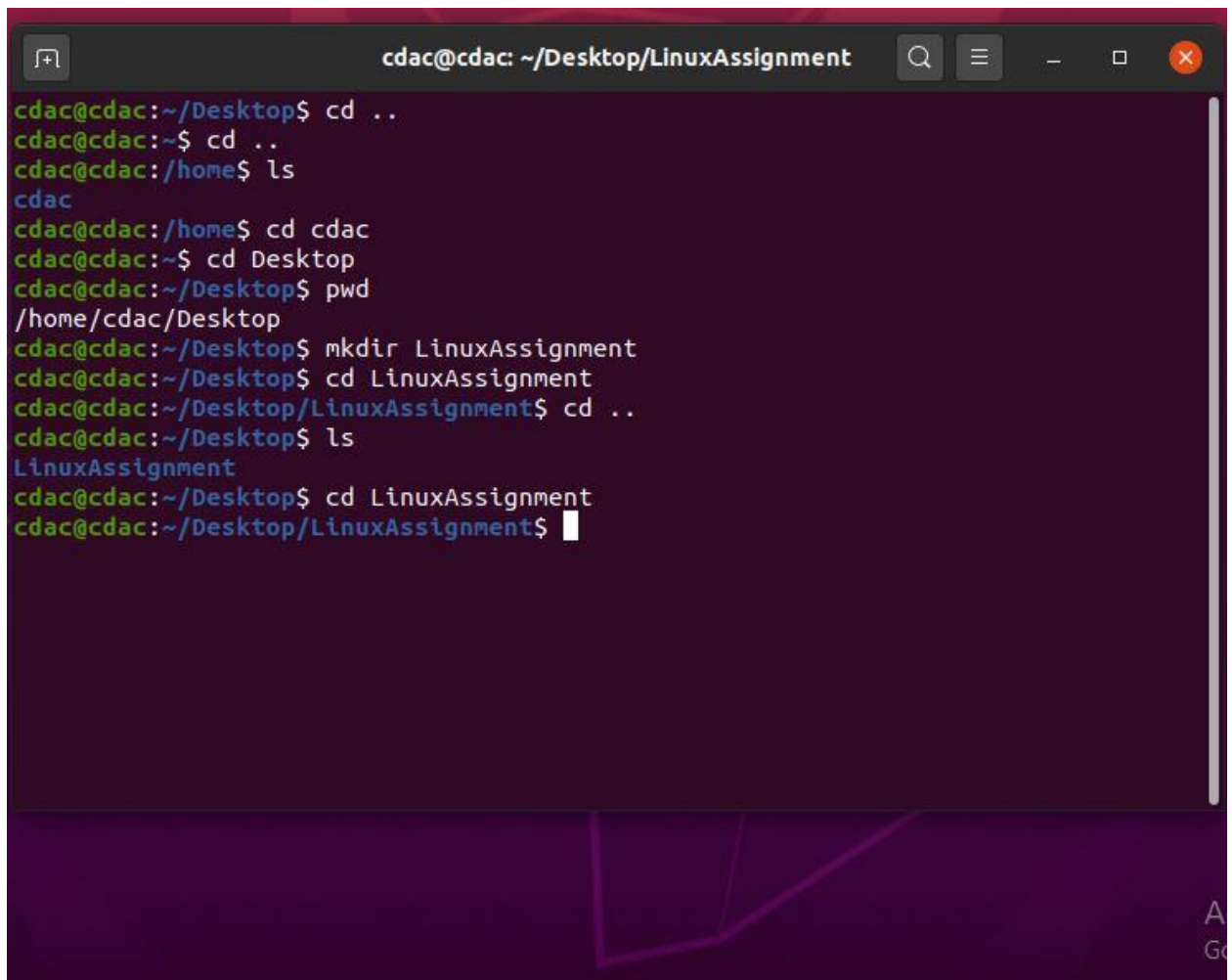# Concepts of Operating System

Assignment 1

Problem 1:

a) Navigate and List: a. Start by navigating to your home directory and list its contents. Then, move into a directory named "LinuxAssignment" if it exists; otherwise, create it.

Explanation:

Step1:

To change directory to home directory, I executed the **cd command** as follows:

Used `..` this as an argument in the `cd` command which is used to move to the parent directory of the current directory, or the directory one level up from the current directory.
Eg: $ **cd ..**
After using "cd .." , I was able to move up from Desktop to home directory.

Step2:

ls – It **list files and directories** in the current directory. So here listed the contents within home directory.

cd - The cd command allows user to **change directories** within the file system. As mentioned above in the image by using cd command , moved from home directory to Desktop. Eg : $ cd cdac ,
$ cd Desktop.

