Process

As we all know Linux is a multitasking and multi-user systems. So, it allows multiple processes to operate simultaneously without interfering with each other. Process is one of the important fundamental concept of the Linux OS. A process is an executing instance of a program and carry out different tasks within the operating system.

Linux provides us a utility called **ps** for viewing information related with the processes on a system which stands as abbreviation for **"Process Status".** ps command is used to list the currently running processes and their PIDs along with some other information depends on different options. It reads the process information from the virtual files in **/proc** file-system. /proc contains virtual files, this is the reason it's referred as a virtual file system.

ps provides numerous options for manipulating the output according to our need.

Syntax –

```
ps [options]
```

Options for ps Command :

Simple process selection : Shows the processes for the current shell –

```
[root@rhel7 ~]# ps
  PID TTY          TIME CMD
12330 pts/0    00:00:00 bash
21621 pts/0    00:00:00 ps
```

Result contains four columns of information.
Where,
PID – the unique process ID
TTY – terminal type that the user is logged into
TIME – amount of CPU in minutes and seconds that the process has been running
CMD – name of the command that launched the process.

Note – Sometimes when we execute **ps** command, it shows TIME as 00:00:00. It is nothing but the total accumulated CPU utilization time for any process and 00:00:00 indicates no CPU time has been given by the kernel till now. In above example we found that, for bash no CPU time has been given. This is because bash is just a parent process for different processes which needs bash for their execution and bash itself is not utilizing any CPU time till now.

View Processes : View all the running processes use either of the following option with ps –

```
[root@rhel7 ~]# ps -A
[root@rhel7 ~]# ps -e
```

View Processes not associated with a terminal : View all processes except both session leaders and processes not associated with a terminal.

```
[root@rhel7 ~]# ps -a
  PID TTY          TIME CMD
27011 pts/0    00:00:00 man
27016 pts/0    00:00:00 less
27499 pts/1    00:00:00 ps
```

> **Note –** You may be thinking that what is session leader? A unique session is assigned to every process group. So, session leader is a process which kicks off other processes. The process ID of first process of any session is similar as the session ID.

View all the processes except session leaders :

```
[root@rhel7 ~]# ps -d
```

View all processes except those that fulfill the specified conditions (negates the selection) :
Example – If you want to see only session leader and processes not associated with a terminal. Then, run

```
[root@rhel7 ~]# ps -a -N
OR
[root@rhel7 ~]# ps -a --deselect
```

View all processes associated with this terminal :

```
[root@rhel7 ~]# ps -T
```

View all the running processes :

```
[root@rhel7 ~]# ps -r
```

View all processes owned by you : Processes i.e same EUID as ps which means runner of the ps command, root in this case –

```
[root@rhel7 ~]# ps -x
```

Process selection by list

Here we will discuss how to get the specific processes list with the help of ps command. These options accept a single argument in the form of a blank-separated or comma-separated list. They can be used multiple times.
For example: ps -p “1 2” -p 3,4

Select the process by the command name. This selects the processes whose executable name is given in cmdlist. There may be a chance you won’t know the process ID and with this command it is easier to search.

Syntax : ps -C command_name

```
Syntax :
ps -C command_name

Example :
[root@rhel7 ~]# ps -C dhclient
  PID TTY          TIME CMD
19805 ?        00:00:00 dhclient
```

Select by group ID or name. The group ID identifies the group of the user who created the process.

```
Syntax :
ps -G group_name
ps --Group group_name

Example :
[root@rhel7 ~]# ps -G root
```

View by group id :

```
Syntax :
ps -g group_id
ps -group group_id

Example :
[root@rhel7 ~]# ps -g 1
  PID TTY          TIME CMD
    1 ?        00:00:13 systemd
```

View process by process ID.

```
Syntax :
ps p process_id
ps -p process_id
ps --pid process_id

Example :
[root@rhel7 ~]#  ps p 27223
  PID TTY      STAT   TIME COMMAND
27223 ?        Ss     0:01 sshd: root@pts/2

[root@rhel7 ~]#  ps -p 27223
  PID TTY          TIME CMD
27223 ?        00:00:01 sshd

[root@rhel7 ~]#  ps --pid 27223
  PID TTY          TIME CMD
27223 ?        00:00:01 sshd
```

You can view multiple processes by specifying multiple process IDs separated by blank or comma –
Example :

```
[root@rhel7 ~]#  ps -p 1 904 27223
PID TTY      STAT   TIME COMMAND
```

```
        1 ?        Ss     0:13 /usr/lib/systemd/systemd --
        switched-root --system --d
    904 tty1     Ssl+   1:02 /usr/bin/X -core -noreset :0 -seat seat0
    -auth /var/r
    27223 ?        Ss     0:01 sshd: root@pts/2
```

Here, we mentioned three process IDs – 1, 904 and 27223 which are separated by blank.

Select by parent process ID. By using this command we can view all the processes owned by parent process except the parent process.

```
[root@rhel7 ~]# ps -p 766
  PID TTY          TIME CMD
  766 ?        00:00:06 NetworkManager

[root@rhel7 ~]# ps --ppid 766
  PID TTY          TIME CMD
19805 ?        00:00:00 dhclient
```

In above example process ID **766** is assigned to NetworkManager and this is the parent process for dhclient with process ID 19805.

