
SYSTEM MONITORING & COMPRESSION

Usage, Options, and Sample Outputs

PART 1 — SYSTEM MONITORING COMMANDS

System monitoring commands are used to check disk space, memory usage, CPU load, and system performance. These commands are very important for system administrators and Linux users to maintain system health.

1. df — Disk Filesystem Usage

df = Disk Filesystem

This command shows how much disk space is used and available on all mounted file systems.

Used when:

- Disk is full
- Monitoring server storage
- Checking available space

Basic Usage

df

Options

- df -h → Human readable format (KB, MB, GB)
- df -T → Show filesystem type
- df -h /home → Check specific directory space

Related Commands

- du -sh * → Check folder sizes
- lsblk → Show disks and partitions

Sample Output

```
(kali@kali) - [~/Documents/project]
$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            884788         0     884788   0% /dev
tmpfs           198156        1208     196948   1% /run
/dev/sda1       82083148 16841252 61026348  22% /
tmpfs           990772         4     990768   1% /dev/shm
none            1024          0       1024   0% /run/credentials/systemd-journald.service
tmpfs           990776         8     990768   1% /tmp
none            1024          0       1024   0% /run/credentials/getty@tty1.service
tmpfs           198152        104     198048   1% /run/user/1000

(kali@kali) - [~/Documents/project]
$ df -h
Filesystem      Size      Used Avail Use% Mounted on
udev            865M         0     865M   0% /dev
tmpfs           194M        1.2M    193M   1% /run
/dev/sda1       79G         17G     59G  22% /
tmpfs           968M        4.0K    968M   1% /dev/shm
none            1.0M         0       1.0M   0% /run/credentials/systemd-journald.service
tmpfs           968M        8.0K    968M   1% /tmp
none            1.0M         0       1.0M   0% /run/credentials/getty@tty1.service
tmpfs           194M       104K    194M   1% /run/user/1000

(kali@kali) - [~/Documents/project]
$ df -T
Filesystem      Type      1K-blocks      Used Available Use% Mounted on
udev            devtmpfs  884788         0     884788   0% /dev
tmpfs           tmpfs     198156        1204     196952   1% /run
/dev/sda1       ext4     82083148 16841312 61026288  22% /
tmpfs           tmpfs     990772         4     990768   1% /dev/shm
none            tmpfs     1024          0       1024   0% /run/credentials/systemd-journald.service
tmpfs           tmpfs     990776         8     990768   1% /tmp
none            tmpfs     1024          0       1024   0% /run/credentials/getty@tty1.service
```

2. du — Disk Usage of Files/Folders

du = Disk Usage

Shows how much space files and folders are using.

Difference:

- df → total disk usage
- du → file/folder usage

Basic Usage

du

Options

- du -h → Human readable
- du -sh → Total size of current folder
- du -h * → Size of each folder
- du -sh /home/user → Specific directory size

Combined Commands

du -h | sort -h

Sort folders by size.

```
du -ah | sort -rh | head -10
```

Show top 10 largest files.

Sample Output

```
(kali㉿kali)-[~/Documents]
└─$ du -h
8.0K    ./project
4.0K    ./salary_details
24K     .

(kali㉿kali)-[~/Documents]
└─$ du -sh
24K     .

(kali㉿kali)-[~/Documents]
└─$ du -h *
4.0K    animals.txt
0       copy1.txt
4.0K    numbers.txt
8.0K    project
0       project.txt
4.0K    salary_details
0       test1.txt

(kali㉿kali)-[~/Documents]
└─$ du -sh /hoe/kali
du: cannot access '/hoe/kali': No such file or directory

(kali㉿kali)-[~/Documents]
└─$ du -sh /home/kali
213M    /home/kali

(kali㉿kali)-[~/Documents]
└─$
```

```
(kali㉿kali)-[~/Documents]
└─$ du -h | sort -h
4.0K    ./salary_details
8.0K    ./project
24K     .
```

3. free — Memory (RAM) Usage

Shows RAM usage, free memory, and swap memory.

