



Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: Isadora Resende Grandaux

RA:

- 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.

```
2025.1.08.012@suporte-OptiPlex-3050:~$ bash
2025.1.08.012@suporte-OptiPlex-3050:~$ nano escrevenome.sh
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ ./escrevenome.sh
bash: ./escrevenome.sh: Permission denied
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano testecompara.sh
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 testecompara.sh
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ ./testecompara.sh
./testecompara.sh: line 2: [-r: command not found
"O usuário isadora não pode alterar o arquivo agora"
2025.1.08.012@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, geleias, tortas, gelatinas, sucos.”

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano frutascase.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ nano frutascase.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 frutascase.sh
chmod: cannot access 'frutascase.sh': No such file or directory
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 frutascase.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./frutascase.sh
A manga é uma fruta tropical, rica em vitaminas A e C, com sabor doce e suculento
2025.1.08.012@suporte-OptiPlex-3050:~$
```

```
GNU nano 6.2 frutascase.sh
fruta="manga"
case $fruta in
"uva")
echo " uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e
"manga")
echo " A manga é uma fruta tropical, rica em vitaminas A e C, com sabor doce e suculento";;
"banana")
echo " A banan é uma fruta rica em potássio, muito consumida no mundo todo, excelente para energia r
"maçã")
echo "A maçã é consumida, em sua maioria, na forma de fruta, ou de suco de maçã. Fermentada, é utili
"abacaxi")
echo "O abacaxi é uma fruta tropical ácida, rica em bromelina, excelente para digestão de proteínas"
*)
echo "Fruta desconhecuiida";;
esac
```

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

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

echo "Múltiplos de 3 (de 1 a 5):"
for i in {1..5}; do
    echo "$((i * 3))";
done
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano for.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 for.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./for.sh
Múltiplos de 3 (de 1 a 5):
3
6
9
12
15
```

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

senha_correta="bash123"
tentativa=""

while [ "$tentativa" != "$senha_correta" ]; do
    read -sp "Digite a senha: " tentativa
    echo
    if [ "$tentativa" != "$senha_correta" ]; then
        echo "Senha incorreta. Tente novamente!"
    fi
done

echo "Acesso permitido."
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano while.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 while.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./while.sh
Digite a senha:
Senha incorreta. Tente novamente!
```

```
GNU nano 6.2                               until.sh *
#!/bin/bash

total=0

echo "Adicionando moedas de R$ 0,25 até atingir R$ 1,00..."

until [ "$total" -ge 100 ]; do
    total=$((total + 25))
    echo "Total atual: R$ $(bc <<< "scale=2; $total/100")"
done

echo "Você atingiu R$ 1,00!"
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano until.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 until.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./until.sh
Adicionando moedas de R$ 0,25 até atingir R$ 1,00...
Total atual: R$ .25
Total atual: R$ .50
Total atual: R$ .75
Total atual: R$ 1.00
Você atingiu R$ 1,00!
2025.1.08.012@suporte-OptiPlex-3050:~$
```

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

R: IFS significa **Internal Field Separator** (Separador de Campo Interno). Ele define **quais** caracteres o shell deve usar para separar palavras em uma linha.

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

# Define o IFS como espaço para dividir o nome completo
IFS=' '

read -p "Digite seu nome completo: " nome_completo

# Salva o IFS atual e altera temporariamente
OLDIFS=$IFS
IFS=' '

# Converte a string em array usando o novo IFS
read -ra partes <<< "$nome_completo"

echo "Partes do nome:"
for parte in "${partes[@]}"; do
    echo "- $parte"
done

# Restaura o IFS original
IFS=$OLDIFS
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano ifs.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ nano ifs.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 ifs.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./ifs.sh
Digite seu nome completo: Isadora Resende Grandaux
Partes do nome:
- Isadora
- Resende
- Grandaux
2025.1.08.012@suporte-OptiPlex-3050:~$
```

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

```
GNU nano 6.2                                C.sh *
#!/bin/bash

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

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano C.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 C.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./C.sh
Numero: 50
Numero: 49
Numero: 48
Numero: 47
Numero: 46
Numero: 45
Numero: 44
Numero: 43
Numero: 42
Numero: 41
Numero: 40
Numero: 39
Numero: 38
Numero: 37
Numero: 36
Numero: 35
Numero: 34
Numero: 33
Numero: 32
Numero: 31
Numero: 30
Numero: 29
Numero: 28
Numero: 27
Numero: 26
Numero: 25
Numero: 24
Numero: 23
Numero: 22
Numero: 21
Numero: 20
2025.1.08.012@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.

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

valor=1

# Verifica se o valor é um número usando regex
if ! [[ $valor =~ ^[0-9]+([.][0-9]+)?$ ]]; then
    echo "Erro: '$valor' não é um número válido."
    exit 2
fi

# Usa bc para comparações com ponto flutuante
if (( $(echo "$valor >= 0 && $valor <= 10" | bc -l) )); then
    resultado=$(echo "$valor * 3" | bc)
    echo "O valor está entre 0 e 10. Triplo: $resultado"
elif (( $(echo "$valor > 10 && $valor < 20" | bc -l) )); then
    resultado=$(echo "$valor * 2" | bc)
    echo "O valor está entre 10 e 20. Dobro: $resultado"
else
    echo "O valor não está entre 0 e 20 nos intervalos definidos."
fi
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano parametro.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 parametro.sh
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ ./parametro.sh
O valor está entre 0 e 10. Triplo: 3
```

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

R:\$# representa a quantidade de parametros passados para o script na linha de comando.

```
GNU nano 6.2 par.sh *
#!/bin/bash

echo "Você passou $# argumentos."
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano par.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 par.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./par.sh arg1 arg2 arg3
Você passou 3 argumentos.
2025.1.08.012@suporte-OptiPlex-3050:~$
```