

SHELL SCRIPT

Rafael antonello

O que é?

- Shell é a interface entre
 - ▣ Usuário e o sistema
 - ▣ Similar ao cmd do windows
- O bash tá instalado em `/bin/bash`
- Para verificar a versão
 - ▣ `% bash --version`

Hello Word em Bash

- ❑ Criar um arquivo qualquer usando o gedit

```
#!/bin/bash
```

```
echo "hello word usando o bash"
```

Relembrando

- O programa shell interpreta os comandos que você digita
 - ▣ e os traduz para comandos que o kernel compreende
- O shell é uma linguagem de programação completa
 - ▣ Possui
 - Variáveis
 - Construções condicionais e iterativas
 - Ambiente adaptável ao usuário.

Relembrando

Nome Shell	Desenvolvido por	Onde	Comentários
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey	Free Software Foundation	Shell mais utilizado nas versões linux, é o shell padrão do Unix. Prompt \$
CSH (C SHell)	Bill Joy	University of California (For BSD)	Sintaxe similar a linguagem C. Traz vantagens adicionais ao Bourne shell como reutilizar e visualizar comandos previamente digitados. Prompt %
KSH (Korn SHell)	David Korn	AT & T Bell Labs	--
TCSH	-	--	TCSH é compatível com UNIX C shell (CSH

Para verificar shell disponível em sua máquina : **\$ cat /etc/shells**

Para verificar o shell ativo **\$ echo \$SHELL**

Quando Usar?



- ❑ Procedimentos complexos usando muitas linhas de comando
- ❑ Uma tarefa numa data planejada
- ❑ Integrar informações de vários sistemas existentes

Escrevendo programas em Shell

1. digitando os comandos e executando-os interativamente
 - ▣ o *prompt* normal \$ troca para > quando se inicia a digitar comandos na Shell
 - ▣ a seguir o programa é executado
2. armazenando os comandos em um arquivo e invocando da mesma forma que um programa
 - ▣ usando um editor de textos comum

Shell Script

Exemplo de shell script

- > arquivo chamado primeiro.sh
- > .sh sem significado
- > # indica comentário
- > /bin/bash =shell default
- > exit 0 código de retorno, 0 indica sucesso

```
#!/bin/bash
# primeiro.sh
echo "Hello Word"
exit 0
```

```
#!/bin/bash
# segundo.sh

clear
echo "Os arquivos do diretório são:"
ls -l|more
```


Como Escrever seu Script (1 / 2)

1. Escolha um editor de textos e inicie sua execução
2. Digitar script
 - ▣ Usando um editor de texto, nano, gedit, vi, etc...
3. Salvar
 - ▣ Obs.: Lembre onde salvou o script, você precisará executá-lo depois 😊
 - ▣ **Salve o script com algum nome sem espaço e com a extensão (.sh) – Ex. meuscript.sh (de preferência tudo em minúsculo)**
4. Acertar permissões de execução para o dono:
 - ▣ **chmod +x nomescript**
 - Troque nomescript pelo nome do script que você salvou no item 3

Como Escrever seu Script (2/2)

- Como executar o seu script
 - ▣ Abra o terminal (shell)
 - ▣ Navegue até a pasta onde você salvou seu script
 - Ex. (se vc salvou seu script home do aluno)
 - `cd /home/aluno/`
 - ▣ Execute o script
 - **`./nomescript`**

Shell Script

Escreva o seguinte script, salve e execute.

Nome do script: **teste.sh**

```
# Script para imprimir informações sobre o usuário logado
clear
echo "Ola $USER"
echo "Hoje é ";date
echo "Calendario"
cal 2011
exit 0
```

Obs.: Lembre-se que todo script deve iniciar com `#!/bin/bash`

Características

Aspas duplas “ “ - string

– mostra o string e substitui variáveis (se existir)

Aspas simples ´ - string

– mostra o string e não substitui variáveis (se existir)

Crase ` - comando

– o resultado do comando é usado em output

**Barra Invertida **

Transforma caracter especial em caracter normal

– permite a inserção de comentários no script

```
# testando variaveis
dia=03
echo " Hoje é dia $dia"
resultado=`Hoje é dia $dia`
maquina=`hostname`
echo $resultado $maquina
```

