CPU Utilization in Linux

One aspect of determining the system's efficiency is CPU performance.

Monitoring CPU performance is critical to debugging system processes, making system decisions, handling system resources, and inspecting and evaluating systems in real- time.

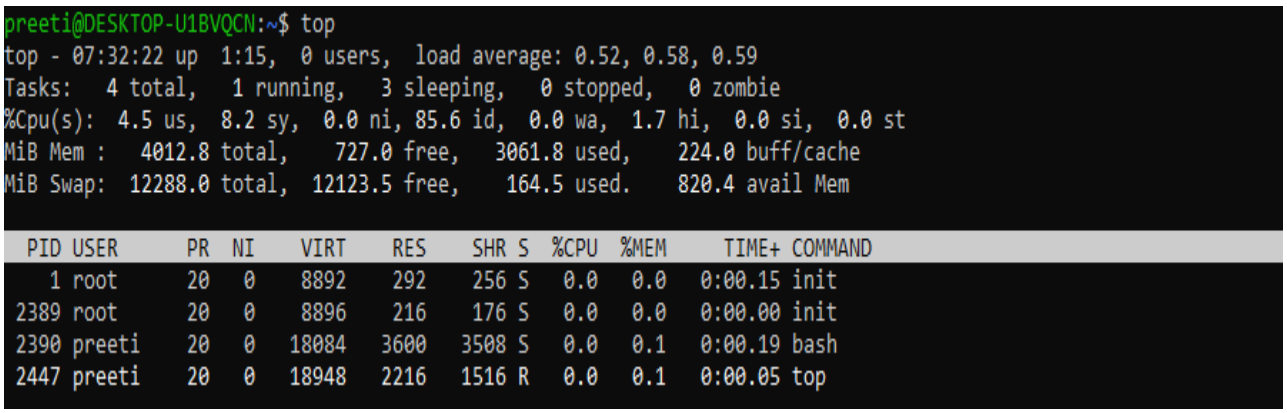
A variety of tools are available to track and view CPU performance. The operating system consists of built-in system calls that rely on these tools to extract performance readings. Now we talk about the great CPU usage monitoring tool and how to use it in any Linux distribution.

Used Top to Check CPU Utilization

With the help of the **top** command, we can monitor the system in real-time. When we execute the top command, it will provide us the summary of the system along with the list of threads and process which are presently being managed by the Linux kernel. It also offers various options in order to change its behaviour and perform several actions.

Syntax

1. $ top



In the above output, the main line to focus on is the line number third. The output values are briefly described. Each value indicates how much time the CPU spends performing a task.

- **us:** The amount of time it takes to run the process for individual users in the "user space."
- **sy:** sy is the time spent running "kernel space" processes.
- **ni:** ni is the time spent to run the process with a custom (manually set) excellent value.

- **id:** id is the time spent idle.
- **wa:** wa means the amount of time spent waiting for I/O request to complete.
- **hi:** hi is the time spent servicing hardware interrupts.
- **si:** si is the time spent servicing software interrupts.
- **st:** st is the time lost for running a virtual machine, which is also called "steal time."

Check CPU Utilization Using htop Command

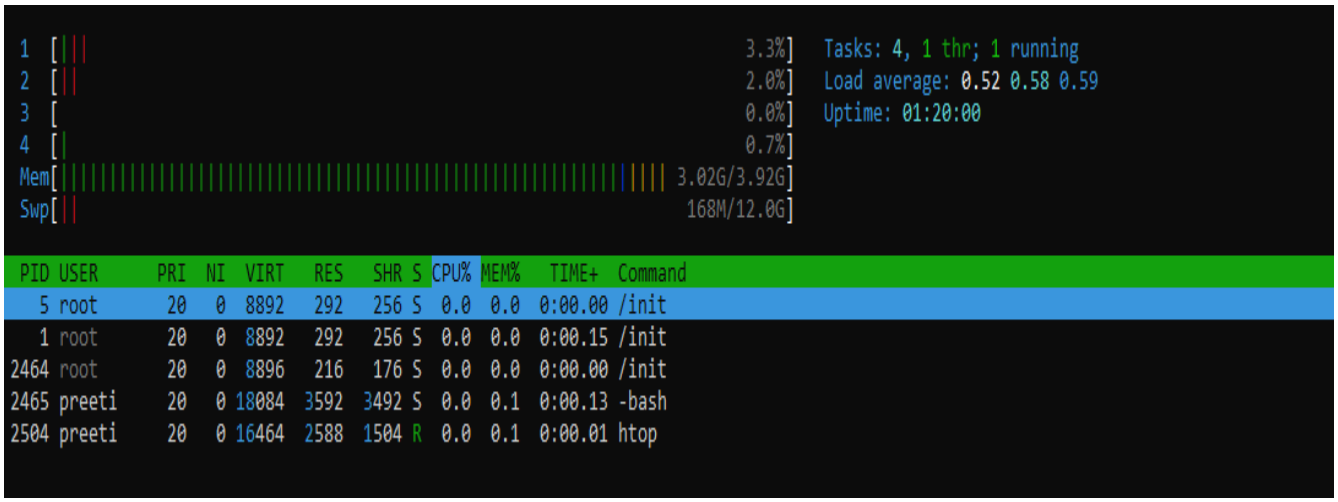
The **top** and **htop** commands are the same. In terms of system monitoring both commands offer the same functionality. On the other hand, the **htop** command offers a better quality-of-life experience. The **htop's** default display is more user-friendly. As compared to the top, the **htop** UI has better quality. In this, we can also scroll horizontal as well as vertical.

