

Programação Shell Script

REDES DE COMPUTADORES

Prof. Dr. João Paulo Aramuni

Iniciando a Programação Shell Script

- * Em computação, para podermos trabalhar com **dados e informações**, precisamos **armazená-los** na **memória RAM** do computador.

O que é memória RAM?

- * A **memória RAM** é um **espaço definido** em seu computador para **armazenar dados e informações** por um tempo determinado.

Como a memória RAM funciona?

- * A **memória RAM** é dividida em **pequenos espaços**:
 - * Cada um desses pequenos espaços recebe um **identificador único** chamado de **endereço**.
 - * Esses espaços são utilizados para **armazenar** nossos dados.

Como a memória RAM funciona?

- * Nós programadores podemos **definir** um nome específico para cada um desses **espaços** em nosso programa.

Como a memória RAM funciona?

- * Esses espaços são chamados de **Variáveis**.

Variáveis

- * Esses **espaços** são o que chamamos de **variáveis de memória** ou simplesmente variáveis.
- * É um **local de armazenamento**, que **possui um nome**, e que pode armazenar valores diferentes, um por vez.

Variáveis em Shell Script

- * Em Shell Script Linux existem dois tipos diferentes de variáveis:
 - * Variáveis **de Sistema**
 - * Criadas e administradas pelo próprio sistema Linux.
 - * Este tipo de variável é sempre definida (declarada) em **LETRAS MAIÚSCULAS**.
 - * Variáveis **definidas pelo Usuário (UDV)**
 - * Criadas e administradas pelo próprio usuário.
 - * Este tipo de variável é definida em **letras minúsculas**.

Variáveis de Sistema

- * Como podemos **exibir** a lista de **variáveis de sistema**?

Variáveis de Sistema

- * Para visualizar a lista de variáveis do sistema digite o comando “**env**” já estudado anteriormente.

- * Ex:

```
$ env
```

Variáveis de Sistema

- * Quais são as variáveis de sistemas já estudadas até agora?

```
$ echo $SHELL
```

```
$ echo $PATH
```

```
$ echo $USER
```

Variáveis de Sistema

- * Algumas variáveis de sistema **importantes**:

<i>Variável de Sistema</i>	<i>Significado</i>
BASH=/bin/bash	Nome do nosso shell
BASH_VERSION=1.14.7(1)	Nome da versão do nosso shell
COLUMNS=80	Nº. de colunas da nossa tela
HOME=/home/rafael	Nosso diretório home
LINES=25	Nº. de linhas da nossa tela
LOGNAME=rafael	Nome do usuário logado
HOSTTYPE=i386-Linux	Nosso tipo de Sistema Op.
PATH=/usr/bin:/sbin:/bin	Nossa configuração de path
PS1=[\u@\h \W]\\$	Nossa configuração de prompt
PWD=/home/rafael	Nosso diretório corrente
SHELL=/bin/bash	Nome do shell corrente
USERNAME=aluno	Nome do usuário

Variáveis de Sistema

- * Note que algumas das configurações apresentadas no seu ambiente Linux **podem ser diferentes**.
- * Você pode **imprimir** na tela qualquer **conteúdo** de qualquer uma das variáveis apresentadas utilizando o comando “**echo**”

```
$echo $USERNAME
```

```
$echo $HOME
```

Exercício

1) Se você quiser imprimir a localização de seu diretório home, qual comando você utilizaria:

a) `$ echo $HOME`

b) `$ echo HOME`

2) Qual dos comandos é o correto e por que?

Importante: Não modifique as suas variáveis de sistema por enquanto, isso pode lhe trazer grandes problemas.

Variáveis definidas pelo Usuário (UDV)

- * Como **definir** as variáveis definidas pelo usuário (UDV)?

Variáveis definidas pelo Usuário (UDV)

- * Sintaxe:
 - * `nomedavariavel=valor`
- * O valor é atribuído ao nome da variável, ou seja, o nome da variável agora passa a **guardar o valor**.
- * O valor sempre deve ser **colocado a direita** do sinal de igualdade (“=”).

Variáveis definidas pelo Usuário (UDV)

* Exemplos de UDV's

\$num=10

#Forma correta

\$10=num

#Erro! O valor deve ser colocado
a direita do sinal de =

Variáveis definidas pelo Usuário (UDV)

- * Para definir uma variável chamada **vetor** contendo o valor Teste

`$ vetor = Teste`

- * Para definir uma variável chamada n contendo o valor 10

`$ n = 10`

Regras para a criação de nomes de variáveis

- * 1º O nome da variável deve sempre **começar com um caractere alfanumérico** ou com o caractere **underscore (_)**, seguido por um ou mais caracteres alfanuméricos.

Regras para a criação de nomes de variáveis

- * Vejamos alguns exemplos de **variáveis válidas** em Shell Script.

