

Linux Process Management — Lab Report

Objective

Understanding and implementing process management concepts in Linux through hands-on examples.

Theory

◊ What is a Process?

A process is an instance of a running program with its own memory space and system resources.

◊ Process States

-  R (Running/Runnable)
-  S (Sleeping — waiting for event)
-  D (Uninterruptible sleep)
-  T (Stopped)
-  Z (Zombie)

◊ Process Types

-  Foreground: Connected to terminal
-  Background: Runs independently

◊ Process Identifiers

-  PID: Unique process ID
-  PPID: Parent process ID

◊ Core Concepts

-  Parent/Child processes
-  Zombie processes
-  Orphan processes

Commands & Examples

View all processes (detailed)

```
ps aux
```

Explanation: Shows all running processes with 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	7.5	0.1	23580	14176	?	Ss	00:09	0:05	/sbin/init sp
root	2	0.1	0.0	0	0	?	S	00:09	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	00:09	0:00	[pool_workque]
root	4	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-rc]
root	5	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-sy]
root	6	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-kv]
root	7	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-sl]
root	8	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-ne]
root	9	0.0	0.0	0	0	?	I	00:09	0:00	[kworker/0:0-]
root	10	0.3	0.0	0	0	?	I	00:09	0:00	[kworker/0:1-]
root	11	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/0:0H]
root	12	0.0	0.0	0	0	?	I	00:09	0:00	[kworker/u20:]
root	13	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/R-mm]
root	14	0.0	0.0	0	0	?	I	00:09	0:00	[rcu_tasks_kt]
root	15	0.0	0.0	0	0	?	I	00:09	0:00	[rcu_tasks_ru]
root	16	0.0	0.0	0	0	?	I	00:09	0:00	[rcu_tasks_tr]
root	17	0.3	0.0	0	0	?	S	00:09	0:00	[ksoftirqd/0]
root	18	0.9	0.0	0	0	?	I	00:09	0:00	[rcu_preempt]
root	19	0.0	0.0	0	0	?	S	00:09	0:00	[rcu_exp_par_]
root	20	0.4	0.0	0	0	?	S	00:09	0:00	[rcu_exp_gp_k]
root	21	0.0	0.0	0	0	?	S	00:09	0:00	[migration/0]
root	22	0.0	0.0	0	0	?	S	00:09	0:00	[idle_inject/]
root	23	0.0	0.0	0	0	?	S	00:09	0:00	[cpuhp/0]
root	24	0.0	0.0	0	0	?	S	00:09	0:00	[cpuhp/1]
root	25	0.0	0.0	0	0	?	S	00:09	0:00	[idle_inject/]
root	26	2.1	0.0	0	0	?	S	00:09	0:01	[migration/1]
root	27	0.1	0.0	0	0	?	S	00:09	0:00	[ksoftirqd/1]
root	28	0.2	0.0	0	0	?	I	00:09	0:00	[kworker/1:0-]
root	29	0.0	0.0	0	0	?	I<	00:09	0:00	[kworker/1:0H]
root	30	0.0	0.0	0	0	?	S	00:09	0:00	[cpuhp/2]
root	31	0.0	0.0	0	0	?	S	00:09	0:00	[idle_inject/]

