



Introdução à Ciência da Computação – Lista 7
Shell script – parte 4

Nome: João Antonio Siqueira Pascuini RA: 2024.1.08.028

- 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 teste compara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevenome. Mostre o resultado da saída.



2024.1.08.028@suporte-OptiPlex-3050: ~



```
2024.1.08.028@suporte-OptiPlex-3050:~$ nano escrevanome.sh
```

```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash escrevanome.sh
```

Digite seu nome:

JOão Antonio Siqueira Pascuini

JOão Antonio Siqueira Pascuini

```
2024.1.08.028@suporte-OptiPlex-3050:~$
```

```
2024.1.08.028@suporte-OptiPlex-3050: ~  
2024.1.08.028@suporte-OptiPlex-3050:~$ nano teste compara.sh  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash escrevanome.sh  
Digite seu nome:  
2024.1.08.028  
2024.1.08.028  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash teste compara.sh  
O script escrevanome não existe.  
2024.1.08.028@suporte-OptiPlex-3050:~$ nano teste compara.sh  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash teste compara.sh  
O usuário 2024.1.08.028 NÃO tem permissão de leitura e/ou execução sobre o scri  
t escrevanome.sh.  
2024.1.08.028@suporte-OptiPlex-3050:~$
```

- 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."

```
fruta=$(echo "$fruta" | tr '[:upper:]' '[:lower:]')

# Usa a estrutura case para mostrar a descrição da fruta
case "$fruta" in
    "maçã")
        echo "A maçã é o pseudofruto pomáceo da macieira, árvore da família Rosaceae"
        ;;
    "banana")
        echo "As bananas formam-se em cachos na parte superior dos pseudocaulos das plantas da família Musaceae"
        ;;
    "laranja")
        echo "A laranja é um fruto de várias espécies cítricas da família Rutaceae"
        ;;
    "uva")
        echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae"
        ;;
    "morango")
        echo "O morango é rico em vitaminas como, por exemplo, vitamina C, A, E"
        ;;
    *)
        echo "Fruta desconhecida"
        ;;
esac
```

```
2024.1.08.028@suporte-OptiPlex-3050:~$ nano frutascase.sh
```

```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
```

Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):
maçã

A maçã é o pseudofruto pomáceo da macieira, árvore da família Rosaceae. É um dos pseudofrutos de árvore mais cultivados, e o mais conhecido dos muitos membros do gênero *Malus* que são usados pelos seres humanos.

```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
```

Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):
banana

As bananas formam-se em cachos na parte superior dos pseudocauls que nascem de um verdadeiro caule subterrâneo (rizoma ou cormo) cuja longevidade chega a 15 anos ou mais. Depois da maturação e colheita do cacho de bananas, o pseudocaul morre (ou é cortado), dando origem, posteriormente, a um novo pseudocaul.

```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
```

Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):
laranja

A laranja é um fruto de várias espécies cítricas da família Rutaceae. Refere-se principalmente ao fruto da espécie *Citrus × sinensis*, que também é chamado de laranja-doce, para distingui-lo do relacionado *Citrus × aurantium*, referido como laranja-azeda.

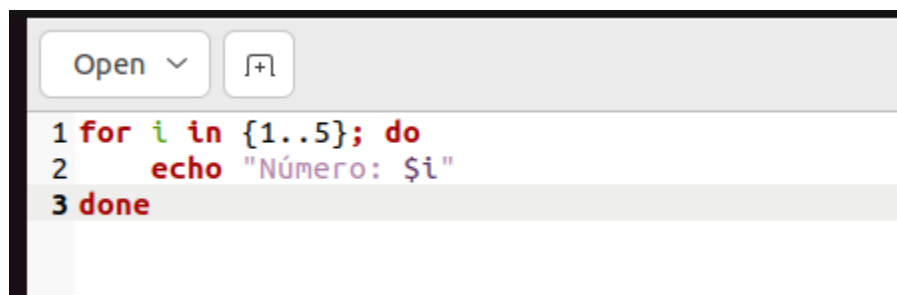
```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
```

Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):

```
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):
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 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.
2024.1.08.028@suporte-OptiPlex-3050:~$ bash frutascase.sh
Digite o nome de uma fruta (maçã, banana, laranja, uva, morango):
morango
O morango é rico em vitaminas como, por exemplo, vitamina C, A, E, B5 e B6. Os principais minerais presentes no morango são: Cálcio, Potássio, Ferro, Selênio e Magnésio. Os morangos também são ricos em flavonoides, importante agente antioxidante no organismo dos seres humanos.
2024.1.08.028@suporte-OptiPlex-3050:~$
```

