WileyNXT

mthree

Linux Processes

4.1 What is a Process?

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.

Types of Processes:

- Foreground Processes: They run on the screen and need input from the user. For example LibreOffice Programs.
- Background Processes: They run in the background and usually do not need user input. For example Antivirus.

The simplest way to start a background process is to add an **ampersand (&)** at the end of the command.

```
File  Edit  View  Search  Terminal  Help
[binayak@localhost Desktop]$ touch f1.txt f2.txt f3.txt f4.txt f5.txt
[binayak@localhost Desktop]$ ls
f1.txt  f2  f2.txt  f3.txt  f4.txt  f5.txt  file3.txt
[binayak@localhost Desktop]$ ls *.txt
f1.txt  f2.txt  f3.txt  f4.txt  f5.txt  file3.txt
[binayak@localhost Desktop]$ ls *.txt &
[1] 5370
f1.txt  f2.txt  f3.txt  f4.txt  f5.txt  file3.txt
[1]+  Done                    ls --color=auto *.txt
[binayak@localhost Desktop]$
```

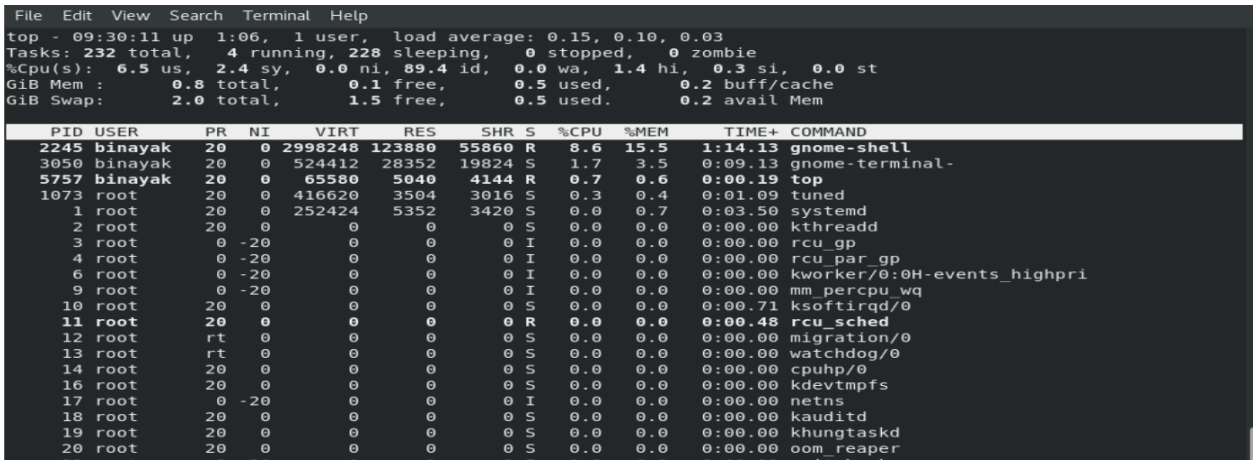
- **ps** command is used to print a snapshot of current running processes.

```
[binayak@localhost Desktop]$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:03 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-events_highpri
      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 watchdog/0
     14 ?        00:00:00 cpuhp/0
     16 ?        00:00:00 kdevtmpfs
     17 ?        00:00:00 netns
```

```
[binayak@localhost Desktop]$ ps -ely
S   UID     PID    PPID  C PRI  NI    RSS    SZ WCHAN  TTY          TIME CMD
S     0       1       0  0  80   0   5360 63106 -      ?        00:00:03 systemd
S     0       2       0  0  80   0      0     0 -      ?        00:00:00 kthreadd
I     0       3       2  0  60 -20      0     0 -      ?        00:00:00 rcu_gp
I     0       4       2  0  60 -20      0     0 -      ?        00:00:00 rcu_par_gp
I     0       6       2  0  60 -20      0     0 -      ?        00:00:00 kworker/0:0H-events_highpri
I     0       9       2  0  60 -20      0     0 -      ?        00:00:00 mm_percpu_wq
S     0      10       2  0  80   0      0     0 -      ?        00:00:00 ksoftirqd/0
R     0      11       2  0  80   0      0     0 -      ?        00:00:00 rcu_sched
S     0      12       2  0 -40   -      0     0 -      ?        00:00:00 migration/0
S     0      13       2  0 -40   -      0     0 -      ?        00:00:00 watchdog/0
S     0      14       2  0  80   0      0     0 -      ?        00:00:00 cpuhp/0
S     0      16       2  0  80   0      0     0 -      ?        00:00:00 kdevtmpfs
I     0      17       2  0  60 -20      0     0 -      ?        00:00:00 netns
S     0      18       2  0  80   0      0     0 -      ?        00:00:00 kauditd
S     0      19       2  0  80   0      0     0 -      ?        00:00:00 khungtaskd
S     0      20       2  0  80   0      0     0 -      ?        00:00:00 oom_reaper
I     0      21       2  0  60 -20      0     0 -      ?        00:00:00 writeback
S     0      22       2  0  80   0      0     0 -      ?        00:00:00 kcompactd0
S     0      23       2  0  85   5      0     0 -      ?        00:00:00 ksmd
```

```
[binayak@localhost Desktop]$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.6 252424  5352 ?        Ss   08:24   0:03 /usr/lib/systemd/systemd --switched-root --sy
root           2  0.0  0.0      0     0 ?        S    08:24   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   08:24   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   08:24   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   08:24   0:00 [kworker/0:0H-events_highpri]
root           9  0.0  0.0      0     0 ?        I<   08:24   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    08:24   0:00 [ksoftirqd/0]
root          11  0.0  0.0      0     0 ?        I    08:24   0:00 [rcu_sched]
root          12  0.0  0.0      0     0 ?        S    08:24   0:00 [migration/0]
root          13  0.0  0.0      0     0 ?        S    08:24   0:00 [watchdog/0]
root          14  0.0  0.0      0     0 ?        S    08:24   0:00 [cpuhp/0]
root          16  0.0  0.0      0     0 ?        S    08:24   0:00 [kdevtmpfs]
root          17  0.0  0.0      0     0 ?        I<   08:24   0:00 [netns]
root          18  0.0  0.0      0     0 ?        S    08:24   0:00 [kauditd]
root          19  0.0  0.0      0     0 ?        S    08:24   0:00 [khungtaskd]
```

