Process Management in Linux

What is a Process?

An instance of a program is called a Process.
In simple terms, any command that you give to your Linux machine starts a new process.
Having multiple processes for the same program is possible.

Types of Processes:

Foreground Processes: They run on the screen and need input from the user. For example Office Programs
Background Processes: They run in the background and usually do not need user input. For example Antivirus.

Running a Foreground Process

To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.
When using the Terminal, you will have to wait, until the foreground process runs.

Running a Background process

If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.

PS

This command stands for ‘Process Status’.

It is similar to the “Task Manager” that pop-ups in a Windows Machine when we use Cntrl+Alt+Del.

This command is similar to ‘top’ command but the information displayed is different.

To check all the processes running under a user, use the command – $ps ux

```
home@VirtualBox:~$ ps ux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
home      1114  0.0  0.8  46548  8512 ?        Ssl  Sep03   0:00 gnome-sess
home      1151  0.0  0.0   3856   140 ?        Ss   Sep03   0:00 /usr/bin/s
home      1154  0.0  0.0   3748   484 ?        S    Sep03   0:00 /usr/bin/d
home      1155  0.1  0.2   6656  3036 ?        Ss   Sep03   0:18 //bin/dbus
home      1157  0.0  0.2   9148  2368 ?        S    Sep03   0:00 /usr/lib/g
home      1162  0.0  0.2  31588  2296 ?        Ssl  Sep03   0:00 /usr/lib/g
home      1174  0.0  1.4 132472 14884 ?        Sl   Sep03   0:03 /usr/lib/g
```

You can also check the process status of a single process, use the syntax – $ps PID

```
guru99@VirtualBox:~$ ps 1268
  PID TTY      STAT   TIME COMMAND
 1268 ?        S<l    0:02 /usr/bin/pulseaudio --start --log-target=syslog
```

Kill

This command terminates running processes on a Linux machine.
To use these utilities you need to know the PID (process id) of the process you want to kill
Syntax – $kill PID

To find the PID of a process simply type
Pidof Process name

```
home@VirtualBox:~$ pidof Photoshop.exe
1525
home@VirtualBox:~$ kill 1525
```

Parent and Child Processes

- Each unix process has two ID numbers assigned to it:
- The Process ID (pid) and the Parent process ID (ppid).
- Each user process in the system has a parent process.
- Most of the commands that you run have the shell as their parent.
- Check the **ps -f** example where this command listed both the process ID and the parent process ID.

Priority of process in Linux | nice value

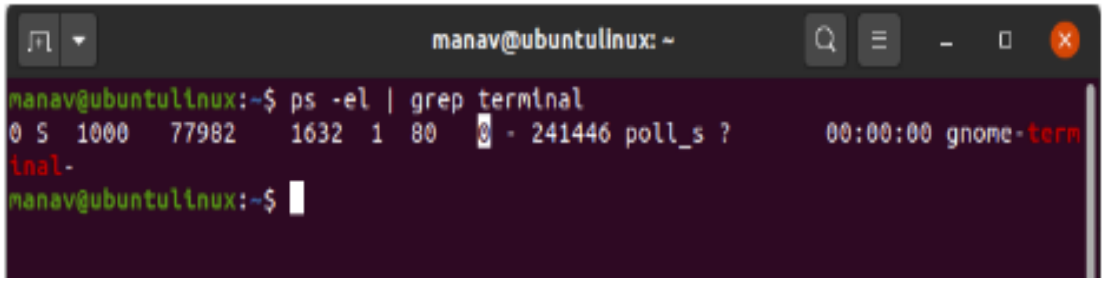
The running instance of program is process, and each process needs space in RAM and CPU time to be executed, each process has its priority in which it is executed.

- The **nice** command lets you run a command at a priority lower than the command's normal priority.
- The *Command* parameter is the name of any executable file on the system.
- If you do not specify an *Increment* value the **nice** command defaults to an increment of 10.
- The priority of a process is often called its nice value.
- The nice value can range from -20 to 19, with 19 being the lowest priority.
- The nice value is used by the system to calculate the current priority of a running process. Use the **ps** command with the **-l** flag to view a command's nice value.
- The nice value appears under the **NI** heading in the **ps** command output.

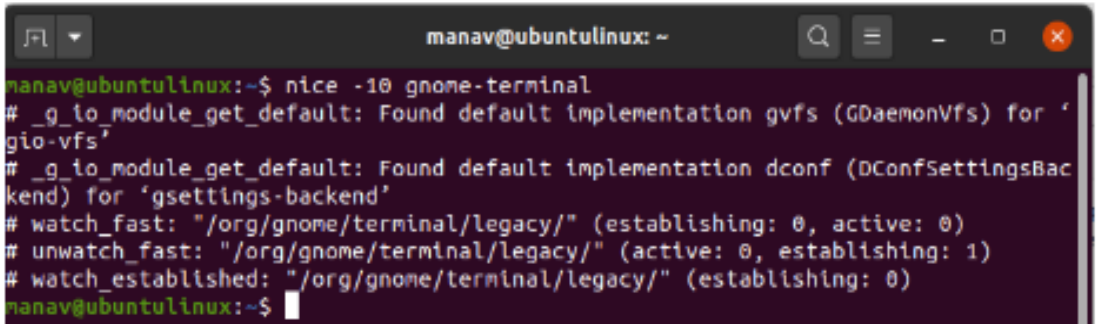
1. To check the nice value of a process.

$ps -el | grep terminal

The eight highlighted value is
the nice value of the process.



2. To set the priority of a process
$nice -10 gnome-terminal



•You can change the scheduling priority of a running process to a value lower or higher than the base scheduling priority by using the **renice** command from the command line.

•This command changes the nice value of a process.

•To change the priority of a running process, type the following:

•$renice *Priority* -p *ProcessID*

- •where ,
- •*Priority* is a number in the range of -20 to 20.
- •The higher the number, the lower the priority.
- •If you use zero, the process will run at its base scheduling priority.
- •*ProcessID* is the PID for which you want to change the priority.

You can observe that nice value of process(PID = 2371) is 0, now let's try to set the new priority of 5 to this process.

$renice 5 2371

Output:
2371 (process ID) old priority 0, new priority 5