

Memory Management

LinuXpert Systems

No.5, Dr. Ambedkar St, Rathinapuram,

Nanganallur, Chennai – 600 061.

Phone: 98841 65649, 90808 01382

E-mail: baskar@linuxpert.in

TABLE OF CONTENTS

1. free 1

2. vmstat 7

3. slabtop 15

4. swapon 22

5. swapoff 27

1. free

The **`free`** command in Linux is used to **display information about system memory (RAM) and swap space**. It shows how much memory is currently being used, how much is free, and how much is available for applications.

Syntax

```
free [OPTIONS]
```

Basic Example

```
free
```

Example Output:

```
               total        used        free      shared  buff/cache   available
Mem:         8077360     2438200     1526348      184320     4112812     5198740
Swap:        2097148           0     2097148
```

Explanation of Columns

Column	Description
total	Total installed physical memory (RAM).
used	Memory currently in use by programs.
free	Completely unused memory.
shared	Memory shared between processes (mainly tmpfs).
buff/cache	Memory used for buffers and filesystem cache. Linux uses free RAM as cache to improve performance.
available	Estimated memory available for starting new applications without swapping.
Swap	Disk space used as virtual memory when RAM is full.

Common Options

1. Display memory in human-readable format

```
free -h
```

Example:

```
               total        used        free      shared  buff/cache   available
Mem:           7.7Gi       2.3Gi       1.5Gi       180Mi       3.9Gi       5.0Gi
Swap:          2.0Gi          0Bi       2.0Gi
```

`-h` displays sizes using KB, MB, GB, etc.

---

2. Display memory in megabytes

```
free -m
```

Example:

```
               total   used   free   shared  buff/cache  available
Mem:            7888   2380   1490      180        4018       5070
Swap:           2048      0   2048
```

---

3. Display memory in gigabytes

```
free -g
```

Example:

```
               total  used  free
Mem:               8     2     1
Swap:              2     0     2
```

---

4. Display memory in kilobytes

```
free -k
```

Shows memory in KB (default on many systems).

5. Refresh continuously

```
free -s 2
```

Refreshes every **2 seconds**.

Stop with:

```
Ctrl + C
```

---

6. Display total memory summary

```
free -t
```

Example:

```
               total        used        free
Mem:         8077360     2438200     1526348
Swap:        2097148           0     2097148
Total:      10174508     2438200     3623496
```

---

7. Combine options

```
free -ht
```

Shows:

- Human-readable output
- Total memory line

Understanding "used" vs "available"

Many beginners think that **low "free" memory means the system is running out of RAM**, but that's usually **not true**.

Linux uses unused RAM as **filesystem cache** to improve performance. This cached memory can be released automatically when applications need it.

For example:

```
Mem:
total:       8 GB
used:        6 GB
free:      200 MB
available: 3.5 GB
```

Although only **200 MB** is completely free, there are still **3.5 GB available** because cached memory can be reclaimed.

The `available` column is generally the best indicator of how much memory is actually usable.

Practical Examples

Check current RAM usage

```
free -h
```

Monitor memory every second

```
free -h -s 1
```

View memory in MB

```
free -m
```

Get total RAM including swap

```
free -ht
```

Typical Use Cases

- Check available RAM before starting a memory-intensive application.
- Monitor memory usage while troubleshooting performance issues.
- Verify whether the system is using swap space.
- Observe memory changes during application execution.
- Check memory usage on servers during maintenance.

Summary

Command	Purpose
`free`	Show memory usage
`free -h`	Human-readable output
`free -m`	Show memory in MB
`free -g`	Show memory in GB
`free -k`	Show memory in KB
`free -t`	Include total memory summary
`free -s 2`	Refresh every 2 seconds
`free -ht`	Human-readable output with total

Key Points

- `free` displays **RAM and swap usage**.
- Use **`free -h`** for the most readable output.
- Focus on the **`available`** column rather than just **`free`** when assessing memory pressure.
- Linux intentionally uses spare RAM for caching to improve performance, so a low **`free`** value is often normal.

