

Linux and Shell Scripting

5: Shells, Commands, and Navigation

Introduction

In this chapter, we will delve into the fundamental aspects of using Linux shells, executing commands, and navigating the file system. Understanding these basics is crucial for efficient system administration and effective use of a Linux environment.

5.1 Shells

5.1.1 What is a Shell?

A shell is a command-line interpreter that provides a user interface for the Unix/Linux operating system. It allows users to execute commands, run scripts, and interact with the system.

5.1.2 Commonly Used Shells

- **Bash (Bourne Again Shell):** The default shell on most Linux distributions.
- **Zsh (Z Shell):** Known for its extended features and user-friendliness.
- **Fish (Friendly Interactive Shell):** Focuses on user-friendly and interactive features.
- **Sh (Bourne Shell):** An older shell that served as a basis for many others.

5.1.3 Switching Between Shells

To change your shell temporarily, type the name of the shell at the command prompt. For example:

```
bash
```

Copy code

```
zsh
```

To change your default shell permanently:

```
bash
```

Copy code

```
chsh -s /bin/zsh
```

Then, log out and log back in for the change to take effect.

5.2 Commands

5.2.1 Basic Command Syntax

A command generally follows the syntax:

```
bash
```

Copy code

```
command [options] [arguments]
```

For example:

bash

Copy code

```
ls -l /home
```

- **ls** is the command.
- **-l** is an option that tells **ls** to use long listing format.
- **/home** is an argument specifying the directory to list.

5.2.2 Common Commands

File and Directory Operations

- **Listing Files:**

bash

Copy code

```
ls ls -l ls -a
```

- **Changing Directory:**

bash

Copy code

```
cd /path/to/directory
```

- **Present Working Directory:**

bash

Copy code

```
pwd
```

- **Creating Directories:**

bash

Copy code

```
mkdir directory_name
```

- **Removing Files and Directories:**

bash

Copy code

```
rm file_name rm -r directory_name
```

- **Copying Files:**

bash

Copy code

cp source_file destination

- **Moving/Renaming Files:**

bash

Copy code

mv source_file destination

Viewing and Editing Files

- **Displaying File Contents:**

bash

Copy code

cat file_name less file_name more file_name

- **Editing Files:**

- **Nano:**

bash

Copy code

nano file_name

- **Vim:**

bash

Copy code

vim file_name

System Information

- **Disk Usage:**

bash

Copy code

df -h du -sh directory_name

- **Memory Usage:**

bash

Copy code

free -h

- **System Uptime:**

bash

Copy code

uptime

- **Current Users:**

bash

Copy code

who

5.2.3 Command History and Aliases

Command History

- **View History:**

bash

Copy code

history

- **Repeat Last Command:**

bash

Copy code

!!

- **Run a Specific Command from History:**

bash

Copy code

!n # where n is the command number from history

Aliases

Aliases allow you to create shortcuts for commands.

- **Creating an Alias:**

bash

Copy code

alias ll='ls -l'

- **Removing an Alias:**

bash

Copy code

unalias ll

- **Making Aliases Permanent:** Add them to your `~/.bashrc` or `~/.zshrc` file.

5.3 Navigation

5.3.1 Navigating the File System

The Linux file system is hierarchical, with the root directory (/) at the top. Important directories include:

- **/bin** - Essential command binaries
- **/etc** - Configuration files
- **/home** - Home directories for users
- **/var** - Variable data files
- **/usr** - User utilities and applications

5.3.2 Pathnames

- **Absolute Path:** A complete path from the root directory, e.g., **/home/user/docs**.
- **Relative Path:** A path relative to the current directory, e.g., **docs** if you're in **/home/user**.

5.3.3 Useful Navigation Commands

- **Change to Home Directory:**

bash

Copy code

```
cd ~
```

- **Go Up One Directory:**

bash

Copy code

```
cd ..
```

- **Return to Previous Directory:**

bash

Copy code

```
cd -
```

5.4 Summary

In this chapter, we covered the basics of Linux shells, executing commands, and navigating the file system. Understanding these concepts is essential for performing everyday tasks and managing a Linux system effectively. With these skills, you can interact with the system more efficiently and automate routine tasks.