

Bartholomeu Vitor Neto  
2024.1.08.043

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?

```
a=20
b=6
c=10
d=4
resultado=$(echo "scale=2; ($a * $b) / ($c + $d)" | bc)
echo "O resultado da operação aritmética é: $resultado"
```

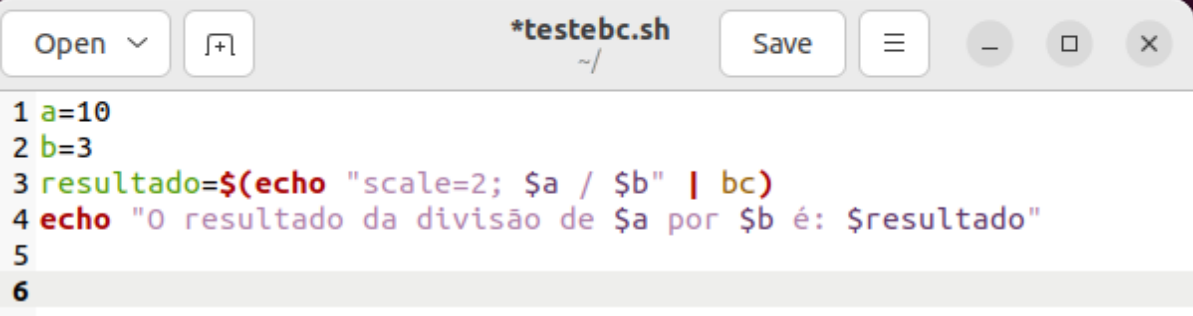
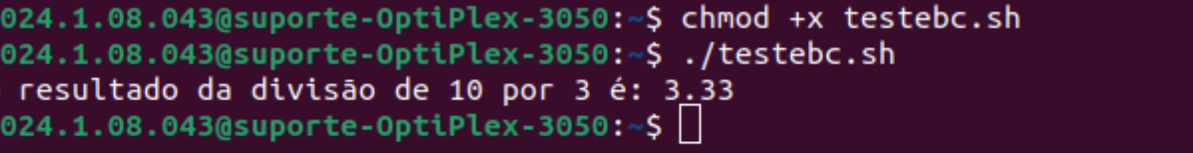
```
2024.1.08.043@suporte-OptiPlex-3050:~$ vim scriptaritmetico.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./scriptaritmetico.sh
O resultado da operação aritmética é: 8.57
```

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

```
2024.1.08.043@suporte-OptiPlex-3050:~$ 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
```

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.

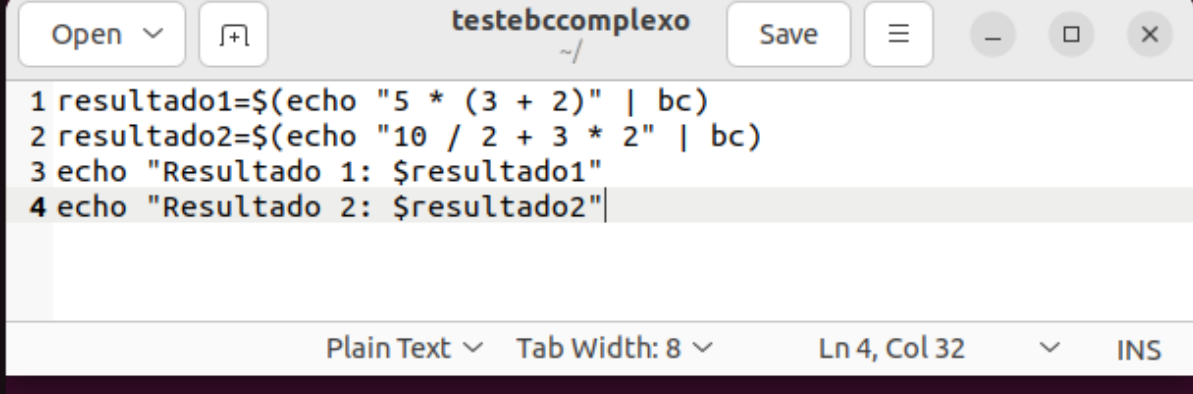
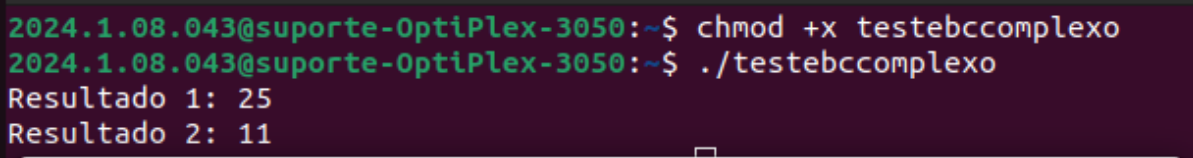
```
024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
024.1.08.043@suporte-OptiPlex-3050:~$ ./testebc.sh
resultado da divisão de 10 por 3 é: 3.33
024.1.08.043@suporte-OptiPlex-3050:~$
```



```
1 a=10
2 b=3
3 resultado=$(echo "scale=2; $a / $b" | bc)
4 echo "O resultado da divisão de $a por $b é: $resultado"
5
6
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testebccomplexo
Resultado 1: 25
Resultado 2: 11
```



```
1 resultado1=$(echo "5 * (3 + 2)" | bc)
2 resultado2=$(echo "10 / 2 + 3 * 2" | bc)
3 echo "Resultado 1: $resultado1"
4 echo "Resultado 2: $resultado2"
```

Plain Text ▾ Tab Width: 8 ▾ Ln 4, Col 32 ▾ INS

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.

