



# **Mawlana Bhashani Science and Technology University**

## **Lab -Report**

Report No:06

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:19-08-2020

Date of Submission:26-08-2020

### **Submitted by**

Name: Ruku Shikder  
ID: IT-18057  
3<sup>th</sup> year 1<sup>st</sup> 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.

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

### Manage Processes from the linux Terminal:

The procedure to monitor the running process in Linux using the command line is as follows:

- Open the terminal window on Linux
- For remote Linux server use the ssh command for log in purpose
- Type the ps aux command to see all running process in Linux
- Alternatively, you can issue the top command or htop command to view running process in Linux

**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.

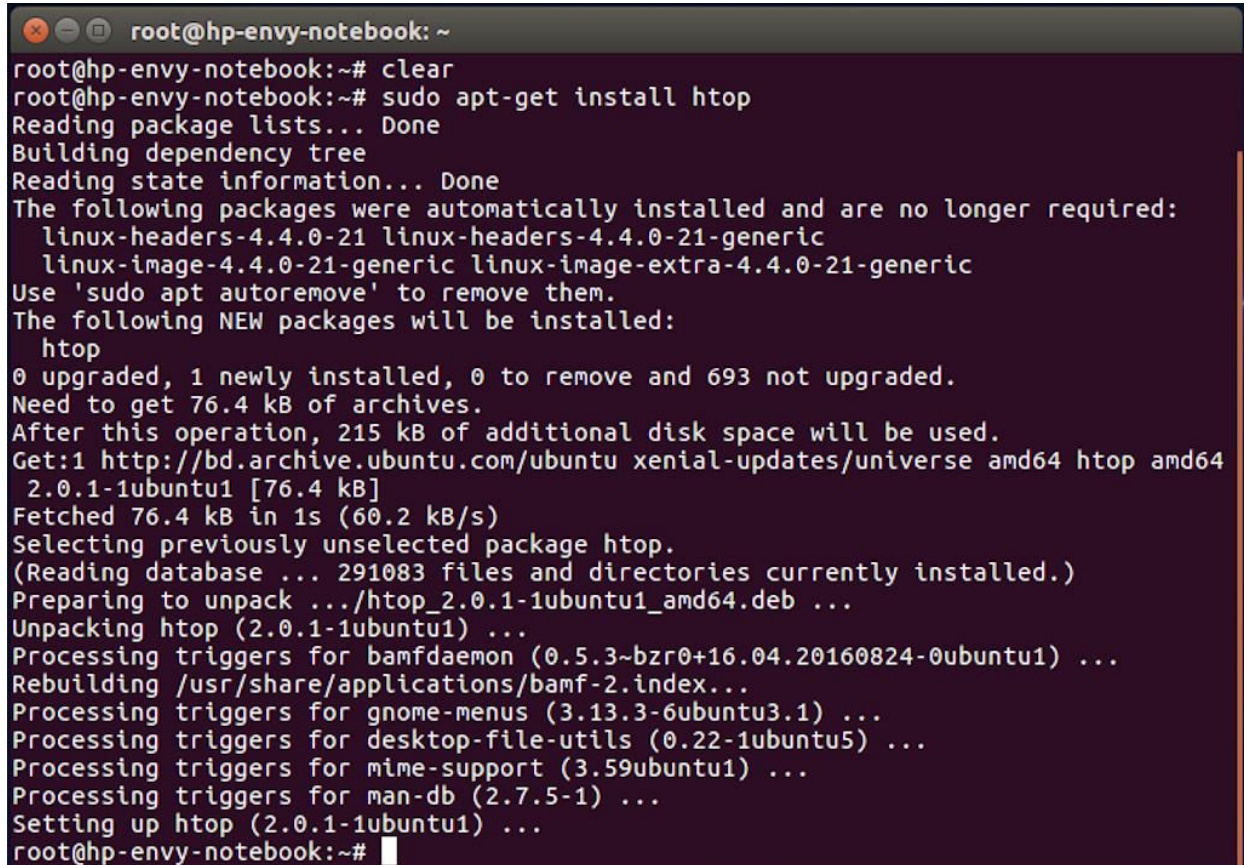
```
ruku@hp-envy-notebook: ~
top - 01:08:44 up 1:18, 2 users, load average: 0.86, 0.49, 0.23
Tasks: 278 total, 1 running, 277 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.5 us, 0.9 sy, 0.0 ni, 97.6 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8071012 total, 4541344 free, 1331024 used, 2198644 buff/cache
KiB Swap: 3998716 total, 3998716 free, 0 used. 5760228 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 6737 root        20   0   382516   73636   62876 S   11.3   0.9   0:22.79 Xorg
