Chapter 12

Deploying Bash Scripts

The Basics of Shell Scripting

- One exciting feature of the Linux command line is that you can enter multiple commands on the same command line and Linux will process them all. Just place a semicolon between each command you enter: $ date ; whoami

- Another building block of shell scripting is the ability to store command output: $ date > today.txt && whoami >> today.txt

- Piping Data: $ ls | sort

Shell Script

- Format: .sh ;
- Running Shell Script: $ ./test1.sh
- Permissions: $ chmod u+x test1.sh
- Environment Variables

  $ which bash ; echo $PATH ; $ echo Home: $HOME ; $ set

- User Variables

  $ days=10

  $ guest=John

  $ echo $guest checked in $days days ago

Command-Line Arguments

```
$ cat test.sh
#!/bin/bash
# Testing command line arguments
echo $1 checked in $2 days ago
$ chmod u+x test.sh
$ ./test.sh Ali 3
Ali checked in 3 days ago
$ ./test.sh Hasan 5
Hasan checked in 5 days ago
```

Performing Math

The bc calculator is a tool in Linux that can perform floating-point arithmetic:

```
$ bc
...
12 * 5.4
64.8
3.156 * (3 + 5)
25.248
quit
```

To include mathematical expressions in your shell scripts, you use a special format. This format places the equation within the $[] characters:

```
result=$[ 25 * 5 ]
```

Logic Statements

- The if Statement : *Listing 25.11*
- The case Statement : *Listing 25.12*
- The for Loop : *Listing 25.13*
- The while Loop : *Listing 25.14*
- Text Manipulation :

  Shell scripts allow you to process large quantities of data files line by line, searching for specific data, or even replacing specific data with just a few simple commands.

- **EXERCISE 25.1**: Writing a Bash Script to View the Password Information for System Users