Chapter Process

///Commands

To see the PID assigned to your shell

```
root@ubuntu:~# echo $$
4420
```

```
root@ubuntu:~/Desktop/shell_scripts# ps -p $$
   PID TTY          TIME CMD
  4420 pts/4    00:00:00 bash
```

Check if the file is executable

ls -l or file command

```
root@ubuntu:~# file /bin/ls
/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/l, for GNU/Linux 2.6.32,
BuildID[sha1]=d0bc0fb9b3f60f72bbad3c5a1d24c9e2a1fde775, stripped
root@ubuntu:~#
```

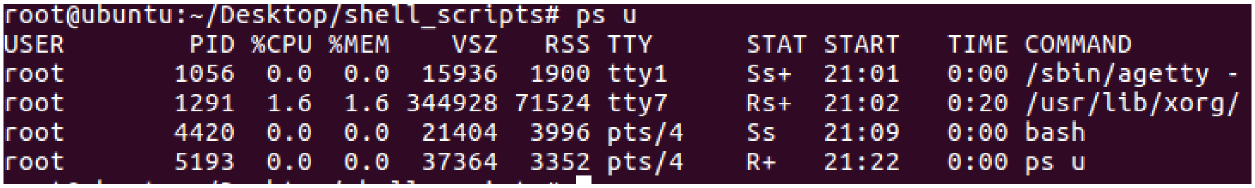
What processes are running?

```
root@ubuntu:~/Desktop/shell_scripts# ps
   PID TTY          TIME CMD
  4420 pts/4    00:00:00 bash
  5148 pts/4    00:00:00 ps
root@ubuntu:~/Desktop/shell_scripts#
```

*****Process states********

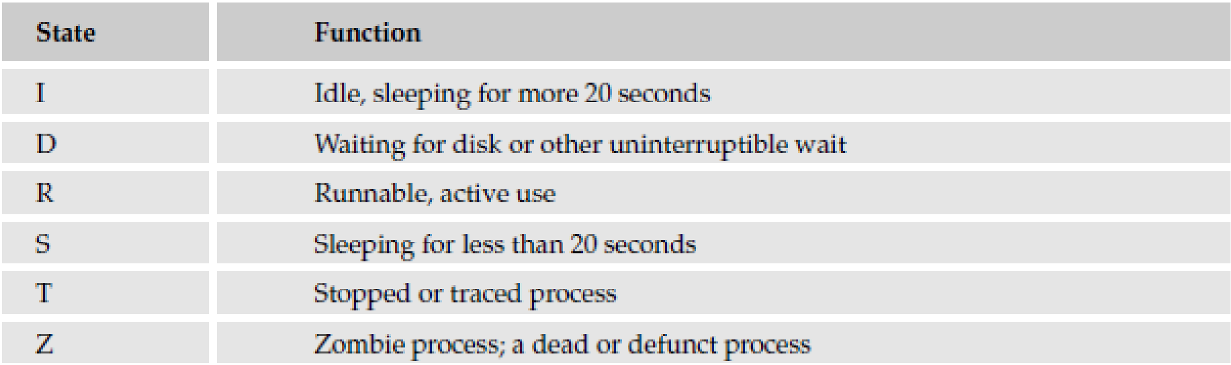
```
root@ubuntu:~/Desktop/shell_scripts# ps u
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1056  0.0  0.0  15936  1900 tty1     Ss+  21:01   0:00 /sbin/agetty -
root      1291  1.6  1.6 344928 71524 tty7     Rs+  21:02   0:20 /usr/lib/xorg/
root      4420  0.0  0.0  21404  3996 pts/4    Ss   21:09   0:00 bash
root      5193  0.0  0.0  37364  3352 pts/4    R+   21:22   0:00 ps u
```

- It reports the user of the running process
- the process ID,
- the percentage of the CPU the process has been using over the past minute,
- the percentage of the real memory,
- the virtual memory size in kilobytes,
- the physical memory used,
- the terminal it is connected to, the states,
- when the process was started, the amount of CPU time used by process (since it was started), and the command name



```
root@ubuntu:~/Desktop/shell_scripts# ps u
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1056  0.0  0.0  15936  1900 tty1     Ss+  21:01   0:00 /sbin/agetty -
root       1291  1.6  1.6 344928 71524 tty7     Rs+  21:02   0:20 /usr/lib/xorg/
root       4420  0.0  0.0  21404  3996 pts/4    Ss   21:09   0:00 bash
root       5193  0.0  0.0  37364  3352 pts/4    R+   21:22   0:00 ps u
```

States of Process



State	Function
I	Idle, sleeping for more 20 seconds
D	Waiting for disk or other uninterruptible wait
R	Runnable, active use
S	Sleeping for less than 20 seconds
T	Stopped or traced process
Z	Zombie process; a dead or defunct process

S --> Interruptible sleep(S) process which is waiting for an event to complete.

