Flow control in Shell

While writing a Shell script, there may be a situation when you need to follow an external path 2 provided. So you need to create usage of conditional commands that allow your program to make accurate decisions and perform the right actions.

While writing a Shell script, there may be a situation when you need to follow an external path 2 provided. So you need to create usage of conditional commands that allow your program to make accurate decisions and perform the right actions.

Unix Shell supports conditional commands that are used to perform various actions based on various conditions. Here we will explain two flow control commands:

If . else statement

The **case . esac** command

If . else command in Unix / Linux

This command is a useful flow control command, used to select an option from a given set of options.

Unix Shell supports the following **if . else** commands:

If . fi command

If . else . fi command

If . elif . else . fi command

Most **if statements** check relationships by using the relational operators mentioned in the previous chapter.

Case . esac command in Unix / Linux

You can use multiple **if . elif** commands to make a branch with multiple choices. However, this is not the best solution, especially when all branches depend on the value of a single variable.

Unix shell supports **case . esac** command which correctly resolves this situation, and it performs in a more efficient way than iterating using multiple **if . elif** commands.

There is only one sample of the **case . esac** command that is listed here:

The **case . esac** command

The **case . esac** command of Unix Shell is similar to the switch command . the case that we have in C or C ++ programs and PERL etc.

According to Tutorialspoint

Previous article: Operator Korn Shell

Next article: Loop in Unix / Linux