How to use the vmstat command on Linux

Suppose your Linux or macOS computer is using virtual memory. Discover how it affects the system's physical memory usage, CPU and resource usage.

Suppose your Linux or macOS computer is using virtual memory. Discover how it affects the system's physical memory usage, CPU and resource usage.

Learn about the vmstat command in Linux

What is virtual memory?

The computer is equipped with a finite amount of physical memory, called Random Access Memory (RAM). This RAM needs to be managed by the kernel, shared between the operating system and any running applications. If these combined needs require more memory than the physical installation in the computer, what can the kernel do?

Linux and Unix-like operating systems like macOS can use the space on the hard drive to help them manage memory needs. A dedicated area for hard drive space called 'swap space', can be used as an extension of RAM. This is virtual memory.

The Linux kernel can write the contents of a memory block into this swap space and release that RAM to use for another process. Memory can be removed from the swap space and restored to RAM when necessary called paged out.

Of course, access speed for paged out memory is slower than memory stored in RAM. And that is not the only tradeoff. While virtual memory provides Linux with a way to manage memory needs, the use of virtual memory will increase the burden on other parts of the computer.

Your hard drive must read and write more. The kernel and CPU must do more when swap in (move data from swap space to main memory) and swap out (move main memory contents to disk swap, when space is available). main memory is full) memory, as well as keeping all disks spinning to meet the memory needs of many different processes.

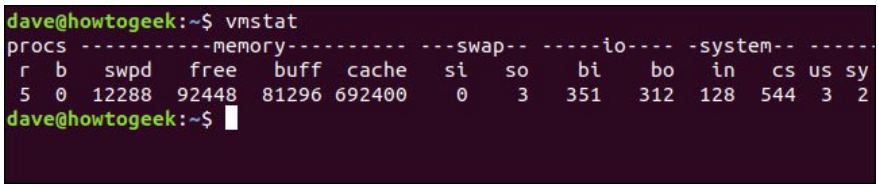
Linux provides a way for you to monitor all of these activities as vmstat commands, reporting virtual memory statistics.

Vmstat command

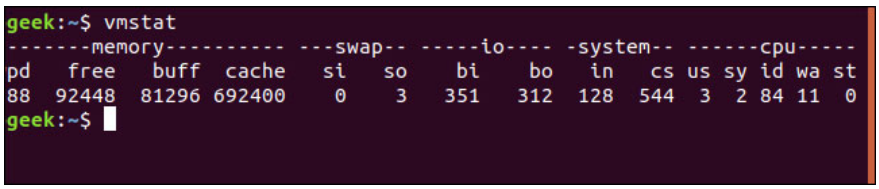
If you type the **vmstat** command without parameters, it will show you a set of values. These values ??are the average for each statistic, since the computer was last restarted. These figures are not current values.

```
vmstat
```

A short value table is displayed.



There are **Procs, Memory, Swap, IO, System** and **CPU** columns. The last column (rightmost column) contains CPU-related data.



Below is a list of data items in each column.

Procs

1. r: Number of processes that can be run. These are processes that are already running or waiting for a new CPU cycle.
2. b: Number of processes in continuous sleep state. In fact, these processes are not 'sleeping', but they are only blocking system calls (when the program calls a function or service that is in the kernel of the operating system) and cannot be interrupted until it executes. The current dynamic is completed. Typically, these processes occur when the device driver is waiting for some 'free' resources. Any disruption to these processes while in the queue is processed when the process continues to operate as usual.

Memory

1. swpd: Virtual memory capacity used. In other words, this is the amount of memory that has been swapped.
2. free: Idle memory capacity (currently not in use).
3. buff: The amount of memory used as a buffer.
4. cache: The amount of memory used to cache.

Swap

1. si: The amount of virtual memory is swapped in from the swap space.
2. so: The amount of virtual memory is swapped out into the swap space.

IO

1. Ball: Blocks received from a block device. The number of data blocks used to exchange virtual memory back to RAM.
2. bo: Blocks sent to a block device. The number of data blocks used to exchange virtual memory out of RAM and into swap space.

System

1. Print: The number of interruptions per second, including the clock.
2. cs: Number of context switches per second. The switch context is when the kernel changes from processing system mode to user mode processing.