Ss --> A session leader(s) process in Interruptible sleep(S).

SN --> A low priority(N) process which is nice to others is in interruptible sleep(S).

S< s --> "<" indicates a high priority process and the process is a session leader(s) in interruptible sleep(S).

S< sl --> A multhi-threaded(l) high priority(<) session leader(s) process in interruptible sleep(S).

S< --> A high priroty(<) process in interruptible sleep(S).

SNl --> A low priority(N) multi-threaded(l) process in interruptible sleep(S) state.

SNs --> A low priority(N) session leader(s) process in interruptible sleep(S).

SN+ --> A low priority(N) interruptible sleep(S) process running in foreground(+).

R+ --> A foreground(+) running(R) process.

The a argument for ps causes it to report information about processes for all users. The x argument tells ps to display information about processes without a controlling terminal. System processes are programs running behind the scenes handling many essential maintenance aspects for your system. ***Normally, system processes do not have a TTY (teletype) in use. Many of these processes are often called daemons, and they do routine work. This is noted by a question mark in the TTY field.***

System process:

```
root@ubuntu:~/Desktop/shell_scripts# ps ax

   PID TTY      STAT   TIME COMMAND

     1 ?        Ss     0:07 /sbin/init auto noprompt

     2 ?        S      0:00 [kthreadd]

     4 ?        S<     0:00 [kworker/0:0H]

     6 ?        S<     0:00 [mm_percpu_wq]
```

How many processes running?

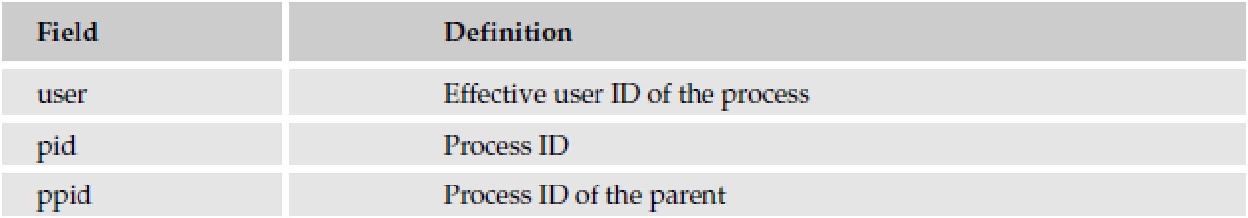
```
root@ubuntu:~/Desktop/shell_scripts# ps ax | wc -l

222
```

[Total -heading 1]

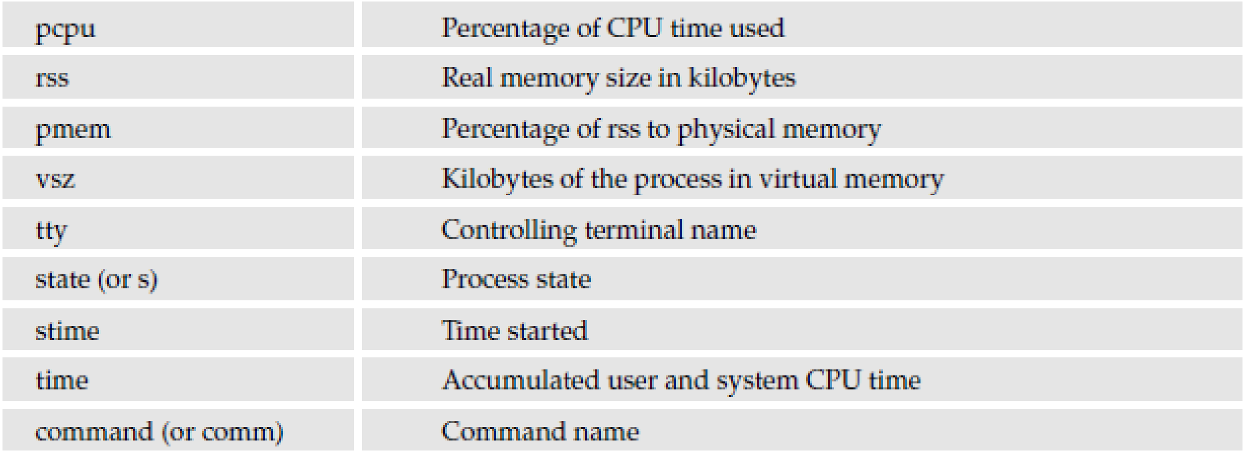
To check Process Attributes: To see parent process id . Use switch -o

```
root@ubuntu:~/Desktop/shell_scripts# ps -o user,pid,ppid,comm

USER        PID   PPID COMMAND
```

```
root      4420   4413 bash

root      5395   4420 ps
```



