

* **Navigate to your home directory**:

**cd**

* **List the contents of your home directory**:

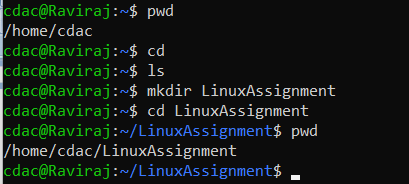
**ls**

* **Check if the "LinuxAssignment" directory exists, and create it if necessary**:

**mkdir LinuxAssignment**

* **Move into the "LinuxAssignment" directory**:

cd **LinuxAssignment**





* **Navigate to the "LinuxAssignment" directory**:

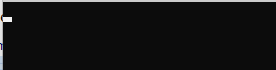
cd LinuxAssignment/

* **Create a new file named "file1.txt"**:

**touch file1**.txt

* **Display the contents of "file1.txt"** (it will be empty initially):

**cat file1.txt**

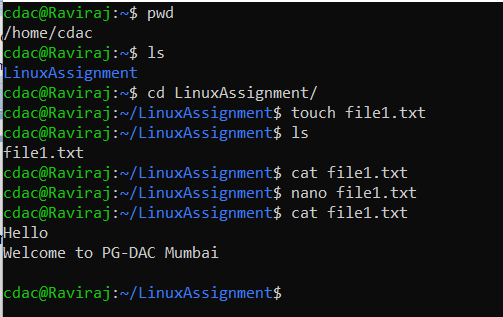
****

* **Add some text to the file**:

nano file1.txt

* **Display the contents again to verify**:

cat file1.txt





* **Navigate to the "LinuxAssignment" directory**:

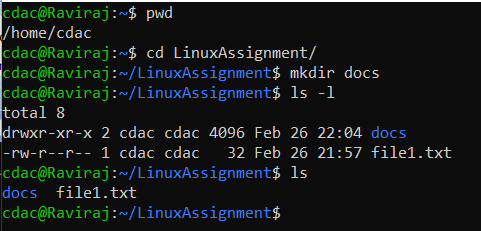
**cd LinuxAssignment/**

* **Create a new directory named "docs" inside "LinuxAssignment"**:

**mkdir docs**

* **Verify that the directory was created**:

**ls**

****



* **Navigate to the "LinuxAssignment" directory**:

**cd LinuxAssignment/**

* **Copy "file1.txt" to the "docs" directory and rename it to "file2.txt"**:

**cp file1.txt docs/file2.txt**

Or

**Copy "file1.txt" to the "docs" directory**:

**cp file1**.txt docs/

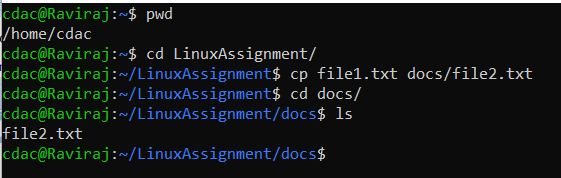
**Rename "file1.txt" to "file2.txt" inside the "docs" directory**:

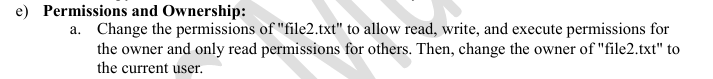
**mv docs/file1**.txt docs/file2.txt

* **Verify that "file2.txt" exists in the "docs" directory**:

**cd docs/**

**ls**

****



* **Change permissions of "file2.txt"**

chmod 744 ~/LinuxAssignment/docs/file2.txt

**Explanation of chmod 744**:

* **7 (Owner)** → The first digit (7) represents the **owner's** permissions:

r(read)

w(write)

x(Execute)

