



## Introdução à Ciência da Computação – Lista 6

### Shell script – parte 3

Nome: Diego Vianna Leite Montemor RA: 2025.1.08.023

- 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?

**R: Uso do utilitário bc com scale para permitir números com ponto flutuante (decimal), pois o bash só faz contas com inteiros por padrão.**

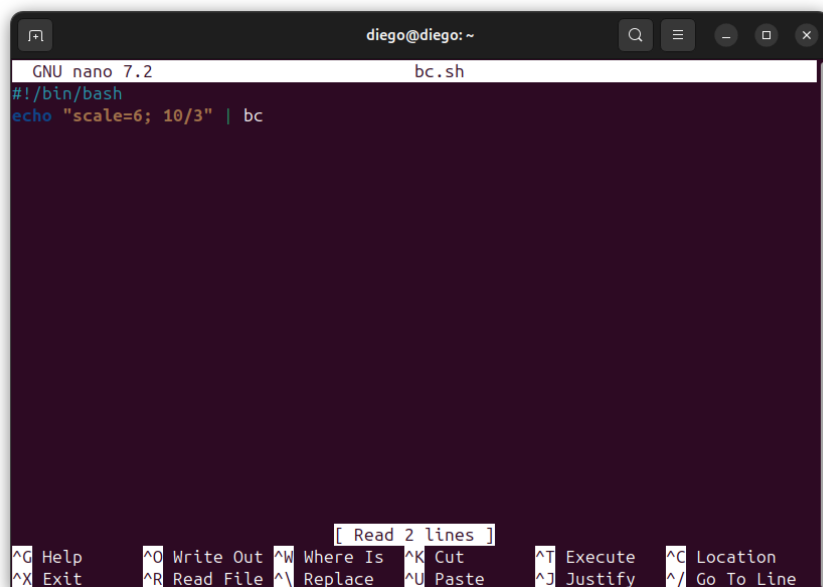
```
GNU nano 7.2 scriptaritmetico.sh
#!/bin/bash

a=10
b=3
c=1
d=1
#((a+c)/(b+d))

resultado=$(echo "scale=2; ($a+$c)/($b+$d)" | bc)
echo "Resultado de divisão: $resultado"
```

```
diego@diego: ~
diego@diego:~$ nano scriptaritmetico.sh
diego@diego:~$ chmod +x scriptaritmetico.sh
diego@diego:~$ ./scriptaritmetico.sh
Resultado de divisão: 2.75
diego@diego:~$
```

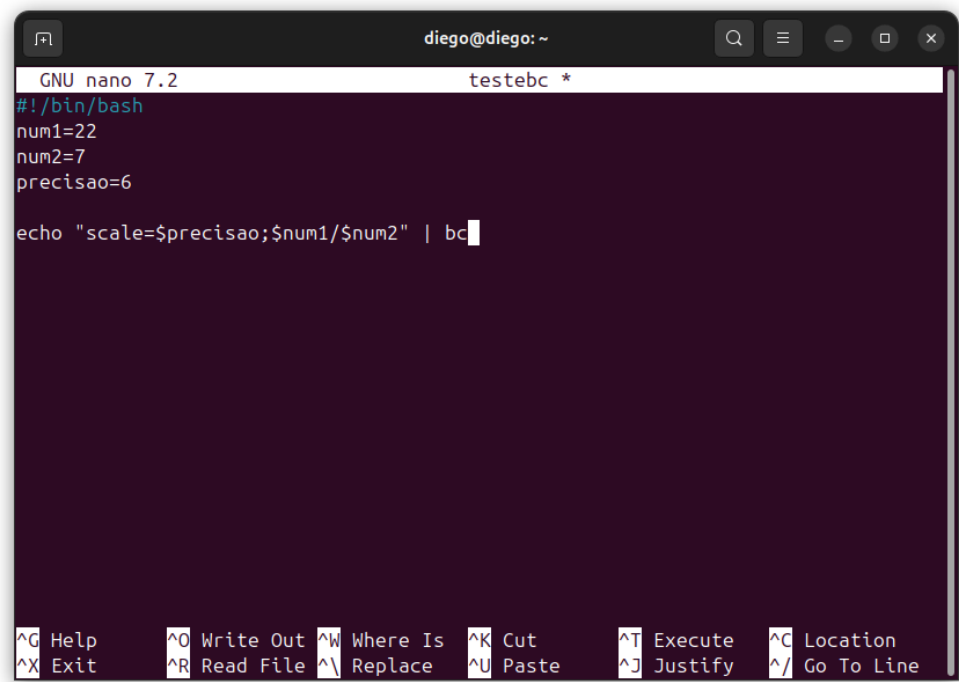
- 2) Ponha em execução a calculadora bc. Mostre o uso da variável scale, exibindo um resultado de operação aritmética com 6 casas decimais.



```
GNU nano 7.2 bc.sh
#!/bin/bash
echo "scale=6; 10/3" | bc
```

```
diego@diego:~$ nano bc.sh
diego@diego:~$ chmod +x bc.sh
diego@diego:~$ ./bc.sh
3.333333
diego@diego:~$
```

