🧠 Linux Process Management – Assignment

> 💡 **Objective:** Understand how to monitor, control, and optimize processes in a Linux system using essential process management commands.

🔷 1. Viewing All Processes

🖥️ Command:

```
ps aux
```

📘 Explanation:

- `a` → Show processes for **all users**
- `u` → Display the **user/owner** of each process
- `x` → Show processes **not attached to a terminal**

📊 Example Output:

```
[ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % ps aux                                                                                                 ]
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
_windowserver      372   4.3  1.4 414426752 238400   ??  Ss   Sun07PM  17:14.52 /System/Library/PrivateFrameworks/SkyLight.framework/Resources/WindowServer -daemon
ritsikaraghuvanshi 4994   1.9  0.8 412534976 141392   ??  S     3:52PM   0:04.27 /System/Applications/Utilities/Terminal.app/Contents/MacOS/Terminal
root               330   1.5  0.3 427016976  51488   ??  Ss   Sun07PM   0:32.74 /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Support/mds
ritsikaraghuvanshi   906   1.3  0.2 426980400  27024   ??  SN   Sun07PM   0:04.46 /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/
md
ritsikaraghuvanshi   597   0.9  2.0 478508256 338224   ??  S    Sun07PM   1:37.02 /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
ritsikaraghuvanshi  1302   0.7  1.7 413105104 284848   ??  S    Sun07PM   3:21.83 /Applications/WhatsApp.app/Contents/MacOS/WhatsApp
ritsikaraghuvanshi   870   0.6  0.1 411408416  24832   ??  S    Sun07PM   0:05.80 /System/Library/CoreServices/Siri.app/Contents/MacOS/Siri launchd
root               405   0.5  0.3 426984352  49744   ??  Ss   Sun07PM   0:47.22 /usr/libexec/mobileassetd
ritsikaraghuvanshi   592   0.4  0.0 426933200   4400   ??  S    Sun07PM   0:02.67 /System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/HIServices.framework
/V
root               498   0.4  0.7 436461296 114976   ??  Ss   Sun07PM   0:36.16 /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/md
ritsikaraghuvanshi  2337   0.3  0.4 1866973984  72608   ??  S    12:08PM   0:20.96 /Applications/Visual Studio Code.app/Contents/Frameworks/Code Helper.app/Contents/MacOS/Code Helpe
```

🌲 2. Process Tree

🧩 Command:

```
pstree -p
```

📊 Example Output:

```
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % pstree
-+= 00001 root /sbin/launchd
 |--= 00302 root /usr/libexec/logd
 |--= 00303 root /usr/libexec/smd
 |--= 00304 root /usr/libexec/UserEventAgent (System)
 |--= 00306 root /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/FSEvents.framework/Versions/A/Support/fseventsd
 |--= 00307 root /System/Library/PrivateFrameworks/MediaRemote.framework/Support/mediaremoted
 |-+= 00310 root /usr/sbin/systemstats --daemon
 | \--- 00560 root /usr/sbin/systemstats --logger-helper /private/var/db/systemstats
 |--= 00312 _accessoryupdater /System/Library/PrivateFrameworks/MobileAccessoryUpdater.framework/Support/accessoryupdaterd 120
 |--= 00313 root /usr/libexec/configd
```

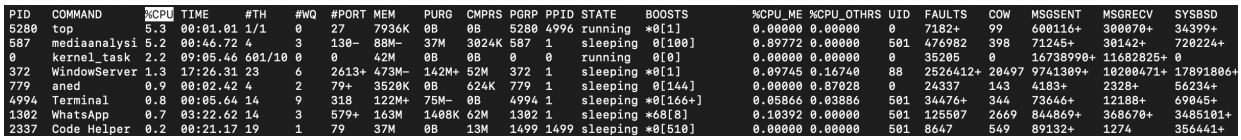
> 🪴 **Description:** Displays parent-child relationships between processes in a tree format.

