Shell Script Programs with a While Loop

June 7, 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.

```
# mk_file is a program that makes empty  files
echo  “This program makes empty text files “                    # program definition
echo -n “How many files do you wish to make?  “                 # choice
read  number                                                    # user prompt
Echo –n “What is the name of the file?  “                       # choice for file name
read filename                                                   # input file name
let counter=0                                                   # set the counter to 0
while [  $counter –lt  $number ]
do
  let counter=counter+1
  touch $filename$counter.txt
done                                                            # 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.

```
# mk_file is a program that makes empty  files
echo  "This program makes empty text files "                    # program definition
echo -n "How many files do you wish to make?  "                 # choice
read  number                                                    # user prompt
Echo –n "What is the name of the file?  "                       # choice for file name
read filename                                                   # input file name
let counter=0                                                   # set the counter to 0
while [  $counter –lt  $number ]
do
  let counter=counter+1
  touch $filename$counter.txt
done                                                            # 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, “how many files they wish to create?” 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 number of files to the variable number.

```
# mk_file is a program that makes empty  files
echo  “This program makes empty text files “                    # program definition
echo -n “How many files do you wish to make?  “                 # choice
read  number                                                    # user prompt
echo –n “What is the name of the file?  “                       # choice for file name
read filename                                                   # input file name
let counter=0                                                   # set the counter to 0
while [  $counter –lt  $number ]
do
  let counter=counter+1
  touch $filename$counter.txt
done                                                            # end of program
```

In the second question, we will ask the computer operator, “what is the name of the file?” 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 name of the file to the variable filename.

The While Loop

The While loop begins with us setting a counter variable to zero. In the While loop, we use a test to determine how many cycles we will make in program. In this while loop, we ask if the counter is less than the number of files. If we want to make 3 files, the first test is 0 < 3, which is true, so the loop is executed.

```
# mk_file is a program that makes empty  files
echo  “This program makes empty text files “                    # program definition
echo -n “How many files do you wish to make?  “                 # choice
read  number                                                    # user prompt
Echo –n “What is the name of the file?  “                       # choice for file name
read filename                                                   # input file name
let counter=0                                                   # set the counter to 0
while [  $counter –lt  $number ]
do
  let counter=counter+1
  touch $filename$counter.txt
done                                                            # end of program
```

The first expression in the while loop is to add one to the counter. Then we make a file called filename1.txt. After that expression, the while statement will test the condition again before looping.

The Condition Test

The second test will have 1 < 3, so the program will continue inside the loop. The counter will increase by one and the new file will be filename2.txt. The third test will have 2 < 3, so the program will continue inside the loop. The counter will increase by one and the new file will be filename3.txt.

```
# mk_file is a program that makes empty  files
echo  “This program makes empty text files “                    # program definition
echo -n “How many files do you wish to make?  “                 # choice
read  number                                                    # user prompt
Echo –n “What is the name of the file?  “                       # choice for file name
read filename                                                   # input file name
let counter=0                                                   # set the counter to 0
while [  $counter –lt  $number ]
do
  let counter=counter+1
  touch $filename$counter.txt
done                                                            # end of program
```

The next test will have 3 < 3, which will return as false. When the while loop condition has a negative response, the program bypass the expression inside the loop which is concluded with the done command and continues. In this program, we are finished.

Other Condition Operators

We used the –lt for less than in the condition test of the while loop. We can see in the table other operators for different scenarios. We place the logical operator between the two numbers. If the condition test is true, the expressions in the while loop are read. If the condition test results are false, then the expressions inside the while loop are bypasses.

Operator	Definition
-eq	Equal to
-ne	Not equal to
-gt	Greater than
-lt	Less than
-ge	Greater than or equal to
-le	Less than or equal to

```
while [  $counter  –lt  $number ]
do
  actions
  actions
done
```

Writing Shell Scripts with Loops

When we are done in the text editor, we can save the file as **mk_file.exe** or just **mk_file**. If we save the file as **mk_file**, we can convert it to an executable file by typing in the Bash command **chmod ugo+x mk_file**. To run the program type **sh mk_file.** When prompted for our choice, we can type D or M and our answer will give us one of the lists.

Write another program. Ask the computer user a question and utilize the while loop.