Used when:

- System slow

- Applications crash
- High RAM usage

Basic Usage

free

Options

- free -h → Human readable
- free -m → Show in MB
- free -g → Show in GB
- free -s 2 → Refresh every 2 seconds

Sample Output

[Insert Screenshot Here – free command output]

4. uptime — System Running Status

Shows:

- How long system is running
- Number of logged-in users
- CPU load average

Usage

uptime

Sample Output

```
(kali㉿kali)-[~/Documents]
$ free
              total        used        free      shared  buff/cache   available
Mem:           1981548      1361624       105476        55024       757336       619924
Swap:           976556        126736       849820

(kali㉿kali)-[~/Documents]
$ free -h
              total        used        free      shared  buff/cache   available
Mem:           1.9Gi       1.3Gi       102Mi        53Mi       739Mi       605Mi
Swap:          953Mi       123Mi       829Mi

(kali㉿kali)-[~/Documents]
$ free -g
              total        used        free      shared  buff/cache   available
Mem:              1          1          0          0          0          0
Swap:              0          0          0

(kali㉿kali)-[~/Documents]
$ free -s 2
              total        used        free      shared  buff/cache   available
Mem:           1981548      1362032       105008        55024       757396       619516
Swap:           976556      126728       849828

              total        used        free      shared  buff/cache   available
Mem:           1981548      1362284       104756        55024       757396       619264
Swap:           976556      126728       849828

              total        used        free      shared  buff/cache   available
Mem:           1981548      1362536       104504        55024       757396       619012
Swap:           976556      126728       849828

              total        used        free      shared  buff/cache   available
Mem:           1981548      1361900       105140        55024       757396       619648
```

5. vmstat — Virtual Memory Statistics

Displays virtual Memory statistics ,system performance including:

- CPU usage
- Memory usage
- Processes
- I/O activity

Usage

vmstat

Live Monitoring

vmstat 2

(refresh every 2 seconds)

Sample Output

```
(kali@kali)-[~/Documents]
$ vmstat
procs-----memory----- --swap-- --io-- --system-- --cpu--
r b swpd free buff cache si so bi bo in cs us sy id wa st gu
1 0 126720 99380 27268 730220 0 1 15 22 322 1 0 1 99 0 0 0

(kali@kali)-[~/Documents]
$ vmstat 2
procs-----memory----- --swap-- --io-- --system-- --cpu--
r b swpd free buff cache si so bi bo in cs us sy id wa st gu
0 0 126672 107520 27468 730240 0 1 15 22 322 1 0 1 99 0 0 0
0 0 126672 107428 27476 730240 0 0 0 28 601 739 0 1 99 0 0 0
0 0 126672 107176 27476 730240 0 0 0 0 335 431 0 0 99 0 0 0
0 0 126672 107176 27476 730240 0 0 0 0 431 517 1 1 97 0 0 0
0 0 126672 107176 27476 730244 0 0 0 0 386 476 1 1 99 0 0 0
0 0 126672 107176 27476 730244 0 0 0 0 305 393 0 1 99 0 0 0
^C
```

Daily Commands Used by System Administrator

These commands are commonly used daily for monitoring:

- df -h → Check disk space
- du -sh /* → Find large folders
- free -h → Check RAM
- uptime → Check system load
- vmstat 1 → Deep monitoring

```
kali@kali: ~/Documents
Session Actions Edit View Help
kali@kali: ~/Documents x kali@kali: ~/Documents x

(kali@kali)-[~/Documents]
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            865M   0 865M   0% /dev
tmpfs           194M  1.2M  193M   1% /run
/dev/sda1       79G   17G   59G  22% /
tmpfs           968M  4.0K  968M   1% /dev/shm
none            1.0M   0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs           968M  8.0K  968M   1% /tmp
none            1.0M   0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs           194M  104K  194M   1% /run/user/1000

(kali@kali)-[~/Documents]
$ vmstat 1
procs-----memory----- --swap-- --io-- --system-- --cpu--
r b swpd free buff cache si so bi bo in cs us sy id wa st gu
2 0 126472 114516 27008 728200 0 1 14 22 322 1 0 1 99 0 0 0
0 0 126472 114264 27008 728200 0 0 0 0 334 422 0 1 99 0 0 0
0 0 126472 114264 27008 728200 0 0 0 0 417 638 1 1 98 0 0 0
0 0 126472 114264 27008 728200 0 0 0 56 474 777 1 1 99 0 0 0
0 0 126472 114264 27008 728200 0 0 0 0 298 388 0 1 99 0 0 0
0 0 126472 114264 27008 728200 0 0 0 472 325 409 0 1 99 0 0 0
0 0 126472 114264 27008 728200 0 0 0 0 283 390 0 1 99 0 0 0
```

PART 2 — COMPRESSION & ARCHIVING

Compression reduces file size.

Archiving combines multiple files into one.

1. tar — Archive Files

tar = Tape Archive

Used to combine multiple files/folders into a single file.

Extension: .tar

Note: tar only archives. It does not compress unless gzip/bzip2 is used.

