

Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: [Isabella Cristina da Silveira](#)

RA: [2019.2.08.004](#)

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.

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
> echo "Isabella Silveira"' > escrevenome.sh
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
> if [ -r escrevenome.sh ] && [ -x escrevenome.sh ]; then
  echo "Você tem permissão de leitura e execução sobre o escrevenome.sh"
else
  echo "Você NÃO tem permissão de leitura e/ou execução sobre o escrevenome.sh"
fi' > testecompara.sh
(base) nelson@Latitude-E5470:~$ bash escrevenome.sh
Isabella Silveira
(base) nelson@Latitude-E5470:~$ bash testecompara.sh
Você NÃO tem permissão de leitura e/ou execução sobre o escrevenome.sh
(base) nelson@Latitude-E5470:~$ s
```

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.”

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
fruta=$1
case "$fruta" in
  uva) echo "A uva é o fruto da videira. Usada para vinho, sucos e doces." ;;
  banana) echo "A banana é rica em potássio, ideal para energia rápida." ;;
  maçã) echo "A maçã é uma fruta crocante e saudável, com antioxidantes." ;;
  morango) echo "O morango é pequeno, vermelho e muito usado em sobremesas." ;;
  abacaxi) echo "O abacaxi é tropical, com sabor doce e ácido, ótimo para sucos." ;;
  *) echo "Fruta desconhecida." ;;
esac' > frutascase.sh
(base) nelson@Latitude-E5470:~$ bash frutascase.sh banana
A banana é rica em potássio, ideal para energia rápida.
(base) nelson@Latitude-E5470:~$
```

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

For: O for é usado para repetir um bloco de comandos para cada item de uma lista. Pode ser usado em dois estilos: tradicional (com lista) ou estilo C (com controle por índice).

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
for i in 1 2 3 4 5
do
    echo "Número: $i"
done' > for_loop.sh
(base) nelson@Latitude-E5470:~$ bash for_loop.sh
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
(base) nelson@Latitude-E5470:~$
```

While: O while executa um bloco de comandos enquanto a condição for verdadeira. É ideal quando não sabemos exatamente quantas vezes o laço será executado.

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
contador=1
while [ $contador -le 5 ]
do
    echo "Contador: $contador"
    ((contador++))
done' > while_loop.sh
(base) nelson@Latitude-E5470:~$ bash while_loop.sh
Contador: 1
Contador: 2
Contador: 3
Contador: 4
Contador: 5
(base) nelson@Latitude-E5470:~$
```

Until: O until é parecido com o while, mas inverte a lógica: ele executa o bloco até a condição se tornar verdadeira (ou seja, roda enquanto a condição for falsa).

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
x=1
until [ $x -gt 5 ]
do
    echo "x é: $x"
    ((x++))
done' > until_loop.sh
(base) nelson@Latitude-E5470:~$ bash until_loop.sh
x é: 1
x é: 2
x é: 3
x é: 4
x é: 5
(base) nelson@Latitude-E5470:~$
```

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

IFS é a variável que define o separador padrão para leitura de strings. O padrão é espaço, tabulação e newline.

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
texto="maçã,banana,laranja,pera"
IFS=","
for fruta in $texto
do
    echo "Fruta: $fruta"
done' > ifs_exemplo.sh
(base) nelson@Latitude-E5470:~$ bash ifs_exemplo.sh
Fruta: maçã
Fruta: banana
Fruta: laranja
Fruta: pera
(base) nelson@Latitude-E5470:~$
```

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

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
for ((i=50; i>=20; i--))
do
    echo $i
done' > for_estilo_c.sh
(base) nelson@Latitude-E5470:~$ bash for_estilo_c.sh
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
(base) nelson@Latitude-E5470:~$
```

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.

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
valor=$1
if [ $valor -ge 0 ] && [ $valor -le 10 ]; then
    echo "Triplo: $((valor * 3))"
elif [ $valor -gt 10 ] && [ $valor -le 20 ]; then
    echo "Dobro: $((valor * 2))"
else
    echo "Valor fora do intervalo permitido."
fi' > valor_intervalo.sh
(base) nelson@Latitude-E5470:~$ bash valor_intervalo.sh 8
Triplo: 24
(base) nelson@Latitude-E5470:~$ bash valor_intervalo.sh 15
Dobro: 30
(base) nelson@Latitude-E5470:~$ bash valor_intervalo.sh 25
Valor fora do intervalo permitido.
(base) nelson@Latitude-E5470:~$
```

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 no Bash (e em outros shells) que representa a quantidade de parâmetros (argumentos) passados para o script na hora da execução. Ela verifica se o número de argumentos está correto, controla a lógica do script dependendo da quantidade de argumentos e evita erros quando faltam parâmetros.

```
(base) nelson@Latitude-E5470:~$ echo '#!/bin/bash
if [ $# -ne 2 ]; then
    echo "Uso: $0 param1 param2"
else
    echo "Parâmetro 1: $1"
    echo "Parâmetro 2: $2"
    soma=$(( $1 + $2 ))
    echo "Soma dos parâmetros: $soma"
fi' > parametros.sh
(base) nelson@Latitude-E5470:~$ bash parametros.sh 5 7
Parâmetro 1: 5
Parâmetro 2: 7
Soma dos parâmetros: 12
(base) nelson@Latitude-E5470:~$
```