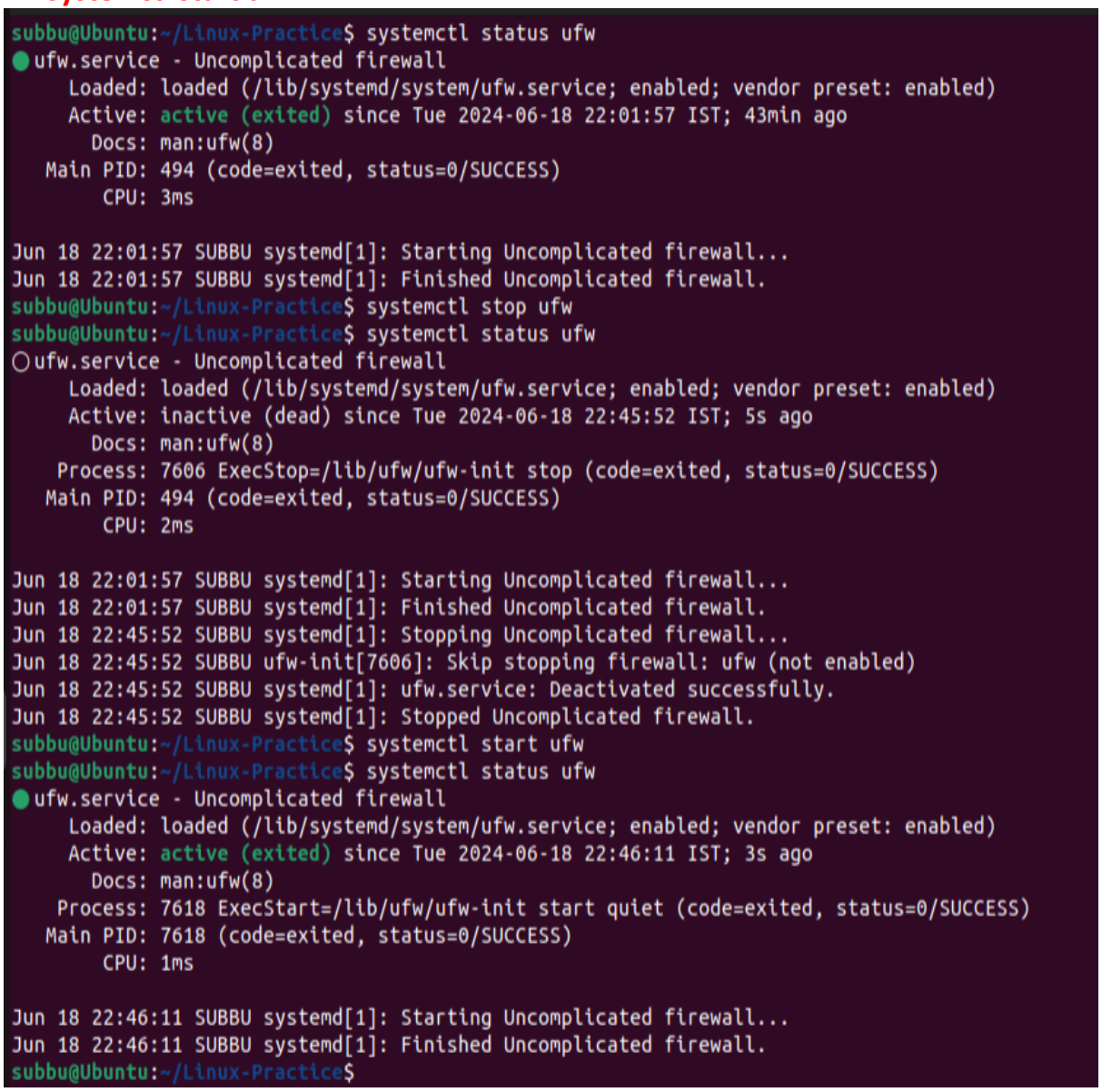
45. systemctl:

systemctl is a command-line utility in Linux used to manage systemd, the init system and service manager. It provides a centralized way of controlling the init system, services, and other units of the systemd system.

For systemd-based systems, you can use systemctl directly instead of service:

systemctl start service_name
systemctl stop service_name
systemctl restart service_name
systemctl status service_name
systemctl enable service_name
systemctl disable service_name

Ex: systemctl status ufw
systemctl stop ufw
systemctl start ufw



46. last:

The last command in Linux is used to display a list of users who have previously logged into the system, as well as the times at which they logged in and out. It reads from the /var/log/wtmp file, which contains a record of all login and logout activity on the system.

```
subbu@Ubuntu:~/Linux-Practice$ last
subbu    tty2         tty2             Tue Jun 18 22:02   still logged in
reboot   system boot  6.5.0-25-generic Tue Jun 18 22:01   still running
subbu    tty2         tty2             Mon Mar 18 07:44 - crash (92+14:17)
reboot   system boot  6.5.0-25-generic Mon Mar 18 07:44   still running
subbu    tty2         tty2             Mon Mar 18 07:38 - down   (00:06)
reboot   system boot  6.5.0-25-generic Mon Mar 18 07:38 - 07:44  (00:06)
subbu    tty2         tty2             Mon Mar 18 07:25 - down   (00:11)
reboot   system boot  6.5.0-25-generic Mon Mar 18 07:25 - 07:36  (00:11)
subbu    tty2         tty2             Mon Mar 18 07:10 - down   (00:14)
reboot   system boot  6.5.0-25-generic Mon Mar 18 07:10 - 07:25  (00:14)
subbu    tty2         tty2             Mon Mar 18 06:34 - down   (00:36)
reboot   system boot  6.5.0-25-generic Mon Mar 18 06:34 - 07:10  (00:36)

wtmp begins Mon Mar 18 06:34:02 2024
subbu@Ubuntu:~/Linux-Practice$
```

47. ps:

The ps command in Linux is used to display information about running processes. It provides a snapshot of the current processes, including details such as the process ID (PID), terminal associated with the process, CPU usage, memory usage, and more.

Basic Usage:

ps

```
subbu@Ubuntu:~/Linux-Practice$ ps
    PID TTY          TIME CMD
   6835 pts/0    00:00:00 bash
   7629 pts/0    00:00:00 ps
subbu@Ubuntu:~/Linux-Practice$
```

2. Full Format Listing: ps -f

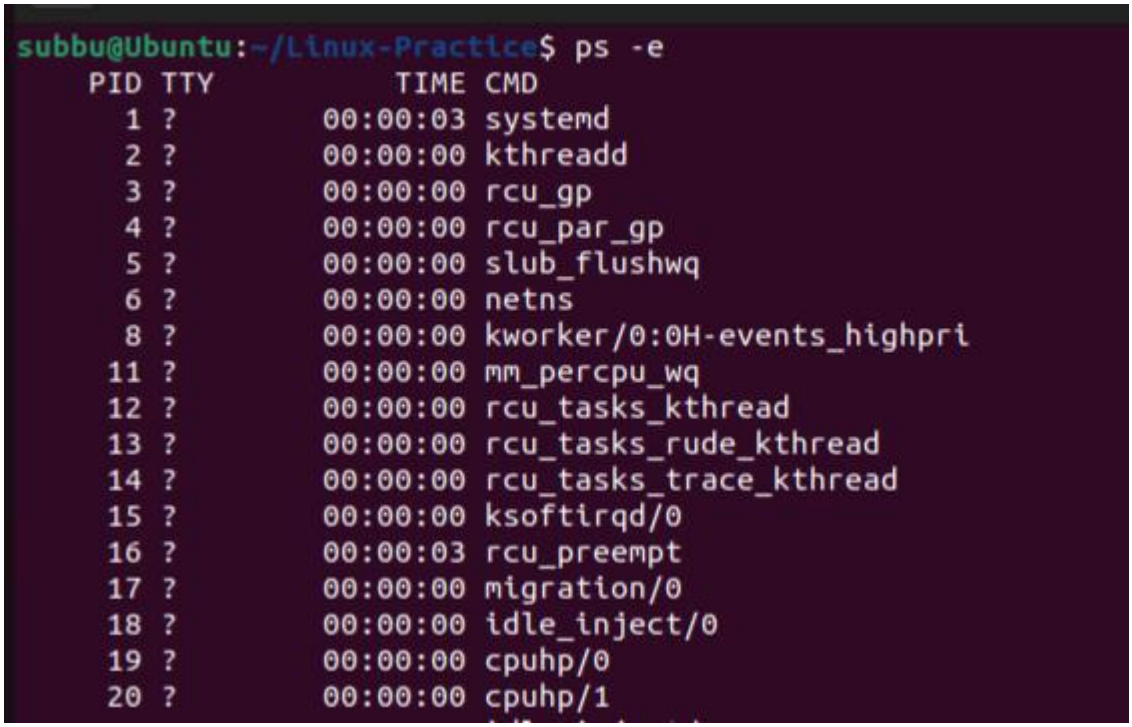
This command provides a full-format listing, including additional details like PPID (parent process ID).

```
subbu@Ubuntu:~/Linux-Practice$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
subbu       6835    6817  0 22:10 pts/0    00:00:00 bash
subbu       7790    6835  0 22:52 pts/0    00:00:00 ps -f
subbu@Ubuntu:~/Linux-Practice$
```

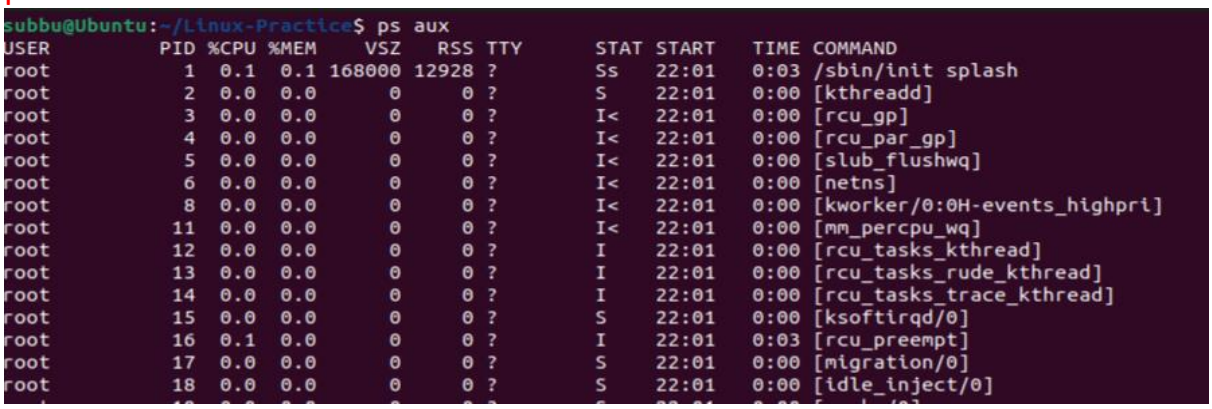
3.Viewing All Processes: ps -e (or) ps aux

 These commands display all processes running on the system.

ps -e:



ps aux:



4. Viewing Processes by User: ps -u username

Replace username with the actual username to see processes owned by that user.

ps -u root

```
subbu@Ubuntu:~$ ps -ef | grep root
root           1       0  0 14:21 ?        00:00:02 /sbin/init splash
root           2       0  0 14:21 ?        00:00:00 [kthreadd]
root           3       2  0 14:21 ?        00:00:00 [rcu_gp]
root           4       2  0 14:21 ?        00:00:00 [rcu_par_gp]
root           5       2  0 14:21 ?        00:00:00 [slub_flushwq]
root           6       2  0 14:21 ?        00:00:00 [netns]
root           7       2  0 14:21 ?        00:00:00 [kworker/0:0-cgroup_destroy]
root           8       2  0 14:21 ?        00:00:00 [kworker/0:0H-events_highpri]
root           9       2  0 14:21 ?        00:00:01 [kworker/0:1-mm_percpu_wq]
root          11       2  0 14:21 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 14:21 ?        00:00:00 [rcu_tasks_kthread]
root          13       2  0 14:21 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2  0 14:21 ?        00:00:00 [rcu_tasks_trace_kthread]
root          15       2  0 14:21 ?        00:00:00 [ksoftirqd/0]
root          16       2  0 14:21 ?        00:00:00 [rcu_preempt]
root          17       2  0 14:21 ?        00:00:00 [migration/0]
root          18       2  0 14:21 ?        00:00:00 [idle_inject/0]
root          19       2  0 14:21 ?        00:00:00 [cpuhp/0]
root          20       2  0 14:21 ?        00:00:00 [cpuhp/1]
root          21       2  0 14:21 ?        00:00:00 [idle_inject/1]
root          22       2  0 14:21 ?        00:00:00 [migration/1]
root          23       2  0 14:21 ?        00:00:00 [ksoftirqd/1]
root          24       2  0 14:21 ?        00:00:00 [kworker/1:0-cgroup_destroy]
root          25       2  0 14:21 ?        00:00:00 [kworker/1:0H-events_highpri]
root          26       2  0 14:21 ?        00:00:00 [cpuhp/2]
```

5. Viewing Processes by User (Alternative Method):

```
subbu@Ubuntu:~$ ps -ef | grep -e "subbu"
subbu       2587       1  0 19:29 ?        00:00:00 /lib/systemd/systemd --user
subbu       2588    2587  0 19:29 ?        00:00:00 (sd-pam)
subbu       2594    2587  0 19:29 ?        00:00:00 /usr/bin/pipewire
subbu       2595    2587  0 19:29 ?        00:00:00 /usr/bin/pipewire-media-session
subbu       2596    2587  0 19:29 ?        00:00:00 /usr/bin/pulseaudio --daemonize=no --log-target=journal
subbu       2615       1  0 19:29 ?        00:00:00 /usr/bin/gnome-keyring-daemon --daemonize --login
subbu       2618    2587  0 19:29 ?        00:00:00 /usr/bin/dbus-daemon --session --address=systemd: --nofork --n
subbu       2620    2576  0 19:29 tty2     00:00:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=
subbu       2623    2587  0 19:29 ?        00:00:00 /usr/libexec/xdg-document-portal
subbu       2624    2620  0 19:29 tty2     00:00:00 /usr/libexec/gnome-session-binary --session=ubuntu
subbu       2631    2587  0 19:29 ?        00:00:00 /usr/libexec/xdg-permission-store
subbu       2677    2587  0 19:29 ?        00:00:00 /usr/libexec/gnome-session-ctl --monitor
subbu       2687    2587  0 19:29 ?        00:00:00 /usr/libexec/gvfsd
subbu       2694    2587  0 19:29 ?        00:00:00 /usr/libexec/gvfsd-fuse /run/user/1000/gvfs -f
subbu       2696    2587  0 19:29 ?        00:00:00 /usr/libexec/gnome-session-binary --systemd-service --session=
subbu       2718    2696  0 19:29 ?        00:00:00 /usr/libexec/at-spi-bus-launcher --launch-immediately
subbu       2719    2587  7 19:29 ?        00:00:23 /usr/bin/gnome-shell
```

