Mawlana Bhashani Science and Technology University

Lab-Report

Report No : 06

Experiment name : Linux command for process.

Course code : ICT-3110

Course title : Operating System Lab.

Date of Performance :

Date of Submission :

Submitted by

Name: Iqbal Hossen

ID: IT-18041

3rd year 1st semester

Session: 2017-18

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

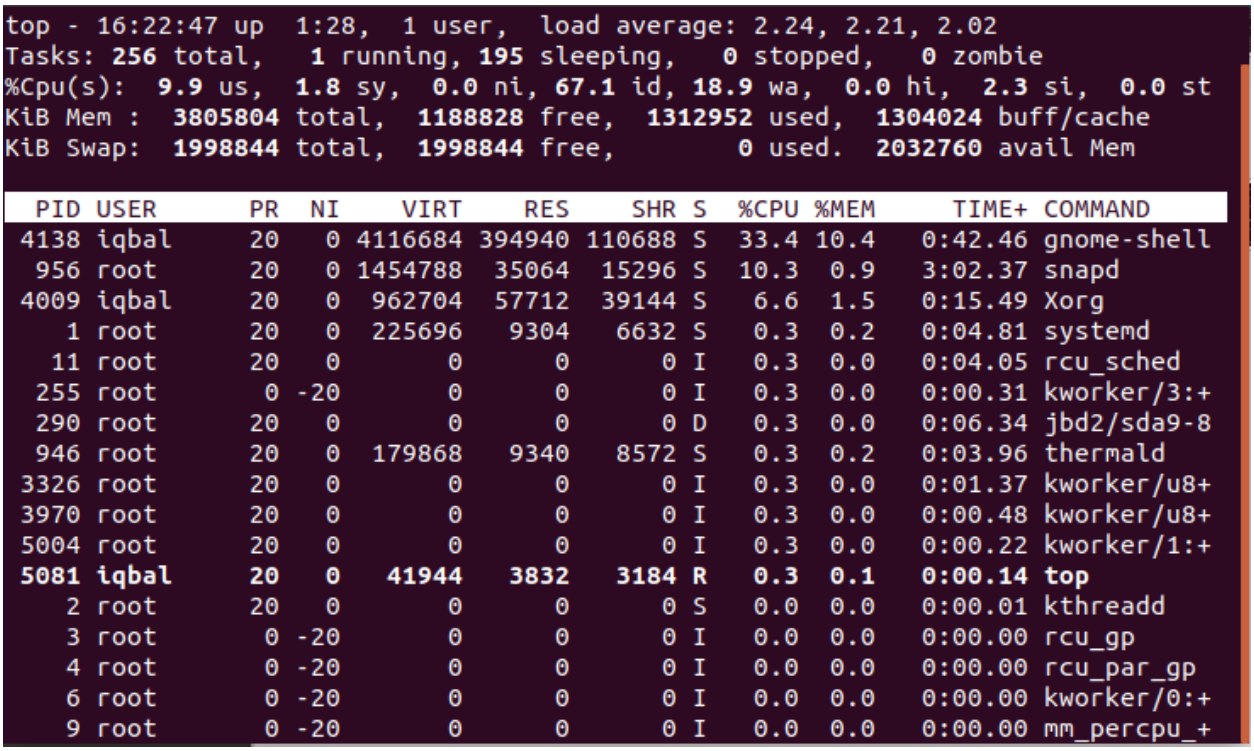
MBSTU.

i) How to Manage Processes from the Linux Terminal?

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. 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.

ii) 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.



