

Basic File System Structure and Permissions (Linux)

1. File System Structure

- The file system defines how files and directories are stored and organized.
- Everything is treated as a file, including devices and processes.
- The root directory (/) is the top-level directory.

2. Important Directories

/ – Root directory, base of the file system
/home – User personal directories
/bin – Essential user commands
/sbin – System administration commands
/etc – Configuration files
/var – Variable data like logs
/tmp – Temporary files
/usr – User-installed software and libraries
/dev – Device files
/proc – Process and system info (virtual)

3. File Types

- Regular files – text, binary, scripts
- Directories – contain files
- Symbolic links – shortcuts to files
- Device files – hardware access
- Sockets & pipes – inter-process communication

4. File Permissions

- Permissions define who can read, write, or execute a file.
- Three entities: Owner, Group, Others
- Permission types:
 - r – Read
 - w – Write
 - x – Execute

5. Permission Representation

- Symbolic: rwxr-xr--
- Numeric: 754
r=4, w=2, x=1

6. Common Permission Commands

ls -l → View permissions
chmod → Change permissions
chown → Change owner
chgrp → Change group

7. Example Commands

```
ls -l
chmod 755 script.sh
chmod u+x file.sh
chown user file.txt
chgrp group file.txt
```

8. Practical Examples

- Make a file executable:
chmod +x app.sh
- Restrict access:
chmod 600 secret.txt
- Grant full access:
chmod 777 test.txt

Terminal Command Screenshots

Screenshot 1: Viewing file permissions using ls -l

```
student@student-virtual-machine:~/25SUB4508_56133$ ls -l
total 4
dr-xr-xr-x 2 student student 4096 Dec 16 12:53 day_09
student@student-virtual-machine:~/25SUB4508_56133$ █
```

Screenshot 2: Changing permissions using chmod

```
student@student-virtual-machine:~/25SUB4508_56133$ touch test.sh
student@student-virtual-machine:~/25SUB4508_56133$ ll
total 12
drwxrwxr-x 3 student student 4096 Dec 16 14:39 .
drwxr-x--- 22 student student 4096 Dec 16 11:39 ..
dr-xr-xr-x 2 student student 4096 Dec 16 12:53 day_09/
-rw-rw-r-- 1 student student 0 Dec 16 14:39 test.sh
student@student-virtual-machine:~/25SUB4508_56133$ chmod 755 test.sh
student@student-virtual-machine:~/25SUB4508_56133$ ll
total 12
drwxrwxr-x 3 student student 4096 Dec 16 14:39 .
drwxr-x--- 22 student student 4096 Dec 16 11:39 ..
dr-xr-xr-x 2 student student 4096 Dec 16 12:53 day_09/
-rwxr-xr-x 1 student student 0 Dec 16 14:39 test.sh*
student@student-virtual-machine:~/25SUB4508_56133$ █
```

Screenshot 3: Changing ownership using chown

```
-rwxr-xr-x 1 student student 0 Dec 16 14:39 test.sh
student@student-virtual-machine:~/25SUB4508_56133$ chown souvik test.sh
```

Exploring Linux File System Hierarchy

1. What is File System Hierarchy

- The file system hierarchy defines the standard directory layout.
- It ensures consistency across systems.
- All files and directories originate from the root (/).

2. Root Directory (/)

- Top-most directory in the hierarchy.
- Contains all other system directories.
- Critical for system boot and operation.

3. Key System Directories

/bin – Essential user commands (ls, cp, mv)
/sbin – System administration commands
/etc – System-wide configuration files
/lib – Shared libraries needed by system binaries
/boot – Bootloader and kernel files

4. User & Application Directories

/home – Personal directories for users
/usr – User applications, libraries, documentation
/opt – Optional or third-party software

5. Variable & Temporary Data

/var – Logs, mail, spool files
/tmp – Temporary files cleared on reboot

6. Virtual File Systems

/proc – Process and kernel info (virtual)
/sys – Hardware and device information

7. Device Management

/dev – Represents hardware devices as files
• Example: hard disks, USB, terminals

8. Navigation Commands (Examples)

pwd → Show current directory
ls / → List root directories
cd /home → Move to home directory
tree / → View directory structure (if installed)

9. Disk Usage & Structure Commands

df -h → Disk space usage
du -sh * → Directory size
mount → View mounted file systems

10. Practical Terminal Examples

```
ls /  
cd /etc  
pwd  
df -h  
du -sh /home
```

Terminal Command Screenshots

Screenshot 1: Listing root directory using ls /

```
student@student-virtual-machine:~/25SUB4508_56133$ ls /  
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var  
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr  
student@student-virtual-machine:~/25SUB4508_56133$
```

Screenshot 2: Navigating directories using cd and pwd

```
student@student-virtual-machine:~$ cd .  
student@student-virtual-machine:~$ cd 25SUB4508_56133/  
student@student-virtual-machine:~/25SUB4508_56133$ pwd  
/home/student/25SUB4508_56133  
student@student-virtual-machine:~/25SUB4508_56133$
```

Standard Directories and Their Roles

1. Overview

- Linux follows a standard directory structure.
- Each directory has a specific purpose.
- This structure helps in system organization and administration.