CPU

These values ??are all percentage of total CPU time.

1. us: Time to run code is not kernel.
2. sy: Runtime of kernel code.
3. id: Idle time.
4. wa: Time to wait for input or output.

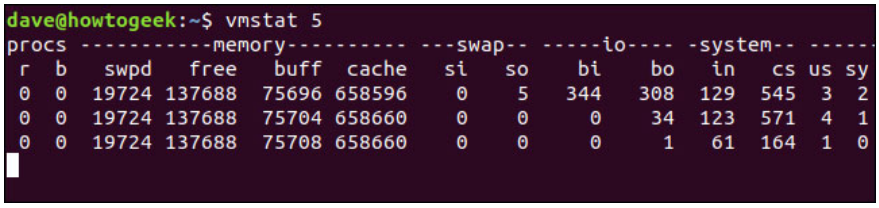
5. st: Time obtained from a virtual machine. This is the time a virtual machine must wait for the hypervisor to complete serving other virtual machines before it can go back and process the virtual machine.

Use a period of time

You can ask **vmstat to** provide regular updates for these metrics by using the **delay** value **.** The **delay** value is provided in seconds. To update statistics every 5 seconds, use the following command:

```
vmstat 5
```

Every 5 seconds **vmstat** will add another row of data to the table. You need to press `Ctrl` + `C` to stop this.



Use the count value

Using a **delay** value too low will put more pressure on the system. If you need to have quick updates to try to diagnose the problem, you should use the **count** value as well as the **delay** value **.**

The **count** value for **vmstat** indicates how many updates it needs to perform before exiting and returning you to the Command Prompt. If you do not provide the **count** value **, vmstat** will run until it is stopped by the `Ctrl` + `C` key combination.

Let **vmstat** provide an update every 5 seconds, but in just 4 updates, use the following command:

```
vmstat 5 4
```

After 4 updates, **vmstat** will stop.

```
dave@howtogeek:~$ vmstat 5 4
procs -----------memory---------- ---swap-- -----io---- -system-- -----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy
 4  0  19724 133704  75728 662092    0    5   342   306  128  545  3  2
 0  0  19724 133704  75736 662076    0    0     0     5  171  874  5  3
 0  0  19724 133704  75744 662068    0    0     0     6   84  293  1  1
 0  0  19724 133580  75744 662144    0    0    13     0   72  142  0  0
dave@howtogeek:~$
```

Change units

You can choose to display memory statistics and swap in kilobytes or megabytes using the **-S** (unit character) option. This option must be followed by one of the units **k, K, m** or **M.** They represent:

1. k: 1000 bytes
2. K: 1024 bytes
3. m: 1000000 bytes
4. M: 1048576 bytes

To update memory statistics and swaps every 10 seconds, displayed in megabytes, use the following command:

```
vmstat 10 -SM
```

```
dave@howtogeek:~$ vmstat 10 -S M
```

Memory and swap statistics are currently displayed in megabytes. Note that the **-S** option does not affect **IO** block statistics **.** They are always displayed in blocks.

```
dave@howtogeek:~$ vmstat 10 -S M
procs -----------memory---------- ---swap-- -----io---- -system-- -----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy
 5  0     19    102     76    668    0    0   328   293  126  533  3  2
 0  0     19    102     76    668    0    0     0     8   90  337  2  1
```

Memory is active and inactive

If you use the **-a** (active) option, the cache and cache columns are replaced with '**inact**' and '**active**' columns **.** As you might have guessed, these columns show the amount of memory inactive and inactive.

To see these two columns instead of **buff** and **cache** columns **,** include the **-a** option **,** as shown:

```
vmstat 5 -a -SM
```

```
dave@howtogeek:~$ vmstat 5 -a -S M
```

'**Inact**' and '**active**' columns are affected by the **-S** (unit character) option.

```
dave@howtogeek:~$ vmstat 5 -a -S M
procs -----------memory---------- ---swap-- -----io---- -system-- ------
 r  b   swpd   free  inact active   si   so    bi    bo   in   cs us sy
 2  0     19     84    591   1129    0    0   321   288  126  529  3  2
 0  0     19     84    591   1129    0    0     0    18   97  429  2  1
 1  0     19     84    591   1129    0    0     0     0  117  475  3  1
```

Fork

Switch **-f** displays the number of fork that has occurred since the computer was started. (Fork is an activity where a process creates a copy of itself).

