



## Introdução à Ciência da Computação – Lista 6

### Shell script – parte 3

Nome: Gustavo Andrade M de Assis RA: 2024.1.08.012

- 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?

```
scriptaritmetico.sh  Ln 6, Col 7
2
3 var1=4
4 var2=3
5 var3=5
6 var4=1
7
8 resposta=$(echo "scale=2; ($var1 * $var2) / ($var3 * $var4)" | bc)
9
10 echo "O resultado é: $resposta"
```

```
gustavo@gustavo-1-2: ~
gustavo@gustavo-1-2:~$ ls
Desktop      intcc          ola_mundo.txt  scriptaritmetico.sh  teste
Documents    Music          Pictures       snap                trabalho
Downloads    NetBeansProjects  Public        Templates           Videos
gustavo@gustavo-1-2:~$ ./scriptaritmetico.sh
O resultado é: 2.40
gustavo@gustavo-1-2:~$
```


**Para fazer com que o valor não inteiro apareça é preciso usar a variável scale e escolher quantas casas decimais aparecerão.**

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



```
gustavo@gustavo-1-2: ~  
gustavo@gustavo-1-2:~$ 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  
10/3  
3.333333  
quit  
gustavo@gustavo-1-2:~$
```

3) Crie um script simples chamado testebc, 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.



```
Open ▾  testebc.sh  Ln 6, Col 39  
1 #!/bin/bash  
2  
3 var1=10  
4 var2=3  
5  
6 resposta=$(echo "scale=6; $var1 / $var2" | bc)  
7 echo "O resultado da divisão entre $var1 e $var2 é: $resposta"
```

```

gustavo@gustavo-1-2:~$ ls
Desktop      Music          Public          teste
Documents    NetBeansProjects scriptaritmetico.sh testebc.sh
Downloads    ola_mundo.txt  snap            trabalho
intcc        Pictures        Templates       Videos
gustavo@gustavo-1-2:~$ ./testebc.sh
0 resultado da divisão entre 10 e 3 é: 3.333333
gustavo@gustavo-1-2:~$

```

- 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.

```

Open  testebccomplexo.sh  Ln 5, Col 37
1  #!/bin/bash
2
3
4  resposta1=$(bc <<< "scale=2; (10 + 5) * 2.5")
5  resposta2=$(bc <<< "scale=2; sqrt(49)")
6
7
8  echo "0 resultado da primeira operação é: $resposta1"
9  echo "0 resultado da segunda operação é: $resposta2"

```

```

gustavo@gustavo-1-2: ~
gustavo@gustavo-1-2:~$ ls
Desktop      Music          Public          teste          Videos
Documents    NetBeansProjects scriptaritmetico.sh testebccomplexo.sh
Downloads    ola_mundo.txt  snap            testebc.sh
intcc        Pictures        Templates       trabalho
gustavo@gustavo-1-2:~$ chmod +x testebccomplexo.sh
gustavo@gustavo-1-2:~$ ls
Desktop      Music          Public          teste          Videos
Documents    NetBeansProjects scriptaritmetico.sh testebccomplexo.sh
Downloads    ola_mundo.txt  snap            testebc.sh
intcc        Pictures        Templates       trabalho
gustavo@gustavo-1-2:~$ ./testebccomplexo.sh
0 resultado da primeira operação é: 37.5
0 resultado da segunda operação é: 7.00
gustavo@gustavo-1-2:~$

```

- 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.

