

Lab-Report

Report No: 06

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission: 11/09/2020

Submitted by

Name: Shakhera khanom

ID: IT-18033

3rd year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No :06

Experiment Name: Linux command for process.

Objectives:

- What How to Manage Processes from the Linux Terminal?
- Run the following process commands in Linux.
Top, htop, Ps, pstree, kill, pgrep, pkill, killall, renice, xkill,

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

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.

```
shakhera@shakhera-HP-Notebook-PC: ~
File Edit View Search Terminal Help
top - 03:19:17 up 1:07, 1 user, load average: 0.30, 0.16, 0.07
Tasks: 227 total, 1 running, 187 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1390180 total, 87640 free, 1042428 used, 260112 buff/cache
KiB Swap: 1942896 total, 1749360 free, 193536 used. 187504 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1717 shakhera  20   0  361136   6376   5332  S   0.0   0.5   0:00.04 gvfsd-trash
 1724 shakhera  20   0  197780    6184   5536  S   0.0   0.4   0:00.93 ibus-engin+
 1756 shakhera  20   0 1128760  48524   9536  S   0.0   3.5   0:00.52 evolution-+
 1779 shakhera  20   0  725580  13200   9832  S   0.0   0.9   0:00.06 evolution-+
 1790 shakhera  20   0  862620  12900   9452  S   0.0   0.9   0:00.07 evolution-+
 1808 shakhera  20   0  197384   5668   5040  S   0.0   0.4   0:00.02 gvfsd-meta+
 1836 shakhera  20   0  663412  23676  18424  S   0.0   1.7   0:00.35 update-not+
 1839 shakhera  20   0 1303788 140028  24068  S   0.0  10.1   0:05.04 gnome-soft+
 1862 root       20   0  563524  13868  11012  S   0.0   1.0   0:00.25 fwupd
 1951 shakhera  20   0  790024  33176  27160  S   0.0   2.4   0:00.14 deja-dup-m+
 2011 shakhera  20   0  426344   8480   7196  S   0.0   0.6   0:00.03 zeitgeist-+
 2017 shakhera  20   0  326700  15944  13544  S   0.0   1.1   0:00.07 zeitgeist-+
 2048 root       20   0         0         0         0  I   0.0   0.0   0:00.00 kworker/1:+
 2142 root       20   0         0         0         0  I   0.0   0.0   0:00.18 kworker/u4+
 2146 root       20   0         0         0         0  I   0.0   0.0   0:00.78 kworker/0:+
 2192 shakhera  20   0   22440   4568   3300  S   0.0   0.3   0:00.04 bash
 2202 root       20   0         0         0         0  I   0.0   0.0   0:00.05 kworker/u4+
```

