

CDAC MUMBAI

Concepts of Operating System

Assignment 1

Problem 1: Read the instructions carefully and answer accordingly. If there is any need to insert some data then do that as well.

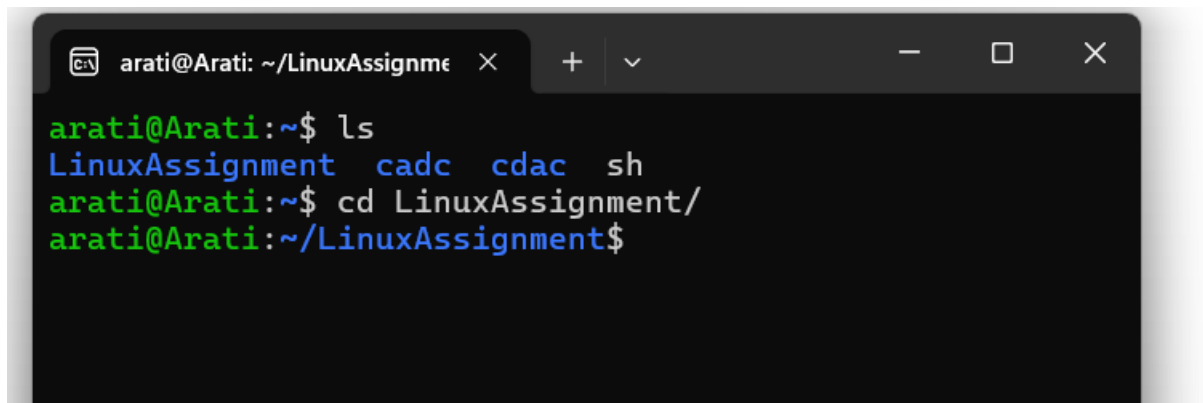
a) Navigate and List:

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.

Answer : -

```
arati@Arati:~$ ls
cadc cdac sh
arati@Arati:~$ mkdir LinuxAssignment
arati@Arati:~$ cd LinuxAssignment/
arati@Arati:~/LinuxAssignment$
```

output:

A screenshot of a terminal window with a dark background. The window title bar shows 'arati@Arati: ~/LinuxAssignme' and standard window controls. The terminal displays the following commands and their outputs: 'arati@Arati:~\$ ls' followed by 'LinuxAssignment cadc cdac sh' on the next line; 'arati@Arati:~\$ cd LinuxAssignment/' followed by 'arati@Arati:~/LinuxAssignment\$' on the next line. The text is color-coded: the prompt is green, the command is blue, and the output is white.

b) File Management:

a. Inside the "LinuxAssignment" directory, create a new file named "file1.txt". Display its contents.

Answer :

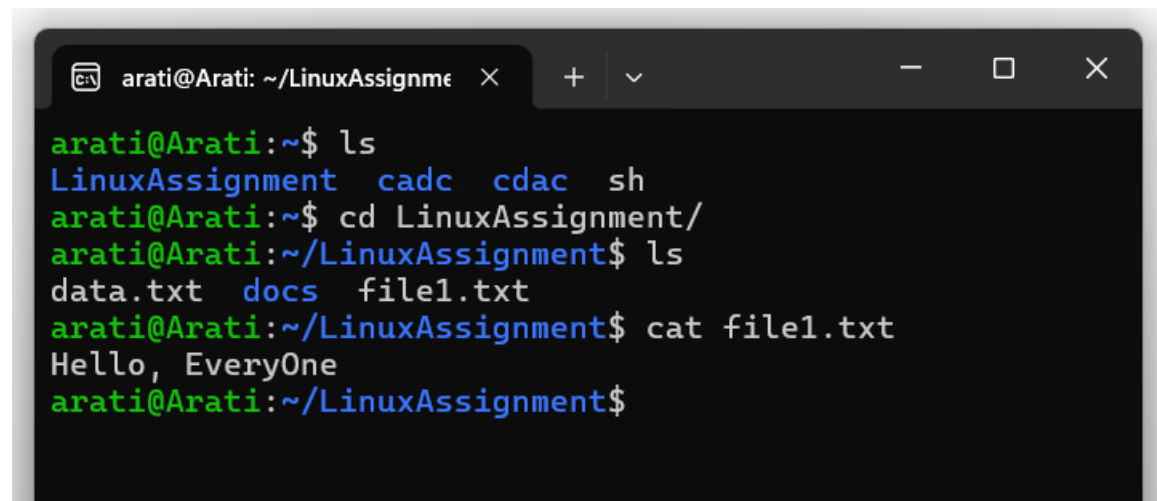
```
arati@Arati:~/LinuxAssignment$ touch file1.txt
```

```
arati@Arati:~/LinuxAssignment$ nano file1.txt
```

```
arati@Arati:~/LinuxAssignment$ cat file1.txt
```

Hello, EveryOne

Output:

A terminal window titled 'arati@Arati: ~/LinuxAssignme' with standard window controls. The terminal shows the following commands and output: 'ls' lists 'LinuxAssignment', 'cadc', 'cdac', and 'sh'; 'cd LinuxAssignment/' changes the directory; 'ls' lists 'data.txt', 'docs', and 'file1.txt'; 'cat file1.txt' outputs 'Hello, EveryOne'.

```
arati@Arati:~$ ls
LinuxAssignment  cadc  cdac  sh
arati@Arati:~$ cd LinuxAssignment/
arati@Arati:~/LinuxAssignment$ ls
data.txt  docs  file1.txt
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, EveryOne
arati@Arati:~/LinuxAssignment$
```

c) Directory Management:

a. Create a new directory named "docs" inside the "LinuxAssignment" directory.