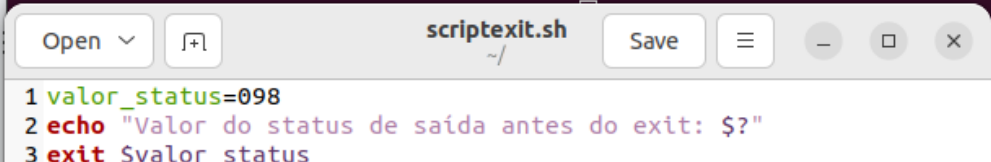
O status de saída de um programa é um valor numérico que indica se a execução do programa foi bem-sucedida ou se ocorreu algum erro durante a execução. Geralmente, um status de saída de 0 indica que o programa foi executado com sucesso, enquanto qualquer outro valor diferente de zero indica que ocorreu algum tipo de erro.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ echo "Status de saída do comando ls: $?"
Status de saída do comando ls: 0
2024.1.08.043@suporte-OptiPlex-3050:~$ xyz
echo "Status de saída do comando xyz: $?"
Command 'xyz' not found, did you mean:
  command 'xye' from deb xye (0.12.2+dfsg-11)
  command 'xz' from deb xz-utils (5.2.5-2ubuntu1)
Try: apt install <deb name>
Status de saída do comando xyz: 127
2024.1.08.043@suporte-OptiPlex-3050:~$
```

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 é usado para encerrar a execução de um script e retornar um status de saída específico para indicar se o script foi executado com sucesso ou se ocorreu algum erro.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ echo "Valor do status de saída antes do exit: $?"
Valor do status de saída antes do exit: 98
```



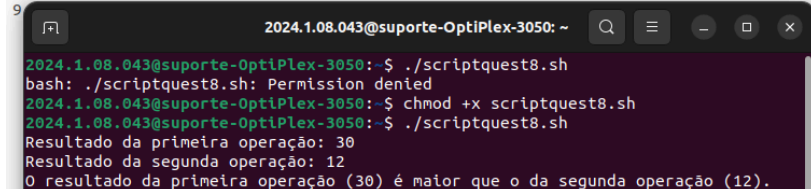
```
1 valor_status=098
2 echo "Valor do status de saída antes do exit: $?"
3 exit $valor_status
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ diretorio="sapato"
2024.1.08.043@suporte-OptiPlex-3050:~$ if [ -d "$diretorio" ]; then
    echo "O diretório $diretorio existe."
> else
> echo "O diretório $diretorio não existe."
> fi
O diretório sapato não existe.
2024.1.08.043@suporte-OptiPlex-3050:~$ diretorio="home"
2024.1.08.043@suporte-OptiPlex-3050:~$ if [ -d "$diretorio" ]; then      echo "O
diretório $diretorio existe.";
> else
> echo "O diretório $diretorio não existe."
> fi
O diretório home existe.
```

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.

```
1 op1=$((10*3))
2 op2=$((7+5))
3 echo "Resultado da primeira operação: $op1"
4 echo "Resultado da segunda operação: $op2"
5 if [ "$op1" -gt "$op2" ]; then echo "O resultado da primeira operação ($op1) é maior que o da segunda operação ($op2)."
```



```
6 elif [ "$op1" -lt "$op2" ]; then echo "O resultado da primeira operação ($op1) é menor que o da segunda operação ($op2)."
```

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.