pwd – It prints the path of the current working directory starting from the root. Referring the given image above after using pwd command we got /home/cdac/Desktop
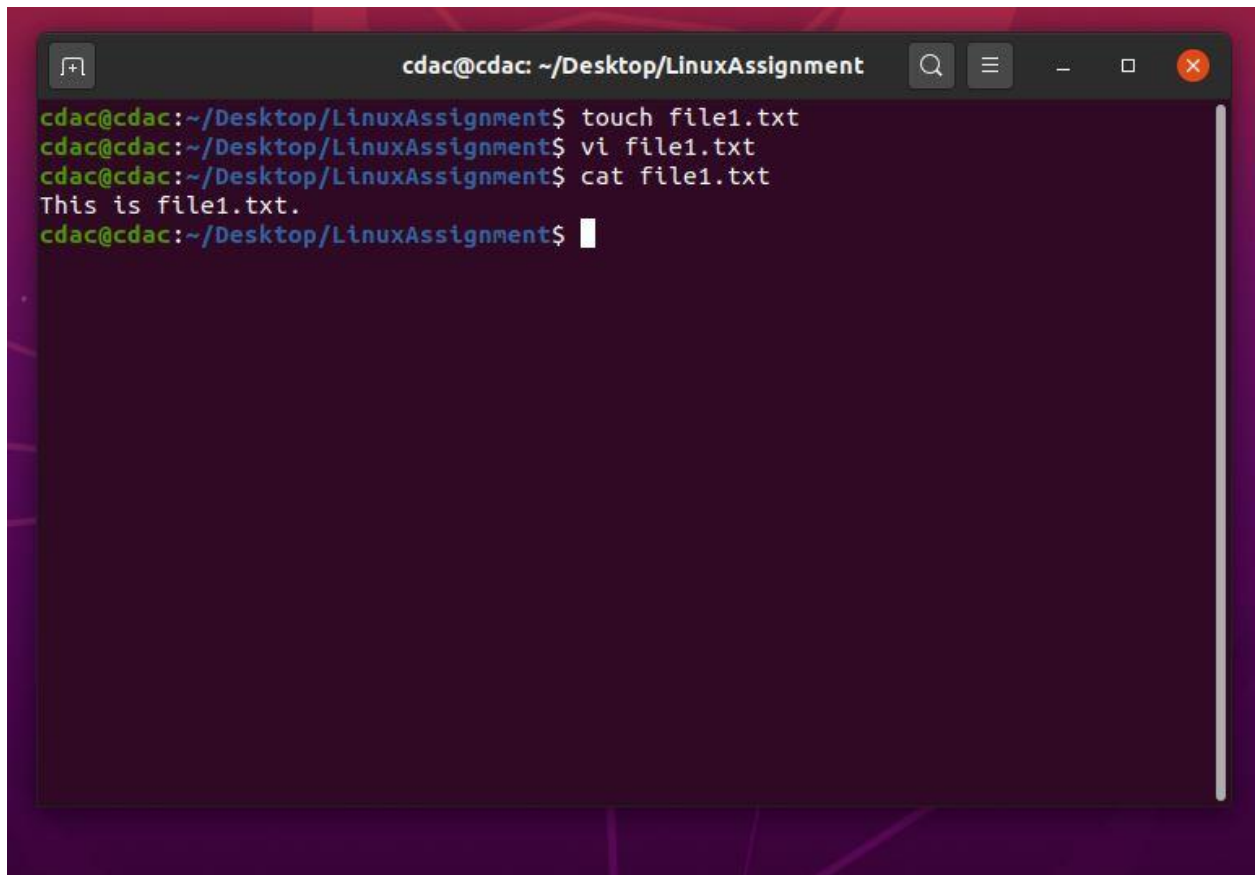
Step3:

mkdir – It is used to create new directories. Created a new directory LinuxAssignment

Eg: $ mkdir LinuxAssignment

Moved into LinuxAssignment using cd command.

Eg: $ cd LinuxAssignment

b) File Management: a. Inside the "LinuxAssignment" directory, create a new file named "file1.txt". Display its contents.

Explanation:

Step1:
touch – It is used to create a file without any content. With the help of touch command I created file1.txt.
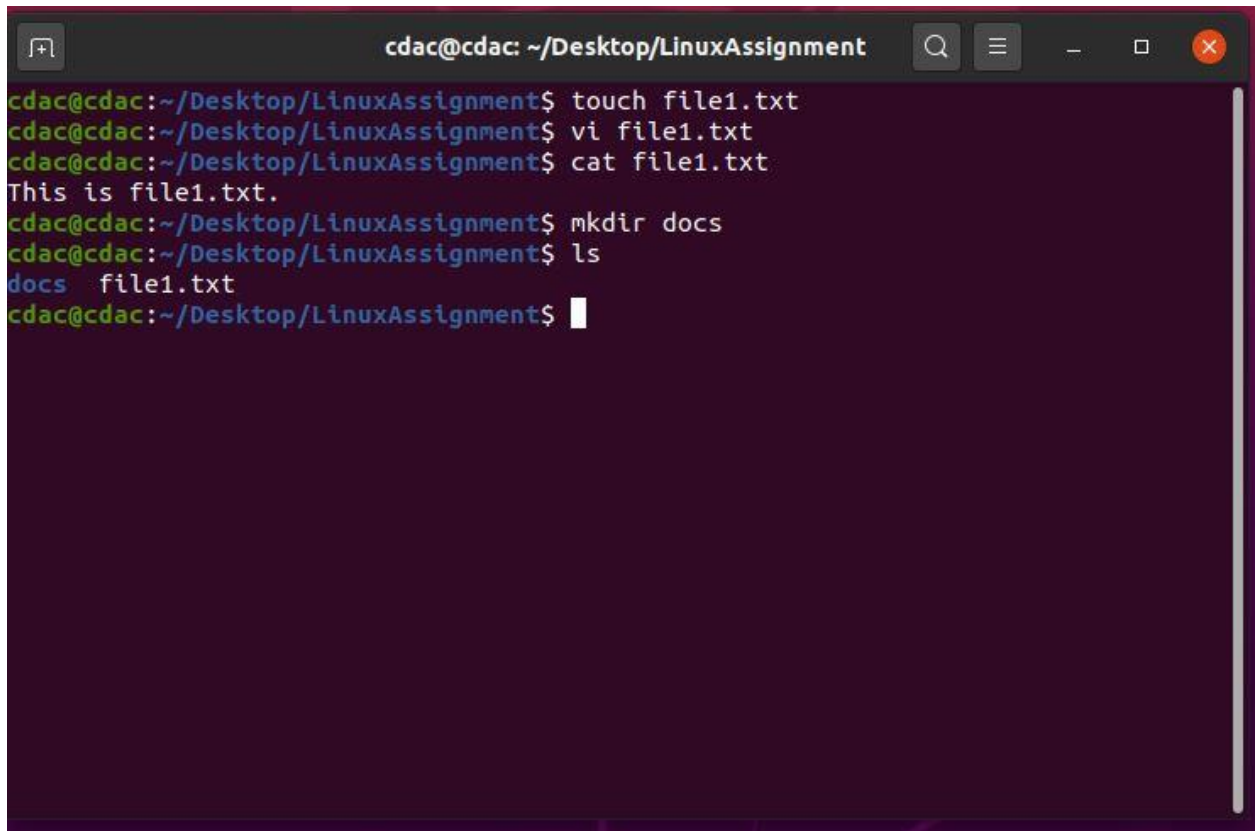
Step2:
vi – Entered the content in file1.txt using "vi command".

Step3:
cat – Using "cat command" , displayed the file contents.