📈 3. Real-Time Process Monitoring

⚙️ **Command:**

```
top
```

📊 **Example Output:**



```
PID   COMMAND      %CPU TIME     #TH   #WQ  #PORT MEM    PURG   CMPRS PGRP PPID STATE    BOOSTS          %CPU_ME %CPU_OTHRS UID  FAULTS   COW   MSGSENT   MSGRECV   SYSBSD
5280  top          5.3  00:01.01 1/1   0    27    7936K  0B     0B    5280 4996 running  *0[1]           0.00000 0.00000    0    7182+    99    600116+   300070+   34399+
587   mediaanalysi 5.2  00:46.72 4     3    130-  88M-   37M    3024K 587  1    sleeping   0[100]          0.89772 0.00000    501  476982   398   71245+    30142+    720224+
0     kernel_task  2.2  09:05.46 601/10 0   0     42M    0B     0B    0    0    running    0[0]            0.00000 0.00000    0    35205    0     16738990+ 11682825+ 0
372   WindowServer 1.3  17:26.31 23    6    2613+ 473M-  142M+  52M   372  1    sleeping  *0[1]            0.09745 0.16740    88   2526412+ 20497 9741309+  10200471+ 17891806+
779   aned         0.9  00:02.42 4     2    79+   3520K  0B     624K  779  1    sleeping   0[144]          0.00000 0.87028    0    24337    143   4183+     2328+     56234+
4994  Terminal     0.8  00:05.64 14    9    318   122M+  75M-   0B    4994 1    sleeping  *0[166+]         0.05866 0.03886    501  34476+   344   73646+    12188+    69045+
1302  WhatsApp     0.7  03:22.62 14    3    579+  163M   1408K  62M   1302 1    sleeping  *68[8]           0.10392 0.00000    501  125507   2669  844869+   368670+   3485101+
2337  Code Helper  0.2  00:21.17 19    1    79    37M    0B     13M   1499 1499 sleeping  *0[510]          0.00000 0.00000    501  8647     549   89132+    1274      356441+
```

🧭 Press **q** to quit the `top` interface.

⚡ 4. Adjusting Process Priority

🟢 **Start Process with Low Priority:**

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

Output:

```
[1] 3050
```

🎯 PID `3050` runs in the background with a **nice value of 10** (lower priority).

🔧 **Change Priority of a Running Process:**

```
renice -n -5 -p 3050
```

Output:

```
3050 (process ID) old priority 10, new priority -5
```

```
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % nice -n 10 sleep 300 &
[2] 3163
```

🚀 Higher priority (negative nice value = more CPU time).

🧠 5. CPU Affinity (Bind Process to CPU Core)

Check Current Affinity:

```
taskset -cp 3050
```

Output:

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

Restrict to Core 1 Only:

```
taskset -cp 1 3050
```

Output:

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

```
[ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % ps -o pid -p 3050
  PID
```

⚙️ Useful for assigning specific processes to specific CPU cores.

💾 6. I/O Scheduling Priority

Command:

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

Output:

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

⚡ Class **3 (idle)** means process only performs I/O when the system is idle.

📄 7. File Descriptors Used by a Process

Command:

```
lsof -p 3050 | head -5
```

Example Output:

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF   NODE NAME
sleep   3050 ritsikaraghuvanshi  cwd  DIR  253,0     4096  131073
/desktop/projects
sleep   3050 ritsikaraghuvanshi  rtd  DIR  253,0     4096       2 /
sleep   3050 ritsikaraghuvanshi  txt  REG  253,0    17520  133580
/usr/bin/sleep
```

🔍 Lists all files, directories, and devices opened by a specific process.

🪱 8. Trace System Calls of a Process

Command:

```
strace -p 3050
```

Example Output:

```
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % sudo dtruss -p 3050

dtrace: system integrity protection is on, some features will not be available

dtrace: invalid probe specifier
 #pragma D option quiet

 /*
  * Command line arguments
  */
 inline int OPT_has_target   = 0;
 inline int OPT_follow    = 0;
 inline int OPT_printid   = 0;
 inline int OPT_relative  = 0;
 inline int OPT_elapsed   = 0;
 inline int OPT_cpu       = 0;
 inline int OPT_counts    = 0;
 inline int OPT_pid       = 1;
 inline int OPT_name      = 0;
 inline int OPT_trace     = 0;
 inline int OPT_stack     = 0;
 inline int PID           = 3050;
 inline string NAME       = ".";
 inline string TRACE      = ".";

