***Lab No-06***

***Name of the lab:*****Linux Command for process**

***Name:******Shuvo Biswas ID:******IT-16014***

***Objective:***

(1) How to Manage Processes from the Linux Terminal?

(2) Run the following process commands in Linux.

Top, htop, Ps, pstree, kill, pgrep, pkill ,killall, renice, xkill .

**(1) How to Manage Processes from the Linux Terminal?**

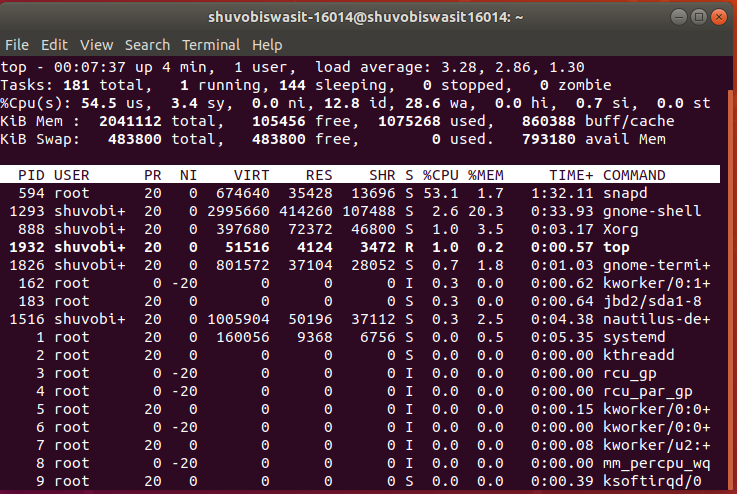
**Ans:**

The Linux terminal has a number of useful commands that can display running processes, kill them, and change their priority level. This post lists the classic, traditional commands, as well as some more useful, modern ones.

Many of the commands here perform a single function and can be combined — that’s the Unix philosophy of designing programs. Other programs, like htop, provide a friendly interface on top of the commands.

**(2) Run the following process commands in Linux .**

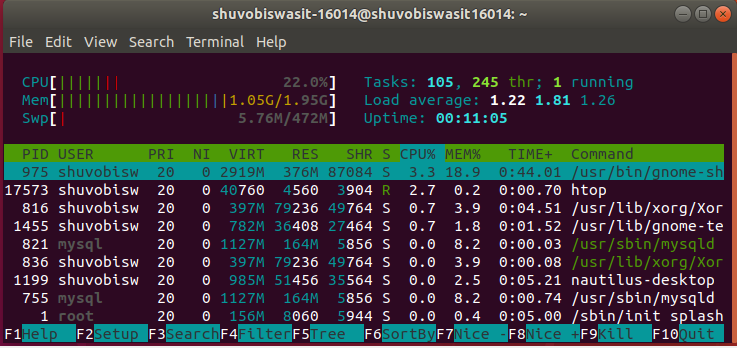
1. **Top :** The top command is the traditional way to view your system’s resource usage and see the processes that are taking up the most system resources. Top displays a list of processes, with the ones using the most CPU at the top.



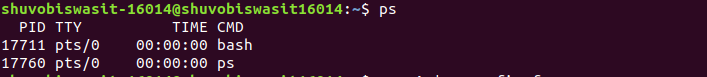
**(ii) htop :** The htop command is an improved top. It’s not installed by default on most Linux distributions . here’s the command you’ll need to install it on Ubuntu:

sudo apt-get install htop

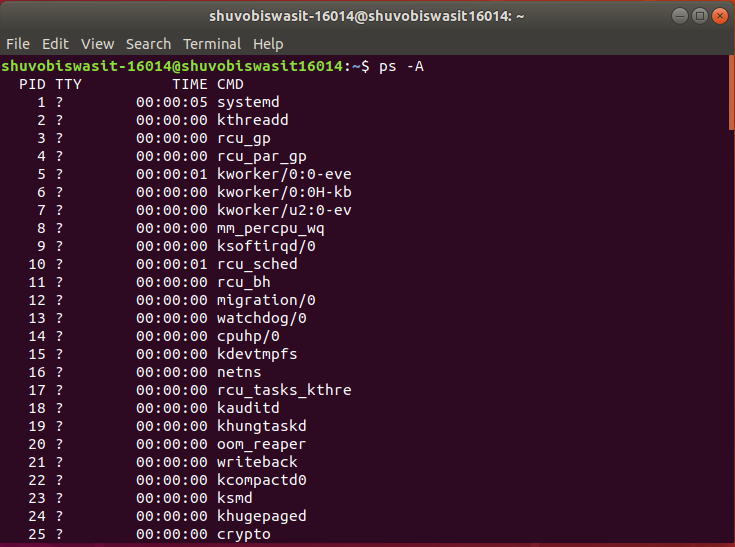
htop displays the same information with an easier-to-understand layout. It also lets you select processes with the arrow keys and perform actions, such as killing them or changing their priority, with the F keys.



