

Programação Shell Script

REDES DE COMPUTADORES

Prof. Dr. João Paulo Aramuni

Construções da Linguagem

- * Shell Script – Linguagem Estruturada
- * **Tomar decisões** é uma parte importante do nosso dia-a-dia.
- * Da mesma maneira, tomar decisões também fazem parte das linguagens que são chamadas de **estruturadas**:
 - * C / Pascal e etc.
- * Shell Script também nos permite construir estruturas de **tomadas de decisão**.

Construções da Linguagem

- * Nesta aula iremos começar a estudar a linguagem estruturada bash e algumas construções como:
- * **Tomadas de decisão**
- * **Loops**

Tomadas de Decisão

- * Existe alguma diferença entre as tomadas de decisão da **vida real** e as tomadas de decisão utilizando **computadores**?

Tomadas de Decisão

- * Tomadas de decisão na vida real são um pouco **complicadas para nós**, imagine para os computadores que nem se quer tem capacidade para entender as nossas decisões na vida real.
- * Computadores conhecem apenas **0 (zero) e 1**, o que significa **Sim e Não**.

Tomadas de Decisão

- * Para esclarecer como isso funciona vamos **praticar um pouco** utilizando um programa de calcular do Linux.
- * `$ bc`

Digite bc em um terminal para iniciar o programa

Tomadas de Decisão

- * Assim que você digitar bc o programa irá esperar até que você digite uma expressão para ele calcular.
- * Vamos praticar:
 - * ***Digite 5 + 7 e tecla ENTER***
 - * ***Digite 5 - 2 e tecla ENTER***
 - * ***Digite 6 / 2 e tecla ENTER***

Tomadas de Decisão

- * Em todos os casos você obteve como resposta a solução para estas expressões aritméticas:

$$5 + 7 = 12$$

$$5 - 2 = 3$$

$$6 / 2 = 3$$

Tomadas de Decisão

- * O que vai acontecer se digitarmos $5 > 2$ para o programa calcular?

Tomadas de Decisão

- * Digite $5 > 2$ e tecle enter para ver o que acontece
 - * Resp: 1
- * Se eu perguntasse se $5 > 2$ todos diriam que SIM!
 - * O programa bc responde SIM **da maneira que ele conhece**, mostrando o **valor 1**

Tomadas de Decisão

- * Digite agora $5 < 2$ e tecle ENTER
 - * Resp: 0
- * O zero (0) indica falso.
 - * Lembre-se que **verdadeiro** é representado no programa pelo **valor 1**
 - * ... e **falso** é representado pelo **valor 0**.

Tomadas de Decisão

- * Essas expressões são chamadas de **relacionais** pois utilizam **operadores lógicos**:

>, <, >=, <=, ==, !=

maior, menor, maior igual, menor igual, igual, diferente

Tomadas de Decisão

- * Faça o teste dessas **operações relacionais** no programa bc, mas antes escreva no papel a sua resposta: SIM ou NAO para cada expressão.

Expressão	Significado para nós
$5 > 12$	5 é maior do que 12?
$5 == 10$	5 é igual a 10?
$5 != 2$	5 é diferente de 2?
$5 == 5$	5 é igual a 5?
$1 < 2$	1 é menor do que 2?

Tomadas de Decisão

Expressão	Sua resposta	Resposta do BC
$5 > 12$	Não	0
$5 == 10$	Não	0
$5 != 2$	Sim	1
$5 == 5$	Sim	1
$1 < 2$	Sim	1

Tomadas de Decisão

- * **ATENCAO, CUIDADO!**

- * O Shell Linux e o programa BC **possuem diferentes** formas de representar seus valores VERDADEIRO/FALSO.

- * Lembrem-se do status de saída de um script!

- * 0: Executou corretamente (Verdadeiro)
 - * 1: Não executou corretamente (Falso)

Tomadas de Decisão

Valor	Programa BC	Linux Shell
Verdadeiro / Sim	1	0
Falso / Não	0	Valor diferente de zero!

A condição If

- * A condição if é utilizada como forma de **tomada de decisão em shell script** e em outras linguagens.
- * Se a condição a ser verificada pelo if for verdadeira então ele executa o bloco verdadeiro (then).

if condição

then

executa se a condição for verdadeira (true) ou se o status de saída da condição for 0 (zero)

...

...

fi

A condição If

- * Uma condição não é nada além do que a *comparação entre dois valores*
- * Na condição do if podemos utilizar:
- * um teste (utilizando o comando *test*)
- * ... ou uma **[expr]** (expressão) que retorne um status de saída

A condição If

- * Definição de expressão:
- * Uma expressão não é nada além do que uma **combinação de:**
 - * Valores, operadores relacionais ($>$, $<$) e operadores matemáticos ($+$, $-$, $/$ e etc)

- * Exemplos:

$$6 > 3$$

$$a < b$$

$$c > 6$$

$$c > 6 + 31 - 1$$

A condição If

- * Crie um arquivo chamado **msg** e coloque a seguinte mensagem dentro dele
 - * Alo! Eu estou aqui!
- * Utilize o comando **cat** para verificar o conteúdo do arquivo e verifique o seu status de saída utilizando o comando: **echo \$?**
- * O seu console deve retornar o valor 0 (zero) indicando que o arquivo foi visualizado corretamente.