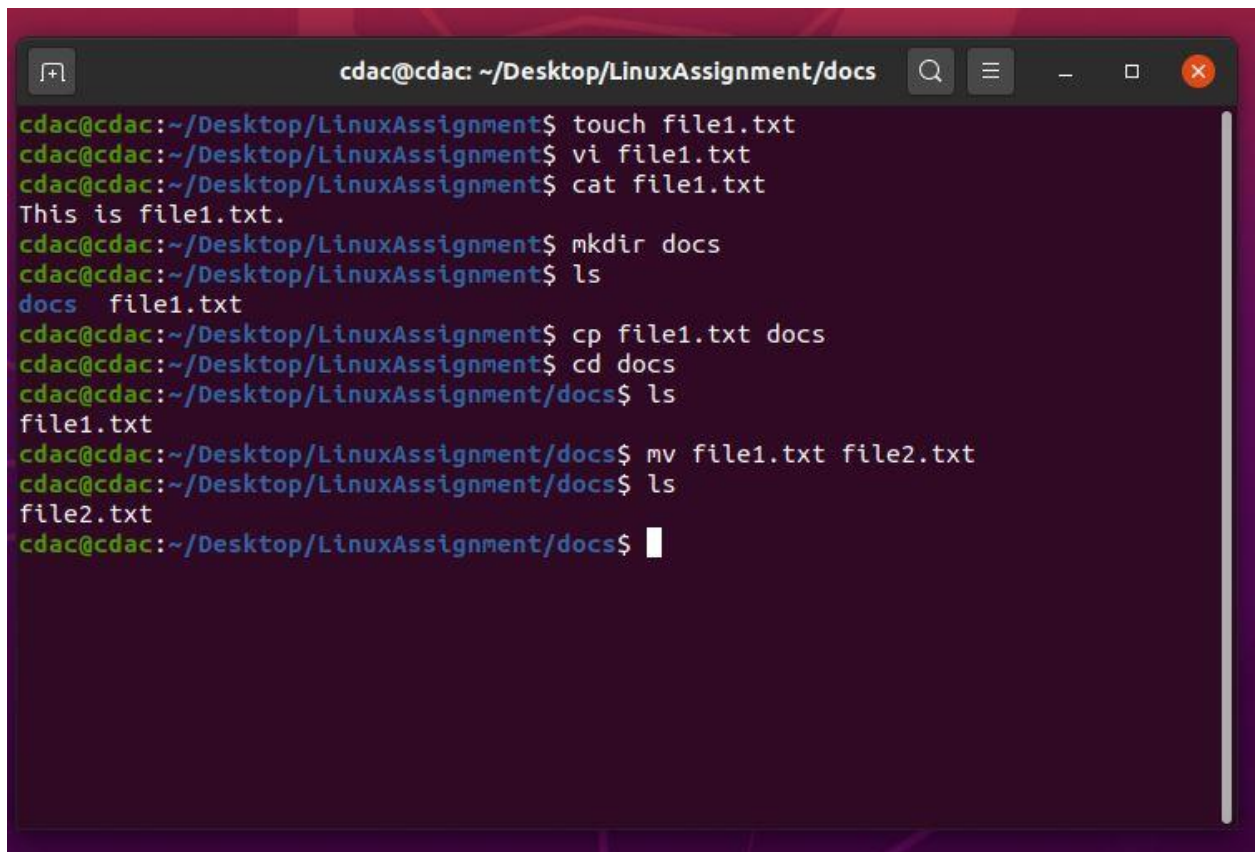c) Directory Management: a. Create a new directory named "docs" inside the "LinuxAssignment" directory.

```
cdac@cdac: ~/Desktop/LinuxAssignment

cdac@cdac:~/Desktop/LinuxAssignment$ touch file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ vi file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ cat file1.txt
This is file1.txt.
cdac@cdac:~/Desktop/LinuxAssignment$ mkdir docs
cdac@cdac:~/Desktop/LinuxAssignment$ ls
docs  file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ 
```

Explanation:

Created a new directory named docs inside LinuxAssignment directory & listed its contents.

d) Copy and Move Files: a. Copy the "file1.txt" file into the "docs" directory and rename it to "file2.txt".

```
cdac@cdac: ~/Desktop/LinuxAssignment/docs

cdac@cdac:~/Desktop/LinuxAssignment$ touch file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ vi file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ cat file1.txt
This is file1.txt.
cdac@cdac:~/Desktop/LinuxAssignment$ mkdir docs
cdac@cdac:~/Desktop/LinuxAssignment$ ls
docs   file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ cp file1.txt docs
cdac@cdac:~/Desktop/LinuxAssignment$ cd docs
cdac@cdac:~/Desktop/LinuxAssignment/docs$ ls
file1.txt
cdac@cdac:~/Desktop/LinuxAssignment/docs$ mv file1.txt file2.txt
cdac@cdac:~/Desktop/LinuxAssignment/docs$ ls
file2.txt
cdac@cdac:~/Desktop/LinuxAssignment/docs$ █
```
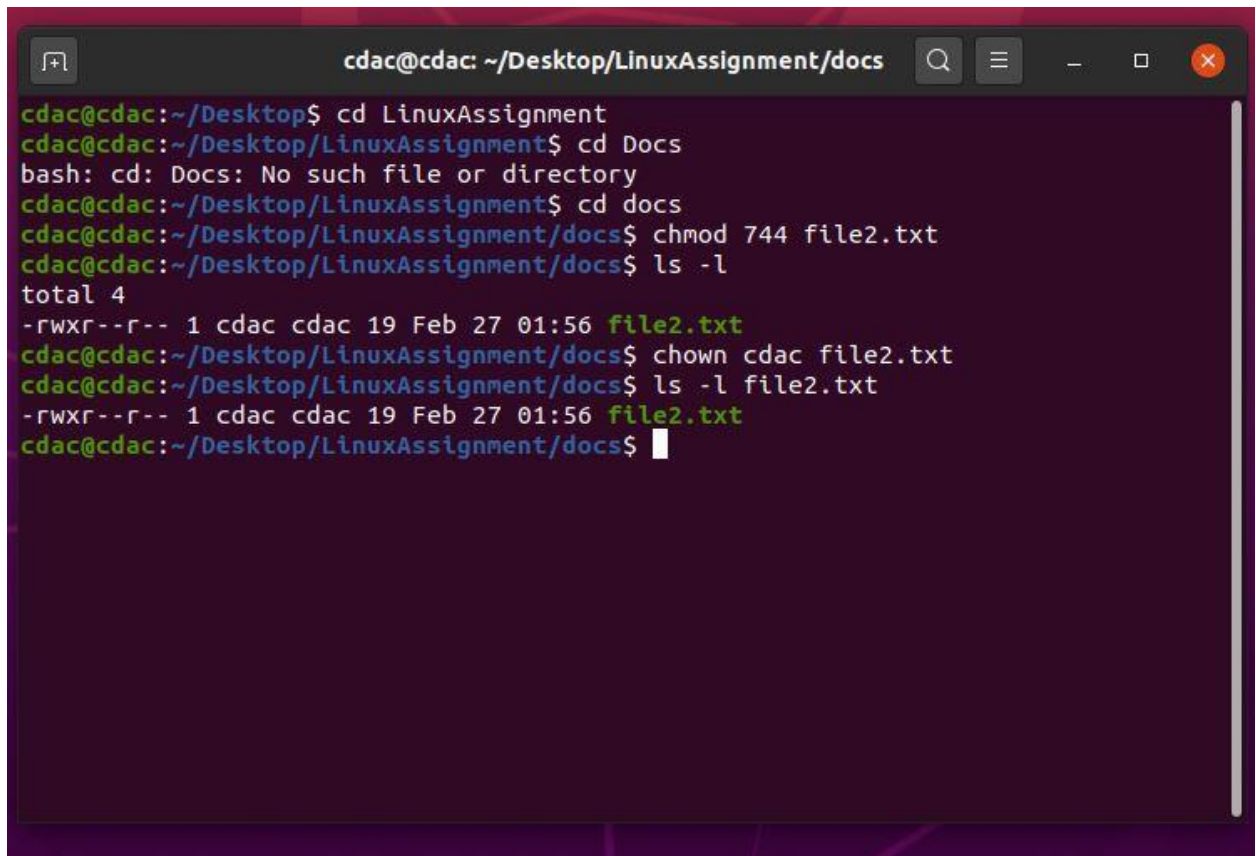
Explanation:

cp – Using cp command I copied "file1.txt" file into docs.
Eg: $ cp file1.txt docs

mv – Using mv command I renamed "file1.txt" to "file2.txt" inside docs directory.

Eg: $ mv file1.txt file2.txt

e) Permissions and Ownership: a. Change the permissions of "file2.txt" to allow read, write, and execute permissions for the owner and only read permissions for others. Then, change the owner of "file2.txt" to the current user.

Explanation:

Step1:

chmod: Every file & directory has a set of permissions that control the permissions like who can read, write & execute the file. Using chmod command I changed the file permissions of file2.txt allowing only the owner to read , write & execute the file, whereas restricting permissions for "groups" & "others" to only read the file without writing & executing.
Did this using octal mode.

Eg: $ chmod 744 file2.txt

Step2:

chown: Using this command I changed the ownership of "file2.txt" to the current user "cdac" as mentioned in the above image.

f) Final Checklist: a. Finally, list the contents of the "LinuxAssignment" directory and the root directory to ensure that all operations were performed correctly.

```
cdac@cdac: ~/Desktop/LinuxAssignment

cdac@cdac:~/Desktop/LinuxAssignment$ ls -l
total 8
drwxrwxr-x 2 cdac cdac 4096 Feb 27 01:59 docs
-rw-rw-r-- 1 cdac cdac   19 Feb 27 01:25 file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ ls -l /
total 2097232
lrwxrwxrwx   1 root root        7 Feb 26 00:41 bin -> usr/bin
drwxr-xr-x   4 root root     4096 Feb 26 01:16 boot
drwxrwxr-x   2 root root     4096 Feb 26 00:47 cdrom
drwxr-xr-x  18 root root     4140 Feb 27 09:54 dev
drwxr-xr-x 131 root root    12288 Feb 26 01:17 etc
drwxr-xr-x   3 root root     4096 Feb 26 00:49 home
lrwxrwxrwx   1 root root        7 Feb 26 00:41 lib -> usr/lib
lrwxrwxrwx   1 root root        9 Feb 26 00:41 lib32 -> usr/lib32
lrwxrwxrwx   1 root root        9 Feb 26 00:41 lib64 -> usr/lib64
lrwxrwxrwx   1 root root       10 Feb 26 00:41 libx32 -> usr/libx32
drwx------   2 root root    16384 Feb 26 00:40 lost+found
drwxr-xr-x   3 root root     4096 Feb 26 01:14 media
drwxr-xr-x   2 root root     4096 Jul 31  2020 mnt
drwxr-xr-x   3 root root     4096 Feb 26 01:14 opt
dr-xr-xr-x 250 root root        0 Feb 27 09:54 proc
drwx------   4 root root     4096 Feb 26 23:17 root
drwxr-xr-x  32 root root      880 Feb 27 12:27 run
lrwxrwxrwx   1 root root        8 Feb 26 00:41 sbin -> usr/sbin
```
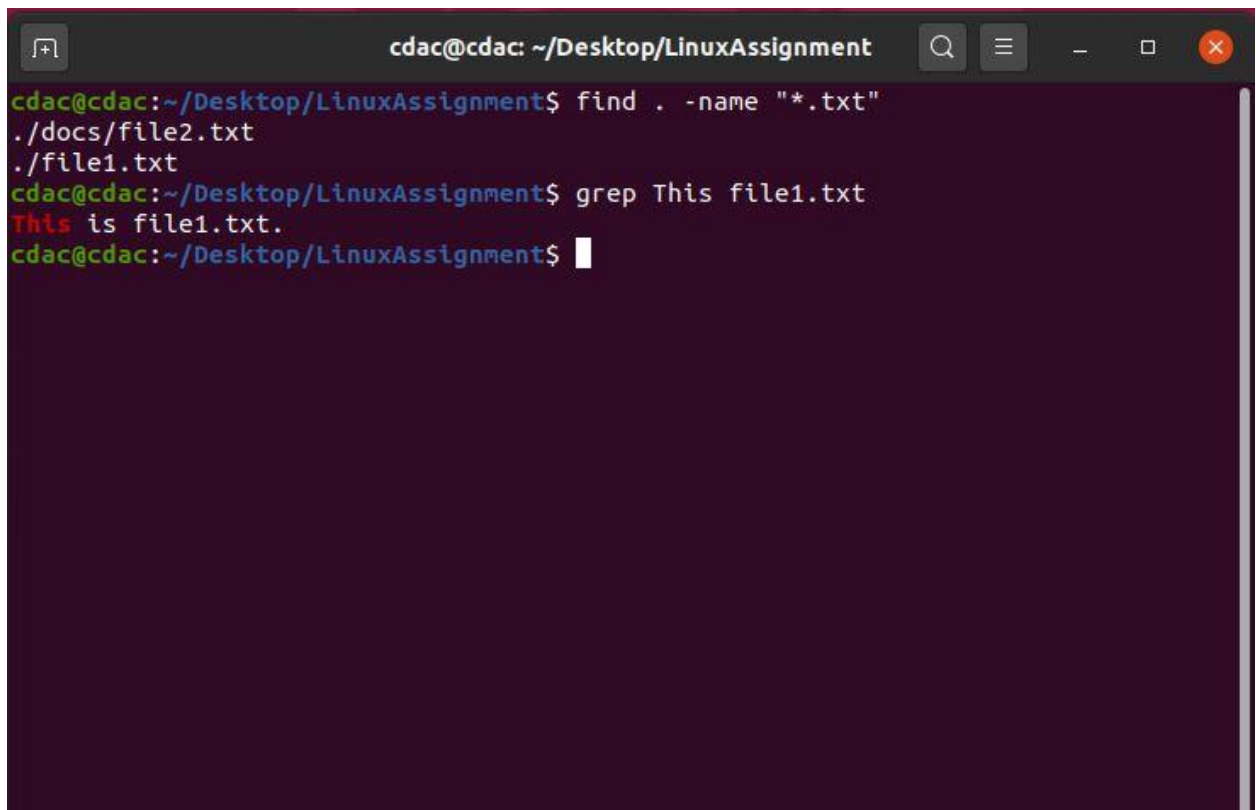
g) File Searching: a. Search for all files with the extension ".txt" in the current directory and its subdirectories. b. Display lines containing a specific word in a file (provide a file name and the specific word to search).