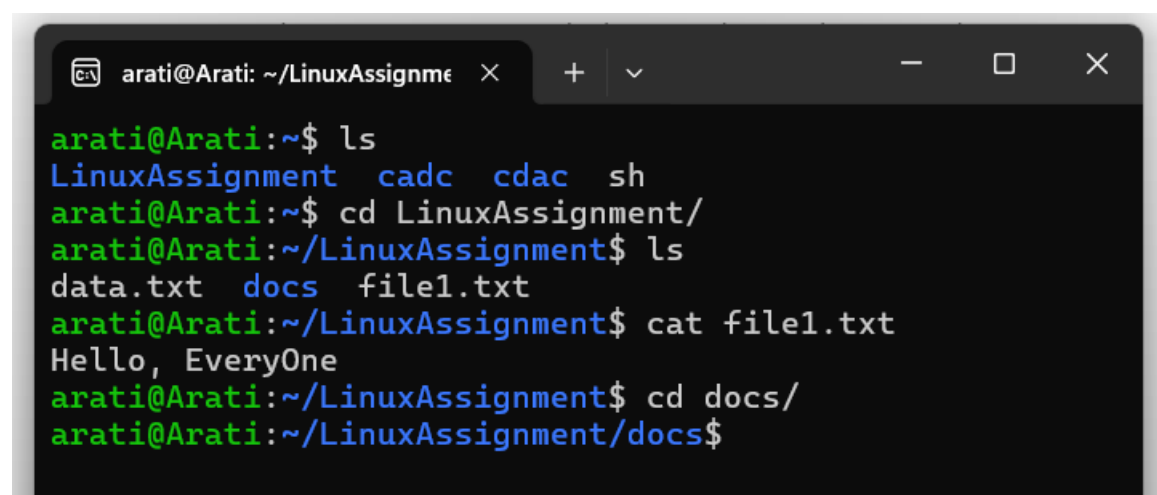
Answer :

```
arati@Arati:~/LinuxAssignment$ mkdir docs
```

```
arati@Arati:~/LinuxAssignment$ cd docs
```

```
arati@Arati:~/LinuxAssignment/docs$
```

Output:

A terminal window titled 'arati@Arati: ~/LinuxAssignme' with standard window controls. The terminal shows the following commands and output: 'ls' lists 'LinuxAssignment', 'cadc', 'cdac', and 'sh'; 'cd LinuxAssignment/' changes the directory; 'ls' lists 'data.txt', 'docs', and 'file1.txt'; 'cat file1.txt' outputs 'Hello, EveryOne'; 'cd docs/' changes to the 'docs' subdirectory. The final prompt is 'arati@Arati:~/LinuxAssignment/docs\$'.

```
arati@Arati:~$ ls
LinuxAssignment  cadc  cdac  sh
arati@Arati:~$ cd LinuxAssignment/
arati@Arati:~/LinuxAssignment$ ls
data.txt  docs  file1.txt
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, EveryOne
arati@Arati:~/LinuxAssignment$ cd docs/
arati@Arati:~/LinuxAssignment/docs$
```

d) Copy and Move Files:

a. Copy the "file1.txt" file into the "docs" directory and rename it to "file2.txt".

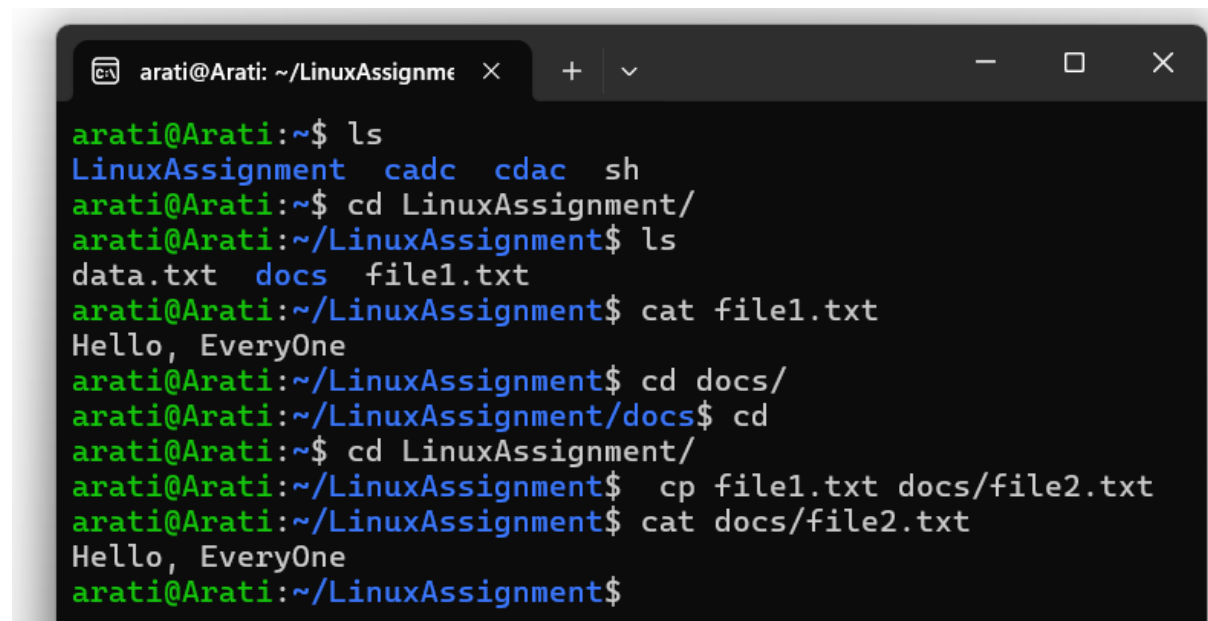
Answer :

```
arati@Arati:~/LinuxAssignment$ cp file1.txt docs/file2.txt
```

```
arati@Arati:~/LinuxAssignment$ cat docs/file2.txt
```

