Collecting User Input and Code Logic

Andrew Mallett
LINUX AUTHOR AND CONSULTANT

@theurbanpenguin www.theurbanpenguin.com

Module Overview

Using variables

Built-in variables

Reading user input

Using conditional statements in BASH

Building the Project Script

As we work our way through the course we will build an application that Linux Operators can use to manage users accounts. We start by creating users

BASH Variables

Setting variables: user_name=bob

Reading variables: echo "$user_name"

Delimiting variables: echo "${user_name}s"

Built-in Variables

Positional parameters: $1, $2 ... ${10}.
$0 is not positional but represents the script name

Number of positional parameters: $#

List of all parameters: $*

Demo

We will now build a new script that will be developed for User Accounts. For the moment we will investigate using positional parameters to gain user input.

```
#!/bin/bash

user_password=${2:-Password1}

echo "$user_password"
```

Providing Default Values

Here the 2nd argument should be the password, if it is not supplied we can set a default value

Demo

Allowing default values for the password is an easy option to add to the script.

```
#!/bin/bash
read -p "Enter a username: " user_name
read -sp "Enter a password: " user_password
echo -e "\n$user_name $user_password"
```

Prompting for User Input

The BASH built-in command read can be used to prompt for user input populating the variable name we supply or the variable REPLY if none is stated. The option -p is for the prompt message and -s to silence the screen input

Demo

We will now test prompting for user input.

```
#!/bin/bash
if [ $# -eq 0 ] ; then
    read -p "Enter a username: " user_name
else
  user_name="$1"
fi
```

Testing Numerical Values

The operators that we can use to test numerical values include -eq, -lt and -gt

```
if [ "$user_password" != "$user_password_check" ] ; then
    echo "$0: Passwords do not match"
    exit 1
fi
```

Testing Strings Values

We may want to have the user enter a password twice for verification.
A simple test could look like this

Demo

Implementing logic within the script allowing command line input or prompts.

Summary

When setting variables do not leave whitespace around the = symbol

When using variables always double-quote them as we do not know what characters they may contain

Provide default values where appropriate

BASH built-in read to prompt for input

The keyword if is used to start an IF conditional statement and fi to close it.

Next up:
Using Functions and Loops in Scripts