# Practical : 03 Date:

**Aim :** Process priorities:

1. Create a new directory and create six pipes in that directory.
2. Bounce a character between two pipes.
3. use top and ps to display.
4. Bounce another character between two other pipes but this same time start the command nice, verify that all cat processes are battling to the CPU.
5. Use ps to verify that the two new cat process have a nice value. Use the o and c.
6. use renice to increase the nice value from 10 to 15. Notice the difference between the usual commands.

# Theory :

Priority value — The priority value is the process’s actual priority which is used by the Linux kernel to schedule a task.

In Linux system priorities are 0 to 139 in which 0 to 99 for real-time and 100 to 139 for users.

Nice value — Nice values are user-space values that we can use to control the priority of a process. The nice value range is -20 to +19 where -20 is highest, 0 default and +19 is lowest.

The relation between nice value and priority is as such - Priority\_value

=Nice\_value+20

All processes have a priority and a nice value. Higher priority processes will get more CPU time than lower priority processes. You can influence this with the nice and renice commands.

**nice and renice command** :  
nice command is used to start a process with specified nice value, which renice command is used to alter priority of running process.

**Usage of nice command :**

programs running on it(processes) are not responding quickly, in that case if you want to kill some of the processes, you need to start a terminal, if you start your bash shell normally, it will also

produce lag but you can avoid this by starting the bash shell with high priority.

For example:

nice -n -5 bash

**Usage of renice command :**  
To alter priority of running process, we use renice command.

renice value PID

value is new priority to be assigned  
PID is PID of process whose priority is to be changed

pipes (mkfi):

a pipe is a connection between two processes, such that the standard output from one process becomes the standard input of the other process. In UNIX Operating System, Pipes are useful for communication between related processes (interprocess communication).

Processes can communicate with each other via pipes. These pipes can be created with the mkfifo command.

Syntax for creating pipes in directory:

~$ mkdir pipes; cd pipes

~$ mkfifo p1 p2 p3….

~$ ls-1 Cat:

To demonstrate the use of the top and renice commands we will make the cat command use the

previously created pipes to generate a full load on the CPU. The cat is copied with a distinct name to the current directory. Syntax: top:

Just running top without options or arguments will display all processes and an overview of

information. The top of the top screen might look something like this.

**nice and renice command** :  
 nice command is used to start a process with specified nice value, which renice command is used to alter priority of running process.

**Usage of nice command :**

programs running on it(processes) are not responding quickly, in that case if you want to kill some of the processes, you need to start a terminal, if you start your bash shell normally, it will also produce lag but you can avoid this by starting the bash shell with high priority.

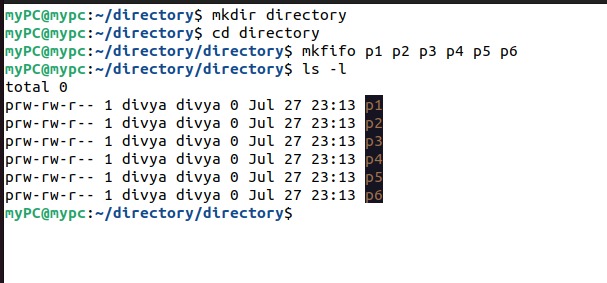
For example:

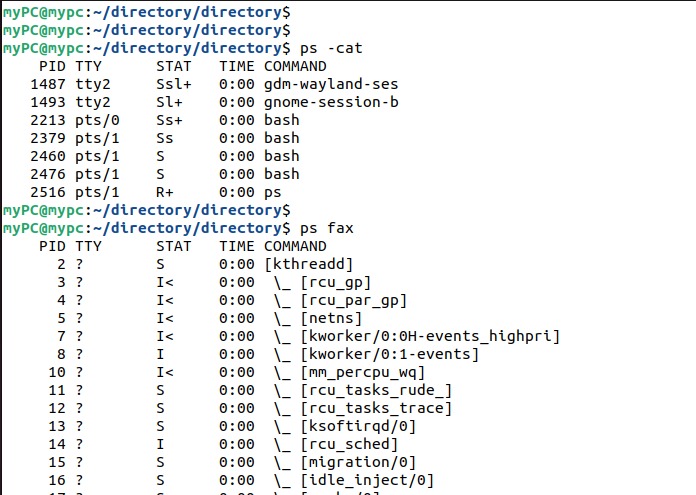
nice -n -5 bash

**Usage of renice command :**  
To alter priority of running process, we use renice command.

renice value PID

value is new priority to be assigned  
PID is PID of process whose priority is to be changed





**Conclusion**: Thus, we have successfully understood the process priorities and implemented all its topics.