In other words, this indicates the number of tasks that were started (and most of them closed) since the system was started. Each process started from the command line will increase this number. Every time a task or process generates or duplicates a new task, this number will increase.

```
vmstat -f
```

```
dave@howtogeek:~$ vmstat -f
         3659 forks
dave@howtogeek:~$
```

The display of fork does not update.

Slabinfo display

Kernel has its own memory management feature to manage the operating system and all applications.

As you can imagine, the kernel is allocating and releasing memory multiple times for many different types of data objects it must handle. To make this as effective as possible, it uses a system called slab. This is a form of cache record.

Memory is allocated, used and no longer needed for a specific type of kernel data object, which can be reused for another data object of the same type without the need for memory to be released and allocated. again. Imagine that the slab has been pre-allocated and measures the RAM segments for the kernel's specific needs.

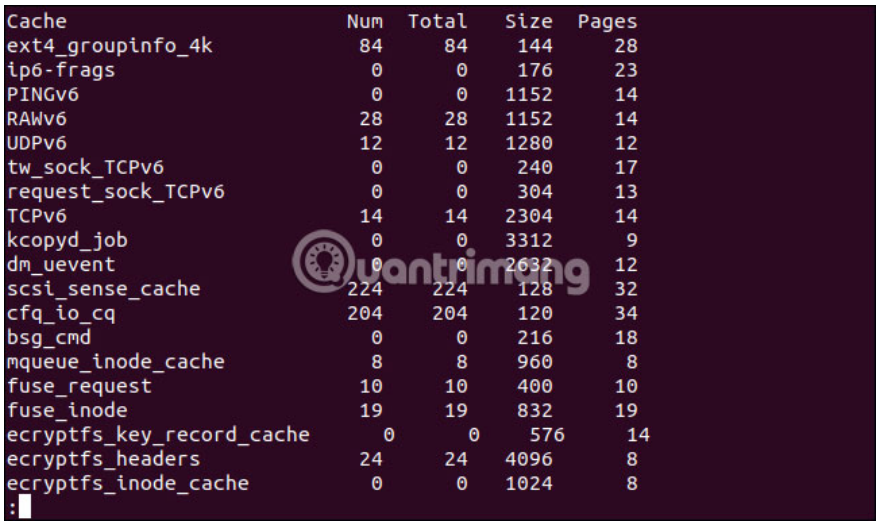
To see statistics for slab, use the **-m** (slab) option. You need to use **sudo** and will be prompted to enter a password. Since the output can be quite long, you should use the **less** option **.**

```
sudo vmstat -m | less
```

```
dave@howtogeek:~$ sudo vmstat -m | less
```

The output has 5 columns, including:

1. Cache: The name of the cache.
2. num: Number of objects currently active in this cache.
3. total: The total number of objects available in this cache.
4. size: Size of each object in the cache.
5. pages: The total number of memory pages that have (at least) an object currently associated with this cache.

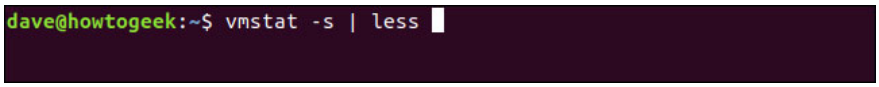


To exit **less** mode **,** press `q` .

Displays event counters and memory statistics

To display a page of event counters and memory statistics, use the **-s** (stats) option. Note that, the letter **s is** lower

```
vmstat -s
```



Although the statistics are reported in much the same way as the information generated by the default **vmstat** command, some of the data is divided in more detail.

