Linux Process Management Commands

This document explains useful Linux commands for monitoring and managing processes, with examples and outputs.

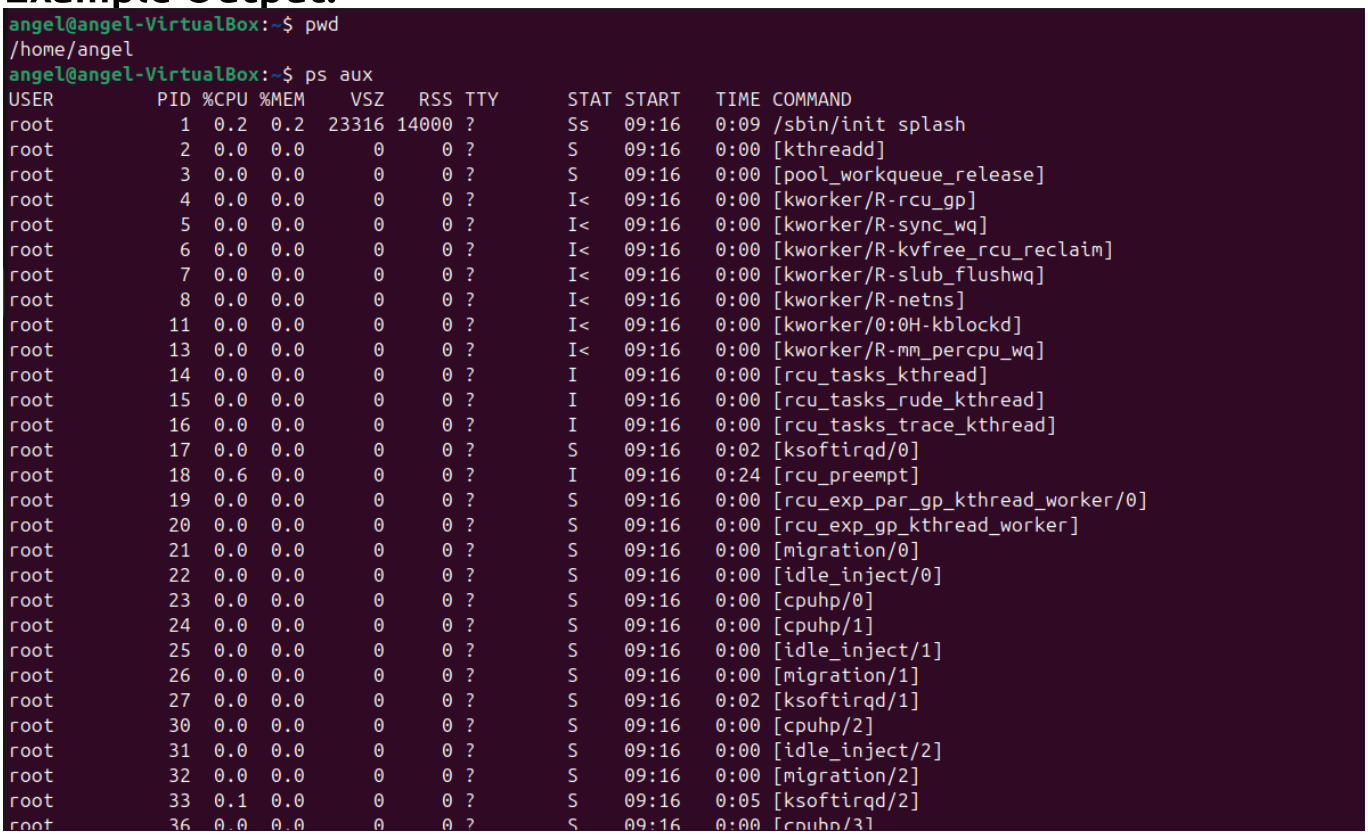
🌐 1. View All Processes

Command:

```
ps aux
```

- **a** → show processes for all users
- **u** → show user/owner of process
- **x** → show processes not attached to a terminal

Example Output:



```
angel@angel-VirtualBox:~$ pwd
/home/angel
angel@angel-VirtualBox:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.2  23316 14000 ?        Ss   09:16   0:09 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    09:16   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    09:16   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-rcu_gp]
root           5  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-sync_wq]
root           6  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-kvfree_rcu_reclaim]
root           7  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-slub_flushwq]
root           8  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-netns]
root          11  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/0:0H-kblockd]
root          13  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-mm_percpu_wq]
root          14  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_kthread]
root          15  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_rude_kthread]
root          16  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_trace_kthread]
root          17  0.0  0.0      0     0 ?        S    09:16   0:02 [ksoftirqd/0]
root          18  0.6  0.0      0     0 ?        I    09:16   0:24 [rcu_preempt]
root          19  0.0  0.0      0     0 ?        S    09:16   0:00 [rcu_exp_par_gp_kthread_worker/0]
root          20  0.0  0.0      0     0 ?        S    09:16   0:00 [rcu_exp_gp_kthread_worker]
root          21  0.0  0.0      0     0 ?        S    09:16   0:00 [migration/0]
root          22  0.0  0.0      0     0 ?        S    09:16   0:00 [idle_inject/0]
root          23  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/0]
root          24  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/1]
root          25  0.0  0.0      0     0 ?        S    09:16   0:00 [idle_inject/1]
root          26  0.0  0.0      0     0 ?        S    09:16   0:00 [migration/1]
root          27  0.0  0.0      0     0 ?        S    09:16   0:02 [ksoftirqd/1]
root          30  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/2]
root          31  0.0  0.0      0     0 ?        S    09:16   0:00 [idle_inject/2]
root          32  0.0  0.0      0     0 ?        S    09:16   0:00 [migration/2]
root          33  0.1  0.0      0     0 ?        S    09:16   0:05 [ksoftirqd/2]
root          36  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/3]
```