7 is derived from: rwx = 4+2+1 = 7

* **4 (Group)** → The second digit (4) represents the **group’s** permissions

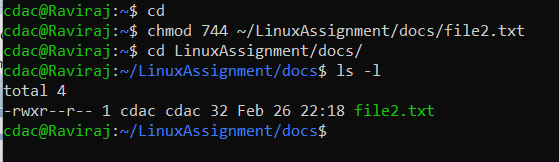
r (read) only

4 is driver from: r-- = 4+0+0 = 4

* **4 (Others)** → The third digit (4) represents **others’** permissions:

r (read) only

4 is driver from: r-- = 4+0+0 = 4

****

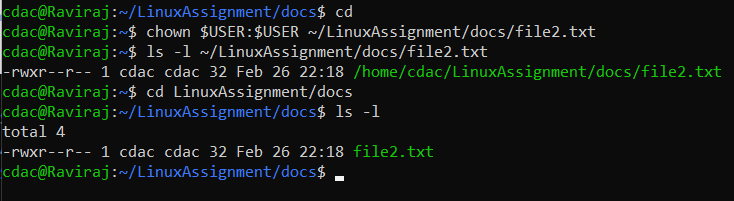
* **Change the owner of "file2.txt" to the current user**

chown $USER:$USER ~/LinuxAssignment/docs/file2.txt

**Explanation of chown $USER:$USER file2.txt**:

* **chown** → Command used to change file ownership
* **$USER** → This automatically substitutes the currently logged-in username.
* **$USER:$USER** →
* The **first $USER** represents the **owner** of the file.
* The **second $USER** represents the **group** to which the owner belongs.
* **~/LinuxAssignment/docs/file2.txt** → Specifies the file whose ownership is being changed.
* **Verify the changes**

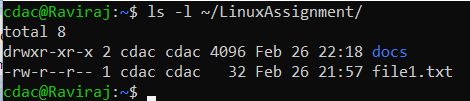
ls -l ~/LinuxAssignment/docs/file2.txt

****

****

* **List contents of "LinuxAssignment" directory**

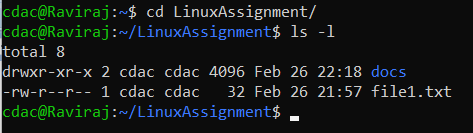
ls -l ~/LinuxAssignment/



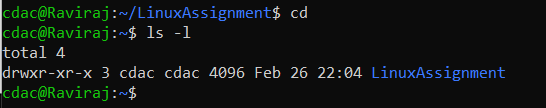
Or

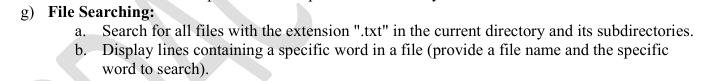
cd LinuxAssignment/

ls –l

****

* **List contents of the root directory**

****

****

* **Search for all .txt files in the current directory and its subdirectories**

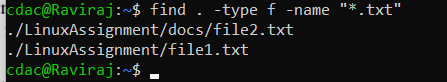
find . -type f -name "\*.txt"

Explanation:

**find .** → Starts searching from the current directory (.).

**-type f** → Looks for files (not directories).

**-name "\*.txt"** → Finds files ending with .txt.

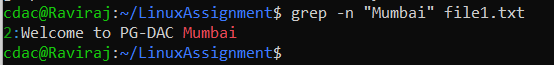
****

* **Display lines containing a specific word in a file**

grep "Mumbai" file1.txt

****

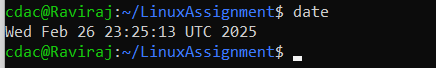
**grep -n “Mumbai” File1.txt**

****

****

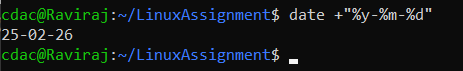
* **Display the current System date and time:**

**date**

****

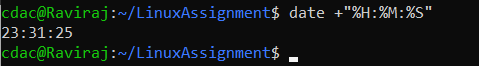
* **Display only date:**

**date +“%y-%m-%d”**

****

* **Display only time:**

**date +“%H:%M:%S”**

****

* **show date in custom format (eg- DD-MM-YYYY)**

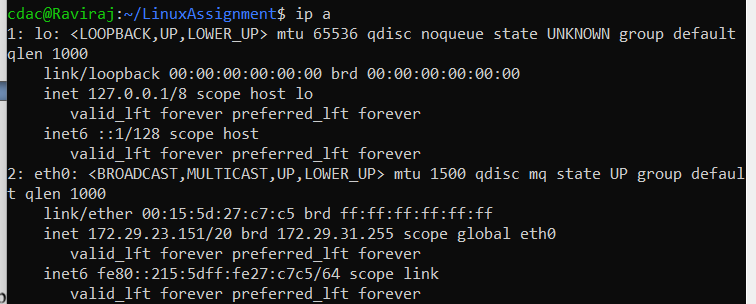
**date +“%d-%m-%y %H:%M:%S”**

****

****

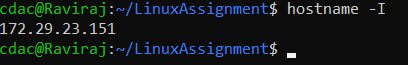
* **Display the IP Address of the System**