Hello, EveryOne

Output:

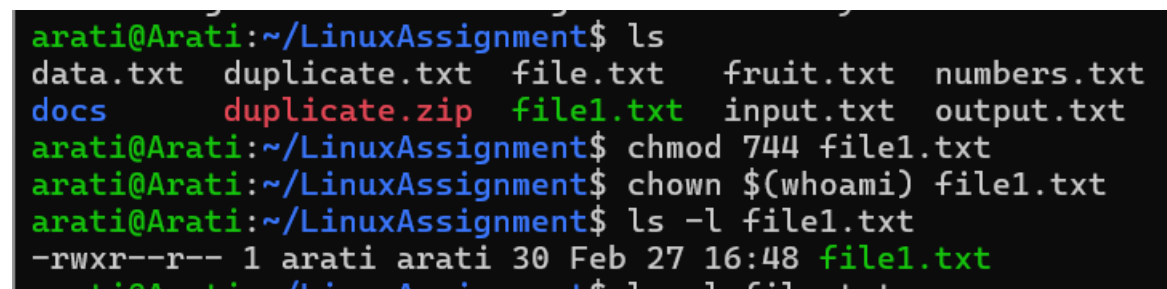
A terminal window with a dark background and light-colored text. The window title is 'arati@Arati: ~/LinuxAssignme'. The terminal shows a series of commands and their outputs: 'ls' lists 'LinuxAssignment', 'cadc', 'cdac', and 'sh'; 'cd LinuxAssignment/' changes the directory; 'ls' lists 'data.txt', 'docs', and 'file1.txt'; 'cat file1.txt' outputs 'Hello, EveryOne'; 'cd docs/' changes to the docs directory; 'cd' shows the current directory as '/LinuxAssignment/docs'; 'cd LinuxAssignment/' returns to the parent directory; 'cp file1.txt docs/file2.txt' copies the file; 'cat docs/file2.txt' outputs 'Hello, EveryOne'; and the prompt returns to '~/.LinuxAssignment\$'.

```
arati@Arati:~$ ls
LinuxAssignment  cadc  cdac  sh
arati@Arati:~$ cd LinuxAssignment/
arati@Arati:~/LinuxAssignment$ ls
data.txt  docs  file1.txt
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, EveryOne
arati@Arati:~/LinuxAssignment$ cd docs/
arati@Arati:~/LinuxAssignment/docs$ cd
arati@Arati:~$ cd LinuxAssignment/
arati@Arati:~/LinuxAssignment$ cp file1.txt docs/file2.txt
arati@Arati:~/LinuxAssignment$ cat docs/file2.txt
Hello, EveryOne
arati@Arati:~/LinuxAssignment$
```

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.

Answer:

A terminal window with a dark background and light-colored text. The window title is 'arati@Arati: ~/LinuxAssignment\$'. The terminal shows commands and their outputs: 'ls' lists various files including 'duplicate.txt', 'file.txt', 'fruit.txt', 'numbers.txt', 'docs', 'duplicate.zip', 'file1.txt', 'input.txt', and 'output.txt'; 'chmod 744 file1.txt' changes permissions; 'chown \$(whoami) file1.txt' changes ownership to the current user; and 'ls -l file1.txt' shows the resulting permissions '-rw-r--r--' and owner 'arati'.

```
arati@Arati:~/LinuxAssignment$ ls
data.txt  duplicate.txt  file.txt  fruit.txt  numbers.txt
docs      duplicate.zip  file1.txt  input.txt  output.txt
arati@Arati:~/LinuxAssignment$ chmod 744 file1.txt
arati@Arati:~/LinuxAssignment$ chown $(whoami) file1.txt
arati@Arati:~/LinuxAssignment$ ls -l file1.txt
-rwxr--r-- 1 arati arati 30 Feb 27 16:48 file1.txt
arati@Arati:~/LinuxAssignment$
```

f) Final Checklist:

- a. Finally, list the contents of the "LinuxAssignment" directory and the root directory to ensure that all operations were performed correctly.

