



REAL TIME SCENARIO BASED LINUX IMPLEMENTATIONS



By DevOps Shack

Real Time Scenario Based Linux Implementations

1. Kill the Process Consuming the Most Memory

Scenario:

Your application server is lagging. You suspect a memory-hogging process.

Command:

```
ps aux --sort=-%mem | head -n 5
```

```
kill -9 <PID>
```

Explanation:

- ps aux: Lists all processes with memory and CPU stats.
- --sort=-%mem: Sorts by memory usage (descending).
- kill -9 <PID>: Forcefully kills the process.

Use Case:

Quick intervention when a Java or Python app leaks memory.

2. Check Listening Ports and Services

Scenario:

Your app fails to start because the port is in use.

Command:

```
ss -tuln
```

Explanation:

Shows TCP/UDP listening ports and prevents port conflicts.

3. Detect Which Process Is Writing Excessively to Disk

Scenario:

EBS disk fills up fast; logs or temp files suspected.

Command:

```
sudo apt update
```

```
sudo apt install iotop -y
```

```
sudo iotop -o
```

Explanation:

Displays real-time I/O activity sorted by process.

4. Find Files Recently Modified (Last 2 Hours)

Scenario:

Something broke after deployment; need to see recent file changes.

Command:

```
find /var/www -type f -mmin -120
```

5. List All Processes of a Specific User

Scenario:

You need to audit or kill all processes owned by a specific user.

Command:

```
ps -u username
```

6. Create a Temporary User With Expiry

Scenario:

A temp contractor or intern needs server access for a week.

Command:

```
sudo useradd -e 2025-06-01 tempuser
```

```
sudo passwd tempuser
```

```
sudo chage -d 0 tempuser
```

7. Find the Process Listening on a Specific Port

Scenario:

Port 8080 is in use and blocking your app.

Command:

```
sudo lsof -i :8080
```

8. Monitor Real-Time CPU Usage by Processes

Scenario:

Server CPU usage is spiking.

Command: top -o %CPU

9. List All Cron Jobs for a Specific User

Scenario:

You suspect a cron job is auto-running at midnight.

Command:

```
crontab -u ubuntu -l
```

10. Backup Only .conf Files With Folder Structure

Scenario:

You need to archive only configuration files.

Command:

```
rsync -av --prune-empty-dirs \  
--include='*/' \  
--include='*.conf' \  
--include='**/*.conf' \  
--exclude='*' \  
/etc/nginx/ backup/nginx/
```

11. Check Logs From Previous Boot

Scenario:

A VM rebooted last night; need logs from that boot.

Command:

```
journalctl -b -1
```

12. Check System Uptime and Load Averages

Scenario:

You're reporting uptime or checking server stability.

Command:

```
uptime
```

13. Find Zombie Processes

Scenario:

System load is low, but defunct processes remain.

Command:

```
ps aux | awk '$8=="Z" { print $0 }'
```

14. Find Directories Using the Most Disk Space

Scenario:

Your disk is 98% full and you need to know where.

Command:

```
du -h --max-depth=1 / | sort -hr
```

15. Display Top 10 Largest Files in a Directory Tree

Scenario:

You want to find large files in /var for cleanup.

Command:

```
find /var -type f -exec du -h {} + | sort -rh | head -n 10
```

16. Check Inode Usage

Scenario:

You're out of inodes even though disk space remains.

Command:

```
df -i
```

17. Monitor Real-Time System Logs

Scenario:

You want to tail live logs from a systemd service.

Command:

```
journalctl -u nginx.service -
```

18. Monitor Real-Time Process Tree

Scenario:

You want to see parent-child process relationships.

Command:

```
pstree -p
```

19. Check Who Is Logged In

Scenario:

You want to verify who is active during a maintenance window.

Command:

```
w
```

20. Kill All Processes Belonging to a User

Scenario:

A test user left rogue processes running.

Command:

```
pkill -u tempuser
```

21. Watch Logs for 'error' Messages in Real Time

Scenario:

You're live-debugging a production issue.

Command:

```
tail -f /var/log/syslog | grep -i --line-buffered "error"
```

22. Check DNS Resolution

Scenario:

Your app can't reach an external API.

Command:

```
dig api.example.com +short
```

23. Limit Bandwidth While Downloading with wget

Scenario:

You're on a shared server and need to download a large file without hogging bandwidth.

Command:

```
wget --limit-rate=1m https://github.com/adoptium/temurin17-binaries/releases/download/jdk-17.0.11%2B9/OpenJDK17U-jdk_x64_linux_hotspot_17.0.11_9.tar.gz
```

24. Reboot a System After 60 Seconds With Warning

Scenario:

You want to notify users and reboot the system gracefully.

