



Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: Gustavo Andrade M. de Assis RA: 2024.1.08.012

- 1) Crie um script chamado escrevenome, faça com que a saída desse script seja seu nome completo. Não utilize o comando chmod. Depois crie um script chamado testecompara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevenome. Mostre o resultado da saída.

```
GNU nano 6.2      escrevanome1.sh
#!/bin/bash

echo "Gustavo Andrade Moreira de Assis"

GNU nano 6.2      testecompara
#!/bin/bash

if [ -r escrevanome1 ] && [ -x escrevanome1 ]; then
echo "O usuário tem permissão de leitura e execução sobre o script escrevanome1."
else
echo "O usuário não tem permissão de leitura e execução sobre o script escrevanome1."
fi

[ Escritas 7 linhas ]
^G Ajuda      ^O Gravar      ^W Onde está?  ^K Recortar    ^T Executar    ^C Local
^X Sair        ^R Ler o arq   ^_ Substituir  ^U Colar       ^J Justificar  ^/ Ir p/ linha
```

```
2024.1.08.012@OptiPlex-3000: ~  
2024.1.08.012@OptiPlex-3000:~$ ./testecompara  
bash: ./testecompara: Permissão negada  
2024.1.08.012@OptiPlex-3000:~$
```

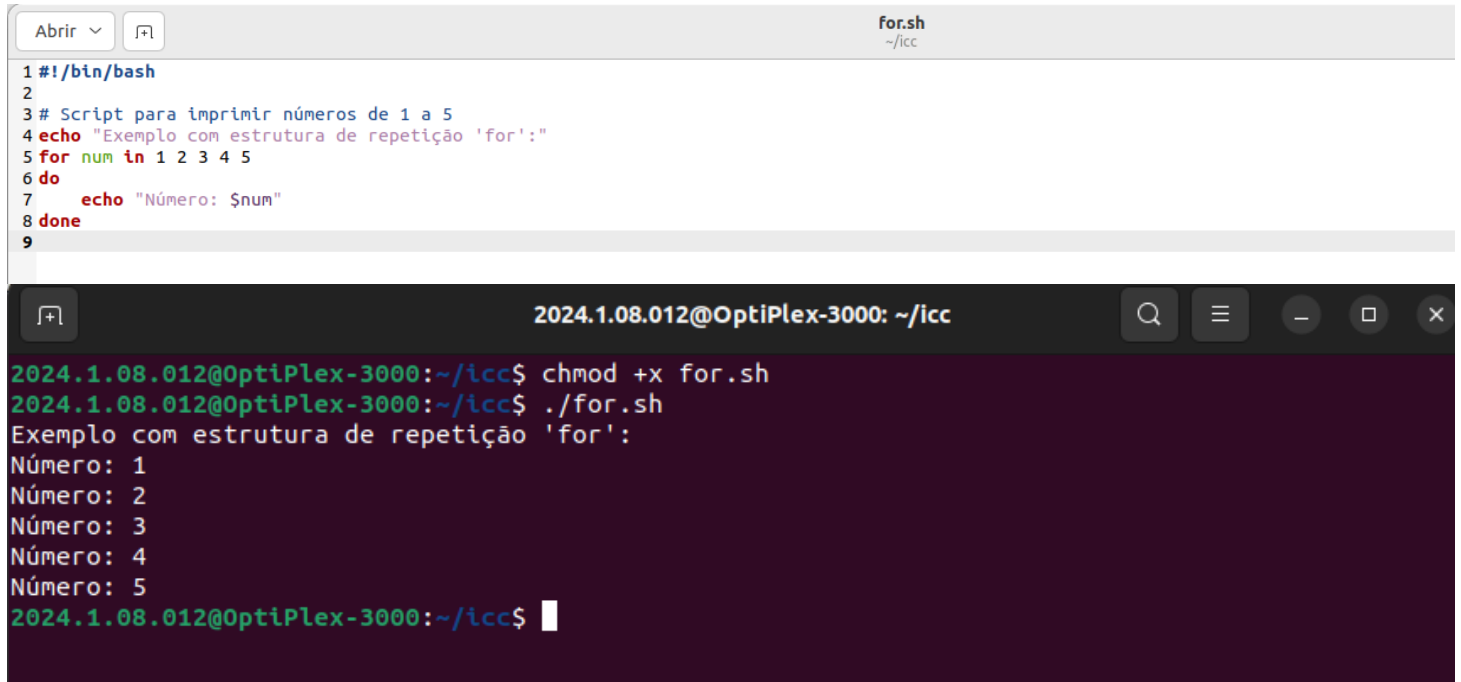
- 2) Crie um script chamado frutascase. Com base no valor da variável fruta mostre uma breve descrição da fruta. Faça com 5 frutas. Exemplo: fruta=uva, echo “A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, musses, geléias, tortas, gelatinas, sucos.”

```
frutascase.sh  
~/.icc  
1 #!/bin/bash  
2  
3  
4 fruta="$1"  
5  
6  
7 case $fruta in  
8     "uva")  
9         echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação  
humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de  
produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, musses, geléias, tortas, gelatinas, sucos."  
10     ;;  
11     "maçã")  
12         echo "A maçã é o fruto da macieira, uma árvore do gênero Malus. É uma das frutas mais cultivadas do mundo. Existem milhares de variedades de maçãs, com  