In most of the distros, the top is installed by default, and we have to install the **htop** manually. The best way to install the **htop** is using snap because it works well on any of the Linux distros. Following is the syntax which we use to install **htop**:

1. $ sudo snap install htop

Syntax of htop:

1. $ htop



Check CPU Utilization Using iostat

The **iostat** command reports CPU and I/O usage statistics. This command is simple with simple output. Although, it will only report data on the moment the

command was run. **iostat** does not provide real-time system monitoring, unlike top and **htop.**

The **iostat** tool comes as a part of the sysstat package. Mostly this command is available on every Linux distro. Suppose we have installed the **sysstat** package, then run.

Launch iostat:

1. $ iostat

```
preeti@DESKTOP-U1BVQCN:~$ iostat
Linux 4.4.0-18362-Microsoft (DESKTOP-U1BVQCN)   04/09/21        _x86_64_        (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          5.08    0.00    5.15    0.00    0.00   89.78

Device            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
```

If we want to know more detail, then we can use the -c flag to check the CPU usage of system processes, I/O wait, idle time, and user processes.

Syntax

1. $ iostat -c

```
preeti@DESKTOP-U1BVQCN:~$ iostat -c
Linux 4.4.0-18362-Microsoft (DESKTOP-U1BVQCN)   04/09/21        _x86_64_        (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          5.07    0.00    5.15    0.00    0.00   89.78
```

The "-x" flag can be used to get more details statistics. The "-t" flag specifies the number of times each report must be shown.

1. $ iostat -xtc 6 2

```
preeti@DESKTOP-U1BVQCN:~$ iostat -xtc 6 2
Linux 4.4.0-18362-Microsoft (DESKTOP-U1BVQCN)   04/09/21        _x86_64_        (4 CPU)

04/09/21 07:41:41
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          5.17    0.00    5.45    0.00    0.00   89.38

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz
aqu-sz  %util
```

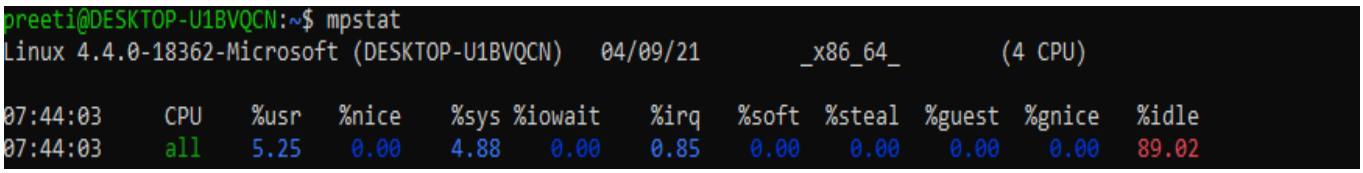
Use mpstat to Check CPU Utilization

The **mpstat** tool is a part of the **sysstat** package. The tool reports the use of an individual processors or processor cores.

If we want to use the **mpstat** command, then it is a must that the **sysstat** package is installed in our system. Let's say we already have the package installed then proceed.

Launch mpstat:

1. $ mpstat



Now we will explain each value in detail, and each value defines the usage of CPU at a specific stage.

- **%usr:** It means the user-level CPU usage.
- **%nice:** It implies the usage of CPU by "nice" user processes.
- **%sys:** %sys means the CPU usage by the kernel.
- **%iowait:** Waiting for disk read/write is expressed as %iowait.
- **%irq:** Handling hardware interrupts is referred to as %irq.
- **%soft:** Handling software interrupts is known as %soft.
- **%steal:** The term %steal refers to having a wait for a hypervisor in order to handle virtual processors.
- **$idle:** %idle means standing idle.

Use sar to Check CPU Utilization

Using the **sar** command, we can gather and report system activity information. This command will provide you a simple and short report related to CPU utilization.

Sar command may also be used to provide the CPU information at a regular interval (in seconds). It is not a real-time report; it is still better to work with it.

1. $ sar **<interval_seconds>**

We are also able to define the number of times the **sar** will print the output. In this below example, the **sar** will print the output at 4 seconds intervals, for 8 times.

1. $ sar 4 8

```
preeti@DESKTOP-U1BVQCN:~$ sar 4 8
Linux 4.4.0-18362-Microsoft (DESKTOP-U1BVQCN)   04/09/21        _x86_64_        (4 CPU)

07:46:57        CPU     %user     %nice   %system   %iowait    %steal     %idle
07:47:01        all      6.65      0.00      5.08      0.00      0.00     88.28
07:47:05        all      5.69      0.00      5.76      0.00      0.00     88.55
07:47:09        all      0.00      0.00      0.50      0.00      0.00     99.50
07:47:13        all      0.06      0.00      0.94      0.00      0.00     99.00
07:47:17        all      0.31      0.00      0.37      0.00      0.00     99.31
07:47:21        all      0.19      0.00      0.37      0.00      0.00     99.44
07:47:25        all      2.55      0.00      3.92      0.00      0.00     93.53
07:47:29        all      0.19      0.00      1.86      0.00      0.00     97.95
Average:        all      1.95      0.00      2.35      0.00      0.00     95.70
```

Check CPU Utilization Using nmon Command

nmon is a system administration tool that we used to get the information about CPU, Top process memory, etc. We can install the nmon command by using the following command:

1. $ nmon