Essential Medium-Level Linux Commands for Everyday Use

find
Used to search for files and directories based on name, type, size, or modification time.

- Find all .log files inside /var/logs/:

```
find /var/logs/ -type f -name "*.log"
```

- Find files modified in the last 7 days:

```
find /home/user/ -type f -mtime -7
```

---

grep
Used to search for specific text patterns inside files.

- Search for "error" in logfile.txt:

```
grep "error" logfile.txt
```

- Search for case-insensitive "warning" in all .log files:

```
grep -i "warning" *.log
```

---

sed

Stream editor for modifying text in files.

- Replace "old" with "new" in file.txt:

```
sed 's/old/new/g' file.txt
```

- Delete lines containing "debug" in file.txt:

```
sed '/debug/d' file.txt
```

---

awk

Powerful text-processing tool, often used for filtering and formatting text.

- Print the second column of a space-separated file:

```
awk '{print $2}' file.txt
```

- Print lines where column 3 is greater than 50:

```
awk '$3 > 50' file.txt
```

---

cut

Extracts specific columns or characters from a file.

- Extract the first 5 characters from each line of file.txt:

```
cut -c 1-5 file.txt
```

- Extract the second column from a CSV file:

```
cut -d ',' -f2 file.csv
```

---

sort
Sorts lines in a file.

- Sort a file alphabetically:

```
sort file.txt
```

- Sort numerically in descending order:

```
sort -nr numbers.txt
```

---

uniq
Removes duplicate lines from a sorted file.

- Remove duplicate lines from a file:

```
sort file.txt | uniq
```

- Count occurrences of each unique line:

```
sort file.txt | uniq -c
```

---

tar
Creates and extracts compressed archives.

- Create a tar archive:

```
tar -cvf archive.tar file1 file2
```

- Extract a tar archive:

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

---

zip/unzip
Used for compressing and extracting ZIP files.

- Create a ZIP archive:

```
zip archive.zip file1 file2
```

- Extract a ZIP file:

```
unzip archive.zip
```

---

df
Displays available disk space.

- Show human-readable disk usage:

```
df -h
```

- Show disk usage of /home:

```
df -h /home
```

---

du

Shows the size of files and directories.

- Show the size of myfolder/:

```
du -sh myfolder/
```

- Show the size of all files in myfolder/:

```
du -ah myfolder/
```

---

ps

Displays running processes.

- Show all running processes:

```
ps aux
```

- Show processes for a specific user:

```
ps -u username
```

---

kill

Terminates running processes.

- Kill a process by its PID:

```
kill 1234
```

- Kill all processes named "firefox":

```
pkill firefox
```

---

top

Monitors real-time CPU and memory usage.

- Run top to see system stats:

```
top
```

- Sort processes by memory usage:

```
top -o %MEM
```

---

htop

Interactive process manager with an easy-to-use interface.

- Start htop:

```
htop
```

- Sort by CPU usage inside htop:

Press F6 → Select CPU%

---

chmod
Changes file permissions.

- Grant execute permission:

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

- Set exact permissions (rwxr-xr--):

```
chmod 755 file.sh
```

---

chown
Changes file ownership and group.

- Change file owner to user1:

```
chown user1 file.txt
```

- Change owner and group:

```
chown user1:group1 file.txt
```

---

scp
Securely copies files between systems.

- Copy file to remote server:

```
scp file.txt user@remote:/home/user/
```

- Copy an entire directory:

```
scp -r folder/ user@remote:/home/user/
```

---

rsync
Synchronizes files and directories efficiently.

- Sync local and remote directories:

```
rsync -avz /local/dir/ user@remote:/remote/dir/
```

- Sync without overwriting unchanged files:

```
rsync -avzu /local/dir/ user@remote:/remote/dir/
```

---

curl
Transfers data from URLs using different protocols.

- Download a file from a URL:

```
curl -O https://example.com/file.zip
```

- Send a GET request to an API:

```
curl https://api.example.com/data
```