



Introdução à Ciência da Computação – Lista 6

Shell script – parte 3

Nome: Isadora Resende Grandaux RA:

- 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?

```
2025.1.08.012@suporte-OptiPlex-3050:~$ bash
2025.1.08.012@suporte-OptiPlex-3050:~$ nano scriptaritmetico.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico
chmod: cannot access 'scriptaritmetico': No such file or directory
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./scriptaritmetico
bash: ./scriptaritmetico: No such file or directory
2025.1.08.012@suporte-OptiPlex-3050:~$ ./scriptaritmetico.sh
Resultado: 3.40
2025.1.08.012@suporte-OptiPlex-3050:~$
```

O comando é o `scale`, que é utilizado para definir o número de casas decimais no resultado.

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

```
2025.1.08.012@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
5/3
1.666666
quit
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ bash
2025.1.08.012@suporte-OptiPlex-3050:~$ nano testebc.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./testebc.sh
Resultado da divisão:
2.3333
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
Resultado:9
3.33
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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. ls

```
2025.1.08.012@suporte-OptiPlex-3050:~$ ls
Desktop      Music          scriptaritimico.sh  testebc.sh
Documents    NetBeansProjects snap               Videos
Downloads    Pictures       Templates          vscode
meuscript.sh Public          testebccomplexo.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ echo "status da saída:$?"
status da saída:0
2025.1.08.012@suporte-OptiPlex-3050:~$ comandoInexistente
comandoInexistente: command not found
2025.1.08.012@suporte-OptiPlex-3050:~$ echo "status da saída não existente:$?"
status da saída não existente:127
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano testeexit.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ echo $?
0
2025.1.08.012@suporte-OptiPlex-3050:~$ nano testeexit.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./testeexit.sh
Saíndo com código 7
2025.1.08.012@suporte-OptiPlex-3050:~$ echo $?
7
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano scriptsimples.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x scriptsimples.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./scriptsimples.sh
./scriptsimples.sh: line 1: {-d: command not found
Diretório não existe
2025.1.08.012@suporte-OptiPlex-3050:~$ nano scriptsimples.sh
```

- 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.

```
a=$((5 + 3))
b=$((2 * 4))

if [ $a -gt $b ]; then
echo "$a é maior que $b"
elif [ $a -lt $b ]; then
echo "$a é menor que $b"
else
echo "$a é igual a $b"
fi
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano condicionais.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x condicionais.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./condicionais.sh
./condicionais.sh: line 6: [8: command not found
8 é igual a 8
2025.1.08.012@suporte-OptiPlex-3050:~$ nano condicionais.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./condicionais.sh
./condicionais.sh: line 6: [8: command not found
8 é igual a 8
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
GNU nano 6.2                                string.sh *
str1="casa"
str2="carro"

if [ "$str1" = "$str2" ]; then
echo "As strings são iguais."
else
echo "As strings são diferentes."
fi
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano string.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x string.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./string.sh
As strings são diferentes.
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano string.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x string.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./string.sh
As strings são diferentes.
2025.1.08.012@suporte-OptiPlex-3050:~$ nano fruta.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x fruta.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./fruta.sh
./fruta.sh: line 3: [fruta: command not found
A palavra não é fruta
2025.1.08.012@suporte-OptiPlex-3050:~$ nano fruta.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./fruta.sh
A palavra é fruta.
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 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).

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano string2.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x string2.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./string2.sh
./string2.sh: line 23: syntax error: unexpected end of file
2025.1.08.012@suporte-OptiPlex-3050:~$ nano string2.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./string2.sh
str1 está vazia
str2 tem conteúdo
2025.1.08.012@suporte-OptiPlex-3050:~$
```

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 permissão de leitura;
- s → verifica se o arquivo não está vazio;

```
GNU nano 6.2                                opcoes.sh *
#!/bin/bash
ARQUIVO="$HOME/.bashrc"

if [ -f "$ARQUIVO" ]; then
    echo "Arquivo $ARQUIVO existe e é um arquivo regular"
else
    echo "Arquivo não existe ou não é regular"
fi
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano opcoes.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod +x opcoes.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./opcoes.sh
Arquivo /home/2025.1.08.012/.bashrc existe e é um arquivo regular
2025.1.08.012@suporte-OptiPlex-3050:~$
```