diferentes cores, sabores e texturas. A maçã é conhecida por ser uma fonte de fibras, vitaminas e antioxidantes. Ela pode ser consumida fresca, cozida, em sucos,  
compotas, tortas, saladas, entre outras preparações."  
13     ;;  
14     "banana")  
15         echo "A banana é o fruto da bananeira, uma planta herbácea da família das Musaceae. Originária do sudeste asiático, a banana é uma das frutas mais consumidas  
no mundo. Ela é uma excelente fonte de potássio, vitaminas e fibras. A banana pode ser consumida fresca, cozida, assada, em sobremesas, saladas, smoothies, entre  
outras preparações."  
16     ;;  
17     "laranja")  
18         echo "A laranja é o fruto da laranjeira, uma árvore do gênero Citrus. Ela é conhecida por ser uma excelente fonte de vitamina C e antioxidantes. Existem  
diversas variedades de laranja, com diferentes cores e sabores. A laranja pode ser consumida fresca, em sucos, sobremesas, saladas, molhos, entre outras  
preparações."  
19     ;;  
20     "morango")  
21         echo "O morango é uma pequena fruta vermelha originária da Europa. Ele é conhecido por seu sabor doce e aroma característico. Os morangos são uma excelente  
fonte de vitamina C, fibras e antioxidantes. Eles podem ser consumidos frescos, em sobremesas, saladas, sucos, geléias, entre outras preparações."  
22     ;;  
23     *)  
24         echo "Não tenho informações sobre essa fruta."  
25     ;;  
26 esac  
27
```

```
2024.1.08.012@OptiPlex-3000: ~/.icc  
2024.1.08.012@OptiPlex-3000:~/.icc$ ./frutascase.sh uva  
A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades  
da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classif  
icada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura  
ou usada na preparação de doce, vinho, passas, musses, geléias, tortas, gelatinas, sucos.  
2024.1.08.012@OptiPlex-3000:~/.icc$
```

3) Cite, explique e faça um script simples para cada estrutura de repetição do shell bash. Use sua criatividade para os scripts.

FOR :



