

Table of Contents

01	time	Measure command execution time
02	timeout	Run a command with a time limit
03	vmstat	Report virtual memory statistics

Part 2 of 2 - Explore all 232+ Linux commands at dargslan.com/learn/linux-commands

Each command includes syntax, options, practical examples with output, and pro tips.

\$ time

Beginner

Measure command execution time

time measures the execution time of a command. It reports real (wall clock) time, user CPU time, and system CPU time, providing insight into command performance.

time is available as both a shell built-in and an external command (/usr/bin/time). The external version provides more detail with -v ...

Options & Flags

command Time a command

-v Verbose (external /usr/bin/time)

-o Write timing to file

-f Custom format

Practical Examples

Example: Time a command

```
$ time find / -name "*.log" 2>/dev/null
real    0m2.345s\uuser    0m0.123s\sys    0m0.456s
```

Measures execution time of the find command.

Example: Verbose timing

```
$/usr/bin/time -v sort large_file.txt > /dev/null
Elapsed (wall clock) time: 0:05.23\nMaximum resident set size: 524288 kB
```

Shows detailed timing including memory usage.

Example: Compare approaches

```
$ echo "grep:"; time grep -r "pattern" /var/log/ 2>/dev/null; echo "rg:"; time rg "pattern" /var/log/ 2>/dev/null
```

Compares execution time of two different commands.

Example: Save timing

```
$/usr/bin/time -o benchmark.txt -a ./deploy.sh
```

Appends timing data to a benchmark file.

Example: Script timing

```
$ time (apt update && apt upgrade -y)
```

Times an entire command group.

Tips & Best Practices

Pro Tip: Interpret the three times: real=wall clock, user=CPU in app code, sys=CPU in kernel. real >> user+sys means I/O bound. user >> real means multi-threaded.

Note: Built-in vs /usr/bin/time: Shell built-in time gives basic info. /usr/bin/time -v gives memory usage, page faults, and more detail.

Warning: Include all costs: time includes startup time. For short commands, startup may dominate. Run multiple iterations for accurate benchmarks.

\$ timeout

Intermediate

Run a command with a time limit

timeout runs a command with a time limit. If the command does not complete within the specified duration, it is terminated. timeout is essential for preventing hung processes in scripts and automation.

timeout sends SIGTERM by default when the time limit expires, giving the process a chance to c...

Options & Flags

DURATION	Time limit
-s SIGNAL	Signal to send (default TERM)
-k DURATION	Send KILL after additional time
--preserve-status	Exit with command status, not 124

Practical Examples

Example: 30-second limit

```
$ timeout 30 curl -s https://slow-api.example.com/data
```

Kills curl if it takes more than 30 seconds.

Example: With kill fallback

```
$ timeout -k 10 60 ./long_process.sh
```

Sends SIGTERM at 60s, SIGKILL at 70s if still running.

Example: 5-minute limit

```
$ timeout 5m ./backup.sh
```

Limits backup script to 5 minutes.

Example: Test connectivity

```
$ timeout 5 bash -c "echo > /dev/tcp/server/80" && echo "Open" || echo "Closed/Timeout"
```

Tests port connectivity with a 5-second timeout.

Example: In a loop

```
$ for host in $(cat servers.txt); do timeout 3 ping -c 1 $host > /dev/null && echo "$host: up" || echo "$host: down"; done
```

Pings each server with a 3-second timeout.

Tips & Best Practices

Pro Tip: Exit code 124: timeout returns exit code 124 when it kills the command. Use this in scripts to detect timeouts.

Note: -k for stubborn processes: Some processes ignore SIGTERM. Use -k 5 to send SIGKILL 5 seconds after SIGTERM if the process is still running.

Warning: Duration units: Default is seconds. Append s/m/h/d for seconds/minutes/hours/days: timeout 5m, timeout 2h.

\$ vmstat

Intermediate

Report virtual memory statistics

vmstat (virtual memory statistics) reports system performance statistics including processes, memory, swap, I/O, and CPU activity. It provides a quick overview of system health in a compact format.

vmstat is excellent for real-time monitoring - run it with an interval to see how system resources...

Options & Flags

N Update every N seconds

N M Update N seconds, M times

-w Wide output

-s Display memory statistics

-d Disk statistics

-t Add timestamp

Practical Examples

Example: Quick snapshot

```
$ vmstat
procs ---memory--- ---swap-- -----io----- -system-- -----cpu-----
r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa
1  0      0 2048000 128000 4096000 0  0   5  10 200 400 10  2 87  1
```

Shows a single snapshot of system statistics.

Example: Continuous monitoring

```
$ vmstat 2 10
```

Shows stats every 2 seconds, 10 times.

Example: Wide output

```
$ vmstat -w 1
```

Wider columns for better readability, updating every second.

Example: Memory summary

```
$ vmstat -s
```

Shows detailed memory statistics.

Example: With timestamps

```
$ vmstat -t 5
```

Shows stats every 5 seconds with timestamp.

Tips & Best Practices

Pro Tip: Key columns to watch: r (runnable processes): >CPU count means overloaded. si/so (swap in/out): >0 means swapping. wa (I/O wait): >10% means I/O bottleneck.

Note: First line is averages: The first line of vmstat output shows averages since boot. Subsequent lines show current values.

Warning:

Swap activity is bad: si/so values > 0 mean the system is actively swapping. This indicates insufficient RAM and severe performance impact.

Ready for more? Explore 200+ professional IT eBooks

Go deeper with comprehensive guides, hands-on projects, and real-world examples

dargslan.com/books