



## Introdução à Ciência da Computação – Lista 6 Shell script – parte 3

Nome: Leticia Santos Alves

RA:2025.1.08.016

- 1) Crie um script chamado scriptaritmetico, com uma operação aritmética arbitrária usando pelo menos 4 variáveis, realizando uma operação de divisão cujo resultado não seja um número inteiro. Execute o script e mostre o resultado. Qual o recurso a ser utilizado caso você queira que o valor não inteiro apareça no resultado? Qual variável eu uso para isso?

```
leticialuthor@Luthor-Notebook2:~$ gedit scriptaritmetico.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x scriptaritmetico.sh
leticialuthor@Luthor-Notebook2:~$ ./scriptaritmetico.sh
Resultado da operação: 3.00
```

```
scriptaritmetico.sh
1 #!/bin/bash
2
3 a=10
4 b=3
5 c=2
6 d=1
7 resultado=$(echo "scale=2; ($a + $c) / ($b + $d)" | bc)
8 echo "Resultado da operação: $resultado"
```

Caso eu queira que o valor inteiro não apareça no resultado, eu devo usar o recurso 'bc', e a variável 'scale'.

- 2) Ponha em execução a calculadora bc. Mostre o uso da variável scale, exibindo um resultado de operação aritmética com 6 casas decimais.

```
leticialuthor@Luthor-Notebook2:~$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software
Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type 'warranty'.
scale=6
100 / 3
33.333333
```


- 3) Crie um script simples chamado testeabc, em que você utilize a calculadora bc dentro dele, envolvendo o uso de algumas variáveis e a operação de divisão, com o direcionamento via pipe. Execute o script, mostrando o resultado.

```
leticialuthor@Luthor-Notebook2:~$ gedit testeabc.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x testeabc.sh
leticialuthor@Luthor-Notebook2:~$ ./testeabc.sh
4.25000
```

```
testeabc.sh
1 #!/bin/bash
2 num1=17
3 num2=4
4 scale_value=5
5 resultado=$(echo "scale=$scale_value; $num1 / $num2" | bc)
6 echo "$resultado"
```

- 4) Crie um script chamado testebccomplexo, em que você utilize operações aritméticas diversas com a calculadora bc (pelo menos duas), armazenando os resultados em variáveis, como mostrado na aula. Neste caso, utilize a técnica de redirecionamento de entrada inline. Execute o script, mostrando o resultado.

```
leticialuthor@Luthor-Notebook2:~$ gedit testeabc.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x testeabc.sh
leticialuthor@Luthor-Notebook2:~$ ./testeabc.sh
4.25000
leticialuthor@Luthor-Notebook2:~$ gedit testeabc.sh
leticialuthor@Luthor-Notebook2:~$ gedit testebccomplexo.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x testebccomplexo.sh
leticialuthor@Luthor-Notebook2:~$ ./teste
testeabc.sh      testebccomplexo.sh
leticialuthor@Luthor-Notebook2:~$ ./testebccomplexo.sh
Resultado: 33.50
```



```
#!/bin/bash
var1=10
var2=3
var3=7
var4=2
resultado=$(bc << EOF
scale=2
a=($var1 * $var2)
b=($var3 / $var4)
a + b
EOF
)
echo "Resultado: $resultado"
```

- 5) O que consiste o status de saída de um programa? Mostre um exemplo de execução de dois comandos (um com sucesso e outro desconhecido) e verifique esse status. Mostre em tela.

O status de saída de um comando executado com sucesso é 0. Se houver erro, será um inteiro positivo.

```
leticialuthor@Luthor-Notebook2:~$ echo "oi meu nome é leticia"
oi meu nome é leticia
leticialuthor@Luthor-Notebook2:~$ echo $?
0
leticialuthor@Luthor-Notebook2:~$ kkkkkkkk
kkkkkkkk: comando não encontrado
leticialuthor@Luthor-Notebook2:~$ echo $?
127
```

