Name : Aditya Rajendra Gadhave

Task 1 : basic File Operations

➢ Linux File Hierarchy Structure:

- Linux adopts a hierarchical directory structure
- That organizes files in a tree-like format, starting from the root (/) directory.
- This structure is standard across all Linux distributions and follows the **FHS** (Filesystem Hierarchy Standard).
- The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems.
- It is maintained by the Linux Foundation.
- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.

Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS and are not considered authoritative for platforms other than Linux.

1. / (Root):

Primary hierarchy root and root directory of the entire file system hierarchy.

- Every single file and directory start from the root directory.
- The only root user has the right to write under this directory.
- /root is the root user’s home directory, which is not the same as /

2. /bin :

Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.

- Contains binary executables.

- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp

3. /boot :

Boot loader files, e.g., kernels, initrd.

- Kernel initrd, vmlinux, grub files are located under /boot
- Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

4. /dev :

Essential device files, e.g., /dev/null.

- These include terminal devices, usb, or any device attached to the system.
- Example: /dev/tty1, /dev/usbmon0

5. /etc :

Host-specific system-wide configuration files.

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- Example: /etc/resolv.conf, /etc/logrotate.conf.

6. /home :

Users' home directories, containing saved files, personal settings, etc.

- Home directories for all users to store their personal files.
- example: /home/kishlay, /home/kv

7. /lib:

Libraries essential for the binaries in /bin/ and /sbin/.

- Library filenames are either ld* or lib*.so.*
- Example: ld-2.11.1.so, libncurses.so.5.7

8./usr :

Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains user's programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2
- /usr/src holds the Linux kernel sources, header-files and documentation.File System in Linux
- The **file system** in Linux refers to how files and data are organized and stored on a disk or storage device.
- It defines the structure and logic rules that dictate how files are named, stored, and retrieved.
- Linux supports multiple file systems, such as **ext4, XFS, Btrfs, FAT32, NTFS**, and many others.

➢ Common Linux File Systems

1. **ext4 (Fourth Extended File System)**:
    - **Default file system** in most Linux distributions like Ubuntu and Fedora.
    - Supports large volumes (up to 1 exabyte) and file sizes (up to 16 terabytes).
    - Provides journaling, which helps recover data after a crash.

- Backward compatibility with older ext2 and ext3 file systems.

2. **XFS**:
    - High-performance file system used in RHEL and CentOS.
    - Known for its scalability and is ideal for large data volumes.
    - Supports snapshotting, a feature often used in backup solutions.
3. **Btrfs (B-tree File System)**:
    - Focuses on fault tolerance, easy management, and advanced features like snapshotting and pooling multiple disks.
    - Great for managing large data sets and frequent snapshots.
    - Still considered experimental in some cases, though widely adopted.
4. **FAT32 and NTFS**:
    - **FAT32** is used for smaller, portable drives (e.g., USB flash drives) and is compatible with many operating systems.
    - **NTFS** is a Windows file system, but Linux can mount and work with NTFS drives using drivers like ntfs-3g.
5. **Swap**:
    - Special partition type in Linux used for swapping memory when RAM is full.
    - Not a traditional file system but plays a role in managing memory and virtual memory.

Basic File Operations in Linux

1. **List Files**:
    - List files in the current directory:

      Command: ls

    - List with more details (permissions, size, etc.):

      Command: ls -l

2. **Create a File**:

```
touch Adi.txt
```

3. **Create a Directory**:

```
mkdir Aditya_G
```

4. **Copy Files**:

```
cp Adi.txt  Adi2.txt
```

5. **Move/Rename Files**:

```
mv Adi.txt new_directory/
```

6. **Delete Files**:

```
rm Aditya.txt
```

7. **View File Contents**:
   - Display the content of a file:

     ```
     cat Adi.txt
     ```

   - View the first 10 lines of a file:

     ```
     head Adi. txt
     ```

   - View the last 10 lines of a file:

     ```
     tail Adi.txt
     ```

➢ File Permissions in Linux :

File permissions determine which users can read, write, or execute a file. Each file or directory has three sets of permissions: **owner, group, and others**.

- **r**: Read permission.
- **w**: Write permission.
- **x**: Execute permission.

Example of file permissions:

```
code-rwxr-xr--
```

- The first character (-) indicates the file type (in this case, a regular file).
- The next three (rwx) represent permissions for the file owner.
- The next three (r-x) represent permissions for the group.
- The last three (r--) represent permissions for others.

You can change file permissions using the chmod command:

```
chmod 755 file.txt
```

Ownership in Linux

Each file in Linux has an owner and an associated group. You can change the owner and group using the chown command:

```
sudo chown user:group file.txt
```

Conclusion :

The **Linux file system** is highly organized and follows a standard hierarchy, making it easier for users and administrators to manage files, directories, and data. Understanding **file architecture** (directories, permissions, and file types) and working with different **file systems** like ext4, XFS, or Btrfs allows you to efficiently manage data storage and perform essential tasks such as mounting devices, formatting partitions, and controlling access to files.