Sistema Operacional

Unidade 11.2 – Shell Script: estruturas condicionais

QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

SHELL SCRIPT COM IF

Na linguagem *Shell Script* podemos utilizar estruturas condicionais, como por exemplo, o **if**. Com ele podemos realizar testes lógicos através dos operadores relacionais.

> *Um teste condicional é composto de uma ou mais condições, sendo que uma condição é composta de no mínimo dois argumentos e um operador relacional.*

OPERADORES RELACIONAIS

Em scripts é comum utilizar comparações entre variáveis, sendo caracteres e/ou números. Para comparar utilizamos condições, onde cada condição é composta de no mínimo dois argumentos e um operador relacional. Todo teste realizado através de uma condição pode retornar duas possibilidades, são elas: **verdadeiro** ou **falso**. Em inglês, ***true*** ou ***false***.

Veja abaixo os operadores relacionais:

== igual

> maior

< menor

>= maior ou igual

<= menor ou igual

!= diferente

Por exemplo: **$nome == "thiago".** Essa condição testa se o conteúdo da variável "nome" é igual a "thiago".

COMPARAÇÕES EM SCRIPTS

Para realizar comparações entre números geralmente utilizamos letras ao invés de operadores relacionais comuns, como o sinal de maior (>) ou menor(<) por exemplo. Claro que é possível criar condições com operadores comuns, porém, os operadores relacionais com letras são mais utilizados.

Por exemplo: vamos comparar se um número é maior que 0 (zero) utilizando o sinal de maior (>) e depois utilizando os operadores através de letras, no caso o "-gt".

Comparando se o conteúdo da variável $numero é maior do que 0 (zero) através do sinal de maior (>).

```
$numero \> 0
```

> *Obs: Para testar números com os sinais, temos que utilizar um caractere de escape, no caso utilizamos o caractere contra barra "\".*

Comparando se o conteúdo da variável $numero é maior do que 0 (zero) utilizando o "-gt".

```
$numero -gt 0
```

> *Obs: Com a utilização do "-gt" não precisamos utilizar o caractere de escape contra barra.*

MAIS OPERADORES

-eq igual

-ne não igual

-gt maior

-ge maior ou igual

-lt menor

-le menor ou igual

-o ou

-d se o arquivo for diretório

-e se existir o arquivo

-z se o arquivo estiver vazio

-f se o arquivo contem algum texto

-o se o usuário for o dono do arquivo

-r se o arquivo pode ser lido

-w se o arquivo pode ser alterado

-x se o arquivo pode ser executado

ESTRUTURA CONDICIONAL IF

A estrutura "***if***" permite testar o resultado das comparações. Como toda comparação retorna dois possíveis valores (***true*** ou ***false***) precisamos avaliar qual foi o resultado e efetuar uma ação a partir do resultado obtido.

Para testar o resultado das comparações podemos utilizar somente o "***if***". O ***if*** é um comando que desvia o fluxo da operação para um determinado conjunto de comandos, conforme o resultado da condição.

Sintaxe básica do if simples

O ***if*** simples testa somente se a condição for verdadeira, ou seja, se ela for verdadeira ele executa a ação previamente programada.

*Obs: Todo **if** no Shell Script deve ser encerrado com a palavra "fi", que no caso significa "if" ao contrário.*

```
if [ condição ]; then
      #ação que será tomada caso a condição seja verdadeira
fi
```

Exemplo if simples

Neste exemplo estamos testando se o conteúdo da variável $num é maior que 5, se for maior, o retorno para o usuário será "o número é maior que cinco".

```
num = 5
if [ $num -gt 5 ]; then
      echo "o número é maior que cinco"
fi
```

Desvio condicional "else"

Também podemos agregar ao "***if***" o comando "***else***" que significa senão, ou seja, o contrário da condição testada. Se a condição for verdadeira é feito alguma ação, se a condição não for verdadeira (falsa) é efetuado outro tipo de ação.

```
if [ condição ]; then
      #ação que será tomada caso a condição seja verdadeira
else
      #ação que será tomada caso a condição seja falsa
fi
```

Exemplo if com else

```
num = 5
if [ $num -gt 5 ]; then
      echo "o número é maior que cinco"
else
      echo "o número é menor ou igual a cinco"
fi
```

UTILIZANDO IF ENCADEADO

O ***if*** encadeado serve para testar várias condições. Para testar várias condições, encadeamos vários testes condicionais, no caso vários "***else if***". O ***else if*** significa "senão se".