2. vmstat

The **`vmstat`** command is a Linux utility that displays information about **system performance**, including:

- Virtual memory usage
- CPU usage
- Processes
- Paging
- Block I/O
- Disk activity
- System interrupts
- Context switches

It is commonly used by **Linux system administrators** to monitor system performance and troubleshoot resource bottlenecks.

Syntax

```
vmstat [options] [delay [count]]
```

- **delay** – Time interval (in seconds) between updates.
- **count** – Number of reports to display.

Basic Examples

1. Display current system statistics

```
vmstat
```

Example output:

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0    0 820000  45000 500000    0    0     2     3  120  240  4  2 94  0  0
```

2. Refresh every 2 seconds

```
vmstat 2
```

Updates continuously every 2 seconds.

---

3. Display five reports

```
vmstat 2 5
```

Shows:

- First report (since boot)
- Four more reports every 2 seconds

---

4. Display memory in MB

```
vmstat -S M
```

Possible units:

```
vmstat -S K
vmstat -S M
vmstat -S G
```

---

5. Show active and inactive memory

```
vmstat -a
```

Adds:

- Active memory
- Inactive memory

6. Show disk statistics

```
vmstat -d
```

Displays read/write statistics for disks.

---

7. Show partition statistics

```
vmstat -p sda1
```

Displays statistics for partition `sda1`.

---

8. Show slab information

```
vmstat -m
```

Displays kernel slab allocator statistics.

---

9. Display event counters

```
vmstat -s
```

Shows totals since boot:

- Total memory
- Free memory
- Swap
- CPU ticks
- Context switches
- Interrupts

Understanding the Output

Example:

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b swpd free buff cache si so bi bo in cs us sy id wa st
 1  0   0 820000 45000 500000 0 0 2 3 120 240 4 2 94 0 0
```

1. Processes

Column	Meaning
r	Number of runnable processes waiting for CPU
b	Processes blocked (usually waiting for I/O)

2. Memory

Column	Meaning
swpd	Swap memory used
free	Free RAM
buff	Memory used as buffers
cache	Memory used as page cache

3. Swap

Column	Meaning
si	Swap in (KB/s)
so	Swap out (KB/s)

Healthy system:

```
si = 0
so = 0
```

High values indicate memory pressure.

4. I/O

Column	Meaning
bi	Blocks received from block devices
bo	Blocks sent to block devices

Large values indicate heavy disk activity.

5. System

Column	Meaning
in	Interrupts per second
cs	Context switches per second

6. CPU

Column	Meaning
Us	Time spent running user processes (%)
sy	Time spent running kernel processes (%)
id	CPU idle time (%)
wa	Time waiting for I/O (%)
st	Time stolen by the hypervisor (virtual machines) (%)

Example Analysis

Output:

```
r b swpd free buff cache si so bi bo in cs us sy id wa st
2 0 0 650000 35000 420000 0 0 4 5 180 300 20 8 70 2 0
```

Interpretation:

- 2 processes are ready to run.
- No blocked processes.
- No swap is being used.
- Plenty of free memory.
- Light disk activity.
- CPU is:
    - 20% user processes
    - 8% kernel processes
    - 70% idle
    - 2% waiting for disk I/O

The system appears healthy with no significant bottlenecks.

Useful Options

Option	Description
`vmstat`	Show current statistics
`vmstat 2`	Refresh every 2 seconds
`vmstat 2 10`	Show 10 reports at 2-second intervals
`vmstat -a`	Show active/inactive memory
`vmstat -d`	Display disk statistics
`vmstat -p sda1`	Show partition statistics
`vmstat -m`	Show slab allocator information
`vmstat -s`	Show summary statistics
`vmstat -S M`	Display values in megabytes
`vmstat -w`	Wide output for better readability

Common Use Cases

Monitor CPU usage

```
vmstat 1
```

Watch the `us`, `sy`, `id`, and `wa` columns to understand CPU load.

Detect memory pressure

```
vmstat 1
```

Look for:

- Increasing `swpd`
- Non-zero `si` or `so`

These may indicate insufficient RAM.

Identify disk bottlenecks

```
vmstat 1
```

High values in:

- `wa` (CPU waiting for I/O)
- `bi`
- `bo`

can point to heavy disk activity or slow storage.