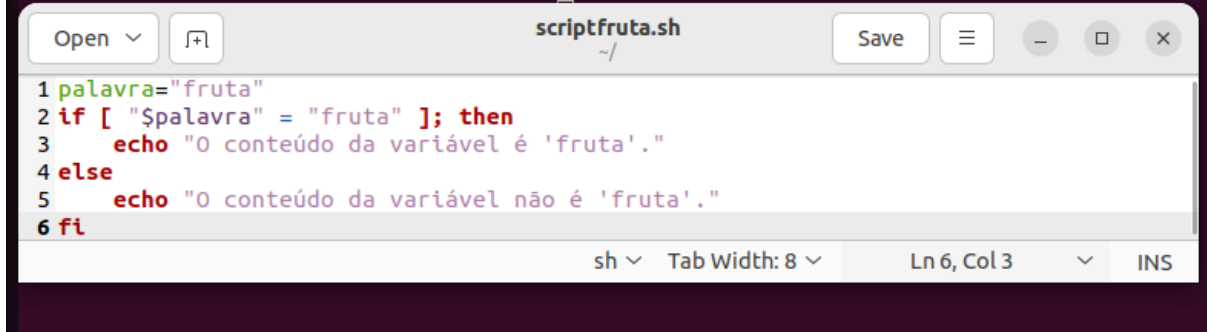
```
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x scriptstring.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./scriptstring.sh
O conteúdo das variáveis é igual.
2024.1.08.043@suporte-OptiPlex-3050:~$
```



```
1 string1="Olá, mundo!"
2 string2="Olá, mundo!"
3 if [ "$string1" = "$string2" ]; then echo "O conteúdo das variáveis é igual."
4 else echo "O conteúdo das variáveis não é igual."
5 fi
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x scriptfruta.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./scriptfruta.sh
O conteúdo da variável é 'fruta'.
```



```
1 palavra="fruta"
2 if [ "$palavra" = "fruta" ]; then
3     echo "O conteúdo da variável é 'fruta'."
4 else
5     echo "O conteúdo da variável não é 'fruta'."
6 fi
```

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).

```
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x scriptquest11.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./scriptquest11.sh
A primeira string está vazia.
A segunda string não está vazia.
```



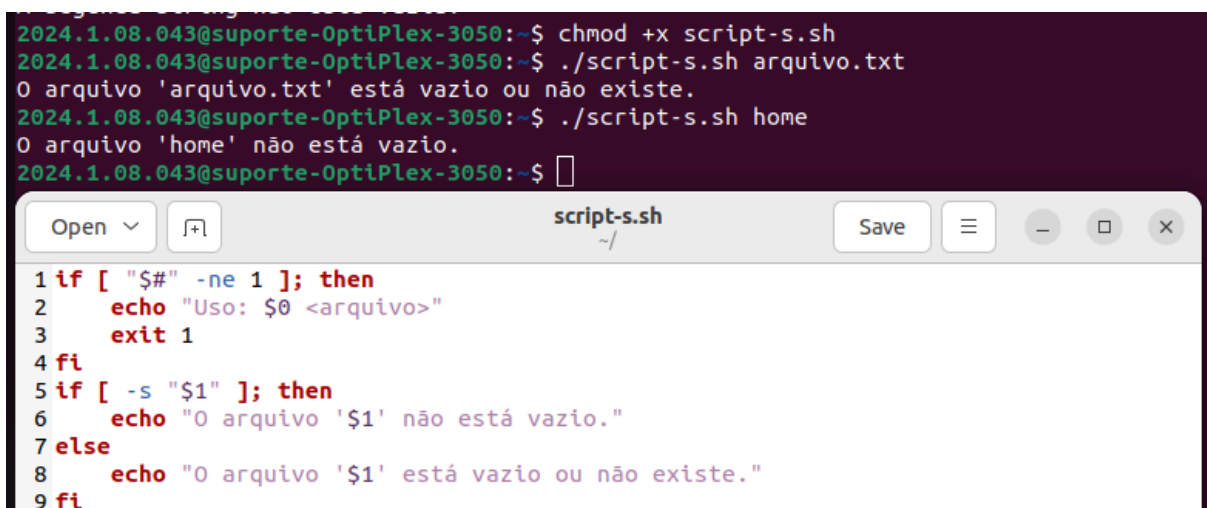
```
1 string_vazia=""
2 string_com_conteudo="Olá, mundo!"
3 if [ -n "$string_vazia" ]; then
4     echo "A primeira string não está vazia."
5 else
6     echo "A primeira string está vazia."
7 fi
8
9 if [ -n "$string_com_conteudo" ]; then
10    echo "A segunda string não está vazia."
11 else
12    echo "A segunda string está vazia."
13 fi
```

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

1. `-e` ou `-f`: Verifica se o arquivo existe.
2. `-d`: Verifica se o arquivo é um diretório.
3. `-s`: Verifica se o arquivo não está vazio.
4. `-r`: Verifica se o arquivo tem permissão de leitura.
5. `-w`: Verifica se o arquivo tem permissão de escrita.

Farei com a opção `-s`.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x script-s.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./script-s.sh arquivo.txt
0 arquivo 'arquivo.txt' está vazio ou não existe.
2024.1.08.043@suporte-OptiPlex-3050:~$ ./script-s.sh home
0 arquivo 'home' não está vazio.
2024.1.08.043@suporte-OptiPlex-3050:~$
```



```
1 if [ "$#" -ne 1 ]; then
2     echo "Uso: $0 <arquivo>"
3     exit 1
4 fi
5 if [ -s "$1" ]; then
6     echo "O arquivo '$1' não está vazio."
7 else
8     echo "O arquivo '$1' está vazio ou não existe."
9 fi
```