Explanation:

To search for all files with the ".txt" extension in the current directory and its subdirectories used the following command: $ find . -name "*.txt".

To search for files containing a specific text or string I used grep command : $ grep This file1.txt

h) System Information: a. Display the current system date and time.

i) Networking: a. Display the IP address of the system. b. Ping a remote server to check connectivity (provide a remote server address to ping).


```
cdac@cdac: ~/Desktop/LinuxAssignment
cdac@cdac:~/Desktop/LinuxAssignment$ ip addr show | grep "inet"
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
    inet6 fd00::ea25:40b0:3c3:554b/64 scope global temporary dynamic
    inet6 fd00::b0f:cb10:a314:99a9/64 scope global dynamic mngtmpaddr noprefixro
ute
    inet6 fe80::d6bf:e000:6e4e:32a8/64 scope link noprefixroute
cdac@cdac:~/Desktop/LinuxAssignment$
```

Explanation : Displayed address using ip addr command & used grep with "inet" to get the address.


```
cdac@cdac: ~/Desktop/LinuxAssignment
cdac@cdac:~/Desktop/LinuxAssignment$ ping google.com
PING google.com (142.250.194.142) 56(84) bytes of data.
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=1 ttl=255 time=35.1 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=2 ttl=255 time=36.6 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=3 ttl=255 time=34.8 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=4 ttl=255 time=37.2 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=5 ttl=255 time=35.7 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=6 ttl=255 time=37.8 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=7 ttl=255 time=35.4 ms
64 bytes from del12s05-in-f14.1e100.net (142.250.194.142): icmp_seq=8 ttl=255 time=35.5 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7028ms
rtt min/avg/max/mdev = 34.768/36.023/37.774/0.999 ms
cdac@cdac:~/Desktop/LinuxAssignment$
```

j) File Compression: a. Compress the "docs" directory into a zip file. b. Extract the contents of the zip file into a new directory.
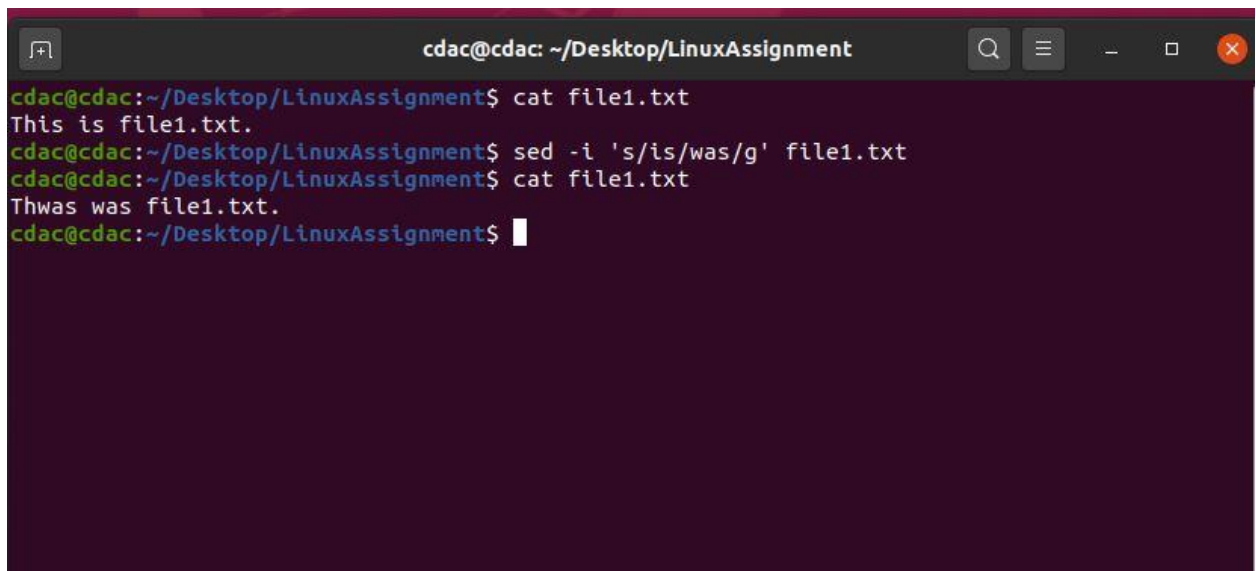
```
                         cdac@cdac: ~/Desktop/LinuxAssignment          Q   ≡   —   □   ✕
cdac@cdac:~/Desktop/LinuxAssignment$ zip docs.zip docs/
  adding: docs/ (stored 0%)
cdac@cdac:~/Desktop/LinuxAssignment$ ls -l
total 12
drwxrwxr-x 2 cdac cdac 4096 Feb 27 01:59 docs
-rw-rw-r-- 1 cdac cdac  160 Feb 27 23:27 docs.zip
-rw-rw-r-- 1 cdac cdac   19 Feb 27 01:25 file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ unzip docs.zip -d doc1
Archive:  docs.zip
   creating: doc1/docs/
cdac@cdac:~/Desktop/LinuxAssignment$ ls -l
total 16
drwxrwxr-x 3 cdac cdac 4096 Feb 27 23:30 doc1
drwxrwxr-x 2 cdac cdac 4096 Feb 27 01:59 docs
-rw-rw-r-- 1 cdac cdac  160 Feb 27 23:27 docs.zip
-rw-rw-r-- 1 cdac cdac   19 Feb 27 01:25 file1.txt
cdac@cdac:~/Desktop/LinuxAssignment$ █
```

