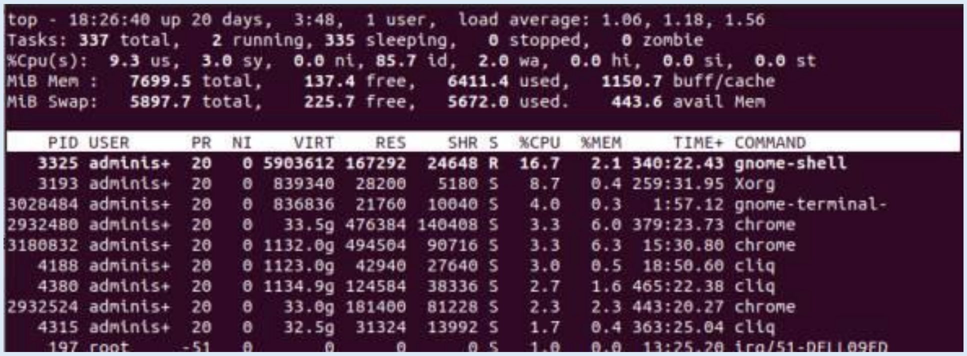
RAJLINUX

LINUX MONITORING AND EXPLANATION

How to Monitor System Activity in linux | top Command

top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the systemand the list of processes or threads which are currently managed by the Linux Kernel. As soon as you will run this command it will open an interactive command mode where the top half portion will contain the statistics of processes and resource usage. And Lower half contains a list of the currently running processes. Pressing q will simply exit the command mode.

top command:--



```
top - 18:26:40 up 20 days,  3:48,  1 user,  load average: 1.06, 1.18, 1.56
Tasks: 337 total,   2 running, 335 sleeping,   0 stopped,   0 zombie
%Cpu(s):  9.3 us,  3.0 sy,  0.0 ni, 85.7 id,  2.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7699.5 total,    137.4 free,   6411.4 used,   1150.7 buff/cache
MiB Swap:   5897.7 total,    225.7 free,   5672.0 used.    443.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3325 adminis+  20   0 5903612 167292  24648 R  16.7   2.1 340:22.43 gnome-shell
   3193 adminis+  20   0  839340  28200   5180 S   8.7   0.4 259:31.95 Xorg
3028484 adminis+  20   0  836836  21760  10040 S   4.0   0.3   1:57.12 gnome-terminal-
2932480 adminis+  20   0   33.5g 476384 140408 S   3.3   6.0 379:23.73 chrome
3180832 adminis+  20   0 1132.0g 494504  90716 S   3.3   6.3  15:30.80 chrome
   4188 adminis+  20   0 1123.0g  42940  27640 S   3.0   0.5  18:50.60 cliq
   4380 adminis+  20   0 1134.9g 124584  38336 S   2.7   1.6 465:22.38 cliq
2932524 adminis+  20   0   33.0g 181400  81228 S   2.3   2.3 443:20.27 chrome
   4315 adminis+  20   0   32.5g  31324  13992 S   1.7   0.4 363:25.04 cliq
    197 root     -51   0       0      0      0 S   1.0   0.0  13:25.20 irq/51-DELL09ED
```

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.

What Is Linux Load Average?

Linux load average is a metric that shows the number of tasks currently executed by the CPU and tasks waiting in the queue.
Unlike CPU usage, which measures system performance at a specific point in time, the load average shows performance over a particular period. The number of processes running on the system changes constantly, and the load average displays that change.
The metric is expressed as the average number of processes in a runnable state over the last 1, 5, or 15 minutes. A higher load average indicates higher resource usage
Checking with the uptime Command

RAJLINUX

Use the uptime command to check the load average for the past 1, 5, and 15 minutes.

uptime

uptime terminal output The output shows that the system has been running for 21 minutes since the last boot and that the number of active users is 1.

The load average for one user is:

0.79 for the past 1 minute.

0.32 for the past 5 minutes.

0.11 for the past 15 minutes.

The results are calculated by dividing the number of running and waiting processes by the number of available CPU cores.

Checking Load Average with cat

Another way to view the load average on a Linux system is by using the cat command. To print the load average in the first three columns, run the following:

cat /proc/loadavg

----

cat proc loadavg command terminal output

While the first three numbers show the load average, the last three represent:

The number of currently running processes: 3.

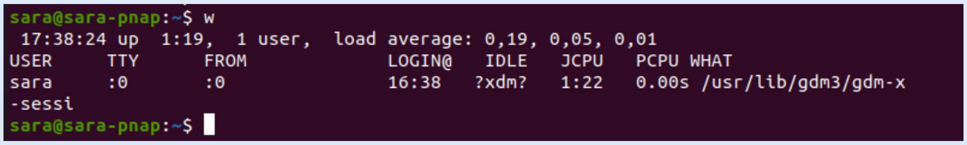
The total number of processes: 472.