For example, the default output combines both nice and non-nice user CPU time into the us column. The **-s** (stats) option lists these statistics separately.

```
      2041092 K total memory
      1178876 K used memory
      1170100 K active memory
       607840 K inactive memory
        79644 K free memory
        74324 K buffer memory
       708248 K swap cache
       483800 K total swap
        25100 K used swap
       458700 K free swap
        13320 non-nice user cpu ticks
          341 nice user cpu ticks
         6194 system cpu ticks
       349554 idle cpu ticks
        41806 IO-wait cpu ticks
            0 IRQ cpu ticks
          152 softirq cpu ticks
            0 stolen cpu ticks
      1277188 pages paged in
      1150904 pages paged out
:
```


Show drive statistics

You can get a similar drive statistics list using the **-d** (disk) option.

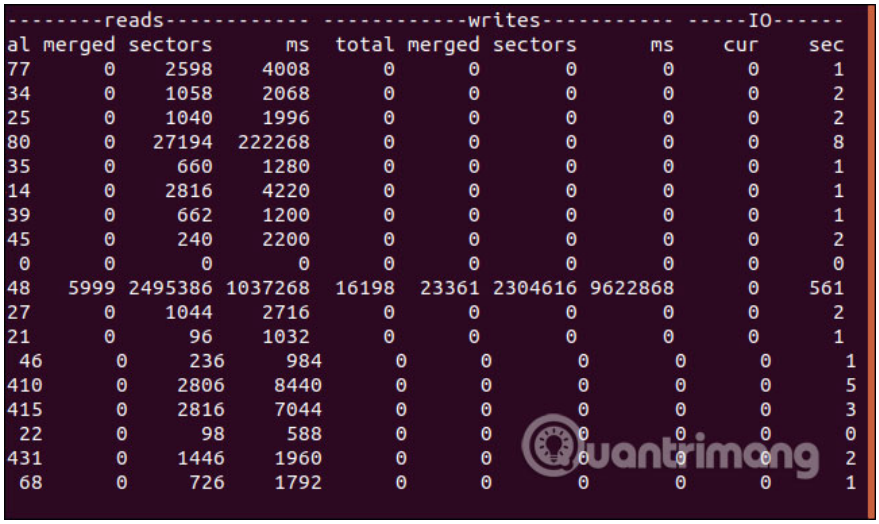
```
vmstat -d | less
```

```
dave@howtogeek:~$ vmstat -d | less
```

For each drive, there are 3 columns displayed, which are **Reads, Writes** and **IO.**

```
disk- ------------reads------------ ------------writes----------- -----
       total merged sectors      ms  total merged sectors      ms    cu
loop0    277      0    2598    4008      0      0       0       0     (
loop1    234      0    1058    2068      0      0       0       0     (
loop2    225      0    1040    1996      0      0       0       0     (
loop3  12580      0   27194  222268      0      0       0       0     (
loop4     35      0     660    1280      0      0       0       0     (
loop5    414      0    2816    4220      0      0       0       0     (
loop6     39      0     662    1200      0      0       0       0     (
loop7     45      0     240    2200      0      0       0       0     (
sr0        0      0       0       0      0      0       0       0     (
sda    43348   5999 2495386 1037268  16198  23361 2304616 9622868     (
loop8    227      0    1044    2716      0      0       0       0     (
loop9     21      0      96    1032      0      0       0       0     (
loop10    46      0     236     984      0      0       0       0
loop11   410      0    2806    8440      0      0       0       0
loop12   415      0    2816    7044      0      0       0       0
loop13    22      0      98     588      0      0       0       0
loop14   431      0    1446    1960      0      0       0       0
loop15    68      0     726    1792      0      0       0       0
:
```


IO is the rightmost column. Note that the **sec** column in **IO** is measured in seconds but time-based statistics in **Reads** and **Writes** columns are measured in milliseconds.



```
--------reads------------ ------------writes----------- -----IO------
al merged sectors      ms  total merged sectors      ms    cur    sec
77      0    2598    4008      0      0       0       0      0      1
34      0    1058    2068      0      0       0       0      0      2
25      0    1040    1996      0      0       0       0      0      2
80      0   27194  222268      0      0       0       0      0      8
35      0     660    1280      0      0       0       0      0      1
14      0    2816    4220      0      0       0       0      0      1
39      0     662    1200      0      0       0       0      0      1
45      0     240    2200      0      0       0       0      0      2
 0      0       0       0      0      0       0       0      0      0
48   5999 2495386 1037268  16198  23361 2304616 9622868      0    561
27      0    1044    2716      0      0       0       0      0      2
21      0      96    1032      0      0       0       0      0      1
 46     0     236     984      0      0       0       0      0      1
410     0    2806    8440      0      0       0       0      0      5
415     0    2816    7044      0      0       0       0      0      3
 22     0      98     588      0      0       0       0      0      0
431     0    1446    1960      0      0       0       0      0      2
 68     0     726    1792      0      0       0       0      0      1
```