Explanation:

To compress the docs directory into a Zip used : $ zip docs.zip docs/ & to
extract the contents of a Zip into a new directory I used : $ unzip docs.zip -d
doc1

Where doc1=The directory where the files will be extracted.

k) File Editing: a. Open the "file1.txt" file in a text editor and add some text
to it. b. Replace a specific word in the "file1.txt" file with another word
(provide the original word and the word to replace it with).

Explanation:

Using sed command with option –i edited the file with the help of 's/is(oldword)/was(newword)/g' (globally) across the file.
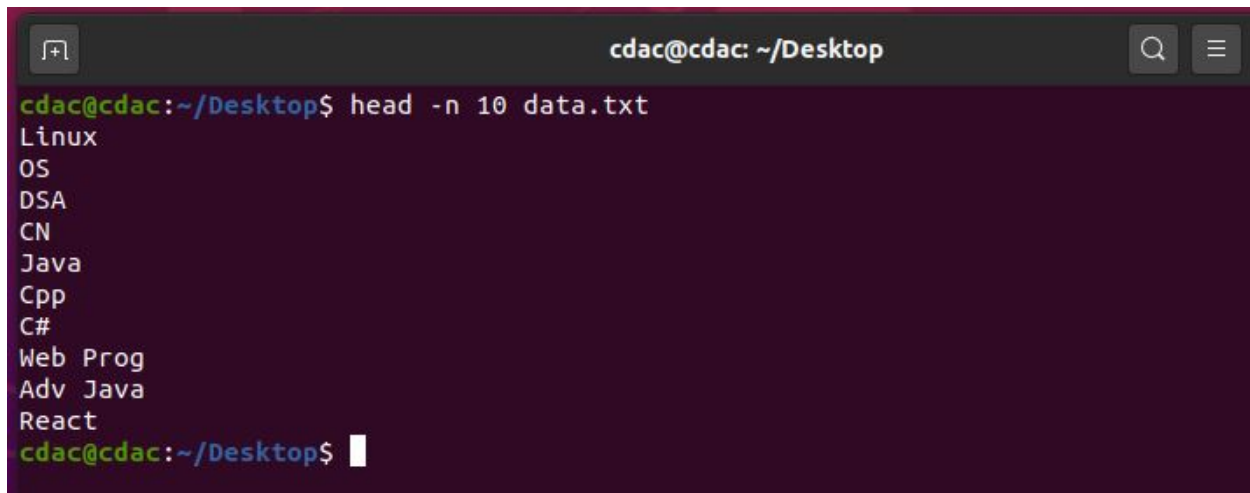
$ cat file1.txt

O/p: This is file1.txt.

$ sed -i 's/is/was/g' file1.txt

$ cat file1.txt

O/P: Thwas was file1.txt.

Problem 2:

a. Suppose you have a file named "data.txt" containing important information. Display the first 10 lines of this file to quickly glance at its contents using a command.
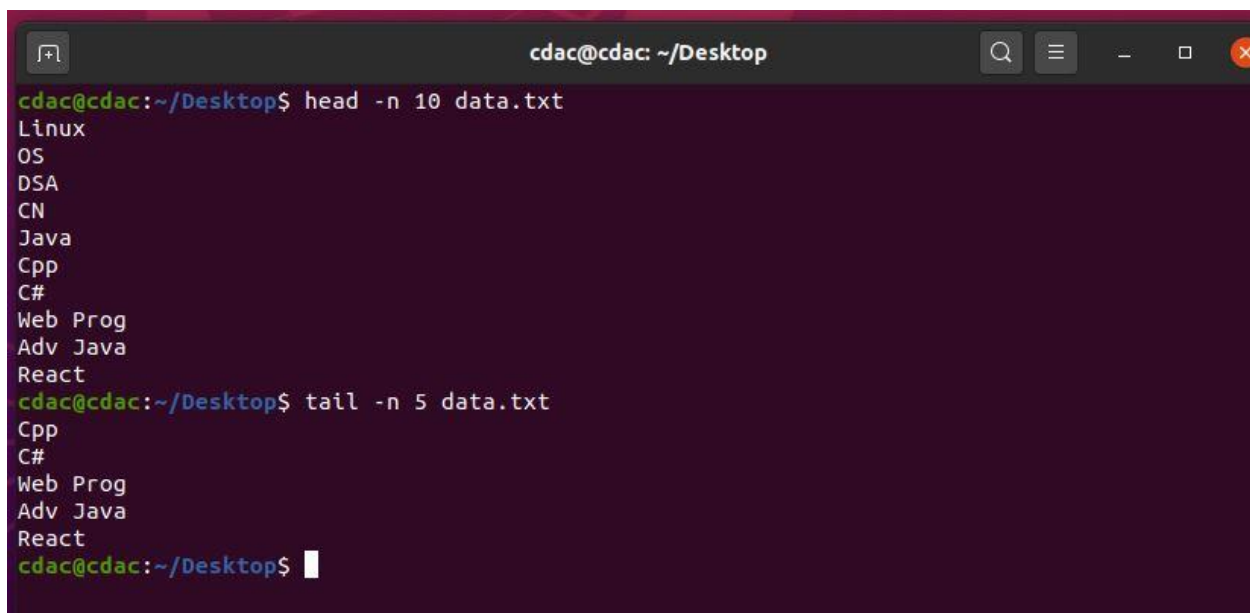
Explanation:

Using head & option –n 10 displayed the first 10 lines of data.txt.