The ID of the most recently created process: 26767.

Check Load Average with w

The w command also prints the load average in the first line of output

w command:--



```
sara@sara-pnap:~$ w
 17:38:24 up  1:19,  1 user,  load average: 0,19, 0,05, 0,01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
sara     :0       :0               16:38   ?xdm?   1:22   0.00s /usr/lib/gdm3/gdm-x
-sessi
sara@sara-pnap:~$ 
```

The first line shows info about currently logged-in users, including system time, uptime, number of users, and the average Linux load.

free Command and Meaning of the Values

The free command in Linux is a commonly used utility that provides information about the system’s memory usage, including both physical and virtual memory. When we run the free command in a terminal, it displays several values related to memory utilization. Here’s a breakdown of the meanings of these values:

total: This value represents the total amount of physical memory (RAM) in the system, measured in kilobytes (KB), megabytes (MB), or gigabytes (GB).

used: The used value shows the amount of physical memory currently in use by various processes and the operating system itself.

free: The free value indicates the amount of physical memory that is currently not being used.

shared: The shared value represents the memory that multiple processes share among themselves.
buffers: The buffers value shows the amount of memory used for buffering disk I/O operations.
cache: The cache value indicates the amount of memory used for caching frequently accessed data from the disk or other storage devices.
available: The available value represents an estimate of the memory that is available for new processes to allocate. It takes into account the memory used for buffers and cache, which can be freed if needed by other applications.

It’s important to note that the values reported by the free command are dynamic and can change as processes and applications allocate or release memory. The values are presented in kilobytes by default, but we can specify different units, such as megabytes or gigabytes, by using appropriate options with the free command (e.g., free -m for megabytes).

Meaning of –/+ buffers/cache

In order to know what the numbers mean, we must understand the virtual memory (VM) subsystem in Linux. Linux, like most modern OS, will use free RAM for caching, so Mem: free will almost always be very low. Caches get freed automatically if memory gets scarce, so they do not really matter. Therefore, the line -/+ buffers/cache shows how much memory is free when ignoring the cache. A Linux system is really low on memory if the free value in -/+ buffers/cache the line gets low:

Difference Between Buffer and Cache

The main distinction between buffer and cache, as reported by the free command, lies in their purpose and the type of data they manage. With respect to the storage devices, buffers read from or write to the disk, thereby optimizing disk I/O operations by temporarily storing the data. On the other hand, caches are focused on enhancing the performance of the system’s main memory by storing frequently accessed data.

While both buffer and cache aim to improve overall system performance, they operate at different levels within the memory hierarchy. Buffers work at the level of storage devices, whereas caches operate at the level of the main memory. Buffers help minimize disk access, while caches aim to reduce memory latency.

When we run the free command, the reported values for buffers and caches represent the amount of memory allocated for these purposes. Monitoring these values can provide insights into the usage of memory resources in our system. It also helps identify potential bottlenecks or areas of improvement.

To understand the concept of buffers, let’s try a little experiment with the reading of free command in Linux:

```
free -m
          total     used     free    shared   buffers    cached
Mem:        2897     1466     1430        0        32      1127
-/+ buffers/cache:      306      2590
```

Different ways to get file size in Linux

ls command

The 'ls' command is perhaps one of the most often used commands on the command line in Linux. It means "to list," as in "to list the files and folders from my current location." It's approximately the same as the

DOS/Windows command line option 'dir'. The man page for 'ls' will provide you with a wide variety of options that you can use with this command. Let's take a look at a few that show you the file size.

```
ls -l <file>
```

The -l options are used to get the size of the specified file.

```
ls -l *
```

-l options are used to get the size of all the files in the current directory.

```
ls -al *
```

-al option is used to get the size of all the files, including hidden files in the current directory.

```
ls -h -l <file>
```

-h option prints human-readable sizes of the files.

du command

The command du is used to obtain information about the disc utilization of specific files and folders. It works best with certain folders and has a lot of options for customizing the output to match your requirements. It usually gives the detail of sizes in terms of blocks.

```
du -h <path>
```

Print file sizes in human-readable form

```
du -s <path>
```

Get memory allocated summary of the file or the directory.

Netstat Command in Linux

`netstat` stands for network statistics. It allows users to display network-related information and diagnose various networking issues. The command has several options that can be combined to retrieve specific details.

Basic Syntax of `netstat`Command in Linux

Below is the general syntax of the netstat command:

```
netstat [options]
```

1) Show Both Listening and Non-listening Sockets Using netstat Command in Linux

-a -all : Show both listening and non-listening sockets. With the –interfaces option, show interfaces that are not up.

```
netstat -a | more
```

2) List All TCP Ports Using netstat Command in Linux

This command specifically lists all TCP ports, giving you information about the TCP connections your system is engaged in.

```
netstat -at
```

```
Swap:        4000        0      4000
```