Output:

```
arati@Arati: ~/LinuxAssignme x + v - □ X
arati@Arati:~/LinuxAssignment$ ls -l
total 32
-rw-r--r-- 1 arati arati 1663 Feb 27 13:18 data.txt
drwxr-xr-x 2 arati arati 4096 Feb 26 17:05 docs
-rw-r--r-- 1 arati arati 141 Feb 27 13:50 duplicate.txt
-rwxr--r-- 1 arati arati 16 Feb 26 17:00 file1.txt
-rw-r--r-- 1 arati arati 152 Feb 27 13:58 fruit.txt
-rw-r--r-- 1 arati arati 22 Feb 27 13:43 input.txt
-rw-r--r-- 1 arati arati 1663 Feb 27 13:31 numbers.txt
-rw-r--r-- 1 arati arati 22 Feb 27 13:44 output.txt
```

g) File Searching:

- a. Search for all files with the extension ".txt" in the current directory and its subdirectories.

Output:

```
arati@Arati: ~/LinuxAssignme x + v - □ X
arati@Arati:~/LinuxAssignment$ ls
data.txt docs file1.txt
arati@Arati:~/LinuxAssignment$ find -name "*.txt"
./docs/file2.txt
./file1.txt
./data.txt
arati@Arati:~/LinuxAssignment$ |
```

- b. Display lines containing a specific word in a file (provide a file name and the specific word to search).

Output:

```
arati@Arati: ~/LinuxAssignme × + ▾  
arati@Arati:~/LinuxAssignment$ ls  
data.txt docs file1.txt  
arati@Arati:~/LinuxAssignment$ grep -n "Hello" file1.txt  
1:Hello, Everyone  
arati@Arati:~/LinuxAssignment$ cat file1.txt  
Hello, Everyone  
arati@Arati:~/LinuxAssignment$ |
```

h) System Information:

a. Display the current system date and time.

Output:

```
arati@Arati: ~/LinuxAssignme × + ▾  
arati@Arati:~/LinuxAssignment$ ls  
data.txt docs file1.txt  
arati@Arati:~/LinuxAssignment$ grep -n "Hello" file1.txt  
1:Hello, Everyone  
arati@Arati:~/LinuxAssignment$ cat file1.txt  
Hello, Everyone  
arati@Arati:~/LinuxAssignment$ man date  
arati@Arati:~/LinuxAssignment$ date  
Thu Feb 27 12:52:49 UTC 2025  
arati@Arati:~/LinuxAssignment$
```

i) Networking:

a. Display the IP address of the system.

Output:

```
arati@Arati: ~/LinuxAssignme × + ▾  
arati@Arati:~/LinuxAssignment$ hostname -I  
172.17.126.11  
arati@Arati:~/LinuxAssignment$ |
```

- b. Ping a remote server to check connectivity (provide a remote server address to ping).**

```
arati@Arati: ~/LinuxAssignme × + ∨  
arati@Arati:~/LinuxAssignment$ hostname -I  
172.17.126.11  
arati@Arati:~/LinuxAssignment$ ping  
ping: usage error: Destination address required  
arati@Arati:~/LinuxAssignment$ ping 172.17.126.11  
PING 172.17.126.11 (172.17.126.11) 56(84) bytes of data.  
64 bytes from 172.17.126.11: icmp_seq=1 ttl=64 time=1.76 ms  
64 bytes from 172.17.126.11: icmp_seq=2 ttl=64 time=0.076 m  
s  
64 bytes from 172.17.126.11: icmp_seq=3 ttl=64 time=0.038 m  
s  
64 bytes from 172.17.126.11: icmp_seq=4 ttl=64 time=0.047 m  
s  
^C  
--- 172.17.126.11 ping statistics ---  
66 packets transmitted, 66 received, 0% packet loss, time 6  
7590ms  
k) rtt min/avg/max/mdev = 0.034/0.083/1.761/0.210 ms  
arati@Arati:~/LinuxAssignment$
```

- a. Compress the "docs" directory into a zip file.

```
a/  
arati@Arati:~/LinuxAssignment$ zip -r duplicate.zip duplica  
te.txt  
    adding: duplicate.txt (deflated 30%)  
arati@Arati:~/LinuxAssignment$ cat duplicate.zip  
duplicate.txtUT  m g m gux  
                  M A  
        zT_h K "   
  RE  ! ! ? %  
? @=    &    A P ?  \fw  X          .      S  U   G | PLn[ZrK&   
  duplicate.txtUT m g x
```

b. Extract the contents of the zip file into a new directory.

Output:

```
arati@Arati:~/LinuxAssignment$ ls
data.txt      duplicate.zip  input.txt
docs          file1.txt     numbers.txt
duplicate.txt  fruit.txt     output.txt
arati@Arati:~/LinuxAssignment$ unzip -d docs duplicate.zip
Archive:  duplicate.zip
  inflating: docs/duplicate.txt
arati@Arati:~/LinuxAssignment$ cd docs/
arati@Arati:~/LinuxAssignment/docs$ ls
duplicate.txt  file2.txt
arati@Arati:~/LinuxAssignment/docs$ |
```

k) File Editing:

a. Open the "file1.txt" file in a text editor and add some text to it.