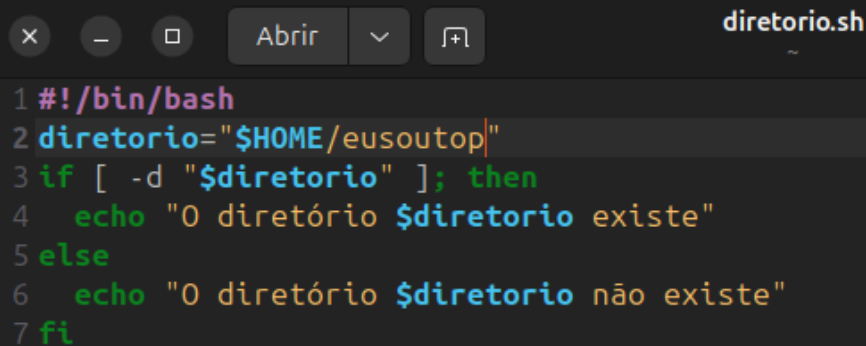
- 6) Qual a função do comando exit? Mostre um exemplo do uso do comando exit dentro de um script, mudando o valor padrão do status de saída. Mostre tanto o uso do exit exibindo um número qualquer até 255, quanto o valor de uma variável que você utilize no script. Execute o script e mostre o valor do status de saída em cada caso.

O comando exit permite especificar um status de saída quando o script finaliza.

```
leticialuthor@Luthor-Notebook2:~$ gedit exit.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x exit.sh
leticialuthor@Luthor-Notebook2:~$ ./exit.sh
Exit com número:
leticialuthor@Luthor-Notebook2:~$ gedit exit.sh
leticialuthor@Luthor-Notebook2:~$ ./exit.sh
Com variável:
leticialuthor@Luthor-Notebook2:~$ echo $?
27
```

- 7) Crie um script simples envolvendo comandos condicionais if then else, para verificar a existência de um diretório específico no seu home. Primeiro procure um diretório inexistente, depois um diretório existente e exiba as mensagens específicas de acordo com o resultado. Execute o script e mostre em tela.

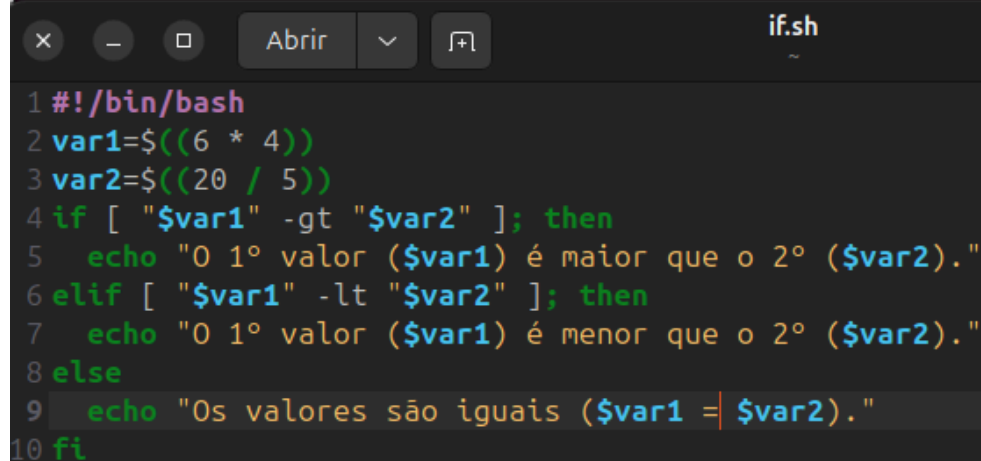
```
leticialuthor@Luthor-Notebook2:~$ gedit diretorio.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x diretorio.sh
leticialuthor@Luthor-Notebook2:~$ ./diretorio.sh
0 diretório /home/leticialuthor/eusoutop1 não existe
leticialuthor@Luthor-Notebook2:~$ gedit diretorio.sh
leticialuthor@Luthor-Notebook2:~$ ./diretorio.sh
0 diretório /home/leticialuthor/eusoutop existe
```



```
1 #!/bin/bash
2 diretorio="$HOME/eusoutop1"
3 if [ -d "$diretorio" ]; then
4     echo "0 diretório $diretorio existe"
5 else
6     echo "0 diretório $diretorio não existe"
7 fi
```

- 8) Crie um script envolvendo várias condicionais usando a estrutura if then elif else, fazendo duas operações aritméticas arbitrárias, verificando o valor das variáveis que armazenam essa operação, checando se o valor da primeira é maior, menor ou igual ao valor da segunda. Execute o script e mostre o resultado em tela.

