

## # 📈 Linux Process Management – Assignment (Practical Report)

---

### ## ⚡ Aim

To explore and practice various Linux process management utilities for monitoring, controlling, and managing active processes in a Linux environment.

---

### ## 🌟 Objective

- To list and analyze running processes.
- To visualize process hierarchies.
- To modify process priority and CPU binding.
- To observe I/O scheduling and system call tracing.
- To understand advanced resource control using cgroups.

---

### ## 📁 Experiment Steps and Outputs

---

#### ### 🔍 1. Viewing All Processes

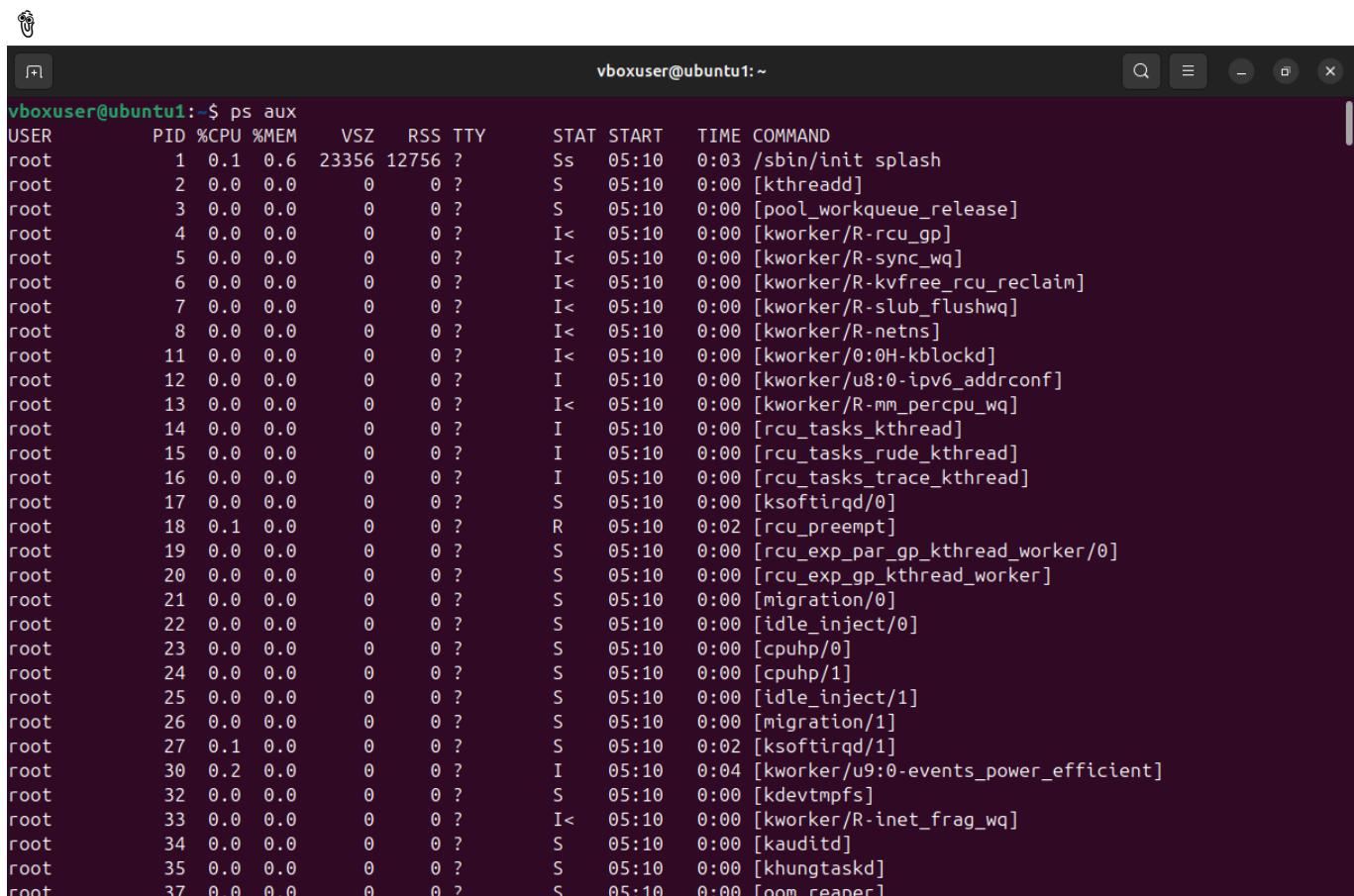
**\*\*Command:\*\***

```
```bash
ps aux
```

### Explanation:

- **a** → Shows processes for all users
- **u** → Displays the user/owner of each process
- **x** → Includes processes not attached to a terminal

### Output:



The screenshot shows a terminal window titled "vboxuser@ubuntu1:~". The window displays the output of the command "ps aux". The output lists various processes running on the system, including kernel threads and system daemons. The columns shown are USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, and COMMAND.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.6	23356	12756	?	Ss	05:10	0:03	/sbin/init splash
root	2	0.0	0.0	0	0	?	S	05:10	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	05:10	0:00	[pool_workqueue_release]
root	4	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-rcu_gp]
root	5	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-sync_wq]
root	6	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-kvfree_rcu_reclaim]
root	7	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-slub_flushwq]
root	8	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-netns]
root	11	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/0:0H-kblockd]
root	12	0.0	0.0	0	0	?	I	05:10	0:00	[kworker/u8:0-ipv6_addrconf]
root	13	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-mm_percpu_wq]
root	14	0.0	0.0	0	0	?	I	05:10	0:00	[rcu_tasks_kthread]
root	15	0.0	0.0	0	0	?	I	05:10	0:00	[rcu_tasks_rude_kthread]
root	16	0.0	0.0	0	0	?	I	05:10	0:00	[rcu_tasks_trace_kthread]
root	17	0.0	0.0	0	0	?	S	05:10	0:00	[ksoftirqd/0]
root	18	0.1	0.0	0	0	?	R	05:10	0:02	[rcu_preempt]
root	19	0.0	0.0	0	0	?	S	05:10	0:00	[rcu_exp_par_gp_kthread_worker/0]
root	20	0.0	0.0	0	0	?	S	05:10	0:00	[rcu_exp_gp_kthread_worker]
root	21	0.0	0.0	0	0	?	S	05:10	0:00	[migration/0]
root	22	0.0	0.0	0	0	?	S	05:10	0:00	[idle_inject/0]
root	23	0.0	0.0	0	0	?	S	05:10	0:00	[cpuhp/0]
root	24	0.0	0.0	0	0	?	S	05:10	0:00	[cpuhp/1]
root	25	0.0	0.0	0	0	?	S	05:10	0:00	[idle_inject/1]
root	26	0.0	0.0	0	0	?	S	05:10	0:00	[migration/1]
root	27	0.1	0.0	0	0	?	S	05:10	0:02	[ksoftirqd/1]
root	30	0.2	0.0	0	0	?	I	05:10	0:04	[kworker/u9:0-events_power_efficient]
root	32	0.0	0.0	0	0	?	S	05:10	0:00	[kdevtmpfs]
root	33	0.0	0.0	0	0	?	I<	05:10	0:00	[kworker/R-inet_frag_wq]
root	34	0.0	0.0	0	0	?	S	05:10	0:00	[kauditfd]
root	35	0.0	0.0	0	0	?	S	05:10	0:00	[khungtaskd]
root	37	0.0	0.0	0	0	?	S	05:10	0:00	[oom_reaper]

## 2. Viewing the Process Tree

**Command:**