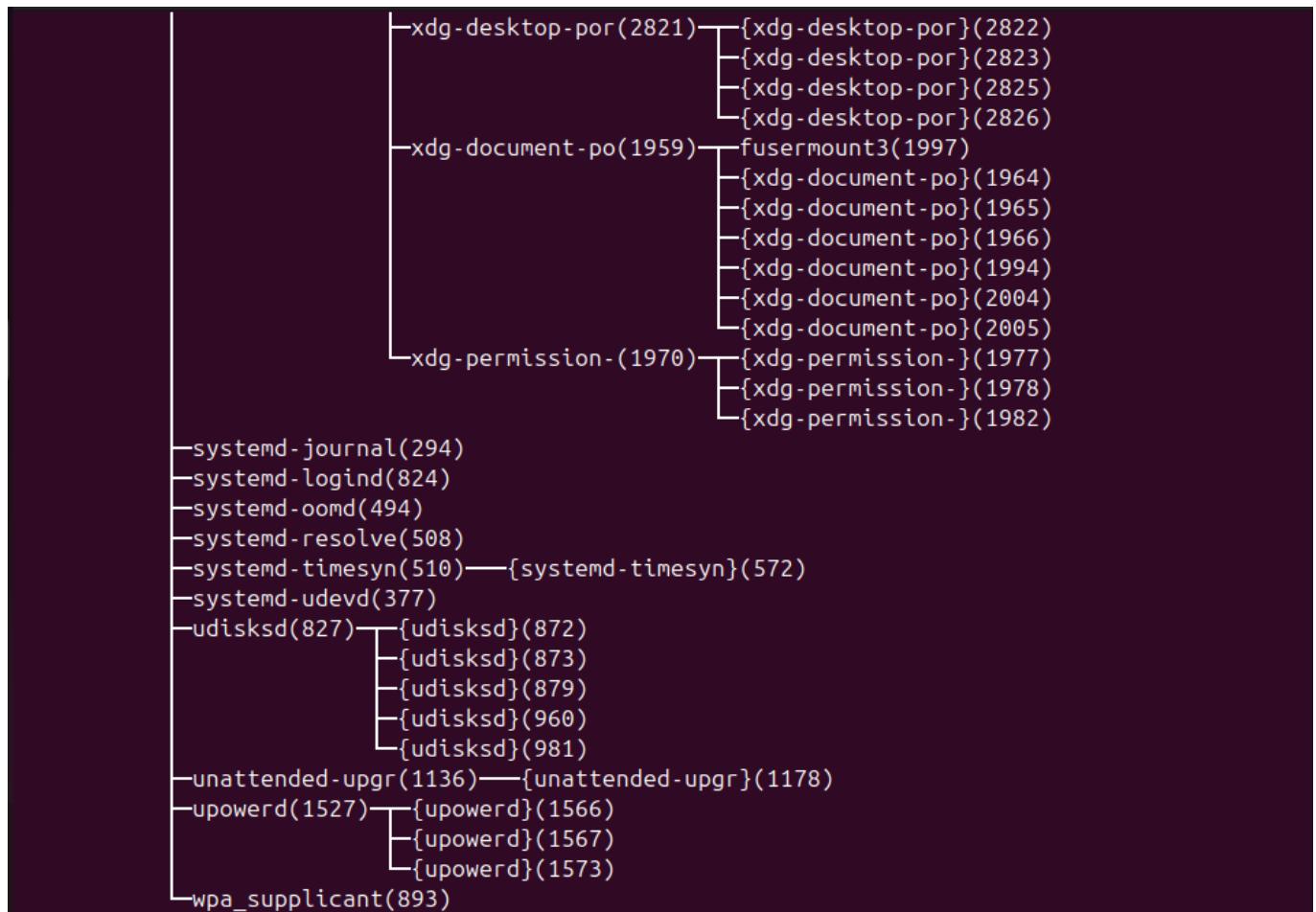
ujjwalt+	2414	2.2	0.1	471720	11164	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2425	0.6	0.1	424904	14808	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2450	0.2	0.0	245312	7104	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2452	42.3	0.3	429636	27968	?	Sl	00:10	0:05	/usr/libexec/
ujjwalt+	2461	0.5	0.0	319092	7036	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2473	1.2	0.2	555328	23980	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2487	3.7	0.1	39136	11848	?	Ss	00:10	0:00	/snap/snapd-d
ujjwalt+	2493	0.3	0.0	318440	6208	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2512	0.3	0.1	397792	8916	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2513	2.6	0.2	899384	24200	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2522	0.4	0.0	318460	6200	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2542	0.8	0.0	398044	7724	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2566	0.6	0.0	319428	6744	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2580	1.9	0.3	833976	29684	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2603	5.3	0.3	429612	30568	?	Sl	00:10	0:00	/snap/snapd-d
ujjwalt+	2642	0.4	0.0	245436	7108	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2665	0.4	0.0	230220	5564	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2674	6.7	0.6	946744	50312	?	Sl	00:10	0:00	/usr/bin/naut
ujjwalt+	2675	0.7	0.0	244936	6148	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2676	1.4	0.1	618108	8884	?	Sl	00:10	0:00	/usr/libexec/
ujjwalt+	2709	2.6	0.1	562804	13816	?	Ssl	00:10	0:00	/usr/libexec/
root	2739	0.0	0.0	0	0	?	I<	00:10	0:00	[kworker/u27:]
root	2740	0.0	0.0	0	0	?	I<	00:10	0:00	[kworker/u27:]
ujjwalt+	2754	4.1	0.2	742932	24924	?	SNsl	00:10	0:00	/usr/libexec/
ujjwalt+	2755	5.4	0.4	925996	40952	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2766	11.3	0.7	3019200	61592	?	Sl	00:10	0:00	gjs /usr/shar
ujjwalt+	2783	2.0	0.3	2544192	27792	?	Sl	00:10	0:00	/usr/bin/gjs
ujjwalt+	2821	4.8	0.2	425872	24388	?	Ssl	00:10	0:00	/usr/libexec/
ujjwalt+	2864	29.2	0.6	561652	51488	?	Ssl	00:10	0:01	/usr/libexec/
ujjwalt+	2871	1.6	0.0	19692	5056	pts/0	Ss	00:10	0:00	bash
ujjwalt+	2879	150	0.0	22284	4620	pts/0	R+	00:10	0:00	ps aux

