



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

Nome: Leticia Santos Alves

RA: 2025.1.08.016

- 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.

```
leticialuthor@Luthor-Notebook2:~$ gedit escrevanome.sh
leticialuthor@Luthor-Notebook2:~$ gedit testecompara.sh
leticialuthor@Luthor-Notebook2:~$ bash escrevanome.sh
Leticia Santos Alves
leticialuthor@Luthor-Notebook2:~$ bash testecompara.sh
O usuário leticialuthor não tem permissão de leitura e execução sobre o script escrevenome.
```

- 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.”

```
leticialuthor@Luthor-Notebook2:~$ gedit frutascase.sh
leticialuthor@Luthor-Notebook2:~$ bash frutascase.sh
0 abacaxi é uma fruta tropical produzida pela planta de mesmo nome, caracterizada como parte da família das bromeliáceas. É um símbolo das regiões tropicais e subtropicais.

1 #!/bin/bash
2 fruta=abacaxi
3 case "$fruta" in
4   uva)
5     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."
6     ;;
7   banana)
8     echo "A banana é uma fruta muito popular, cultivada em diversas regiões tropicais e subtropicais. É rica em nutrientes, especialmente em potássio, e também contém vitaminas, fibras e carboidratos."
9     ;;
10  maçã)
11    echo "A maçã é um fruto da macieira, árvore da família Rosaceae, muito consumido em todo o mundo. É uma fruta rica em nutrientes, fibras e antioxidantes, com baixo teor calórico."
12    ;;
13  laranja)
14    echo "A laranja é uma fruta cítrica rica em vitamina C, fibras e antioxidantes, com diversos benefícios para a saúde, como fortalecer o sistema imunológico, proteger contra doenças e melhorar a saúde cardiovascular."
15    ;;
16  abacaxi)
17    echo "O abacaxi é uma fruta tropical produzida pela planta de mesmo nome, caracterizada como parte da família das bromeliáceas. É um símbolo das regiões tropicais e subtropicais."
18    ;;
19  *)
20    echo "Fruta não reconhecida."
21  esac
```

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

O **for** percorre uma lista de itens ou um intervalo, repetindo comandos para cada item.

O **while** executa comandos enquanto uma condição for verdadeira.

O **until** executa comandos até que a condição se torne verdadeira.

```

leticialuthor@Luthor-Notebook2:~$ gedit for.sh
leticialuthor@Luthor-Notebook2:~$ bash for.sh
Contagem de 0 a 5:
0
1
2
3
4

```

```

1 #!/bin/bash
2 echo "Contagem de 0 a 5:"
3 for i in {0..4}; do
4     echo "$i"
5 done

```

```

leticialuthor@Luthor-Notebook2:~$ gedit while.sh
leticialuthor@Luthor-Notebook2:~$ bash while.sh
Contagem regressiva:

```

```

3
2
1
BOMMMMMMM

```

```

1 #!/bin/bash
2 echo "Contagem regressiva:"
3 contador=3
4 while [ $contador -gt 0 ]; do
5     echo "$contador"
6     contador=$((contador - 1))
7 done
8 echo "BOMMMMMMM"

```

```

leticialuthor@Luthor-Notebook2:~$ gedit until.sh
leticialuthor@Luthor-Notebook2:~$ bash until.sh
Contagem crescente:
1
2
3
4
5
FIM.

```

until.sh

```

1 #!/bin/bash
2 echo "Contagem crescente:"
3 contador=1
4 until [ $contador -gt 5 ]; do
5     echo "$contador"
6     contador=$((contador + 1))
7 done
8 echo "FIM."

```

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

IFS define quais caracteres são usados como separadores de campo ao ler dados, como em loops for ou comandos read. Por padrão, o IFS contém espaço, tabulação e quebra de linha.

```
leticialuthor@Luthor-Notebook2:~$ gedit ifs.sh
leticialuthor@Luthor-Notebook2:~$ bash ifs.sh
Os TOPS:
Kara Zor-El
Lena Luthor
Diana Prince
Lex Luthor
Harleen Quinzel
Pamela Isley
```

```
ifs.sh
~
1 #!/bin/bash
2 pessoas="Kara Zor-El,Lena Luthor,Diana Prince,Lex Luthor,Harleen Quinzel,Pamela Isley"
3 IFS=', '
4 echo "Os TOPS:"
5 for pessoas in $pessoas; do
6     echo "$pessoas"
7 done
```

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

```
leticialuthor@Luthor-Notebook2:~$ gedit c.sh
leticialuthor@Luthor-Notebook2:~$ bash c.sh
```

Contagem de 50 até 20:

```
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
```

```
c.sh
~
1 #!/bin/bash
2 echo "Contagem de 50 até 20:"
3 for (( i=50; i>=20; i-- )); do
4     echo "$i"
5 done
6 echo "FIM bb"
```

- 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.

```
leticialuthor@Luthor-Notebook2:~$ gedit parametro.sh
leticialuthor@Luthor-Notebook2:~$ bash parametro.sh
Uso: parametro.sh <número>
leticialuthor@Luthor-Notebook2:~$ gedit parametro.sh
leticialuthor@Luthor-Notebook2:~$ bash parametro.sh 6
0 valor 6 está entre 0 e 10. Seu triplo é: 18.
leticialuthor@Luthor-Notebook2:~$ bash parametro.sh 12
0 valor 12 está entre 10 e 20. Seu dobro é: 24.
leticialuthor@Luthor-Notebook2:~$ bash parametro.sh 27
0 valor 27 não está entre 0 e 20.
```

```
1 #!/bin/bash
2 if [ -z "$1" ]; then
3     echo "Uso: $0 <número>"
4     exit 1
5 fi
6 valor=$1
7 if [ $valor -ge 0 ] && [ $valor -le 10 ]; then
8     triplo=$((valor * 3))
9     echo "0 valor $valor está entre 0 e 10. Seu triplo é: $triplo."
10 elif [ $valor -gt 10 ] && [ $valor -le 20 ]; then
11     dobro=$((valor * 2))
12     echo "0 valor $valor está entre 10 e 20. Seu dobro é: $dobro."
13 else
14     echo "0 valor $valor não está entre 0 e 20."
15 fi
```

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

A variável especial \$# contém o número de parâmetros de linhas de comando fornecidos ao rodar o script.

```
leticialuthor@Luthor-Notebook2:~$ gedit ultimo.sh
leticialuthor@Luthor-Notebook2:~$ bash ultimo.sh 2 3
A multiplicação dos números é 6
Os números ao quadrado são 4 e 9
leticialuthor@Luthor-Notebook2:~$ bash ultimo.sh 12 15
A multiplicação dos números é 180
Os números ao quadrado são 144 e 225
```

```
1 #!/bin/bash
2 if [ $# -ne 2 ]; then
3     echo "Usando $0 <numero1> <numero2>"
4     exit 1
5 fi
6 num1=$1
7 num2=$2
8 multi=$((num1 * num2))
9 elevar1=$((num1 * num1))
10 elevar2=$((num2 * num2))
11 echo "A multiplicação dos números é $multi"
12 echo "Os números ao quadrado são $elevar1 e $elevar2"
```