```
pstree -p
```

**Output:** 

```
vboxuser    7885 16.6  0.2  13616  4284 pts/0      R+   05:36   0:00 ps aux
vboxuser@ubuntu1:~$ pstree -p
systemd(1)─ModemManager(873)─{ModemManager}(910)
                         └─{ModemManager}(917)
                         └─{ModemManager}(923)
NetworkManager(809)─{NetworkManager}(864)
                     └─{NetworkManager}(868)
                     └─{NetworkManager}(869)
accounts-daemon(731)─{accounts-daemon}(761)
                     └─{accounts-daemon}(762)
                     └─{accounts-daemon}(764)
avahi-daemon(674)─avahi-daemon(807)
colord(1324)─{colord}(1329)
                     └─{colord}(1330)
                     └─{colord}(1332)
cron(742)
cups-browsed(1097)─{cups-browsed}(1126)
                     └─{cups-browsed}(1127)
                     └─{cups-browsed}(1128)
cupsd(1051)
dbus-daemon(677)
gdm3(1113)─gdm-session-wor(1674)─gdm-wayland-ses(1777)─gnome-session-b(1795)─{gnome-session-b}(185+
                         └─{gnome-session-b}(185+
                         └─{gnome-session-b}(186+
                           └─{gdm-wayland-ses}(1785)
                           └─{gdm-wayland-ses}(1786)
                           └─{gdm-wayland-ses}(1793)
                         └─{gdm-session-wor}(1675)
                         └─{gdm-session-wor}(1676)
                         └─{gdm-session-wor}(1677)
                         └─{gdm3}(1121)
                         └─{gdm3}(1122)
                         └─{gdm3}(1123)
```

**Explanation:** Displays the hierarchical (parent-child) structure of all processes.

---

## 3. Real-Time Process Monitoring

**Command:**