6. Viewing Processes by Multiple Users:

```
subbu@Ubuntu:~/Linux-Practice$ ps -ef |grep -iE "subbu|root"
root           1       0  0 19:27 ?        00:00:04 /sbin/init splash
root           2       0  0 19:27 ?        00:00:00 [kthreadd]
root           3       2  0 19:27 ?        00:00:00 [rcu_gp]
root           4       2  0 19:27 ?        00:00:00 [rcu_par_gp]
root           5       2  0 19:27 ?        00:00:00 [slub_flushwq]
root           6       2  0 19:27 ?        00:00:00 [netns]
root           8       2  0 19:27 ?        00:00:00 [kworker/0:0H-events_highpri]
root          11       2  0 19:27 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 19:27 ?        00:00:00 [rcu_tasks_kthread]
root          13       2  0 19:27 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2  0 19:27 ?        00:00:00 [rcu_tasks_trace_kthread]
```

5.Detailed Information: ps -ef

This command provides detailed information about all processes.

```
subbu@Ubuntu:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 19:27 ?        00:00:02 /sbin/init splash
root           2       0  0 19:27 ?        00:00:00 [kthreadd]
root           3       2  0 19:27 ?        00:00:00 [rcu_gp]
root           4       2  0 19:27 ?        00:00:00 [rcu_par_gp]
root           5       2  0 19:27 ?        00:00:00 [slub_flushwq]
root           6       2  0 19:27 ?        00:00:00 [netns]
root           7       2  0 19:27 ?        00:00:00 [kworker/0:0-rcu_gp]
root           8       2  0 19:27 ?        00:00:00 [kworker/0:0H-events_highpri]
root           9       2  0 19:27 ?        00:00:00 [kworker/0:1-rcu_gp]
root          10       2  0 19:27 ?        00:00:00 [kworker/u10:0-events_unbound]
root          11       2  0 19:27 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 19:27 ?        00:00:00 [rcu_tasks_kthread]
root          13       2  0 19:27 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2  0 19:27 ?        00:00:00 [rcu_tasks_trace_kthread]
root          15       2  0 19:27 ?        00:00:00 [ksoftirqd/0]
```

Common Options

- -A or -e: Select all processes.
- -a: Select all processes except session leaders and processes not associated with a terminal.
- -u: Display user-oriented format.
- -x: Also show processes not attached to a terminal.
- -f: Display full format listing.
- -o: Specify user-defined format.

48. kill:

The kill command in Linux is used to terminate or send signals to processes. It is often used to gracefully stop or forcefully terminate processes running on the system.

The basic syntax of the kill command is: **kill [options] PID**

- PID: Process ID of the target process.
- options: Various options to specify the signal to send to the process.

Here are some common options you can use with the kill command:

- **-l, --list**: List all available signal names.
- **-s, --signal SIGNAL**: Specify the signal to send. You can use either signal names or their corresponding numbers.
- **-9, --signal SIGKILL**: Send a SIGKILL signal to terminate the process immediately (forceful termination).
- **-15, --signal SIGTERM**: Send a SIGTERM signal to terminate the process gracefully (soft termination).

Examples:

1. **Send SIGTERM Signal (Soft Termination)**: **kill PID Ex: kill 34585**

This command sends a SIGTERM signal to gracefully terminate the process.

```
  34585 ?        00:00:00 ssh-agent
  34728 ?        00:00:00 gjs
  34789 pts/0    00:00:00 ps
subbu@Ubuntu:~/Linux-Practice$ kill 34585
subbu@Ubuntu:~/Linux-Practice$ ps -u subbu
    PID TTY          TIME CMD
   2587 ?        00:00:01 systemd
   2588 ?        00:00:00 (sd-pam)
   2594 ?        00:00:00 pipewire
   2595 ?        00:00:00 pipewire-media-
   2596 ?        00:00:00 pulseaudio
   3385 ?        00:00:00 gnome-terminal.
   3390 ?        00:00:12 gnome-terminal-
   3408 pts/0    00:00:00 bash
  34728 ?        00:00:00 gjs
  34790 pts/0    00:00:00 ps
subbu@Ubuntu:~/Linux-Practice$
```

2. **Send SIGKILL Signal (Forceful Termination)**: **kill -9 PID Ex: kill -9 34728**

This command sends a SIGKILL signal to forcefully terminate the process.

```
subbu@Ubuntu:~/Linux-Practice$ kill -9 34728
subbu@Ubuntu:~/Linux-Practice$ ps -ef |grep -ie "subbu" -e "root"
root           1       0  0 19:27 ?        00:00:04 /sbin/init splash
root           2       0  0 19:27 ?        00:00:00 [kthreadd]
root           3       2  0 19:27 ?        00:00:00 [rcu_gp]
root           4       2  0 19:27 ?        00:00:00 [rcu_par_gp]
root           5       2  0 19:27 ?        00:00:00 [slub_flushwq]
root       34702       2  0 21:17 ?        00:00:00 [kworker/u10:2-events_unbound]
root       34780       2  0 21:19 ?        00:00:00 [kworker/0:1]
root       34786       2  0 21:22 ?        00:00:00 [kworker/u10:0-events_unbound]
root       34788       2  0 21:24 ?        00:00:00 [kworker/0:2-events]
root       34800       2  0 21:30 ?        00:00:00 [kworker/u10:3-events_unbound]
root       34807       1  0 21:32 ?        00:00:00 /usr/sbin/anacron -d -q -s
subbu      34817    2719  3 21:33 ?        00:00:00 gjs /usr/share/gnome-shell/extensions/ding@raster
:0:70:0:0
subbu      34829    2587  2 21:33 ?        00:00:00 /usr/bin/nautilus --gapplication-service
subbu      34838    2587  1 21:33 ?        00:00:00 /usr/bin/file-roller --service
subbu      34853    3408  0 21:33 pts/0    00:00:00 ps -ef
subbu      34854    3408  0 21:33 pts/0    00:00:00 grep --color=auto -ie subbu -e root
subbu@Ubuntu:~/Linux-Practice$
```

3. List Available Signal Names: kill -l

This command lists all available signal names and their corresponding numbers.

```
subbu@Ubuntu:~/Linux-Practice$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       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) SIGIO       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
subbu@Ubuntu:~/Linux-Practice$
```

49.top:

The top command in Linux is a real-time process monitoring tool that displays information about running processes and system resource usage. It provides an interactive interface that continuously updates to show the most relevant system statistics.

Ex: top

```
top - 21:39:49 up  2:12,  1 user,  load average: 0.16, 0.13, 0.13
Tasks: 215 total,   1 running, 214 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.4 us,  0.6 sy,  0.0 ni, 97.8 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :  10189.5 total,   6081.3 free,    960.7 used,   3147.4 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   8902.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2719 subbu     20   0 5447604 430684 147652 S  21.0   4.1   3:59.16 gnome-shell
   3390 subbu     20   0  556484  50448  38288 S   3.0   0.5   0:17.15 gnome-terminal-
    535 systemd+  20   0   14836   6784   6016 S   0.3   0.1   0:15.81 systemd-oomd
   3028 subbu     20   0  207376  64288  49864 S   0.3   0.6   0:07.55 Xwayland
   3170 subbu     20   0  227520   3228   2816 S   0.3   0.0   0:38.59 VBoxClient
  34675 root      20   0       0      0      0 I   0.3   0.0   0:00.48 kworker/3:0-events
  34702 root      20   0       0      0      0 I   0.3   0.0   0:00.59 kworker/u10:2-flush-8:0
  34800 root      20   0       0      0      0 I   0.3   0.0   0:00.25 kworker/u10:3-events_freezable_power_
      1 root      20   0  167996  13048   8312 S   0.0   0.1   0:04.90 systemd
```

This will display a dynamic view of the running processes and system resource usage. By default, processes are sorted by CPU usage, with the most CPU-intensive processes listed at the top.

Load Average Concept:

```
top - 21:43:48 up  2:16,  1 user,  load average: 0.15, 0.16, 0.13
Tasks: 215 total,   1 running, 214 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  0.6 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  10189.5 total,   6080.9 free,    961.1 used,   3147.5 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   8902.2 avail Mem
```

In the top command output, the "load average" represents the average system load over the last 1, 5, and 15 minutes. It provides an indication of how busy the system has been over different time intervals.

Understanding Load Average Values

- **Low Values**: A load average below 1.0 generally indicates that the system is not heavily loaded and there are sufficient resources available to handle additional workload.
- **High Values**: A load average greater than 1.0 indicates that the system is under load, and the higher the value, the more loaded the system is. If the load average consistently exceeds the number of CPU cores, it may indicate that the system is overloaded

Factors Affecting Load Average

Several factors can contribute to high load averages, including:

- CPU-bound processes consuming a significant amount of CPU time.
- I/O-bound processes performing heavy disk I/O operations.
- Network-bound processes handling a large volume of network traffic

50. zip:

The zip command in Linux is used to compress files and directories into a single compressed archive file with the .zip extension.

1.Recursively Compress Directory: **zip -r archive.zip directory**

Ex: zip -r bash.zip BASH/

```
subbu@Ubuntu:~/Linux-Practice$ tree
.
├── app.py
├── BASH
│   └── Scripts
│       └── chkerrors.sh
├── dev-app.java
└── testfile.txt

2 directories, 4 files
subbu@Ubuntu:~/Linux-Practice$ zip -r bash.zip BASH/
  adding: BASH/ (stored 0%)
  adding: BASH/Scripts/ (stored 0%)
  adding: BASH/Scripts/chkerrors.sh (deflated 15%)
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 12K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu  537 Jun 19 21:50 bash.zip
subbu@Ubuntu:~/Linux-Practice$
```

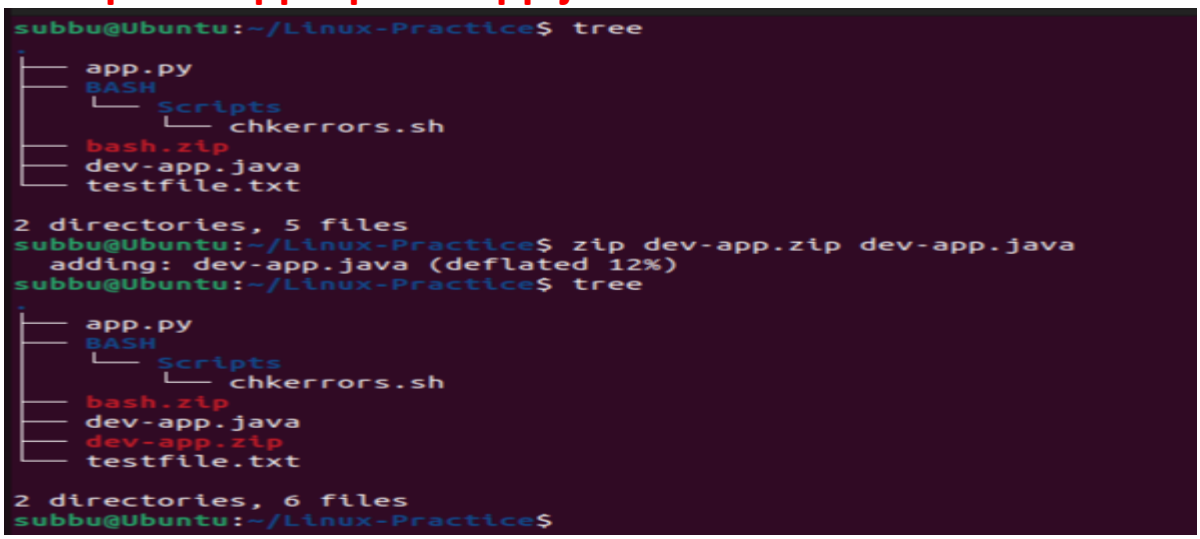
Common Options:

Here are some common options you can use with the zip command:

- **-r**: Recursively include all files and subdirectories within directories.
- **-q**: Quiet mode. Suppress output messages during compression.
- **-9**: Use maximum compression level (slower but produces smaller archives).
- **-u**: Update existing archive files with newer files. Add new files or update existing files within the archive.
- **-m**: Move files into the archive. Delete original files after compression.