b. Now, to check the end of the file for any recent additions, display the last 5 lines of "data.txt" using another command.
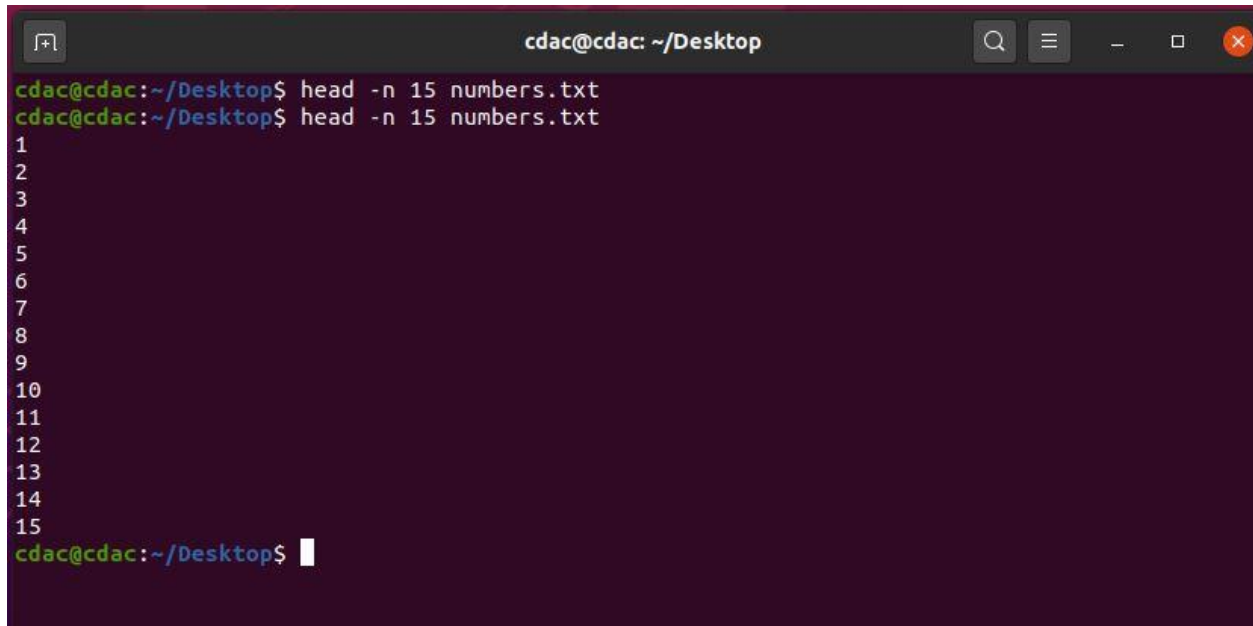
Explanation:

Using tail & option –n 5 displayed the last 5 lines of data.txt.

c. In a file named "numbers.txt," there are a series of numbers. Display the first 15 lines of this file to analyze the initial data set.
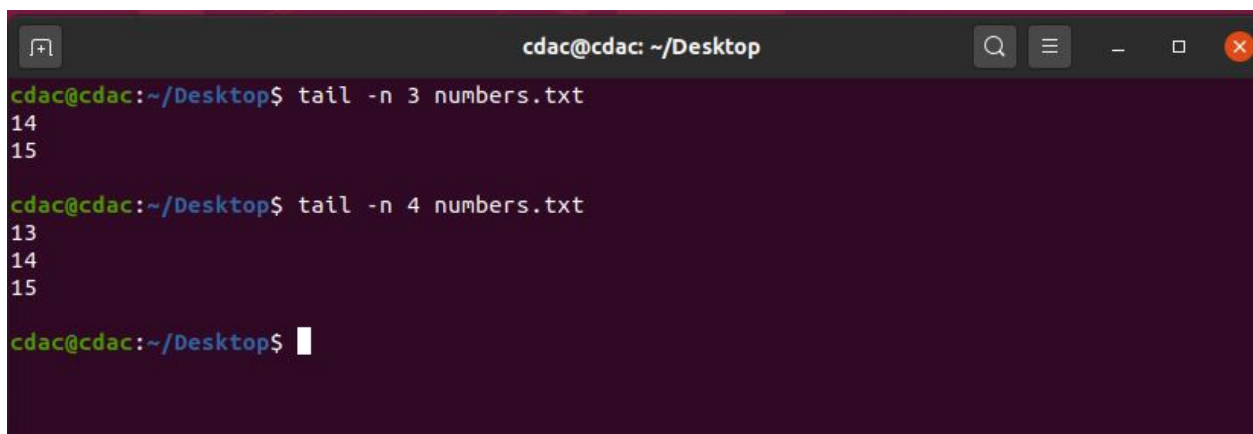
Solution:

```
cdac@cdac: ~/Desktop

cdac@cdac:~/Desktop$ head -n 15 numbers.txt
cdac@cdac:~/Desktop$ head -n 15 numbers.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
cdac@cdac:~/Desktop$
```

d. To focus on the last few numbers of the dataset, display the last 3 lines of "numbers.txt".
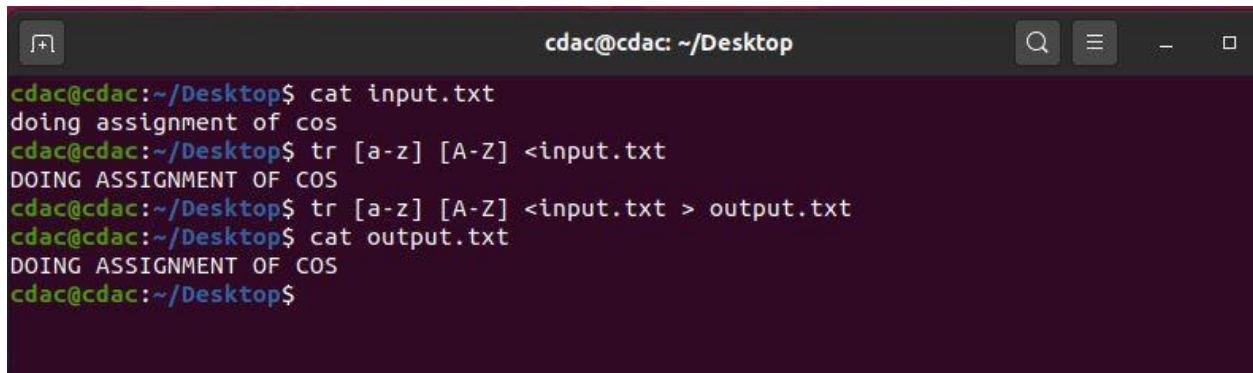
Solution:

```
cdac@cdac: ~/Desktop

cdac@cdac:~/Desktop$ tail -n 3 numbers.txt
14
15

cdac@cdac:~/Desktop$ tail -n 4 numbers.txt
13
14
15

cdac@cdac:~/Desktop$
```