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.



```
GNU nano 7.2 testebc *
#!/bin/bash
num1=22
num2=7
precisao=6

echo "scale=$precisao;$num1/$num2" | bc
```

```
diego@diego:~$ nano testebc
diego@diego:~$ chmod +x testebc
diego@diego:~$ ./testebc
3.142857
diego@diego:~$
```

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.



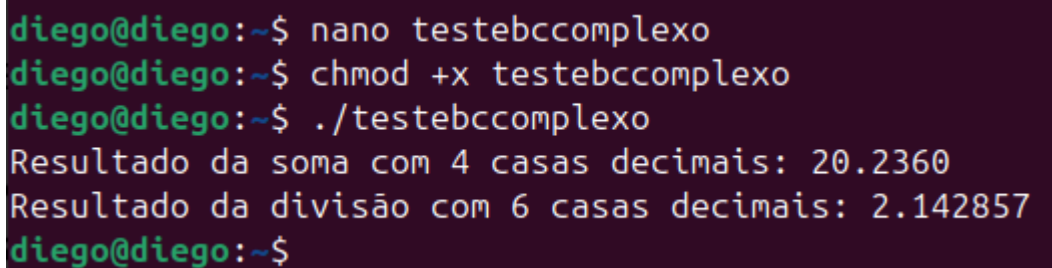
```
GNU nano 7.2 testebccomplexo *
#!/bin/bash

resultado1=$(bc << EOF
scale=4
a=12.3450
b=7.8910
a+b
EOF
)

resultado2=$(bc << EOF
scale=6
x=15
y=7
x/y
EOF
)

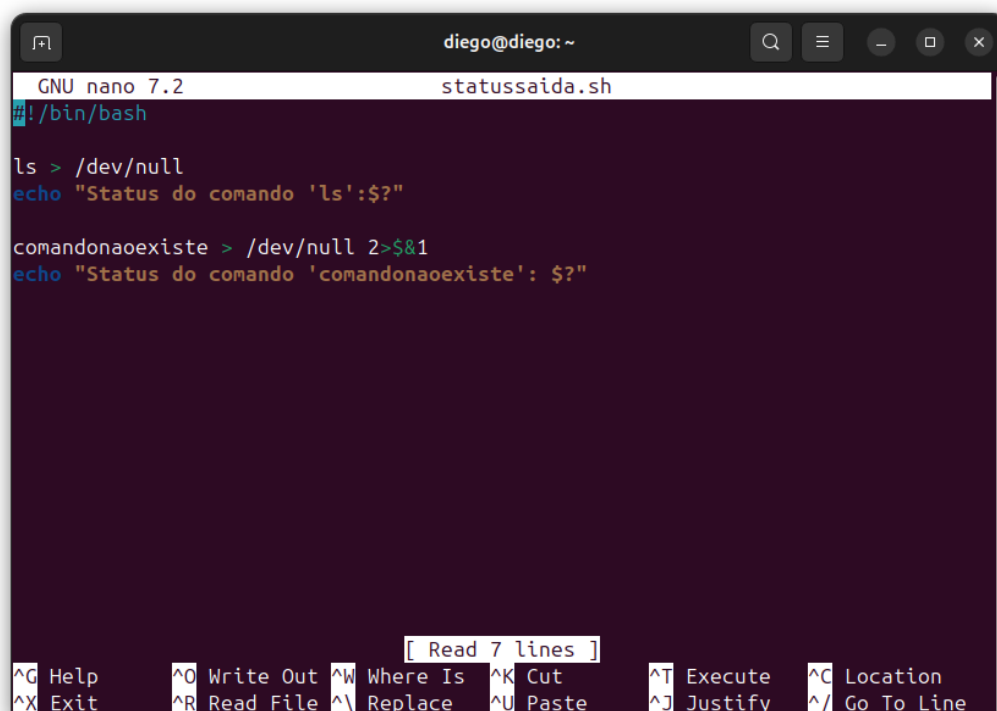
echo "Resultado da soma com 4 casas decimais: $resultado1"
echo "Resultado da divisão com 6 casas decimais: $resultado2"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```



```
diego@diego:~$ nano testebccomplexo
diego@diego:~$ chmod +x testebccomplexo
diego@diego:~$ ./testebccomplexo
Resultado da soma com 4 casas decimais: 20.2360
Resultado da divisão com 6 casas decimais: 2.142857
diego@diego:~$
```

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.