2.Compress a Single File: zip archive.zip file.txt

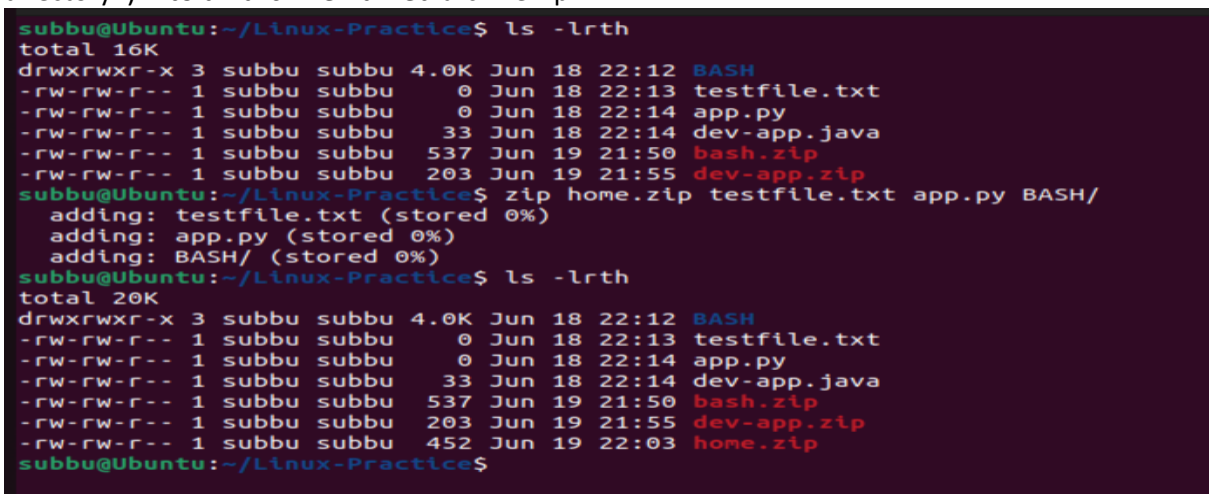
Ex: zip dev-app.zip dev-app.java



3. Compress Multiple Files and Directories:

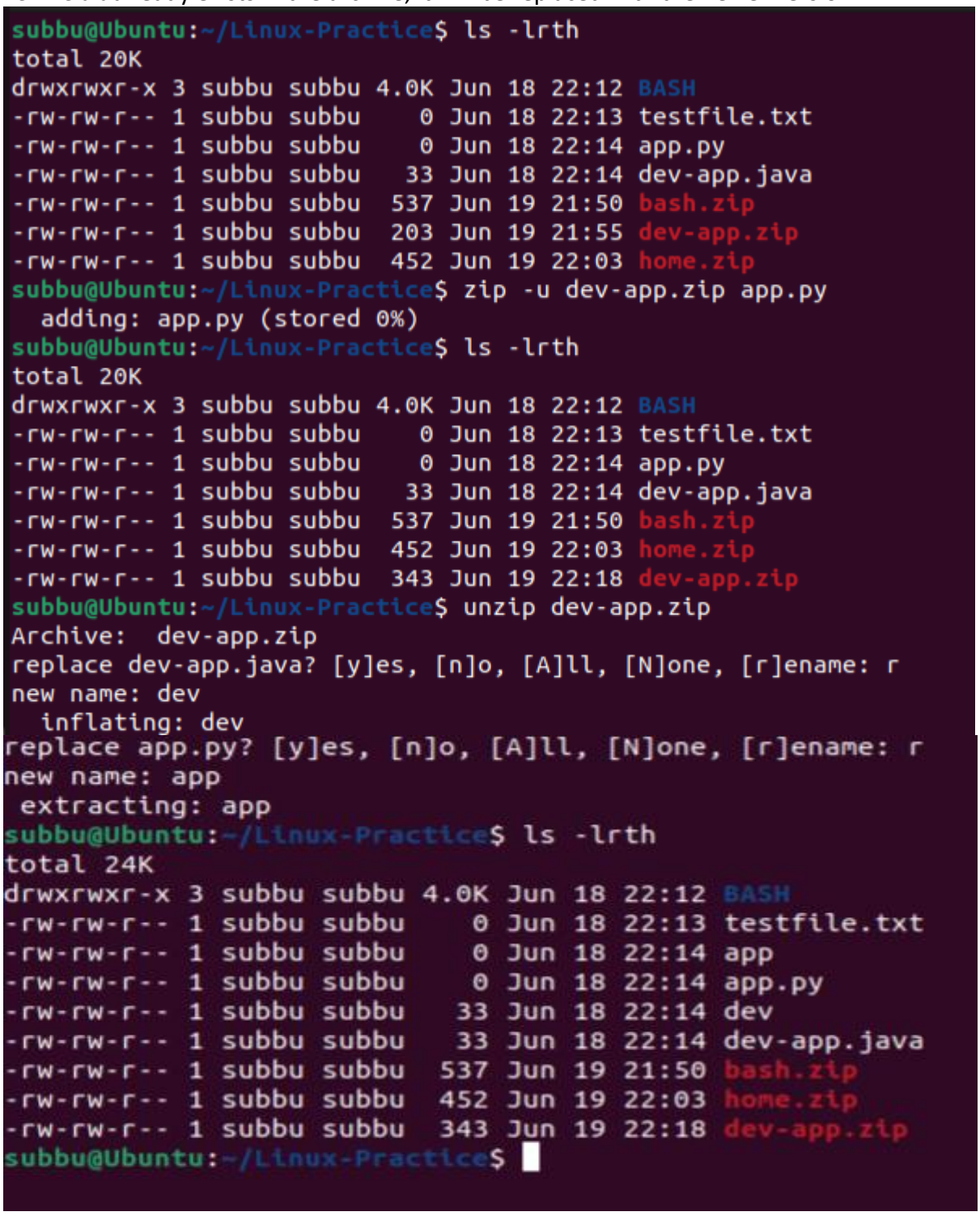
zip archive.zip file1.txt file2.txt directory1 directory2

This command compresses multiple files (file1.txt, file2.txt) and directories (directory1, directory2) into an archive named archive.zip.



4. Update Existing Archive: zip -u archive.zip newfile.txt

This command updates an existing archive archive.zip by adding a new file newfile.txt to it. If newfile.txt already exists in the archive, it will be replaced with the newer version.



51.unzip:

The unzip command in Linux is used to extract files from ZIP archives. It's the counterpart to the zip command, allowing you to decompress and extract files and directories from compressed ZIP archives. **Syntax: unzip [options] zipfile.zip**

```
subbu@Ubuntu:~/Linux-Practice$ whatis unzip
unzip (1)            - list, test and extract compressed files in a ZIP archive
```

Ex: unzip dev-app.zip

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu  537 Jun 19 21:50 bash.zip
-rw-rw-r-- 1 subbu subbu  452 Jun 19 22:03 home.zip
-rw-rw-r-- 1 subbu subbu  343 Jun 19 22:18 dev-app.zip
subbu@Ubuntu:~/Linux-Practice$ unzip dev-app.zip
Archive:  dev-app.zip
replace dev-app.java? [y]es, [n]o, [A]ll, [N]one, [r]ename: r
new name: dev
replace dev? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: dev
replace app.py? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
 extracting: app.py
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu  537 Jun 19 21:50 bash.zip
-rw-rw-r-- 1 subbu subbu  452 Jun 19 22:03 home.zip
-rw-rw-r-- 1 subbu subbu  343 Jun 19 22:18 dev-app.zip
subbu@Ubuntu:~/Linux-Practice$
```

Here are some common options you can use with the unzip command:

- **-d directory**: Extract files into the specified directory instead of the current directory.
- **-q**: Quiet mode. Suppress output messages during extraction.
- **-o**: Overwrite existing files without prompting.
- **-l**: List contents of the ZIP archive without extracting.

1. **Extract ZIP Archive to a Specific Directory**:

unzip archive.zip -d /path/to/directory

This command extracts files and directories from the archive archive.zip into the specified directory /path/to/directory.

Ex: unzip dev-app.zip -d BASH/Scripts/

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu  537 Jun 19 21:50 bash.zip
-rw-rw-r-- 1 subbu subbu  452 Jun 19 22:03 home.zip
-rw-rw-r-- 1 subbu subbu  343 Jun 19 22:18 dev-app.zip
subbu@Ubuntu:~/Linux-Practice$ unzip dev-app.zip -d BASH/Scripts/
Archive:  dev-app.zip
  inflating: BASH/Scripts/dev-app.java
 extracting: BASH/Scripts/app.py
subbu@Ubuntu:~/Linux-Practice$ tree
.
├── app
├── app.py
├── BASH
│   └── Scripts
│       ├── app.py
│       ├── chkerrors.sh
│       └── dev-app.java
├── bash.zip
├── dev
├── dev-app.java
├── dev-app.zip
├── home.zip
└── testfile.txt

2 directories, 11 files
subbu@Ubuntu:~/Linux-Practice$
```

2. List Contents of ZIP Archive: unzip -l archive.zip

This command lists the contents (files and directories) of the archive archive.zip without extracting them.

Ex: **unzip -l bash.zip**

```
subbu@Ubuntu:~/Linux-Practice$ unzip -l bash.zip
Archive:  bash.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2024-06-18 22:12   BASH/
        0  2024-06-19 21:49   BASH/Scripts/
       53  2024-06-19 21:49   BASH/Scripts/chkerrors.sh
---------                     -------
       53                     3 files
subbu@Ubuntu:~/Linux-Practice$
```

3. Extract Specific File from ZIP Archive: unzip archive.zip path/to/file.txt

This command extracts only the file file.txt from the archive archive.zip into the current directory.

Ex: unzip bash.zip BASH/Scripts/chkerros.sh

```
subbu@Ubuntu:~/Linux-Practice$ tree
.
├── app
├── app.py
├── BASH
│   └── Scripts
│       ├── app.py
│       ├── chkerrors.sh
│       └── dev-app.java
├── bash.zip
├── dev
├── dev-app.java
├── dev-app.zip
├── home.zip
└── testfile.txt

2 directories, 11 files
subbu@Ubuntu:~/Linux-Practice$ unzip bash.zip BASH/Scripts/chkerrors.sh
Archive:  bash.zip
replace BASH/Scripts/chkerrors.sh? [y]es, [n]o, [A]ll, [N]one, [r]ename: r
new name: check.sh
  inflating: check.sh
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  537 Jun 19 21:50 bash.zip
-rw-rw-r-- 1 subbu subbu  452 Jun 19 22:03 home.zip
-rw-rw-r-- 1 subbu subbu  343 Jun 19 22:18 dev-app.zip
subbu@Ubuntu:~/Linux-Practice$
```

52.tar :

The tar command in Linux is used to create, extract, and manage archive files. The name "tar" stands for "tape archive," and it is commonly used to collect many files into a single archive file, often for distribution or backup purposes. The tar format can also be compressed to reduce the archive size.

Basic Syntax

The basic syntax of the tar command is:

tar [OPTIONS] [ARCHIVE_NAME] [FILE/DIRECTORY]

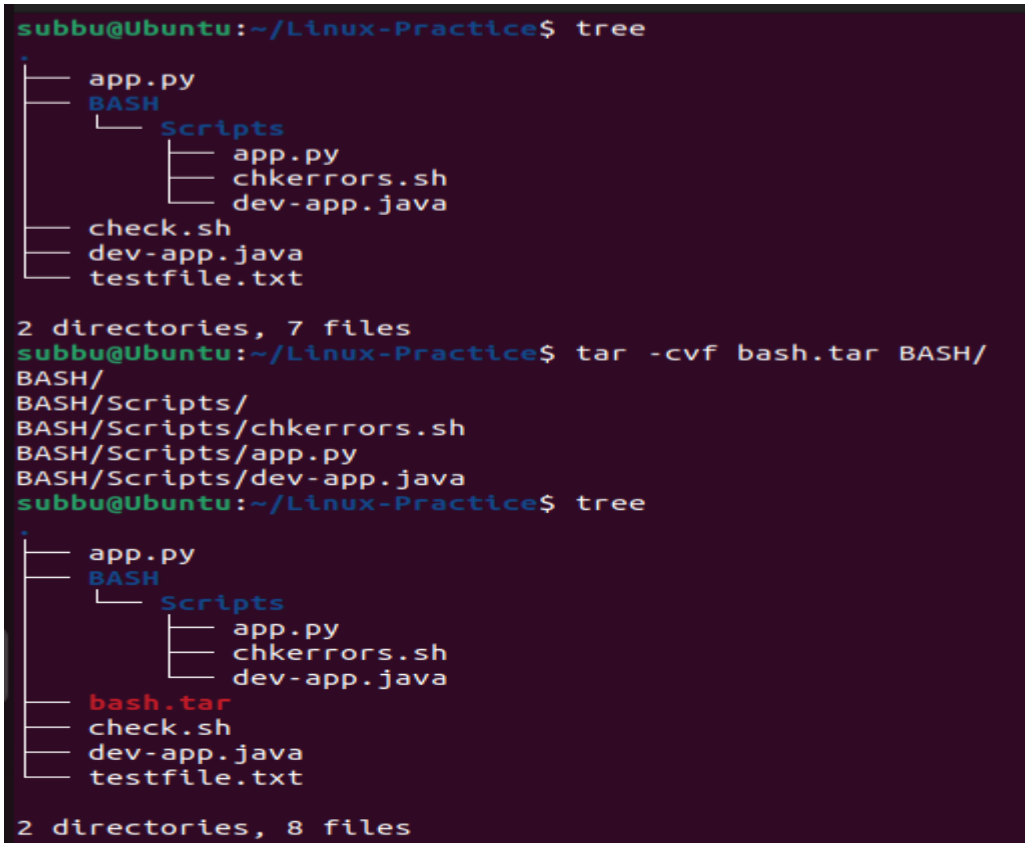
Here are some commonly used options with the tar command:

- **-c**: Create a new archive.
- **-x**: Extract files from an archive.
- **-t**: List the contents of an archive.
- **-v**: Verbosely list files processed.
- **-f**: Specify the name of the archive file.
- **-z**: Compress the archive using gzip.
- **-j**: Compress the archive using bzip2.
- **-J**: Compress the archive using xz.

1. Creating an Archive: tar -cvf archive_name.tar example_directory/

To create a tar archive of a directory named example_directory:

Ex: tar -cvf bash.tar BASH/



2. Creating a Compressed Archive:

To create a gzip-compressed tar archive:

tar -czvf archive_name.tar.gz example_directory/

Ex: tar -czvf test.tar testfile.txt

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 24K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  10K Jun 19 22:45 bash.tar
subbu@Ubuntu:~/Linux-Practice$ tar -czvf test.tar testfile.txt
testfile.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  10K Jun 19 22:45 bash.tar
-rw-rw-r-- 1 subbu subbu  118 Jun 19 22:52 test.tar
subbu@Ubuntu:~/Linux-Practice$
```

3. Extracting an Archive: tar -xvf archive_name.tar

To extract a tar archive:

Ex: tar -xvf test.tar

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  10K Jun 19 22:45 bash.tar
-rw-rw-r-- 1 subbu subbu  118 Jun 19 22:52 test.tar
subbu@Ubuntu:~/Linux-Practice$ tar -xvf test.tar
testfile.txt
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  10K Jun 19 22:45 bash.tar
-rw-rw-r-- 1 subbu subbu  118 Jun 19 22:52 test.tar
subbu@Ubuntu:~/Linux-Practice$
```

4. Listing the Contents of an Archive: tar -tvf archive_name.tar

Ex: tar -tvf bash.tar

```
subbu@Ubuntu:~/Linux-Practice$ ls -lrth
total 28K
drwxrwxr-x 3 subbu subbu 4.0K Jun 18 22:12 BASH
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:13 testfile.txt
-rw-rw-r-- 1 subbu subbu    0 Jun 18 22:14 app.py
-rw-rw-r-- 1 subbu subbu   33 Jun 18 22:14 dev-app.java
-rw-rw-r-- 1 subbu subbu   53 Jun 19 21:49 check.sh
-rw-rw-r-- 1 subbu subbu  10K Jun 19 22:45 bash.tar
-rw-rw-r-- 1 subbu subbu  118 Jun 19 22:52 test.tar
subbu@Ubuntu:~/Linux-Practice$ tar -tvf bash.tar
drwxrwxr-x subbu/subbu       0 2024-06-18 22:12 BASH/
drwxrwxr-x subbu/subbu       0 2024-06-19 22:37 BASH/Scripts/
-rw-rw-r-- subbu/subbu      53 2024-06-19 21:49 BASH/Scripts/chkerrors.sh
-rw-rw-r-- subbu/subbu       0 2024-06-18 22:14 BASH/Scripts/app.py
-rw-rw-r-- subbu/subbu      33 2024-06-18 22:14 BASH/Scripts/dev-app.java
subbu@Ubuntu:~/Linux-Practice$
```

53. useradd:

The useradd command in Linux is used to create a new user account on the system. It's a fundamental command for system administration, allowing administrators to manage user accounts and permissions.

