



# CPU

System parameters monitoring in Linux

# How to Find Out CPU Load

## Linux Terminal

```
[root@localhost ~]# uptime  
09:15:21 up 24 min,  1 user,  load average: 0.00, 0.01, 0.02
```

What does the **uptime** show?

- Current time
- How long does the system work without a reboot
- Logged users
- Load average – represent the average system load over a period of time 1, 5, and 15 minutes respectively

The values are indicators of the **CPU queue length**.

## Load Average and Multi-Core CPUs

On a single-core CPU effective load average is 1,00. That meaning all queue of processes do not have awaiting process and have full utilization of CPU.

On a ten-core CPU effective load average will be 10,00. The value 5,00 on a ten-core CPU will indicate that it is only half-loaded.

## How Many Cores CPU Has

### Linux Terminal

```
[root@localhost ~]# grep -c processor /proc/cpuinfo
4
[root@localhost ~]# nproc
4
```

Two ways to determine the number of CPU cores

# Why Load Average

---

**Load average** values due to storage i/o and network workload, not just CPU demand.

Despite **Load Average** is an abstract value, it lets one quickly detect bottlenecks and assess overall system load.

# Additional Information

---

## **Load (computing) – Wikipedia**

[https://en.wikipedia.org/wiki/Load\\_\(computing\)](https://en.wikipedia.org/wiki/Load_(computing))

## **Understanding the Load Average on Linux and Other Unix-like Systems**

<https://www.howtogeek.com/194642/understanding-the-load-average-on-linux-and-other-unix-like-systems/>



# Memory

System parameters monitoring in Linux

# How to Find Out Memory Usage

## Linux Terminal

```
[root@localhost ~]# free -h
```

	total	used	free	shared	buff/cache	available
Mem:	990M	162M	725M	6.7M	102M	703M
Swap:	819M	0B	819M			

What does the **free** show?

- Total, used, free **Mem** statues
- Total, used, free **Swap** usage

-h show all output fields automatically scaled to shortest three digit unit and display the units of print out.

# What Free Command Showing

<b>total</b>	Total installed memory
<b>used</b>	Used memory (calculated as <b>used = total - free - buffers - cache</b> )
<b>free</b>	Unused memory
<b>shared</b>	Memory used (mostly) by tmpfs
<b>buff/cache</b>	The combined memory used by the kernel buffers and page cache and slabs This memory can be reclaimed at any time if needed by the applications
<b>available</b>	Estimation of how much memory is available for starting new applications, without swapping.



# What are we Monitoring

---

## **Memory and Swap usage**

- How much free memory is available for operating system and applications
- How much swap usage by operating system and applications

# Additional Information

---

## **Linux Find Out What Process Are Using Swap Space**

<https://www.cyberciti.biz/faq/linux-which-process-is-using-swap/>

## **Exploring virtual memory with vmstat**

<https://www.redhat.com/sysadmin/linux-commands-vmstat>



# Disks

System parameters monitoring in Linux

# How to Find Out Disks Usage

## Linux Terminal

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        484M   0    484M   0% /dev
tmpfs           496M   0    496M   0% /dev/shm
tmpfs           496M  6.8M   489M   2% /run
tmpfs           496M   0    496M   0% /sys/fs/cgroup
/dev/mapper/centos-root 6.2G  1.4G   4.9G  23% /
/dev/sda1       1014M  138M   877M  14% /boot
tmpfs           100M   0    100M   0% /run/user/0
```

What does the **df** show?

- Filesystems
- **Size, Used, Avail, Use%** – disk size, used space, available space and used space in percent
- **Mount on** – where filesystem is mounted

The key **-i** of the **df** command is shown inodes (inodes store the attributes and disk block locations of the object's data) usage status.

# How to Find Who Use a Disk

## Linux Terminal