Usage

```
tar -cvf archive.tar files/
```

Options

- -c → Create archive
- -v → Verbose (show process)
- -f → File name

Extract

```
tar -xvf archive.tar
```

Sample Output

```
(kali㉿kali)-[~/Documents/project]
$ ls
file1.txt  file2.txt  file3.txt

(kali㉿kali)-[~/Documents/project]
$ tar -cvf archivefile.tar .
./
./file3.txt
./file2.txt
tar: ./archivefile.tar: archive cannot contain itself; not dumped
./file1.txt

(kali㉿kali)-[~/Documents/project]
$ ls
archivefile.tar  file1.txt  file2.txt  file3.txt

(kali㉿kali)-[~/Documents/project]
$ tar -xvf archivefile.tar
./
./file3.txt
./file2.txt
./file1.txt

(kali㉿kali)-[~/Documents/project]
$ ls
archivefile.tar  file1.txt  file2.txt  file3.txt

(kali㉿kali)-[~/Documents/project]
$
```

2. gzip — Compress File

Compresses a file and reduces its size.

Extension: .gz

Usage

gzip filename

Decompress

gunzip filename.gz

Sample Output


```
kali@kali: ~/Documents/project x kali@kali: ~/Documents x
(kali@kali)-[~/Documents/project]
$ ls
file1.txt file2.txt file3.txt
(kali@kali)-[~/Documents/project]
$ tar -cvzf backup.tar.gz .
./
./file3.txt
./file2.txt
./file1.txt
tar: .: file changed as we read it
(kali@kali)-[~/Documents/project]
$ ls
backup.tar.gz file1.txt file2.txt file3.txt
(kali@kali)-[~/Documents/project]
$ tar -xvzf backup.tar.gz .
./
./file3.txt
./file2.txt
./file1.txt
(kali@kali)-[~/Documents/project]
$ ls
backup.tar.gz file1.txt file2.txt file3.txt
(kali@kali)-[~/Documents/project]
$
```

3. bzip2 — Better Compression

Provides higher compression than gzip but slower.

Extension: .bz2

Usage

bzip2 filename

Decompress

bunzip2 filename.bz2

Sample Output

```
kali@kali: ~/Documents/project x  kali@kali: ~/Documents x
(kali@kali)-[~/Documents/project]
$ tar -cvjf backup.tar.bz2 .
./
./file3.txt
./file2.txt
./file1.txt
tar: .: file changed as we read it

(kali@kali)-[~/Documents/project]
$ ls
backup.tar.bz2  file1.txt  file2.txt  file3.txt

(kali@kali)-[~/Documents/project]
$
```

4. zip — Create Zip File

Used to compress and archive files together.

Extension: .zip

Usage

zip archive.zip file1 file2

Zip folder

zip -r archive.zip folder/

Sample Output

```
(kali㉿kali)-[~/Documents/project]
└─$ ls
file1.txt  file2.txt  file3.txt

(kali㉿kali)-[~/Documents/project]
└─$ zip file.zip file1.txt
  adding: file1.txt (stored 0%)

(kali㉿kali)-[~/Documents/project]
└─$ ls
file1.txt  file2.txt  file3.txt  file.zip

(kali㉿kali)-[~/Documents/project]
└─$ unzip file.zip
Archive:  file.zip
replace file1.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n

(kali㉿kali)-[~/Documents/project]
└─$ ls
file1.txt  file2.txt  file3.txt  file.zip
```

5. unzip — Extract Zip File

Extract files from a zip archive.

Usage

unzip archive.zip

Extract to specific folder

unzip archive.zip -d /home/user/

Sample Output

```
(kali㉿kali)-[~/Documents/project]
$ ls
file1.txt  file2.txt  file3.txt

(kali㉿kali)-[~/Documents/project]
$ zip file.zip file1.txt
adding: file1.txt (stored 0%)

(kali㉿kali)-[~/Documents/project]
$ ls
file1.txt  file2.txt  file3.txt  file.zip

(kali㉿kali)-[~/Documents/project]
$ unzip file.zip
Archive:  file.zip
replace file1.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n

(kali㉿kali)-[~/Documents/project]
$ ls
file1.txt  file2.txt  file3.txt  file.zip
```

Conclusion

System monitoring commands help in checking disk, memory, CPU, and system performance.

Compression and archiving commands help reduce file size and manage multiple files efficiently.

These commands are essential for:

- System administrators
 - Linux users
 - Server management
 - Performance troubleshooting
-