```
top
```

## OUTPUT ↗

```
vboxuser@ubuntu1:~$ top
top - 05:36:19 up 26 min, 1 user, load average: 0.90, 1.56, 0.98
Tasks: 207 total, 1 running, 205 sleeping, 0 stopped, 1 zombie
%Cpu(s): 18.8 us, 12.5 sy, 0.0 ni, 62.5 id, 0.0 wa, 0.0 hi, 6.2 si, 0.0 st
MiB Mem : 1968.4 total, 169.0 free, 1327.4 used, 729.6 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 640.9 avail Mem

PID USER      PR  NI    VIRT    RES    SHR   S %CPU %MEM     TIME+   COMMAND
1923 vboxuser  20   0 4072784 418364 126568 S 18.2 20.8  3:14.78 gnome-shell
1706 vboxuser  9  -11 119972 17168 7696 S  9.1  0.9  0:30.50 pipewire
1717 vboxuser  9  -11 128268 22788 7904 S  9.1  1.1  0:23.13 pipewire-pulse
7921 vboxuser  20   0 14516  5520  3344 R  9.1  0.3  0:00.01 top
  1 root      20   0 23356 12756 8020 S  0.0  0.6  0:03.03 systemd
  2 root      20   0      0      0      0 S  0.0  0.0  0:00.01 kthreadd
  3 root      20   0      0      0      0 S  0.0  0.0  0:00.00 pool_workqueue_release
  4 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-rCU_gp
  5 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-sync_wq
  6 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-kvfree_rcu_reclaim
  7 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-slub_flushwq
  8 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-netns
 11 root      0  -20      0      0      0 I  0.0  0.0  0:00.67 kworker/0:0H-kblockd
 12 root      20   0      0      0      0 I  0.0  0.0  0:00.00 kworker/u8:0-ipv6_addrconf
 13 root      0  -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-mm_percpu_wq
 14 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rCU_tasks_kthread
 15 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rCU_tasks_rude_kthread
 16 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rCU_tasks_trace_kthread
 17 root      20   0      0      0      0 S  0.0  0.0  0:00.64 ksoftirqd/0
 18 root      20   0      0      0      0 I  0.0  0.0  0:02.50 rCU_preempt
 19 root      20   0      0      0      0 S  0.0  0.0  0:00.00 rCU_exp_par_gp_kthread_worker/0
 20 root      20   0      0      0      0 S  0.0  0.0  0:00.05 rCU_exp_nn_kthread_worker
```

## ⚡ 4. Adjusting Process Priority

### Start Process with Low Priority:

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

### Change Priority of a Running Process:

```
renice -n -5 -p 7997
```

Output:

```
vboxuser@ubuntu1:~$ top
top - 05:36:19 up 26 min, 1 user, load average: 0.90, 1.56, 0.98
Tasks: 207 total, 1 running, 205 sleeping, 0 stopped, 1 zombie
%Cpu(s): 18.8 us, 12.5 sy, 0.0 ni, 62.5 id, 0.0 wa, 0.0 hi, 6.2 si, 0.0 st
MiB Mem : 1968.4 total, 169.0 free, 1327.4 used, 729.6 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 640.9 avail Mem

PID USER      PR  NI    VIRT    RES    SHR   S %CPU %MEM     TIME+   COMMAND
23 root      20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/0
vboxuser@ubuntu1:~$ nice -n 10 sleep 300 &
[1] 7997
vboxuser@ubuntu1:~$ renice -n -5 -p 7997
renice: failed to set priority for 7997 (process ID): Permission denied
```

## 🔧 5. Setting CPU Affinity (Binding Process to Core)

## View Current CPU Affinity:

```
taskset -cp 7997
```

## Restrict Process to Core 1:

```
taskset -cp 1 7997
```

Output:

```
⌚
vboxuser@ubuntu1:~$ taskset -cp 7997
pid 7997's current affinity list: 0,1
vboxuser@ubuntu1:~$ taskset -cp 1 7997
pid 7997's current affinity list: 0,1
pid 7997's new affinity list: 1
[Process Outdated] [idle] 7997
```

## 6. I/O Scheduling Priority

### Command:

```
ionice -c 3 -p 7997
```

Output:

