Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 06

Lab Report Name: Linux command for processes .

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 15-09-2020

Date of Submission:

Submitted by

Name: Md Masud Rana

ID: IT-18002

3rd year 1st semester

Session: 2017-18

Dept. of ICT.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Experiment No: 06

Experiment Name: Linux command for processes.

Theory: 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.

1) **ps:** The ps command is used to view currently running processes on the system. It helps us to determine which process is doing what in our system, how much memory it is using, how much CPU space it occupies, user ID, command name, etc .

The ps command may display different results for different systems because it displays information about the currently running process of a system

```
masud@masud-VirtualBox:~$ ps
  PID TTY          TIME CMD
21398 pts/0    00:00:00 bash
22231 pts/0    00:00:00 ps
masud@masud-VirtualBox:~$ ps -A
  PID TTY          TIME CMD
    1 ?        00:00:11 systemd
    2 ?        00:00:00 kthreadd
    4 ?        00:00:00 kworker/0:0H
    6 ?        00:00:00 mm_percpu_wq
    7 ?        00:00:02 ksoftirqd/0
    8 ?        00:00:04 rcu_sched
    9 ?        00:00:00 rcu_bh
   10 ?        00:00:00 migration/0
   11 ?        00:00:00 watchdog/0
   12 ?        00:00:00 cpuhp/0
   13 ?        00:00:00 kdevtmpfs
   14 ?        00:00:00 netns
   15 ?        00:00:00 rcu_tasks_kthre
   16 ?        00:00:00 kauditd
   17 ?        00:00:00 khungtaskd
   18 ?        00:00:00 oom_reaper
   19 ?        00:00:00 writeback
   20 ?        00:00:00 kcompactd0
   21 ?        00:00:00 ksmd
   22 ?        00:00:00 khugepaged
   23 ?        00:00:00 crypto
   24 ?        00:00:00 kintegrityd
   25 ?        00:00:00 kblockd
```

2) **ps –ef/ps –aux:** To display all currently running processes in full format on a system two types of commands are used.

ps -ef

```
masud@masud-VirtualBox:~$ ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 05:07 ?        00:00:12 /sbin/init splash
root           2     0  0 05:07 ?        00:00:00 [kthreadd]
root           4     2  0 05:07 ?        00:00:00 [kworker/0:0H]
root           6     2  0 05:07 ?        00:00:00 [mm_percpu_wq]
root           7     2  0 05:07 ?        00:00:02 [ksoftirqd/0]
root           8     2  0 05:07 ?        00:00:05 [rcu_sched]
root           9     2  0 05:07 ?        00:00:00 [rcu_bh]
root          10     2  0 05:07 ?        00:00:00 [migration/0]
root          11     2  0 05:07 ?        00:00:00 [watchdog/0]
root          12     2  0 05:07 ?        00:00:00 [cpuhp/0]
root          13     2  0 05:07 ?        00:00:00 [kdevtmpfs]
root          14     2  0 05:07 ?        00:00:00 [netns]
root          15     2  0 05:07 ?        00:00:00 [rcu_tasks_kthre]
root          16     2  0 05:07 ?        00:00:00 [kauditd]
root          17     2  0 05:07 ?        00:00:00 [khungtaskd]
root          18     2  0 05:07 ?        00:00:00 [oom_reaper]
root          19     2  0 05:07 ?        00:00:00 [writeback]
root          20     2  0 05:07 ?        00:00:00 [kcompactd0]
root          21     2  0 05:07 ?        00:00:00 [ksmd]
```

ps -aux

```
masud@masud-VirtualBox:~$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2 160188  9304 ?        Ss   05:07   0:12 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    05:07   0:00 [kthreadd]
root           4  0.0  0.0      0     0 ?        I<   05:07   0:00 [kworker/0:0H]
root           6  0.0  0.0      0     0 ?        I<   05:07   0:00 [mm_percpu_wq]
root           7  0.0  0.0      0     0 ?        S    05:07   0:02 [ksoftirqd/0]
root           8  0.0  0.0      0     0 ?        I    05:07   0:05 [rcu_sched]
root           9  0.0  0.0      0     0 ?        I    05:07   0:00 [rcu_bh]
root          10  0.0  0.0      0     0 ?        S    05:07   0:00 [migration/0]
root          11  0.0  0.0      0     0 ?        S    05:07   0:00 [watchdog/0]
root          12  0.0  0.0      0     0 ?        S    05:07   0:00 [cpuhp/0]
root          13  0.0  0.0      0     0 ?        S    05:07   0:00 [kdevtmpfs]
root          14  0.0  0.0      0     0 ?        I<   05:07   0:00 [netns]
root          15  0.0  0.0      0     0 ?        S    05:07   0:00 [rcu_tasks_kthre]
root          16  0.0  0.0      0     0 ?        S    05:07   0:00 [kauditd]
root          17  0.0  0.0      0     0 ?        S    05:07   0:00 [khungtaskd]
root          18  0.0  0.0      0     0 ?        S    05:07   0:00 [oom_reaper]
```

3) **ps –eo:** If you want to list different types of information to check who is logged in to your system, use **eo** option. Where **e** show all the processes and **o** controls the output.

```
masud@masud-VirtualBox:~$ ps -eo pid,user
  PID USER
    1 root
    2 root
    4 root
    6 root
    7 root
    8 root
    9 root
   10 root
   11 root
   12 root
```