The basic syntax of the useradd command is:

sudo useradd [options] username

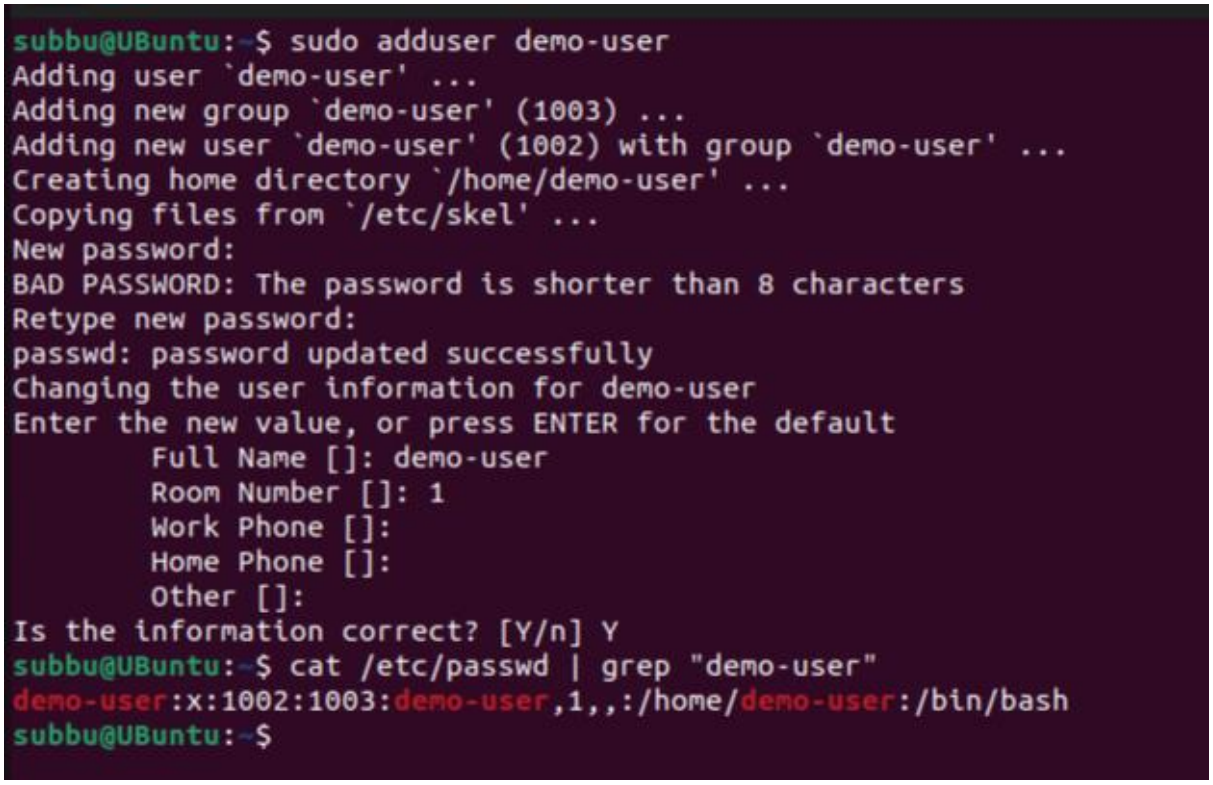
username: The username of the new user account to be created.

Here are some common options you can use with the useradd command:

- **-m, --create-home**: Create the user's home directory if it doesn't already exist.
- **-s, --shell SHELL**: Specify the login shell for the user. The default shell is typically /bin/bash.
- **-g, --gid GROUP**: Specify the primary group for the user.
- **-G, --groups GROUPS**: Specify additional groups for the user.
- **-c, --comment COMMENT**: Add a descriptive comment for the user.
- **-u, --uid UID**: Specify the user ID for the new user.
- **-d, --home HOME_DIR**: Specify the home directory for the user.

1. Create a New User:

```
subbu@Ubuntu:~/Linux-Practice$ sudo useradd Devops
[sudo] password for subbu:
subbu@Ubuntu:~/Linux-Practice$ cat /etc/passwd | grep "Devops"
Devops:x:1001:1001::/home/Devops:/bin/sh
subbu@Ubuntu:~/Linux-Practice$
```



1.1 List All Users:

cat /etc/passwd

less /etc/passwd

grep /etc/passwd

1.2 List Logged-In Users:

Who

w

1.3 List Users with Human-Readable Output:

awk -F':' '{ print $1 }' /etc/passwd

This command prints only the usernames from the /etc/passwd file.

1.4 List Users in Specific Groups:

If you want to list users belonging to specific groups, you can use the getent command along with grep:

getent group groupname

54. passwd:

The passwd command in Linux is used to change the password of a user account. It allows users to update their own passwords and system administrators to change passwords for other users.

The basic syntax of the passwd command is:

passwd [options] [username]

username: (Optional) The username of the user whose password you want to change. If omitted, passwd changes the password for the current user.

Common Options

Here are some common options you can use with the passwd command:

- **-l, --lock**: Lock the password of the specified account, preventing the user from logging in with a password.
- **-u, --unlock**: Unlock the password of the specified account, allowing the user to log in with a password again.
- **-d, --delete**: Delete the password of the specified account, effectively disabling password-based login for the user.

1. Change Your Own Password:
To change your own password, simply run the passwd command without any options:
passwd
You will be prompted to enter your current password and then to enter and confirm your new password.
2. Change Another User's Password:
To change the password of another user (requires root or superuser privileges), specify the username as an argument:
sudo passwd username
Replace username with the username of the user whose password you want to change. You will be prompted to enter and confirm the new password.
3. Lock or Unlock an Account:
To lock or unlock an account (requires root or superuser privileges), use the -l (lock) or -u (unlock) options:
sudo passwd -l username # Lock the account
sudo passwd -u username # Unlock the account
4. Delete a Password:
To delete the password of an account (requires root or superuser privileges), use the -d (delete) option:
sudo passwd -d username

55. groupadd:

Creating a new group in Linux can be done using the groupadd command. This command is used to create new groups in the system, and it requires superuser (root) privileges.

The basic syntax of the groupadd command is:
sudo groupadd [options] group_name

1. Creating a Simple Group:

To create a new group named developers, you would use:

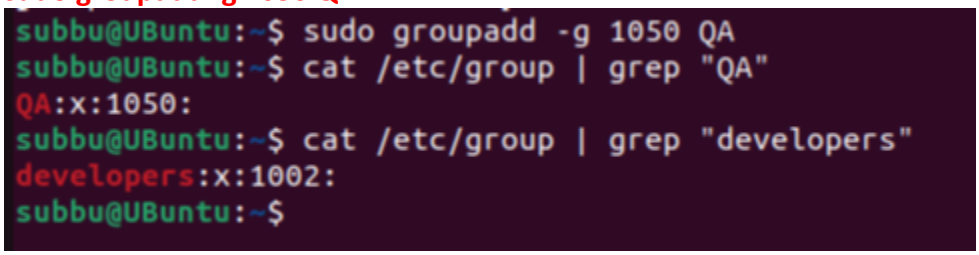
sudo groupadd developers

Group information available in /etc/group

2. Creating a Group with a Specific GID:

If you want to create a group with a specific Group ID (GID), you can use the -g option:

sudo groupadd -g 1050 QA



56. usermod:

The usermod command in Linux is used to modify the properties of an existing user account. It can change a user's home directory, login name, group memberships, shell, and other attributes.

The basic syntax of the usermod command is:

sudo usermod [options] username

Here are some common options you can use with the usermod command:

- **-a, --append**: Add the user to the supplementary group(s). Use with the -G option.
- **-c, --comment**: Change the user's comment (also known as GECOS).
- **-d, --home**: Change the user's home directory.
- **-e, --expiredate**: Set the account expiration date.
- **-g, --gid**: Change the user's primary group.
- **-G, --groups**: Change the user's supplementary groups.
- **-l, --login**: Change the user's login name.
- **-L, --lock**: Lock the user's account.
- **-p, --password**: Change the user's encrypted password.
- **-s, --shell**: Change the user's login shell.
- **-u, --uid**: Change the user's user ID.
- **-U, --unlock**: Unlock the user's account.

Steps to Add a User to Multiple Groups:

Use usermod with the -aG Option: To add a user to multiple groups, you use the usermod command followed by the -aG option and the list of groups.

sudo usermod -aG group1, group2, group3 username

Ex: sudo usermod -aG developers, QA Subbu

```
subbu@UBuntu:~$ cat /etc/passwd  | awk -F ':' '{print $1}'
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
tss
uuidd
systemd-oom
tcpdump
avahi-autoipd
usbmux
dnsmasq
kernoops
avahi
cups-pk-helper
rtkit
whoopsie
sssd
speech-dispatcher
fwupd-refresh
nm-openvpn
saned
colord
geoclue
pulse
gnome-initial-setup
hplip
gdm
subbu
vboxadd
Devops
subbu@UBuntu:~$ sudo usermod -aG developers,QA subbu
```

```
subbu@UBuntu:~$ getent group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,subbu
subbu:x:1000:
sambashare:x:136:subbu
vboxsf:x:999:
vboxdrmipc:x:998:
Devops:x:1001:
developers:x:1002:subbu
QA:x:1050:subbu
subbu@UBuntu:~$
```

59. lid or id :

Id: Print real and effective user and group IDs

```
subbu@UBuntu:~$ id
uid=1000(subbu) gid=1000(subbu) groups=1000(subbu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),135(lxd),136(sambashare)
subbu@UBuntu:~$ whatis id
id (1)               - print real and effective user and group IDs
subbu@UBuntu:~$ whatis lid
lid (1)              - Query ID database and report results
```

60. userdel:

The userdel command in Linux is used to delete a user account from the system. This command removes the user's entry from the system files (/etc/passwd, /etc/shadow, /etc/group, and others), effectively deleting the user from the system.

The basic syntax of the userdel command is:

sudo userdel [options] username

Here are some common options you can use with the userdel command:

- **-r, --remove**: Remove the user's home directory and mail spool.
- **-f, --force**: Force removal of the user account, even if the user is currently logged in.
- **-r, --root CHROOT_DIR**: Apply changes in the CHROOT_DIR directory and use the configuration files from the CHROOT_DIR directory.

1.Delete a User Without Removing Home Directory:

sudo userdel username

Ex: sudo userdel demo-user

```
subbu@UBuntu:~$ sudo userdel demo-user
subbu@UBuntu:~$ cat /etc/passwd | grep "demo-user"
subbu@UBuntu:~$ cat /etc/passwd | awk -F ':' '{print $1}'
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
tss
uuidd
systemd-oom
tcpdump
avahi-autoipd
usbmux
dnsmasq
kernoops
avahi
cups-pk-helper
rtkit
whoopsie
sssd
speech-dispatcher
fwupd-refresh
nm-openvpn
saned
colord
geoclue
pulse
gnome-initial-setup
hplip
gdm
subbu
vboxadd
Devops
subbu@UBuntu:~$
```

2. Delete a User and Remove Home Directory:

sudo userdel -r username

This command deletes the user account and also removes the user's home directory and mail spool.

3. Forcefully Delete a User Account:

sudo userdel -f username

This command forcefully deletes the user account, even if the user is currently logged in.

61.groupdel:

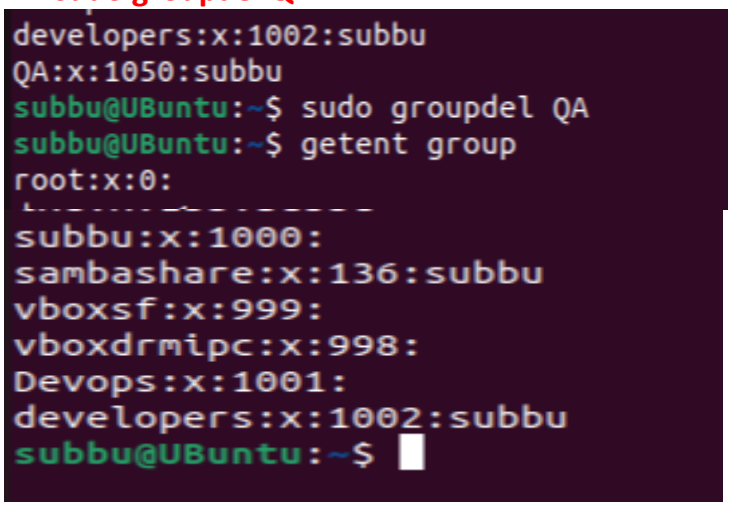
The groupdel command in Linux is used to delete a group from the system. Only the root user or a user with sudo privileges can execute this command.

The basic syntax of the groupdel command is:

sudo groupdel groupname

groupname: The name of the group to delete.

Ex: sudo groupdel QA



62. sudo visudo:

/etc/sudoers

Granting administrative (admin) access to a user in Linux typically involves adding the user to the sudo group. Users in the sudo group have permission to run commands as the superuser (root) or another user, as specified by the sudoers file.

