

How to Use While Loop in Linux Shell Scripting

The "while loop" is a fundamental control flow statement that allows code to be executed repeatedly based on a given Boolean condition. In the context of Linux, while loops are particularly useful in shell scripting for automating repetitive tasks, managing system processes, and handling file operations. Understanding how to implement while loops in Linux can significantly enhance your scripting capabilities and system management efficiency.

In Linux, while loops are typically used in bash scripts. These scripts can automate tasks such as monitoring system resources, processing files, and performing routine maintenance. Mastering while loops in bash can help you create more dynamic and responsive scripts.

Examples:

1. Basic While Loop Example

```
#!/bin/bash

counter=1

while [ $counter -le 5 ]
do
    echo "Counter: $counter"
    ((counter++))
done
```

In this example, the script initializes a variable counter to 1. The while loop continues to execute as long as the value of counter is less than or equal to 5. Each iteration prints the current value of counter and then increments it by 1.

2. While Loop with User Input

```
#!/bin/bash

read -p "Enter a number (0 to exit): " number

while [ $number -ne 0 ]
do
    echo "You entered: $number"
    read -p "Enter another number (0 to exit): " number
done
```

```
echo "Exited the loop."
```

This script prompts the user to enter a number. The while loop continues to execute as long as the entered number is not 0. Each iteration prints the entered number and prompts the user for another input.

3. Monitoring System Resources

```
#!/bin/bash

while true
do
    echo "CPU Usage:"
    top -bn1 | grep "Cpu(s)" | \
        sed "s/.*, *\[0-9.\]*\)%* id.*\/\1/" | \
        awk '{print 100 - $1"%"}'
    sleep 5
done
```

This script continuously monitors CPU usage. The while loop runs indefinitely (while true), and every 5 seconds, it prints the current CPU usage. The top command is used to get the CPU statistics, which are then processed using grep, sed, and awk.