System Programming - Linux
Week 6 - Working with Directories

Lecturer: Dr. Aggrey Obbo (PhD)

April 22, 2025

Contents

1 Introduction

For effective file management and system navigation, it is essential to understand Linux folders. The hierarchical file system, directory permissions, and key tools like cd, ls, and mkdir are all covered in this lesson. Understanding these ideas gives the ability to efficiently manage data and use the Linux environment.

Objectives

- Comprehend the Linux Filesystem Hierarchy
- Navigate through the Directory Structure Efficiently
- Manage Directories Using Command-Line Tools
- Understand and Modify Directory Permissions
- Locate Specific Directories

2 Introduction to the Linux Filesystem Hierarchy

The root directory (/) is the base of the Linux file system, which has a hierarchical, tree-like structure. Data is organized by folders / directories, which branch out from root and contain files and subdirectories. With important directories like /home, /bin, and /etc fulfilling particular system roles and adhering to the Filesystem Hierarchy Standard (FHS), this layout offers a clear path for exploring and accessing data.

The Root Directory (/)

This is the foundation of everything. The top-level directory in the Linux file system hierarchy is the root directory, denoted by /. It acts as the hub from which all other files and directories radiate out, creating a structure like a tree. Root contains all of the necessary system directories, including /bin, /etc, and /home. The top-level directory in the Linux file system hierarchy is the root directory, denoted by /. It acts as the hub from which all other files and directories radiate out, creating a structure like a tree. Root contains all of the necessary system directories, including /bin, /etc, and /home.

The other standard linux directories and their purposes include:

/etc:

This the main location for system-wide configuration files, the /etc directory is essential to Linux. These files affect everything from network configurations to user management, controlling how the operating system and apps behave. It's where administrators modify system behavior.

/home:

Each user in Linux has their own unique space in the home directory. It is located under /home/ and contains configuration settings, documents, and files unique to the individual. Users can personalize their surroundings without impacting the system or other users thanks to this division, which guarantees privacy and organization.

/bin and /usr/bin:

/bin contains essential command-line utilities needed for basic system functions and boot processes. /usr/bin holds a wider range of executable programs for users, including applications and common tools. Historically separated for system recovery, they now often overlap, but serve distinct organizational roles.

/sbin and /usr/sbin:

System administration binaries that are necessary for boot, recovery, and repair are located in /sbin. Non-essential system administration binaries are stored in /usr/sbin and are utilized following the mounting of the /usr partition. While /sbin is required for early system operations, both are mainly for root and preserving system integrity.

/var:

Variable data that changes while the system is operating is kept in the /var directory. This comprises temporary files, spool files (such as print queues), log files, and caches. It is essential to system operation because, in contrast to the static data in /usr, it contains information that is always changing.

/tmp: Temporary

In Linux, temporary files generated by the operating system and applications are stored in the /tmp directory. Usually, these files are erased when the system restarts or on occasion. It offers a place for temporary data that doesn't have to last.

/mnt and /media:

Both /mnt and /media are mount points for file systems in Linux. /mnt is typically used to manually mount filesystems, such as network shares or hard drives. The typical place for automatically mounting and making accessible removable media, including CDs and USB devices, is /media.

/opt:

In Linux, optional or add-on software packages that are not included in the standard system installation are kept in the /opt directory. This maintains the primary system directories tidy and well-structured.

Knowledge of the tree structure helps in visualizing the interconnectedness of directories.

3 Navigating the Directory Structure

Understanding the fundamental structure of any complicated system is essential for navigating it, whether it's a corporate hierarchy, a file system on your computer, or even the architecture of a new city. This entails understanding the connections and organization of various components. Understanding these connections will help you find what you need quickly, comprehend how things operate, and navigate the system with assurance. Some of these commands are as follows:

The pwd Command:

In Linux, the current directory you are working in is shown by the pwd command. For instance, entering pwd will display /home/user/Documents, verifying your location in the file system, if you use cd Documents to navigate to your "Documents" folder. It is helpful when navigating intricate directory hierarchies or in scripts.

The cd Command:

In Linux, the cd command modifies the directory you are currently in. Cd Documents, for example, places you in the "Documents" folder. CD elevates you to a new level. You can access the root directory by using cd /. It is necessary for using the command line to navigate the file system.

The syntax is:

cd [directory_name]: Moving to a specific subdirectory.
cd ..: Moving one level up (to the parent directory).
cd or cd: Returning to the user's home directory.
cd -: Returning to the previous directory.

Absolute vs. Relative Paths:

Absolute paths provide a comprehensive, clear path by indicating a file or directory's precise location from the root (e.g., /home/user/document.txt).

On the other hand, relative paths specify the position in relation to the current directory (such as document.txt or../). While absolute pathways guarantee that you arrive at your destination regardless of where you start, they make navigating easier in a local context.

4 Listing Directory Contents

Linux listing commands, like ls, show the contents of directories. The current location's files and subdirectories are displayed by ls. Options like -l and -a include hidden files and comprehensive information, respectively. To navigate and comprehend the file system, you must know these instructions. Some of the listing commands include:

The basic ls command can utilize the following flags / options:
-l: Long listing, showing details including permissions and ownership
-a: Showing all files and directories, including hidden ones (starting with a.)
-d: Listing only the directory itself, not its contents.
-h: Human-readable file sizes.
-r: Reverse order.
-t: Sort by modification time.