This is the meaning of the columns:

Reads

1. total: The total number of read drives.
2. merged: The total number of group readings.
3. sectors: The total number of sectors read.
4. ms: Total time in milliseconds used to read data from the drive.

Writes

1. total: The total number of drive writes.
2. merged: The total number of records grouped.
3. sectors: The total number of sectors recorded.
4. ms = Total time, in milliseconds, was used to write data to the drive.

IO

1. cur: The number of reads or writes of the current drive.
2. sec: Time in seconds for any reading or writing being performed.

Show summary drive statistics

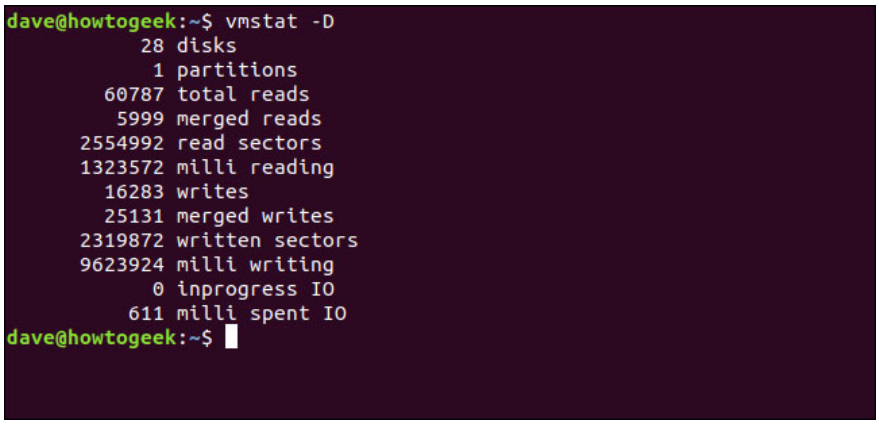
To quickly display summary statistics for disk activity, use the **-D** (disk-sum) option. Note that the letter **D** is capitalized.

```
vmstat -D
```

```
dave@howtogeek:~$ vmstat -D
```

The number of drives may look unusually high. The computer used as an example in this article is running Ubuntu. With Ubuntu, every time you install a Snap application, a file system created with the **squashfs pseudo-filesystem** will be attached to the device **/ dev / loop.**

The annoying thing is that these device items are counted as hard drive devices by many Linux commands and utilities.



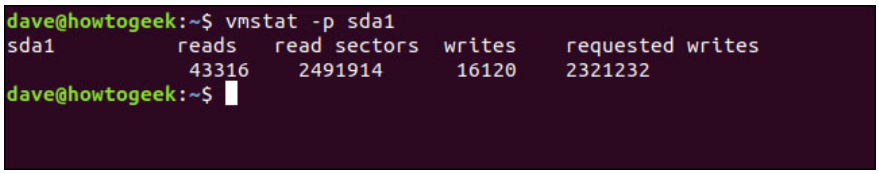
Show partition statistics

To see statistics related to a specific partition, use the **-p** (partition) option and provide the partition identifier as the command line parameter.

Here we will look at the partition **sda1.** The first digit indicates that this is the first partition on the **sda** device **,** this is the main hard drive for this computer.

```
vmstat -p sda1
```

The return information shows the total number of reads, writes the drive to and from that partition, as well as the number of sectors included in the read and write operations.



In-depth understanding of a problem is always encouraged. Sometimes you will try to solve the problem or not care about it anymore (because you just want to know how your computer works).

The **vmstat** command can give you a lot of useful information. Now, you know how to access this command and its meaning. You need to take some time to learn more and make some diagnoses, to really capture **vmstat.**

Good luck!