 7317 ruku        20   0  1298028   68260   49592 S    4.6   0.8   0:05.62 nautilus
 7151 ruku        20   0   565528   34144   25836 S    3.3   0.4   0:01.34 unity-pane+
 7253 ruku        20   0  1518780  116540   64068 S    2.6   1.4   0:21.88 compiz
 9586 ruku        20   0   657064   35672   28428 S    0.7   0.4   0:00.38 gnome-term+
  846 avahi      20   0    44912    3492    3100 S    0.3   0.0   0:00.40 avahi-daem+
 6594 mysql      20   0  1378648  167892   18188 S    0.3   2.1   0:04.70 mysqld
 7057 ruku        20   0   418964    9068    5464 S    0.3   0.1   0:01.07 ibus-daemon
    1 root        20   0   185228    5960    4084 S    0.0   0.1   0:02.01 systemd
    2 root        20   0         0         0        0 S    0.0   0.0   0:00.00 kthreadd
    3 root        20   0         0         0        0 S    0.0   0.0   0:00.02 ksoftirqd/0
    5 root        0 -20         0         0        0 S    0.0   0.0   0:00.00 kworker/0:0+
    7 root        20   0         0         0        0 S    0.0   0.0   0:02.46 rcu_sched
    8 root        20   0         0         0        0 S    0.0   0.0   0:00.00 rcu_bh
    9 root        rt    0         0         0        0 S    0.0   0.0   0:00.00 migration/0
   10 root        rt    0         0         0        0 S    0.0   0.0   0:00.03 watchdog/0
   11 root        rt    0         0         0        0 S    0.0   0.0   0:00.03 watchdog/1
```

