



Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: 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 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.

```
diego@diego: ~  
GNU nano 7.2 escrevenome *  
#!/bin/bash  
echo "Diego Vianna Leite Montemor"
```

```
diego@diego: ~  
GNU nano 7.2 testecompara *  
#!/bin/bash  
  
SCRIPT="./escrevenome"  
  
[ -r "$SCRIPT" ] && [ -x "$SCRIPT" ] && echo "O usuário tem permissão de leitura"
```

```
diego@diego: ~  
diego@diego:~$ nano escrevenome  
diego@diego:~$ nano testecompara  
diego@diego:~$ ./testecompara  
bash: ./testecompara: Permission denied
```

- 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."

```
diego@diego: ~  
GNU nano 7.2 frutacase *  
#!/bin/bash  
  
fruta=banana  
  
case $fruta in  
uva)  
    echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae"  
    ;;  
banana)  
    echo "A banana é uma fruta tropical muito consumida no mundo todo. Rica em"  
    ;;  
maçã)  
    echo "A maçã é uma fruta originária da Ásia e amplamente cultivada em região"  
    ;;  
manga)  
    echo "A manga é uma fruta tropical suculenta, originária do sul da Ásia, mu"  
    ;;  
abacaxi)  
    echo "O abacaxi é uma fruta tropical com sabor agridoce, cultivada principa"  
    ;;  
*)  
    echo "Fruta desconhecida"  
    ;;  
esac
```

```
diego@diego:~$ nano frutacase  
diego@diego:~$ bash frutacase  
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, mussels, geléias, tortas, gelatinas, sucos.  
diego@diego:~$ nano frutacase  
diego@diego:~$ bash frutacase  
A banana é uma fruta tropical muito consumida no mundo todo. Rica em potássio, fibras e vitaminas, é excelente para fornecer energia rapidamente. É produzida pela bananeira, planta da família Musaceae. Pode ser consumida in natura, em vitaminas, bolos, frita ou assada.  
diego@diego:~$ nano frutacase  
diego@diego:~$ bash frutacase  
A manga é uma fruta tropical suculenta, originária do sul da Ásia, muito apreciada por seu sabor doce. É rica em vitaminas A e C, além de antioxidantes. A manga pode ser consumida fresca, em sucos, saladas, doces e molhos.  
diego@diego:~$
```

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

sua criatividade para os scripts.

```
GNU nano 7.2 for *
#!/bin/bash

echo "Dias da semana:"
for dia in Segunda Terça Quarta Quinta Sexta Sábado Domingo
do
    echo "- $dia"
done
```

```
diego@diego:~$ nano for
diego@diego:~$ bash for
Dias da semana:
- Segunda
- Terça
- Quarta
- Quinta
- Sexta
- Sábado
- Domingo
diego@diego:~$
```

```
diego@diego: ~  
GNU nano 7.2 while *  
#!/bin/bash  
  
contador=5  
echo "Iniciando contagem regressiva:"  
while [ $contador -gt 0 ]  
do  
    echo "$contador..."  
    sleep 1  
    contador=$((contador - 1))  
done  
echo "FIM!"
```

```
diego@diego:~$ nano while  
diego@diego:~$ bash while  
Iniciando contagem regressiva:  
5...  
4...  
3...  
2...  
1...  
FIM!
```

```
diego@diego: ~  
GNU nano 7.2 until *  
#!/bin/bash  
  
senha=""  
senha_correta="abc123"  
  
until [ "$senha" = "$senha_correta" ]  
do  
    read -sp "Digite a senha: " senha  
    echo ""  
done  
  
echo "Senha correta. Acesso liberado!"
```

```
diego@diego:~$ nano until  
diego@diego:~$ bash until  
Digite a senha:  
Digite a senha:  
Digite a senha:  
Senha correta. Acesso liberado!  
diego@diego:~$
```

