Commands for the Week

Find the processes running on a Linux machine

ps -aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.1 1.1 167788 11144 ? Ss 12:09 0:01 /sbin/init

root 2 0.0 0.0 0 0 ? S 12:09 0:00 [kthreadd]

root 3 0.0 0.0 0 0 ? I< 12:09 0:00 [rcu\_gp]

root 4 0.0 0.0 0 0 ? I< 12:09 0:00 [rcu\_par\_gp]

root 5 0.0 0.0 0 0 ? I< 12:09 0:00 [slub\_flushwq]

root 6 0.0 0.0 0 0 ? I< 12:09 0:00 [netns]

root 7 0.0 0.0 0 0 ? I 12:09 0:00 [kworker/0:0-events]

root 8 0.0 0.0 0 0 ? I< 12:09 0:00 [kworker/0:0H-events\_highpri]

root 9 0.0 0.0 0 0 ? I 12:09 0:00 [kworker/u2:0-events\_unbound]

root 10 0.0 0.0 0 0 ? I< 12:09 0:00 [mm\_percpu\_wq]

Find the users currently logged in

Command:

w

11:41:16 up 5:13, 2 users, load average: 0.01, 0.02, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

ubuntu pts/0 192.168.64.1 Mon13 22:17m 0.05s 0.05s -bash

ubuntu pts/1 192.168.64.1 11:41 1.00s 0.01s 0.00s w

Find zombie processes on a machine

Command:

ps aux | awk '$8=="Z"'

Find the uptime of the machine

Command:

uptime

11:43:43 up 5:15, 2 users, load average: 0.01, 0.02, 0.00

Find the ram usage

Command:

free -h

total used free shared buff/cache available

Mem: 962Mi 166Mi 123Mi 1.0Mi 672Mi 699Mi

Find the disk usage

Command:

df -h

Filesystem Size Used Avail Use% Mounted on

tmpfs 97M 1.2M 96M 2% /run

/dev/sda1 4.7G 2.1G 2.7G 45% /

tmpfs 482M 0 482M 0% /dev/shm

tmpfs 5.0M 0 5.0M 0% /run/lock

/dev/sda15 98M 6.3M 92M 7% /boot/efi

Find the inode usage

Command:

df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

tmpfs 123150 666 122484 1% /run

/dev/sda1 648320 107663 540657 17% /

tmpfs 123150 1 123149 1% /dev/shm

tmpfs 123150 3 123147 1% /run/lock

/dev/sda15 0 0 0 - /boot/efi

Here are the commands to find those details.

Find the ulimit of a user

Command:

ulimit -a

real-time non-blocking time (microseconds, -R) unlimited

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 3563

max locked memory (kbytes, -l) 123148

max memory size (kbytes, -m) unlimited

open files (-n) 1024

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 8192

cpu time (seconds, -t) unlimited

max user processes (-u) 3563

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

Find the ulimit of a process

cat /proc/1112/limits

Limit Soft Limit Hard Limit Units

Max cpu time unlimited unlimited seconds

Max file size unlimited unlimited bytes

Max data size unlimited unlimited bytes

Max stack size 8388608 unlimited bytes

Max core file size 0 unlimited bytes

Max resident set unlimited unlimited bytes

Max processes 3563 3563 processes

Max open files 1024 1048576 files

Max locked memory 126103552 126103552 bytes

Max address space unlimited unlimited bytes

Max file locks unlimited unlimited locks

Max pending signals 3563 3563 signals

Max msgqueue size 819200 819200 bytes

Max nice priority 0 0

Max realtime priority 0 0

Max realtime timeout unlimited unlimited us

Here are the commands to find those details.

Find the ulimit of a user

Command:

Bash

ulimit -a

Output:

The terminal will show a list of all resource limits for the current user. This includes things like max user processes, open files, and cpu time, along with their specific values.

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 31066

max locked memory (kbytes, -l) 65536

max memory size (kbytes, -m) unlimited

open files (-n) 1024

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 8192

cpu time (seconds, -t) unlimited

max user processes (-u) 31066

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

Find the ulimit of a process

Command:

Replace <PID> with the actual Process ID.

Bash

cat /proc/<PID>/limits

Output:

The terminal will display a table showing the specific resource limits applied to that single process. It lists each Limit, its Soft Limit (the effective value), its Hard Limit (the ceiling), and the Units.

Limit Soft Limit Hard Limit Units

Max cpu time unlimited unlimited seconds

Max file size unlimited unlimited bytes

Max data size unlimited unlimited bytes

Max stack size 8388608 unlimited bytes

Max core file size 0 unlimited bytes

Max resident set unlimited unlimited bytes

Max processes 31066 31066 processes

Max open files 1024 4096 files

Max locked memory 65536 65536 bytes

Max address space unlimited unlimited bytes

Max file locks unlimited unlimited locks

Max pending signals 31066 31066 signals

Max msgqueue size 819200 819200 bytes

Max nice priority 0 0

Max realtime priority 0 0

Max realtime timeout unlimited unlimited us

Find the file descriptors used by a process

Command:

sudo ls -l /proc/1112/fd

total 0

lr-x------ 1 root root 64 Sep 16 12:10 0 -> /dev/null

lrwx------ 1 root root 64 Sep 16 12:25 1 -> /dev/null

lrwx------ 1 root root 64 Sep 16 12:25 2 -> /dev/null

Find the top 5 process by memory usage

Command:

ps aux --sort=-%mem | head -n 6

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 10184 0.1 3.6 1773904 36248 ? Ssl 11:38 0:00 /usr/lib/snapd/snapd

root 415 0.0 2.6 289864 25676 ? SLsl 06:28 0:05 /sbin/multipathd -d -s

root 1752 0.0 1.7 297616 16832 ? Ssl 06:28 0:00 /usr/libexec/packagekitd

root 686 0.0 1.6 32888 16468 ? Ss 06:28 0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers

root 750 0.0 1.5 109916 15544 ? Ssl 06:28 0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal

Top 5 Processes by CPU Usage

ps aux --sort=-%cpu | head -n 6

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.1 1.1 167788 11144 ? Ss 12:09 0:01 /sbin/init

root 612 0.1 3.5 1773648 34536 ? Ssl 12:09 0:01 /usr/lib/snapd/snapd

root 2 0.0 0.0 0 0 ? S 12:09 0:00 [kthreadd]

root 3 0.0 0.0 0 0 ? I< 12:09 0:00 [rcu\_gp]

root 4 0.0 0.0 0 0 ? I< 12:09 0:00 [rcu\_par\_gp]

Top 5 Processes by Network Usage

sudo nethogs

PID USER PROGRAM DEV SENT RECEIVED

1158 ubuntu sshd: ubuntu@pts/0enp0s12.0252.032 KB/sec

root unknown TCP0.0000.000 KB/sec

Here are the commands to find the top processes by resource usage.

Top 5 Processes by CPU Usage

To find the top 5 processes currently using the most CPU, you can use the ps command.

Command:

Bash

ps aux --sort=-%cpu | head -n 6

What It Does:

This command lists all running processes (ps aux), sorts them in descending order by their CPU usage (--sort=-%cpu), and then displays the top 5 entries plus a header row (head -n 6).

Example Output:

The output is a static table. The %CPU column shows the processes with the highest usage at that moment.

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

jdoe 4567 25.1 4.2 1895420 345120 ? Sl 12:20 12:34 /usr/bin/ffmpeg -i video.mp4 out.avi

root 1122 5.5 2.5 1258892 205100 ? Sl 11:56 3:05 /usr/bin/gnome-shell

jdoe 2345 2.2 8.1 2358892 645100 ? Sl 11:58 2:45 /usr/lib/firefox/firefox

mysql 987 0.9 1.8 1100456 148765 ? Sl 11:55 1:02 /usr/sbin/mysqld

jdoe 8765 0.5 0.8 450123 65432 ? R 12:25 0:05 ps aux --sort=-%cpu

Top 5 Processes by Network Usage 🌐

Standard tools don't easily show real-time, per-process network usage. You'll need to install a specialized tool like nethogs.

Command (after installation):

Bash

sudo nethogs

What It Does:

nethogs provides a live, interactive display that groups bandwidth usage by process, making it easy to see which application is using your network.

Example Output:

The terminal will show a constantly updating list of processes. The Sent and Received columns show the data transfer rate for each process in KB/sec.

NetHogs version 0.8.5

PID USER PROGRAM DEV SENT RECEIVED

2345 jdoe /usr/lib/firefox/firefox eth0 25.125 350.450 KB/sec

5432 jdoe /usr/bin/ssh eth0 2.450 1.987 KB/sec

6789 root /usr/bin/apt eth0 0.000 50.123 KB/sec

TOTAL 27.575 402.560 KB/sec

Top 5 Processes by Disk I/O Usage

sudo iotop

Total DISK READ:0.00 B/s | Total DISK WRITE: 0.00 B/s

Current DISK READ:0.00 B/s | Current DISK WRITE: 0.00 B/s

TID PRIO USER DISK READ DISK WRITE SWAPIN IO> COMMAND

1234 be/4 root 0.00 B/s 95.21 K/s 0.00 % 8.15 % jbd2/sda1-8

find the network traffic and bandwidth usage of the machine

sudo iftop

Given a file as input, find the processes using that file

Command:

lsof /lib/systemd/systemd

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

systemd 796 ubuntu txt REG 8,1 1785544 3360 /usr/lib/systemd/systemd

List processes listening on a specific port (ex: 22)

Command:

lsof -i :22

List open ports

Command:

ss -tulpn

Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:\*

udp UNCONN 0 0 192.168.64.14%enp0s1:68 0.0.0.0:\*

tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:\*

tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:\*

tcp LISTEN 0 128 [::]:22 [::]:\*

Find the status of a service (ex: httpd)

Command:

systemctl status apache2

apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Active: active (running) since Tue 2025-09-16 12:11:32 IST; 1min 5s ago

Docs: https://httpd.apache.org/docs/2.4/

Main PID: 2938 (apache2)

Tasks: 55 (limit: 1069)

Memory: 4.8M

CPU: 20ms

CGroup: /system.slice/apache2.service

├─2938 /usr/sbin/apache2 -k start

├─2940 /usr/sbin/apache2 -k start

└─2941 /usr/sbin/apache2 -k start

Find the permissions set for a file

Command:

ls -l /etc/passwd

-rw-r--r-- 1 root root 1766 Sep 15 13:19 /etc/passwd