Monitor virtual machines

```
vmstat 1
```

Pay attention to:

- `st` (steal time)

If `st` is consistently high, the virtual machine may not be receiving enough CPU time from the host.

Advantages

- Lightweight and available on most Linux distributions.
- Provides a quick overview of CPU, memory, swap, and I/O activity.
- Useful for real-time performance monitoring.
- Helps identify CPU, memory, or disk bottlenecks.

Related Commands

- `top` / `htop` – Interactive process and system monitoring.
- `free` – Memory usage.
- `iostat` – Disk I/O statistics.
- `sar` – Historical system performance data.
- `mpstat` – CPU statistics per processor.
- `pidstat` – Per-process performance statistics.
- `dstat` – Combined system resource statistics.

`vmstat` is one of the first commands Linux administrators use when diagnosing performance issues because it provides a concise view of the system's overall health.

3. slabtop

The **`slabtop`** command is a **real-time monitoring tool** used to display **kernel slab cache information**. It is similar to the `top` command, but instead of showing CPU and memory usage by processes, it shows **memory used by the Linux kernel's slab allocator**.

It helps system administrators understand how kernel memory is being used and detect kernel memory leaks.

Syntax

```
slabtop [options]
```

What is Slab Memory?

Linux uses a **slab allocator** to efficiently manage memory for frequently used kernel objects.

Instead of repeatedly allocating and freeing memory, the kernel keeps caches (called **slabs**) containing preallocated objects.

Examples of objects stored in slabs include:

- Process descriptors
- File descriptors
- Inodes
- Dentries (directory cache)
- Network buffers
- Task structures

Example

```
slabtop
```

Example output:

```
Active / Total Objects (% used)    : 985432 / 1023456 (96.3%)
Active / Total Slabs (% used)      : 25412 / 26001 (97.7%)
Active / Total Caches (% used)     : 142 / 170 (83.5%)
Active / Total Size (% used)       : 235.6M / 248.4M (94.9%)
Minimum / Average / Maximum Object : 0.02K / 0.24K / 8.00K

  OBJS ACTIVE  USE OBJ SIZE  SLABS OBJ/SLAB CACHE SIZE NAME
250000 245000 98%   0.19K   12500       20      50M dentry
150000 149500 99%   0.06K    2500       60      10M kmalloc-64
120000 118000 98%   0.50K    3000       40      60M inode_cache
```

The display refreshes every few seconds until you quit.

Press **q** to exit.

Understanding the Columns

Column	Description
OBJS	Total number of objects in the cache
ACTIVE	Objects currently in use
USE	Percentage of objects in use
OBJ SIZE	Size of each object
SLABS	Number of slab pages allocated
OBJ/SLAB	Number of objects stored in one slab
CACHE SIZE	Total memory used by that cache
NAME	Name of the slab cache

Header Information

Example:

```
Active / Total Objects
```

Shows:

- Total allocated kernel objects
- Objects currently being used

```
Active / Total Slabs
```

Shows:

- Total slab pages
- Active slab pages

```
Active / Total Size
```

Shows the total amount of kernel memory occupied by slab caches.

Example:

```
210M / 225M
```

means:

- 210 MB actively used
- 225 MB allocated

Common Slab Cache Names

Cache	Purpose
dentry	Directory entry cache
inode_cache	Filesystem inode cache
kmalloc-32	32-byte kernel allocations
kmalloc-64	64-byte kernel allocations
kmalloc-128	128-byte kernel allocations
kmalloc-256	256-byte kernel allocations
kmalloc-512	512-byte kernel allocations
kmalloc-1024	1 KB kernel allocations
task_struct	Process structures
cred_jar	Process credentials
sock_inode_cache	Socket-related inodes
buffer_head	Filesystem buffer metadata

Interactive Commands

While `slabtop` is running:

Key	Action
q	Quit
s	Change refresh delay
d	Sort by active objects
o	Sort by object count
u	Sort by object usage (%)
c	Sort by cache size
a	Sort by active slabs

Useful Options

Display once and exit

```
slabtop -o
```

Useful for scripting or capturing a snapshot.

Help

```
slabtop --help
```

Displays available options.

Version

```
slabtop --version
```

Shows the installed version.