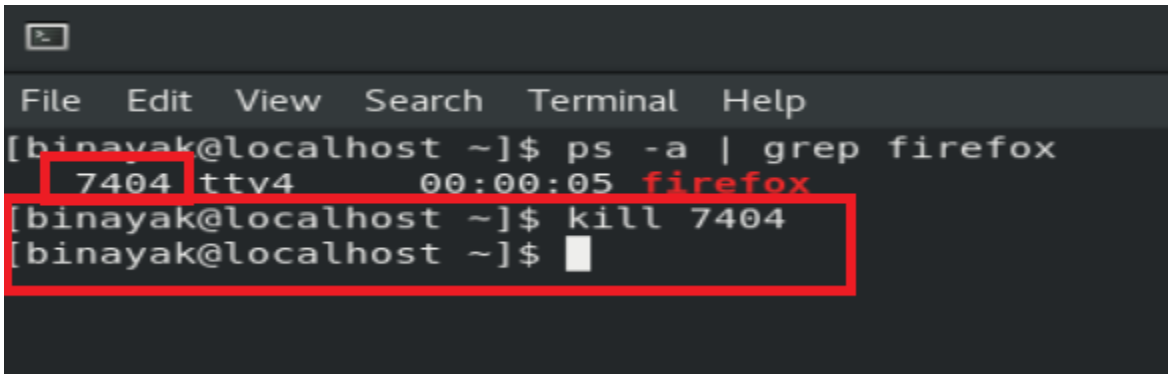
- **top** command is the utility which tells the user about all the running processes on the Linux machine.



Tip: use Shift+ E to change Memory, Swap usage between various units of storage.

- **Kill command is used to terminate running processes on a linux machine**.

- Start mozilla firefox and search its process ID using ps -a | grep firefox
- Use kill command for terminating firefox.