View all the processes belongs to any session ID.

```
Syntax :
ps -s session_id
ps --sid session_id

Example :
[root@rhel7 ~]# ps -s 1248
  PID TTY          TIME CMD
 1248 ?        00:00:00 dbus-daemon
 1276 ?        00:00:00 dconf-service
 1302 ?        00:00:00 gvfsd
 1310 ?        00:00:00 gvfsd-fuse
 1369 ?        00:00:00 gvfs-udisks2-vo
 1400 ?        00:00:00 gvfsd-trash
 1418 ?        00:00:00 gvfs-mtp-volume
 1432 ?        00:00:00 gvfs-gphoto2-vo
 1437 ?        00:00:00 gvfs-afc-volume
 1447 ?        00:00:00 wnck-applet
 1453 ?        00:00:00 notification-ar
 1454 ?        00:00:02 clock-applet
```

Select by tty. This selects the processes associated with the mentioned tty :

```
Syntax :
ps t tty
ps -t tty
ps --tty tty

Example :
[root@rhel7 ~]# ps -t pts/0
  PID TTY          TIME CMD
31199 pts/0    00:00:00 bash
31275 pts/0    00:00:00 man
```

```
31280 pts/0    00:00:00 less
```

Select by effective user ID or name.
Syntax :
ps U user_name/ID
ps -U user_name/ID
ps -u user_name/ID
ps –User user_name/ID
ps –user user_name/ID

Output Format control

These options are used to choose the information displayed by ps. There are multiple options to control output format. These option can be combined with any other options like **e, u, p, G, g** etc, depends on our need.

Use **-f** to view full-format listing.

```
[tux@rhel7 ~]$ ps -af
tux      17327 17326  0 12:42 pts/0    00:00:00 -bash
tux      17918 17327  0 12:50 pts/0    00:00:00 ps -af
```

Use **-F** to view Extra full format.

```
[tux@rhel7 ~]$ ps -F
UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME
CMD
tux      17327 17326  0 28848  2040   0 12:42 pts/0    00:00:00 -
bash
tux      17942 17327  0 37766  1784   0 12:50 pts/0    00:00:00 ps
-F
```

To view process according to user-defined format.

```
Syntax :
[root@rhel7 ~]#  ps --formate column_name
[root@rhel7 ~]#  ps -o column_name
[root@rhel7 ~]#  ps o column_name

Example :
[root@rhel7 ~]#  ps -aN --format cmd,pid,user,ppid
CMD                          PID USER      PPID
/usr/lib/systemd/systemd --    1 root         0
[kthreadd]                     2 root         0
[ksoftirqd/0]                  3 root         2
[kworker/0:0H]                 5 root         2
[migration/0]                  7 root         2
[rcu_bh]                       8 root         2
[rcu_sched]                    9 root         2
[watchdog/0]                  10 root         2
```

In this example I wish to see command, process ID, username and parent process ID, so I pass the arguments cmd, pid, user and ppid respectively.

View in BSD job control format :

```
[root@rhel7 ~]# ps -j
  PID  PGID   SID TTY          TIME CMD
16373 16373 16373 pts/0    00:00:00 bash
19734 19734 16373 pts/0    00:00:00 ps
```

Display BSD long format :

```
[root@rhel7 ~]# ps l
F   UID   PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY
TIME COMMAND
4     0   904   826  20   0 306560 51456 ep_pol Ssl+ tty1
1:32 /usr/bin/X -core -noreset :0 -seat seat0 -auth
/var/run/lightdm/root/:0 -noli
4     0 11692 11680  20   0 115524  2132 do_wai Ss   pts/2
0:00 -bash
```

Add a column of security data.

```
[root@rhel7 ~]# ps -aM
LABEL                                                PID  TTY
TIME    CMD
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 19534 pts/2
00:00:00 man
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 19543 pts/2
00:00:00 less
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 20469 pts/0
00:00:00 ps
```

View command with signal format.

```
[root@rhel7 ~]# ps s 766
```

Display user-oriented format

```
[root@rhel7 ~]# ps u 1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME
COMMAND
root         1  0.0  0.6 128168  6844 ?        Ss   Apr08   0:16
/usr/lib/systemd/systemd --switched-root --system --deserialize 21
```

Display virtual memory format

```
[root@rhel7 ~]# ps v 1
  PID TTY      STAT   TIME  MAJFL   TRS   DRS   RSS %MEM COMMAND
    1 ?        Ss     0:16     62  1317 126850 6844  0.6
/usr/lib/systemd/systemd --switched-root --system --deserialize 21
```

If you want to see environment of any command. Then use option **e** –

```
[root@rhel7 ~]# ps ev 766
  PID TTY      STAT   TIME  MAJFL   TRS   DRS   RSS %MEM COMMAND
  766 ?        Ssl    0:08     47  2441 545694 10448  1.0
/usr/sbin/NetworkManager --no-daemon LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
```

View processes using highest memory.

```
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem
```

12 – print a process tree

```
[root@rhel7 ~]# ps --forest -C sshd
PID TTY          TIME CMD
797 ?        00:00:00 sshd
11680 ?        00:00:03  \_ sshd
16361 ?        00:00:02  \_ sshd
```

List all threads for a particular process. Use either the **-T or -L** option to display threads of a process.

```
[root@rhel7 ~]# ps -C sshd -L
  PID   LWP TTY          TIME CMD
  797   797 ?        00:00:00 sshd
11680 11680 ?        00:00:03 sshd
16361 16361 ?        00:00:02 sshd
```

Note – For the explanation of different column contents refer **man page.**