Practical Examples

1. Monitor kernel slab usage

```
slabtop
```

Watch slab memory usage in real time.

---

2. Save a snapshot

```
slabtop -o > slab_report.txt
```

Stores the current slab statistics in a file.

3. Find the largest slab cache

Run:

```
slabtop
```

Then press:

```
c
```

to sort by **cache size** and identify which slab cache is consuming the most kernel memory.

---

4. Investigate a memory leak

If the output shows something like:

```
dentry      800M
inode_cache 500M
```

and these values keep growing over time without dropping, it may indicate excessive filesystem caching or a potential kernel memory leak.

When to Use `slabtop`

- Investigating high kernel memory usage.
- Diagnosing suspected kernel memory leaks.
- Monitoring filesystem caches.
- Analysing memory usage by kernel objects.
- Performance tuning on busy Linux servers.
- Troubleshooting storage, networking, or filesystem-related memory consumption.

Difference Between `top`, `free`, and `slabtop`

Command	Shows
`top`	CPU, memory, and processes
`free`	Overall RAM and swap usage
`vmstat`	Virtual memory and system statistics
`slabtop`	Kernel slab cache memory usage

Summary

- **Purpose:** Displays real-time kernel slab cache usage.
- **Used by:** System administrators and kernel developers.
- **Interactive:** Yes, similar to `top`.
- **Primary use:** Monitor kernel object caches and diagnose kernel memory usage or slab-related memory leaks.
- **Requires:** Usually root privileges for complete visibility, although many systems allow non-root users to view basic slab statistics.

4. swapon

The `swapon` command is used to **enable swap space** on a Linux system. Swap space is disk space that the kernel uses as **virtual memory** when physical RAM becomes full.
Once enabled, the operating system can move inactive memory pages from RAM to the swap area, freeing RAM for active processes.

Syntax

```
swapon [OPTIONS] <device|file>
```

Common Options

Option	Description
`-a`, `--all`	Enable all swap areas listed in `/etc/fstab`.
`-s`	Display swap summary (deprecated; use `--show`).
`--show`	Show information about active swap devices/files.
`--noheadings`	Suppress the table header.
`--bytes`	Display sizes in bytes.
`--priority <N>`	Set swap priority. Higher priority swap is used first.
`--discard`	Enable discard (TRIM) support for SSD-backed swap.
`--verbose`	Display detailed information.
`--help`	Display help information.
`--version`	Show version information.

Basic Examples

1. Enable a swap partition

```
sudo swapon /dev/sdb2
```

Enables the swap partition located on `/dev/sdb2`.

---

2. Enable a swap file

```
sudo swapon /swapfile
```

Enables a swap file named `/swapfile`.

---

3. Enable all swap devices from `/etc/fstab`

```
sudo swapon -a
```

This is commonly executed during system startup.

Example `/etc/fstab` entry:

```
/swapfile none swap sw 0 0
```

---

4. Display active swap areas

```
swapon --show
```

Example output:

```
NAME       TYPE  SIZE   USED PRIO
/swapfile  file    4G   256M   -2
```

---

5. Display swap in bytes

```
swapon --show --bytes
```

Example:

```
NAME       TYPE          SIZE         USED
/swapfile  file 4294967296 268435456
```

6. Display only specific columns

```
swapon --show=NAME,SIZE,USED
```

Example:

```
NAME       SIZE USED
/swapfile   4G 256M
```

---

7. Enable swap with a priority

```
sudo swapon --priority 10 /dev/sdb2
```

The kernel prefers higher-priority swap areas.

View Current Swap Usage

Using `free`

```
free -h
```

Example:

```
               total        used        free
Mem:            16Gi       5.2Gi       9.8Gi
Swap:            4Gi       256Mi       3.7Gi
```

Using `/proc/swaps`

```
cat /proc/swaps
```

Example:

```
Filename      Type      Size    Used Priority
/swapfile     file    4194300 262144   -2
```

Related Command: `swapoff`

To disable swap:

```
sudo swapoff /swapfile
```

Disable all swap:

```
sudo swapoff -a
```

Typical Workflow for Creating and Enabling a Swap File

```
sudo fallocate -l 4G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
```

Verify:

```
swapon --show
```

Exit Status

- `0` – Success
- Non-zero – An error occurred (for example, insufficient permissions, invalid swap area, or the device/file is already active).