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

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
```

```
shakhera@shakhera-HP-Notebook-PC: ~
File Edit View Search Terminal Help

1  [||| 2.0%] Tasks: 145, 365 thr; 1 running
2  [||| 2.6%] Load average: 1.25 1.15 0.61
Mem[|||||||||||||||||906M/1.33G] Uptime: 00:06:19
Swp[||||| 268M/1.85G]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
2630 shakhera    20   0 26952  4116  3320 R   2.6  0.3   0:00.15 htop
1417 shakhera    20   0 3449M  223M  40396 S   2.0 16.5   0:12.29 /usr/bin/gnome-sh
1423 shakhera    20   0 3449M  223M  40396 S   0.7 16.5   0:00.97 /usr/bin/gnome-sh
1104 gdm         20   0  708M   8236  2908 S   0.7  0.6   0:00.30 /usr/lib/gnome-se
1424 shakhera    20   0 3449M  223M  40396 S   0.0 16.5   0:01.09 /usr/bin/gnome-sh
1009 gdm         20   0 3342M  96860 12044 S   0.0  7.0   0:07.25 /usr/bin/gnome-sh
1231 shakhera    20   0  685M  54980 18240 S   0.0  4.0   0:02.42 /usr/lib/xorg/Xor
 865 mysql      20   0 1328M   572   488 S   0.0  0.0   0:00.02 /usr/sbin/mysqld
 828 mysql      20   0 1328M   572   488 S   0.0  0.0   0:00.95 /usr/sbin/mysqld
1238 shakhera    20   0  685M  54980 18240 S   0.0  4.0   0:00.09 /usr/lib/xorg/Xor
1785 shakhera    20   0  773M 17880   8836 S   0.0  1.3   0:01.26 /usr/lib/gnome-te
1373 shakhera    20   0  181M   380     0 S   0.0  0.0   0:00.59 /usr/bin/VBoxClie
1358 shakhera    20   0  181M   380     0 S   0.0  0.0   0:00.59 /usr/bin/VBoxClie
1449 shakhera    20   0 3449M  223M  40396 S   0.0 16.5   0:00.02 /usr/bin/gnome-sh
   1 root        20   0  220M   3624  1744 S   0.0  0.3   0:01.96 /sbin/init splash
 232 root        19  -1 95256  6152  5492 S   0.0  0.4   0:00.45 /lib/systemd/syst
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

htop displays the same information with an easier-to-understand layout. It also lets you select processes with the arrow keys and perform actions.

ps: The **ps** command lists running processes. The following command lists all processes running on your system:

```
ps -A
```

```
shakhera@shakhera-HP-Notebook-PC: ~  
File Edit View Search Terminal Help  
shakhera@shakhera-HP-Notebook-PC:~$ ps  
  PID TTY          TIME CMD  
 3055 pts/2    00:00:00 bash  
 3064 pts/2    00:00:00 ps  
shakhera@shakhera-HP-Notebook-PC:~$ ps -A  
  PID TTY          TIME CMD  
    1 ?           00:00:02 systemd  
    2 ?           00:00:00 kthreadd  
    3 ?           00:00:00 rcu_gp  
    4 ?           00:00:00 rcu_par_gp  
    6 ?           00:00:00 kworker/0:0H-kb  
    7 ?           00:00:00 kworker/0:1-eve  
    9 ?           00:00:00 mm_percpu_wq  
   10 ?           00:00:00 ksoftirqd/0  
   11 ?           00:00:00 rcu_sched  
   12 ?           00:00:00 migration/0  
   13 ?           00:00:00 idle_inject/0  
   14 ?           00:00:00 cpuhp/0  
   15 ?           00:00:00 cpuhp/1  
   16 ?           00:00:00 idle_inject/1  
   17 ?           00:00:00 migration/1  
   18 ?           00:00:00 ksoftirqd/1  
   20 ?           00:00:00 kworker/1:0H-kb  
   21 ?           00:00:00 kdevtmpfs
```

This may be too many processes to read at one time, so you can pipe the output through the **less** command to scroll through them at your own pace

```
ps -A | less
```

```
shakhera@shakhera-HP-Notebook-PC: ~
File Edit View Search Terminal Help
PID TTY      TIME CMD
  1 ?        00:00:02 systemd
  2 ?        00:00:00 kthreadd
  3 ?        00:00:00 rcu_gp
  4 ?        00:00:00 rcu_par_gp
  6 ?        00:00:00 kworker/0:0H-kb
  7 ?        00:00:00 kworker/0:1-eve
  9 ?        00:00:00 mm_percpu_wq
 10 ?        00:00:00 ksoftirqd/0
 11 ?        00:00:00 rcu_sched
 12 ?        00:00:00 migration/0
 13 ?        00:00:00 idle_inject/0
 14 ?        00:00:00 cpuhp/0
 15 ?        00:00:00 cpuhp/1
 16 ?        00:00:00 idle_inject/1
 17 ?        00:00:00 migration/1
 18 ?        00:00:00 ksoftirqd/1
 20 ?        00:00:00 kworker/1:0H-kb
 21 ?        00:00:00 kdevtmpfs
 22 ?        00:00:00 netns
 23 ?        00:00:00 rcu_tasks_kthre
 24 ?        00:00:00 kauditd
 25 ?        00:00:00 khungtaskd
:
```

Press **q** to exit when you're done.