Output:

```
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, Everyone
arati@Arati:~/LinuxAssignment$ nano file.txt
arati@Arati:~/LinuxAssignment$ nano file1.txt
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, Everyone Cdac Students
arati@Arati:~/LinuxAssignment$ |
```

b. Replace a specific word in the "file1.txt" file with another word (provide the original word and the word to replace it with).

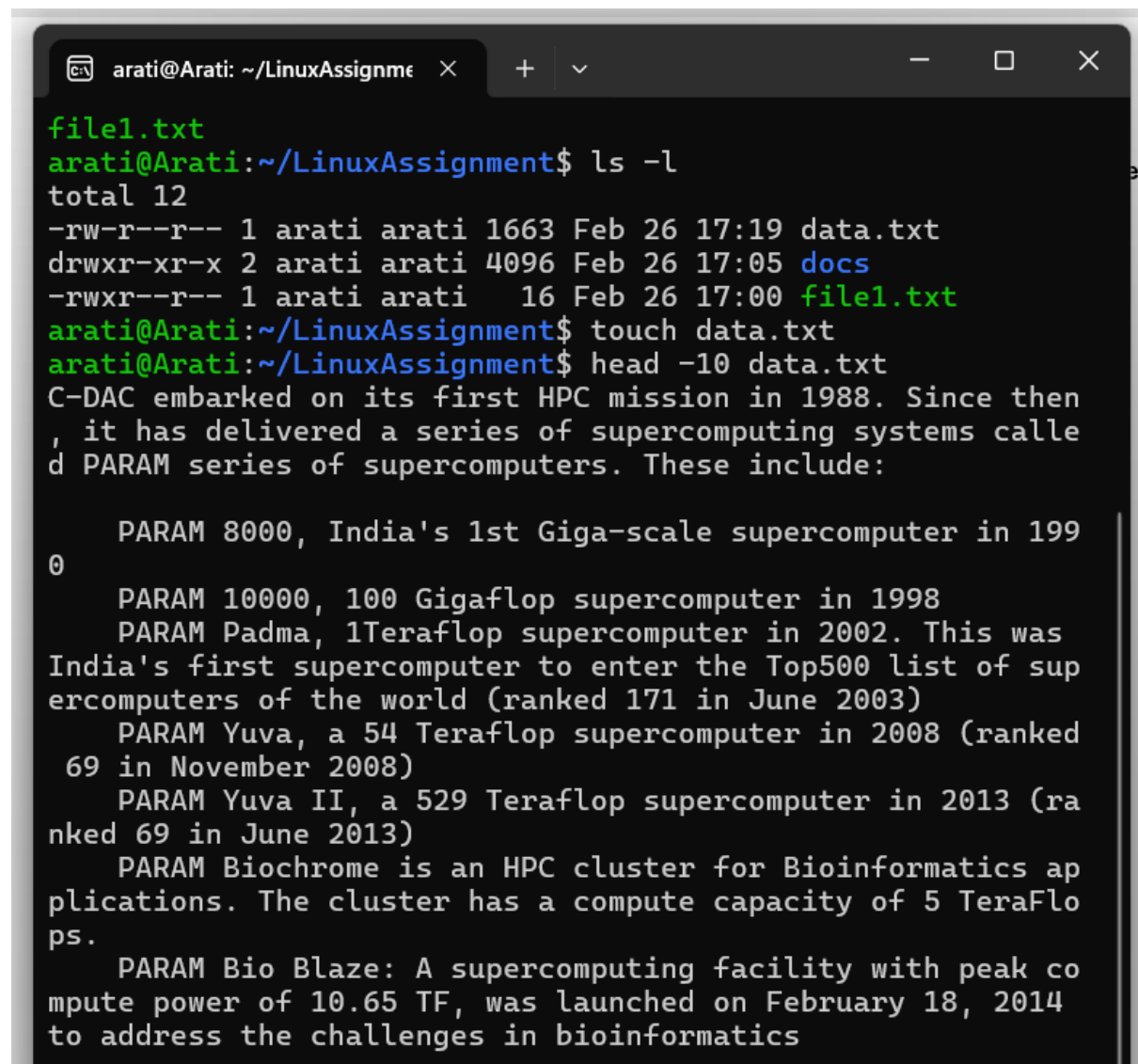
Output:

```
arati@Arati:~/LinuxAssignment$ cat file1.txt
Hello, Everyone Cdac Students
arati@Arati:~/LinuxAssignment$ sed 's/hello/world/' file1.t
xt
Hello, Everyone Cdac Students
arati@Arati:~/LinuxAssignment$ sed 's/Hello/world/' file1.t
xt
world, Everyone Cdac Students
arati@Arati:~/LinuxAssignment$ |
```

Problem 2: Read the instructions carefully and answer accordingly. If there is any need to insert some data then do that as well.

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.