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`

A terminal window titled 'root@hp-envy-notebook: ~' with a dark background and light text. The user has entered the command 'sudo apt-get install htop'. The terminal output shows the package manager's response, including reading package lists, building a dependency tree, and listing packages to be installed. It indicates that htop will be installed along with some other packages that are no longer required. The disk space requirements are shown as 76.4 kB. The source of the package is identified as 'http://bd.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 htop amd64 2.0.1-1ubuntu1 [76.4 kB]'. The package is then fetched and unpacked, and the terminal shows the processing of various triggers for other installed packages like bamfdaemon, gnome-menus, desktop-file-utils, mime-support, and man-db. Finally, htop is set up, and the prompt returns to the user.

```
root@hp-envy-notebook: ~
root@hp-envy-notebook:~# clear
root@hp-envy-notebook:~# sudo apt-get install htop
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.4.0-21 linux-headers-4.4.0-21-generic
  linux-image-4.4.0-21-generic linux-image-extra-4.4.0-21-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  htop
0 upgraded, 1 newly installed, 0 to remove and 693 not upgraded.
Need to get 76.4 kB of archives.
After this operation, 215 kB of additional disk space will be used.
Get:1 http://bd.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 htop amd64
  2.0.1-1ubuntu1 [76.4 kB]
Fetched 76.4 kB in 1s (60.2 kB/s)
Selecting previously unselected package htop.
(Reading database ... 291083 files and directories currently installed.)
Preparing to unpack .../htop_2.0.1-1ubuntu1_amd64.deb ...
Unpacking htop (2.0.1-1ubuntu1) ...
Processing triggers for bamfdaemon (0.5.3~bZR0+16.04.20160824-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5) ...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up htop (2.0.1-1ubuntu1) ...
root@hp-envy-notebook:~#
```

```
root@hp-envy-notebook: ~

1 [ 0.0%] Tasks: 188, 438 thr; 1 running
2 [ 0.0%] Load average: 0.18 0.30 0.25
3 [ 0.0%] Uptime: 01:35:35
4 [|||||||||||||||||100.0%]
Mem[|||||||||||||1.98G/7.70G]
Swp[0K/3.81G]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
11301 root         20    0 26172  4040  3204  R 100.  0.1   0:00.05 htop
   1 root         20    0  180M  5960  4084  S   0.0  0.1   0:02.17 /sbin/init splash
 218 root         20    0 35508  4916  4372  S   0.0  0.1   0:00.37 /lib/systemd/syst
 246 root         20    0 46204  5352  3016  S   0.0  0.1   0:01.12 /lib/systemd/syst
 468 systemd-t    20    0   99M  2568  2344  S   0.0  0.0   0:00.00 /lib/systemd/syst
 446 systemd-t    20    0   99M  2568  2344  S   0.0  0.0   0:00.01 /lib/systemd/syst
 804 root         20    0  269M  6440  5628  S   0.0  0.1   0:00.18 /usr/lib/accounts
 830 root         20    0  269M  6440  5628  S   0.0  0.1   0:00.12 /usr/lib/accounts
 732 root         20    0  269M  6440  5628  S   0.0  0.1   0:00.45 /usr/lib/accounts
 828 root         20    0  329M 10332  6680  S   0.0  0.1   0:00.00 /usr/sbin/ModemMa
 831 root         20    0  329M 10332  6680  S   0.0  0.1   0:00.01 /usr/sbin/ModemMa
 741 root         20    0  329M 10332  6680  S   0.0  0.1   0:00.02 /usr/sbin/ModemMa
 744 root         20    0 31956  4296  3944  S   0.0  0.1   0:00.01 /usr/lib/bluetoot
 960 whoopsie      20    0  365M 14212 10560  S   0.0  0.2   0:00.00 /usr/bin/whoopsie
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`

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ ps  
  PID TTY          TIME CMD  
 9646 pts/1        00:00:00 bash  
 9673 pts/1        00:00:00 ps  
ruku@hp-envy-notebook:~$ ps -A  
  PID TTY          TIME CMD  
   1 ?            00:00:02 systemd  
   2 ?            00:00:00 kthreadd  
   3 ?            00:00:00 ksoftirqd/0  
   5 ?            00:00:00 kworker/0:0H  
   7 ?            00:00:02 rcu_sched  
   8 ?            00:00:00 rcu_bh  
   9 ?            00:00:00 migration/0  
  10 ?            00:00:00 watchdog/0  
  11 ?            00:00:00 watchdog/1  
  12 ?            00:00:00 migration/1  
  13 ?            00:00:00 ksoftirqd/1  
  15 ?            00:00:00 kworker/1:0H  
  16 ?            00:00:00 watchdog/2  
  17 ?            00:00:00 migration/2  
  18 ?            00:00:00 ksoftirqd/2  
  20 ?            00:00:00 kworker/2:0H  
  21 ?            00:00:00 watchdog/3  
  22 ?            00:00:00 migration/3  
  23 ?            00:00:00 ksoftirqd/3  
  25 ?            00:00:00 kworker/3:0H  
  26 ?            00:00:00 kdevtmpfs  
  27 ?            00:00:00 netns  
  28 ?            00:00:00 perf  
  29 ?            00:00:00 khungtaskd  
  30 ?            00:00:00 writeback  
  31 ?            00:00:00 ksmd  
  32 ?            00:00:00 khugepaged  
  33 ?            00:00:00 crypto  
  34 ?            00:00:00 kintegrityd  
  35 ?            00:00:00 bioset  
  36 ?            00:00:00 kblockd
```