ip a



Or

hostname -I

****

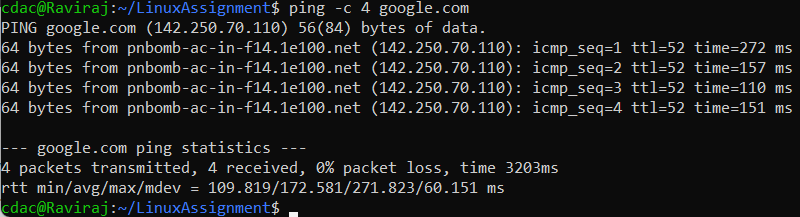
* **Ping a Remote Server to Check Connectivity**

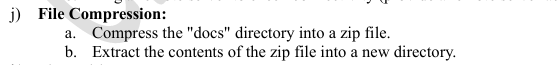
ping -c 4 google.com

**Explanation:**

ping sends packets to the given server to check if it is reachable.

-c 4 sends only 4 packets and stops.

****

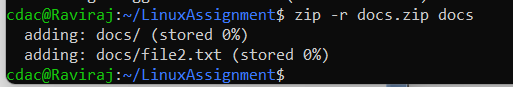
****

* **Compress the "docs" directory into a zip file**

zip -r docs.zip docs

Explanation:

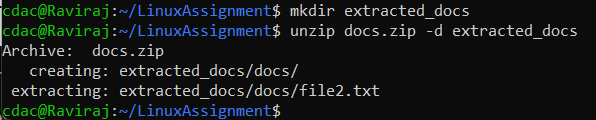
* zip is the command to create a ZIP archive.
* -r (recursive) ensures that all files and subdirectories are included.
* docs.zip is the name of the output ZIP file.
* docs is the directory being compressed.

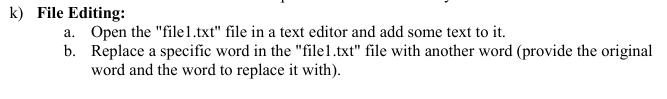
****

* **Extract the contents of the zip file into a new directory**

mkdir extracted\_docs

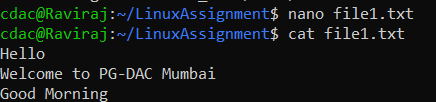
unzip docs.zip -d extracted\_docs

****

****

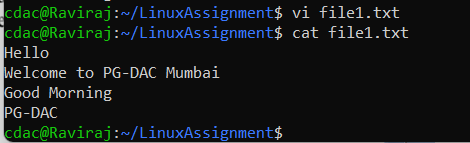
* **Open "file1.txt" in a text editor and add text**
* **using nano**

nano file1.txt

****

* **using vi**

**vi file1txt**

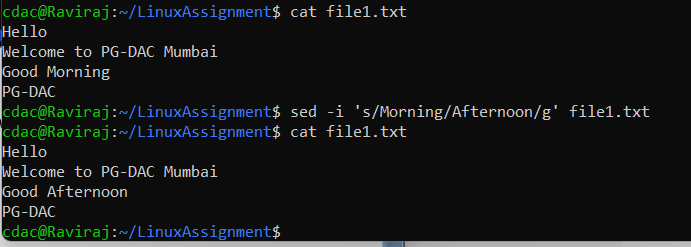
****

* **Replace a specific word in "file1.txt" with another word**

Syntax -

sed -i 's/oldword/newword/g' file1.txt

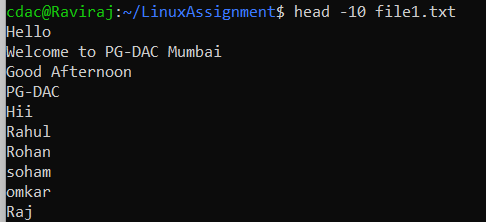
sed -i 's/Morning/Afternoon/g' file1.txt

