Advanced Linux Commands

1. Process Management

Linux is a multitasking OS where many processes run simultaneously. These commands help monitor and control them.

- ps aux → Shows all running processes with details (PID, user, CPU/memory usage).
- top → Real-time view of active processes.
- htop (if installed) → Interactive process viewer with colors and navigation.
- kill <PID> → Terminate a process by its process ID.
- kill -9 <PID> → Force kill a process.
- jobs → Show background jobs started in the current session.
- fg %1 → Bring job number 1 to the foreground.
- bg %1 → Resume a stopped job in the background.

Use Case: If a program freezes, find its PID using ps aux and kill it.

2. Disk & File System Management

Efficient disk monitoring and file system operations are critical for system administration.

- df -h → Show disk usage in human-readable format.
- du -sh <dir> → Show total size of a directory.
- mount <device> <dir> → Mount a storage device.
- umount <device> → Unmount a storage device.
- lsblk → List available block devices (disks, partitions).
- fdisk -l → Show partition details.

Use Case: Before adding a new disk to a server, check partitions with lsblk and mount it properly.

3. Searching & Filtering

Working with logs or large data requires searching.

- grep "error" logfile.log → Find lines containing “error.”
- grep -i "error" logfile.log → Case-insensitive search.
- grep -r "TODO" /project → Search recursively in a directory.

- find /home -name "*.txt" → Find all .txt files under /home.
- locate filename → Quickly find files (uses prebuilt database).
- wc -l file.txt → Count lines in a file.

Use Case: Developers use grep to locate error messages in log files.

4. Archiving & Compression

Linux supports multiple compression and packaging utilities.

- tar -cvf archive.tar files/ → Create tar archive.
- tar -xvf archive.tar → Extract tar archive.
- tar -czvf archive.tar.gz files/ → Create compressed tarball.
- tar -xzvf archive.tar.gz → Extract compressed tarball.
- zip archive.zip file1 file2 → Create zip file.
- unzip archive.zip → Extract zip file.

Use Case: System admins compress log files before sending them for backup.

5. Networking Commands

Networking is essential for troubleshooting connectivity and security.

- ping google.com → Test connectivity.
- curl https://example.com → Fetch web content.
- wget https://example.com/file → Download a file.
- ifconfig or ip addr → Show IP and network configuration.
- netstat -tulnp → Show open ports and services.
- ss -tulnp → Modern replacement for netstat.
- scp file.txt user@remote:/path → Copy files over SSH.
- rsync -av source/ destination/ → Sync files efficiently.
- ssh user@host → Remote login.

Use Case: If a web service is down, ping tests network reachability, and netstat checks if the port is listening.

6. User & Permission Management

Beyond basics, admins need advanced control.

- useradd newuser → Add new user.
- passwd newuser → Set password.
- usermod -aG sudo newuser → Add user to sudo group.
- deluser newuser → Delete a user.
- groups → Show groups current user belongs to.
- umask → Show/modify default permissions for new files.

Use Case: A DevOps engineer gives a developer sudo rights with usermod.

7. System Monitoring & Logs

Logs help diagnose issues and maintain security.

- dmesg → Show kernel messages (hardware, drivers).
- journalctl → View systemd logs.
- tail -f /var/log/syslog → Continuously monitor system logs.
- uptime → Show system load and running time.
- free -h → Show memory usage.
- iostat (if installed) → Show CPU/disk performance.

Use Case: To troubleshoot a server crash, admins check /var/log/ with journalctl and dmesg.

8. Package Management

Package managers differ by distribution.

Debian/Ubuntu

- apt update → Update package list.
- apt upgrade → Upgrade installed packages.
- apt install nginx → Install package.
- apt remove nginx → Remove package.

RHEL/CentOS/Fedora

- yum install httpd or dnf install httpd → Install Apache.
- yum update → Update system packages.
- yum remove httpd → Remove Apache.

Use Case: Installing and updating software like nginx or mysql.

9. Advanced File Handling

- ln file linkfile → Create hard link.
- ln -s file symlink → Create symbolic (soft) link.
- stat file → Show detailed file information.
- file filename → Show file type.
- diff file1 file2 → Compare two files line by line.
- cmp file1 file2 → Compare files byte by byte.

Use Case: Developers use diff before merging code changes.

10. Shell Scripting (Intro)

Once students know commands, automation with scripts is next.

- A script is just a text file with commands.
- Start with #!/bin/bash at the top.
- Example:

```
#!/bin/bash
echo "Backup starting..."
tar -czvf backup.tar.gz /home/student
echo "Backup completed."
```

Run it with:

```
chmod +x backup.sh
./backup.sh
```

Use Case: Automating daily backup tasks.

Practical Tasks – Advanced Linux Commands

1. Process Management

- List all running processes with ps aux.
- Start a program (e.g., gedit or sleep 1000) and find its PID.
- Kill it using kill <PID>.
- Run sleep 2000 & and bring it to foreground with fg.

---

2. Disk & File System

- Check total disk usage with df -h.
- Find out the size of /var/log using du -sh /var/log.
- Plug in a USB drive (if available) and check its details with lsblk.

---

3. Searching & Filtering

- Create a file sample.txt with multiple lines.
- Search for the word Linux in it using grep Linux sample.txt.
- Count how many lines it has with wc -l sample.txt.
- Use find to locate all .sh files under /home.

---

4. Archiving & Compression

- Create a directory backup_test with 3 files.
- Archive it into backup.tar using tar -cvf.
- Compress it into backup.tar.gz and then extract it back.
- Create a zip file of the directory using zip -r backup.zip backup_test/.

---

5. Networking

- Use ping google.com and stop it with Ctrl+C.
- Download a webpage using curl https://example.com.
- Check your system's IP address with ip addr.
- Use ssh user@localhost (if SSH is enabled) to log into your own system.

6. User & Permissions

- Create a new user testuser with sudo adduser testuser.
- Switch to that user using su testuser.
- Create a file and try changing its permissions with chmod 700 filename.
- Add testuser to the sudo group using usermod -aG sudo testuser.

---

7. System Monitoring

- Run top and observe which processes use most CPU and memory.
- Monitor system logs in real-time with tail -f /var/log/syslog.
- Check how long the system has been running using uptime.
- View kernel messages with dmesg | less.

---

8. Package Management

- Update your system with sudo apt update && sudo apt upgrade (Ubuntu/Debian).
- Install htop and run it.
- Remove a package you don't need with sudo apt remove <package>.

---

9. Advanced File Handling

- Create a file notes.txt.
- Create a hard link notes_hard and a soft link notes_soft pointing to it.
- Compare two text files with diff file1.txt file2.txt.
- Use stat notes.txt to check inode and permissions.

---

10. Shell Scripting

- Write a script myscript.sh that:
    1. Prints today's date.
    2. Creates a directory mybackup.
    3. Archives /home/<youruser> into mybackup/home_backup.tar.gz.
- Make it executable and run it.