

Shell Script

Estruturas condicionais



Shell Script – Estruturas condicionais

Uma linguagem de programação não pode sobreviver sem estruturas condicionais. As estruturas condicionais são usadas em tarefas muito corriqueiras.

Para podermos compreender a estrutura condicional do Shell Script, primeiro temos que saber como ele testa se uma condição é falsa ou verdadeira. Para isso, o shell script trabalha com um código de retorno.

Esse código de retorno fica guardado dentro de uma variável que é representada por **\$?** (cifrão + ponto de interrogação).

De modo que se um comando no shell for executado com sucesso, o código de retorno será igual a ZERO.

Se um comando no shell falhar, o valor retornado será DIFERENTE DE ZERO.



Shell Script – Estruturas condicionais

Exemplo:

Listando um arquivo que **NÃO** existe:

```
$ ls nada
```

```
ls: nada: Arquivo ou diretório não encontrado
```

Qual deve ser o código de retorno?

Diferente de 0!

Verificando:

```
$ echo $?
```

```
2
```

O código de retorno foi **diferente de zero**, então o comando **NÃO** foi executado com sucesso.



Shell Script – Estruturas condicionais

O comando if

```
if <comando> Se <comando> for bem sucedido, isto é, $? igual a zero
then
    <comando_1>
    <comando_2>
    <comando_n>
else Se <comando> NÃO for bem sucedido, isto é, $? diferente de zero
    <comando_3>
    <comando_4>
    <comando_z>
fi
```



Shell Script – Estruturas condicionais

O comando if do Shell, no seu formato geral, não testa uma condição e sim se uma instrução foi executada com sucesso ou não, isto é, se seu código de retorno é ou não igual a zero, mas é possível testar condições também, veremos mais na frente.

Use o editor de textos VI para escrever esse script:

```
$ vi talogado.sh
# Verifica se um usuário está logado ou não no sistema
#!/bin/bash
if who | grep $1
    then
        echo $1 esta logado
    else
        echo $1 nao esta logado
    fi
```



Shell Script – Estruturas condicionais

O comando test

- Ora, mas você deve estar se perguntando se o comando if testa o conteúdo da variável **\$?**, como farei para testar condições???

Sintaxe:

```
$ test <expressão>
```

Exemplo:

```
$ resposta=N
```

Como é atribuição acima, sem brancos antes e depois do igual

```
$ test $resposta = N
```

Como é comparação acima, com brancos antes e depois do igual

```
$ echo $?
```

0 (Se a saída é ZERO, logo é verdadeiro o teste.)



Shell Script – Estruturas condicionais

Principais opções do comando test

test -r <arquivo> Testa se tem permissão de leitura
test -w <arquivo> Testa se tem permissão de escrita
test -x <arquivo> Testa se tem permissão de execução
test -f <arquivo> Testa se é um arquivo regular
test -d <arquivo> Testa se é um diretório
test -u <arquivo> Testa se seu SUID está ativado
test -g <arquivo> Testa se seu SGID está ativado
test -s <arquivo> Testa se seu tamanho é maior que zero

Exemplo:

```
$ test -r wget-log
```

```
$ echo $?
```

```
0
```



Shell Script – Estruturas condicionais

num1 -eq num2 : num1 é igual a num2 (equal to)
num1 -ne num2 : num1 é diferente de num2 (not equal to)
num1 -gt num2 : num1 é maior que num2 (greater than)
num1 -ge num2 : num1 é maior ou igual a num2 (greater or equal)
num1 -lt num2 : num1 é menor que num2 (less than)
num1 -le num2 : num1 é menor ou igual a num2 (less or equal)

Exemplo:

```
$ test 1 -eq 2
```

```
$ echo $?
```

1 (diferente de zero, pode ter como saída outro número)



Shell Script – Estruturas condicionais

&& (and / “e” lógico)

Um teste entre duas condições com **&&** só retorna verdadeiro se AMBAS condições forem se e somente se verdadeiras de acordo com a tabela verdade.

```
$ a=10
```

```
$ b=9
```

```
$ echo $a && echo $b
```

```
10
```

```
9
```



Shell Script – Estruturas condicionais

|| (or / ou lógico)

Um teste entre duas condições com **||** retorna verdadeiro se alguma dessas for verdadeira.

Se a primeira for verdadeira a segunda nem é executada.

```
$ a=10
```

```
$ b=9
```

```
$ echo $a || echo $b
```

```
10
```



Shell Script – Estruturas condicionais

O comando case

Usamos o comando case quando a quantidade de comandos condicionais (if) seja maior que três. Isso agiliza a execução, aumenta a legibilidade e diminui o tamanho do código.

Sintaxe:

case valor in

padr1)

<comando1>

<...>

<comandon>

::

padr2)

<comando1>

<...>

<comandon>

::

padr3)

<comando1>

<...>

<comandon>

::

esac



Shell Script – Estruturas condicionais

O valor é comparado a cada um dos padrões (padr1,padr2, ...,padrN), até que satisfaça a um deles, quando, então, passará a executar os comandos subseqüentes até que dois pontos-e-vírgulas sucessivos (;;) sejam encontrados.

Veremos um exemplo do case em Chat. :-)



Bibliografia

Programação Shell Linux 3ª edição

Autor: Julio Cezar Neves

Editora: BRASPORT