Vamos utilizar este retorno para validar a nossa condição do if como verdadeira.
Digite o script abaixo e veja o que acontece:

```
# !/bin/bash
# -----
# Nome: mostra_arq.sh
# -----
# Script para imprimir um arquivo na tela através da condição if
```

```
if cat $1
then
echo -e "\n\n Arquivo $1, encontrado e visualizado com sucesso!"
fi
```

```
#-----
```

Rode o script acima:

```
$ chmod 755 mostra_arq.sh
$ ./mostra_arq.sh msg
```

A condição If

- * Na programação shell script, o **nome do arquivo** que contem o script é equivalente a **\$0**.
- * O **argumento**, **\$1**, é passado logo após o nome do script que será executado:

```
$ ./<script> <argumento>
```

A condição If

- * O if imprime o arquivo com o comando "cat <argumento>" e compara se o **status de saída** do comando "cat msg" foi **0** (sucesso)
- * ... , ou seja, se o shell conseguiu visualizar o arquivo na tela através do comando cat o if se torna **verdadeiro**.

A condição If

- * A partir daí ele entra na **condição verdadeira** do if e mostra a mensagem através do comando echo.
- * Se a **condição for falsa**, ele não mostra a mensagem, pois não acessa o bloco verdadeiro (then).

Exercício

- * Crie um arquivo para ser apagado.
- * Logo após crie um script que remova qualquer arquivo através do argumento passado para ele na linha de comando
- * Se o script apagar o arquivo, mostre para o usuário:
“Arquívoo apagado com sucesso!”

* **RESP:**

```
# !/bin/bash
# -----
# Nome: testrm.sh
# -----
# Script para testar o status de saída do comando rm
# Para executar este programa passe o arquivo a ser apagado como
# argumento
# Ex: ./testerm.sh <arquivoaserapagado>

if rm $1
then
echo "$1 apagado com sucesso!"
fi
```

Exercício

- * Responda as questões abaixo referente ao script anterior (testerm.sh)
 - * a) Suponha que o arquivo aluno.txt exista no seu hd. Qual será a saída do comando `$./testerm.sh aluno.txt`
 - * b) Suponha que o arquivo aluno2.txt não exista no seu hd. Qual será a saída do comando `$./testerm.sh aluno.txt`
 - * c) Qual será a saída do comando `$./testerm.sh`

A condição If

- * Como construir *testes melhores* para a condição if?
- * Utilizando o *comando test* ou `[]`

Comando test ou [expr]

- * O comando test ou [expr] é utilizado para ver se uma expressão é verdadeira.
- * Se a expressão for verdadeira, o teste irá retornar zero (0).
- * Se o resultado do teste da expressão for falso irá retornar um valor diferente de zero.
- * **Sintaxe:**
- * test expressão OU [expressão]

Exemplo

- * O script abaixo determina se um numero, passado como argumento, é positivo.

```
# !/bin/bash
# -----
# Nome: ehpositivo.sh
# -----
# Script para verificar se um numero eh positivo
#
if test $1 -gt 0
then
echo "$1 eh positivo"
fi
```

Argumentos

- * O que é um **argumento**?
- * Como vamos fazer para rodar este script?

Argumentos

- * Não se esqueça que argumento é o que **passamos para funções** ou programas para serem processados.
- * Geralmente esses programas ou funções geram resultados após processar o argumento.

Exemplo

- * Para rodar o script

```
$ chmod 755 ehpositivo
```

- * Faça os testes abaixo e veja o resultado

```
$ ./ehpositivo.sh 5
```

```
$ ./ehpositivo.sh -5
```

```
$ ./ehpositivo.sh
```

Exemplo

- * Entendendo o script...

Exemplo

```
#!/bin/bash
# -----
# Nome: ehpositivo.sh
# -----
# Script para verificar se um numero eh positivo
#
if test $1 -gt 0
then
echo "$1 eh positivo"
fi
```

* Atenção!

O comando test ou [expr] funciona com:

1. Inteiros (Números sem casas decimais)
2. Tipos de arquivo
3. Cadeias de caracteres (strings)

A condição If

- * Utilizando testes condicionais em SHELL SCRIPT

Operador matemático	Significado	Operação Matemática tradicional	SHELL SCRIPT	
			Utilizando o comando test	Utilizando o comando [expr]
-eq	É igual a	$6==7$	if test 6 -eq 7	if [6 -eq 7]
-ne	Não é igual a	$6!=7$	if test 6 -ne 7	if [6 -ne 7]
-lt	É menor que	$6 < 7$	if test 6 -lt 7	if [6 -lt 7]
-le	É menor igual	$6 \leq 7$	if test 6 -le 7	if [6 -le 7]
-gt	É maior que	$6 > 7$	if test 6 -gt 7	if [6 -gt 7]
-ge	É maior igual	$6 \geq 7$	if test 6 -ge 7	if [6 -ge 7]

Obrigado.

Contato: joaopauloaramuni@gmail.com