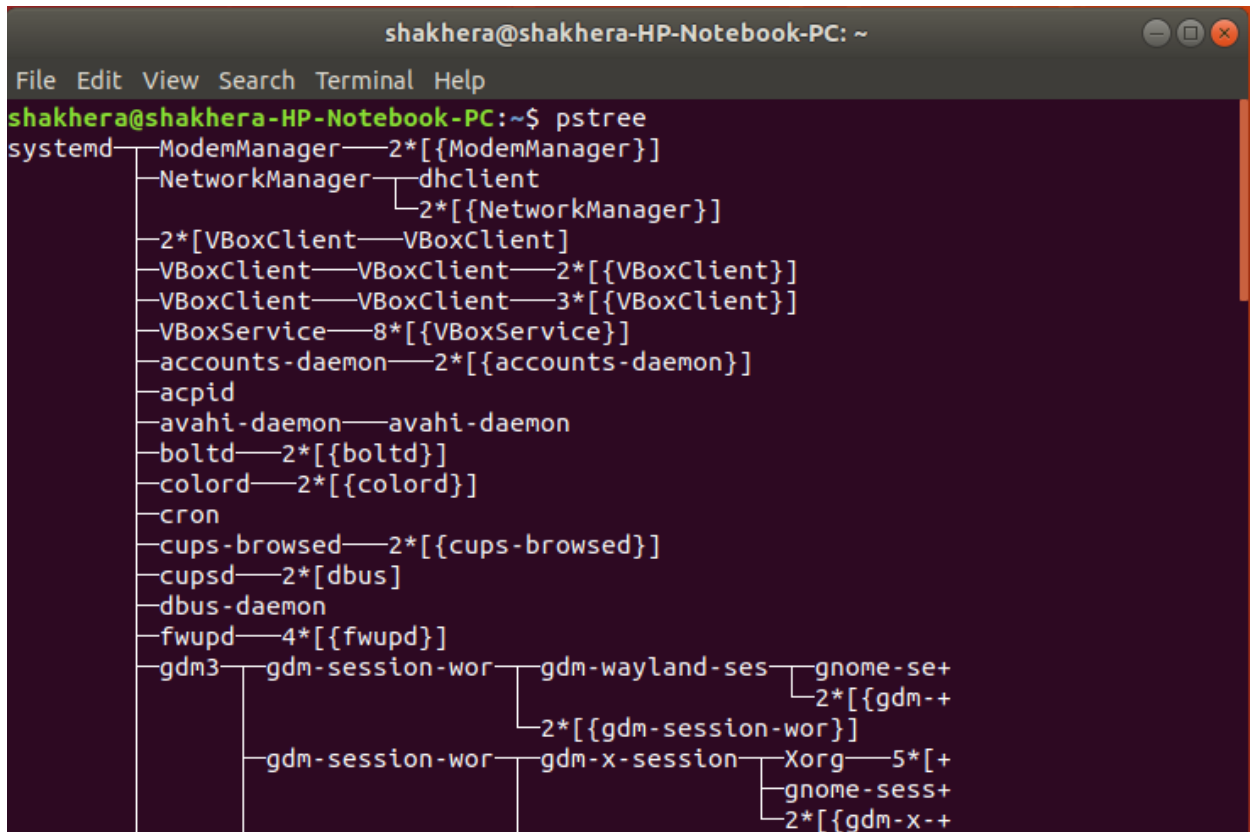
You could also pipe the output through **grep** to search for a specific process without using any other commands. The following command would search for the Firefox process:

```
ps -A | grep firefox
```

```
shakhera@shakhera-HP-Notebook-PC: ~
File Edit View Search Terminal Help
shakhera@shakhera-HP-Notebook-PC:~$ ps -A|less
shakhera@shakhera-HP-Notebook-PC:~$ ps aux | grep firefox
shakhera 1975  0.0  0.0 14424  956 pts/0    S+   09:30   0:00 grep --color=au
to firefox
```

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.

