
KNOWLEDGE BASE ARTICLE

Linux "top" command

The "top" command is essential for Linux system administrators, offering real-time monitoring and management of processes and resource usage. It features an interactive interface that provides detailed insights into system performance and allows process termination.

PUBLISHED

6 May 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from EssingtonITS

Linux "top" command

6 May 2026

The `top` command is an invaluable tool for system administrators who need to keep an eye on processes and resource usage in Linux. It's a staple in the sysadmin toolkit, and you'll find it pre-installed on virtually every Linux distribution. What makes `top` so handy is its interactivity; not only can you view and monitor processes, but you can also terminate them if necessary.

Getting Started with Top

To fire up `top`, simply open your terminal and type `top`. This launches the interactive interface, which may seem a bit overwhelming at first glance, but it's quite straightforward once you get the hang of it.

```
top - 15:39:37 up 90 days, 15:26,  2 users,  load average: 0.00, 0.00, 0.00
Tasks:  27 total,   1 running, 26 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  524288 total,   22792 free,  119380 used,  382116 buff/cache
KiB Swap:  131072 total,   43716 free,   87356 used.  322002 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	37168	1192	680	S	0.0	0.2	2:21.51	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd/646
3	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khelper/646
62	root	20	0	38896	1316	1188	S	0.0	0.3	1:08.85	systemd-journal
219	root	20	0	26012	328	200	S	0.0	0.1	0:11.09	cron
226	root	20	0	65464	924	220	S	0.0	0.2	0:13.11	sshd
229	syslog	20	0	184632	1524	464	S	0.0	0.3	0:28.85	rsyslogd
231	root	20	0	47572	504	40	S	0.0	0.1	0:07.80	rpcbind
274	root	20	0	12788	8	4	S	0.0	0.0	0:00.00	agetty
275	root	20	0	12788	8	4	S	0.0	0.0	0:00.00	agetty
293	root	20	0	308984	12800	2248	S	0.0	2.4	27:15.96	fail2ban-server
4452	root	20	0	92996	3124	3120	S	0.0	0.6	0:00.03	sshd
4461	supriyo	20	0	92996	1000	996	S	0.0	0.2	0:00.00	sshd
4462	supriyo	20	0	19472	1604	1600	S	0.0	0.3	0:00.05	bash
4696	root	20	0	92996	4036	3132	S	0.0	0.8	0:00.02	sshd
4705	supriyo	20	0	92996	1952	1008	S	0.0	0.4	0:00.02	sshd
4706	supriyo	20	0	19472	3364	1600	S	0.0	0.6	0:00.05	bash
4718	supriyo	20	0	36608	1784	1324	R	0.0	0.3	0:00.31	top
5830	root	20	0	41532	728	320	S	0.0	0.1	0:01.25	systemd-udevd
13879	www-data	20	0	290032	2632	2612	S	0.0	0.5	0:01.18	php-fpm7.0
14031	cloud-t+	20	0	19788	9736	3276	S	0.0	1.9	10:11.27	cloud-torrent
14089	root	20	0	286060	560	452	S	0.0	0.1	1:11.67	php-fpm7.0
14091	www-data	20	0	289508	2168	2064	S	0.0	0.4	0:02.32	php-fpm7.0

Deciphering the Interface

Before diving into the details, you might want to check the version of `top` you're using by typing `top -v`. Once you're in, the screen is divided into two main sections: the *summary* area at the top and the *task* area below, which lists all the processes. By default, this information refreshes every three seconds, providing a near real-time view of your system's performance.

The Summary Area

The first line of the summary area displays the current time, the system uptime, and the number of users currently logged in. It also shows the load averages for the past 1, 5, and 15 minutes, which can help you understand how heavily your system has been used over those time frames.

The second line provides a snapshot of the number of tasks and their states: running, sleeping, stopped, or zombie. This is crucial for understanding how many processes are actively using resources versus those that are idle.

Next, you'll find the CPU usage breakdown:

- **us:** Time spent on user processes.
- **sy:** Time spent on system (kernel) processes.
- **ni:** Time spent on processes with adjusted priority (nice value).
- **id:** Idle CPU time.
- **wa:** Time waiting for I/O operations to complete.
- **hi:** Time handling hardware interrupts.
- **si:** Time handling software interrupts.
- **st:** Time stolen from this VM by the hypervisor (applicable in virtualised environments).

The fourth line details memory usage, showing total physical memory and how much is free, used, or cached. The fifth line does the same for swap memory, including how much is recoverable from caches.

The Task or Process Area

Below the summary, the task area provides a list of all processes, each with several columns of information:

- **PID:** Unique Process ID.
- **USER:** Username of the process owner.
- **PR:** Priority of the process.
- **NI:** Nice value, which affects scheduling priority.
- **VIRT:** Virtual memory used.
- **RES:** Resident memory used (physical memory).
- **SHR:** Shared memory used.
- **S:** Current status of the process.
- **%CPU:** CPU usage percentage since the last update.
- **%MEM:** Percentage of physical memory used.
- **TIME+:** Total CPU time consumed by the process, shown in hundredths of a second.
- **COMMAND:** The command that started the process.

Process statuses include:

- **D:** Uninterruptible sleep (usually I/O).
- **R:** Running or runnable (on run queue).
- **S:** Sleeping (idle).
- **T:** Stopped by job control signal.
- **Z:** Zombie process, terminated but not reaped by its parent.

When you're ready to exit `top`, just press `q`. Avoid using `ctrl+x`, as this can leave a process in a stopped state.