Hierarchical process view

```
pstree -p
```

Explanation: Displays processes in a tree form with PIDs — useful to see parent/child relations.

```
ujjwaltyagi@ujjwaltyagi:~$ pstree -p
systemd(1)─ ModemManager(976)─ {ModemManager}(1006)
              └─ {ModemManager}(1013)
                  └─ {ModemManager}(1017)
              └─ NetworkManager(878)─ {NetworkManager}(978)
                  └─ {NetworkManager}(983)
                      └─ {NetworkManager}(984)
              └─ accounts-daemon(801)─ {accounts-daemon}(857)
                  └─ {accounts-daemon}(861)
                      └─ {accounts-daemon}(869)
              └─ avahi-daemon(768)─ avahi-daemon(880)
              └─ canonical-livep(1132)─ {canonical-livep}(1425)
                  └─ {canonical-livep}(1429)
                  └─ {canonical-livep}(1430)
                  └─ {canonical-livep}(1432)
                  └─ {canonical-livep}(1442)
                  └─ {canonical-livep}(1443)
                  └─ {canonical-livep}(1444)
                  └─ {canonical-livep}(1445)
                  └─ {canonical-livep}(1446)
                  └─ {canonical-livep}(1447)
              └─ colord(1480)─ {colord}(1485)
                  └─ {colord}(1486)
                      └─ {colord}(1488)
              └─ cron(804)
              └─ cups-browsed(1158)─ {cups-browsed}(1185)
                  └─ {cups-browsed}(1186)
                      └─ {cups-browsed}(1191)
              └─ cupsd(1128)─ dbus(1155)
              └─ dbus-daemon(769)
              └─ fwupd(4425)─ {fwupd}(4426)
                  └─ {fwupd}(4427)
                      └─ {fwupd}(4428)
```