```
GNU nano 7.2 statussaida.sh
#!/bin/bash

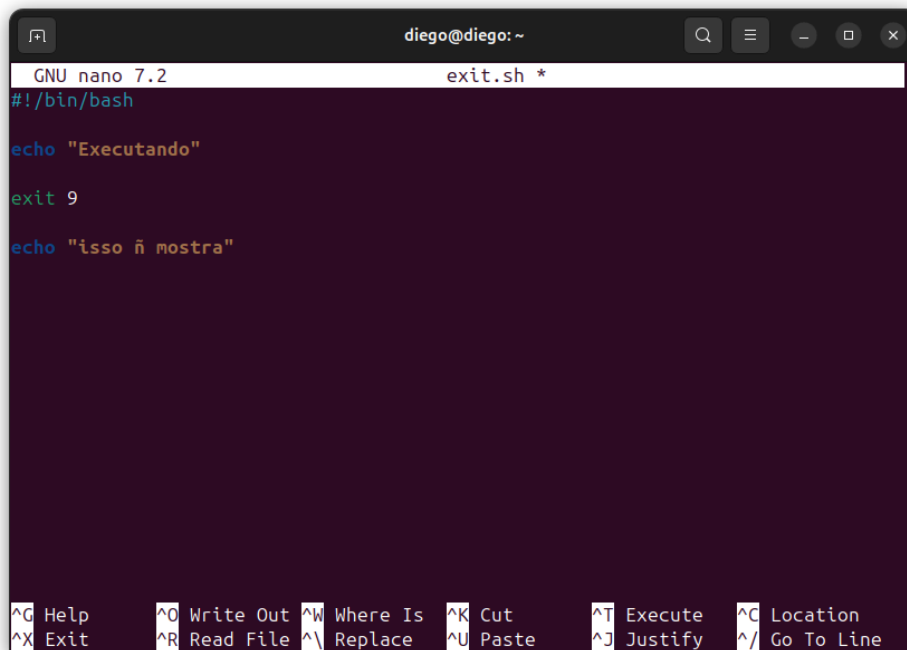
ls > /dev/null
echo "Status do comando 'ls':$?"

comandonaoexiste > /dev/null 2>$&1
echo "Status do comando 'comandonaoexiste': $?"

[ Read 7 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

```
diego@diego:~$ nano statussaida.sh
diego@diego:~$ chmod +x statussaida.sh
diego@diego:~$ ./statussaida.sh
Status do comando 'ls':0
./statussaida.sh: line 6: 1: command not found
Status do comando 'comandonaosexiste': 127
diego@diego:~$
```

- 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.



```
diego@diego: ~
GNU nano 7.2 exit.sh *
#!/bin/bash

echo "Executando"

exit 9

echo "isso ã mostra"
```

Help Write Out Where Is Cut Execute Location  
Exit Read File Replace Paste Justify Go To Line

```
diego@diego:~$ nano exit.sh
diego@diego:~$ chmod +x exit.sh
diego@diego:~$ ./exit.sh
Executando
diego@diego:~$
```

```
diego@diego: ~  
GNU nano 7.2 exit.sh *  
#!/bin/bash  
  
status=100  
  
echo "Saindo com status armazenado: $status"  
  
exit $status  
  
echo "isso ã mostra"  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
diego@diego:~$ nano exit.sh  
diego@diego:~$ ./exit.sh  
Saindo com status armazenado: 100  
diego@diego:~$
```

- 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.

```
diego@diego: ~  
GNU nano 7.2 diretorio.sh *  
#!/bin/bash  
  
DIR1="$HOME/diretorio_inexistente"  
  
if [ -d "$DIR1" ]; then  
    echo "O diretório $DIR1 existe."  
else  
    echo "O diretório $DIR1 NÃO existe."  
fi  
  
DIR2="$HOME/Downloads"  
  
if [ -d "$DIR2" ]; then  
    echo "O diretório $DIR2 existe."  
else  
    echo "O diretório $DIR2 NÃO existe."  
fi  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
diego@diego:~$ nano diretorio.sh  
diego@diego:~$ chmod +x diretorio.sh  
diego@diego:~$ ./diretorio.sh  
O diretório /home/diego/diretorio_inexistente NÃO existe.  
O diretório /home/diego/Downloads existe.  
diego@diego:~$
```

