Linux and Shell Scripting

3: Basics of Linux

Introduction

This chapter will introduce you to the fundamentals of Linux, covering essential concepts, commands, and the file system structure. By the end of this chapter, you will have a solid understanding of how to navigate and manage a Linux system.

3.1 Linux File System Structure

3.1.1 Directory Hierarchy

Linux uses a hierarchical file system structure, starting with the root directory denoted by **/**. Here are some key directories:

- **/**: Root directory, the top-level directory in the file system.
- **/bin**: Essential command binaries (programs) needed for booting and repairing the system.
- **/boot**: Files needed for the boot process, such as the kernel and boot loader files.
- **/dev**: Device files representing hardware components.
- **/etc**: System configuration files.
- **/home**: Home directories for users.
- **/lib**: Essential shared libraries and kernel modules.
- **/media**: Mount points for removable media such as USB drives.
- **/mnt**: Temporarily mounted file systems.
- **/opt**: Optional software packages.
- **/proc**: Virtual file system providing process and kernel information.
- **/root**: Home directory for the root user.
- **/sbin**: System binaries for root or superuser.
- **/tmp**: Temporary files.
- **/usr**: User programs and data.
- **/var**: Variable data such as logs and spool files.

3.1.2 Path Types

- **Absolute Path**: Specifies the complete path from the root directory. For example, **/home/user/documents**.
- **Relative Path**: Specifies the path relative to the current directory. For example, **documents** (if currently in **/home/user**).

3.2 Essential Linux Commands

3.2.1 Navigation Commands

- **pwd**: Print Working Directory. Displays the current directory.

bash

code

pwd

- **ls**: List directory contents.

bash

code

ls

Common options:

    - **-l**: Long listing format.
    - **-a**: Include hidden files.

- **cd**: Change Directory. Navigates to a different directory.

bash

code

cd /path/to/directory

3.2.2 File and Directory Management

- **mkdir**: Make Directory. Creates a new directory.

bash

code

mkdir new_directory

- **rmdir**: Remove Directory. Deletes an empty directory.

bash

Copy code

rmdir empty_directory

- **rm**: Remove files or directories.

bash

Copy code

rm file_name rm -r directory_name # Recursively remove a directory and its contents

- **cp**: Copy files or directories.

bash

Copy code

cp source_file destination_file cp -r source_directory destination_directory # Recursively copy a directory

- **mv**: Move or rename files or directories.

bash

Copy code

mv old_name new_name mv file_name /new/path/

3.2.3 Viewing and Editing Files

- **cat**: Concatenate and display file contents.

bash

Copy code

cat file_name

- **less**: View file contents one page at a time.

bash

Copy code

less file_name

- **head**: Display the first part of a file.

bash

Copy code

head file_name

Common option:

    - **-n**: Specify the number of lines to display.

- **tail**: Display the last part of a file.

bash

Copy code

tail file_name

Common option:

    - **-f**: Follow the file as it grows, useful for log files.

- **nano**: A simple text editor.

bash

Copy code

nano file_name

- **vim**: A powerful text editor (advanced users).

bash

Copy code

vim file_name

3.2.4 File Permissions and Ownership

- **chmod**: Change file mode (permissions).

bash

Copy code

chmod 755 file_name

Common permission values:

    - **r**: Read (4)
    - **w**: Write (2)
    - **x**: Execute (1)
- **chown**: Change file owner and group.

bash

Copy code

chown user:group file_name

3.2.5 System Information

- **uname**: Display system information.

bash

Copy code

uname -a

- **df**: Display disk space usage.

bash

Copy code

df -h

- **du**: Display disk usage of files and directories.

bash

Copy code

du -sh directory_name

- **top**: Display real-time system processes.

bash

Copy code

top

3.3 Linux Shells

3.3.1 Introduction to Shells

A shell is a command-line interpreter that provides a user interface for the Unix/Linux operating system. Common shells include:

- **Bash (Bourne Again Shell)**: The default shell for many Linux distributions.
- **Zsh (Z Shell)**: Known for its powerful features and customization.
- **Fish (Friendly Interactive Shell)**: User-friendly and interactive.

3.3.2 Basic Shell Commands

- **echo**: Print a message to the terminal.

bash

Copy code

echo "Hello, World!"

- **which**: Locate a command.

bash

Copy code

which ls

- **alias**: Create an alias for a command.

bash

Copy code

alias ll='ls -l'

3.4 Navigating the Linux File System

3.4.1 Using Wildcards

Wildcards are symbols that represent other characters in file names. Common wildcards include:

- *****: Matches any number of characters.

bash

Copy code

ls *.txt

- **?**: Matches a single character.

bash

Copy code

ls file?.txt

- **[]**: Matches any one of the characters inside the brackets.

bash

Copy code

ls file[1-3].txt

3.4.2 Redirection and Piping

- **Redirection**: Redirect the output of a command to a file.

bash

Copy code

command > output_file # Overwrite the file command >> output_file # Append to the file

- **Piping**: Pass the output of one command as input to another command.

bash

Copy code

command1 | command2

Example:

bash

Copy code

ls -l | grep "Jan"

3.5 Summary

In this chapter, we covered the basics of Linux, including the file system structure, essential commands, shells, and navigation techniques. You should now be comfortable performing basic tasks in the Linux environment, such as navigating directories, managing files, and viewing system information.