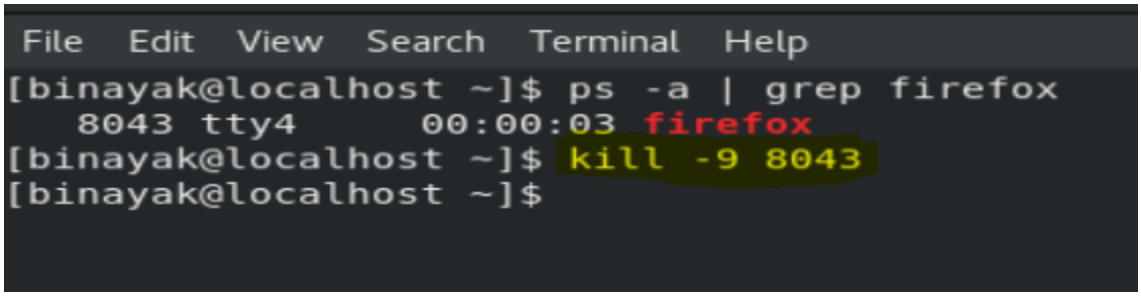


- Signal Number is a non-negative decimal integer, specifying the signal to be sent instead of the default .
- Signal Name is a symbolic signal name specifying the signal to be sent instead of the default.

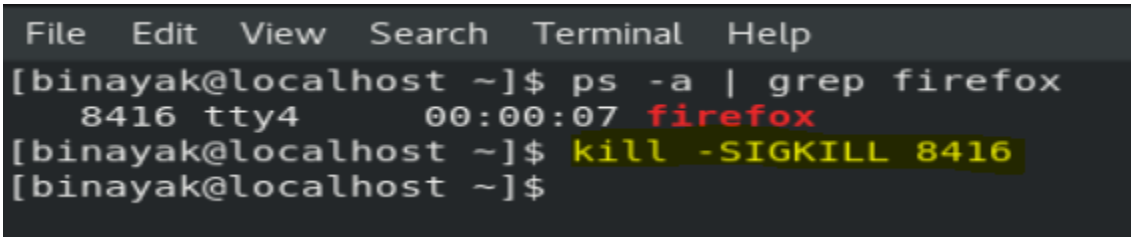
Common Linux and UNIX signal names and numbers

Signal Name	Signal Number	Description
SIGHUP	1	Hang up detected on controlling terminal or death of controlling process
SIGINT	2	Issued if the user sends an interrupt signal (Ctrl + C)
SIGQUIT	3	Issued if the user sends a quit signal (Ctrl + D)
SIGFPE	8	Issued if an illegal mathematical operation is attempted
SIGKILL	9	If a process gets this signal it must quit immediately and will not perform any clean-up operations
SIGALRM	14	Alarm clock signal (used for timers)
SIGTERM	15	Software termination signal (sent by kill by default)

- Killing a process using signal number.



- Killing the process using signal name.



- The nice command lets you execute a program/process with modified scheduling priority, the renice command allows you to change the scheduling priority of an already running process.

,

- ## Zombie and Orphan Processes

Normally, when a child process is killed, the parent process process is updated via a SIGCHLD signal.Then the parent can do some other task or restart a new child as needed. needed. However, sometimes the parent process is killed before its child is killed. killed. In this case, the "parent "parent of all processes," processes,"The init process, process, becomes the new PPID (parent (parent process process ID). In some cases, these processes are called orphan processes. When a process is killed, a ps listing may still show the process process with a Z state. This is a zombie or defunct defunct process. process. The process is dead and not being used. These processes are different from the orphan processes. processes. They have completed execution but still find an entry in the process table.

- ## Daemon Processes

Daemons Daemons are system-related system-related background background processes processes that often run with the permissions permissions of root and services requests from other processes. A daemon has no controlling terminal. terminal. It cannot open /dev/tty. If you do a "ps -ef" and look at the tty field, all daemons will have a ? for the tty. To be precise, precise, a daemon is a process process that runs in the background, background, usually usually waiting waiting for something something to happen that it is capable capable of working working with. For example, a printer printer daemon waiting waiting for print commands.If you have a program that calls for lengthy lengthy processing, then it’s worth it to make it a daemon and run it in the background.

- ## Job ID Versus Process ID

Background Background and suspended processes are usually usually manipulated via job number (job ID). This number is different from the process ID and is used because it is shorter. In addition, a job can consist of multiple processes running in a series or at the same time, in parallel. Using the job ID is easier than tracking individual processes.