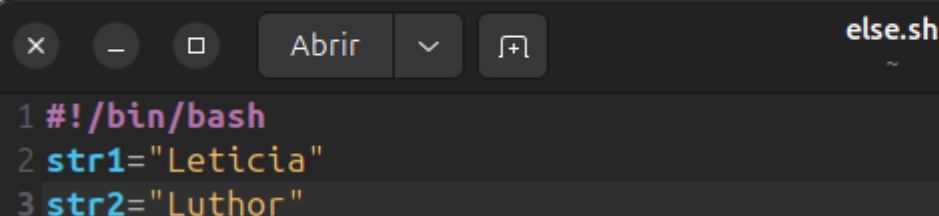
```
leticialuthor@Luthor-Notebook2:~$ gedit if.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x if.sh
leticialuthor@Luthor-Notebook2:~$ ./if.sh
0 1º valor (24) é maior que o 2º (4).
```



```
1 #!/bin/bash
2 var1=$((6 * 4))
3 var2=$((20 / 5))
4 if [ "$var1" -gt "$var2" ]; then
5     echo "0 1º valor ($var1) é maior que o 2º ($var2)."
```

- 9) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas variáveis string arbitrárias e verificando seus valores, checando se o conteúdo das variáveis é igual. Execute o script e mostre o resultado em tela.

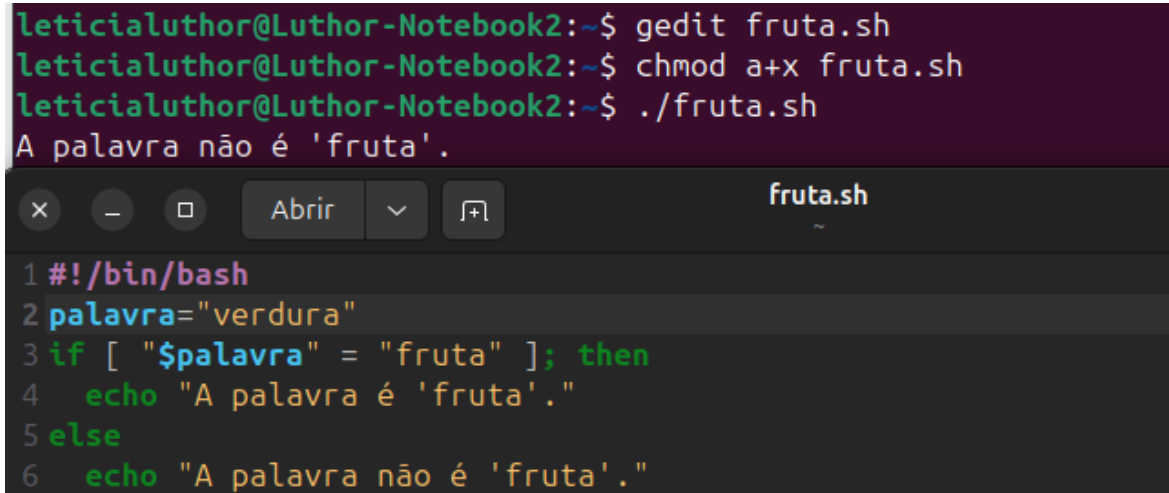
```
leticialuthor@Luthor-Notebook2:~$ gedit else.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x else.sh
leticialuthor@Luthor-Notebook2:~$ ./else.sh
As strings são diferentes.
```



```
1 #!/bin/bash
2 str1="Leticia"
3 str2="Luthor"
```

- 10) Crie um script envolvendo condicionais usando a estrutura if then else, criando uma string com um conteúdo, verificando se seu valor é "fruta". Execute o script e mostre o resultado em tela.