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

```
diego@diego:~$ nano ifs
diego@diego:~$ bash ifs
Digite seus filmes favoritos separados por ponto e vírgula (;)
batman;superman;SenhorDosAneis
🎬 batman
🎬 superman
🎬 SenhorDosAneis
diego@diego:~$
```

```
GNU nano 7.2 ifs *
#!/bin/bash

echo "Digite seus filmes favoritos separados por ponto e vírgula (;)"
read filmes

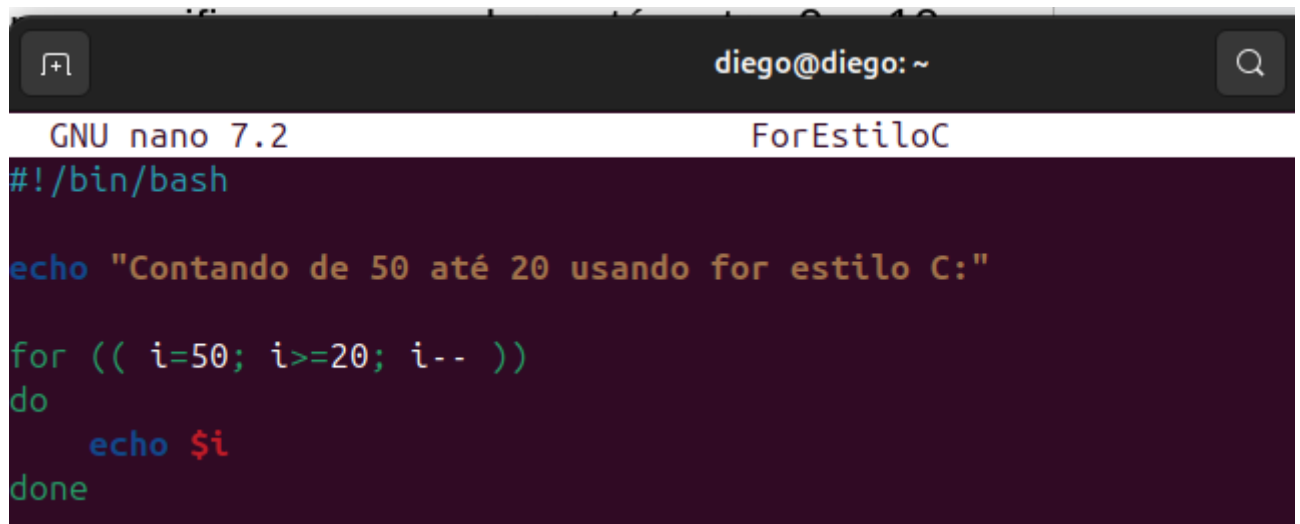
OLDIFS=$IFS

IFS=';'

for filme in $filmes
do
    echo "🎬 $filme"
done

IFS=$OLDIFS
```

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



The image shows a terminal window with a dark background. At the top, there is a title bar with a window icon on the left, the text "diego@diego: ~" in the center, and a search icon on the right. Below the title bar, the text "GNU nano 7.2" is on the left and "ForEstiloC" is on the right. The main area of the terminal contains a bash script written in a monospaced font with syntax highlighting. The script starts with a shebang line, followed by an echo statement, a for loop with C-style syntax, and a do-done block containing an echo statement.

```
#!/bin/bash

echo "Contando de 50 até 20 usando for estilo C:"

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



```
diego@diego:~$ bash ForEstiloC
Contando de 50 até 20 usando for estilo C:
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
```

- 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 7.2                               VerificaValor *
#!/bin/bash

if [ -z "$1" ]; then
    echo "Uso: $0 <número>"
    exit 1
fi

valor=$1

if ! [[ "$valor" =~ ^[0-9]+([.][0-9]+)?$ ]]; then
    echo "Erro: o valor informado não é numérico."
    exit 1
fi

if (( $(echo "$valor >= 0" | bc -l) )) && (( $(echo "$valor <= 10" | bc -l) ));>
    triplo=$(echo "$valor * 3" | bc -l)
    echo "O valor está entre 0 e 10. Triplo: $triplo"

elif (( $(echo "$valor > 10" | bc -l) )) && (( $(echo "$valor <= 20" | bc -l) ));>
    dobro=$(echo "$valor * 2" | bc -l)
    echo "O valor está entre 10 e 20. Dobro: $dobro"

else
    echo "O valor não está entre 0 e 20."
fi
```

```
diego@diego:~$ nano VerificaValor
diego@diego:~$ bash VerificaValor
Uso: VerificaValor <número>
diego@diego:~$ bash VerificaValor 7
0 valor está entre 0 e 10. Triplo: 21
diego@diego:~$
```

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

```
diego@diego:~$ nano ComparaPalavras.sh
diego@diego:~$ bash ComparaPalavras.sh oi ola
Você digitou 2 palavras: 'oi' e 'ola'
As palavras são diferentes.
Tamanhos: 'oi' = 2 letras, 'ola' = 3 letras.
'oi' vem antes de 'ola' no dicionário.
diego@diego:~$ bash ComparaPalavras.sh oi oi
Você digitou 2 palavras: 'oi' e 'oi'
As palavras são iguais.
diego@diego:~$
```

```
diego@diego: ~
GNU nano 7.2 ComparaPalavras.sh
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Uso: $0 <palavra1> <palavra2>"
    exit 1
fi

palavra1=$1
palavra2=$2

echo "Você digitou $# palavras: '$palavra1' e '$palavra2'"

if [ "$palavra1" = "$palavra2" ]; then
    echo "As palavras são iguais."
else
    echo "As palavras são diferentes."

    tam1=${#palavra1}
    tam2=${#palavra2}
    echo "Tamanhos: '$palavra1' = $tam1 letras, '$palavra2' = $tam2 letras."

    if [[ "$palavra1" < "$palavra2" ]]; then
        echo "'$palavra1' vem antes de '$palavra2' no dicionário."
    else
        echo "'$palavra2' vem antes de '$palavra1' no dicionário."
    fi
fi
```