Command:

```
sudo shutdown -r +1 "Rebooting for maintenance. Save your work."
```

25. Watch Memory and CPU Usage of a Process

Scenario:

You suspect a specific PID is leaking memory.

Command:

```
watch -n 2 "ps -p <PID> -o %cpu,%mem,cmd"
```

26. Schedule a One-Time Task With at

Scenario:

You want to execute a command 15 minutes from now.

Command:

```
echo "reboot" | at now + 15 minutes
```

27. Run a Script Only if the Previous Command Succeeded

Scenario:

You want to build only if git pull is successful.

Command:

```
git pull && ./build.sh
```

28. Retry Until a Command Succeeds

Scenario:

You're waiting for a service to come online.

Command:

```
until curl -s http://localhost:8080/health; do sleep 5; done
```

29. Create a Compressed Tarball of a Directory

Scenario:

You want to archive and compress an entire folder (e.g., before deployment or backup).

Command:

```
tar -czvf backup.tar.gz /path/to/folder
```

Explanation:

- tar: Archive tool.
 - -c: Create a new archive.
 - -z: Compress using gzip.
 - -v: Verbose — shows files being added.
 - -f: Specifies the filename for the archive.
 - backup.tar.gz: The output file.
 - /path/to/folder: The directory you're archiving.
-

30. Extract a Tar.gz Archive

Scenario:

You need to extract a .tar.gz archive you downloaded or backed up.

Command:

```
tar -xzvf backup.tar.gz
```

Explanation:

- -x: Extract.
 - -z: Use gzip.
 - -v: Verbose output.
 - -f: Specifies the file to extract.
-

31. Show Disk Usage of All Mounted Partitions

Scenario:

Disk space is running out, and you want an overview of each partition.

Command:

```
df -h
```

Explanation:

- df: Disk filesystem usage.
 - -h: Human-readable sizes (e.g., GB/MB).
-

32. Show Mounted Disks and Filesystems

Scenario:

You want to list all block devices, filesystems, and mount points.

Command:

```
lsblk -f
```

Explanation:

- lsblk: Lists block devices (disks, partitions).
 - -f: Shows filesystem info like mount point and type.
-

33. Show Memory Usage Summary

Scenario:

You want to quickly check how much RAM is used/free.

Command:

```
free -h
```

Explanation:

- free: Displays memory usage.
- -h: Human-readable output.

34. List All Installed Packages (Debian-based)

Scenario:

You want to audit installed software on a system.

Command:

```
dpkg -l
```

35. Update and Upgrade All Packages

Scenario:

You want to ensure all packages are up to date.

Command:

```
sudo apt update && sudo apt upgrade -y
```

36. Check Last 100 Lines of Nginx Log

Scenario:

You want to view recent web server activity.

Command:

```
tail -n 100 /var/log/nginx/access.log
```

37. Search Files for a Keyword Recursively

Scenario:

You need to find all places where "TODO" comments exist in a project.

Command:

```
grep -r "TODO" .
```

38. Securely Copy a File to a Remote Server

Scenario:

You need to move a local file to a remote host.

Command:

```
scp file.txt user@host:/remote/path/
```

39. Find All .log Files Larger Than 100MB

Scenario:

You're auditing large log files that may be filling disk space.

Command:

```
find /var/log -name "*.log" -size +100M
```

40. Show All Open Network Connections

Scenario:

You want to audit running services and their ports.

Command:

```
netstat -tulnp
```

41. Start a Service With systemd

Scenario:

You want to manually start a service (e.g., nginx).

Command:

```
sudo systemctl start nginx
```

42. Enable a Service to Start on Boot

Scenario:

You want a service to auto-start after reboot.

Command:

```
sudo systemctl enable nginx
```

43. Check the Status of a Service

Scenario:

You want to see if a service is running and healthy.

Command:

```
sudo systemctl status nginx
```

44. Monitor Real-Time Network Usage

Scenario:

You're diagnosing live network throughput per interface.

Command:

```
sudo apt install iftop -y
```

```
sudo iftop
```

45. Show Environment Variables

Scenario:

You need to check environment variables for debugging.

Command:

```
printenv
```

46. Set an Environment Variable Temporarily

Scenario:

You want to set a variable for the current session only.

Command:

```
export MY_VAR=value
```

47. List All Users on the System

Scenario:

You want to audit which users exist.

Command:

```
cut -d: -f1 /etc/passwd
```

48. Find Out Which User Owns a File

Scenario:

You want to check file ownership for access control.

Command:

```
ls -l filename
```

49. Change Ownership of a File

Scenario:

You're assigning a file to a different user or group.

Command:

```
sudo chown user:group file
```

50. Change Permissions of a Script to Executable

Scenario:

You've created a shell script and want to make it runnable.

Command:

```
chmod +x script.sh
```