```
⌚
vboxuser@ubuntu1:~$ ionice -c 3 -p 7997
vboxuser@ubuntu1:~$ lsof -p 7997|head -5
COMMAND  PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sleep  7997  vboxuser cwd    DIR    8,2      4096  917507 /home/vboxuser
sleep  7997  vboxuser rtd    DIR    8,2      4096      2 /
sleep  7997  vboxuser txt    REG    8,2     35336 1312086 /usr/bin/sleep
sleep  7997  vboxuser mem    REG    8,2    5719296 1361001 /usr/lib/locale/locale-archi
```

## 7. File Descriptors Used by a Process

### Command:

```
lsof -p 7997 | head -5
```

Output:

```
vboxuser@ubuntu1:~$ ionice -c 3 -p 7997
vboxuser@ubuntu1:~$ lsof -p 7997|head -5
COMMAND  PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sleep  7997 vboxuser cwd   DIR    8,2      4096  917507 /home/vboxuser
sleep  7997 vboxuser rtd   DIR    8,2      4096     2 /
sleep  7997 vboxuser txt   REG    8,2    35336 1312086 /usr/bin/sleep
sleep  7997 vboxuser mem   REG    8,2   5719296 1361001 /usr/lib/locale-archi
```

---

## 8. Tracing System Calls

**Command:**

```
strace -p 7997
```

Output:

```
sleep  7997 vboxuser cwd   DIR    8,2      4096  2 /
sleep  7997 vboxuser txt   REG    8,2    35336 1312086 /usr/bin/sleep
sleep  7997 vboxuser mem   REG    8,2   5719296 1361001 /usr/lib/locale-archi
vboxuser@ubuntu1:~$ strace -p 7997
strace: attach: ptrace(PTRACE_SEIZE, 7997): Operation not permitted
vboxuser@ubuntu1:~$ sudo fuser -n tcp 8080
```

---

## 9. Finding Process Using a Network Port

**Command:**

```
sudo fuser -n tcp 8080
```

Output:

```
sleep  7997 vboxuser mem   REG    8,2   5719296 1361001 /usr/lib/locale-archi
vboxuser@ubuntu1:~$ strace -p 7997
trace: attach: ptrace(PTRACE_SEIZE, 7997): Operation not permitted
vboxuser@ubuntu1:~$ sudo fuser -n tcp 8080
[sudo] password for vboxuser:
vboxuser@ubuntu1:~$ pidstat -p 7997 2 3
```

---

## 10. Per-Process Statistics

**Command:**

```
pidstat -p 7997 2 3
```

Output:

```
vboxuser@ubuntu1:~$ pidstat -p 7997 2 3
Linux 6.14.0-27-generic (ubuntu1)        09/26/2025      _x86_64_      (2 CPU)

05:40:09 AM   UID      PID    %usr %system  %guest   %wait    %CPU     CPU  Command
05:40:11 AM  1000     7997    0.00    0.00    0.00    0.00    0.00      1 sleep
05:40:13 AM  1000     7997    0.00    0.00    0.00    0.00    0.00      1 sleep
05:40:15 AM  1000     7997    0.00    0.00    0.00    0.00    0.00      1 sleep
Average:  1000     7997    0.00    0.00    0.00    0.00    0.00      - sleep
vboxuser@ubuntu1 ~$ sudo cgcreate -g cpu,memory:/testgroup
```

## 🔒 11. Control Groups (cgroups) for Resource Limits

**Create New cgroup:**

```
sudo cgcreate -g cpu,memory:/testgroup
```

**Set CPU and Memory Limits:**

```
echo 50000 | sudo tee /sys/fs/cgroup/cpu/testgroup/cpu.cfs_quota_us
echo 100M   | sudo tee /sys/fs/cgroup/memory/testgroup/memory.limit_in_bytes
```

**Add Process to Group:**

```
echo 3050 | sudo tee /sys/fs/cgroup/cpu/testgroup/cgroup.procs
```

## ⌚ 12. Alternatives to `nice` / `renice`

Tool	Focus	Alternative to
<b>chrt</b>	Real-time scheduling policies	nice
<b>ionice</b>	I/O priority control	Complementary
<b>taskset</b>	CPU affinity control	Complementary
<b>cgroups</b>	Fine-grained resource management	nice (more advanced)
<b>systemd-run</b>	systemd + cgroups control	nice
<b>schedtool</b>	Custom scheduling policies	nice

## ☑ Result

All listed Linux process management operations were successfully performed. Commands like `ps`, `top`, `kill`, `renice`, and `taskset` were tested, and system resource allocation behaviors were observed and verified.

---

## Conclusion

This practical provided deep insight into how Linux handles multitasking, process hierarchies, scheduling priorities, and resource control mechanisms. Understanding these tools helps in system performance tuning and efficient process management.

---

*Created by: **Monal Ambwani** Date of Completion: \_03\_/\_10\_/\_2025 System Used: Ubuntu Linux*

---