

Unix Scenarios with Solutions

1. Delete All `.log` Files Older Than 7 Days

Scenario: Clean up old logs automatically.

```
find /var/logs -name "*.log" -type f -mtime +7  
-exec rm {} \;
```

2. Count Number of Lines in All `.txt` Files in a Directory

Scenario: Need to know how many lines of text are processed.

```
find . -name "*.txt" -exec wc -l {} + | awk  
'{total += $1} END {print total}'
```

3. Monitor Disk Space and Email Alert If Usage > 90%

Scenario: Prevent server crash due to full disk.

```
df -h | awk '$5+0 > 90 {print $0}' | mail -s  
"Disk Usage Alert!" admin@example.com
```

4. Find the Top 5 Largest Files in **/home**

Scenario: Investigate space consumption.

bash

CopyEdit

```
find /home -type f -exec du -h {} + | sort -rh |  
head -n 5
```

5. Remove Blank Lines From a File

Scenario: Clean up a CSV or config file.

```
sed -i '/^$/d' inputfile.txt
```

6. Schedule a Shell Script to Run Daily at 2 AM

Scenario: Automate backups, jobs.

```
# Edit crontab  
crontab -e  
  
# Add entry  
0 2 * * * /path/to/script.sh
```

7. Extract Columns from a CSV

Scenario: Extract just name and age from a user file.

```
bash
CopyEdit
cut -d', ' -f1,2 users.csv
```

8. Replace a Word in All `.conf` Files Recursively

Scenario: Mass update config values.

```
find . -name "*.conf" -exec sed -i
's/oldword/newword/g' {} +
```

9. Track a Live Log File for Error Entries

Scenario: Check for real-time failures in an app.

```
tail -f app.log | grep --line-buffered "ERROR"
```

10. Kill a Process Consuming Too Much CPU

Scenario: Prevent system slowdown.

```
ps aux --sort=-%cpu | head -n 5  
kill -9 <PID>
```

11. Create a Password-Protected Tar File

Scenario: Secure log files before sending.

```
tar -czf - logs/ | openssl enc -e -aes256 -out  
logs.tar.gz.enc
```

12. Compare Two Files and Show Differences

Scenario: Validate changes between config versions.

```
diff file1.conf file2.conf
```

13. Check If a Service Is Running and Restart If Not

Scenario: Ensure web server uptime.

```
if ! pgrep apache2 > /dev/null; then  
    systemctl restart apache2  
fi
```

14. Archive Files Modified in the Last 1 Day

Scenario: Daily archive of new files.

```
find /data -type f -mtime -1 -exec tar -rvf  
daily_backup.tar {} +
```

15. Create Users in Bulk from a File

Scenario: Onboard employees quickly.

```
while IFS= read -r user; do  
    useradd "$user"  
done < users.txt
```