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`

```
ruku@hp-envy-notebook: ~  
PID TTY          TIME CMD  
  1 ?             00:00:02 systemd  
  2 ?             00:00:00 kthreadd  
  3 ?             00:00:00 ksoftirqd/0  
  5 ?             00:00:00 kworker/0:0H  
  7 ?             00:00:02 rcu_sched  
  8 ?             00:00:00 rcu_bh  
  9 ?             00:00:00 migration/0  
 10 ?            00:00:00 watchdog/0  
 11 ?            00:00:00 watchdog/1  
 12 ?            00:00:00 migration/1  
 13 ?            00:00:00 ksoftirqd/1  
 15 ?            00:00:00 kworker/1:0H  
 16 ?            00:00:00 watchdog/2  
 17 ?            00:00:00 migration/2  
 18 ?            00:00:00 ksoftirqd/2  
 20 ?            00:00:00 kworker/2:0H  
 21 ?            00:00:00 watchdog/3  
 22 ?            00:00:00 migration/3  
 23 ?            00:00:00 ksoftirqd/3  
 25 ?            00:00:00 kworker/3:0H  
 26 ?            00:00:00 kdevtmpfs  
 27 ?            00:00:00 netns  
 28 ?            00:00:00 perf  
 29 ?            00:00:00 khungtaskd  
 30 ?            00:00:00 writeback  
 31 ?            00:00:00 ksm  
:  
█
```

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`

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ ps -A|grep firefox  
ruku@hp-envy-notebook:~$ ps -aux|grep firefox  
ruku      9775  0.0  0.0 14224  964 pts/1    S+   01:13   0:00 grep --color=au  
to firefox  
ruku@hp-envy-notebook:~$ █
```

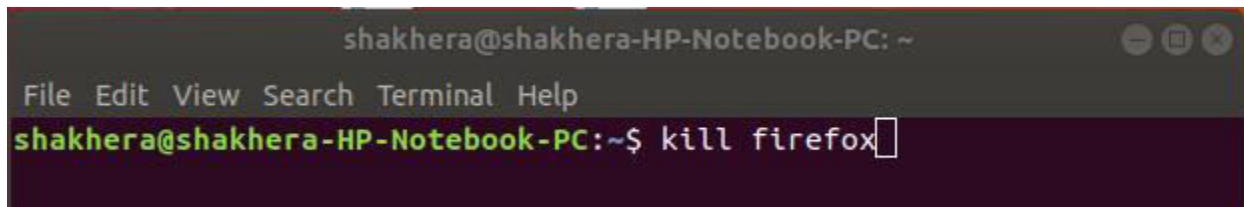
**ps tree:** The **ps tree** command is another way of visualizing processes. It displays them in tree format. So, for example, your X server and graphical environment would appear under the display manager that spawned them.

## ps tree

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ pstree  
systemd--ModemManager--{gdbus}  
--{gmain}  
--NetworkManager--2*[dhclient]  
--dnsmasq  
--{gdbus}  
--{gmain}  
--accounts-daemon--{gdbus}  
--{gmain}  
--acpid  
--agetty  
--avahi-daemon--avahi-daemon  
--bluetoothd  
--colord--{gdbus}  
--{gmain}  
--cron  
--cups-browsed--{gdbus}  
--{gmain}  
--dbus-daemon  
--fwupd--3*[GUSBEventThread]  
--{fwupd}  
--{gdbus}  
--{gmain}  
--2*[gnome-keyring-d--{gdbus}]  
--{gmain}]  
--{timer}]  
--irqbalance  
--lightdm--2*[Xorg--{Xorg}]  
--lightdm--lightdm--upstart--at-spi-bus-laun--dbus-daemon  
--{dconf worker}  
--{gdbus}
```

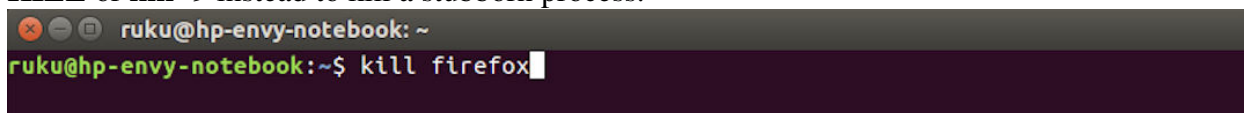
**kill:** The **kill** command can kill a process, given its process ID. You can get this information from the **ps -A**, **top** or **pgrep** commands.