Real-time monitoring

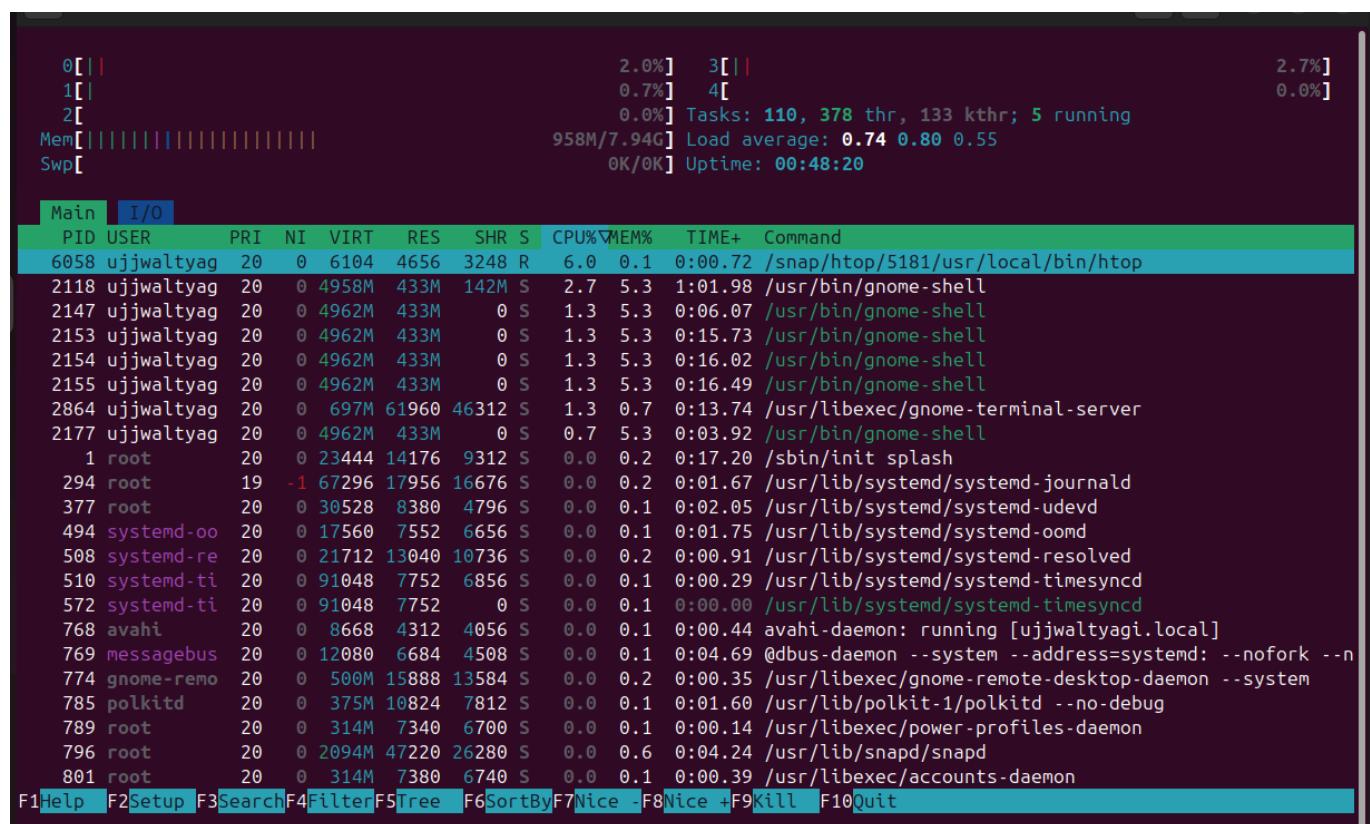
```
top
```

Explanation: Interactive display of processes sorted by CPU usage. Press **q** to quit.

top - 00:52:03 up 42 min, 1 user, load average: 0.06, 0.31, 0.37													
Tasks: 235 total, 1 running, 234 sleeping, 0 stopped, 0 zombie													
%Cpu(s): 0.2 us, 0.4 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st													
MiB Mem : 8128.5 total, 5341.6 free, 1179.7 used, 1886.5 buff/cache													
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 6948.8 avail Mem													
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND		
2118	ujjwalt+	20	0	5077084	444020	145712	S	4.0	5.3	1:55.29	gnome-shell		
2864	ujjwalt+	20	0	638184	60264	45000	S	0.5	0.7	0:09.03	gnome-terminal-		
3525	root	20	0	0	0	0	I	0.3	0.0	0:00.54	kworker/u21:1-flush-8:0		
1	root	20	0	23444	14176	9312	S	0.0	0.2	0:10.56	systemd		
2	root	20	0	0	0	0	S	0.0	0.0	0:00.19	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-slab_flushwq		
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns		
10	root	20	0	0	0	0	I	0.0	0.0	0:05.36	kworker/0:1-events		
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u20:0-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	0:04.77	ksoftirqd/0		
18	root	20	0	0	0	0	I	0.0	0.0	0:03.84	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	0:00.42	rcu_exp_gp_kthread_worker		
21	root	rt	0	0	0	0	S	0.0	0.0	0:00.12	migration/0		
22	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0		
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0		
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1		

htop

Explanation: Enhanced interactive viewer with process tree, colors, and easier navigation.



