LINUX STUDY NOTES

1. FILE SYSTEM STRUCTURE

Linux follows a hierarchical directory structure starting from root (/).

Key Directories:
/bin -> Essential user commands
/etc -> Configuration files
/home -> User directories
/var -> Logs and variable data
/usr -> Installed software
/opt -> Optional packages
/tmp -> Temporary files

Real-time Example:
Logs are stored in /var/log. In production servers, DevOps engineers check logs using:
cd /var/log
ls -lrt

Interview Question (Basic):
Q: What is the difference between /bin and /usr/bin?
A: /bin contains essential system binaries needed during boot, while /usr/bin contains non-essential user binaries.

---

2. NAVIGATION & FILE OPERATIONS

pwd -> Print working directory
ls -> List files
cd -> Change directory
cp -> Copy
mv -> Move/rename
rm -> Delete
mkdir -> Create directory
touch -> Create file

Real-time Use Case:
During deployment:
cp app.war /opt/tomcat/webapps/

Common Mistake:
Using rm -rf without checking path. Always verify with pwd before deleting.

---

3. FILE PERMISSIONS (chmod, chown)

Permission Types:
r = read
w = write

x = execute

Numeric values:
4 = read
2 = write
1 = execute

Example:
chmod 755 script.sh
chown user:group file.txt

Interview Question (Intermediate):
Q: What does chmod 777 mean?
A: It gives read, write, execute permissions to owner, group, and others. It is insecure in production.

---

4. PROCESS MANAGEMENT
---
ps aux -> List processes
top -> Real-time process monitor
kill PID -> Kill process
kill -9 PID -> Force kill

Real Scenario:
Application not responding:
ps aux | grep java
kill -9 PID

---

5. DISK & MEMORY COMMANDS
---
df -h -> Disk usage
du -sh * -> Folder size
free -m -> Memory usage
uptime -> Load average

Troubleshooting:
If disk is full:
du -sh /var/log/*

---

6. USER & GROUP MANAGEMENT
---
useradd username
passwd username
groupadd groupname
usermod -aG group user

Real-time:
Grant sudo:
usermod -aG wheel user

---

7. SSH & SCP
---
ssh user@ip
scp file user@ip:/path

Key-Based Authentication:
ssh-keygen
ssh-copy-id user@ip

---

8. CRON JOBS
---
crontab -e

Example:
0 2 * * * /home/user/backup.sh

---

9. ENVIRONMENT VARIABLES
---
echo $PATH
export VAR=value
source ~/.bashrc

SHELL SCRIPTING
==========

1. VARIABLES
---
name="DevOps"
echo $name

2. IF ELSE
---
if [$a -gt 10]; then
echo "Greater"
else
echo "Smaller"
fi

3. LOOPS
---
for i in {1..5}
do
echo $i
done

```
while [ $a -lt 5 ]
do
((a++))
done
```

4. FUNCTIONS

```
function greet() {
echo "Hello"
}
```

5. TEXT PROCESSING

```
grep "error" file.log
awk '{print $1}' file.txt
sed 's/old/new/g' file.txt
cut -d',' -f1 file.csv
```

---

6. EXIT CODES

```
echo $?
```

0 -> Success
1 -> Failure

REAL DEPLOYMENT SCRIPT EXAMPLE

```
#!/bin/bash
git pull
docker build -t app .
docker stop app
docker run -d -p 80:80 app
```

GIT & VERSION CONTROL
==========

1. CORE WORKFLOW

```
git clone
git add .
git commit -m "msg"
git push
```

2. BRANCHING STRATEGIES

GitFlow -> Feature, develop, main branches
Trunk-Based -> Single main branch

3. UNDOING MISTAKES
---
git reset
git revert
git checkout -- file

4. AZURE REPOS

Branch Policies:
- Minimum reviewers
- Build validation
- PR approvals

INTERVIEW QUESTIONS
==========

Basic:
Q: What is CI/CD?
A: Continuous Integration and Continuous Deployment automate build and release process.

Intermediate:
Q: How do you secure SSH access?
A: Disable password auth, use key-based auth, restrict root login.

Advanced:
Q: How do you troubleshoot high CPU usage?
A: Use top, ps aux, identify process, check logs, scale if required.

PRACTICE ASSIGNMENTS
==========

Easy:
- Create users and assign permissions
- Write script to print numbers 1-10

Medium:
- Write backup script using cron
- Parse logs using awk and grep

Difficult:
- Write full deployment script
- Implement GitFlow in sample project

MINI PROJECT IDEAS
==========
- Create automated log monitoring script
- Build CI pipeline with Git and shell
- Setup cron-based backup system

KEY REVISION POINTS
==========
- File permissions numeric values
- Common Linux commands
- Git branching basics
- Exit codes
- SSH key setup

END OF NOTES