Chapter 9

Shell Scripting: While Loops

In this chapter, you will learn the following to World Class standards:

1. **Adding Comments**
2. **Defining the Program**
3. **Prompting the User**
4. **Computing the Formula**
5. **Displaying the Answer**
6. **Creating an Executable File**

A While Loop Program

A While loop allows us to repeat actions in our program while a designated condition remains constant. In our example program, we used a While loop with a counter to create a certain number of new files as chosen by the 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
```

Figure 9.1 – The MK_File Program

Adding Comments

Before beginning our script, we'll add a comment to define the function of the program. We will use comments to describe the purpose of certain lines of code as well. Remember that comments, designated with the pound sign #, can be used to include any information inside of the script.

```
# 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
```

Figure 9.2 – Adding Comments to Define the Program and Lines of Code

Defining the Program

The second line of code displays the purpose of the program for the user. This is considered proper programming etiquette, and is especially useful if there are more than one executable shell files in a single directory. Use the **echo** command and enter the text in quotation marks as

shown below.

```
# 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
```

Figure 9.3 – Outputting the Program Definition

Prompting the User

Next we will prompt the user to supply the two pieces of data necessary for our program. First we will ask the user how many files he/she wants the program to generate, storing the number to the variable **number** using the **read** command. Next, we ask for the name of the new files, once again using the **read** command and making the variable **filename**.

```
# 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
```

Figure 9.4 – Prompting the User

The While Loop

Now we are ready to set up the While loop. First we will set a counter by typing **let counter=0**. The counter is the element upon which our loop will be dependent; when the counter reaches a certain value then the loop will be complete and the program will continue. Next we need to create our While statement. Type **while**, and then inside brackets type **$counter –lt $number**. This program function means that as long as the **counter** variable value is less than the **number** variable value, the program will execute the While loop.

```
# 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
```

Figure 9.5 – The While Loop

The second part of the While loop is the action that the program will take as long as the while condition is still in effect. For this program we want to create new text files, but we must also keep track of how many we have created, so the action of the While loop will have two parts. The first part will increase the counter by one; type **let counter=$counter+1**. This will increase the counter by one every time the while Loop is executed. Next type **touch $filename$counter.txt**. This will create a new text file with the filename designated by the user as well as a number from the counter. Type **done** afterwards to end the program.

There are a number of operators we can use to define the condition in our while loop. We used **–lt** in this program to create a new file as long as the counter is **less than** the number of files the user wants to create. Once the While loop has been executed enough times to make the counter equal to the number files input by the user, the program will move on to the next command. While loop operators work much in the same way that conditional statements operators do.

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

Making the Shell Script an Executable File

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 by the program, we can type the number of files we want to make and the name of the file, and the program will create those files.

*** World Class CAD Challenge 9-1 * - Write a Script that displays two message boxes, the first will contain the script name, copyright date and author. The second message will display information from the computer.**

Continue this drill four times using some other messages, each time completing the VBScript in less than 30 minutes to maintain your World Class ranking.