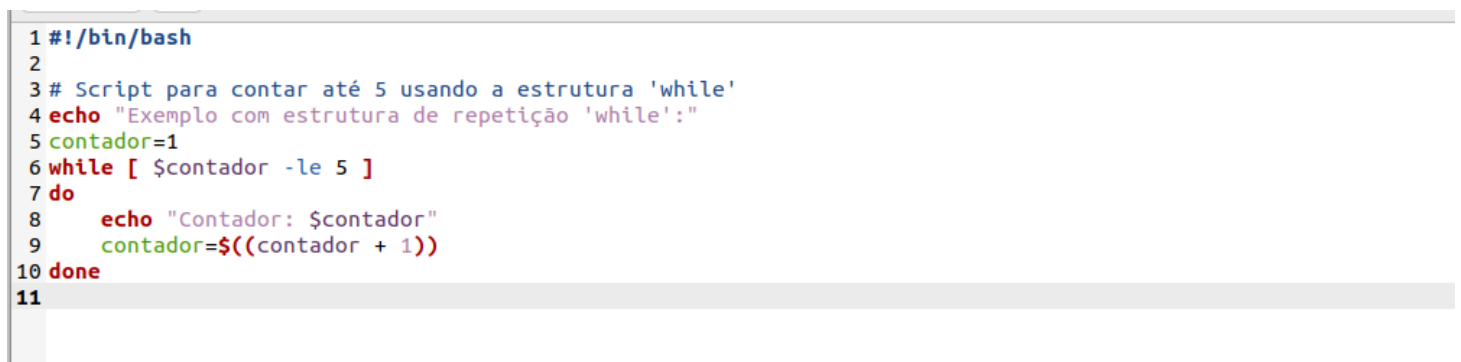
The screenshot shows a terminal window with a title bar containing 'Abrir', a file icon, and 'for.sh ~/.icc'. The terminal content is as follows:

```
1 #!/bin/bash
2
3 # Script para imprimir números de 1 a 5
4 echo "Exemplo com estrutura de repetição 'for':"
5 for num in 1 2 3 4 5
6 do
7     echo "Número: $num"
8 done
9
```

Below the script editor, the terminal execution is shown with a dark background. The prompt is '2024.1.08.012@OptiPlex-3000: ~/.icc\$'. The user enters 'chmod +x for.sh' and then './for.sh'. The output is:

```
2024.1.08.012@OptiPlex-3000:~/.icc$ chmod +x for.sh
2024.1.08.012@OptiPlex-3000:~/.icc$ ./for.sh
Exemplo com estrutura de repetição 'for':
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
2024.1.08.012@OptiPlex-3000:~/.icc$
```

WHILE:



The screenshot shows a terminal window with a title bar containing 'Abrir', a file icon, and 'for.sh ~/.icc'. The terminal content is as follows:

```
1 #!/bin/bash
2
3 # Script para contar até 5 usando a estrutura 'while'
4 echo "Exemplo com estrutura de repetição 'while':"
5 contador=1
6 while [ $contador -le 5 ]
7 do
8     echo "Contador: $contador"
9     contador=$((contador + 1))
10 done
11
```

```
2024.1.08.012@OptiPlex-3000: ~/icc
2024.1.08.012@OptiPlex-3000:~/icc$ ./while.sh
Exemplo com estrutura de repetição 'while':
Contador: 1
Contador: 2
Contador: 3
Contador: 4
Contador: 5
2024.1.08.012@OptiPlex-3000:~/icc$
```

UNTIL :

```
1 #!/bin/bash
2
3 # Script para contar até 5 usando a estrutura 'until'
4 echo "Exemplo com estrutura de repetição 'until':"
5 contador=1
6 until [ $contador -gt 5 ]
7 do
8     echo "Contador: $contador"
9     contador=$((contador + 1))
10 done
11
```

```
2024.1.08.012@OptiPlex-3000:~/icc$ chmod +x until.sh
2024.1.08.012@OptiPlex-3000:~/icc$ ./until.sh
Exemplo com estrutura de repetição 'until':
Contador: 1
Contador: 2
Contador: 3
Contador: 4
Contador: 5
2024.1.08.012@OptiPlex-3000:~/icc$
```

4) Explique o que é IFS e faça um script diferente do que foi visto em aula. Use sua criatividade.

O IFS (Internal Field Separator) é uma variável de ambiente utilizada no shell bash para determinar como os campos (ou partes) de uma string são separados quando esta é dividida em partes menores. Por padrão, o valor de IFS é configurado para espaço em branco (espaço, tabulação e nova linha), mas pode ser modificado para qualquer caractere desejado

```
1 #!/bin/bash
2
3
4 entrada="maçã,banana,uva,morango"
5
6
7 IFS_antigo=$IFS
8
9
10 IFS=', '
11
12
13 echo "Frutas:"
14 for fruta in $entrada; do
15     echo "- $fruta"
16 done
17
18 IFS=$IFS_antigo
19
```

