



Introduction to Linux

Jayesh Deep Dubey
Technical Officer
C-DAC Patna

What is Linux?

- Linux is a free and open-source operating system that was created by Linus Torvalds in 1991
- It is based on the Unix operating system and is known for its stability, security, and flexibility
- Main advantages of Linux is its open-source nature, which means that the source code of the operating system is freely available for anyone to use, modify, and distribute
- Linux distribution is the Linux kernel and collection of software packages.

GNU

- GNU stands for "GNU's Not Unix"
- It is a free and open-source software ecosystem that was created by Richard Stallman and his team at the Free Software Foundation (FSF) in the 1984.
- The GNU project has developed a wide range of software tools and utilities, including compilers, text editors, command-line tools, and desktop environments.
- One of the most well-known components of the GNU project is the GNU General Public License (GPL), which is a widely-used software license that ensures that software released under the GPL remains free and open-source.

Features of Linux

- Open Source
- Secure
- Light Weight
- Multiuser and Multitasking
- Multiple distributions (RedHat, Debian, Ubuntu)
- Robust command-line interface
- Efficiency and performance
- It also has a wide range of graphical user interfaces available, such as GNOME, KDE, and Xfce

File System Structure in Linux

- In Linux, the file system structure is hierarchical and follows the Filesystem Hierarchy Standard (FHS), which specifies the standard directory layout for Linux-based operating systems.
- Following are main directories and their functions in the Linux file system:
 - **/ (root)**: The top-level directory in the file system hierarchy. All other directories and files are contained within this directory
 - **/root**: The home directory for the root user, which is the system administrator.
 - **/home**: Contains user home directories, where users can store their personal files and settings

File System Structure in Linux

- Following are main directories and their functions in the Linux file system:
 - **/bin**: Contains essential system binaries (programs) that are required to boot the system and perform basic system maintenance.
 - **/boot**: Contains files used by the boot loader to start the operating system, such as the kernel, bootloader configuration files, and boot loader menu.
 - **/dev**: Contains device files that represent hardware devices connected to the system, such as hard drives, USB drives, and other peripherals.
 - **/etc**: Contains system configuration files that are used by the system and applications.

File System Structure in Linux

- Following are main directories and their functions in the Linux file system:
 - **/lib and /lib64**: Contains essential shared library files that are required by the system and applications
 - **/media**: Used to automatically mount removable media devices such as USB drives and CD/DVDs.
 - **/usr**: Contains user-related programs, libraries, documentation, and shared resources.
 - **/opt**: Contains optional software packages that are not installed by default, but can be installed by the user

File System Structure in Linux

- Following are main directories and their functions in the Linux file system:
 - **/sbin**: Contains system binaries that are used for system administration tasks.
 - **/tmp**: Contains temporary files that are created and used by the system and applications.
 - **/var**: Contains variable data files, such as log files, mail spools, and printer spools

Basic Linux Commands

- 1) **ls**: This command is used to list the contents of a directory. By default, it displays the files and directories in the current directory.

Syntax: `ls [options] [file/directory name]`

Example: `ls -l /home/user/Desktop`

Basic Linux Commands

2) **cd**: This command is used to change the current directory.

Syntax: cd [directory name]

Example: cd /home/user/Documents

3) **pwd**: This command is used to display the current directory.

Syntax: pwd

Example: pwd

Basic Linux Commands

4) **mkdir**: This command is used to create a new directory.

Syntax: mkdir [directory name]

Example: mkdir my_new_directory

5) **touch**: This command is used to create an empty file.

Syntax: touch [file name]

Example: touch my_new_file.txt

Basic Linux Commands

6) **cat**: This command is used to display the contents of a file.

Syntax: cat [file name]

Example: cat file.txt

7) **rm**: This command is used to remove files or directories.

Syntax: rm [options] [file/directory name]

Example: rm -r my_directory

Basic Linux Commands

8) **cp**: This command is used to copy files or directories from one location to another.

Syntax: `cp [options] [source file/directory] [destination directory]`

Example: `cp file.txt /home/user/Documents`

9) **mv**: This command is used to move files or directories from one location to another.