Steps to Add a User to the sudo Group:

Add the User to the sudo Group: Use the usermod command to add the user to the sudo group. Replace username with the actual username of the user.

sudo usermod -aG sudo username

Ex: sudo usermod -aG Devops

```
subbu@UBuntu:~$ cat /etc/passwd | awk -F ':' '{print $1}'
root
daemon
gdm
subbu
vboxadd
Devops
subbu@UBuntu:~$ sudo usermod -aG sudo Devops
subbu@UBuntu:~$
```

Visudo:

1. **Specialized Editor for sudoers File**:
   - visudo is a specialized command for editing the /etc/sudoers file, which defines user permissions for the sudo command.
   - It uses the user's preferred editor (by default vi, but it can be set to another editor through the EDITOR environment variable).
2. **Syntax Checking**:
   - The primary advantage of visudo is that it performs syntax checking on the file before saving.
   - If you make an error in the sudoers file, visudo will catch it and prevent you from saving the incorrect file, avoiding potential system access issues.
3. **Usage**:
   - To edit the sudoers file, you would use:

sudo visudo

Practical Example of Using visudo:

Open sudoers File:

sudo visudo

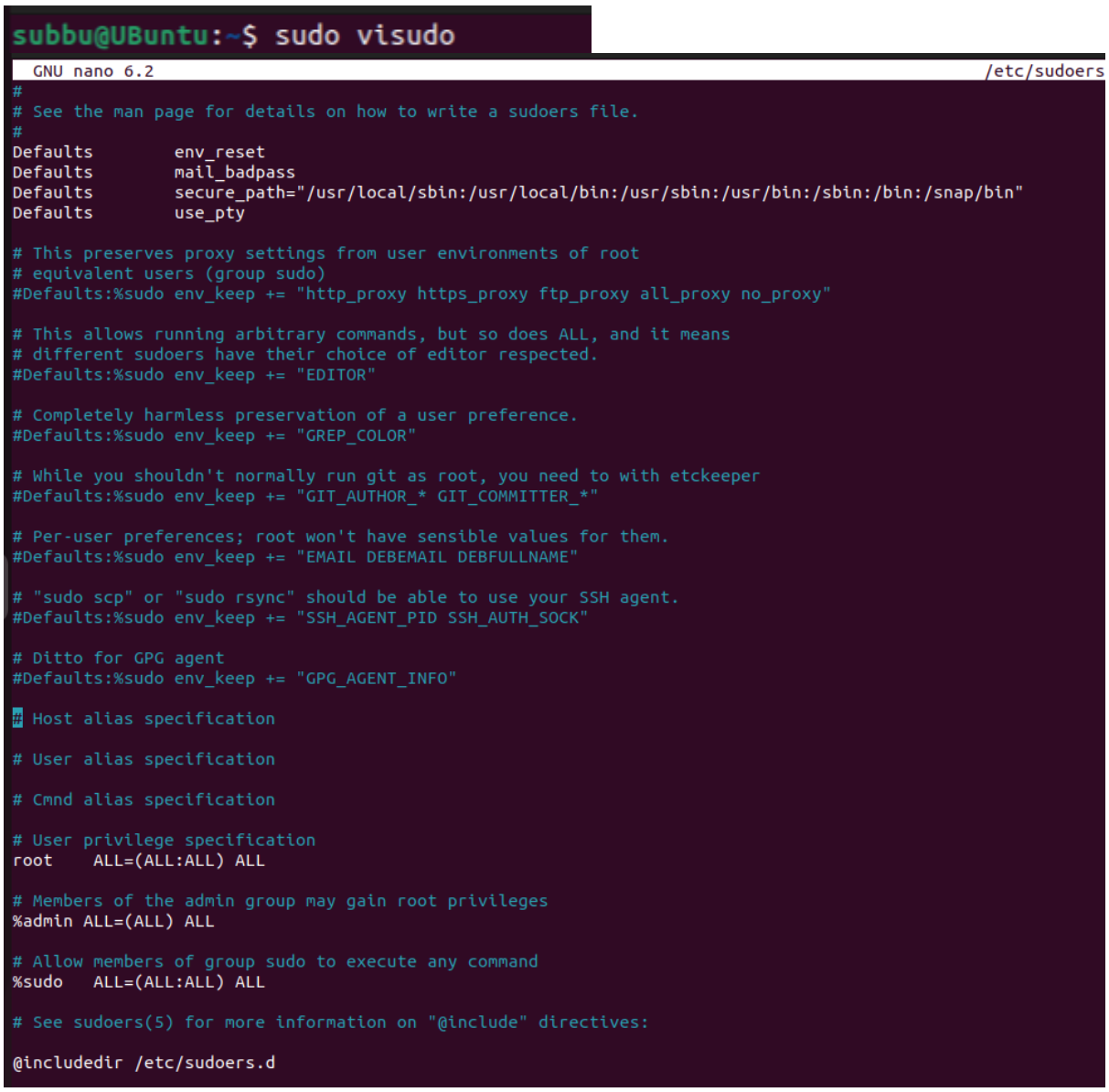
Edit Using Preferred Editor:

- This will open the sudoers file in vi or your preferred text editor.
- Make the necessary changes. For example, adding a user john to have all sudo privileges:

john ALL=(ALL:ALL) ALL

Save and Exit:

- Save the file and exit the editor. In vi, you would typically press Esc, type :wq, and then press Enter.



63. awk:

awk is a powerful text-processing language in Linux. It's commonly used for pattern scanning and processing, where it searches through text files and extracts, processes, and manipulates data based on specified patterns.

Basic Syntax:

awk '/pattern/{action}' file

- pattern: A pattern to search for in the file.
- action: What to do when the pattern is found.

Common Use Cases and Examples:

1. awk '{print}' emp-table.txt

```
subbu@Ubuntu:~$ awk '{print}' table.txt
EmployeeID      FirstName       LastName        DateOfBirth     Position        Department      HireDate        Salary
1               John            Doe             1985-02-15      Software-Engineer       IT              2010-06-01      75000
2               Jane            Smith           1990-08-22      Data-Analyst    Business-Intelligence   2015-04-15      65000
3               Mike            Johnson         1987-12-05      Project-Manager IT              2012-09-23      85000
4               Sarah           Brown           1988-11-19      Accountant      Finance         2013-07-10      62000
5               Chris           Miller          1995-03-11      Marketing-Specialist    Marketing       2018-01-20      60000
```

or awk '{print $0}' emp-table.txt

```
subbu@Ubuntu:~$ awk '{print}' table.txt
EmployeeID      FirstName       LastName        DateOfBirth     Position        Department      HireDate        Salary
1               John            Doe             1985-02-15      Software-Engineer       IT              2010-06-01      75000
2               Jane            Smith           1990-08-22      Data-Analyst    Business-Intelligence   2015-04-15      65000
3               Mike            Johnson         1987-12-05      Project-Manager IT              2012-09-23      85000
4               Sarah           Brown           1988-11-19      Accountant      Finance         2013-07-10      62000
5               Chris           Miller          1995-03-11      Marketing-Specialist    Marketing       2018-01-20      60000
```

2. awk '/Manger/ {print}' table.txt

```
subbu@Ubuntu:~$ awk '/Manager/{print}' table.txt
3               Mike            Johnson         1987-12-05      Project-Manager IT              2012-09-23      85000
subbu@Ubuntu:~$
```

3. awk '{print $1}' table.txt---To Print First Column

```
subbu@Ubuntu:~$ awk '{print $1}' table.txt
EmployeeID
1
2
3
4
5
```

4. awk '{print $1,$4}' table.txt—To Print Multiple Columns

```
subbu@Ubuntu:~$ awk '{print $2,$8}' table.txt
FirstName Salary
John 75000
Jane 65000
Mike 85000
Sarah 62000
Chris 60000
subbu@Ubuntu:~$
```

5. awk ‘{print NF}’ table.txt—To Print last column

The AWK command awk -F ',' '{print NF}' emp-table.txt is used to print the number of fields in each line of the file emp-table.txt, with fields separated by commas.

```
subbu@Ubuntu:~$ awk '{print NF}' table.txt
8
8
8
8
8
8
subbu@Ubuntu:~$
```

6. awk ‘{print NR}’ table.txt—To Display the Row or line number of each line

```
subbu@Ubuntu:~$ awk '{print NR}' table.txt
1
2
3
4
5
6
```

7. awk ‘{print NR,$6}’ table.txt

```
subbu@Ubuntu:~$ awk '{print NR, $6}' table.txt
1 Department
2 IT
3 Business-Intelligence
4 IT
5 Finance
6 Marketing
subbu@Ubuntu:~$
```

8. awk ‘NR==4, NR==7 {print NR,$1,$4}’ table.txt

The AWK command awk 'NR==4, NR==7 {print NR, $1, $4}' table.txt is used to print specific fields from lines 4 to 7 of the file table.txt.

```
subbu@Ubuntu:~$ awk 'NR==4, NR==7 {print NR,$2,$8}' table.txt
4 Mike 85000
5 Sarah 62000
6 Chris 60000
subbu@Ubuntu:~$
```

9. **awk '{print $2,"->",$6,"->",$8}' table.txt**

```
subbu@Ubuntu:~$ awk '{print $2,"->",$6,"->",$8}' table.txt
FirstName -> Department -> Salary
John -> IT -> 75000
Jane -> Business-Intelligence -> 65000
Mike -> IT -> 85000
Sarah -> Finance -> 62000
Chris -> Marketing -> 60000
subbu@Ubuntu:~$
```

64. wget:

The wget command in Linux is a powerful utility used to download files from the internet. It's especially useful for automating downloads and working in scripts.

Basic Usage:

1. Download a Single File:

For example, to download the file to install the Puppet server package on Ubuntu, enter:

wget https://apt.puppetlabs.com/puppet8-release-jammy.deb

```
subbu@Ubuntu:~$ wget https://apt.puppetlabs.com/puppet8-release-jammy.deb
--2024-06-23 22:24:13--  https://apt.puppetlabs.com/puppet8-release-jammy.deb
Resolving apt.puppetlabs.com (apt.puppetlabs.com)... 2600:9000:2634:2a00:1d:fc37:1cc0:93a1, 2600:9000:2634:ee00
Connecting to apt.puppetlabs.com (apt.puppetlabs.com)|2600:9000:2634:2a00:1d:fc37:1cc0:93a1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11530 (11K) [application/vnd.debian.binary-package]
Saving to: 'puppet8-release-jammy.deb'

puppet8-release-jammy.deb                          100%[==========================================================

2024-06-23 22:24:14 (198 MB/s) - 'puppet8-release-jammy.deb' saved [11530/11530]

subbu@Ubuntu:~$
```

65. crontab:

The crontab command in Linux is used to schedule tasks to be executed at specific times. This is done using a configuration file called a "crontab", which specifies shell commands to run periodically on a given schedule.

Practical Usage in Real Time:

DB Backups

Unused Docker Images, containers removing.

Deploying some application at a particular time.

1. View Your Crontab:

crontab -l

This command lists your current crontab entries.

2. Edit Your Crontab:

crontab -e

This command opens your crontab file in the default text editor, allowing you to add or modify cron jobs.

3. Remove Your Crontab:

crontab -r

Crontab File Format

Each line in a crontab file represents a task and follows this format:

```
* * * * * command
- - - - -
| | | | |
| | | | +----- Day of the week (0 - 7) (Sunday = 0 or 7)
| | | +------- Month (1 - 12)
| | +--------- Day of the month (1 - 31)
| +----------- Hour (0 - 23)
+------------- Minute (0 - 59)
```

To Create a cron.allow file /etc directory:

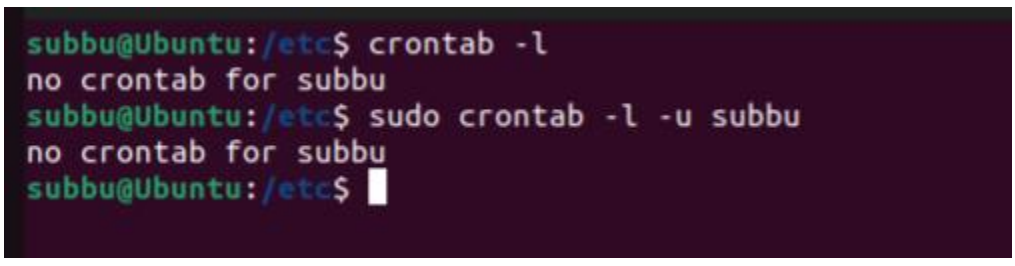
Sudo touch /etc/cron.allow

Add a user--subbu

List All Crontabs (Superuser):

sudo crontab -l -u username

This command lists the crontab entries for a specific user.