A terminal window with a dark purple background and white text. The title bar shows '2024.1.08.012@OptiPlex-3000: ~/lcc'. The terminal shows the execution of a script named 'ifs.sh'. The first command is 'chmod +x ifs.sh' and the second is './ifs.sh'. The output of the script is 'Frutas:' followed by a list of fruits: '- maçã', '- banana', '- uva', and '- morango'. The prompt '2024.1.08.012@OptiPlex-3000:~/lcc\$' is visible at the bottom.

```
2024.1.08.012@OptiPlex-3000: ~/lcc
2024.1.08.012@OptiPlex-3000:~/lcc$ chmod +x ifs.sh
2024.1.08.012@OptiPlex-3000:~/lcc$ ./ifs.sh
Frutas:
- maçã
- banana
- uva
- morango
2024.1.08.012@OptiPlex-3000:~/lcc$
```

5) Crie um script for no estilo C que mostre na tela os números de 50 a 20.

```
2024.1.08.012@OptiPlex-3000: ~/icc
GNU nano 6.2 contagem
#!/bin/bash

echo " Contagem de 50 a 20:"

for ((i = 50; i >= 20; i--))
do
echo $i
done
```

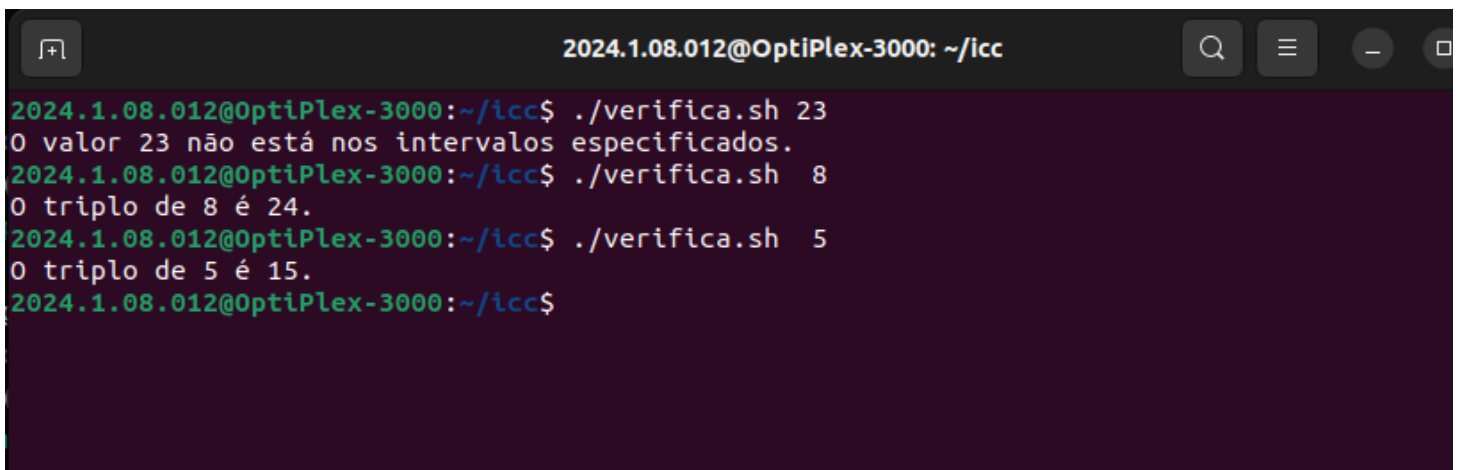
[Escritas 8 linhas]

^G Ajuda	^O Gravar	^W Onde está?	^K Recortar	^T Executar	^C Local
^X Sair	^R Ler o arq	^_ Substituir	^U Colar	^J Justificar	^_ Ir p/ linha

```
2024.1.08.012@OptiPlex-3000: ~/icc
2024.1.08.012@OptiPlex-3000:~/icc$ chmod +x contagem
2024.1.08.012@OptiPlex-3000:~/icc$ ./contagem
Contagem de 50 a 20:
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
2024.1.08.012@OptiPlex-3000:~/icc$
```