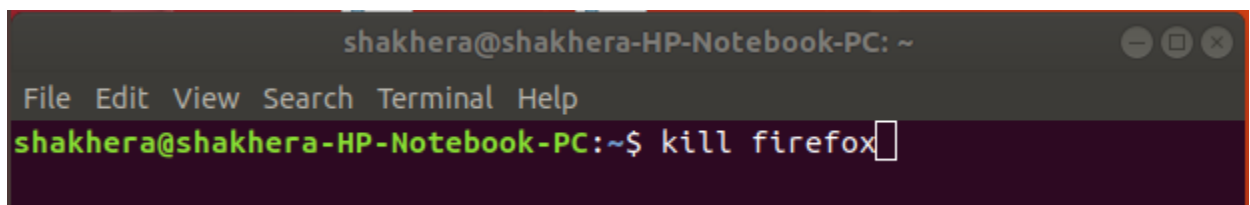
pstree



```
shakhera@shakhera-HP-Notebook-PC: ~  
File Edit View Search Terminal Help  
shakhera@shakhera-HP-Notebook-PC:~$ pstree  
systemd--ModemManager--2*[{ModemManager}]  
--NetworkManager--dhclient  
--2*[{NetworkManager}]  
--2*[VBoxClient--VBoxClient]  
--VBoxClient--VBoxClient--2*[{VBoxClient}]  
--VBoxClient--VBoxClient--3*[{VBoxClient}]  
--VBoxService--8*[{VBoxService}]  
--accounts-daemon--2*[{accounts-daemon}]  
--acpid  
--avahi-daemon--avahi-daemon  
--boltd--2*[{boltd}]  
--colord--2*[{colord}]  
--cron  
--cups-browsed--2*[{cups-browsed}]  
--cupsd--2*[dbus]  
--dbus-daemon  
--fwupd--4*[{fwupd}]  
--gdm3--gdm-session-wor--gdm-wayland-ses--gnome-se+  
--2*[{gdm-session-wor}]  
--gdm-session-wor--gdm-x-session--Xorg--5* [+  
--gnome-sess+  
--2*[{gdm-x-+
```

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

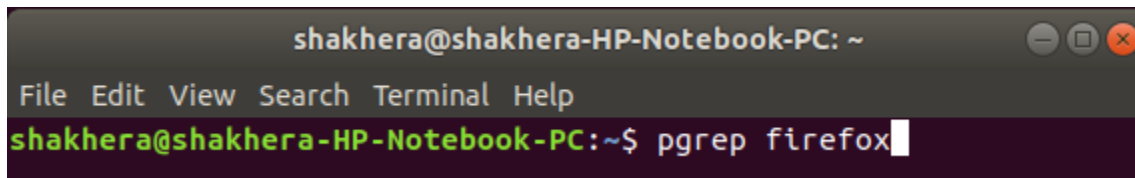


```
shakhera@shakhera-HP-Notebook-PC: ~  
File Edit View Search Terminal Help  
shakhera@shakhera-HP-Notebook-PC:~$ kill firefox
```

Technically speaking, the **kill** command can send any signal to a process. You can use **kill -KILL** or **kill -9** instead to kill a stubborn process.

