



Introdução à Ciência da Computação – Lista 6

Shell script – parte 3

Nome: RA:

- 1) Crie um script chamado script aritmético, 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?

```
2025.1.08.025@suporte-OptiPlex-3050:~$ nano script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./script_aritmetico.sh
./script_aritmetico.sh: line 11: unexpected EOF while looking for matching `"'
./script_aritmetico.sh: line 12: syntax error: unexpected end of file
2025.1.08.025@suporte-OptiPlex-3050:~$ nano script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./script_aritmetico.sh
Resultado da operação: .63
2025.1.08.025@suporte-OptiPlex-3050:~$
```

```
GNU nano 6.2
#!/bin/bash

a=3
b=4
c=5
d=6

resultado=$(echo "scale=2; ($a + $b) / ($c + $d)" | bc)

echo "Resultado da operação: $resultado"
```

R: O recurso que se utiliza é o comando que se utiliza para o valor não inteiro aparecer no resultado, é o bc mais o scale.

R: A variável utilizada é “resultado”.

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

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$
2025.1.08.025@suporte-OptiPlex-3050:~$ nano script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./script_aritmetico.sh
Resultado da operação: .636363
2025.1.08.025@suporte-OptiPlex-3050:~$

```

```

GNU nano 6.2
#!/bin/bash

a=3
b=4
c=5
d=6

resultado=$(echo "scale=6; ($a + $b) / ($c + $d)"|bc)

echo "Resultado da operação: $resultado"

```

- 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.

```

Resultado da operação: .636363
2025.1.08.025@suporte-OptiPlex-3050:~$ nano script_aritmetico.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebc.sh
Runtime error (func=(main), adr=14): Divide by zero
./testebc.sh: line 8: syntax error near unexpected token `"0 resultado da expressao é: $resultado"'
./testebc.sh: line 8: `echo ("0 resultado da expressao é: $resultado")'
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebc.sh
Runtime error (func=(main), adr=14): Divide by zero
0 resultado da expressao é:
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebc.sh
0 resultado da expressao é: 1.73
2025.1.08.025@suporte-OptiPlex-3050:~$

```

```
a=17
b=23

resultado=$(echo "scale = 2; ($a+$b)/$b"|bc)

echo "O resultado da expressao é: $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.

```
#!/bin/bash

read resultado1, resultado2 <<EOF
$(bc<< CALC scale=2
a=36/5
b=12 + 13
print a," ", b
CALC
)
EOF

echo "o resultado de 36/5 é: $resultado1"
echo "o resultado de 12+13 é: $resultado2"
```

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testecomplexo.sh
chmod: cannot access 'testecomplexo.sh': No such file or directory
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
bash: ./testebccomplexo.sh: No such file or directory
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
File scale=2 is unavailable.
./testebccomplexo.sh: line 4: read: `resultado1,': not a valid identifier
o resultado de 36/5 é:
o resultado de 12+13 é:
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
File scale=2 is unavailable.
o resultado de 36/5 é:
o resultado de 12+13 é:
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
o resultado de 36/5 é: 7.20
o resultado de 12+13 é: 25
2025.1.08.025@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.

R: O status de saída é um inteiro entre 0 e 255.

