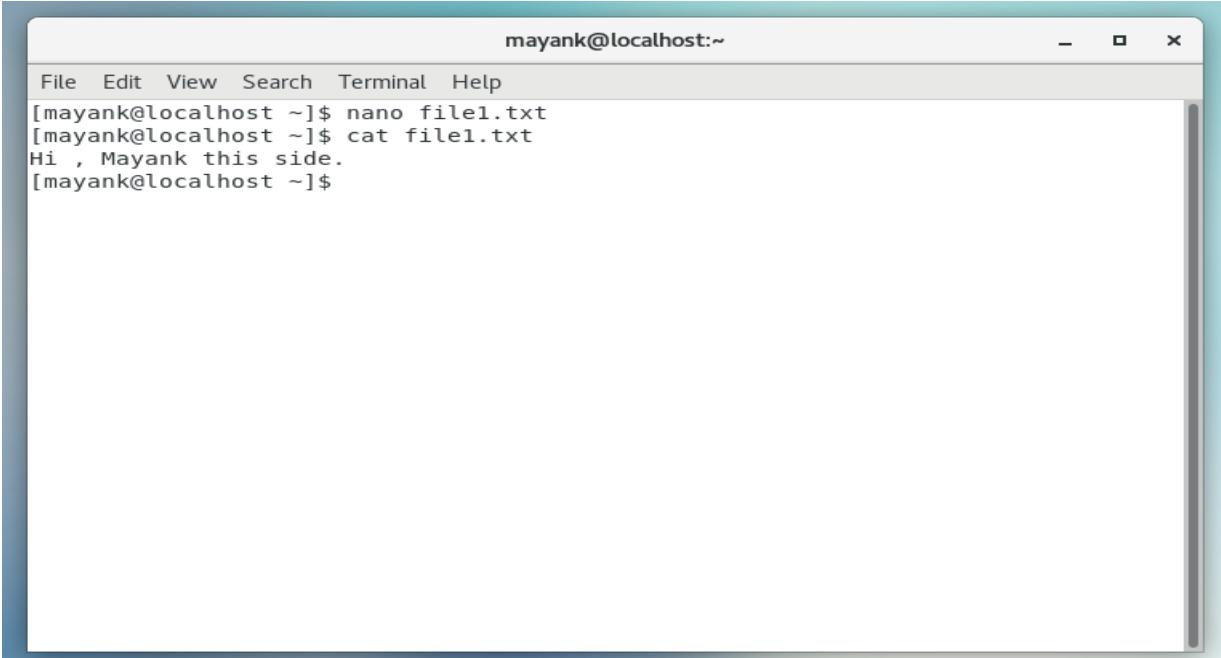


Assignment 2

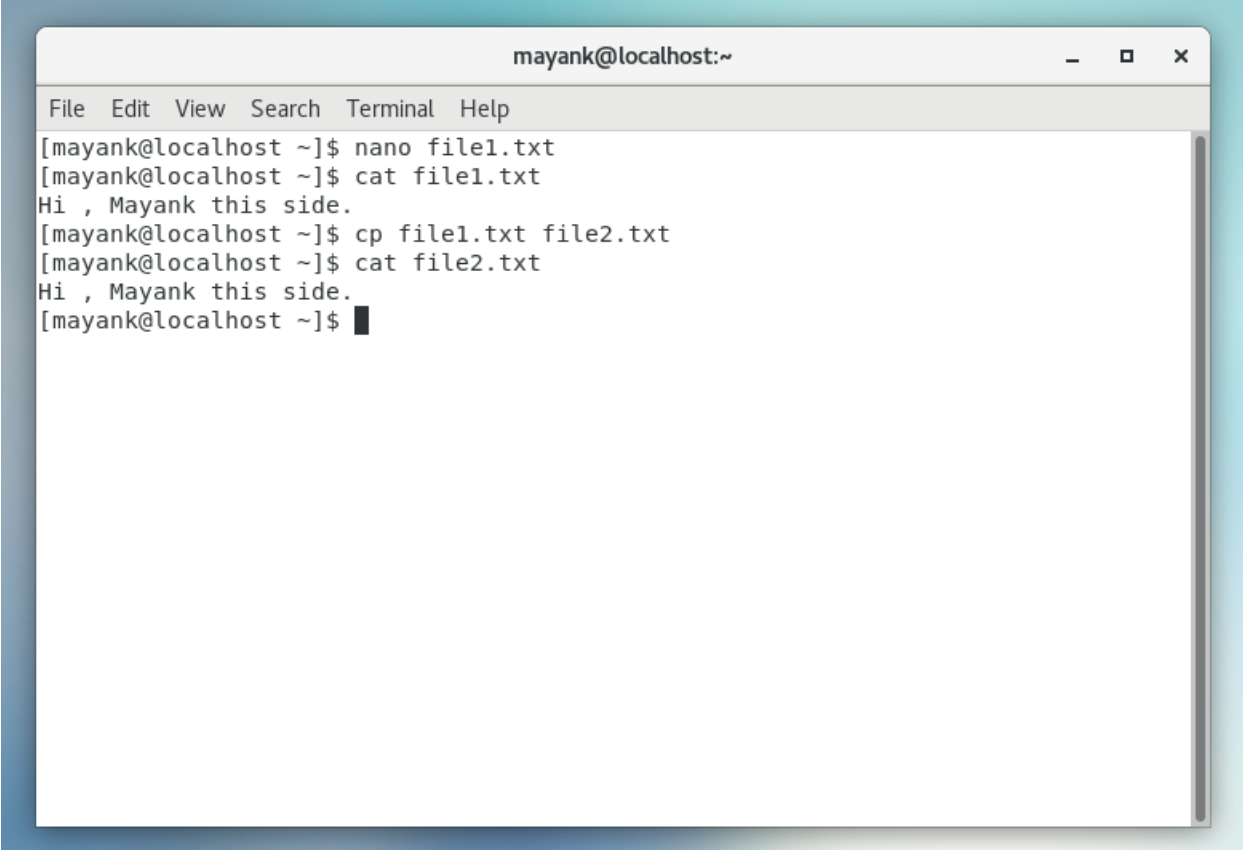
Playing with files

1. Create a file like nano file1.txt

A terminal window titled 'mayank@localhost:~' with standard window controls. The terminal shows a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The command history includes: '[mayank@localhost ~]\$ nano file1.txt', '[mayank@localhost ~]\$ cat file1.txt', and the output 'Hi , Mayank this side.' followed by the prompt '[mayank@localhost ~]\$'.

```
mayank@localhost:~
File Edit View Search Terminal Help
[mayank@localhost ~]$ nano file1.txt
[mayank@localhost ~]$ cat file1.txt
Hi , Mayank this side.
[mayank@localhost ~]$
```

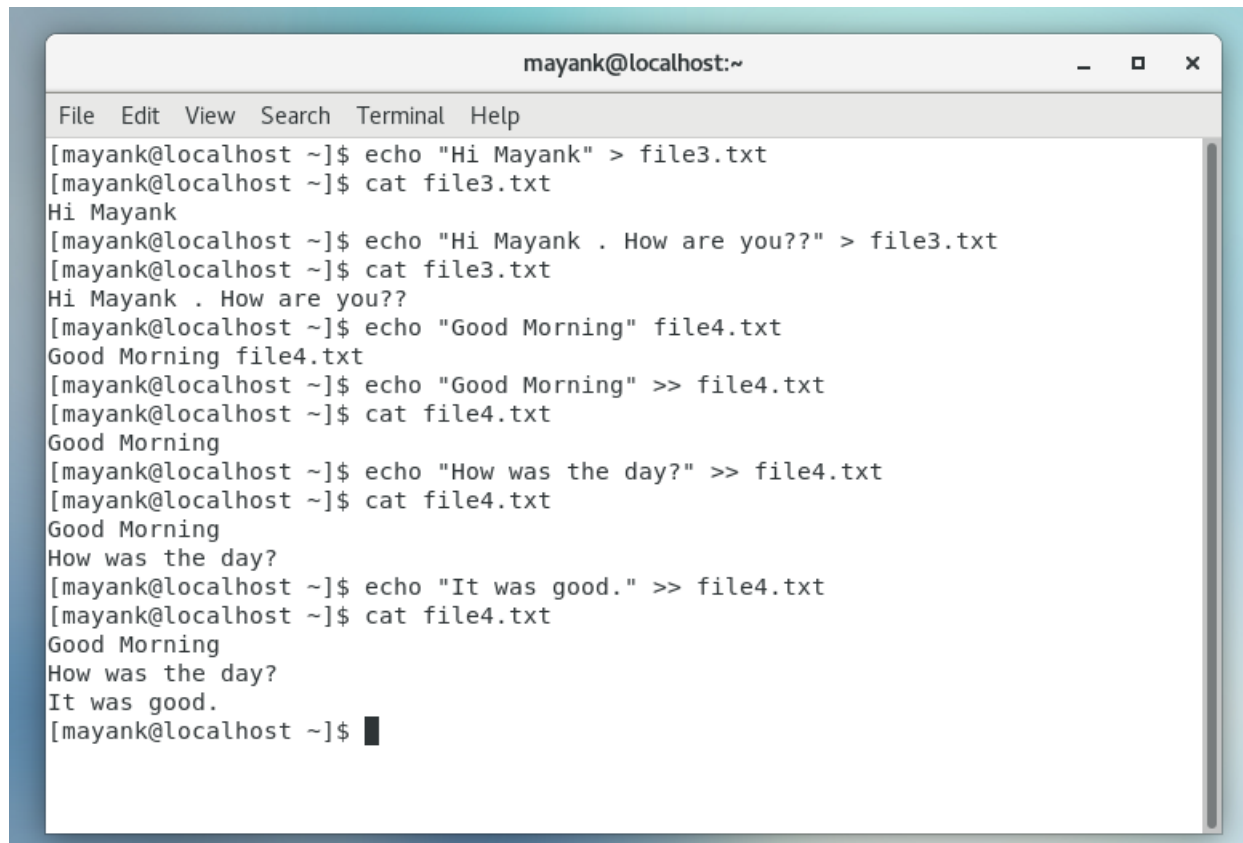
2. Now we will copy data from file1 to new file2

