Shell Scripting - Module 5

Looping statements

There are 3 looping statements which can be used in bash programming.

- **While statement**
- **For statement**
- **Until statement**

  To alter the flow of loop statement, two commands are used they are

  - **break**
  - **continue**

While statement

Here command is evaluated and based on the result loop will executed, if command raise to false then loop will be terminated.

Syntax :

```
while command
do
   Statement to be executed
done
```

For statement

- F**or loop operate on lists of items.**
- **It repeats a set of commands for every item in a list.**

Syntax :

```
for var in word1 word2 ...wordn
do
   Statement to be executed
done
```

Until statement

- **The until loop is executed as many as times the condition/command evaluates to false.**
- **The loop terminates when the condition/command becomes true.**

Syntax :

```
until command
do
   Statement to be executed until command is true
done
```

Conditionals

There are total 5 conditional statements which can be used in bash programming

if statement

This block will process if specified condition is true.

Syntax :

```
if [ expression ]
then
   statement
fi
```

if-else statement

If specified condition is not true in if part then else part will be execute.

Syntax :

```
if [ expression ]
then
   statement1
else
   statement2
fi
```

if..elif..else..fi statement (Else If)

- **To use multiple conditions in one if-else block, then elif keyword is used in shell.**
- **If expression1 is true then it executes statement 1 and 2, and this process continues.**
- **If none of the condition is true then it processes else part.**

Syntax :

```
if [ expression1 ]
then
   statement1
   statement2
   .
   .
elif [ expression2 ]
then
   statement3
   statement4
   .
   .
else
```

```
   statement5
fi
```

if..then..else..if..then..fi..fi..(Nested if)

- **Nested if-else block can be used when, one condition is satisfies then it again checks another condition.**
- **In the syntax, if expression1 is false then it processes else part, and again expression2 will be check.**

Syntax :

```
if [ expression1 ]
then
   statement1
   statement2
   .
else
   if [ expression2 ]
   then
      statement3
      .
   fi
fi
```

switch statement

- **case statement works as a switch statement if specified value match with the pattern then it will execute a block of that particular pattern**
- **When a match is found all of the associated statements until the double semicolon (;;) is executed.**
- **A case will be terminated when the last command is executed.**
- **If there is no match, the exit status of the case is zero.**

Syntax :

```
case  in
   Pattern 1) Statement 1;;
   Pattern n) Statement n;;
esac
```