Introduction

Anurag S Chiplunkar

Technical Trainer

Unix vs. Linux Comparison

- Unix and Linux are popular operating systems (OS) with similar purposes.
- Unix is used for enterprise-level servers and workstations, personal computers (far less).
- Large organisation tend to use Unix because of its high-performance, multitasking oriented design.
- UNIX supports multiple users on the same machine, whether they are logging into it directly or accessing it remotely.
- Like Unix, Linux is a popular platform among organizations that need to host data, applications, or services on high-volume servers. Linux is also a popular operating system choice for personal computers.
- Linux OS versions are the primary option for users worldwide who prefer to use open-source software.
- Linux developers aimed to keep its commands as similar as possible to Unix commands by adhering to the portable operating system interface (POSIX) standards set by the Institute of Electrical and Electronics Engineers. Many core commands are identical in syntax and functionality, including tar and lpr. Small differences still exist, not only between Unix and Linux but among different Unix variants as well.

What is a Shell?

A *shell* is a program that takes commands typed by the user and calls the operating system to run those commands. The shell interprets your commands. For example, you may use the shell to enter a command to list the files in a directory, such as `ls`, or a command to copy a file, such as `cp`.

Shells were designed long before graphical interfaces existed. As graphical environments mature, most users explicitly run shells less and less for their daily work. But a shell can automate some very complex sequences of commands. In addition, most Linux systems are designed to be updated from typed-in commands—that is, from a shell. Furthermore, whether you know it or not, a shell often powers many of the graphical commands users run. Learning the shell can help you better understand your computer.

What Kind of Shells Are There?

Since there is no monopoly on shells, you are free to run any shell you desire. That's all well and good, but choosing a shell without knowing the alternatives isn't very helpful.

1. ***The Bourne Shell***
2. ***The C Shell***
3. ***The Korn Shell***
4. ***Bash, the Bourne Again Shell***

1. The Bourne Shell

- The original Unix shell is known as sh, short for *shell* or the Bourne shell, named for

Steven Bourne, the creator of sh. As shells go, sh remains fairly primitive, but it was quite advanced for the 1970.

- The Bourne shell has been considered a standard part of Unix for decades. Thus, sh should be available on almost all systems that support Unix or Unix-like commands, including Linux, Unix, and Mac OS X systems.
- The basic Bourne shell supports only the most limited command-line editing.
- Wide use today, especially for system administration scripts.

1. The C Shell

1. Designed by Bill Joy at the University of California at Berkeley.
2. Similarity to C programming.
3. Mostly used by C programmers.
4. Added some neat features to Bourne Shell
    a) ability to recall previous commands.
5. Quickly became default shell.

For many years C shell and Bourne shell were the only games in town. However, Bourne shell scripts will not run on the C shell because of differences in syntax.

1. The Korn Shell

1. It offers the same kind of enhancements offered by the C shell, with one important difference.
2. Korn is backward compatible with the older Bourne shell syntax.
3. It followed the older Bourne syntax, unlike C shell having its own syntax.
4. This means that the Korn shell can run most Bourne shell scripts. The C shell cannot.

1. The Bash Shell, the Bourne Again Shell

1. The Korn shell was king of the shells on proprietary Unix, but that now pales in comparison to the installed base of Linux. Linux, a Unix work-alike operating system, grew faster than anyone predicted, and Linux users wanted an advanced shell with features like that of the Korn shell. But Linux users needed a shell that was freely available under an open-source license. This led to the development of bash.
2. Where the Korn shell was a form of answer to the success of the C shell, the bash shell can be considered an answer to the Korn shell.
3. Bourne Shell filled in the gap left by the lack of Korn Shell.

Determining Which Shell You Are Running

Which Shell Am I Running?

If you don't already know what kind of shell it is, try the following command:

$ **echo $SHELL**

/bin/bash

What are Shell Scripts?

1. A shell script is a text file that contains one or more commands.
2. In a shell script, the shell assumes each line of the text file holds a separate command. These commands appear for the most part as if you had typed them in at a shell window.

Advantages of using Shell Scripts

1. Automation of Tasks: Shell scripts can automate repetitive tasks, reducing the need for manual intervention. This saves time and minimizes the risk of human error.
2. Ease of Use: Shell scripts are straightforward to write and execute. They do not require extensive programming knowledge, making them accessible to users with basic scripting skills.
3. Efficiency: Shell scripts can execute commands and processes much faster than doing the same tasks manually. They can chain commands together and handle complex workflows efficiently.
4. Portability: Shell scripts are usually portable across different Unix-like operating systems (such as Linux, macOS, and BSD). This allows scripts written on one system to run on another with minimal modifications.
5. Integration: Shell scripts can easily integrate with other software and utilities. They can call external programs, handle command-line arguments, and use output from other commands within the script.
6. Customization: Users can tailor shell scripts to fit their specific needs, allowing for highly customized workflows and solutions that might not be possible with standard software.

Introduction

This topic ends here.

Hope this **empowering**, will shape your tomorrow. **Best of Luck** and **Thank You** for your attention.

Anurag S. Chiplunkar
Subject Matter Expert