Run process in background

```
sleep 300 &
```

Explanation: Starts `sleep` in background; shell prints the job number and PID.

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 300 &
[1] 6131
sleep 300 &
```

List jobs in current shell

```
jobs -l
```

Explanation: Lists jobs started from the current shell with job IDs and PIDs.

```
ujjwaltyagi@ujjwaltyagi:~$ jobs -l
[1]+ 6131 Running                  sleep 300 &
ujjwaltyagi@ujjwaltyagi:~$
```

Bring job to foreground / send to background

```
fg %1
bg %1
```

Explanation: `fg` brings job 1 to foreground; `bg` resumes job 1 in background.

```
ujjwaltyagi@ujjwaltyagi:~$ fg %1
sleep 300
```

Dismiss a background job

```
disown %1
```

Explanation: Removes job 1 from shell's job table so it will not receive SIGHUP on shell exit.

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 400 &
[1] 3458
ujjwaltyagi@ujjwaltyagi:~$ jobs
[1]+  Running                  sleep 400 &
ujjwaltyagi@ujjwaltyagi:~$ disown %1
ujjwaltyagi@ujjwaltyagi:~$ jobs
ujjwaltyagi@ujjwaltyagi:~$ █
```

Run detached (nohup)

```
nohup python3 server.py > server.log 2>&1 &
```

Explanation: Starts `server.py` immune to hangups; output redirected to `server.log`.

```
ujjwaltyagi@ujjwaltyagi:~$ nohup python3 server.py > server.log 2>1 &
[1] 7049
ujjwaltyagi@ujjwaltyagi:~$ █
```

Find process by name / PID

```
pgrep -a sshd
pidof bash
```

Explanation: `pgrep -a` lists PIDs and command lines matching `sshd`. `pidof` returns PID(s) for a program.

```
ujjwaltyagi@ujjwaltyagi:~$ pgrep -a sshd
[1]+  Exit 2                  nohup python3 server.py > server.log 2> 1
ujjwaltyagi@ujjwaltyagi:~$ pidof bash
6260
ujjwaltyagi@ujjwaltyagi:~$ █
```

Kill a process (graceful then force)

```
kill PID                      # sends SIGTERM by default
kill -9 PID                    # sends SIGKILL (force)
pkill -f processname
```

Explanation: `kill` sends signals to PIDs; `pkill -f` matches the full command line.

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 1200 &
[1] 9184
ujjwaltyagi@ujjwaltyagi:~$ ps
\ PID TTY          TIME CMD
 6260 pts/0    00:00:00 bash
 9184 pts/0    00:00:00 sleep
 9277 pts/0    00:00:00 ps
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  9184  0.0  0.0  16956  1908 pts/0      S     01:17   0:00 sleep 1200
ujjwalt+  9524  0.0  0.0  17812  2288 pts/0      S+    01:18   0:00 grep --color=
auto sleep
ujjwaltyagi@ujjwaltyagi:~$ kill 9184
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  9688  0.0  0.0  17812  2272 pts/0      S+    01:19   0:00 grep --color=
auto sleep
[1]+  Terminated                  sleep 1200
```

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 1345 &
[1] 10099
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  10099  0.0  0.0  16956  1980 pts/0      S     01:20   0:00 sleep 1345
ujjwalt+  10122  0.0  0.0  17812  2284 pts/0      S+    01:20   0:00 grep --color=
auto sleep
ujjwaltyagi@ujjwaltyagi:~$ kill -9 10099
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  10208  0.0  0.0  17812  2288 pts/0      S+    01:21   0:00 grep --color=
auto sleep
[1]+  Killed                      sleep 1345
```