Field	Definition
user	Effective user ID of the process
pid	Process ID
ppid	Process ID of the parent



pcpu	Percentage of CPU time used
rss	Real memory size in kilobytes
pmem	Percentage of rss to physical memory
vsz	Kilobytes of the process in virtual memory
tty	Controlling terminal name
state (or s)	Process state
stime	Time started
time	Accumulated user and system CPU time
command (or comm)	Command name

Terminate a process

CTRL + C keystroke ends the process and exits

Technically, the kill command does not kill a command, but sends a special signal (SIGTERM- to *terminate*) to the process. Signals are used for simple communication between processes.

To give SIGTERM signal to the process ID 5432
kill 5432

What are the signals available - root@ubuntu:~/Desktop/shell_scripts# kill –l

```
 1) SIGHUP     2) SIGINT      3) SIGQUIT    4) SIGILL       5) SIGTRAP
 6) SIGABRT    7) SIGBUS      8) SIGFPE     9) SIGKILL    10) SIGUSR1
11) SIGSEGV  12) SIGUSR2  13) SIGPIPE   14) SIGALRM 15) SIGTERM
16) SIGSTKFLT         17) SIGCHLD 18) SIGCONT 19) SIGSTOP  20) SIGTSTP
21) SIGTTIN   22) SIGTTOU 23) SIGURG    24) SIGXCPU  25) SIGXFSZ
26) SIGVTALRM        27) SIGPROF  28) SIGWINCH          29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN          35) SIGRTMIN+1       36) SIGRTMIN+2       37)
SIGRTMIN+3
38) SIGRTMIN+4       39) SIGRTMIN+5       40) SIGRTMIN+6       41) SIGRTMIN+7       42)
SIGRTMIN+8
```

```
43) SIGRTMIN+9      44) SIGRTMIN+10     45) SIGRTMIN+11     46) SIGRTMIN+12     47)
SIGRTMIN+13
48) SIGRTMIN+14     49) SIGRTMIN+15     50) SIGRTMAX-14     51) SIGRTMAX-13     52)
SIGRTMAX-12
53) SIGRTMAX-11     54) SIGRTMAX-10     55) SIGRTMAX-9      56) SIGRTMAX-8      57)
SIGRTMAX-7
58) SIGRTMAX-6      59) SIGRTMAX-5      60) SIGRTMAX-4      61) SIGRTMAX-3      62)
SIGRTMAX-2
63) SIGRTMAX-1      64) SIGRTMAX
```

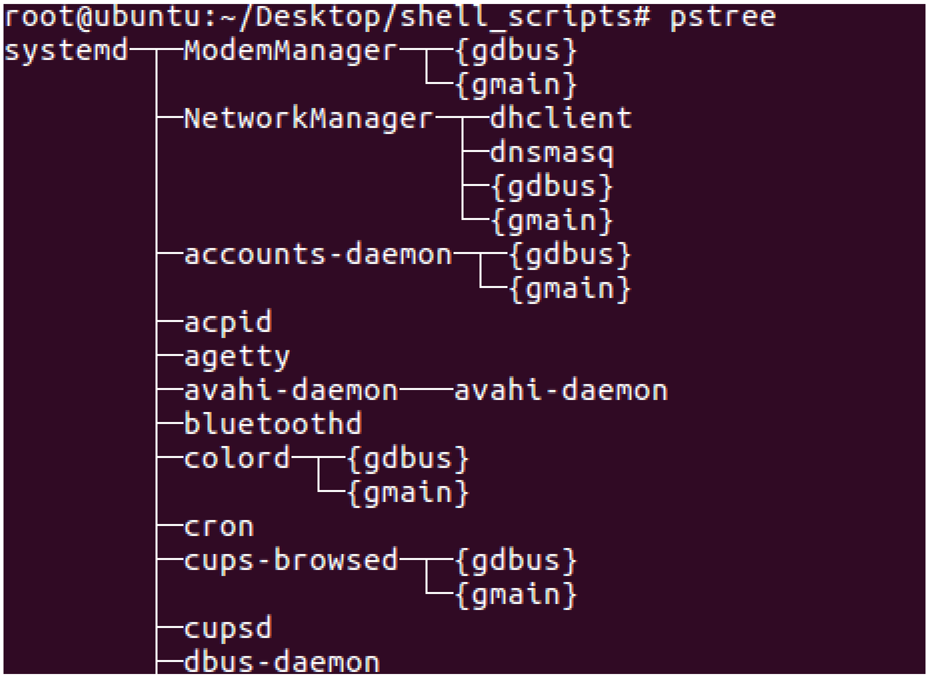
kill -SIGTERM 5432 Usual Kill
If you attempt to kill a process and it does not die, you can try using the unignorable SIGKILL signal
To find the process ID of the FireFox Web browser, you might issue this command

```
root@ubuntu:~/Desktop/shell_scripts# ps aux | grep firefox
root       5469  0.0  0.0  14224  1008 pts/4    S+   22:01   0:00 grep --color=auto firefox
```