Syntax: `mv [options] [source file/directory] [destination directory]`

Example: `mv file.txt /home/user/Documents`

Basic Linux Commands

10) **ps**: This command is used to display the running processes.

Syntax: ps [options]

Example: ps aux

11) **top**: This command is used to display the running processes and their resource usage in real-time. It provides a dynamic view of the system's performance.

Syntax: top

Example: top

Basic Linux Commands

12) **grep**: This command is used to search for a specific pattern or string in a file.

Syntax: `grep [options] [pattern] [file name]`

Example: `grep "example" file.txt`

13) **ssh**: This command is used to establish a secure remote connection to another system. It encrypts all data transmitted between the client and server.

Syntax: `ssh [username]@[IP address]`

Example: `ssh user1@192.168.0.1`

Basic Linux Commands

14) **ifconfig**: This command is used to display network interface information, including IP address, netmask, and network status.

Syntax: ifconfig [options]

Example: ifconfig

15) **ping**: This command is used to test network connectivity by sending ICMP packets to a specified host.

Syntax: ping [options] [host name/IP address]

Example: ping google.com

Basic Linux Commands

16) **telnet**: This command is used to initiate a Telnet session with a remote server. Telnet is a protocol used for connecting to remote servers over the internet. It allows users to establish a connection to a remote host and access its services

Syntax: telnet [IP address/hostname] [port]

Example: telnet 192.168.0.1 80

Basic Linux Commands

17) **ftp**: The ftp command is used to connect to an FTP server and transfer files between the client and server. FTP (File Transfer Protocol) is a protocol used for transferring files over the internet.

Syntax: ftp [IP address/hostname]

Example: ftp ftp.example.com

Basic Linux Commands

18) **sftp**: This command uses SSH for encryption and authentication, providing a secure way to transfer files between systems. SFTP (Secure File Transfer Protocol) is a protocol used for secure file transfer over the internet.

Syntax: `sftp [user]@[IP address/hostname]`

Example: `sftp user@example.com`

Basic Linux Commands

19) **finger**: finger is a command used to retrieve information about a user on a remote system. It displays information such as the user's full name, login name, and last login time.

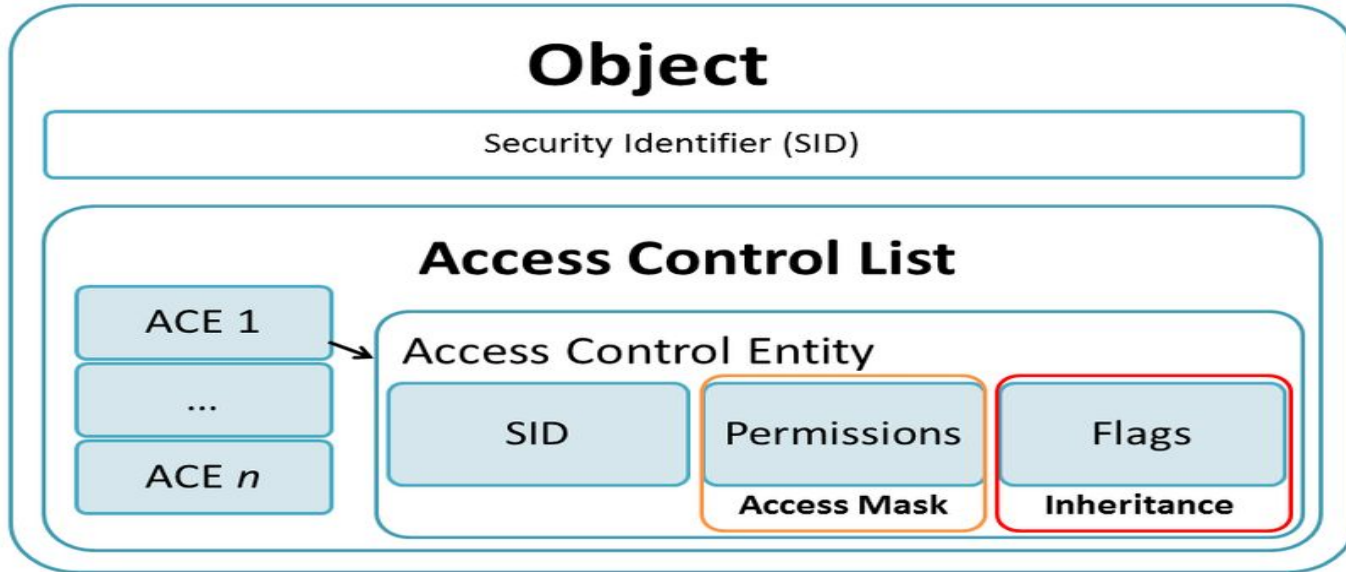
Syntax: `finger [user]@[IP address/hostname]`