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 2312 &
[2] 10704
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  10393  0.0  0.0  16956  1980 pts/0      S+    01:22   0:00 sleep 2345
ujjwalt+  10542  0.0  0.0  16956  1980 pts/1      S     01:22   0:00 sleep 2312
ujjwalt+  10704  0.0  0.0  16956  1980 pts/1      S     01:23   0:00 sleep 2312
ujjwalt+  10739  0.0  0.0  17812  2288 pts/1      S+    01:23   0:00 grep --color=
auto sleep
ujjwaltyagi@ujjwaltyagi:~$ pkill -f 10393
ujjwaltyagi@ujjwaltyagi:~$ ps aux | grep sleep
ujjwalt+  10393  0.0  0.0  16956  1980 pts/0      S+    01:22   0:00 sleep 2345
ujjwalt+  10542  0.0  0.0  16956  1980 pts/1      S     01:22   0:00 sleep 2312
ujjwalt+  10704  0.0  0.0  16956  1980 pts/1      S     01:23   0:00 sleep 2312
ujjwalt+  10889  0.0  0.0  17812  2288 pts/1      S+    01:23   0:00 grep --color=
auto sleep
```

Adjust process priority at start

```
nice -n 10 ./compute-heavy.sh
```

Explanation: Launches `compute-heavy.sh` with niceness 10 (lower priority).

```
ujjwaltyagi@ujjwaltyagi:~$ nice -n 10 ./gcd_lcm.sh 34 56 &
[1] 3511
ujjwaltyagi@ujjwaltyagi:~$ GCD(34,56) = 2
LCM(34,56) = 952
```

Change priority of running process

```
renice +5 -p 12345
renice -n -10 -p 12345
renice -n 5 -p 12345
```

Explanation: `renice` changes niceness of PID 12345 (positive increases nice value = lower priority).

```
ujjwaltyagi@ujjwaltyagi:~$ nice -n 10 sleep 300 &
[2] 3531
ujjwaltyagi@ujjwaltyagi:~$ sudo renice -n 5 -p 3531
[sudo] password for ujjwaltyagi:
3531 (process ID) old priority 10, new priority 5
ujjwaltyagi@ujjwaltyagi:~$ █
```

View process open files and resources

```
lsof -p 12345
```

Explanation: Lists files, sockets, and resources opened by PID 12345.

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 450 &
[4] 3540
ujjwaltyagi@ujjwaltyagi:~$ lsof -p 3540
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sleep  3540 ujjwaltyagi cwd   DIR    8,2      4096  920385 /home/ujjwaltyagi
sleep  3540 ujjwaltyagi rtd   DIR    8,2      4096     2 /
sleep  3540 ujjwaltyagi txt   REG    8,2     35336  525644 /usr/bin/sleep
sleep  3540 ujjwaltyagi mem   REG    8,2  14596880  526973 /usr/lib/locale/loc
le-archive
sleep  3540 ujjwaltyagi mem   REG    8,2   2125328  535783 /usr/lib/x86_64-linu
x-gnu/libc.so.6
sleep  3540 ujjwaltyagi mem   REG    8,2     613  567516 /usr/share/locale-la
ngpack/en/LC_MESSAGES/coreutils.mo
sleep  3540 ujjwaltyagi mem   REG    8,2   236616  535601 /usr/lib/x86_64-linu
x-gnu/ld-linux-x86-64.so.2
sleep  3540 ujjwaltyagi  0u   CHR   136,1      0t0      4 /dev/pts/1
sleep  3540 ujjwaltyagi  1u   CHR   136,1      0t0      4 /dev/pts/1
sleep  3540 ujjwaltyagi  2u   CHR   136,1      0t0      4 /dev/pts/1
uijwaltyagi@uijwaltyagi:~$
```

Show process tree for a PID

```
ps f -p 12345
```

Explanation: Shows process hierarchy for PID 1 (or change PID as needed).

```
ujjwaltyagi@ujjwaltyagi:~$ sleep 700 &
[1] 3558
ujjwaltyagi@ujjwaltyagi:~$ ps f -p 3558
  PID TTY      STAT   TIME COMMAND
 3558 pts/0    S      0:00 sleep 700
ujjwaltyagi@ujjwaltyagi:~$
```

📊 Observations & Results

// ...existing observations and image paths stay exactly the same...

🎓 Conclusion

This practical lab demonstrated essential Linux process management concepts through hands-on examples.

📘 References & Resources

- 📘 man pages: ps, top, kill, nice, renice

-  Linux kernel documentation
 -  System administration guides
-

~ End of Lab Report ~