🌲 2. Process Tree

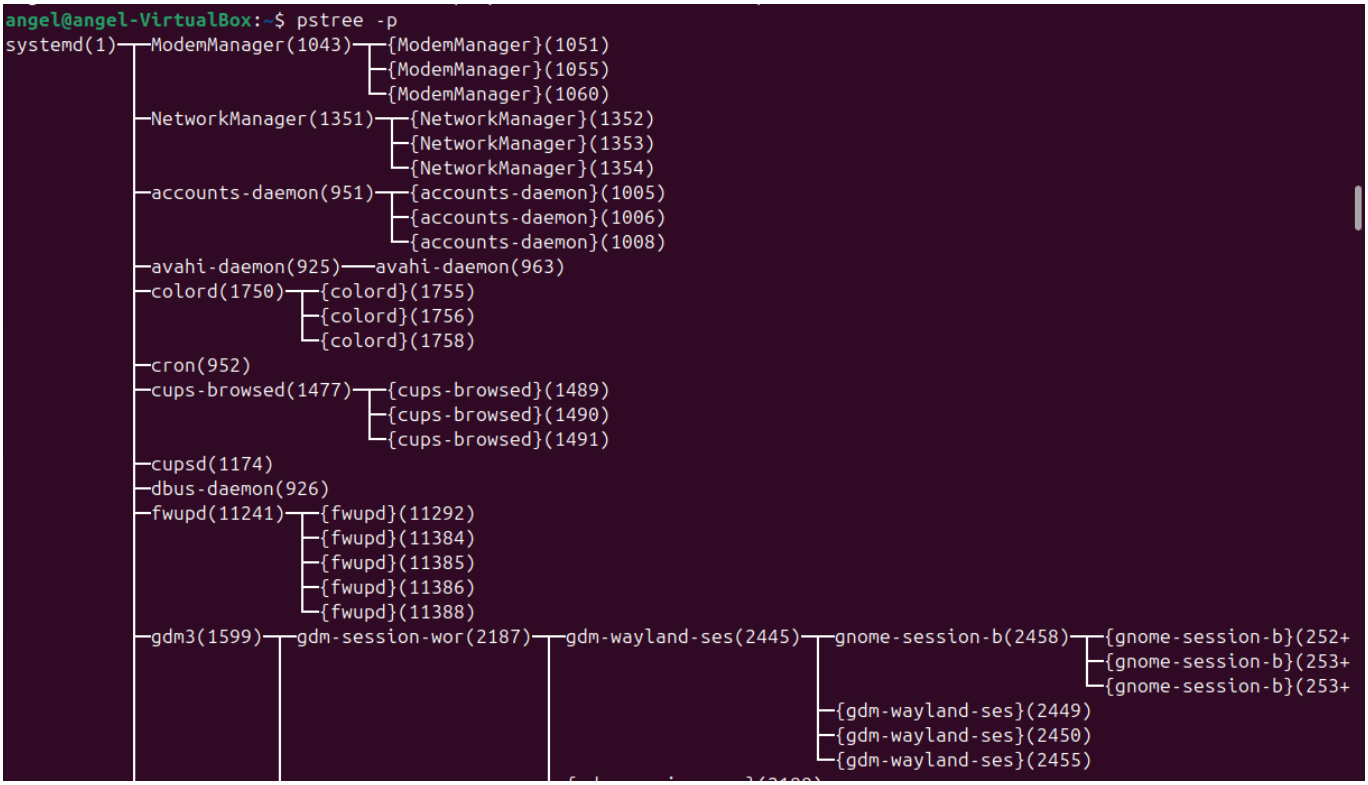
Command:

```
pstree -p
```

Shows parent-child relationships.