Example: `finger user@example.com`

Access Control List

- An Access Control List (ACL) is a mechanism in an operating system that controls access to system resources such as files and directories.
- It is a list of permissions associated with an object, which specifies the users and groups that are granted access to that object and the type of access that they are granted
- ACL allows for more fine-grained control over access to system resources

Access Control List



Access Control List

The ACL system typically includes the following components:

- **An object:** This is the resource that the ACL is associated with, such as a file or directory.
- **An access control entry (ACE):** This is a record that specifies a particular user or group and the type of access that they are granted to the object.
- **A security identifier (SID):** This is a unique identifier that is assigned to each user or group on the system

Access Control List

The ACL system typically includes the following components:

- **A permission:** This specifies the type of access that is granted, such as read, write, or execute.
- **Flags:** They are used to control how the permissions specified in the ACL are applied to the object associated with it.
 - **Inheritable:** This flag is used to indicate whether the ACE can be inherited by child objects. If the flag is set, the ACE is propagated to child objects when they are created. If the flag is not set, the ACE is not propagated

Files, Directories and Inodes

- **File:** A file represents a sequence of bytes.
- **Directory:** A directory represents a list of files.
- **Inode:** An inode (Index Node) contains information about a file (metadata) – File permissions, UID, Size, Time etc. The information about all the files will be maintained in a table called “**Inode Table**”

Users and Groups

Users: Users can be either people or accounts which exist to use specific applications of Linux.

- Each user will be given a unique user ID (UID).
- A root user will also be present and he has all the administrative privileges.

Groups: Users can be tied together into groups for a common purpose.

- Each group is associated with a group ID (GID).

Access Permissions

File Permissions: There are 3 permissions for any file r, w, x.

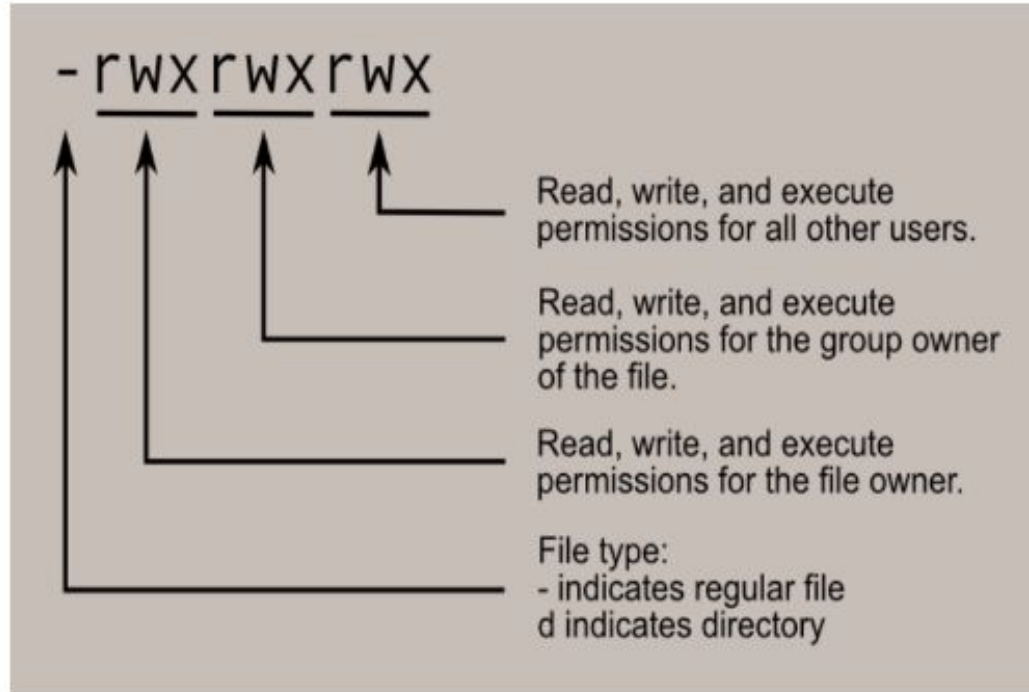
1. Read (r) - Indicates that a given category of user can read a file.
2. Write (w) - Indicates that a given category of user can write to a file.
3. Execute (x)- Indicates that a given category of user can execute the file.