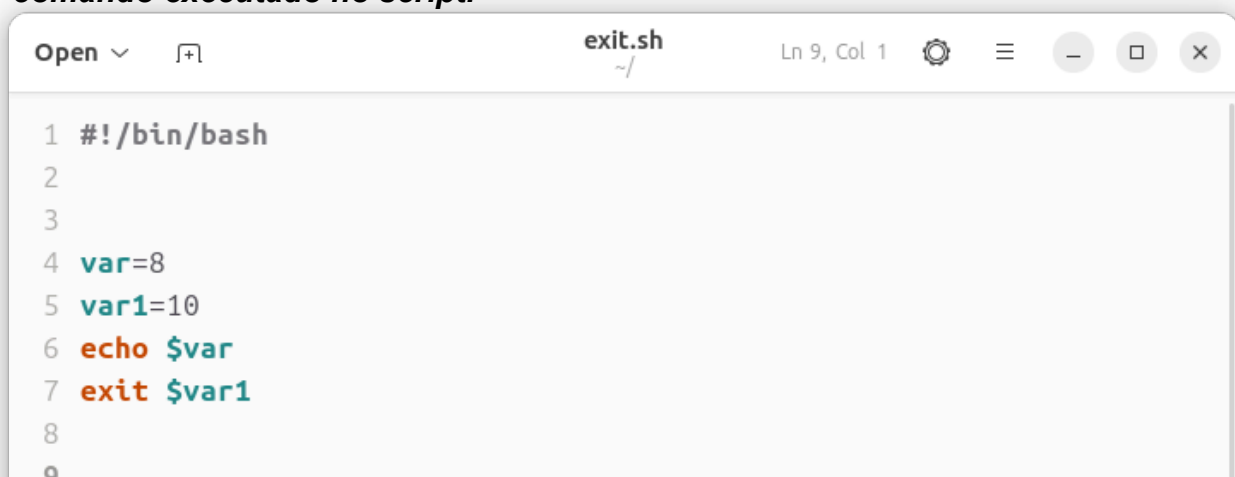
**Consiste em um valor numérico que indica se o programa foi executado com sucesso ou se ocorreu algum erro durante a execução.**



```
gustavo@gustavo-1-2: ~  
gustavo@gustavo-1-2:~$ ls  
Desktop      Music          Public          teste          Videos  
Documents    NetBeansProjects scriptaritmetico.sh testebccomplexo.sh  
Downloads    ola_mundo.txt  snap            testebc.sh  
intcc        Pictures       Templates       trabalho  
gustavo@gustavo-1-2:~$ echo $?  
0  
gustavo@gustavo-1-2:~$ command_aaa  
command_aaa: command not found  
gustavo@gustavo-1-2:~$ echo $?  
127  
gustavo@gustavo-1-2:~$
```

- 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.

**A função do comando exit é finalizar o shell script com o status de saída do último comando executado no script.**



```
Open  exit.sh  Ln 9, Col 1  
1 #!/bin/bash  
2  
3  
4 var=8  
5 var1=10  
6 echo $var  
7 exit $var1  
8  
9
```

```
gustavo@gustavo-1-2: ~  
gustavo@gustavo-1-2:~$ ls  
Desktop      intcc          Pictures       Templates     trabalho  
Documents    Music          Public         teste         Videos  
Downloads    NetBeansProjects scriptaritmetico.sh testebccomplexo.sh  
exit.sh      ola_mundo.txt  snap          testebc.sh  
gustavo@gustavo-1-2:~$ ./exit.sh  
8  
gustavo@gustavo-1-2:~$ echo $?  
10  
gustavo@gustavo-1-2:~$
```

- 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.

```
Open  ▾  verificadir.sh  Ln 9, Col 27  
1  #!/bin/bash  
2  
3  
4  dir="/home/gustavo/intcc"  
5  
6  if [ -d "$dir" ]; then  
7      echo "O diretório $dir existe."  
8  else  
9      echo "O diretório $dir não existe."  
10 fi
```

```
gustavo@gustavo-1-2:~$ ./verificadir.sh  
O diretório /home/gustavo/intcc existe.  
gustavo@gustavo-1-2:~$
```

```
Open  ▾  [🔍]  verificadir.sh  Ln 4, Col 23  ⚙️  ☰  -  □  ×

1  #!/bin/bash
2
3
4  dir="/home/gustavo/dir"
5
6  if [ -d "$dir" ]; then
7      echo "O diretório $dir existe."
8  else
9      echo "O diretório $dir não existe."
10 fi
11
```

```
gustavo@gustavo-1-2:~$ ./verificadir.sh
O diretório /home/gustavo/dir não existe.
gustavo@gustavo-1-2:~$
```