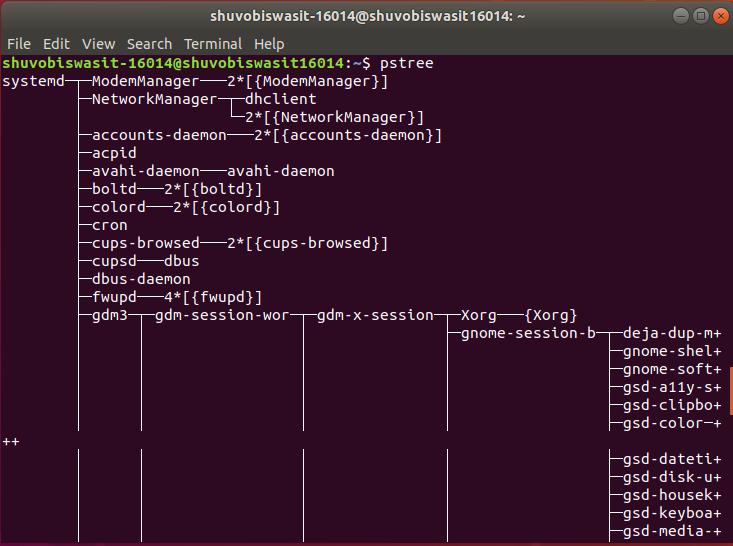
**(iii) ps :** The ps command lists running processes.



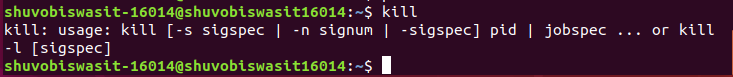
The following command lists all processes running on your system: ps -A



**(iv) pstree :** The pstree command is another way of visualizing processes. It displays them in tree format. So, for example, your X server and graphical environment would appear under the display manager that spawned them.



**(v) kill :** The kill command can kill a process, given its process ID.



You can get this information from the ps -A, top or pgrep commands.

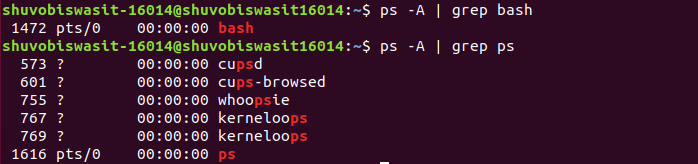
kill PID



**(vi) ps -A | grep** : We could also pipe the output through grep to search for a specific process without using any other commands. The following command would search for the bash and ps process:

ps -A | grep bash

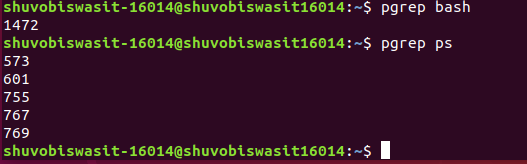
ps -A | grep ps



**(vii) pgrep :** Given a search term, pgrep returns the process IDs that match it. For example, you could use the following command to find bash and ps PID:

pgrep bash

pgrep ps



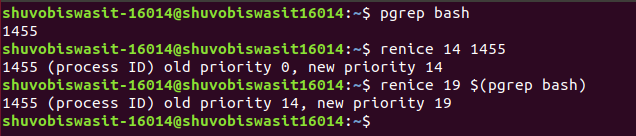
**(viii) pkill & killall :** The pkill and killall commands can kill a process, given its name. Use either command to kill bash:

pkill bash  
 killall bash



**(ix) renice :** The renice command changes the nice value of an already running process. The nice value determines what priority the process runs with. A value of -19 is very high priority, while a value of 14 is very low priority. A value of 0 is the default priority.

The renice command requires a process’s PID. The following command makes a process run with very low priority: renice 14 PID



**(x) xkill :** The *xkill* command is a way of easily killing graphical programs. Run it and your cursor will turn into an *x* sign. Click a program’s window to kill that program. If you don’t want to kill a program, you can back out of xkill by right-clicking instead.



**Conclusion :** From this lab, I know some commands which is used to handle process . By using these useful commands we can display running processes, kill them, and change their priority level. By using these commands, we can solve many complex problem in our operating system.