- 6) Desenvolva um script que receba um parâmetro e verifique se o valor está entre 0 e 10. Caso sim mostre o triplo do valor. Caso ele esteja entre 10 e 20 mostre o dobro. Caso não esteja nos anteriores apresente uma mensagem.

```
1 #!/bin/bash
2
3 if [ $# -ne 1 ]; then
4     echo "Uso: $0 <valor>"
5     exit 1
6 fi
7
8 valor=$1
9
10 if [ $valor -ge 0 ] && [ $valor -le 10 ]; then
11     # Se sim, mostra o triplo do valor
12     resultado=$((valor * 3))
13     echo "O triplo de $valor é $resultado."
14
15 elif [ $valor -gt 10 ] && [ $valor -le 20 ]; then
16     # Se sim, mostra o dobro do valor
17     resultado=$((valor * 2))
18     echo "O dobro de $valor é $resultado."
19 else
20
21     echo "O valor $valor não está nos intervalos especificados."
22 fi
23
```

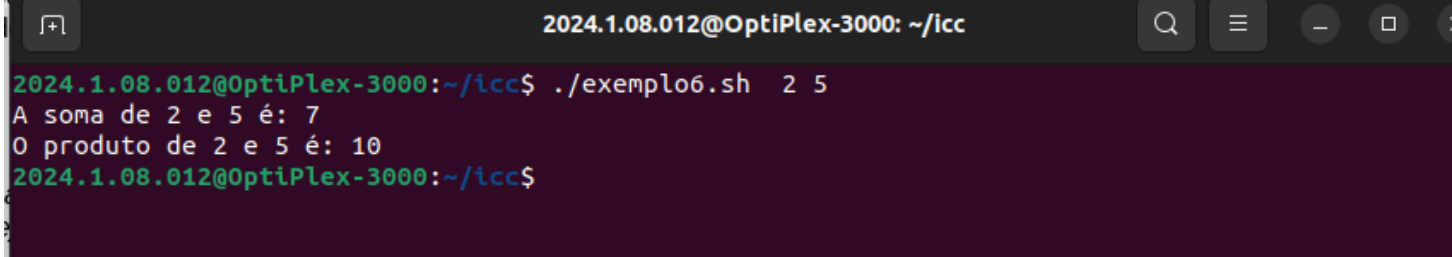
A terminal window with a dark background and light green text. The title bar shows '2024.1.08.012@OptiPlex-3000: ~/icc'. The terminal shows three commands being executed: './verifica.sh 23', './verifica.sh 8', and './verifica.sh 5'. The outputs are: 'O valor 23 não está nos intervalos especificados.', 'O triplo de 8 é 24.', and 'O triplo de 5 é 15.'.

```
2024.1.08.012@OptiPlex-3000:~/icc$ ./verifica.sh 23
O valor 23 não está nos intervalos especificados.
2024.1.08.012@OptiPlex-3000:~/icc$ ./verifica.sh 8
O triplo de 8 é 24.
2024.1.08.012@OptiPlex-3000:~/icc$ ./verifica.sh 5
O triplo de 5 é 15.
2024.1.08.012@OptiPlex-3000:~/icc$
```

- 7) Explique o que é \$# e faça um script diferente do que foi visto em aula. Faça com dois parâmetros. Use sua criatividade.

É uma variável especial em shell bash que representa o número de parâmetros passados para o script ou para uma função. Ou seja, \$# retorna a quantidade de argumentos passados para o script ou função na linha de comando.

```
1 #!/bin/bash
2
3 # Verifica se o número de argumentos é diferente de 2
4 if [ $# -ne 2 ]; then
5     echo "Uso: $0 <numero1> <numero2>"
6     exit 1
7 fi
8
9 # Verifica se ambos os argumentos são números
10 if [[ ! "$1" =~ ^[0-9]+$ ]] || [[ ! "$2" =~ ^[0-9]+$ ]]; then
11     echo "Erro: Ambos os parâmetros devem ser números inteiros."
12     exit 1
13 fi
14
15 # Calcula a soma dos números
16 soma=$(( $1 + $2 ))
17
18 # Calcula o produto dos números
19 produto=$(( $1 * $2 ))
20
21 echo "A soma de $1 e $2 é: $soma"
22 echo "O produto de $1 e $2 é: $produto"
23
```



2024.1.08.012@OptiPlex-3000: ~/icc

```
2024.1.08.012@OptiPlex-3000:~/icc$ ./exemplo6.sh 2 5
A soma de 2 e 5 é: 7
O produto de 2 e 5 é: 10
2024.1.08.012@OptiPlex-3000:~/icc$
```