Examples:

1. **Run a Command Every Minute:**

   * * * * * /path/to/command

2. **Run a Command at 2:30 AM Every Day:**
   30 2 * * * /path/to/command
3. **Run a Command at 3 PM Every Friday:**
   0 15 * * 5 /path/to/command
4. **Run a Command at 1 AM on the First Day of Every Month:**
   0 1 1 * * /path/to/command
5. **Run a Command Every 15 Minutes:**
   */15 * * * * /path/to/command
6. **Run a Command Every Sunday at Midnight:**
   0 0 * * 0 /path/to/command

To run a script for every minute using crontab:

Crontab -e

*/1 * * * * /home/subbu/test.sh > /home/subbu/test.log

Save and quit

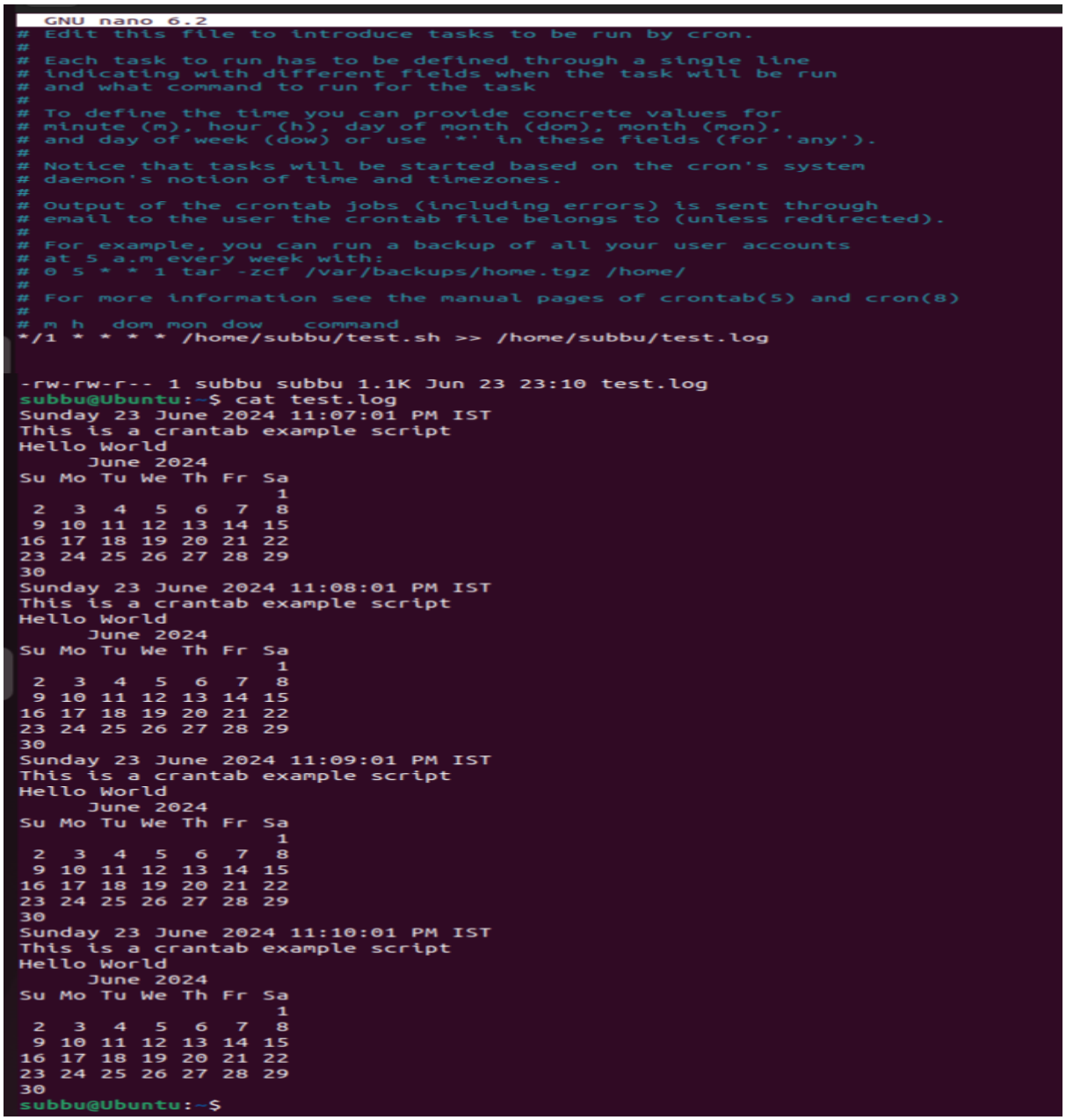
```
subbu@Ubuntu:~$ /home/subbu/test.sh > /home/subbu/test.log
subbu@Ubuntu:~$ ls -lrth
total 72K
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Videos
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Templates
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Public
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Pictures
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Music
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Downloads
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Documents
drwx------ 4 subbu subbu 4.0K Mar 18 06:41 snap
-rw-rw-r-- 1 subbu subbu  12K Jun 18 12:07 puppet8-release-jammy.deb
drwxr-xr-x 3 subbu subbu 4.0K Jun 18 22:06 Desktop
drwxrwxr-x 3 subbu subbu 4.0K Jun 19 22:58 Linux-Practice
drwxrwxr-x 3 subbu subbu 4.0K Jun 21 15:49 Python
-rw-rw-r-- 1 subbu subbu  498 Jun 23 21:20 emp-table.txt
-rw-rw-r-- 1 subbu subbu  430 Jun 23 22:06 table.txt
-r-x--x--x 1 subbu subbu   81 Jun 23 22:29 test.sh
-rw-rw-r-- 1 subbu subbu  265 Jun 23 23:03 test.log
subbu@Ubuntu:~$ cat test.log
Sunday 23 June 2024 11:03:27 PM IST
This is a crantab example script
Hello World
     June 2024
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
subbu@Ubuntu:~$
```

To run a script for every minute using crontab and append the result to a file:

Crontab -e

*/1 * * * * /home/subbu/test.sh >> /home/subbu/test.log

Save and quit



Another Example Crontab Entry:

Edit crontab

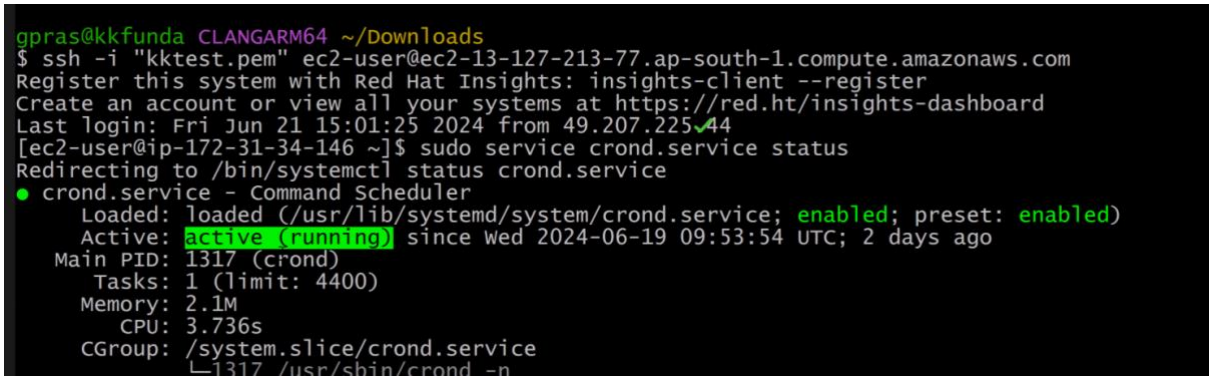
crontab -e

Add the following line to the crontab file to run a backup script every day at 2:30 AM

30 2 * * * /home/user/scripts/backup.sh >> /home/user/logs/backup.log 2>&1

To check the status of cron.service:

Sudo service crond.service status



Special Strings

Crontab also supports several special strings which can replace the five time-and-date fields:

- **@reboot :** Run once, at startup.
- **@yearly or @annually** : Run once a year, "0 0 1 1 *".
- **@monthly :** Run once a month, "0 0 1 * *".
- **@weekly :** Run once a week, "0 0 * * 0".
- **@daily or @midnight** : Run once a day, "0 0 * * *".
- **@hourly :** Run once an hour, "0 * * * *".

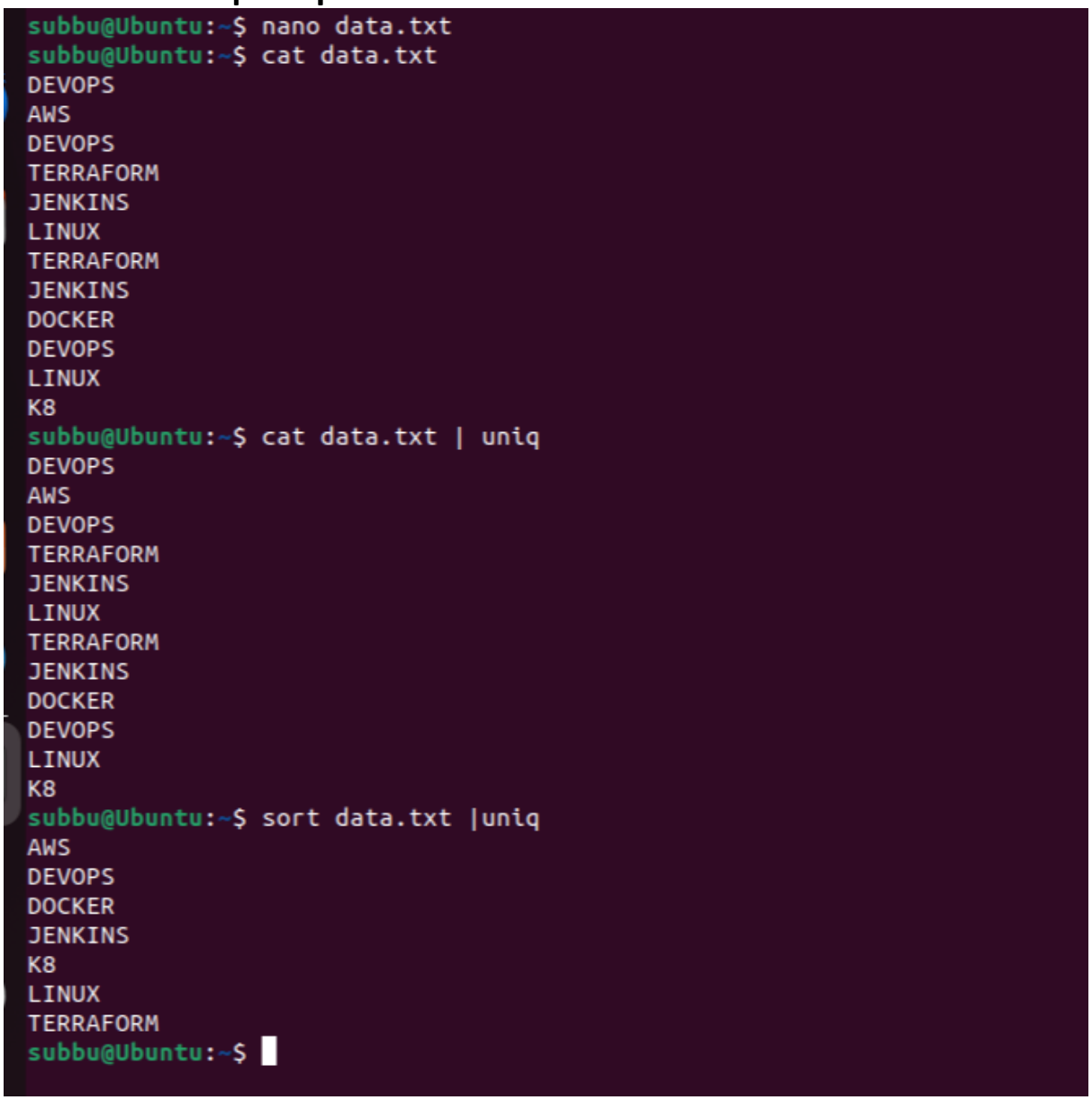
Example using special strings:

@daily /path/to/command

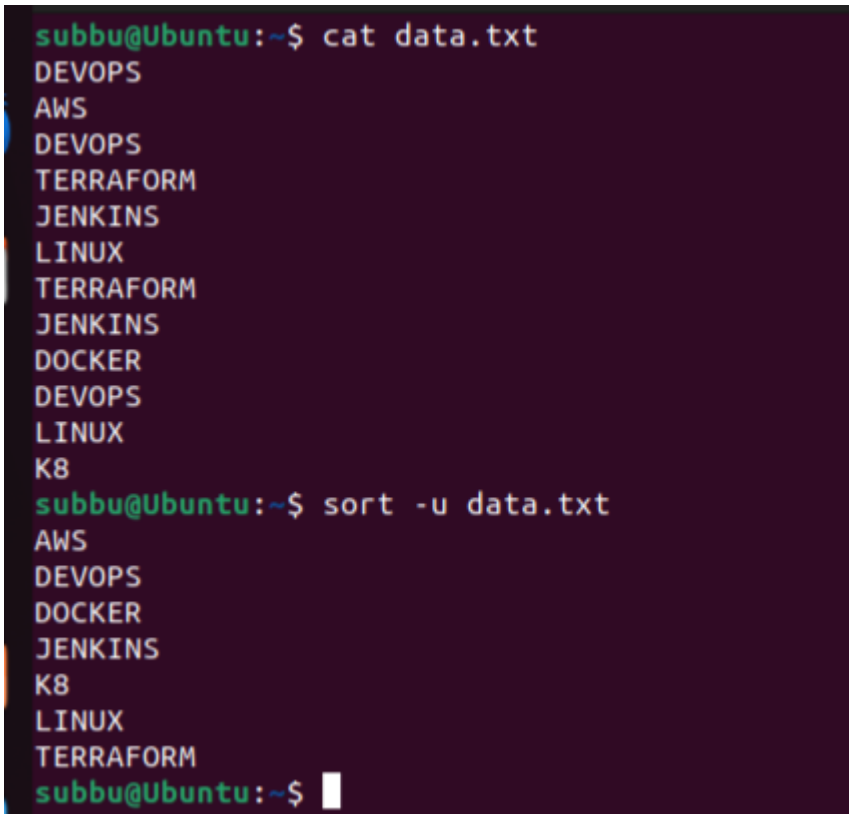
66. uniq :

The uniq command in Linux is used to report or filter out repeated lines in a file. It is often used in combination with other commands, especially those that produce sorted output, such as sort.

Sort filename | uniq



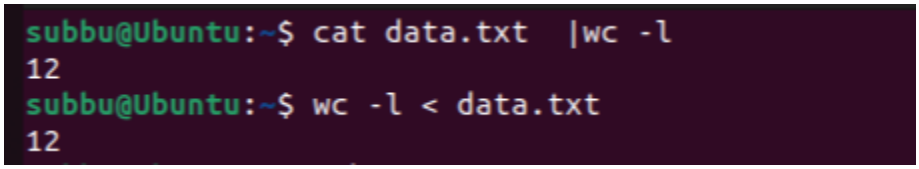
Alternative way to get the uniq values from a dataset using sort command:



67. '<' :passing input to a command

Ex: wc -l < data.txt

The command wc -l < data.txt is used to count the number of lines in the file data.txt.



68. tee

The tee command in Linux reads from the standard input and writes to both the standard output and one or more files simultaneously. It's a useful command when you want to see the output of a command on the terminal while also saving it to a file.

tee [OPTIONS] [FILE...]

- OPTIONS: Various options that modify the behaviour of tee.
- FILE: The file(s) to write to. If not specified, tee only writes to the standard output.

