Lab 6 - Top Command

The top **command is a tool for displaying system-performance information. It dynamically shows administrators which processes are consuming processor and memory resources.**

The **top** command allows changing the process **priority** (**Nice Value**). By pressing key **r** then, enter the process ID and press **Enter**. After that, the program prompts for a new **nice** value. Enter a new value and press **Enter**.

```
top - 14:05:08 up 1 min,  1 user,  load average: 2.56, 1.69, 0.67
Tasks: 281 total,   1 running, 280 sleeping,   0 stopped,   0 zombie
%Cpu(s):  8.8 us,  3.0 sy,  0.0 ni, 88.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3928.7 total,    499.8 free,   1481.0 used,   1948.0 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   2197.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2190 bosko     20   0 4507960 369540 129716 S  19.3   9.2   0:07.05 gnome-shell
   1544 bosko     20   0 1012080  86956  50644 S   5.0   2.2   0:01.67 Xorg
   7510 bosko     20   0 1142728  72280  48040 S   4.0   1.8   0:01.25 nautilus
   6555 bosko     20   0  817396  50692  38188 S   2.0   1.3   0:00.46 gnome-termin+
    932 root      20   0 1094548  41044  18844 S   0.3   1.0   0:02.78 snapd
   1207 mysql     20   0 2077480 385408  35440 S   0.3   9.6   0:00.77 mysqld
   1509 bosko      9 -11 1674144  19956  15216 S   0.3   0.5   0:00.99 pulseaudio
   1813 bosko     20   0  158232   2708   2340 S   0.3   0.1   0:00.10 VBoxClient
   2756 bosko     20   0  980444  78352  48848 S   0.3   1.9   0:02.74 snap-store
      1 root      20   0  168184  12196   8440 S   0.0   0.3   0:01.00 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
```

The five lines displayed at the top of top:

The first line

- top displays uptime information
- Tasks displays process status information
- %Cpu(s) displays various processor values
- MiB Mem displays physical memory utilization
- MiB Swap displays virtual memory utilization

```
top - 23:01:54 up 3 min,  1 user,  load average: 1.21, 0.57, 0.22
Tasks: 221 total,   2 running, 219 sleeping,   0 stopped,   0 zombie
%Cpu(s): 94.7 us,  4.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.7 hi,  0.0 si,  0.0 st
MiB Mem :   3898.5 total,   1737.0 free,   1142.0 used,   1019.5 buff/cache
MiB Swap:   3898.0 total,   3898.0 free,      0.0 used.   2509.6 avail Mem
```

Tasks:

The second line is the **Tasks** output, and it's broken down into five states. These five states display the status of processes on the system:

- total shows the sum of the processes from any state.
- running shows how many processes are handling requests, executing normally, and have CPU access.
- sleeping indicates processes awaiting resources, which is a normal state.
- stopped reports processes exiting and releasing resources; these send a termination message to the parent process.
- zombie refers to a process waiting for its **parent** process to release it; it may become orphaned if the parent exits first. Zombie processes usually mean an application or service didn't exit gracefully.

Example: Tasks: 220 total, 3 running, 217 sleeping, 0 stopped, 0 zombie

Values related to processor utilization are displayed on the third line. They provide insight into exactly what the CPUs are doing.

- us is the percent of time spent running user processes.
- sy is the percent of time spent running the kernel.
- **ni is the percent of time spent running processes with manually configured nice values.**
- id is the percent of time idle (if high, CPU may be overworked).
- wa is the percent of wait time (if high, CPU is waiting for I/O access).
- hi is the percent of time managing hardware interrupts.
- si is the percent of time managing software interrupts.
- st is the percent of virtual CPU time waiting for access to physical CPU.

Values such as id, wa, and st help identify whether the system is overworked.

Example: %Cpu(s): 19.3 us, 4.0 sy, 0.0 ni, 74.7 id, 0.0 wa, 0.3 hi, 1.7 si, 0.0 st

MiB Memory : displays physical memory utilization. This value is based on the total amount of physical RAM installed on the system.

