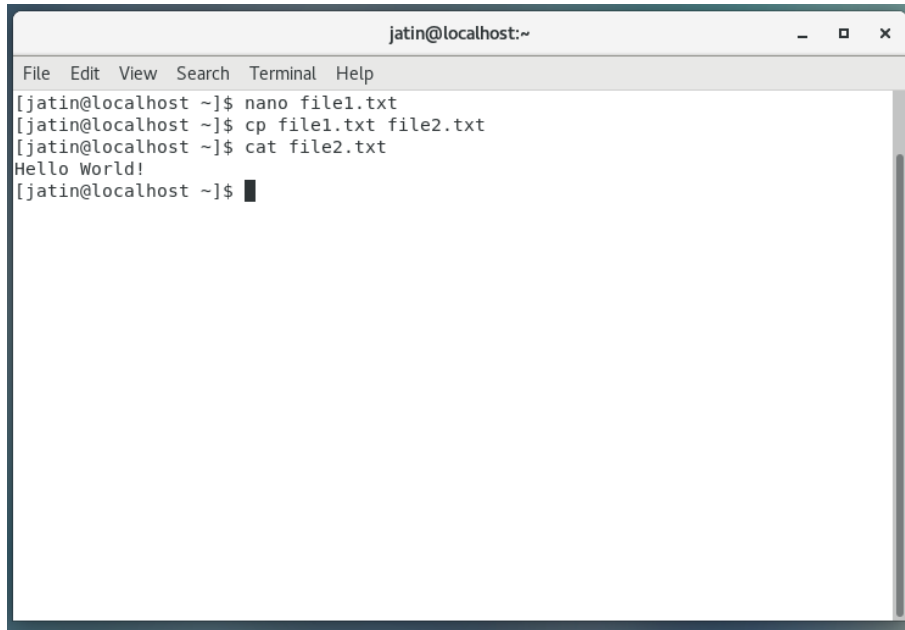


# ASSIGNMENT-2

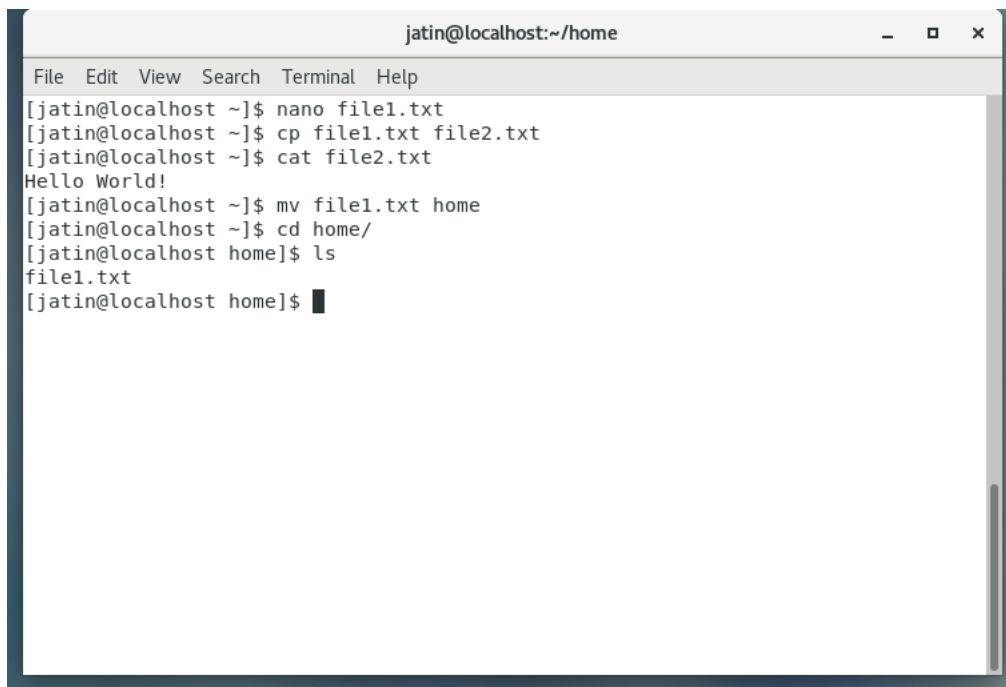
## Playing with files

### Copying data from file

A terminal window titled 'jatin@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[jatin@localhost ~]$ nano file1.txt
[jatin@localhost ~]$ cp file1.txt file2.txt
[jatin@localhost ~]$ cat file2.txt
Hello World!
[jatin@localhost ~]$
```

### Move File

A terminal window titled 'jatin@localhost:~/home' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

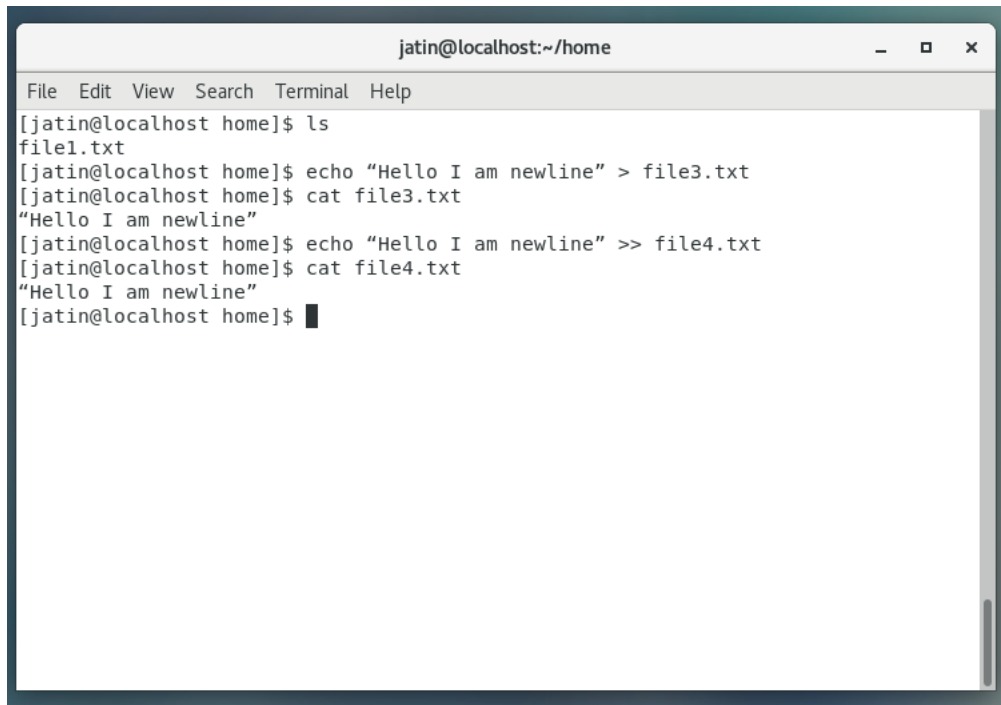
```
[jatin@localhost ~]$ nano file1.txt
[jatin@localhost ~]$ cp file1.txt file2.txt
[jatin@localhost ~]$ cat file2.txt
Hello World!
[jatin@localhost ~]$ mv file1.txt home
[jatin@localhost ~]$ cd home/
[jatin@localhost home]$ ls
file1.txt
[jatin@localhost home]$
```

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 terminal window titled 'jatin@localhost:~/home' with standard window controls. The terminal shows a series of commands and their outputs. First, 'ls' shows 'file1.txt'. Then, 'echo "Hello I am newline" > file3.txt' is executed, followed by 'cat file3.txt' which outputs 'Hello I am newline'. Next, 'echo "Hello I am newline" >> file4.txt' is executed, followed by 'cat file4.txt' which also outputs 'Hello I am newline'. The prompt returns after each command.

```
jatin@localhost:~/home
File Edit View Search Terminal Help
[jatin@localhost home]$ ls
file1.txt
[jatin@localhost home]$ echo "Hello I am newline" > file3.txt
[jatin@localhost home]$ cat file3.txt
"Hello I am newline"
[jatin@localhost home]$ echo "Hello I am newline" >> file4.txt
[jatin@localhost home]$ cat file4.txt
"Hello I am newline"
[jatin@localhost home]$
```