Output :

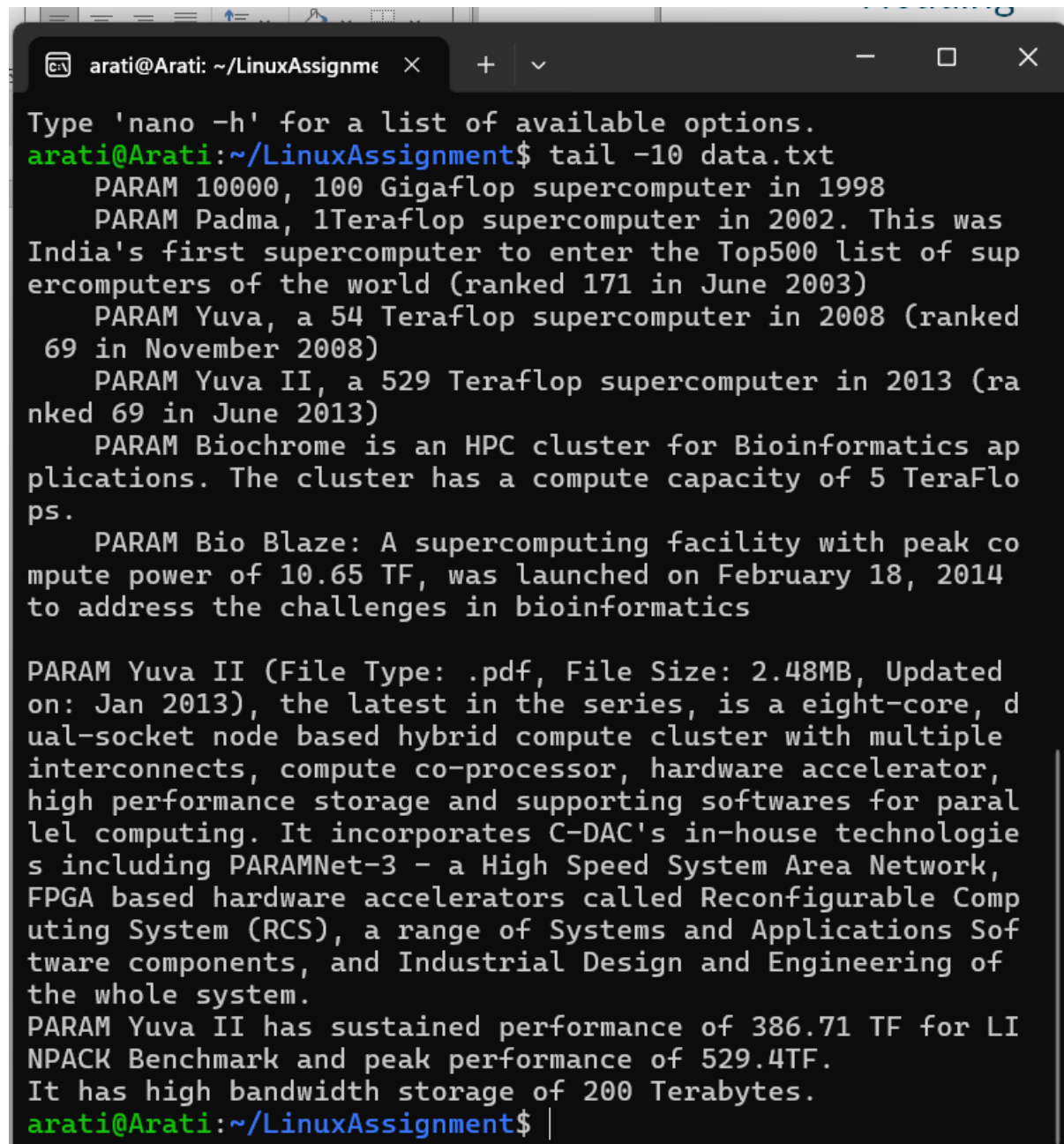
A terminal window titled 'arati@Arati: ~/LinuxAssignme' with standard window controls. The terminal shows a sequence of commands and their outputs. First, 'ls -l' is run, listing files 'data.txt', 'docs', and 'file1.txt'. Then, 'touch data.txt' is executed. Finally, 'head -10 data.txt' is run, displaying the first 10 lines of the file. The file content describes the PARAM series of supercomputers, listing models like PARAM 8000, 10000, Padma, Yuva, Yuva II, Biochrome, and Bio Blaze with their respective specifications and launch dates.

```
file1.txt
arati@Arati:~/LinuxAssignment$ ls -l
total 12
-rw-r--r-- 1 arati arati 1663 Feb 26 17:19 data.txt
drwxr-xr-x 2 arati arati 4096 Feb 26 17:05 docs
-rwxr--r-- 1 arati arati 16 Feb 26 17:00 file1.txt
arati@Arati:~/LinuxAssignment$ touch data.txt
arati@Arati:~/LinuxAssignment$ head -10 data.txt
C-DAC embarked on its first HPC mission in 1988. Since then
, it has delivered a series of supercomputing systems calle
d PARAM series of supercomputers. These include:

    PARAM 8000, India's 1st Giga-scale supercomputer in 199
0
    PARAM 10000, 100 Giga-flop supercomputer in 1998
    PARAM Padma, 1Teraflop supercomputer in 2002. This was
India's first supercomputer to enter the Top500 list of sup
ercomputers of the world (ranked 171 in June 2003)
    PARAM Yuva, a 54 Teraflop supercomputer in 2008 (ranked
69 in November 2008)
    PARAM Yuva II, a 529 Teraflop supercomputer in 2013 (ra
nked 69 in June 2013)
    PARAM Biochrome is an HPC cluster for Bioinformatics ap
plications. The cluster has a compute capacity of 5 TeraFlo
ps.
    PARAM Bio Blaze: A supercomputing facility with peak co
mpute power of 10.65 TF, was launched on February 18, 2014
to address the challenges in bioinformatics
```