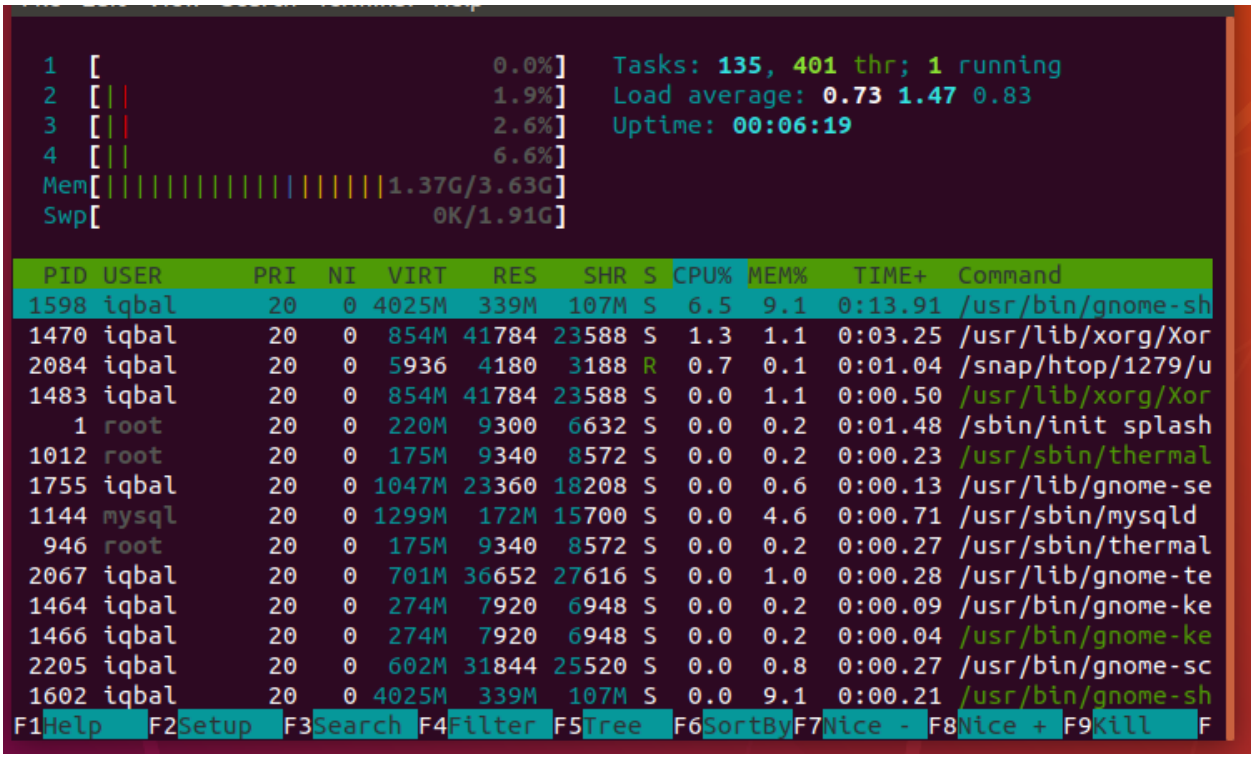
```
top - 16:22:47 up  1:28,  1 user,  load average: 2.24, 2.21, 2.02
Tasks: 256 total,   1 running, 195 sleeping,   0 stopped,   0 zombie
%Cpu(s):  9.9 us,  1.8 sy,  0.0 ni, 67.1 id, 18.9 wa,  0.0 hi,  2.3 si,  0.0 st
KiB Mem :  3805804 total,  1188828 free,  1312952 used,  1304024 buff/cache
KiB Swap:  1998844 total,  1998844 free,        0 used.  2032760 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 4138 iqbal     20   0 4116684 394940 110688 S  33.4 10.4   0:42.46 gnome-shell
  956 root      20   0 1454788  35064  15296 S  10.3  0.9   3:02.37 snapd
 4009 iqbal     20   0  962704  57712  39144 S   6.6  1.5   0:15.49 Xorg
    1 root      20   0  225696   9304   6632 S   0.3  0.2   0:04.81 systemd
   11 root      20   0       0      0      0 I   0.3  0.0   0:04.05 rcu_sched
  255 root       0 -20       0      0      0 I   0.3  0.0   0:00.31 kworker/3:+
  290 root      20   0       0      0      0 D   0.3  0.0   0:06.34 jbd2/sda9-8
  946 root      20   0  179868   9340   8572 S   0.3  0.2   0:03.96 thermald
 3326 root      20   0       0      0      0 I   0.3  0.0   0:01.37 kworker/u8+
 3970 root      20   0       0      0      0 I   0.3  0.0   0:00.48 kworker/u8+
 5004 root      20   0       0      0      0 I   0.3  0.0   0:00.22 kworker/1:+
 5081 iqbal     20   0   41944   3832   3184 R   0.3  0.1   0:00.14 top
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:+
    9 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_+
```

