



Introdução à Ciência da Computação – Lista 6

Shell script – parte 3

Nome: João Antônio de Souza Martins

RA: 2025.1.08.013

- 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?

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat scriptaritmetico.sh
#!/bin/bash

a=0
b=9999
c=22
d=7

e=$((a * b + c / d))

echo "Resultado = $e"
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./scriptaritmetico.sh
Resultado = 3
```

É possível salvar o resultado em uma variável 'e' e, usando o "scale=n", é possível mostrar um valor com resultado decimal, em que n é a quantidade de casas decimais que serão exibidas.

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

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat scriptaritmetico.sh
#!/bin/bash

a=0
b=9999
c=22
d=7

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

echo "Resultado = $e"
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./scriptaritmetico.sh
Resultado = 3.142857
```

- 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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat testebc.sh
#!/bin/bash

a=22
b=7
decimais=2

echo "scale=$decimais; $a / $b" | bc
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x testebc.sh
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./testebc.sh
3.14
```

- 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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > testebccomplexo
#!/bin/bash
numA=250
numB=7
numC=5
resultado=`bc << FIM
scale=4
x=($numA / $numB)
y=(x * $numC)
y
FIM
`
echo "Resultado: $resultado"
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x testebccomplexo
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./testebccomplexo
Resultado: 178.5710
```

- 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.

É um número inteiro que o programa retorna como um resultado específico ao sistema operacional ao terminar sua execução.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ls
scriptaritmetico.sh testebccomplexo testebccomplexo.sh testebc.sh
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ echo $?
0
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ soumpato
soumpato: command not found
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ echo $?
127
```

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 encerra a execução do script imediatamente.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > script
echo "olá"
exit 5
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x script
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./script
olá
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ echo $?
5
```

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > numerosortido
exit 44
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x numerosortido
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./numerosortido
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ echo $?
44
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > exitvariavel
var=88
exit $var
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x exitvariavel
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./variavel
bash: ./variavel: No such file or directory
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./exitvariavel
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ echo $?
88
```

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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ if ls /home/2025.1.08.013/ForShell
> then
> echo "Diretório existente"
else
> echo "Diretório não existente"
> fi
exitvariavel    script    testebccomplexo    testebc.sh
numerosortido  scriptaritmetico.sh  testebccomplexo.sh
Diretório existente
```

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ if ls /home/2025.1.08.013/Unicornio
> then
> echo "Diretório existente"
> else
> echo "Diretório não existente"
> fi
ls: cannot access '/home/2025.1.08.013/Unicornio': No such file or directory
Diretório não existente
```

- 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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > comparision
#!/bin/bash

# Primeira operação: soma
a=$((10 + 5))

# Segunda operação: multiplicação
b=$((3 * 5))

echo "Resultado da primeira operação: $a"
echo "Resultado da segunda operação: $b"

# Estrutura condicional
if [ $a -gt $b ]; then
    echo "A primeira operação é MAIOR que a segunda."
elif [ $a -lt $b ]; then
    echo "A primeira operação é MENOR que a segunda."
else
    echo "As duas operações são IGUAIS."
fi

2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x comparision
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./comparision
Resultado da primeira operação: 15
Resultado da segunda operação: 15
As duas operações são IGUAIS.
```

- 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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > strings
#!/bin/bash

# Variáveis string
str1="goku"
str2="dinossauro"

echo "String 1: $str1"
echo "String 2: $str2"

# Estrutura condicional
if [ "$str1" = "$str2" ]; then
    echo "As strings são IGUAIS."
else
    echo "As strings são DIFERENTES."
fi

2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x strings
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./strings
String 1: goku
String 2: dinossauro
As strings são DIFERENTES.
```

- 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.

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > frutayn
#!/bin/bash

str="priacanjuba"

echo "Valor da string: $str"

if [ "$str" = "fruta" ]; then
    echo "A string é 'fruta'."
else
    echo "A string NÃO é 'fruta'."
fi

2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x frutayn
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./frutayn
Valor da string: priacanjuba
A string NÃO 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).

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ cat > vazio
#!/bin/bash

str1=""
str2="CHEIA"

# Verifica str1
if [ -z "$str1" ]; then
    echo "str1 está VAZIA."
else
    echo "str1 tem CONTEÚDO."
fi

# Verifica str2
if [ -z "$str2" ]; then
    echo "str2 está VAZIA."
else
    echo "str2 tem CONTEÚDO."
fi

2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ chmod +x vazio
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ ./vazio
str1 está VAZIA.
str2 tem CONTEÚDO.
```

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

Cinco dessas opções são: a opção -e, que verifica se o arquivo existe; a opção -f, que verifica se o caminho corresponde a um arquivo comum; a opção -d, que verifica se o caminho é um diretório; a opção -r, que verifica se o arquivo é legível pelo usuário; e a opção -w, que verifica se o arquivo é gravável

```
2025.1.08.013@suporte-OptiPlex-3050:~/ForShell$ #!/bin/bash

arquivo="vazio"

if [ -e "$arquivo" ]; then
    echo "O arquivo '$arquivo' EXISTE."
else
    echo "O arquivo '$arquivo' NÃO existe."
fi

O arquivo 'vazio' EXISTE.
```