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

For: É utilizado em repetições, geralmente quando se sabe o número de itens dentro de uma repetição:



```
Open  [+]
```

```
1 for i in {1..5}; do
2     echo "Número: $i"
3 done
```

```
2024.1.08.028@suporte-OptiPlex-3050: ~  
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit for.sh  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash for.sh  
Número: 1  
Número: 2  
Número: 3  
Número: 4  
Número: 5
```

While: É utilizado para executar um bloco de código enquanto uma condição estipulada for verdadeira

```
1 count=1  
2 while [ $count -le 5 ]; do  
3     echo "Número: $count"  
4     ((count++))  
5 done
```

```
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit while.sh
2024.1.08.028@suporte-OptiPlex-3050:~$ bash while.sh
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
2024.1.08.028@suporte-OptiPlex-3050:~$
```

Until: Semelhante ao “while”, porém executa o bloco enquanto a condição for falsa

```
1 count=1
2 until [ $count -gt 5 ]; do
3     echo "Número: $count"
4     ((count++))
5 done
```



```
2024.1.08.028@suporte-OptiPlex-3050: ~  
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit untill.sh  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash untill.sh  
Número: 1  
Número: 2  
Número: 3  
Número: 4  
Número: 5
```

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

O IFS é uma variável de ambiente no Bash que define os caracteres usados como delimitadores para separação de palavras em um arquivo ou script

```
Open ▾  
1 numeros="1 2 3 4 5 6 7 8 9"  
2 IFS=" "  
3 echo "$numeros"
```


```
2024.1.08.028@suporte-OptiPlex-3050: ~  
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit IFS.sh  
2024.1.08.028@suporte-OptiPlex-3050:~$ bash IFS.sh  
1 2 3 4 5 6 7 8 9  
2024.1.08.028@suporte-OptiPlex-3050:~$
```

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

```
Open ▾ [⌘+]  
1 echo "Números de 50 a 20:"  
2  
3 for (( i = 50; i >= 20; i-- )); do  
4     echo $i  
5 done
```

```
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit numeros_em_C.sh
2024.1.08.028@suporte-OptiPlex-3050:~$ bash numeros_em_C.sh
Números 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
2024.1.08.028@suporte-OptiPlex-3050:~$
```

- 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.

```
Open ▾ 
1 echo "Digite o valor desejado entre 1 e 20: "
2 read valor
3
4 # Verifica o intervalo do valor e mostra o resultado correspondente
5 if (( valor >= 0 && valor <= 10 )); then
6     echo "Triplo do valor: $((valor * 3))"
7 elif (( valor > 10 && valor <= 20 )); then
8     echo "Dobro do valor: $((valor * 2))"
9 else
10     echo "Fora dos intervalos especificados."
11 fi
```

```
 2024.1.08.028@suporte-OptiPlex-3050: ~ 🔍 ≡ - □
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit teste.sh
2024.1.08.028@suporte-OptiPlex-3050:~$ bash teste.sh
Digite o valor desejado entre 1 e 20:
28
Fora dos intervalos especificados.
2024.1.08.028@suporte-OptiPlex-3050:~$ bash teste.sh
Digite o valor desejado entre 1 e 20:
18
Dobro do valor: 36
2024.1.08.028@suporte-OptiPlex-3050:~$ bash teste.sh
Digite o valor desejado entre 1 e 20:
8
Triplo do valor: 24
2024.1.08.028@suporte-OptiPlex-3050:~$
```

7) Explique o que é `##` e faça um script diferente do que foi visto em aula. Faça com dois

parâmetros. Use sua criatividade.

O \$# é uma variável especial que representa o número de argumentos passados para um script em uma determinada linha de comando:

```
Open ▾ [icon]
1 echo "Digite o primeiro nome de um carro que venha à sua cabeça: "
2 read parametro1
3
4 # Solicita que o usuário digite o segundo parâmetro
5 echo "Digite o segundo nome de um outro carro que venha à sua cabeça: "
6 read parametro2
7
8 # Exibe os parâmetros
9 echo "O primeiro carro citado é: $parametro1"
10 echo "O segundo carro citado é: $parametro2"
```

```
2024.1.08.028@suporte-OptiPlex-3050: ~ [icon] [search] [menu] [minus] [maximize] [close]
2024.1.08.028@suporte-OptiPlex-3050:~$ gedit parametro.sh
2024.1.08.028@suporte-OptiPlex-3050:~$ bash parametro.sh
Digite o primeiro nome de um carro que venha à sua cabeça:
Siena
Digite o segundo nome de um outro carro que venha à sua cabeça:
Porsche 911 Cabriolet v6 com 654 cavalos de roda
O primeiro carro citado é: Siena
O segundo carro citado é: Porsche 911 Cabriolet v6 com 654 cavalos de roda
2024.1.08.028@suporte-OptiPlex-3050:~$
```