## kill PID

A terminal window titled 'shakhera@shakhera-HP-Notebook-PC: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'kill firefox' is entered at the prompt.

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

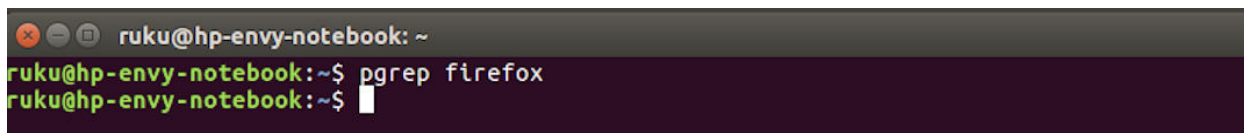
Technically speaking, the **kill** command can send any signal to a process. You can use **kill -KILL** or **kill -9** instead to kill a stubborn process.

A terminal window titled 'ruku@hp-envy-notebook: ~' with standard window controls. The command 'kill firefox' is entered at the prompt.

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ kill firefox
```

**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 'ruku@hp-envy-notebook: ~' with standard window controls. The command 'pgrep firefox' is entered at the prompt.

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ pgrep firefox  
ruku@hp-envy-notebook:~$
```

**pkill & killall:** The **pkill** and **killall** commands can kill a process, given its name.

Use either command to kill Firefox:

`pkill firefox`

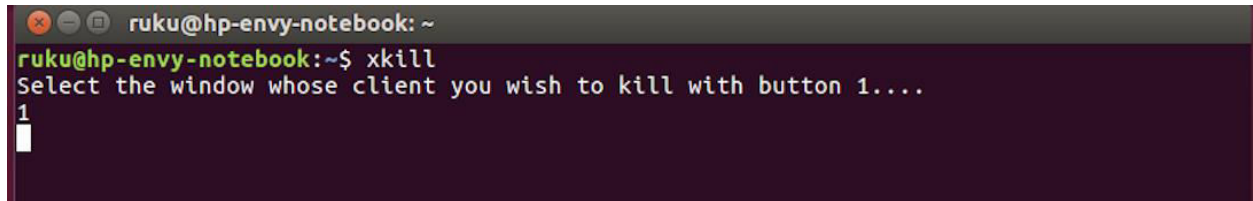
`killall firefox`

A terminal window titled 'ruku@hp-envy-notebook: ~' with standard window controls. The commands 'pkill firefox' and 'killall firefox' are entered at the prompt.

```
ruku@hp-envy-notebook: ~  
ruku@hp-envy-notebook:~$ pkill firefox  
ruku@hp-envy-notebook:~$ killall firefox
```



**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 with a dark background. The title bar shows window control icons and the text 'ruku@hp-envy-notebook: ~'. The prompt is 'ruku@hp-envy-notebook:~\$'. The command 'xkill' has been entered. Below the command, a message says 'Select the window whose client you wish to kill with button 1....'. The number '1' has been entered, and a white cursor is visible on the line.

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:

In this lab, I will walk through a basic understanding of processes and briefly look at how to manage processes in Linux using certain commands.

A process refers to a program in execution; it's a running instance of a program. It is made up of the program instruction, data read from files, other programs or input from a system user. There are fundamentally two types of processes in Linux-Foreground processes (also referred to as interactive processes) – these are initialized and controlled through a terminal session. In other words, there has to be a user connected to the system to start such processes; they haven't started automatically as part of the system functions/services. Background processes (also referred to as non-interactive/automatic processes) – are processes not connected to a terminal; they don't expect any user input.