```
[root@localhost log]# du -h /etc/  
72K      /etc/grub.d  
16K      /etc/pki/rpm-gpg  
160K     /etc/pki/ca-trust/extracted/java  
248K     /etc/pki/ca-trust/extracted/openssl  
376K     /etc/pki/ca-trust/extracted/pem  
788K     /etc/pki/ca-trust/extracted  
0        /etc/pki/ca-trust/source/anchors  
0        /etc/pki/ca-trust/source/blacklist  
4.0K     /etc/pki/ca-trust/source  
800K     /etc/pki/ca-trust
```

What does the **du** show?

- Size and path to file

It is useful to find out the names of directories and files that consume large amounts of space on a disk.

# How to Monitor Disk Activities

## IOSTAT

- **iostat** can be used to report the disk read/write rates and counts for an interval continuously. It collects disk statistics, waits for the given amount of time, collects them again and displays the difference.

## IOTOP

- **iostat** is a top-like utility for displaying real-time disk activity. It can list the processes that are performing I/O, alongwith the disk bandwidth they are using.

## DSTAT

- **dstat** is a little more user-friendly version of iostat and can show much more information than just disk bandwidth.

## ATOP

- **atop** is particularly good for quickly grasping changes happening to the system. It does an excellent job of summarizing changes in each interval.

# What are we Monitoring

---

## **Disk usage and empty space**

- How many empty space is available at each drive on the system
- Which files or directories consume large amount of a disk
- How many inodes is available
- Real-time disk activity

# Additional Information

---

## **Linux Check Disk Space Command To View System Disk Usage**

<https://www.cyberciti.biz/faq/linux-check-disk-space-command/>

## **5 tools for monitoring disk activity in Linux**

<https://www.opsdash.com/blog/disk-monitoring-linux.html>





# Network

System parameters monitoring in Linux

# How to Find Out Network Stats

## Linux Terminal

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe08:9212  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:08:92:12  txqueuelen 1000  (Ethernet)
    RX packets 46  bytes 4602 (4.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 61  bytes 5310 (5.1 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

What does the **ifconfig** show?

- Interface name and status
- **inet** (IPv4) and **inet6** (IPv6) settings
- **MAC** address and queue length
- Interface stats **RX** (Receive) and **TX** (Transmit)

# How to Monitor Network Bandwidth

## Linux Terminal

Linux network traffic monitoring is one of the main parts of Linux troubleshooting. And can be done by many network monitoring utilities.

application	description
iptraf	Monitor tool show many detailed information on the IP traffic (on screenshot)
nload	monitors network traffic and bandwidth usage in real time
iftop	monitor tool that produces a frequently updated list of network connections

```
iptraf-ng 1.2.1
Statistics for enp0s3

      Total      Total      Incoming      Incoming      Outgoing      Outgoing
      Packets    Bytes    Packets    Bytes    Packets    Bytes
Total:    6169    51831835    3539    58887427    2630    144488
IPv4:      0      0      0      0      0      0
IPv6:      0      0      0      0      0      0
TCP:      6188    51816732    3488    58872476    2628    144256
UDP:       61    14391      59    14239      2      152
ICMP:      0      0      0      0      0      0
Other IP:  0      0      0      0      0      0
Non-IP:    0      0      0      0      0      0
Broadcast: 57    14887      57    14887      0      0

Total rates:      4.71 kbps      Broadcast rates:      4.71 kbps
                  1 pps                  1 pps

Incoming rates:   4.71 kbps
                  1 pps

Outgoing rates:   0.00 kbps
                  0 pps

IP checksum errors: 0

Time: 0:00 Drops: 0
X-exit
```

# What are we Monitoring

## **RX (Receive) and TX (Transmit) bandwidth and errors**

<b>bandwidth</b>	maximum amount of data transmitted over an connection in a given amount of time
<b>packets</b>	total number of packets received or transmitted
<b>bytes</b>	total number of bytes received or transmitted over interface
<b>errors</b>	total number of packets with error
<b>overruns</b>	number of received packets that experienced fifo overruns, caused by rate at which a buffer gets full and kernel is not able to empty it

# Benefits of network monitoring

## Clear visibility into the network

- Through network monitoring, administrators can get a clear picture of all the connected devices in the network, see how data is moving among them, and quickly identify and correct issues that can undermine performance and lead to outages.

## Better use of IT resources

- The hardware and software tools in network monitoring systems reduce manual work for IT teams. That means valuable IT staff have more time to devote to critical projects for the organization.

## Early insight into future infrastructure needs

- Network monitoring systems can provide reports on how network components have performed over a defined period. By analyzing these reports, network administrators can anticipate when the organization may need to consider upgrading or implementing new IT infrastructure.

## The ability to identify security threats faster

- Network monitoring helps organizations understand what "normal" performance looks like for their networks. So, when unusual activity occurs, such as an unexplained increase in network traffic levels, it's easier for administrators to identify the issue quickly--and to determine whether it may be a security threat.

# Additional Information

---

## **Monitoring Network Usage in Linux**

<https://www.baeldung.com/linux/monitor-network-usage>

## **How to monitor network activity on a Linux system**

<https://linuxconfig.org/how-to-monitor-network-activity-on-a-linux-system>



# Processes

System parameters monitoring in Linux

# How to Find Process List

## Linux Terminal