For example, suppose you have a file named example.txt and you want to append some text to it while also seeing the output on the terminal:

echo "Hello, World!" | tee -a example.txt

```
subbu@Ubuntu:~$ echo "Hello, World!" | tee -a example.txt
Hello, World!
subbu@Ubuntu:~$ cat example.txt
Hello, World!
subbu@Ubuntu:~$ ls -lrth
total 84K
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Videos
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Templates
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Public
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Pictures
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Music
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Downloads
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Documents
drwx------ 4 subbu subbu 4.0K Mar 18 06:41 snap
-rw-rw-r-- 1 subbu subbu  12K Jun 18 12:07 puppet8-release-jammy.deb
drwxr-xr-x 3 subbu subbu 4.0K Jun 18 22:06 Desktop
drwxrwxr-x 3 subbu subbu 4.0K Jun 19 22:58 Linux-Practice
drwxrwxr-x 3 subbu subbu 4.0K Jun 21 15:49 Python
-rw-rw-r-- 1 subbu subbu  498 Jun 23 21:20 emp-table.txt
-rw-rw-r-- 1 subbu subbu  430 Jun 23 22:06 table.txt
-r-x--x--x 1 subbu subbu   81 Jun 23 22:29 test.sh
-rw-rw-r-- 1 subbu subbu   83 Jun 23 23:14 data.txt
-rw-rw-r-- 1 subbu subbu 4.4K Jun 23 23:23 test.log
-rw-rw-r-- 1 subbu subbu   14 Jun 23 23:23 example.txt
subbu@Ubuntu:~$
```

Ex: ls -lrth | tee directory_listing.txt

```
subbu@Ubuntu:~$ ls -lrth | tee directory_listing.txt
total 84K
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Videos
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Templates
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Public
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Pictures
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Music
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Downloads
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Documents
drwx------ 4 subbu subbu 4.0K Mar 18 06:41 snap
-rw-rw-r-- 1 subbu subbu  12K Jun 18 12:07 puppet8-release-jammy.deb
drwxr-xr-x 3 subbu subbu 4.0K Jun 18 22:06 Desktop
drwxrwxr-x 3 subbu subbu 4.0K Jun 19 22:58 Linux-Practice
drwxrwxr-x 3 subbu subbu 4.0K Jun 21 15:49 Python
-rw-rw-r-- 1 subbu subbu  498 Jun 23 21:20 emp-table.txt
-rw-rw-r-- 1 subbu subbu  430 Jun 23 22:06 table.txt
-r-x--x--x 1 subbu subbu   81 Jun 23 22:29 test.sh
-rw-rw-r-- 1 subbu subbu   83 Jun 23 23:14 data.txt
-rw-rw-r-- 1 subbu subbu   14 Jun 23 23:23 example.txt
-rw-rw-r-- 1 subbu subbu 4.7K Jun 23 23:24 test.log
-rw-rw-r-- 1 subbu subbu    0 Jun 23 23:24 directory_listing.txt
```

69. zcat:

It is used to display the contents of compressed files (usually compressed with gzip) to the standard output (usually the terminal). It is essentially the same as using cat on an uncompressed file, but it works for compressed files.

```
subbu@Ubuntu:~$ ls -lrth
total 92K
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Videos
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Templates
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Public
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Pictures
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Music
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Downloads
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Documents
drwx------ 4 subbu subbu 4.0K Mar 18 06:41 snap
-rw-rw-r-- 1 subbu subbu  12K Jun 18 12:07 puppet8-release-jammy.deb
drwxr-xr-x 3 subbu subbu 4.0K Jun 18 22:06 Desktop
drwxrwxr-x 3 subbu subbu 4.0K Jun 19 22:58 Linux-Practice
drwxrwxr-x 3 subbu subbu 4.0K Jun 21 15:49 Python
-rw-rw-r-- 1 subbu subbu  498 Jun 23 21:20 emp-table.txt
-rw-rw-r-- 1 subbu subbu  430 Jun 23 22:06 table.txt
-r-x--x--x 1 subbu subbu   81 Jun 23 22:29 test.sh
-rw-rw-r-- 1 subbu subbu   83 Jun 23 23:14 data.txt
-rw-rw-r-- 1 subbu subbu   14 Jun 23 23:23 example.txt
-rw-rw-r-- 1 subbu subbu 1.1K Jun 23 23:24 directory_listing.txt
-rw-rw-r-- 1 subbu subbu 5.7K Jun 23 23:28 test.log
-rw-rw-r-- 1 subbu subbu  432 Jun 23 23:28 testlog.zip
subbu@Ubuntu:~$ zcat testlog.zip
Sunday 23 June 2024 11:07:01 PM IST
This is a crantab example script
Hello World
     June 2024
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
Sunday 23 June 2024 11:08:01 PM IST
This is a crantab example script
Hello World
     June 2024
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

70. zgrep :

zgrep: It is used to search for a pattern in one or more compressed files using grep. It's particularly useful when you have large log files compressed with gzip and you want to search for specific patterns without decompressing the files first.

zgrep [OPTIONS] PATTERN [FILE...]

OPTIONS: Various options that modify the behavior of zgrep.

PATTERN: The pattern to search for within the compressed file(s).

FILE: The gzip-compressed file(s) to search within.

Some common options of the zgrep command include:

- -i: Ignore case distinctions in both the pattern and input files.
- -n: Prefix each line of output with the 1-based line number within its input file.
- -c: Suppress normal output; instead, print a count of matching lines for each input file.
- -v: Invert the sense of matching, to select non-matching lines.

```
subbu@Ubuntu:~$ ls -lrth
total 92K
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Videos
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Templates
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Public
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Pictures
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Music
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Downloads
drwxr-xr-x 2 subbu subbu 4.0K Mar 18 06:34 Documents
drwx------ 4 subbu subbu 4.0K Mar 18 06:41 snap
-rw-rw-r-- 1 subbu subbu  12K Jun 18 12:07 puppet8-release-jammy.deb
drwxr-xr-x 3 subbu subbu 4.0K Jun 18 22:06 Desktop
drwxrwxr-x 3 subbu subbu 4.0K Jun 19 22:58 Linux-Practice
drwxrwxr-x 3 subbu subbu 4.0K Jun 21 15:49 Python
-rw-rw-r-- 1 subbu subbu  498 Jun 23 21:20 emp-table.txt
-rw-rw-r-- 1 subbu subbu  430 Jun 23 22:06 table.txt
-r-x--x--x 1 subbu subbu   81 Jun 23 22:29 test.sh
-rw-rw-r-- 1 subbu subbu   83 Jun 23 23:14 data.txt
-rw-rw-r-- 1 subbu subbu   14 Jun 23 23:23 example.txt
-rw-rw-r-- 1 subbu subbu 1.1K Jun 23 23:24 directory_listing.txt
-rw-rw-r-- 1 subbu subbu  432 Jun 23 23:28 testlog.zip
-rw-rw-r-- 1 subbu subbu 6.5K Jun 23 23:31 test.log
subbu@Ubuntu:~$ zgrep "Hello World" testlog.zip
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
subbu@Ubuntu:~$
```

You can also specify multiple gzip-compressed files to search within:

zgrep "error" file1.zip file2.zip

71. nohup:

nohup stands for "no hang up". It is a command used in Unix-like operating systems to run a command immune to hangups, or more generally, to protect the command from being terminated if the user's session is disconnected or ended.

Syntax: nohup command [arguments] &

- command [arguments]: The command you want to run with nohup.
- &: This puts the command in the background, allowing you to continue using the terminal while the command runs.
- When you run a command with nohup, it redirects standard output (stdout) and standard error (stderr) to a file named nohup.out in the current directory. This allows the command to continue running even if the terminal session is terminated.

Here's an example of how you can use nohup:

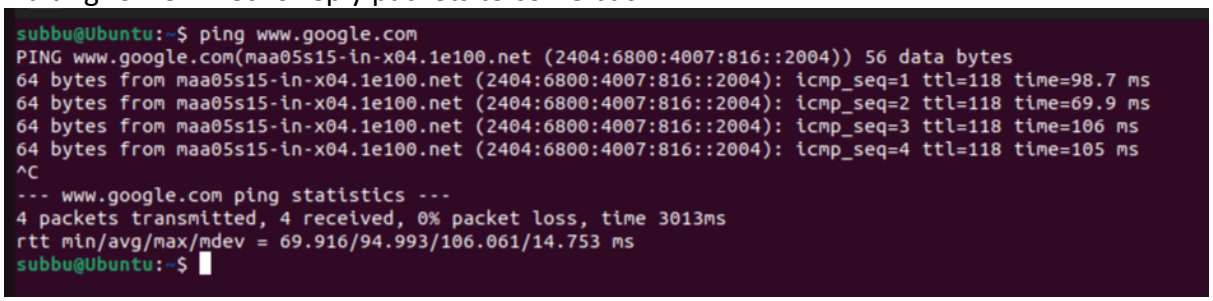
nohup ./myscript.sh &

In this example, ./myscript.sh is the script you want to run in the background using nohup. The & at the end puts the command in the background, allowing you to continue using the terminal while the script runs.

The output of the script, including any stdout and stderr messages, will be captured in the nohup.out file in the current directory.

72. ping :

The ping command in Linux is used to test the reachability of a host on a network by sending ICMP (Internet Control Message Protocol) echo request packets to the target host and waiting for ICMP echo reply packets to come back.



Some common options of the ping command include:

- -c COUNT: Specify the number of echo requests to send before stopping.
- -i SECONDS: Set the interval between echo requests (in seconds).
- -w TIMEOUT: Set the timeout value (in seconds) for each echo request.
- -s SIZE: Set the size of the ICMP packet (in bytes).
- -q: Quiet output. Only display summary statistics at the end.

Ex: ping -c 5 -w 2 example.com

73. curl:

The curl command in Linux is a powerful tool used to transfer data to or from a server using various protocols, including HTTP, HTTPS, FTP, SCP, SFTP, and more. It supports a wide range of options and features, making it versatile for various use cases, such as testing APIs, downloading files, and automation tasks.

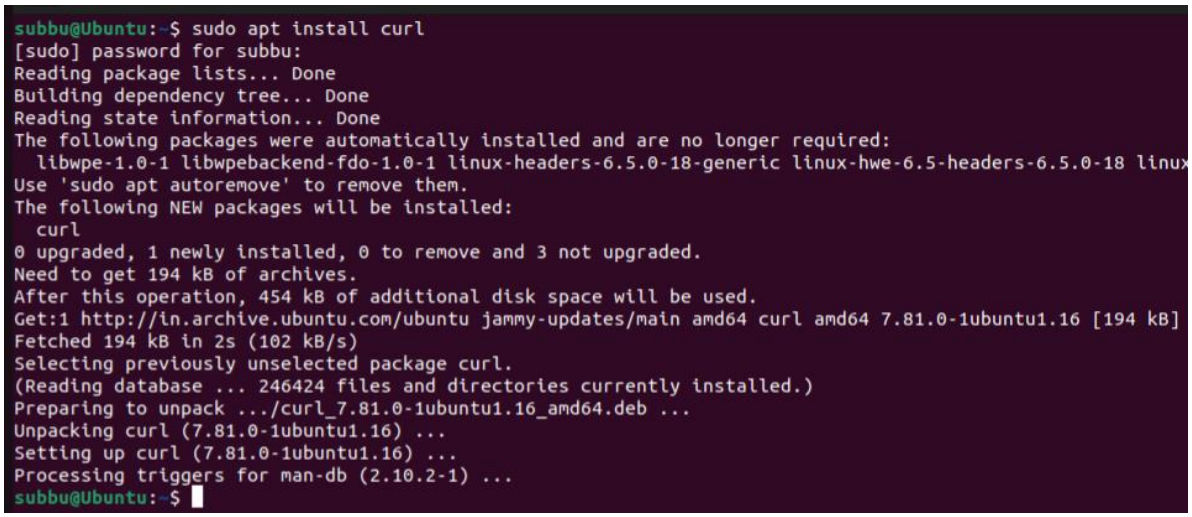
curl [OPTIONS] URL

- OPTIONS: Various options that modify the behaviour of curl.
- URL: The URL of the resource you want to transfer data to or from.

When you run the curl command, it sends an HTTP request to the specified URL by default and displays the response body to the standard output (stdout).

Some common options of the curl command include:

- -X METHOD: Specify the HTTP request method (e.g., GET, POST, PUT).
- -H HEADER: Add custom headers to the request.
- -d DATA: Send data in the request body (e.g., for POST requests).
- -o FILE: Write output to a file instead of stdout.
- -L: Follow redirects.
- -i: Include HTTP response headers in the output.
- -v: Verbose mode. Display additional information about the request and response.



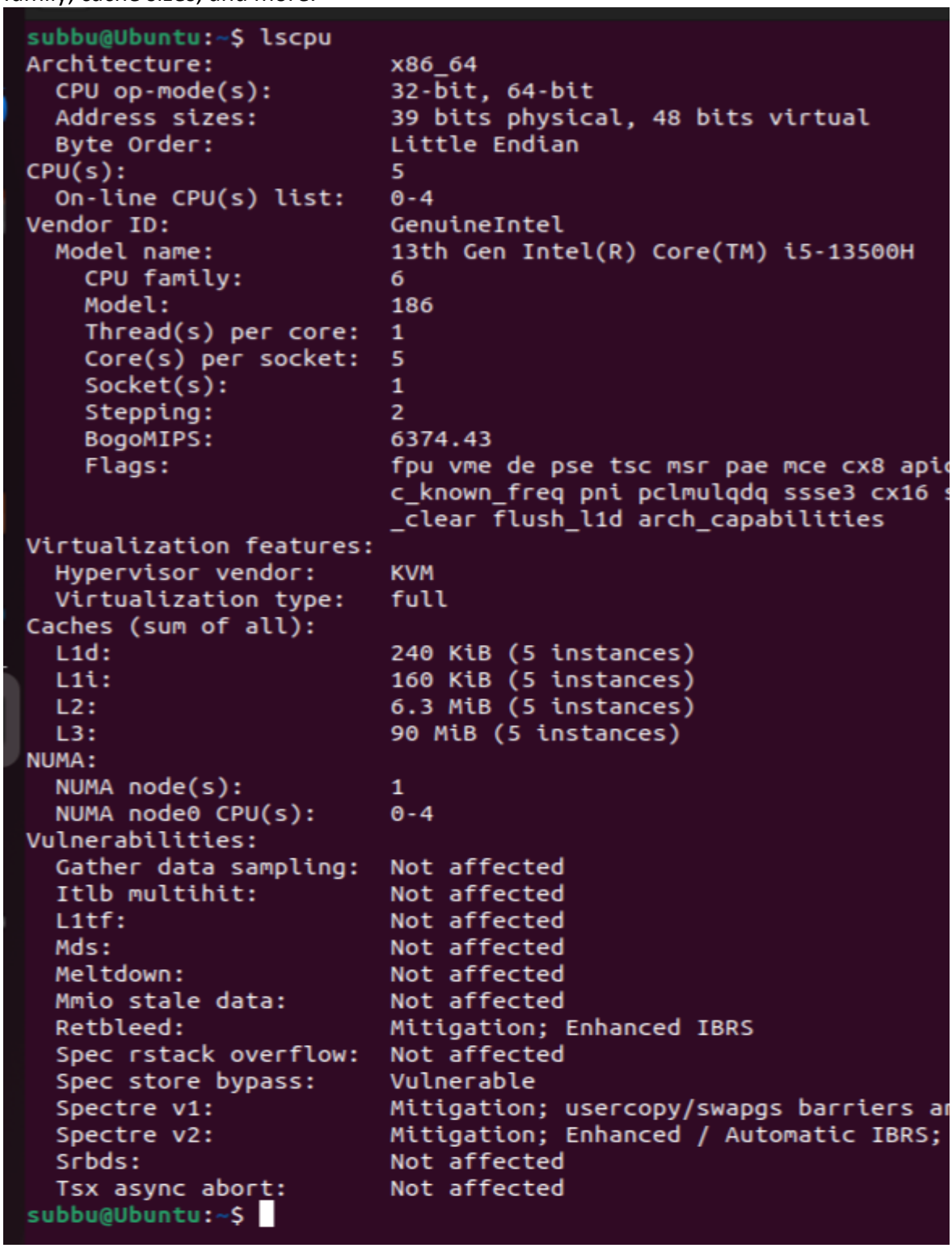
```
subbu@Ubuntu:~$ whatis curl
curl (1)             - transfer a URL
subbu@Ubuntu:~$ curl https://www.gnu.org/gnu/gnu.html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="author" href="mailto:webmasters@gnu.org" />
<link rel="icon" type="image/png" href="/graphics/gnu-head-mini.png" />
<meta name="ICBM" content="42.355469,-71.058627" />
<link rel="stylesheet" type="text/css" href="/layout.min.css" media="screen" />
<link rel="stylesheet" type="text/css" href="/print.min.css" media="print" />