Example Output:

```
systemd(1)─┬─NetworkManager(778)
           ├─sshd(895)─┬─sshd(1023)───bash(1024)───pstree(1101)
           ├─mysqld(2001)
           └─python3(1234)
```

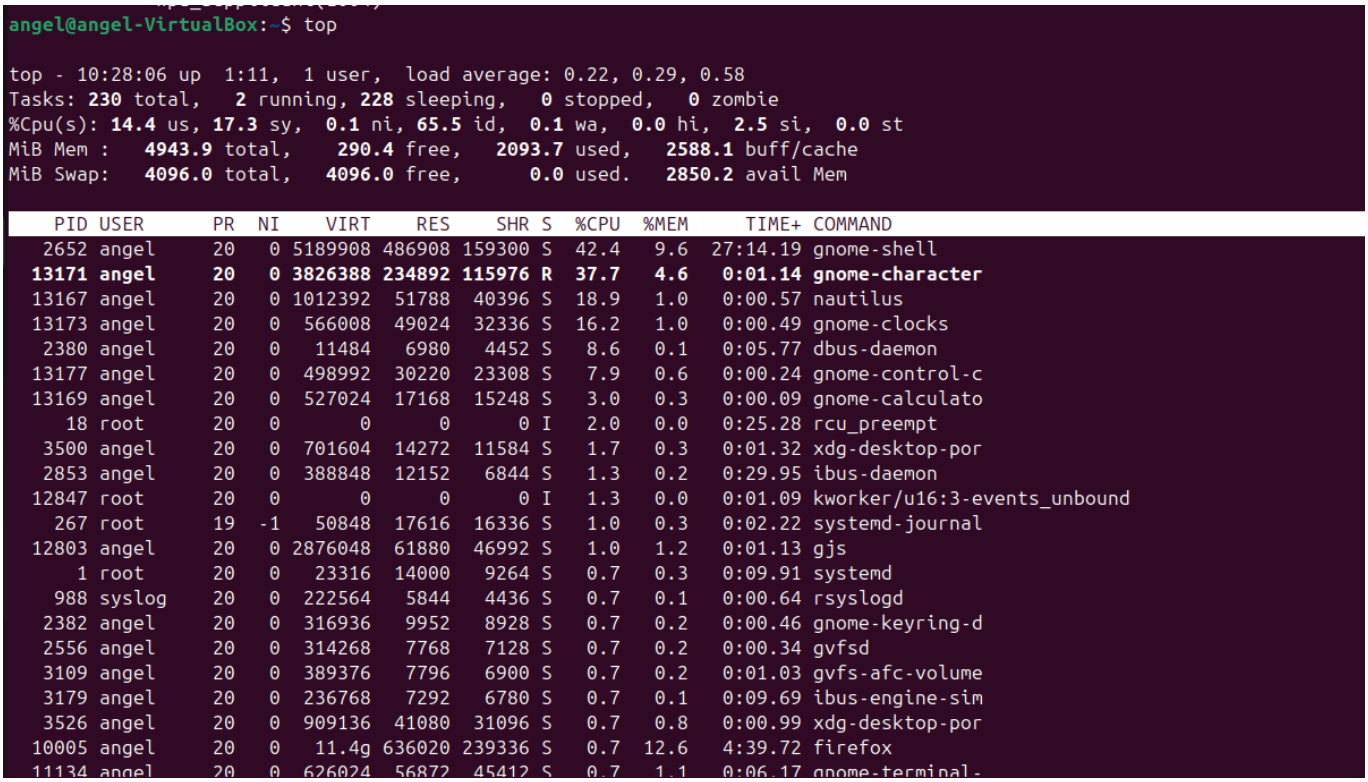


📊 3. Real-Time Monitoring

Command:

```
top
```

Displays live system resource usage.



Example Output:

```
top - 10:20:51 up 2 days, 3:12, 2 users, load average: 0.22, 0.33, 0.45
Tasks: 197 total, 1 running, 196 sleeping, 0 stopped, 0 zombie
%Cpu(s): 12.3 us, 5.4 sy, 0.0 ni, 80.1 id, 2.2 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8045632 total, 3564980 free, 1876324 used, 2604328 buff/cache

PID  USER   PR  NI  VIRT   RES  SHR S %CPU %MEM TIME+    COMMAND
1234 vibhu  20   0 274532 15632 7892 R 45.0 1.5  0:12.34 python3
2001 mysql  20   0 450000 20988 7564 S 25.0 2.0  1:02.11 mysqld
```

☞ Press **q** to quit.

⚡ 4. Adjust Process Priority

Start process with low priority:

```
nice -n 10 sleep 300 &
```

Output:

```
[1] 13749
```

☞ Process with PID **13749** runs with nice value 10.

Change priority:

```
renice -n -5 -p 13749
```

Output:

```
13749 (process ID) old priority 10, new priority -5
```

🔧 5. CPU Affinity

Show which CPUs process can use:

```
taskset -cp 3050
```

Output:

```
pid 3050's current affinity list: 0-3
```

Restrict to core 1:

```
taskset -cp 1 3050
```

Output:

```
pid 3050's current affinity list: 1
```

📂 6. I/O Scheduling Priority

```
ionice -c 3 -p 3050
```

Output:

```
successfully set pid 3050's IO scheduling class to idle
```

☞ Class 3 = only gets I/O when system is idle.
