

# top – display tasks and system status in Unix

**top** is a [basic Unix command](#) which is very useful for observing the current state of your Unix system, by default presenting you the list of top users of your system's resources – CPU shares and memory.

## Basic usage of the top command

By default, you run **top** without any parameters, and it shows you a full screen (or full window of your terminal) with the current status of your system and a list of processes using most of its CPU:

```
ubuntu$ top
top - 13:29:09 up 2 days, 7:13, 4 users, load average:
0.07, 0.02, 0.00
Tasks: 148 total, 1 running, 147 sleeping, 0 stopped, 0
zombie
Cpu(s): 0.6%us, 0.5%sy, 0.0%ni, 97.3%id, 1.6%wa, 0.0%hi,
0.0%si, 0.0%st
Mem: 4051792k total, 4026104k used, 25688k free,
359168k buffers
Swap: 4096492k total, 24296k used, 4072196k free,
2806484k cached
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+
COMMAND
 7629 greys    20   0 749m 291m 28m  S   1   7.4 16:51.40
firefox
19935 greys  20   0 133m  14m 10m  S   0   0.4  2:38.52
smplayer
   1 root     20   0 4020  880 592  S   0   0.0  0:00.96
init
   2 root     15  -5     0     0     0  S   0   0.0  0:00.00
kthreadd
   3 root     RT  -5     0     0     0  S   0   0.0  0:00.04
migration/0
   4 root     15  -5     0     0     0  S   0   0.0  0:00.90
```



# Output of the top command explained

These are the elements which default **top** output consists of:

## Unix system uptime and average load

This is the line of **top** which confirms how many hours (or even days!) your system has been up, shows you the number of logged in users, and reports the average system load numbers for the last minute, 5 minutes and 15 minutes.

```
top - 13:29:09 up 2 days, 7:13, 4 users, load average:
0.07, 0.02, 0.00
```

In this line:

- *13:29:09* is the current time
- *2 days, 7:13* is the uptime
- *4 users* shows how many users currently use your system
- *0.07* – average load for the last minute
- *0.02* – average load for the last 5 minutes
- *0.00* – average load for the last 15 minutes

## Unix tasks stats

Here you can see how many tasks are currently running on your system. Tasks here mean processes, and the main listing will show you the task names (in the COMMAND column) and the PIDs.

```
Tasks: 148 total, 1 running, 147 sleeping, 0 stopped, 0
zombie
```

## CPU(s) status

Current CPU state, averaged for the number of CPUs installed in your system, is represented in this line:

```
Cpu(s): 0.6%us, 0.5%sy, 0.0%ni, 97.3%id, 1.6%wa, 0.0%hi,
0.0%si, 0.0%st
```

Here are the explanations for each parameter:

- **us** – *User CPU time*. The time the CPU has spent running users' processes with default priorities
- **sy** – *System CPU time*. The time the CPU has spent running the kernel and its processes
- **ni** – *Nice CPU time*. The time the CPU has spent running users' process that have been prioritized up using **nice** command
- **wa** – *I/O wait*. Amount of time the CPU has been waiting for I/O operations to complete
- **hi** – *Hardware IRQ*. The amount of time the CPU has been servicing hardware interrupts
- **si** – *Software Interrupts*. The amount of time the CPU has been servicing software interrupts
- **st** – *Steal Time*. The amount of CPU 'stolen' from this virtual machine by the hypervisor for other tasks (such as running another virtual machine) – a fairly recent addition to the top command, introduced with the increased virtualization focus in modern operating systems

## Physical memory usage stats

Memory stats line gives you a summary of how much physical memory you have on your system, and how much of it is currently used or available for the use.

Modern Linux systems are buffering quite a lot for improved performance, which means you rarely get to see all your physical RAM free – the more your system stays up and running, the more of its recently used data ends up being buffered.

In this line, you can see how quite a bit is taken up by the buffers:

```
Mem:    4051792k total,  4026104k used,    25688k free,
359168k buffers
```

## Swap usage stats

Swap statistics highlight how actively your system uses the swap space – most of it should not be used on a healthy system, although seeing substantial amount of swap memory cached is quite normal. Bear in mind that these are caches held in physical memory, so in this example these 2.8Gb of cached swap is responsible for most of the 4Gb physical RAM taken up and reported as used in the above stats for memory

```
Swap: 4096492k total, 24296k used, 4072196k free,
2806484k cached
```

## List of the tasks (processes) running on your system

This is the main part of the top output, which looks like this (output is abridged):

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
7629	greys	20	0	749m	291m	28m	S	1	7.4	16:51.40
firefox										
19935	greys	20	0	133m	14m	10m	S	0	0.4	2:38.52
smplayer										
1	root	20	0	4020	880	592	S	0	0.0	0:00.96
init										
2	root	15	-5	0	0	0	S	0	0.0	0:00.00
kthreadd										
3	root	RT	-5	0	0	0	S	0	0.0	0:00.04
migration/0										
4	root	15	-5	0	0	0	S	0	0.0	0:00.90
ksoftirqd/0										
5	root	RT	-5	0	0	0	S	0	0.0	0:00.00
watchdog/0										
6	root	RT	-5	0	0	0	S	0	0.0	0:00.06
migration/1										

As you can see from this list, you're given all the vital information about each process running on your Unix system:

- **PID** – process ID

- **USER** – username for the owner of each process
- **PR** – process priority (RT means a Real Time priority class – used for system processes)
- **NI** – priority set by nice utility
- **VIRT** – the amount of virtual memory used by a process: code, data and shared libraries plus pages that have been swapped out
- **RES** – the resident part of a process – how much of it resides in the physical memory (non-swapped memory)
- **SHR** – shows you the size of potentially shared memory segments for a process
- **S** – the current state of each process
- **%CPU** – percentage of the time shares CPU spends running a particular process
- **%MEM** – percentage of the physical memory of your system which is used by each process
- **%TIME+** – total time CPUs spent running each process
- **COMMAND** – a command used to initiate each process.

I'll be sure to revisit and expand this page at some later stage.

## See also:

- [uptime](#) – find out how long the system has been up
- [uname](#) – print Unix system information
- [basic Unix commands](#)
- [Unix Commands](#)
- [Unix System Monitoring Commands](#)