4) **pstree:** In normal ps command we have to look manually on PID and PPID number to know the relation between processes. In hierarchial format, child processes are shown under the parent process which makes it easy for us to look upon.

```
masud@masud-VirtualBox:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager─┬─dhclient
        │                └─2*[{NetworkManager}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─apt.systemd.dai───apt.systemd.dai───unattended-upgr─┬─2*[http]
        │                                                     └─2*[{unattended-upgr}]
        ├─avahi-daemon───avahi-daemon
        ├─boltd───2*[{boltd}]
        ├─colord───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd───4*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b─┬─gnome-shell─┬─Xwayland
```

5)**top:** The top command displays all the running process within the environment of your system. It helps in monitoring system usage and performances. It is mainly used to detect load on the server by system administrators.

```
top - 21:18:47 up 16:10,  1 user,  load average: 0.23, 0.15, 0.18
Tasks: 214 total,   1 running, 182 sleeping,   0 stopped,   0 zombie
%Cpu(s): 11.3 us,  0.7 sy,  0.0 ni, 88.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  4039732 total,   129408 free,  1844664 used,  2065660 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used.  1890820 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1197 masud     20   0 3157528 561188  56560 S 10.3 13.9  48:23.50 gnome-shell
 1054 masud     20   0  617532 239448  29796 S  1.3  5.9   6:13.87 Xorg
21388 masud     20   0  802944  37928  28104 S  0.7  0.9   0:02.72 gnome-terminal-
22770 masud     20   0   51664   4356   3600 R  0.3  0.1   0:00.20 top
    1 root      20   0  160188   9304   6504 S  0.0  0.2   0:12.43 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0H
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
    7 root      20   0       0      0      0 S  0.0  0.0   0:02.74 ksoftirqd/0
    8 root      20   0       0      0      0 I  0.0  0.0   0:05.08 rcu_sched
    9 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_bh
   10 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
```

6)**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:

```
  CPU[||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||1.82G/3.85G]
  Swp[                                                                                 0K/2.00G]

  PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
23552 root        39  19  217M  116M 73936 R 74.0  3.0  0:18.09 /usr/bin/python3 /usr/bin/unattended
 1197 masud       20   0 3083M  543M 52628 S 21.4 13.8 48:59.18 /usr/bin/gnome-shell
 1054 masud       20   0  603M  232M 28828 S  2.6  5.9  6:19.30 /usr/lib/xorg/Xorg vt2 -displayfd 3
23489 masud       20   0 41688  5348  3788 R  1.3  0.1  0:00.33 htop
21388 masud       20   0  784M 37772 27280 S  0.6  0.9  0:04.16 /usr/lib/gnome-terminal/gnome-termin
23417 masud       20   0 1152M  188M 39756 S  0.6  4.8  0:01.27 /usr/bin/gnome-software --gapplicati
 1388 masud       20   0 1152M  188M 39756 S  0.0  4.8  0:50.36 /usr/bin/gnome-software --gapplicati
21566 root        20   0  683M 36192 14476 S  0.0  0.9 10:57.72 /usr/lib/snapd/snapd
 1060 masud       20   0  603M  232M 28828 S  0.0  5.9  0:16.13 /usr/lib/xorg/Xorg vt2 -displayfd 3
    1 root        20   0  156M  9304  6504 S  0.0  0.2  0:12.52 /sbin/init splash
  888 gdm         20   0 2858M  162M 49028 S  0.0  4.1  0:00.61 /usr/bin/gnome-shell
```

7)**pgrep**: pgrep is a command-line utility that allows you to find the process IDs of a running program based on given criteria.

```
masud@masud-VirtualBox:~$ pgrep -u masud
1034
1048
1052
1054
1059
1063
1156
1159
1164
1173
```

8) **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.

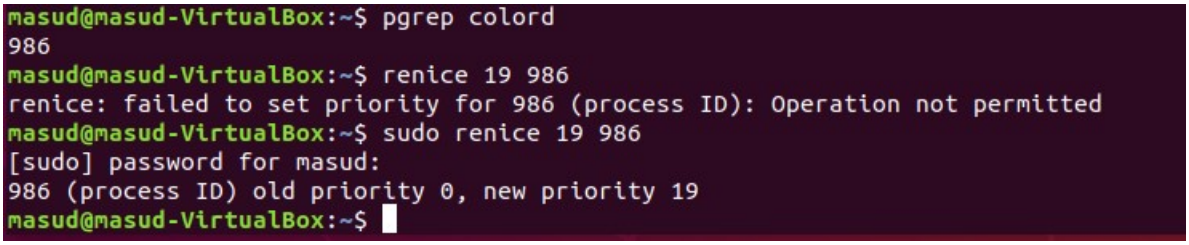
```
masud@masud-VirtualBox:~$ kill mysqld
bash: kill: mysqld: arguments must be process or job IDs
masud@masud-VirtualBox:~$ ps -A | grep mysqld
22081 ?        00:00:32 mysqld
masud@masud-VirtualBox:~$ sudo kill 22081
[sudo] password for masud:
masud@masud-VirtualBox:~$
```

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

```
masud@masud-VirtualBox:~$ sudo killall cupsd
masud@masud-VirtualBox:~$ pkill cupsd
masud@masud-VirtualBox:~$
```

10) **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 19 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:



11)**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.

Discussion: From this lab we learn how we can manage processes from the linux terminal.For this we run some process commands such as top,htop,ps,pgrep,kill,pkill,killall etc.I have faced problem in kill and killall commands.In stead of kill,killall commands i have used sudo kill and sudo killall commands.