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testecomplexo.sh
chmod: cannot access 'testecomplexo.sh': No such file or directory
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
bash: ./testebccomplexo.sh: No such file or directory
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
File scale=2 is unavailable.
./testebccomplexo.sh: line 4: read: `resultado1,': not a valid identifier
o resultado de 36/5 é:
o resultado de 12+13 é:
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
File scale=2 is unavailable.
o resultado de 36/5 é:
o resultado de 12+13 é:
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
o resultado de 36/5 é: 7.20
o resultado de 12+13 é: 25
2025.1.08.025@suporte-OptiPlex-3050:~$ echo $?
0
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh

```

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testebccomplexo.sh
./testebccomplexo.sh: line 15: unexpected EOF while looking for matching `"'
./testebccomplexo.sh: line 16: syntax error: unexpected end of file
2025.1.08.025@suporte-OptiPlex-3050:~$ echo $?
2
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testebccomplexo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ nano testesaida1.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x testesaida1.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./testesaida1.sh
./testesaida1.sh: line 6: comandox: command not found
Status de saída do comando inválido: 127
2025.1.08.025@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.

R: O comando exit em scripts shell (bash) é usado para encerrar a execução do script e retornar um status de saída ao sistema operacional ou ao processo que chamou o script. Esse status é um número de 0 a 255

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./exit.sh
Executando o primeiro bloco...
2025.1.08.025@suporte-OptiPlex-3050:~$ ./exit.sh
Executando o primeiro bloco...
2025.1.08.025@suporte-OptiPlex-3050:~$ nano exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./exit.sh
Executando o primeiro bloco...
2025.1.08.025@suporte-OptiPlex-3050:~$ bash exit.sh
Executando o primeiro bloco...
2025.1.08.025@suporte-OptiPlex-3050:~$ echo $?
42
2025.1.08.025@suporte-OptiPlex-3050:~$

```

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano exitvariavel.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ nano exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x exit.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ bash exit.sh
Executando o segundo bloco...
2025.1.08.025@suporte-OptiPlex-3050:~$ echo $?
7
2025.1.08.025@suporte-OptiPlex-3050:~$

```

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.

```
2025.1.08.025@suporte-OptiPlex-3050:~$ nano ver_diretorio.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod ver_diretorio.sh
chmod: missing operand after 'ver_diretorio.sh'
Try 'chmod --help' for more information.
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x ver_diretorio.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./ver_diretorio.sh
❌ O diretório '/home/2025.1.08.025/meu_diretorio_teste' NÃO existe.
2025.1.08.025@suporte-OptiPlex-3050:~$ rm -rf ~/meu_diretorio_teste
2025.1.08.025@suporte-OptiPlex-3050:~$ mkdir ~/testediretorio
2025.1.08.025@suporte-OptiPlex-3050:~$ nano ver_diretorio.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x ver_diretorio.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./ ver_diretorio.sh
bash: ./: Is a directory
2025.1.08.025@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.025@suporte-OptiPlex-3050:~$ nano comparar_operacoes.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x comparar_operacoes.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./comparar_operacoes.sh
Resultado da primeira operação: 17
Resultado da segunda operação: 18
O primeiro valor (17) é MENOR que o segundo (18).
2025.1.08.025@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.

```
O primeiro valor (17) é menor que o segundo (18).
2025.1.08.025@suporte-OptiPlex-3050:~$ nano comparar_strings.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x comparar_strings.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./comparar_strings.sh
Valor da STRING1: ChatGPT
Valor da STRING2: chatgpt
As strings são DIFERENTES.
2025.1.08.025@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.

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano verifica.sh
2025.1.08.025@suporte-OptiPlex-3050:~$
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x verifica.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./verifica.sh
Valor da variável: fruta
A variável contém a palavra 'fruta'.
2025.1.08.025@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).

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano verifica_conteudo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x verifica_conteudo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./verifica_conteudo.sh
Verificando a STRING_VAZIA...
STRING_VAZIA está vazia.

Verificando a STRING_COM_CONTEUDO...
STRING_COM_CONTEUDO tem conteúdo: ICC
2025.1.08.025@suporte-OptiPlex-3050:~$

```

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

5 opções:

Se é um arquivo comum

Se o arquivo existe

Se o arquivo é um diretório

Se o arquivo é de leitura

Se o arquivo não está vazio

```

2025.1.08.025@suporte-OptiPlex-3050:~$ nano verificaarquivo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ chmod +x verificaarquivo.sh
2025.1.08.025@suporte-OptiPlex-3050:~$ ./verificaarquivo.sh
Verificando o arquivo: /home/2025.1.08.025/exemplo.txt
O arquivo não existe ou está vazio.
2025.1.08.025@suporte-OptiPlex-3050:~$

```