Shell Script Programs with If – Then – Else Statements

June 3, 2010

Adding Comments

When we want to place a comment in a Shell Script, we type the pound sign (#) in front of the phrase or sentence. The first comment usually defines the program to anyone opening and reading the executable file. We can also write copyright information in this area. Comments for each line can be written at the end of the coded expression as shown in the example.

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
                                                                # read day
echo -n "What is the day of the week? Full name please."
read day
                                                                # user prompt
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$day" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

Output the Program's Purpose

We can use the Echo command to output what the program will do. When we have more than one Shell Script in a folder, we can first display on the monitor what the software will do on our computer before answering the user's prompts. This is good programming etiquette. We use the echo command to output data to the computer user.

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
echo -n "What is the day of the week? Full name please."
                                                                # read day
                                                                # user prompt
read day
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$day" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

Prompting the User

We can also employ the Echo command to ask the user for input. In our example, we will ask the computer operator, "what is the day of the week using the full name?" The next line in the code after the user prompt is the read command where we set the user's input to a variable. In our program, we set the day of the week to the variable day.

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
echo -n "What is the day of the week? Full name please."
                                                                # read day
                                                                # user prompt
read day
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$day" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

Obtaining the System Date

We can also employ the 'date' command to get the system date off of the computer and we can add the plus percent sign and capital A to extract the full system day such as Sunday, Monday or Tuesday. We assign the data to the variable sdate.

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
echo -n "What is the day of the week? Full name please."
                                                                # read day
read day
                                                                # user prompt
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$dav" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

Comparing Days with the If Statement

We can compare two numbers or strings using the If statement. In this case, we are using the equal operator, but we could use greater than (>), less than (<), or not equal (!=) in other programs. An **if** statement begins with the logical comparison which will result in a true or false when the expression is read. The **then** statement is executed if the logical test is affirmative. In this case, if we typed Thursday and the system date is Thursday, the computer program will echo "the system date is correct"

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
echo -n "What is the day of the week? Full name please."
                                                                # read day
read day
                                                                # user prompt
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$day" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

The If Statement (continued)

The **else** statement is executed if the logical test is negative. In this example, if we typed Thursday and the system date is Friday, the computer program will echo "check the system date for error" We conclude the if —then —else statement with fi.

```
# dotw is a program that check the day of the week against the system day
echo "This program checks the day of the week"
                                                                # program definition
echo -n "What is the day of the week? Full name please."
                                                                # read day
read day
                                                                # user prompt
sdate=$(date +"%A")
                                                                # retrieve the system day
if [ "$day" = "$sdate" ]
then
echo "System date is correct"
else
echo "Check system date for error"
fi
                                                                # end of program
```

Program another Shell Script

When we are done in the text editor, we can save the file as **dotw.exe** or just **dotw**. If we save the file as **dotw**, we can convert it to an executable file by typing in the Bash command **chmod ugo+x dotw**. To run the program type **sh dotw**. When prompted for the day of the week, we can type Thursday and the answer will return as the system date is correct if your system date matches the day you type.

Write another program. Ask the computer user a question and utilize the if-then-else statement to compare the answer.