A terminal window titled 'mayank@localhost:~' with standard window controls. The terminal shows a sequence of commands: 'nano file1.txt', 'cat file1.txt' (outputting 'Hi , Mayank this side.'), 'cp file1.txt file2.txt', and 'cat file2.txt' (also outputting 'Hi , Mayank this side.'). The prompt '[mayank@localhost ~]\$' is visible at the end of each line.

```
mayank@localhost:~  
File Edit View Search Terminal Help  
[mayank@localhost ~]$ nano file1.txt  
[mayank@localhost ~]$ cat file1.txt  
Hi , Mayank this side.  
[mayank@localhost ~]$ cp file1.txt file2.txt  
[mayank@localhost ~]$ cat file2.txt  
Hi , Mayank this side.  
[mayank@localhost ~]$
```

3. Then we create a new file3.txt and file4.txt in home directory and add content in it.

Difference b/w > and >> while doing echo Both > and >> are used to provide the output to a file. But > will override the existing content of file and add the new content. Whereas >> will add the new contents at end of file

A screenshot of a terminal window titled 'mayank@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows a series of commands and their outputs. First, 'echo "Hi Mayank" > file3.txt' is executed, followed by 'cat file3.txt' which outputs 'Hi Mayank'. Then, 'echo "Hi Mayank . How are you??" > file3.txt' is executed, followed by 'cat file3.txt' which outputs 'Hi Mayank . How are you??'. Next, 'echo "Good Morning" file4.txt' is executed, followed by 'cat file4.txt' which outputs 'Good Morning file4.txt'. Then, 'echo "Good Morning" >> file4.txt' is executed, followed by 'cat file4.txt' which outputs 'Good Morning'. Next, 'echo "How was the day?" >> file4.txt' is executed, followed by 'cat file4.txt' which outputs 'Good Morning' and 'How was the day?'. Then, 'echo "It was good." >> file4.txt' is executed, followed by 'cat file4.txt' which outputs 'Good Morning', 'How was the day?', and 'It was good.'. The terminal ends with a prompt '[mayank@localhost ~]\$' and a cursor.

```
mayank@localhost:~
File Edit View Search Terminal Help
[mayank@localhost ~]$ echo "Hi Mayank" > file3.txt
[mayank@localhost ~]$ cat file3.txt
Hi Mayank
[mayank@localhost ~]$ echo "Hi Mayank . How are you??" > file3.txt
[mayank@localhost ~]$ cat file3.txt
Hi Mayank . How are you??
[mayank@localhost ~]$ echo "Good Morning" file4.txt
Good Morning file4.txt
[mayank@localhost ~]$ echo "Good Morning" >> file4.txt
[mayank@localhost ~]$ cat file4.txt
Good Morning
[mayank@localhost ~]$ echo "How was the day?" >> file4.txt
[mayank@localhost ~]$ cat file4.txt
Good Morning
How was the day?
[mayank@localhost ~]$ echo "It was good." >> file4.txt
[mayank@localhost ~]$ cat file4.txt
Good Morning
How was the day?
It was good.
[mayank@localhost ~]$
```

4. For remove a file or directory you can use the below two commands

```
mayank@localhost ~]$ rm file4.txt
mayank@localhost ~]$ rm file3.txt
mayank@localhost ~]$ mkdir ineuron2
mayank@localhost ~]$ rmdir ineuron2
mayank@localhost ~]$
```