Practical Use Cases

- Add virtual memory when RAM is exhausted.
- Improve system stability by reducing out-of-memory (OOM) situations.
- Configure temporary swap for memory-intensive workloads.
- Enable swap after creating a new swap partition or swap file.
- Activate swap automatically during boot using `/etc/fstab`.

Notes

- `swapon` **enables** swap space; it does **not create** it. A swap area must first be prepared with `mkswap`.
- Root (or `sudo`) privileges are typically required.
- Excessive swapping can significantly reduce performance because disk access is much slower than RAM.
- Modern Linux systems commonly use a **swap file** instead of a dedicated swap partition due to its flexibility.

Summary

Command	Purpose
`swapon /swapfile`	Enable a swap file
`swapon /dev/sda2`	Enable a swap partition
`swapon -a`	Enable all swap areas from `/etc/fstab`
`swapon --show`	List active swap devices/files
`swapon --show --bytes`	Show swap information in bytes
`swapoff -a`	Disable all swap areas
`free -h`	View RAM and swap usage
`cat /proc/swaps`	Display currently active swap areas

5. swapoff

The **`swapoff`** command is used to **disable one or more swap areas** on a Linux system.

Swap space is disk space that the operating system uses as **virtual memory** when the physical RAM is full. When you run `swapoff`, Linux stops using the specified swap partition or swap file.

Syntax

```
swapoff [OPTIONS] <swap_device_or_swap_file>
```

Or disable **all** swap areas:

```
swapoff -a
```

Common Options

Option	Description
`-a`	Disable all swap areas listed in `/etc/fstab`
`-v`	Verbose mode (shows what is being disabled)
`--help`	Display help information
`--version`	Display version information

Examples

1. Disable a swap partition

```
sudo swapoff /dev/sda2
```

Disables the swap partition `/dev/sda2`.

---

2. Disable a swap file

```
sudo swapoff /swapfile
```

Stops using the swap file named `/swapfile`.

3. Disable all swap

```
sudo swapoff -a
```

Disables every active swap area configured on the system.

---

4. Verbose output

```
sudo swapoff -v /swapfile
```

Displays information while disabling the swap file.

Check Active Swap

Before running `swapoff`:

```
swapon --show
```

or

```
cat /proc/swaps
```

Example output:

```
NAME       TYPE  SIZE   USED PRIO
/swapfile  file    2G   512M   -2
```

Verify Swap is Disabled

Run:

```
swapon --show
```

If no output appears, no swap areas are active.

You can also check:

```
free -h
```

Example:

Before:

```
Swap: 2.0Gi  512Mi  1.5Gi
```

After `swapoff`:

```
Swap: 0B 0B 0B
```

Important Notes

- **Requires root privileges**, so use `sudo`.
- If the system doesn't have enough free RAM, `swapoff` may fail because all data currently stored in swap must be moved back into physical memory.
- Disabling swap on a system with limited RAM can lead to memory exhaustion and processes being terminated by the **Out-Of-Memory (OOM) Killer**.

Practical Uses

- Testing application performance without swap.
- Temporarily disabling swap before resizing or recreating a swap partition/file.
- Migrating from one swap device to another.
- System administration and troubleshooting.

Related Commands

Command	Purpose
`swapon`	Enable swap space
`swapon --show`	Display active swap areas
`free -h`	Show RAM and swap usage
`cat /proc/swaps`	List currently active swap areas
`mkswap`	Create a swap area on a partition or file
`vmstat`	Display virtual memory statistics

Example Workflow

```
# Check current swap
swapon --show

# Disable all swap
sudo swapoff -a

# Verify swap is disabled
free -h

# Re-enable all configured swap
sudo swapon -a
```

Summary

- **Command:** `swapoff`
- **Purpose:** Disables swap partitions or swap files.
- **Most common usage:** `sudo swapoff -a`
- **Requires:** Root (`sudo`) privileges.
- **Use with caution:** Ensure sufficient free RAM before disabling swap, or the command may fail or the system may become unstable.