2. Root Level Directories

/bin – Essential user commands required for basic operations
/sbin – System administration commands
/boot – Bootloader and kernel-related files
/lib – Shared libraries needed for system binaries

3. Configuration and System Files

- /etc – System-wide configuration files
- Stores settings for services, users, and network
 - Mostly text-based configuration files

4. User and Application Data

- /home – Home directories for all users
- /usr – User-installed applications, binaries, and documentation
- /opt – Optional or third-party application software

5. Variable and Runtime Data

- /var – Variable data such as logs, mail, and cache
- /run – Runtime data created after boot

6. Temporary Storage

- /tmp – Temporary files created by applications
- Files are usually deleted after reboot

7. Device and Virtual Directories

- /dev – Device files representing hardware
- /proc – Process and kernel information (virtual)
- /sys – Hardware and kernel interface (virtual)

8. Common Directory Commands (Examples)

- | | |
|-------------|----------------------------|
| ls /etc | → List configuration files |
| ls /home | → List user directories |
| ls /var/log | → View system logs |
| ls /dev | → View device files |

9. Practical Terminal Examples

```
ls /  
cd /var/log  
ls  
cd /home  
Pwd
```

File Permissions and Ownership

1. Overview

- File permissions control who can access a file or directory.
- Ownership determines which user and group control the file.
- These mechanisms ensure system security and controlled access.

2. Types of Permissions

- Read (r) – Allows viewing file contents or listing a directory
- Write (w) – Allows modifying file contents or directory structure
- Execute (x) – Allows running a file or accessing a directory

3. Permission Categories

- Owner – The user who owns the file
- Group – Users belonging to the file's group
- Others – All other users

4. Permission Representation

- Symbolic format: rwxr-xr--
 - Numeric format: 754
- r = 4, w = 2, x = 1

5. Viewing Permissions

ls -l → Displays permissions and ownership information

6. Changing Permissions (chmod)

chmod 755 file.sh → Set permissions using numeric method
chmod u+x file.sh → Add execute permission to owner
chmod go-r file.txt → Remove read permission from group and others

7. File Ownership

- Every file has an owner and a group
- Ownership affects permission enforcement

8. Changing Ownership (chown, chgrp)

chown user file.txt → Change file owner
chown user:group file.txt → Change owner and group
chgrp group file.txt → Change group ownership

9. Recursive Operations

chmod -R 755 folder_name → Change permissions recursively
chown -R user folder_name → Change ownership recursively

10. Practical Terminal Examples

```
ls -l
chmod 644 notes.txt
chmod +x script.sh
chown user file.txt
chgrp developers file.txt
```

Changing File Ownership

1. Overview

- File ownership defines which user and group control a file or directory.
- Ownership works together with permissions to enforce access control.
- Only the superuser can change ownership of files.

2. File Owner and Group

- Owner – The user who owns the file
- Group – A collection of users sharing access rights
- Ownership is displayed using the ls -l command

3. Viewing Ownership

ls -l file.txt → Shows owner and group of the file
stat file.txt → Displays detailed ownership information

4. Changing File Owner (chown)

chown user file.txt → Change only the owner
chown user:group file.txt → Change owner and group together

5. Changing Group Ownership (chgrp)

chgrp group file.txt → Change only the group ownership

6. Recursive Ownership Changes

chown -R user folder_name → Change ownership of all files inside a directory

7. Ownership with Permissions

- Ownership determines which permission set applies.
- Owner permissions override group and others permissions.

Reading and Modifying File Permissions

1. Overview

- File permissions determine how files and directories can be accessed.
- Reading permissions helps understand access rights.
- Modifying permissions allows controlled access management.

2. Reading File Permissions

ls -l file.txt → Displays permissions, owner, group, and size
stat file.txt → Shows detailed permission and file metadata

3. Understanding Permission Output

Example: -rwxr-xr--

- First character: file type
- Next 3: owner permissions
- Next 3: group permissions
- Last 3: others permissions

4. Modifying Permissions Using Numeric Method

chmod 644 file.txt → Owner read/write, others read
chmod 755 script.sh → Owner full access, others read/execute

5. Modifying Permissions Using Symbolic Method

chmod u+r file.txt → Add read permission to owner
chmod g-w file.txt → Remove write permission from group
chmod o+x script.sh → Add execute permission for others

6. Directory Permission Behavior

- Read: list directory contents
- Write: create/delete files
- Execute: access directory contents

7. Recursive Permission Changes

chmod -R 700 private_dir → Apply permissions to all files inside

8. Practical Terminal Examples

```
ls -l
stat notes.txt
chmod 600 secret.txt
chmod u+x app.sh
chmod -R 755 project_dir
```

Terminal Command Screenshots

Screenshot : Modifying permissions using chmod (numeric)

```
student@student-virtual-machine:~/25SUB4508_56133$ touch test.sh
student@student-virtual-machine:~/25SUB4508_56133$ ll
total 12
drwxrwxr-x  3 student student 4096 Dec 16 14:39 .
drwxr-x--- 22 student student 4096 Dec 16 11:39 ../
dr-xr-xr-x  2 student student 4096 Dec 16 12:53 day_09/
-rw-rw-r--  1 student student    0 Dec 16 14:39 test.sh
student@student-virtual-machine:~/25SUB4508_56133$ chmod 755 test.sh
student@student-virtual-machine:~/25SUB4508_56133$ ll
total 12
drwxrwxr-x  3 student student 4096 Dec 16 14:39 .
drwxr-x--- 22 student student 4096 Dec 16 11:39 ../
dr-xr-xr-x  2 student student 4096 Dec 16 12:53 day_09/
-rwxr-xr-x  1 student student    0 Dec 16 14:39 test.sh*
student@student-virtual-machine:~/25SUB4508_56133$ █
```

References

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
- Love, R. (2010). Linux Kernel Development (3rd ed.). Addison-Wesley.
- Shotts, W. E. (2019). The Linux Command Line (2nd ed.). No Starch Press.
- Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook (5th ed.). Pearson.
- Linux Foundation. Filesystem Hierarchy Standard (FHS), Version 3.0.