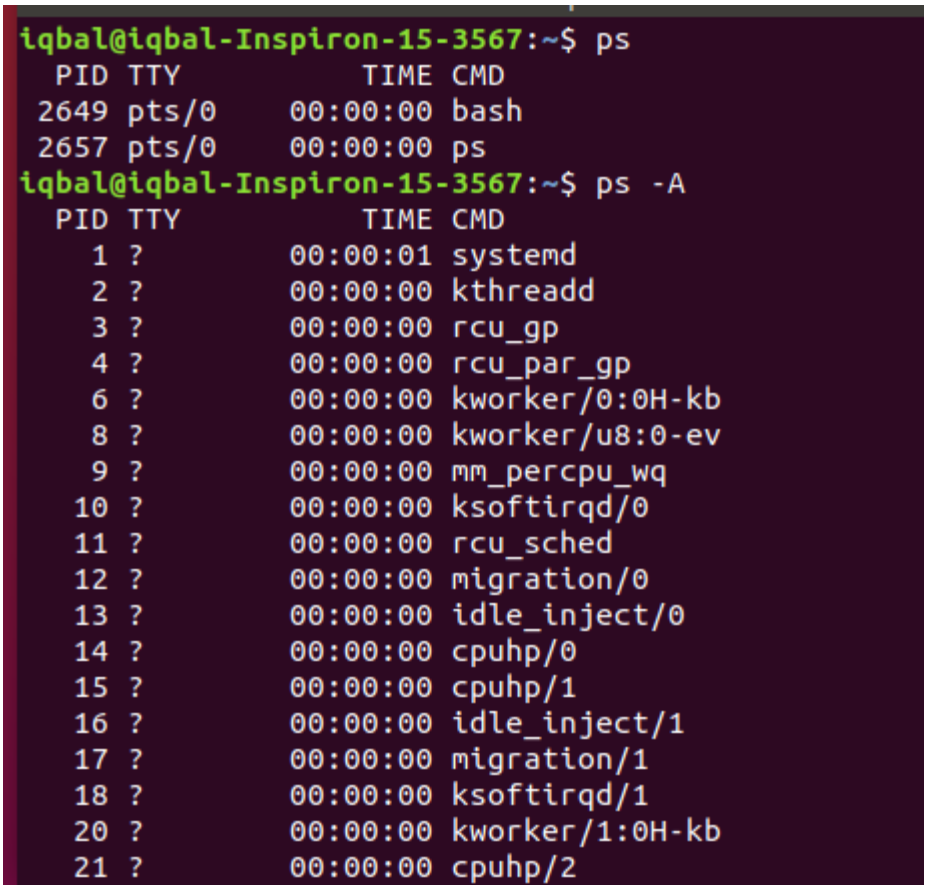
To exit top or htop, use the Ctrl-C keyboard shortcut. This keyboard shortcut usually kills the currently running process in the terminal.