e. Imagine you have a file named "input.txt" with text content. Use a command to translate all lowercase letters to uppercase in "input.txt" and save the modified text in a new file named "output.txt."
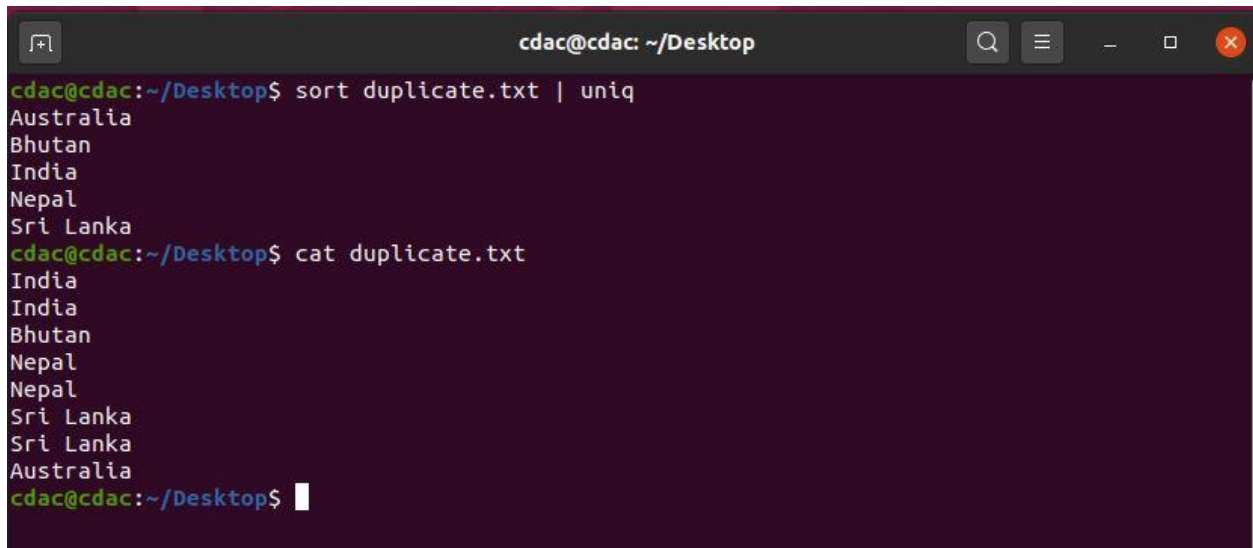
```
cdac@cdac: ~/Desktop                    Q  ≡   _   □
cdac@cdac:~/Desktop$ cat input.txt
doing assignment of cos
cdac@cdac:~/Desktop$ tr [a-z] [A-Z] <input.txt
DOING ASSIGNMENT OF COS
cdac@cdac:~/Desktop$ tr [a-z] [A-Z] <input.txt > output.txt
cdac@cdac:~/Desktop$ cat output.txt
DOING ASSIGNMENT OF COS
cdac@cdac:~/Desktop$
```

Explanation:

To convert all lowercase letters to uppercase in input.txt I used tr command with [a-z] [A-Z] taking input from input.txt using Redirection. And later transferred the modified text to the output.txt file.

f. In a file named "duplicate.txt," there are several lines of text, some of which are duplicates. Use a command to display only the unique lines from "duplicate.txt."
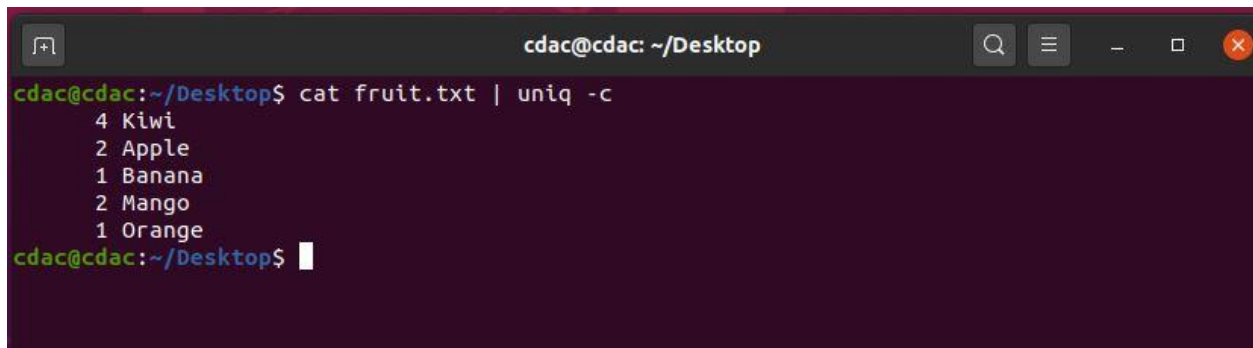
```
cdac@cdac: ~/Desktop                    Q  ≡   _   □   ✕
cdac@cdac:~/Desktop$ sort duplicate.txt | uniq
Australia
Bhutan
India
Nepal
Sri Lanka
cdac@cdac:~/Desktop$ cat duplicate.txt
India
India
Bhutan
Nepal
Nepal
Sri Lanka
Sri Lanka
Australia
cdac@cdac:~/Desktop$
```

Explanation:

With the help of uniq command we are able to filter out duplicate lines as seen above. Now to display file contents I only used cat without uniq which ended up with all duplicate lines.

g. In a file named "fruit.txt," there is a list of fruits, but some fruits are repeated. Use a command to display each unique fruit along with the count of its occurrences in "fruit.txt."

```
cdac@cdac:~/Desktop$ cat fruit.txt | uniq -c
      4 Kiwi
      2 Apple
      1 Banana
      2 Mango
      1 Orange
cdac@cdac:~/Desktop$
```

Explanation:

With the help of –c we're able to count the occurrences of each fruit & print the count alongside the fruit name. Avoided duplication with the help of uniq.