****

****

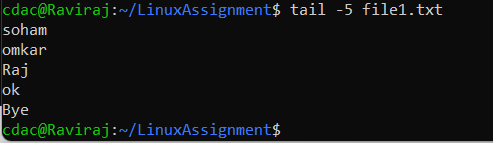
****

* **head -10 file1.txt**

****

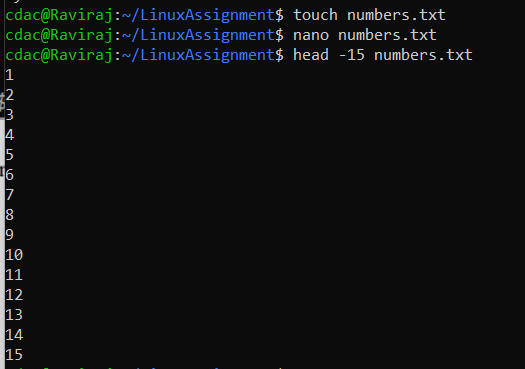
****

* **tail -5 data.txt**

****

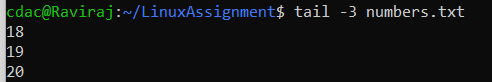
****

* **touch numbers.txt**
* **head -15 numbers.txt**

****

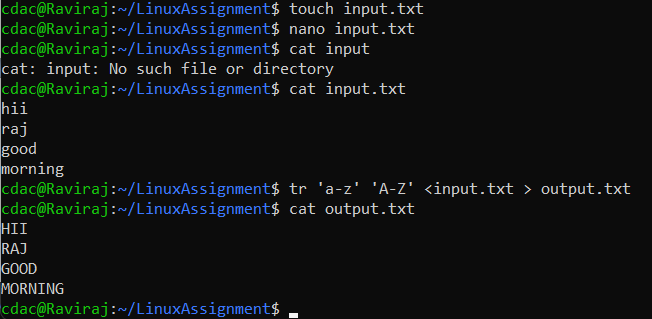
****

* **tail -3 numbers.txt**

****

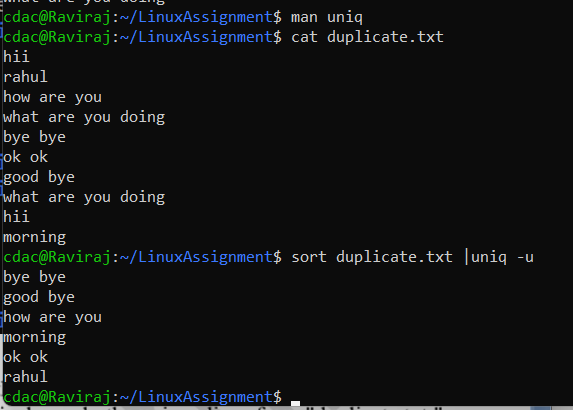
****

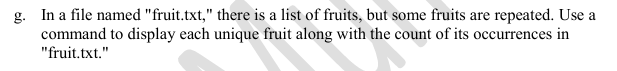
* **touch input.txt**
* **tr 'a-z' 'A-Z' < input.txt > output.txt**

****

****

* **sort duplicate.txt | uniq –u**

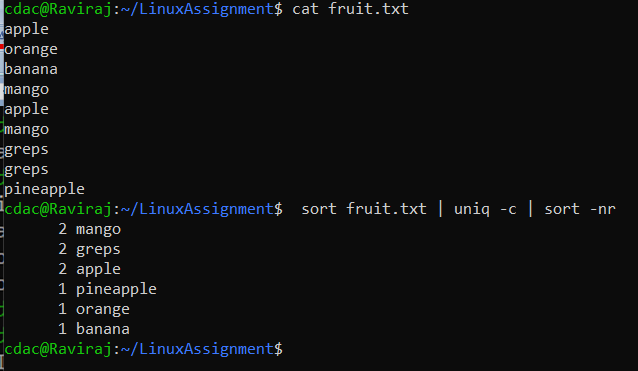
****

****

* **sort fruit.txt | uniq –c | sort –nr**

**Explanation:**

* sort fruit.txt → Sorts the file so that duplicate fruit names are grouped together.
* uniq -c → Counts occurrences of each unique fruit.
* sort -nr → Sorts the output numerically (-n) in descending order (-r), so the most frequent fruit appears first.

****