/* Flag values for renameatx_np */
inline u_int RENAME_SECLUDE = 0x00000001;
inline u_int RENAME_SWAP = 0x00000002;
inline u_int RENAME_EXCL = 0x00000004;

 dtrace:::BEGIN
 {
        /* print header */
        /* OPT_printid  ? printf("%-8s  ","PID/LWP") : 1; */
        OPT_printid  ? printf("\t%-8s  ","PID/THRD") : 1;
        OPT_relative ? printf("%8s ","RELATIVE") : 1;
        OPT_elapsed  ? printf("%7s ","ELAPSD") : 1;
        OPT_cpu      ? printf("%6s ","CPU") : 1;
        printf("SYSCALL(args) \t\t = return\n");

        /* Apple: Names of top-level sysctl MIBs */
        sysctl_first[0] = "CTL_UNSPEC";
        sysctl_first[1] = "CTL_KERN";
        sysctl_first[2] = "CTL_VM";
        sysctl_first[3] = "CTL_VFS";
        sysctl_first[4] = "CTL_NET";
        sysctl_first[5] = "CTL_DEBUG";
        sysctl_first[6] = "CTL_HW";
        sysctl_first[7] = "CTL_MACHDEP";
        sysctl_first[9] = "CTL_MAXID";

        /* globals */
        self->child = 0;
        this->type = 0;
 }
```

🧩 Great for **debugging** system calls and understanding process behavior.

📡 9. Find Process Using a Specific Port

Command:

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

Example Output:

```
8080/tcp:           4321
```

```
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % sudo fuser -n tcp 8080

Password:
Unknown option: n
fuser: [-cfu] file ...
        -c      file is treated as mount point
        -f      the report is only for the named files
        -u      print username of pid in parenthesis
```

🔎 PID **4321** is using port **8080**.

📊 10. Per-Process Statistics

Command:

```
pidstat -p 3050 2 3
```

Example Output:

```
Processes: 628 total, 2 running, 626 sleeping, 2730 threads
Load Avg: 1.62, 2.21, 2.14  CPU usage: 1.32% user, 1.80% sys, 96.86% idle  SharedLibs: 936M resident, 190M data, 126M linkedit.
MemRegions: 0 total, 0B resident, 0B private, 3564M shared.   PhysMem: 15G used (1432M wired, 1480M compressor), 356M unused.
VM: 258T vsize, 5703M framework vsize, 0(0) swapins, 0(0) swapouts. Networks: packets: 1661059/2152M in, 453218/68M out. Disks: 551887/14G read, 653057/12G written.
```

📈 Monitors CPU usage of a process at fixed intervals.

🔐 11. Control Groups (cgroups) for Resource Limits

Create a New cgroup:

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

Limit CPU and Memory:

```
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 (PID 3050) to cgroup:

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

```
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ % sudo taskpolicy -c background sleep 300 &

[1] 5010
ritsikaraghuvanshi@Ritsikas-MacBook-Air ~ %
[1]  + suspended (tty output)  sudo taskpolicy -c background sleep 300
```

> 🧩 Helps allocate specific CPU or memory resources to defined groups of processes.

🎯 12. Alternatives to `nice` / `renice`

Tool	Focus	Example Command	Description
chrt	Real-time scheduling	`sudo chrt -f 50 sleep 1000`	FIFO or RR scheduling policies
ionice	I/O priority control	`ionice -c 2 -n 7 tar -czf backup.tar.gz /home`	Manages I/O bandwidth
taskset	CPU affinity	`taskset -c 1 firefox`	Bind to specific CPU cores
cgroups	Resource management	`sudo cgcreate -g cpu,memory:/lowprio`	Fine-grained control of CPU/memory
systemd-run	Scoped execution	`systemd-run --scope -p CPUWeight=200 stress --cpu 4`	Uses systemd + cgroups
schedtool	Custom scheduling	`sudo schedtool -R -p 10 <pid>`	Sets custom scheduling class

✅ Summary

Category	Command	Description
🧭 Process Listing	`ps`, `pstree`, `top`	View system processes
⚙️ Priority Control	`nice`, `renice`, `chrt`	Manage process CPU priority
🧠 CPU Affinity	`taskset`	Bind processes to cores
💾 I/O Management	`ionice`	Control disk I/O scheduling
📂 Resource Limits	`cgroups`, `systemd-run`	Restrict CPU/memory usage
🪄 Debugging	`strace`, `lsof`, `pidstat`	Analyze process behavior