Example: MiB Mem: 3898.5 total, 385.2 free, 1167.0 used, 2346.2 buff/cache

Note: The term *mebibyte* (and similar units, such as kibibytes and gibibytes) differs slightly from measurements such as megabytes. Mebibytes are based on 1024 units, and megabytes are based on 1000 units (decimal).

- total shows total installed memory.
- free shows available memory.
- used shows consumed memory.

- buff/cache shows the amount of information buffered to be written.

MiB SwapThe process of swapping data back and forth between physical RAM and storage drives is time-consuming and uses system resources, so it's best to minimize the use of virtual memory.

Example: MiB Swap: 3898.0 total, 3898.0 free, 0.0 used, 2433.1 avail Mem

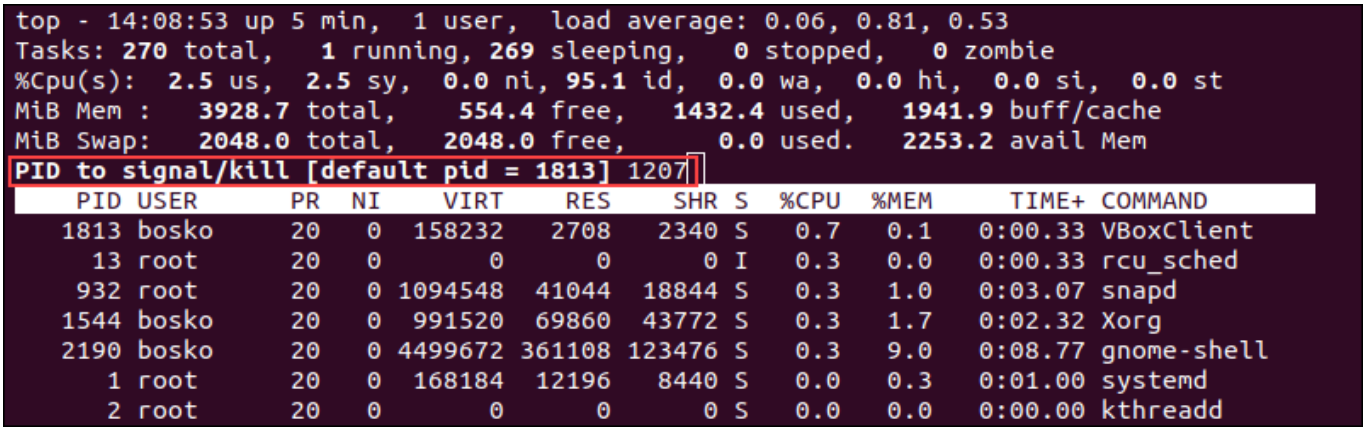
- total shows total swap space.
- free shows available swap space.
- **used** shows consumed swap space.

The information of top command Process Command Line:

- **PID:** Shows task's unique process id.
- **PR:** The process's priority. The lower the number, the higher the priority.
- **VIRT:** Total virtual memory used by the task.
- **USER:** User name of owner of task.
- **%CPU:** Represents the CPU usage.
- **TIME+:** CPU Time, the same as 'TIME', but reflecting more granularity through hundredths of a second.
- **SHR:** Represents the Shared Memory size (kb) used by a task.
- **NI:** Represents a Nice Value of task. A Negative **Nice** value implies higher priority, and positive **Nice** value means lower priority.
- **%MEM:** Shows the Memory usage of task.
- **RES:** How much physical RAM the process is using, measured in kilobytes.
- **COMMAND:** The name of the command that started the process.

Send a Signal

Use the **top** command to send any signal to a running process. Press the **k** key and enter the process PID. **top** gives you a chance to type the signal you want to send. Not entering a specific signal kills the process.



Filter Processes by Specific User

The `-u` option allows you to display all user-specific processes. Press the **u** key while `top` is running or use the following syntax: top -u [user name]