- 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.



```
GNU nano 7.2      condicionais.sh *
#!/bin/bash

VALOR1=$((10 + 5))
VALOR2=$((3 * 5))

echo "Resultado da primeira operação (10 + 5): $VALOR1"
echo "Resultado da segunda operação (3 * 5): $VALOR2"

if [ "$VALOR1" -gt "$VALOR2" ]; then
    echo "O VALOR1 ($VALOR1) é MAIOR que o VALOR2 ($VALOR2)."
elif [ "$VALOR1" -lt "$VALOR2" ]; then
    echo "O VALOR1 ($VALOR1) é MENOR que o VALOR2 ($VALOR2)."
else
    echo "O VALOR1 ($VALOR1) é IGUAL ao VALOR2 ($VALOR2)."
fi

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
diego@diego:~$ nano condicionais.sh
diego@diego:~$ chmod +x condicionais.sh
diego@diego:~$ ./condicionais.sh
Resultado da primeira operação (10 + 5): 15
Resultado da segunda operação (3 * 5): 15
O VALOR1 (15) é IGUAL ao VALOR2 (15).
diego@diego:~$
```

- 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.

```
diego@diego: ~  
GNU nano 7.2 ifelse.sh *  
#!/bin/bash  
  
STRING1="string"  
STRING2="string"  
  
echo "String 1: $STRING1"  
echo "String 2: $STRING2"  
  
if [ "$STRING1" = "$STRING2" ]; then  
    echo "As strings são IGUAIS."  
else  
    echo "As strings são DIFERENTES."  
fi  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
diego@diego:~$ nano ifelse.sh  
diego@diego:~$ chmod +x ifelse.sh  
diego@diego:~$ ./ifelse.sh  
String 1: string  
String 2: string  
As strings são IGUAIS.  
diego@diego:~$
```

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

```
diego@diego: ~  
GNU nano 7.2 fruta.sh *  
#!/bin/bash  
  
MINHA_STRING="fruta"  
  
echo "Valor da string: $MINHA_STRING"  
  
if [ "$MINHA_STRING" = "fruta" ]; then  
    echo "A string contém a palavra 'fruta'.  
else  
    echo "A string NÃO contém a palavra 'fruta'.  
fi  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

tela.

```
diego@diego:~$ nano fruta.sh
diego@diego:~$ chmod +x fruta.sh
diego@diego:~$ ./fruta.sh
Valor da string: fruta
A string contém a palavra 'fruta'.
diego@diego:~$
```

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

```
diego@diego: ~
GNU nano 7.2      DuasStrings.sh *
#!/bin/bash

STRING1=""
STRING2="exemplo"

echo "Verificando STRING1..."
if [ -z "$STRING1" ]; then
    echo "STRING1 está vazia."
else
    echo "STRING1 contém: $STRING1"
fi

echo "Verificando STRING2..."
if [ -z "$STRING2" ]; then
    echo "STRING2 está vazia."
else
    echo "STRING2 contém: $STRING2"
fi

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

os casos).

```
diego@diego:~$ nano DuasStrings.sh
diego@diego:~$ chmod +x DuasStrings.sh
diego@diego:~$ ./DuasStrings.sh
Verificando STRING1...
STRING1 está vazia.
Verificando STRING2...
STRING2 contém: exemplo
diego@diego:~$
```

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

- e : Verifica se o arquivo existe
- f : Verifica se é um arquivo regular
- d : Verifica se é um diretório
- r : Verifica se o arquivo tem permissão de leitura
- s : Verifica se o arquivo não está vazio



```
diego@diego: ~  
GNU nano 7.2 perm.sh *  
#!/bin/bash  
  
ARQUIVO="$HOME/teste.txt"  
  
if [ -e "$ARQUIVO" ]; then  
    echo "O arquivo '$ARQUIVO' existe."  
  
    if [ -r "$ARQUIVO" ]; then  
        echo "Você TEM permissão de leitura neste arquivo."  
    else  
        echo "Você NÃO TEM permissão de leitura neste arquivo."  
    fi  
else  
    echo "O arquivo '$ARQUIVO' NÃO existe."  
fi  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

```
diego@diego:~$ nano perm.sh  
diego@diego:~$ chmod +x perm.sh  
diego@diego:~$ ./perm.sh  
O arquivo '/home/diego/teste.txt' NÃO existe.  
diego@diego:~$
```