



Introdução à Ciência da Computação – Lista 6

Shell script – parte 3

Nome: Filipe Bessa Carvalho

RA: 2024.1.08.009

- 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?

```
1 #!/bin/bash
2 var1=13
3 var2=34
4 var3=2
5 var4=3
6 resultado=`bc << TRUNCA
7 scale=2
8 a=($var1*$var2)
9 b=($var3*$var4)
10 a/b
11 TRUNCA
12 `
13 echo "Resultado: $resultado"
```

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ gedit scriptaritmetico.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ chmod 755 scriptaritmetico.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./scriptaritmetico.sh
Resultado: 73.66
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

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

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ 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.

```
1 #!/bin/bash
2 var1=10
3 var2=3
4 resultado=`echo "scale=3; $var1/$var2" | bc`
5 echo "Resultado: $resultado"
```

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ gedit testebc.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ chmod 755 testebc.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./testebc.sh
Resultado: 3.333
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

- 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.

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ gedit testebccomplexo.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./testebccomplexo.sh
Produto sobre soma: 5
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

```
1 #!/bin/bash
2 var1=12
3 var2=10
4 r='bc << TRUNCA
5 produto=$var1*$var2
6 soma=$var1+$var2
7 produto/soma
8 TRUNCA
9 `
10 echo "Produto sobre soma: $r"|
```

- 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 indica para o shell que o comando terminou e se teve sucesso.

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ls
calculo.sh  logusuario.sh      testebccomplexo.sh  testevariaveisambiente.sh
cdlsa.sh    meimprima.txt      testebc.sh          testevariaveis.sh
log.txt     scriptaritmetico.sh testecrashes.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ echo $?
0
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ echo ^C
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ echo $?
130
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

- 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 permite a alteração do status de saída

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./testesaida.sh
Digite cifrão + ? para ver o status de saída
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ echo $?
12
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$
```

```

1 #!/bin/bash
2 echo "Digite cifrao + ? para ver o status de saida"
3 exit 12

```

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ gedit testesaida.sh
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./testesaida.sh
Digite cifrao + ? para ver o status de saida
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ echo $?
5
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$

```

```

1 #!/bin/bash
2 echo "Digite cifrao + ? para ver o status de saida"
3 var1=34
4 var2=12
5 r=`bc << TRUNCA
6 a=($var1/2)
7 a-$var2
8 TRUNCA
9 `
10 exit $r

```

- 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.

```

2024.1.08.009@suporte-OptiPlex-3050:~$ ./intcc/scripts/dircond.sh
./intcc/scripts/dircond.sh: line 2: cd: /home/2024.1.08.009/ABABAB: No such file or directory
Diretório inexistente
Diretório intcc encontrado
2024.1.08.009@suporte-OptiPlex-3050:~$

```

```

1 #!/bin/bash
2 if cd ~/ABABAB
3 then
4     echo "Diretório ABABAB encontrado"
5 else
6     echo "Diretório inexistente"
7 fi
8 if cd ~/intcc
9 then
10    echo "Diretório intcc encontrado"
11 else
12    echo "Diretório inexistente"
13 fi

```

- 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.

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./aritcond.sh
0 produto é igual à soma
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$

```

```

1 #!/bin/bash
2 prod=$((2*2))
3 soma=$((2+2))
4 if [ $prod -gt $soma ]
5 then
6     echo "O produto é maior que a soma"
7 elif [ $prod -lt $soma ]
8 then
9     echo "O produto é menor que a soma"
10 else
11     echo "O produto é igual à soma"
12 fi

```

- 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.

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./strcond.sh
./strcond.sh: line 4: test: too many arguments
DIFERENTES
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$

```

```

1 #!/bin/bash
2 str1="Eu sou uma string"
3 str2="Eu sou uma string diferente"
4 if test $str1 = $str2
5 then
6     echo "IGUAIS"
7 else
8     echo "DIFERENTES"
9 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.

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./strcond.sh
A string é Fruta
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$

```

```

1 #!/bin/bash
2 str1=Fruta
3 if [ $str1 = Fruta ]
4 then
5     echo "A string é Fruta"
6 else
7     echo "A string não é Fruta"
8 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).

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./strcond.sh
str1 é vazia
str2 não é vazia
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$

```

```

1 #!/bin/bash
2 str1=''
3 str2=NÃOVAZIA
4 if [ -z $str1 ]
5 then
6     echo "str1 é vazia"
7 else
8     echo "str1 não é vazia"
9 fi
10 if [ -n $str2 ]
11 then
12     echo "str2 não é vazia"
13 else
14     echo "str2 é vazia"
15 fi

```

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

-d verifica se existe e é diretório

-f verifica se existe e é arquivo

-e verifica se existe

-nt verifica se o primeiro é mais novo que o segundo

-ot verifica se o primeiro é mais velho que o segundo

```

1 #!/bin/bash
2 if [ -d ~/intcc ]
3 then
4     echo "intcc existe e é diretório"
5 else
6     echo "intcc não existe ou não é diretório"
7 fi

```

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/scripts$ ./arqcond.sh
intcc existe e é diretório

```