-R: Recursively listing subdirectories.

5 Creating and Removing Directories

Directories, which are similar to folders, are crucial for file organization in Linux. By creating new folders, the mkdir command enables you to organize your data. For instance, creating a new directory called "my_folder" is done with mkdir my_folder.

By moving directories, the mv command modifies where they are located in the file system. To relocate "my_folder" to the /home/user/ directory, for example, use mv my_folder /home/user/.

For effective file management and system organization in Linux, it is essential to become proficient with these commands. They aid in maintaining the accessibility and organization of your data.

The mkdir Command:

In Linux, new directories (folders) are created with the mkdir command. It enables users to arrange the file system's files and subdirectories. Options offer flexible directory creation from the command line by setting permissions or creating parent directories if none already exist. These options are as follows:

Syntax:

mkdir <directory_name>

Similar to folders, directories are essential to Linux file organizing. You can organize your files by using the mkdir command to create new folders. For example, mkdir my_folder is used to create a new directory named "my_folder".

The mv command changes the location of directories in the file system by relocating them. Use mv my_folder /home/user/, for instance, to move "my_folder" to the /home/user/ directory.

Gaining proficiency with these commands is crucial for efficient file management and system structure in Linux. They support you in keeping your data organized and accessible.

To create multiple directories:

Syntax:
mkdir dirA dirK

Creating parent directories:

Syntax:
mkdir -p path/to/new/directory

The rmdir Command:

Removing empty directories.
Syntax:
rmdir ¡directory_name¿.
Its limitation however, is that it only works on empty directories.

The rm Command for Directories:

Removing directories and their contents. It employs the following flags in doing so:

- Using the -r or -R option for recursive removal (use with caution!).
- Using the -f option to force removal (even more caution advised!).
- Combining options: rm -rf ¡directory_name¿ (understand the implications!).

6 Directory Permissions

Linux directory permissions regulate who can access the contents of a directory. Directories have read, write, and execute rights for users, groups, and other entities, much like files. But they work in different ways:'read' lists files, 'write' permits adding or removing files inside the directory, and 'execute' permits using cd to enter the directory. Access to the data in a file is managed by its permissions.

Permissions

Like file permissions, directory permissions are:

- Read (r): Allows listing the contents of the directory.
- Write (w): Allows creating, deleting, and renaming files and subdirectories within the directory.
- Execute (x): Allows entering (changing into) the directory.

To view permissions, use ls -l. The output's initial character—"d" for directory and "-" for file—indicates the nature of the file. Permissions for users, groups, and others are indicated by the next nine characters.

The chmod command in Linux changes directory permissions, limiting who has the ability to see and alter the contents of a directory. You can set permissions numerically (chmod 777) or symbolically (chmod u+rwx). This is essential for file management and system security since it establishes read, write, and execute access for the owner, group, and others.

And to change the directory permissions, we use the chmod command. In so doing, we can also opt for either.

- Symbolic mode (e.g., chmod u+w <directory_name>), or
- Numeric mode (e.g., chmod 755 <directory_name>).

7 Finding Directories

Linux's find command is an effective way to find files and directories inside a file system. Name, kind, size, and permissions are just a few of the parameters it uses for its searches.

To find a file called "myfile.txt," for instance, use find / -name "myfile.txt" to search the entire system. Directories within the current directory can be found with find. -type d. Files greater than 10MB are found in the /home directory using find /home -size +10M. Additionally, find can do actions on files it has located, such as removing them. File management and system administration require it.

The find Command: A powerful tool for locating files and directories based on various criteria. Basic syntax: find <path> -name <pattern> Finding only directories: find <path> -type d -name <pattern> Other useful options for directories could be: -empty, -perm, -user, -group, -mtime.

8 Good Practices

Recognizing the Current Working Directory: Always know where you are in the file system wherever you go. Effective Use of Relative Paths: For project portability and convenience. Exercise Caution When Using rm -rf: Verify your target once again before using this command. Logically arranging your directories will improve file management. For ease of recognition and clarity, choose meaningful directory names.

9 Conclusion

Linux manages files and directories via commands. Pwd displays the current location, cd switches directories, and ls lists the contents of a directory. Rm deletes files and directories, while mkdir creates them. mv transfers them, and cp copies them.

Chmod modifies file permissions (rwx), which govern user, group, and other access. discover uses criteria to find files.

Paths might be relative (from the current directory) or absolute (from the root). These ideas and commands are essential for using the command line to navigate and manage a Linux system.

In Linux, organizing and navigating directories effectively is essential. To see the current directory, use pwd; to switch between them, use cd. While mkdir generates new directories and rmdir eliminates empty ones, ls lists the contents of folders. For more complex scenarios, rm -r recursively deletes directories and their contents.

Relative pathways are relative to the present position, but absolute paths (beginning with /) offer a direct path. To efficiently organize and access files, it is essential to comprehend these commands and path types.

List of References

1. Ramey, C. (1998). Bash reference manual. Network Theory Limited, 15.

2. Blum, R. (2008). Linux command line and shell scripting bible (Vol. 481). John Wiley Sons.