Directory permissions:

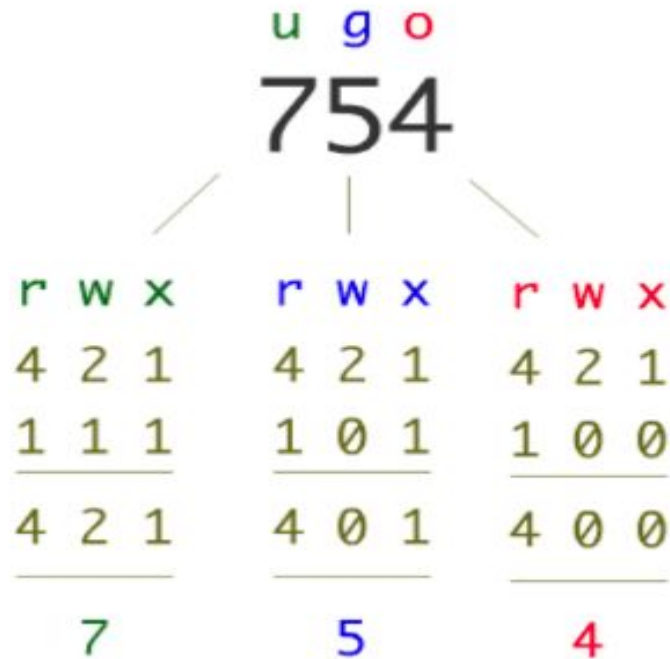
1. Read (r) - The directory can be read.
2. Write (w) - The directory can be updated, renamed or deleted.
3. Execute (x)- Operations can be performed on the files of the directories.

Categories of users: All of these three permissions are assigned to three categories of users – User (U), Group(G), Others(O)

Access Permissions



Access Permissions



Changing Permissions

- **chmod:** The chmod command changes the permissions associated with a file or directory. This command can be used to modify ACLs

Syntax: chmod mode file

Example: chmod 755 public_html

chmod – numeric modes

Consider permission for each set of users (user, group, other) as:

- r – 4
- w – 2
- x – 1

A permission (mode) for all 3 classes is:

- 755 – rwxr-xr-x
- 644 – rw-r--r--
- 700 – rwx-----

chown and chgrp

- **chown** – change file owner and group

Syntax: chown OWNER:GROUP FILE

Example: chown root:root myfile.txt

- **chgrp** - command is used to change the group of a file or directory

Syntax: chgrp <groupname> <file /directoryname>

Example: chgrp purchase s1.txt

Pipe Operator

- The pipe operator (|) is a powerful tool in Linux that allows you to send the output of one command as input to another command.
- This operator is useful in chaining multiple commands.

Example: Suppose you want to list all the files in a directory that contain the word "example". You can use the pipe operator to combine the ls and grep commands like this:

```
ls | grep example
```

Redirection Operator

- Redirection operator (>) is used to redirect the output of a command to a file instead of the terminal. The basic syntax of the redirection operator is as follows:

Syntax: command > filename

- If you want to append the output to the end of the file instead of overwriting it, you can use the double redirection operator (>>).

Syntax: command >> filename

Thank You !!