HOME
SYSTEM_VERSION
vetor
num
_diretorio
_usuario

Regras para a criação de nomes de variáveis

- * 2º Nunca **coloque espaços** em nenhum dos lados do **sinal de igual** quando for **atribuir valor** a uma variável.

Regras para a criação de nomes de variáveis

- * O exemplo abaixo mostra a maneira correta de declarar variáveis:

- * `$ num=10` **#correto**

- * **Problemas!**

- * `$ num =10`

- * `$ num= 10`

- * `$ num = 10`

Regras para a criação de nomes de variáveis

- * 3º As variáveis são **case-sensitive** (igualmente ao sistema de arquivos em Linux). No exemplo abaixo todas as variáveis **são diferentes**:

\$ num=10

\$ Num=11

\$ NUM=20

\$ nUM=2

Regras para a criação de nomes de variáveis

- * Em relação ao exemplo anterior, o que devemos fazer para **imprimir** apenas o valor 20?

Regras para a criação de nomes de variáveis

- * Para mostrarmos na tela (imprimirmos) apenas o valor 20 devemos visualizar o conteúdo da variável **NUM**:
 - * `$ echo $ NUM`
- * Para imprimirmos os **valores restantes**:
 - * `$ echo $num # imprime 10 e não 20`
 - * `$ echo $Num # imprime 11 e não 20`
 - * `$ echo $nUM # imprime 2 e não 20`

Regras para a criação de nomes de variáveis

- * 4º Podemos criar variáveis nulas, NULL. (Uma variável nula é aquela que não possui valor quando é criada).

- * Ex:

- ```
$ vetor=
$ vetor=""
```

- \* Tente imprimir o valor da variável acima:

- \* 

```
$ echo $vetor
```

- \* **Nada será mostrado** porque a variável não possui valor nenhum, ou seja, seu **conteúdo é nulo**, NULL.

# Regras para a criação de nomes de variáveis

- \* 5º Nunca utilize **caracteres especiais** nos nomes das variáveis: **\*, ?, ^** e etc.

# Variáveis definidas pelo Usuário (UDV)

- \* Como podemos acessar ou **mostrar na tela** (imprimir) o **conteúdo das variáveis** definidas pelo Usuário (UDV)?

# Variáveis definidas pelo Usuário (UDV)

- \* Para imprimir ou **acessar o conteúdo** de uma UDV utilize a seguinte sintaxe:

- \* **Sintaxe:**

\$nomedavariavel

Obs: O acesso ao conteúdo é feito através do caractere \$. Basta colocar na frente da variável .

# Variáveis definidas pelo Usuário (UDV)

- \* Defina (Declare) as variáveis **vetor** e **n**, conforme abaixo:

```
$ vetor=Teste
```

```
$ n=10
```

- \* Para imprimir o conteúdo da variável **vetor** digite

```
$ echo $vetor
```

- \* Para imprimir o conteúdo da variável **n** digite o comando abaixo

```
$ echo $n
```

# Variáveis definidas pelo Usuário (UDV)

- \* ***Nunca tente*** imprimir o conteúdo de uma variável utilizando ***apenas o nome*** da variável, veja o exemplo:

\$ echo vetor **#Erro!**

- \* Isto irá imprimir na tela apenas o nome "vetor" ***e não o conteúdo*** da variável vetor.



# Variáveis definidas pelo Usuário (UDV)

- \* De maneira similar, se você tentar imprimir a variável **n** utilizando

`$ echo n` **#Erro!**

- \* O shell irá mostrar apenas o "n" na tela **e não o conteúdo** da variável.

# Variáveis definidas pelo Usuário (UDV)

- \* NUNCA se esqueça de **utilizar o caractere '\$'** seguido do nome da variável para **mostrar o conteúdo**.

- \* Ex:

\$nomedavariavel

\$vetor

\$n

# Exercícios

1. Declare a variável  $x$  com o valor 10 e mostre-a na tela.
2. Declare a variável  $xn$  com o valor "Aluno" e mostre-a na tela.
3. Mostre a soma de dois números na tela, utilize 6 e 3 por exemplo.
4. Declare duas variáveis  $x=20$  e  $y=5$ . Mostre na tela o resultado da divisão de  $x$  com  $y$ .
5. Modifique o exercício anterior para armazenar o resultado da divisão em uma variável chamada  $z$ .

# Exercícios

6. Indique, se existir, os erros no script abaixo:

```
#!/bin/bash
Testando o Meu conhecimento em Variaveis!
#
meunome=Aluno
meusistema = RuWinOS
meunum=5
echo "Meu nome eh $meunome"
echo "Meu sistema eh $meusistema"
echo "Minha matricula eh meunum, !!! "
```

Obrigado.

*Contato:* [joaopauloaramuni@gmail.com](mailto:joaopauloaramuni@gmail.com)