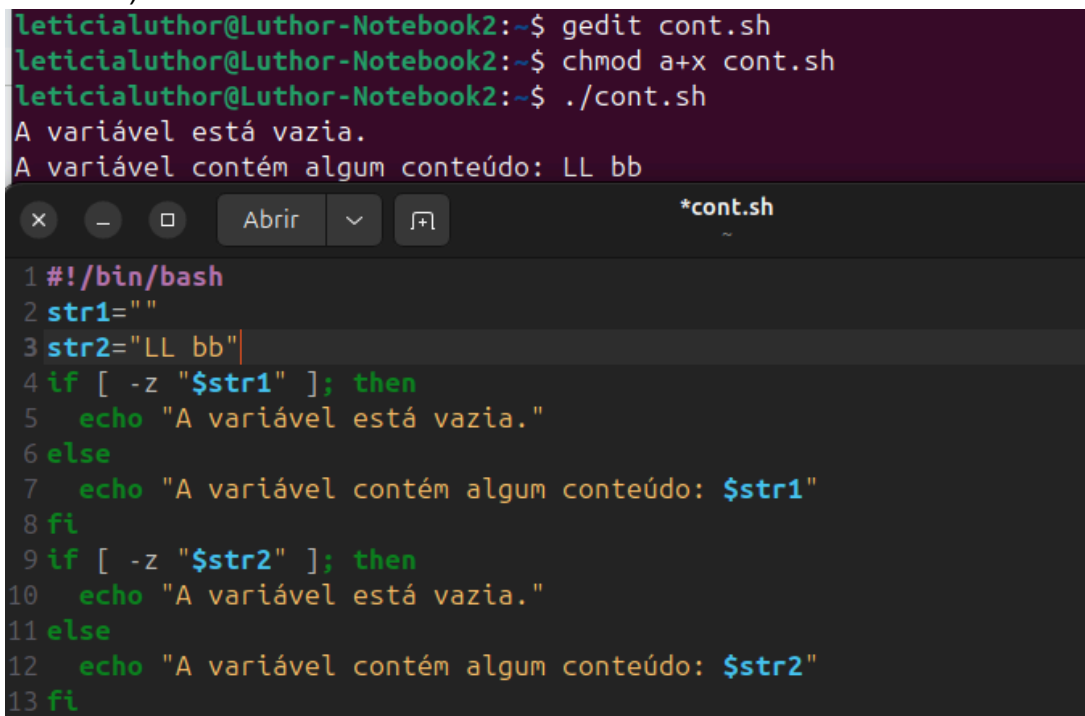
```
leticialuthor@Luthor-Notebook2:~$ gedit fruta.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x fruta.sh
leticialuthor@Luthor-Notebook2:~$ ./fruta.sh
A palavra não é 'fruta'.
```



```
1 #!/bin/bash
2 palavra="verdura"
3 if [ "$palavra" = "fruta" ]; then
4     echo "A palavra é 'fruta'."
5 else
6     echo "A palavra não é 'fruta'."
```

- 11) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas strings, uma vazia, outra com conteúdo e verificando estes resultados (se tem conteúdo em ambos os casos).

```
leticialuthor@Luthor-Notebook2:~$ gedit cont.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x cont.sh
leticialuthor@Luthor-Notebook2:~$ ./cont.sh
A variável está vazia.
A variável contém algum conteúdo: LL bb
```



```
1 #!/bin/bash
2 str1=""
3 str2="LL bb"
4 if [ -z "$str1" ]; then
5     echo "A variável está vazia."
6 else
7     echo "A variável contém algum conteúdo: $str1"
8 fi
9 if [ -z "$str2" ]; then
10    echo "A variável está vazia."
11 else
12    echo "A variável contém algum conteúdo: $str2"
13 fi
```

- 12) Cite 5 opções de comparações envolvendo arquivos. Escolha uma das opções e crie um script envolvendo essa opção.

-d arquivo → Verifica se o arquivo existe e se é um diretório;

-e arquivo → Verifica se o arquivo existe;

-f arquivo → Verifica se o arquivo existe e se é um arquivo;

-s arquivo → Verifica se o arquivo existe e não está vazio;

-w arquivo → Verifica se o arquivo existe e se tem permissão de escrita.

```
leticialuthor@Luthor-Notebook2:~$ gedit comp.sh
leticialuthor@Luthor-Notebook2:~$ chmod a+x comp.sh
leticialuthor@Luthor-Notebook2:~$ ./comp.sh
Testando: /home/leticialuthor/eusoutop
O caminho /home/leticialuthor/eusoutop é um diretório.
```

comp.sh

```
1 #!/bin/bash
2 dir="$HOME/eusoutop"
3 echo "Testando: $dir"
4 if [ -d "$dir" ]; then
5     echo "O caminho $dir é um diretório."
6 else
7     echo "O caminho $dir não é um diretório."
8 fi
```