The pkill command is used like the kill command but instead of using the PID we use command name.

```
root@ubuntu:~/Desktop/shell_scripts# pkill firefox
```

Process Tree
Each parent can have multiple children. To see the process hierarchy - shows the parent and grandparents.
Command is   pstree



```
root@ubuntu:~/Desktop/shell_scripts# pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─dnsmasq
        │                ├─{gdbus}
        │                └─{gmain}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─acpid
        ├─agetty
        ├─avahi-daemon───avahi-daemon
        ├─bluetoothd
        ├─colord─┬─{gdbus}
        │        └─{gmain}
        ├─cron
        ├─cups-browsed─┬─{gdbus}
        │              └─{gmain}
        ├─cupsd
        ├─dbus-daemon
```

Zombie Process:

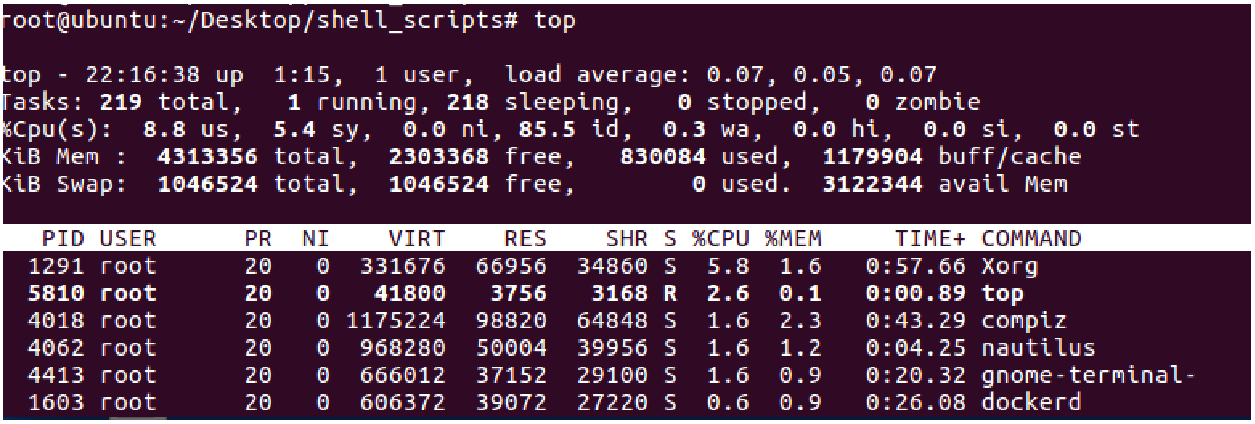
Normally, when a child process is killed, the parent process is told via a SIGCHLD signal. Then the parent can do some other task or restart a new child as needed. However, sometimes the parent process is killed. In this case, the “parent of all processes,” init, becomes the new PPID (parent process ID). this indicated by a process ID of 1 as the PPID of some other process

When a process is killed, a ps listing may still show the process with a Z state. This is a *zombie*, or defunct, process. The process is dead and not being used

If a process is hung, first try sending it a couple of -SIGTERM signals. If you wait and then verify that the process has not yet quit, try sending a -SIGKILL signal. If you find that the process stubbornly refuses to die, you may need to reboot your system.

Top Command:

top command is a very useful tool for quickly showing processes sorted by various criteria



```
oot@ubuntu:~/Desktop/shell_scripts# top

top - 22:16:38 up  1:15,  1 user,  load average: 0.07, 0.05, 0.07
Tasks: 219 total,   1 running, 218 sleeping,   0 stopped,   0 zombie
%Cpu(s):  8.8 us,  5.4 sy,  0.0 ni, 85.5 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  4313356 total,  2303368 free,   830084 used,  1179904 buff/cache
KiB Swap:  1046524 total,  1046524 free,        0 used.  3122344 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1291 root      20   0  331676  66956  34860 S  5.8  1.6   0:57.66 Xorg
 5810 root      20   0   41800   3756   3168 R  2.6  0.1   0:00.89 top
 4018 root      20   0 1175224  98820  64848 S  1.6  2.3   0:43.29 compiz
 4062 root      20   0  968280  50004  39956 S  1.6  1.2   0:04.25 nautilus
 4413 root      20   0  666012  37152  29100 S  1.6  0.9   0:20.32 gnome-terminal-
 1603 root      20   0  606372  39072  27220 S  0.6  0.9   0:26.08 dockerd
```

```
root@ubuntu:~/Desktop/shell_scripts# top

top - 22:16:16 up  1:15,  1 user,  load average: 0.00, 0.04, 0.07
Tasks: 219 total,   1 running, 218 sleeping,   0 stopped,   0 zombie
%Cpu(s): 14.9 us,  9.8 sy,  0.0 ni, 75.0 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  4313356 total,  2303740 free,   829716 used,  1179900 buff/cache
KiB Swap:  1046524 total,  1046524 free,        0 used.  3122716 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1291 root      20   0  331676  66956  34860 S 10.1  1.6   0:57.02 Xorg
 4018 root      20   0 1175224  98820  64848 S  6.8  2.3   0:43.06 compiz
 4413 root      20   0  665820  37152  29100 S  4.9  0.9   0:19.97 gnome-terminal-
 5810 root      20   0   41800   3756   3168 R  2.6  0.1   0:00.50 top
 1603 root      20   0  606372  39072  27220 S  0.6  0.9   0:25.97 dockerd
 4054 root      20   0  496872  28800  24892 S  0.6  0.7   0:21.14 vmtoolsd
    7 root      20   0       0      0      0 S  0.3  0.0   0:01.68 ksoftirqd/0
    8 root      20   0       0      0      0 S  0.3  0.0   0:02.07 rcu_sched
  985 rstudio+  20   0  129124   7372   6560 S  0.3  0.2   0:02.38 rserver
 1016 root      20   0  185608   9672   8496 S  0.3  0.2   0:19.62 vmtoolsd
 1922 root      20   0  284040   9872   7588 S  0.3  0.2   0:19.50 docker-containe
 5474 root      20   0       0      0      0 S  0.3  0.0   0:01.98 kworker/0:2
    1 root      20   0  119624   5824   4060 S  0.0  0.1   0:07.34 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.03 kthreadd
    4 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
```

The /proc file system contains a directory entry for active processes named after the PID. These directories contain files that provide various attributes about the process. For example, the following is the directory listing of a /proc file system for a personal shell process on

```
root@ubuntu:~/Desktop/shell_scripts# ls -l /proc/$$
total 0
dr-xr-xr-x 2 root root 0 Aug 14 23:24 attr
-rw-r--r-- 1 root root 0 Aug 14 23:24 autogroup
-r-------- 1 root root 0 Aug 14 23:24 auxv
-r--r--r-- 1 root root 0 Aug 14 23:24 cgroup
--w------- 1 root root 0 Aug 14 23:24 clear_refs
-r--r--r-- 1 root root 0 Aug 14 21:22 cmdline
-rw-r--r-- 1 root root 0 Aug 14 23:24 comm
-rw-r--r-- 1 root root 0 Aug 14 23:24 coredump_filter
-r--r--r-- 1 root root 0 Aug 14 23:24 cpuset
lrwxrwxrwx 1 root root 0 Aug 14 23:24 cwd -> /root/Desktop/shell_scripts
```

```
root@ubuntu:~/Desktop/shell_scripts# sudo ls -l /proc/1
total 0
dr-xr-xr-x 2 root root 0 Aug 14 23:26 attr
-rw-r--r-- 1 root root 0 Aug 14 23:26 autogroup
-r-------- 1 root root 0 Aug 14 23:26 auxv
-r--r--r-- 1 root root 0 Aug 14 21:01 cgroup
--w------- 1 root root 0 Aug 14 23:26 clear_refs
-r--r--r-- 1 root root 0 Aug 14 21:01 cmdline
-rw-r--r-- 1 root root 0 Aug 14 21:01 comm
-rw-r--r-- 1 root root 0 Aug 14 23:26 coredump_filter
-r--r--r-- 1 root root 0 Aug 14 23:26 cpuset
```

Run sleep at background for 60 sec

```
root@ubuntu:~/Desktop/shell_scripts# sleep 60 &
[1] 6385
root@ubuntu:~/Desktop/shell_scripts# jobs
[1]+  Running                 sleep 60 &
root@ubuntu:~/Desktop/shell_scripts# bg
bash: bg: job 1 already in background
root@ubuntu:~/Desktop/shell_scripts# fg
sleep 60
```