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

Output :

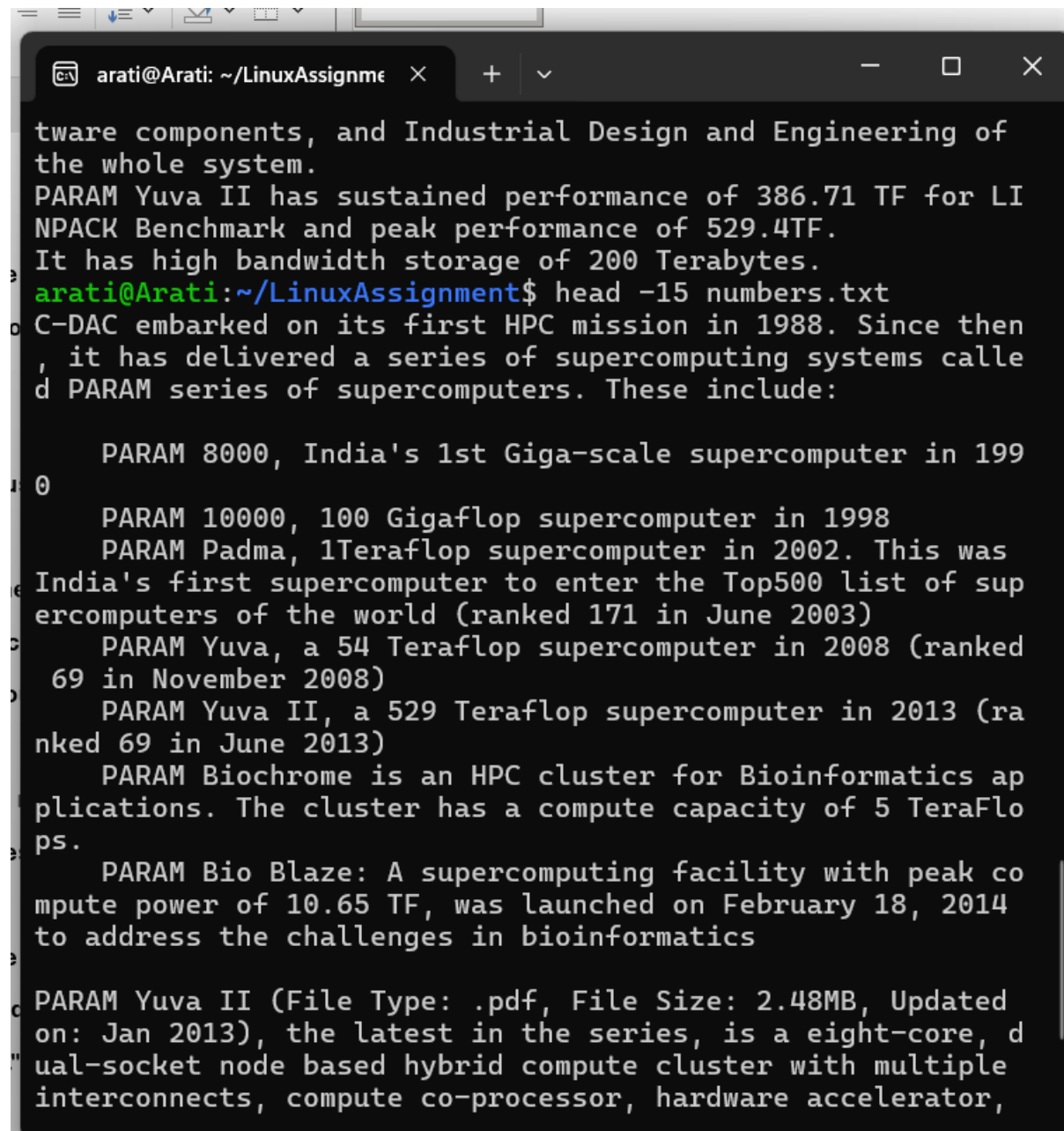


```
arati@Arati: ~/LinuxAssignme x + v - □ ×
Type 'nano -h' for a list of available options.
arati@Arati:~/LinuxAssignment$ tail -10 data.txt
PARAM 10000, 100 GigaFlop supercomputer in 1998
PARAM Padma, 1TeraFlop supercomputer in 2002. This was
India's first supercomputer to enter the Top500 list of supercomputers of the world (ranked 171 in June 2003)
PARAM Yuva, a 54 TeraFlop supercomputer in 2008 (ranked 69 in November 2008)
PARAM Yuva II, a 529 TeraFlop supercomputer in 2013 (ranked 69 in June 2013)
PARAM Biochrome is an HPC cluster for Bioinformatics applications. The cluster has a compute capacity of 5 TeraFlops.
PARAM Bio Blaze: A supercomputing facility with peak compute power of 10.65 TF, was launched on February 18, 2014 to address the challenges in bioinformatics

PARAM Yuva II (File Type: .pdf, File Size: 2.48MB, Updated on: Jan 2013), the latest in the series, is a eight-core, dual-socket node based hybrid compute cluster with multiple interconnects, compute co-processor, hardware accelerator, high performance storage and supporting softwares for parallel computing. It incorporates C-DAC's in-house technologies including PARAMNet-3 - a High Speed System Area Network, FPGA based hardware accelerators called Reconfigurable Computing System (RCS), a range of Systems and Applications Software components, and Industrial Design and Engineering of the whole system.
PARAM Yuva II has sustained performance of 386.71 TF for LINPACK Benchmark and peak performance of 529.4TF.
It has high bandwidth storage of 200 Terabytes.
arati@Arati:~/LinuxAssignment$ |
```