2) 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:

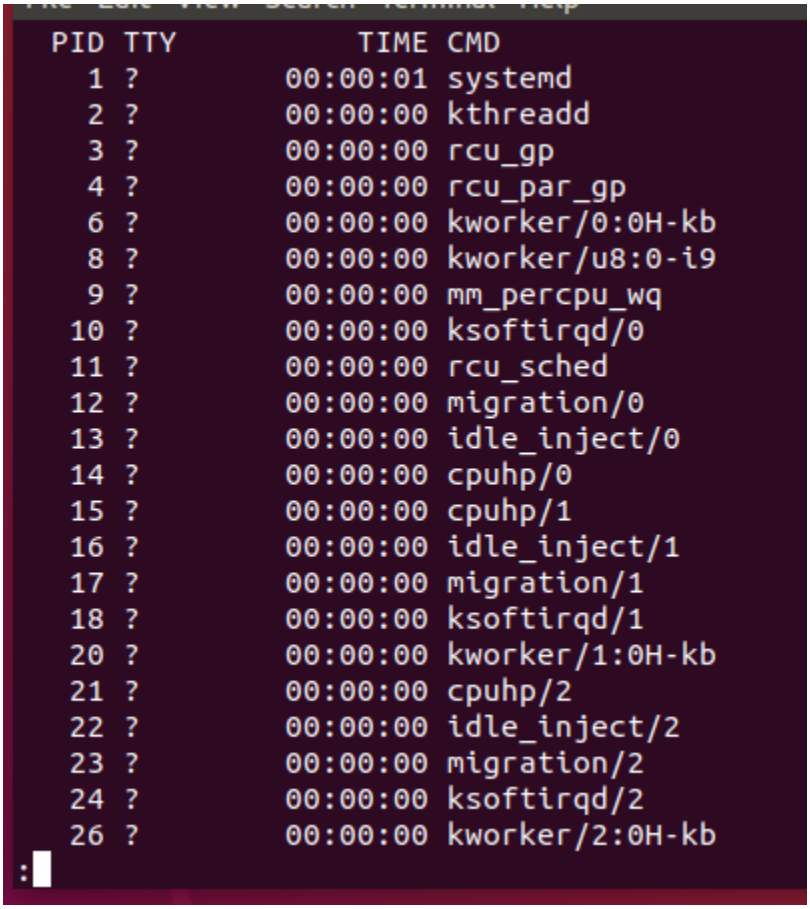
** sudo apt-get install htop.**



3) ps -A : The **ps** command lists running processes. The following command lists all processes running on your system:

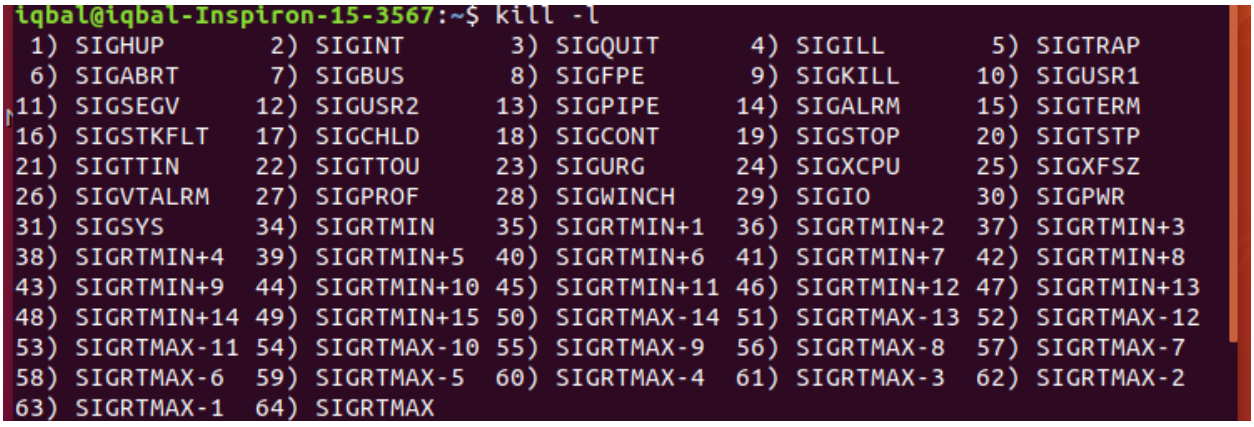


4) ps -A | less: ps -A may be too many processes to read at one time, so we can pipe the output through the **less** command to scroll through them at own pace.ps -A | less:



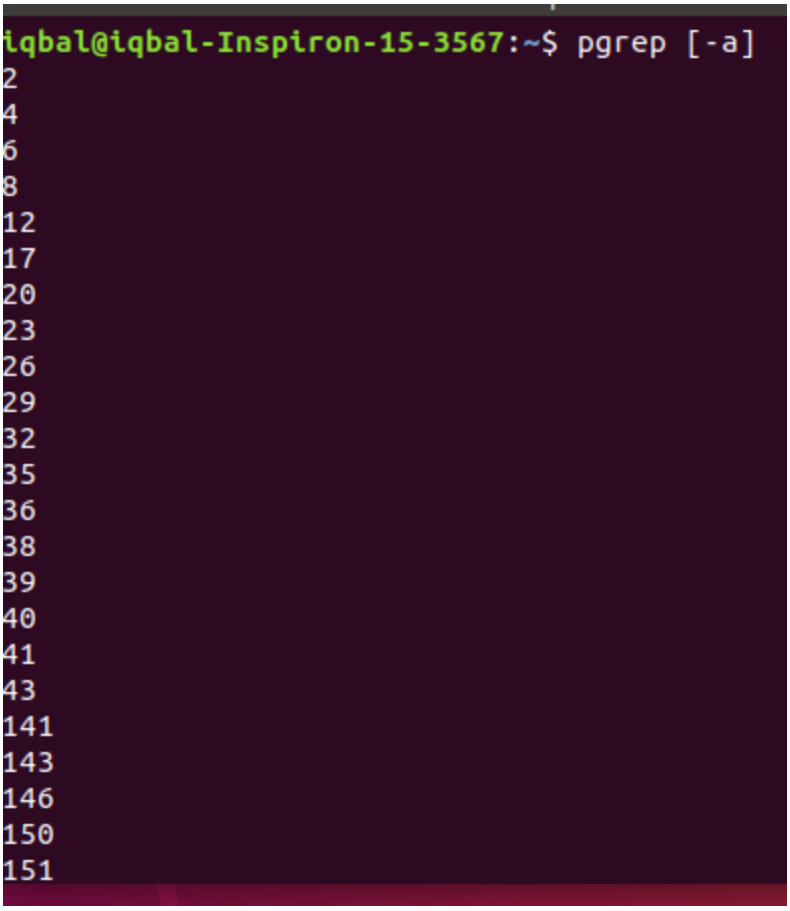
5) pstree: The **pstree** command is another way of visualizing processes. It displays them in tree format.

6)kill : "kill [-l]"command List the single names.If arguments follow '-l' they are assumed to be signal numbers for which names should be listed.



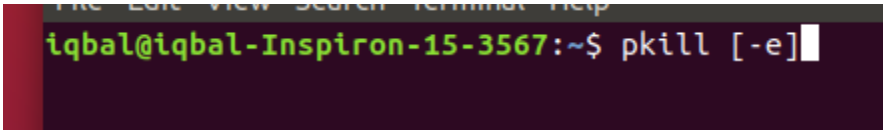
7) pgrep:

pgrep [-a] : This command list PID and full command line.



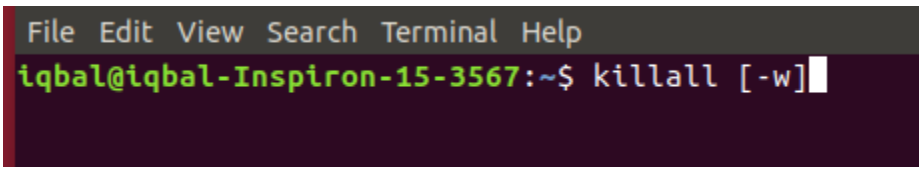
8) pkill :

pkill [-e] : Display what is killed.



9)killall:

killall [-w] :Wait for process to die.



10) “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.