- 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.

```
Open  ▾  [🔍]  conta.sh  Ln 12, Col 41  ⚙️  ☰  -  □  ×

1  #!/bin/bash
2
3  op1=$((15 * 5))
4  op2=$((5 + 15))
5
6  # Verificar qual operação é maior, menor ou igual
7  if [ "$op1" -gt "$op2" ]; then
8      echo "A operação 1 é maior que a operação 2."
9  elif [ "$op1" -lt "$op2" ]; then
10     echo "A operação 1 é menor que a operação 2."
11 else
12     echo "As operações dão o mesmo valor."
13 fi
14
```

```
gustavo@gustavo-1-2:~$ ls
conta.sh  Documents  intcc  NetBeansProjects  Public  Templates  trabalho
Desktop   Downloads  Music  Pictures          snap    teste      Videos
gustavo@gustavo-1-2:~$ chmod +x conta.sh
gustavo@gustavo-1-2:~$ ./conta.sh
A operação 1 é maior que a operação 2.
gustavo@gustavo-1-2:~$
```

- 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.

A screenshot of a code editor window titled 'string.sh'. The editor shows a bash script with line numbers 1 through 13. The script sets two variables, string1 and string2, and uses an if-then-else structure to compare them. The window has a standard Linux-style title bar with 'Open', a file icon, the filename 'string.sh', and window controls. The status bar at the bottom right shows 'Ln 11, Col 25' and some icons.

```
1 #!/bin/bash
2
3
4 string1="banana"
5 string2="laranja"
6
7 # Verifique se o conteúdo das variáveis é igual
8 if [ "$string1" = "$string2" ]; then
9     echo "As strings são iguais."
10 else
11     echo "As strings são diferentes."
12 fi
13
```

```
gustavo@gustavo-1-2:~$ ls
conta.sh  Downloads  NetBeansProjects  snap    teste
Desktop   intcc      Pictures          string.sh  trabalho
Documents Music     Public           Templates  Videos
gustavo@gustavo-1-2:~$ chmod +x string.sh
gustavo@gustavo-1-2:~$ ./string.sh
As strings são diferentes.
gustavo@gustavo-1-2:~$
```

- 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.



The image shows a code editor window titled 'stringfruta.sh' with the following content:

```
6
7 if [ "$string" = "fruta" ]; then
8     echo "A string é 'fruta'."
9 else
10    echo "A string não é 'fruta'."
11 fi
12
```

Below the editor is a terminal window showing the execution of the script:

```
gustavo@gustavo-1-2:~$ chmod +x stringfruta.sh
gustavo@gustavo-1-2:~$ ./stringfruta.sh
A string 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).



The image shows a code editor window titled 'verificastring.sh' with the following content:

```
1 #!/bin/bash
2
3
4 string1=""
5 string2="Aopa"
6
7
8 if [ -z "$string1" ]; then
9     echo "A string 1 está vazia."
10 else
11     echo "A string 1 não está vazia."
12 fi
13
14 if [ -z "$string2" ]; then
15     echo "A string 2 está vazia."
16 else
17     echo "A string 2 não está vazia."
18 fi
19
```



```
gustavo@gustavo-1-2:~$ chmod +x verificastring.sh
gustavo@gustavo-1-2:~$ ./verificastring.sh
A string 1  está vazia.
A string 2 não está vazia.
gustavo@gustavo-1-2:~$
```

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;
- f arquivo : Verifica se o arquivo existe e se é um arquivo;
- s arquivo : Verifica se o arquivo existe e não está vazio;
- x arquivo : Verifica se o arquivo existe e tem permissão de execução;
- G arquivo : Verifica se o arquivo existe e se seu grupo padrão é o mesmo do usuário atual;

```
1 #!/bin/bash
2
3
4 if [ -x "$(nome_arquivo)" ]; then
5     echo "O arquivo $(nome_arquivo) é executável."
6 else
7     echo "O arquivo $(nome_arquivo) não é executável."
8 fi
9 |
```