<!-- Parent-Version: 1.98 -->
<!-- This page is derived from /server/standards/boilerplate.html -->
<title>About the GNU Operating System
- GNU project - Free Software Foundation</title>
<style type="text/css" media="print,screen"><!--
#dynamic-duo { display: none; }
@media (min-width: 45em) {
   .short-lines { width: 48em; max-width: 100%; }
   #dynamic-duo {
      display: block;
      padding: .9em;
      background: #f9f9f9;
      border: .3em solid #acc890;
      margin-top: 5em;
   }
   #dynamic-duo p strong {
      font-size: 1.3em;
   }
   #dynamic-duo img { width: 100%; }
}
--></style>
<!-- begin translist file -->
```

74. cat /proc/cpuinfo

```
subbu@Ubuntu:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 186
model name      : 13th Gen Intel(R) Core(TM) i5-13500H
stepping        : 2
microcode       : 0xffffffff
cpu MHz         : 3187.216
cache size      : 18432 KB
physical id     : 0
siblings        : 5
core id         : 0
cpu cores       : 5
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
```

75. lscpu :

The lscpu command in Linux is used to display information about the CPU architecture and processing units on a system. It provides detailed information about the CPU such as architecture, model name, number of CPU sockets, cores per socket, threads per core, CPU family, cache sizes, and more.



76. lshw

The lshw command in Linux is a powerful tool used to display detailed information about hardware components on a system. It stands for "list hardware" and provides comprehensive information about various hardware components such as the CPU, memory, storage devices, network interfaces, motherboard, and more.

```
subbu@Ubuntu:~$ lshw
WARNING: you should run this program as super-user.
subbu
    description: Computer
    width: 64 bits
    capabilities: smp vsyscall32
  *-core
       description: Motherboard
       physical id: 0
     *-memory
          description: System memory
          physical id: 0
          size: 10GiB
     *-cpu
          product: 13th Gen Intel(R) Core(TM) i5-13500H
          vendor: Intel Corp.
          physical id: 1
          bus info: cpu@0
          version: 6.186.2
          width: 64 bits
          capabilities: fpu fpu_exception wp vme de pse tsc msr
nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 cx16 sse4_
r flush_l1d arch_capabilities
          configuration: microcode=4294967295
     *-pci
          description: Host bridge
          product: 440FX - 82441FX PMC [Natoma]
          vendor: Intel Corporation
          physical id: 100
          bus info: pci@0000:00:00.0
          version: 02
          width: 32 bits
          clock: 33MHz
        *-isa
             description: ISA bridge
             product: 82371SB PIIX3 ISA [Natoma/Triton II]
             vendor: Intel Corporation
             physical id: 1
             bus info: pci@0000:00:01.0
             version: 00
             width: 32 bits
             clock: 33MHz
             capabilities: isa bus_master
             configuration: latency=0
```

77. sudo shutdown

The shutdown command in Linux is used to bring the system down in a safe and controlled manner. It allows you to schedule a system shutdown, restart, or power off operation. Typically, this command requires superuser privileges, so you need to prepend sudo to it.

sudo shutdown [OPTIONS] TIME [MESSAGE]

- OPTIONS: Various options that modify the behaviour of shutdown.
- TIME: The time when the shutdown should occur. This can be a specific time (e.g., 10:00, 23:30) or the number of minutes until shutdown (e.g., +5, +60 for 5 or 60 minutes from now).
- MESSAGE: An optional message to display to users before the shutdown.

Some common options of the shutdown command include:

- -r: Restart the system after shutdown.
- -h: Poweroff the system after shutdown.
- -c: Cancel a pending shutdown.

For example, to shutdown the system immediately:
sudo shutdown now

To schedule a shutdown for 10 minutes from now:
sudo shutdown +10

To schedule a shutdown for 10:00 PM:
sudo shutdown 22:00

You can also include an optional message to be displayed to users before the shutdown:
sudo shutdown +5 "System maintenance in progress. Please save your work."

78. cut :

The cut command in Linux is used to extract specific parts of lines from files or from the standard input (stdin). It's particularly useful for working with structured data, such as delimited text files, where you want to extract specific fields or columns.
Here's the basic syntax of the cut command:

cut [OPTIONS] [FILE]

- OPTIONS: Various options that modify the behaviour of cut.
- FILE: The file from which to extract data. If not specified, cut reads from the standard input.

The primary function of cut is to extract portions of each line of a file or stdin based on specified criteria, such as byte position, character position, or field delimiter. The extracted portions can then be printed to the standard output.

Some common options of the cut command include:

- -f FIELDS: Select only these fields; also supports a comma-separated list or a range of fields (e.g., 1,3, 1-3).
- -d DELIMITER: Specify the field delimiter character.
- -c CHARACTERS: Select only these characters.
- -b BYTES: Select only these bytes.
- -s: Suppress lines with no field delimiter character.
- --output-delimiter: Specify the output delimiter character.

Example Data:

```
subbu@Ubuntu:~$ cat info.txt
Name,Age,Gender
John,30,Male
Alice,25,Female
subbu@Ubuntu:~$
```

1.To extract the first column (Name) from the file:

cut -d',' -f1 info.txt

```
subbu@Ubuntu:~$ cut -d ',' -f1 info.txt
Name
John
Alice
subbu@Ubuntu:~$
```

79. cal:

The cal command in Linux is used to display a calendar of a specific month or year. It displays the calendar for the current month by default.

cal [OPTIONS] [MONTH] [YEAR]

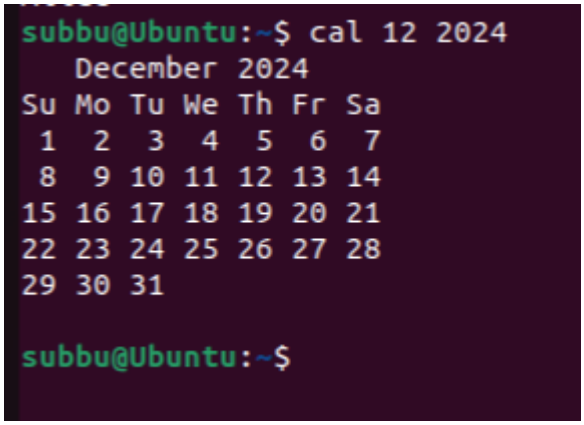
- **OPTIONS**: Various options that modify the behavior of cal.
- **MONTH**: The month to display (1-12). If not specified, the current month is displayed.
- **YEAR**: The year to display. If not specified, the current year is used.

Some common options of the cal command include:

- -1: Display the calendar in a single-column format.
- -3: Display the calendar in a three-month format (current month and previous and next months).
- -j: Display the Julian calendar (number of days since January 1, 4713 BC).
- -y: Display the calendar for the entire year.
- -A NUM: Display the specified number of months after the current month.
- -B NUM: Display the specified number of months before the current month.

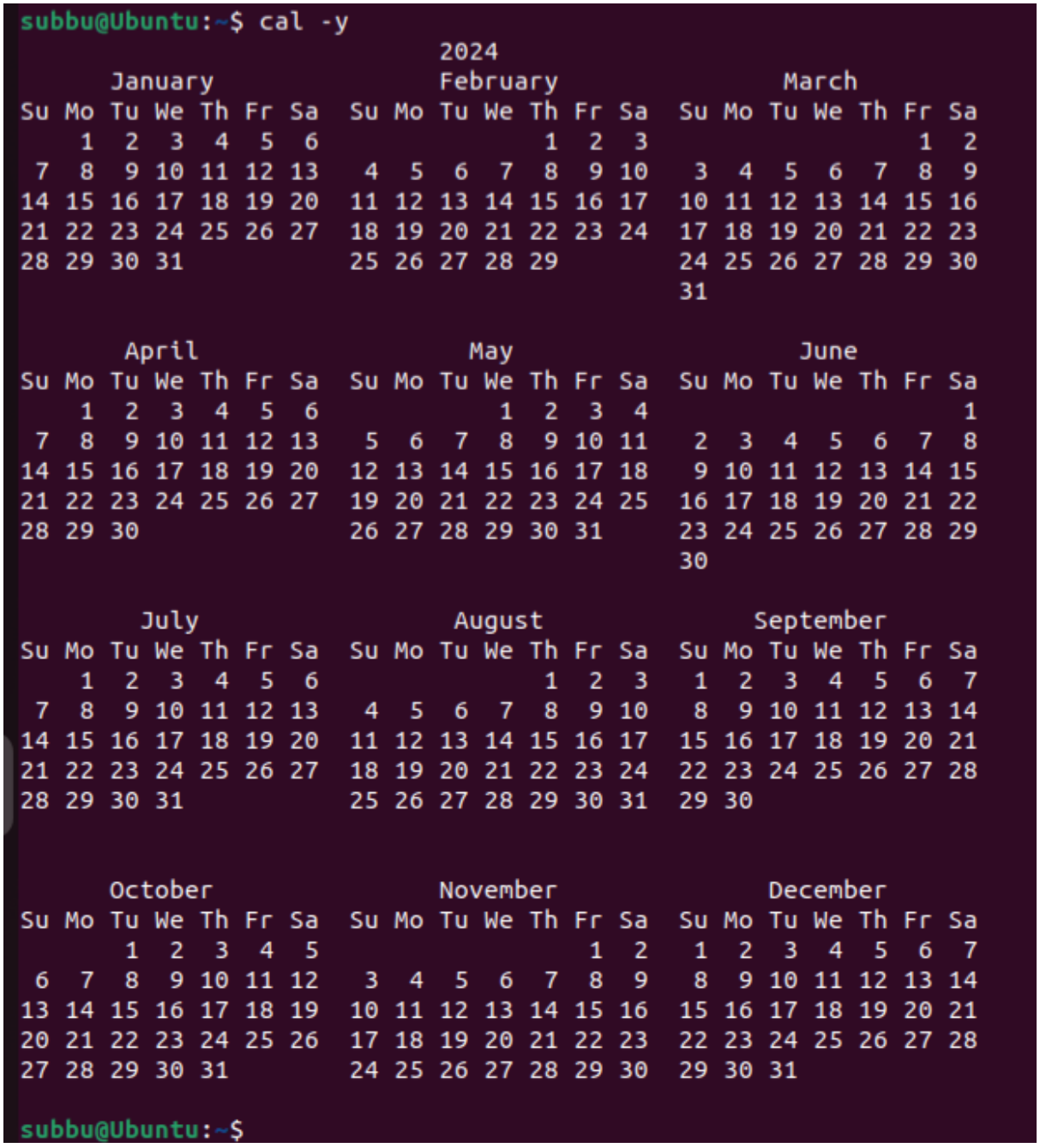
1. For example, to display the calendar for the current month:

2. To display the calendar for a specific month and year (e.g., dec 2024):



3. To display the calendar for the entire year:

cal -y



```
subbu@Ubuntu:~$ cal -y
                                2024
      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                  1  2  3                  1  2
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   3  4  5  6  7  8  9
14 15 16 17 18 19 20  11 12 13 14 15 16 17  10 11 12 13 14 15 16
21 22 23 24 25 26 27  18 19 20 21 22 23 24  17 18 19 20 21 22 23
28 29 30 31           25 26 27 28 29        24 25 26 27 28 29 30
                                            31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6              1  2  3  4                     1
 7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20  12 13 14 15 16 17 18   9 10 11 12 13 14 15
21 22 23 24 25 26 27  19 20 21 22 23 24 25  16 17 18 19 20 21 22
28 29 30              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

        July                 August               September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                  1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17  15 16 17 18 19 20 21
21 22 23 24 25 26 27  18 19 20 21 22 23 24  22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31  29 30

      October              November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                  1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31

subbu@Ubuntu:~$
```

Other Commands

ssh

ssh username@hostname

ssh username@ip

etc/ssh/sshd_config

change------Not working in Ubuntu

/etc/shadow

dnf----Installing Packages