Manipulando Variáveis

- ❑ Variáveis locais
- ❑ Atribuição de valor

```
nome="fulano"  
idade=16
```

Manipulando Variáveis

- Um detalhe importante: **NÃO PODE EXISTIR ESPAÇOS ENTRE OS ELEMENTOS DA DECLARAÇÃO/ATRIBUIÇÃO**
- Para acessar o valor da variável temos que usar o **\$** na frente dela

```
msg="Hello Word"  
echo $msg
```

- As variáveis são **case-sensitive**, isto é, existe diferenças entre maiúsculas e minúsculas.

```
A=5  
a=5
```

Treinando...

- Definir uma variável J com o valor 5 e mostrar na tela
- Definir uma variável SALA cujo conteúdo deve ser “Laboratorio 71” e a seguir mostrar na tela

Realizando Cálculos

- Use o a construção `$ ((expressao))`
- Supondo que temos a variável TAMANHO já definida
 - ▣ Soma
 - `var=$ (($TAMANHO + 5))`
 - ▣ Subtração
 - `var=$ (($TAMANHO - 5))`
 - ▣ Multiplicação
 - `var=$ (($TAMANHO * 5))`
 - ▣ Divisão
 - `var=$ (($TAMANHO / 5))`
 - ▣ Módulo
 - `var=$ (($TAMANHO % 5))`
 - ▣ Exponenciação
 - `var=$ (($TAMANHO ** 5))`

Exemplos

```
#!/bin/bash  
  
res=$(( 2 + 3 ))  
echo $res
```

```
#!/bin/bash  
  
echo "Digite um valor"  
read a  
res=$(( $a * 3 ))  
echo $res
```

Exercitando

- Crie scripts para...
 - ▣ Mostrar o resultado da soma de $256 + 123$
 - ▣ Ler dois números quaisquer e mostrar a soma deles
 - ▣ Ler dois números quaisquer e mostrar a subtração entre eles
 - ▣ Ler 3 números e mostrar a soma deles
 - ▣ Ler 4 números e mostrar a soma deles
 - ▣ Ler dois números A e B quaisquer e trocar o valor de A com B
 - Ou seja, o valor de B vai pra A, e o valor de A vai pra B
 - ▣ Ler um número e incrementá-lo em 4 unidades

Exercitando

- Crie scripts para...
 - ▣ Mostrar o resultado da multiplicação de $2 * 9$
 - ▣ Ler dois números quaisquer e mostrar a multiplicação entre eles
 - ▣ Ler dois números quaisquer mostrar a divisão entre eles
 - ▣ Ler 3 números e mostrar a multiplicação deles
 - ▣ Ler 4 números e mostrar a multiplicação deles
 - ▣ Calcular o resultado de 2 elevado a 3
 - ▣ Ler dois números x e y e calcular o valor de x elevado a y

Entrada e Saída

20

read -> captura de variável no teclado, termina ao pressionarmos o return

echo -> impressão no vídeo

```
echo "digite o seu nome : "  
read nome  
echo "digite a sua idade : "  
read idade  
echo " bom dia $nome voce tem apenas $idade anos?"
```

Faça um script e grave-o no arquivo esp01.sh. O Script deve solicitar as seguintes informações:

Digite Primeiro Numero:

Digite Segundo Numero:

Após a entrada de dados imprima no vídeo a seguinte frase:

O Usuario XXX digitou os numeros XX e XX

A soma dos números foi XXX

Entrada e Saída

□ echo

▣ Opções

- -n -> Não pula linha

▣ Ex.: Veja a diferença entre usar echo com e sem o -n

```
echo "digite sua idade >"  
read idade  
echo "sua idade é $idade"
```

```
echo -n "digite sua idade > "  
read idade  
echo "sua idade é $idade"
```

Sleep

- sleep <n>
 - ▣ Faz o script parar por n segundos
 - ▣ Ex.: Leia um número qualquer e mostre a mensagem “O número lido foi : <numero>”.
 - Porém a mensagem deve ser apresentada 10 segundos após o usuário digitar o número

SHELL SCRIPT

Rafael antonello