```
top - 09:57:01 up 32 min,  2 users,  load average: 0.00, 0.01, 0.05
Tasks:  94 total,   4 running,  90 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 : 1014756 total,  631104 free,  158600 used,  225052 buff/cache
KiB Swap:  839676 total,  839676 free,      0 used.  709656 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
399	root	20	0	0	0	0	R	0.3	0.0	0:00.14	xfssaild/dm-0
1387	root	20	0	161968	2180	1540	R	0.3	0.2	0:00.06	top
1	root	20	0	128012	6620	4144	S	0.0	0.7	0:01.59	systemd

Main points for monitor process of the system by using **top** command

- **VIRT, RES, SHR** – virtual, real RAM, shared memory usage
- **S** – current state of the process
- **%CPU, %MEM** – used CPU and MEMORY in percentage
- **TIME+** – CPU time by command
- **COMMAND** – application name which uses resources of the system



# How to Find Which Process Use Memory or CPU

---

The **top** command can show sorted list of processes by memory or CPU usage:

- **top -o %MEM** – sort process list by usage memory in percent
- **top -o %CPU** – sort process list by CPU usage, it can be more than 100% if process use more than one CPU core

Use **Shift + >** or **<** to switch the column which sorted in rotation

# Who to Find Process Status

## Linux Terminal

```
[root@localhost ~]# ps aux | less
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.2   0.6 128012  6620 ?        Ss   01:26   0:00 /usr/lib/systemd/systemd --switched
-root --system --deserialize 22
root         2   0.0   0.0      0     0 ?        S    01:26   0:00 [kthreadd]
root         4   0.0   0.0      0     0 ?        S<   01:26   0:00 [kworker/0:0H]
root         5   0.0   0.0      0     0 ?        S    01:26   0:00 [kworker/u2:0]
root         6   0.0   0.0      0     0 ?        S    01:26   0:00 [ksoftirqd/0]
root         7   0.0   0.0      0     0 ?        S    01:26   0:00 [migration/0]
root         8   0.0   0.0      0     0 ?        S    01:26   0:00 [rcu_bh]
```

The **ps** command can show displays the currently-running processes. Top useful **ps** options:

- **ps aux | less** – show full list of processes thought less command
- **ps aux | grep <PID|program name|user>** – show processes filtered by PID or application name or username

# Why we Need Monitor Processes

## CAUSES OF HIGH CPU USAGE

- if the process uses a high CPU this may be showing many problems like working with many tasks, bottleneck with memory, disks or network issues. Investigation of high usage of CPU need to know the type of application and minimal system requirements. Sometimes of high CPU usage may depend on high usage of memory and swap.

## CAUSES OF HIGH MEMORY USAGE

- If an application uses a big amount of memory and uses a swap, it indicates an improperly configured application or low physical memory at the host machine. For example, a database can work with a big amount of data and improper configuration can fill all available memory and swap.

# Additional Information

---

## **top command in Linux with Examples**

<https://www.geeksforgeeks.org/top-command-in-linux-with-examples/>

## **30 useful commands of «ps» for process monitoring in Linux (Russian)**

<https://blog.sedicomm.com/2018/05/28/30-poleznyh-komand-ps-dlya-monitoringa-protssessov-linux/>

**THANK YOU**