No *Shell Script* representamos o ***else if*** através do comando: "***elif***".

Sintaxe if else if

Obs: podemos ter "n" ***elifs*** no ***if*** encadeado, mas sempre a primeira condição será testada através de um "***if***", e no final teremos um "***else***" para executar uma ação caso nenhuma das condições tenham sido verdadeiras.

```
if [ condição1 ]; then
      #ação que será tomada caso a condição 1 seja verdadeira
elif [ condição2 ]; then
      #ação que será tomada caso a condição 2 seja verdadeira
elif [ condição3 ]; then
      #ação que será tomada caso a condição 3 seja verdadeira
else
      #ação que será tomada caso nenhuma condição seja verdadeira
fi
```

Exemplo if else if

```
num = 5
if [ $num -gt 5 ]; then
      echo "o número é maior que cinco"
elif [ $num -lt 5 ]; then
      echo "o número é menor que cinco"
else
      echo "o número é igual a cinco"
fi
```

EXEMPLOS

Exemplo 1

Vamos criar um *script* que receba dois números do usuário e coloca os mesmos em ordem crescente.

```
#!/bin/bash
clear
echo "Digite o primeiro número: "
read n1
echo "Digite o primeiro número: "
read n2
if [ $n1 -eq $n2 ]; then
      echo "Números iguais"
      sleep 5s
      clear
elif [ $n2 -gt $n1 ]; then
      echo "Ordem crescente: $n1 - $n2"
      sleep 5s
      clear
elif [ $n1 -gt $n2 ]; then
      echo "Ordem crescente: $n2 - $n1"
      sleep 5s
      clear
fi
```

Exemplo 2

Vamos criar um *script* que testa se um diretório existe, se existir ele avisa o usuário que existe, caso contrário ele cria o diretório.

```
#!/bin/bash
clear
echo "Digite o nome do diretório que deseja criar: "
read nome
if [ -e $nome  ]; then
      echo "diretório já existe!"
      sleep 5s
      clear
else
      mkdir $nome
      echo "Diretório criado com sucesso!"
fi
```

Exemplo 3

Vamos utilizar o mesmo exemplo da aula passada, porém vamos adicionar a estrutura condicional ***if***. Com isso apresentaremos para o usuário o seguinte menu:

```
Escolha:
1 – ver data do sistema
2 – ver calendário
```

Se o usuário digitar 1, aparecerá somente a data, se o usuário digitar 2 aparecerá o calendário.

Script mostrando o calendário e a data do sistema. Esse *script* não envia nenhuma informação para o usuário.

```
#!/bin/bash
clear
echo "Escolha:"
echo "1 – ver data do sistema"
echo "2 – ver calendário"
read op
if [ $op = 1 ]; then
      echo "Visualizando a data"
      date
      sleep 5s
      clear
elif [ $op = 2 ]; then
      echo "Visualizando o calendário"
      cal
      sleep 5s
      clear
else
      echo "Opção inválida"
fi
```

Obs: Note que todo conteúdo que está dentro do ***if***, ***elif*** e ***else*** está recuado para a direita, ou seja, utilizamos um "*tab*" para identar o código. A identação é uma forma de organização do código para que possamos ler o código mais rápido. Por exemplo, em uma redação, a entrada de parágrafo é uma forma de identação, ou seja, serve para visualizar melhor o texto que estamos lendo.