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.

Output:



```
arati@Arati: ~/LinuxAssignme
tware components, and Industrial Design and Engineering of
the whole system.
PARAM Yuva II has sustained performance of 386.71 TF for LI
NPack Benchmark and peak performance of 529.4TF.
It has high bandwidth storage of 200 Terabytes.
arati@Arati:~/LinuxAssignment$ head -15 numbers.txt
C-DAC embarked on its first HPC mission in 1988. Since then
, it has delivered a series of supercomputing systems calle
d PARAM series of supercomputers. These include:

    PARAM 8000, India's 1st Giga-scale supercomputer in 199
0
    PARAM 10000, 100 GigaFlop supercomputer in 1998
    PARAM Padma, 1TeraFlop supercomputer in 2002. This was
India's first supercomputer to enter the Top500 list of sup
ercomputers of the world (ranked 171 in June 2003)
    PARAM Yuva, a 54 TeraFlop supercomputer in 2008 (ranked
69 in November 2008)
    PARAM Yuva II, a 529 TeraFlop supercomputer in 2013 (ra
nked 69 in June 2013)
    PARAM Biochrome is an HPC cluster for Bioinformatics ap
plications. The cluster has a compute capacity of 5 TeraFlo
ps.
    PARAM Bio Blaze: A supercomputing facility with peak co
mpute power of 10.65 TF, was launched on February 18, 2014
to address the challenges in bioinformatics

PARAM Yuva II (File Type: .pdf, File Size: 2.48MB, Updated
on: Jan 2013), the latest in the series, is a eight-core, d
ual-socket node based hybrid compute cluster with multiple
interconnects, compute co-processor, hardware accelerator,
```

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

Output:

```
arati@Arati:~/LinuxAssignment$ tail -3 numbers.txt
PARAM Yuva II (File Type: .pdf, File Size: 2.48MB, Updated
on: Jan 2013), the latest in the series, is a eight-core, d
ual-socket node based hybrid compute cluster with multiple
interconnects, compute co-processor, hardware accelerator,
high performance storage and supporting softwares for paral
lel computing. It incorporates C-DAC's in-house technologie
s including PARAMNet-3 - a High Speed System Area Network,
FPGA based hardware accelerators called Reconfigurable Comp
uting System (RCS), a range of Systems and Applications Sof
tware components, and Industrial Design and Engineering of
the whole system.
PARAM Yuva II has sustained performance of 386.71 TF for LI
NPACK Benchmark and peak performance of 529.4TF.
It has high bandwidth storage of 200 Terabytes.
arati@Arati:~/LinuxAssignment$ |
```

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."

Output:

```
arati@Arati:~/LinuxAssignment$ cat input.txt
Hello , Cdac Students
arati@Arati:~/LinuxAssignment$ tr '[:lower:]' '[:upper:]' <
input.txt > output.txt
arati@Arati:~/LinuxAssignment$ cat output.txt
HELLO , CDAC STUDENTS
arati@Arati:~/LinuxAssignment$ |
```

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."

Output:

```
arati@Arati: ~/LinuxAssignme × + ▾ - □ ×
sudo apt install john
arati@Arati:~/LinuxAssignment$ cat duplicate.txt | uniq
India
China
Nepal
Maldives
Afghanistan
Bangladesh
Bhutan
China
Maldives
Myanmar
Nepal
Pakistan
Sri Lanka
Afghanistan
Australia
arati@Arati:~/LinuxAssignment$ cat duplicate.txt
India
India
China
Nepal
Maldives
Afghanistan
Bangladesh
Bhutan
China
Maldives
Myanmar
Nepal
Pakistan
Sri Lanka
Afghanistan
Australia
arati@Arati:~/LinuxAssignment$ |
```

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."

Output :

```
arati@Arati:~/LinuxAssignment$ sort fruit.txt | uniq -c
  2 Apple
  1 Apricot
  1 Avocado
  1 Banana
  1 Banana
  1 Blueberry
  2 Cherry
  1 Grapefruit
  2 Kiwi
  1 Mango
  1 Orange
  1 Papaya
  1 Pineapple
  2 Strawberry
  1 Watermelon
```