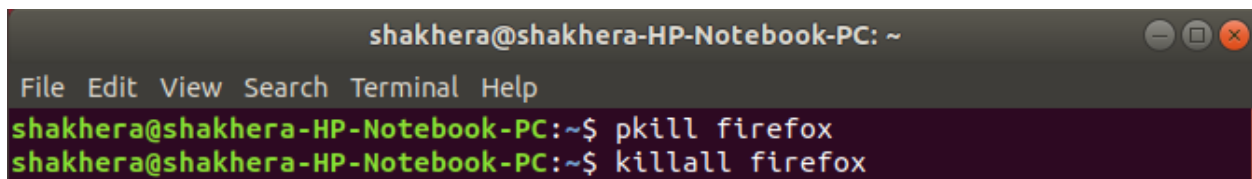
pgrep: Given a search term, **pgrep** returns the process IDs that match it. For example, you could use the following command to find Firefox's PID:

```
pgrep firefox
```

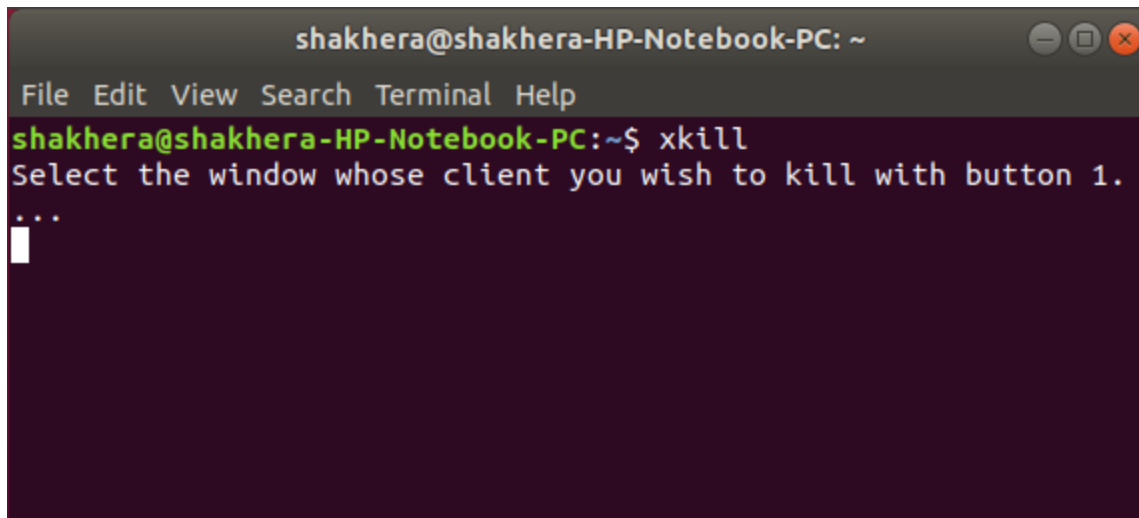
A terminal window titled 'shakhera@shakhera-HP-Notebook-PC: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'shakhera@shakhera-HP-Notebook-PC:~\$' and the command 'pgrep firefox' is entered, followed by a cursor.

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

```
pkill firefox  
killall firefox
```

A terminal window titled 'shakhera@shakhera-HP-Notebook-PC: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'shakhera@shakhera-HP-Notebook-PC:~\$' and the commands 'pkill firefox' and 'killall firefox' are entered on separate lines.

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.

A terminal window titled 'shakhera@shakhera-HP-Notebook-PC: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'shakhera@shakhera-HP-Notebook-PC:~\$' and the command 'xkill' has been entered. The output is 'Select the window whose client you wish to kill with button 1.' followed by three dots and a cursor.

```
shakhera@shakhera-HP-Notebook-PC: ~
File Edit View Search Terminal Help
shakhera@shakhera-HP-Notebook-PC:~$ xkill
Select the window whose client you wish to kill with button 1.
...
█
```

You don't have to run this command from a terminal — you can also press Alt-F2, type **xkill** and press Enter to use it from a graphical desktop.

Discussion: This lab help to learn Linux command for process. Linux processes carry out various tasks/jobs within the Linux distribution. Since Linux is a multiprocessing operating system, one can run multiple tasks in the background. Hence it is essential to know how to show all running processes in Linux.