

## Practical - 4

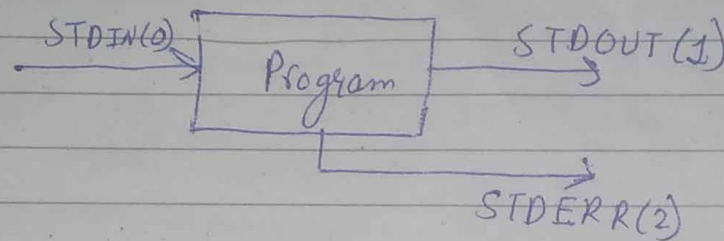
Date:-

॥ नमामि देवि सरस्वति ॥

Aim: To understand Redirection and Piping in UNIX-based system.

→ Every program we run on the command line automatically has three data streams connected to it.

STDIN (0) - Standard Input (data fed into the program)  
STDOUT (1) - Standard output (data printed by the program)  
STDERR (2) - Standard Error (for error messages)



→ Piping and redirection is the means by which we may connect these streams between programs and files to direct data.

### \* Redirecting to file

→ To redirect any data to a file instead of getting it in our output screen, we use ">" (greater than) operator).

→ It indicates to the command line that the output needs to be saved in a file.

1) > operator :- saves the output to a file.

e.g. `snch@snake-vm:~$ cat file.txt`  
`snch@snake-vm:~$ echo "Hello World" > file.txt`  
 Hello World

## \* Saving to an Existing File

- If we redirect to a file which does not exist, it will be created automatically.
- If we save into a file which already exists, then its contents will be cleared, then the new output <sup>will be</sup> saved to it.

## 2) >> operator

Description:- Appends the output to an existing file.

Output:-

```
sneh@snake-vm:~$ echo "Calm down" >> file.txt
Hello World
Calm Down
```

## \* Redirecting from a file

- We can read data from the file and feed it into the program via STDIN stream.

## 3) < operator

Description:- Redirecting the input ~~to~~ from a file to a command

Output:-

```
sneh@snake-vm:~$ wc -l file.txt
8 file.txt
sneh@snake-vm:~$ wc -l < file.txt
8
```



## \* Piping

- A pipe is a form of redirection used to send the output of one command/program to another command/program for further processing.
- Pipes are unidirectional i.e. data flows from left to right through pipeline

● Symbol: '|'

Output :-

```
snch @ snake-vm : ~ $ cat ls ls
dim.txt fleet.txt out.txt in.txt temp.txt temp2.txt
snch @ snake-vm : ~ $ ls | grep "temp"
temp.txt
temp2.txt
```