

## **Lista de Exercícios 08**

### **Estruturas de Fluxo Condicionais**

**Exercício 01)** Escreva um *shell script* que recebe um arquivo como argumento em linha de comando e verifica se existe ou não. Caso exista, verifique se é um arquivo, um diretório ou desconhecido mostrando quais as permissões que o **usuário atual (não o dono do arquivo)** possui sobre ele. Caso não exista, imprima que é inválido. **Cuidado**, essas informações não são obtidas com comandos como ls, stat ou file.

```
$ ./checkfile                               $ ./checkfile /dev/random
Usar: ./checkfile [Arquivo]                  /dev/random: desconhecido (rw-)
$ ./checkfile /etc/passwd                   $ ./checkfile /private
/etc/passwd: arquivo (r--)                 /private: inválido
$ ./checkfile /tmp                          /tmp: diretório (rwx)
```

```
#!/bin/bash
```

```
# Verificando se o usuário digitou o comando corretamente e o arquivo
# passado existe
```

```
if [ $# -eq 0 ]; then
    echo "Usar: ./checkfile [Arquivo]"
    exit 1
elif [[ 2! (-e "$1") ]]; then
    echo "$1: inválido"
    exit 1
fi
```

```
# Verifica as permissões do usuário
```

```
if [ -r "$1" ]; then
    permr="r"
else
    permr="-"
fi
```

```
if [ -w "$1" ]; then
    permw="w"
```

```
else
    permw="-"
fi

if [ -x "$1" ]; then
    permx="x"
else
    permx="-"
fi

# Printa o tipo do arquivo e as permissões
echo -n "$1: "
if [ -d "$1" ]; then
    echo -n "diretorio"
elif [ -f "$1" ]; then
    echo -n "arquivo"
else
    echo -n "desconhecido"
fi
echo " ($permr$permw$permx)"
```

**Exercício 02)** Escreva um *shell script* que recebe três números inteiros pela **entrada padrão** (e não via argumentos em linha de comando) e mostre qual o maior deles.

```
$ ./higher
Primeiro número: 5
Segundo número: 12
Terceiro número: 8

O maior número é o 12
```

```
#!/bin/bash
```

```
echo -n "Primeiro número: "
read num1
echo -n "Segundo número: "
read num2
echo -n "Terceiro número: "
read num3

echo
echo -n "O maior número é o "
if [ $num1 -ge $num2 ]; then
    if [ $num1 -ge $num3 ]; then
        echo "$num1"
    else
        echo "$num3"
    fi
elif [ $num2 -ge $num3 ]; then
    echo "$num2"
else
    echo "$num3"
fi
```

**Exercício 03)** Escreva um *shell script* para cumprimentar um usuário com Bom dia, Boa tarde ou Boa noite dependendo do horário do sistema. A saudação deve incluir o nome completo do usuário atualmente logado. Dica: use o comando whoami para obter o usuário atual e depois consulte o arquivo /etc/passwd na quinta coluna para obter o seu nome completo. As horas do dia são divididas da seguinte maneira:

- **Dia:** até 12:00
- **Tarde:** a partir de 12:00 e até 18:00
- **Noite:** a partir das 18:00

Exemplo de utilização do *shell script*.

```
$ date
Sun Nov 30 12:40:01 BRST 2014
$ ./greeting
Boa tarde João do Caminhão, seja bem vindo.
```

```
#!/bin/bash
```

```
data=$(date | awk '{print $5}' | awk -F ':' '{print $1}')
nome=$(whoami)
```

```
if [ $data -lt 12 -a $data -ge 00 ]; then
    echo -n "Bom dia"
elif [ $data -lt 18 ]; then
    echo -n "Boa tarde"
else
    echo -n "Boa noite"
fi
```

```
echo " \"$nome\", seja bem vindo"
```

**Exercício 04)** O comando `expr` é capaz de realizar cálculos com expressões aritméticas inteiras. Reescreva um programa similar em *shell script* para calcular uma expressão com apenas um operador com dois números em ponto flutuante e imprima o resultado em ponto flutuante. Os operadores suportados com seus respectivos símbolos são: adição (+), subtração (-), multiplicação (x), e divisão (/). **Você deve usar obrigatoriamente a estrutura case como controle de fluxo condicional.** O programa deve ler os dois números e o operador via parâmetros em linha de comando. Caso não sejam especificados os três parâmetros, deve-se mostrar uma mensagem de ajuda. Se o operador não for um dos quatro suportados deve-se mostrar uma mensagem de erro. Dica: use o comando `bc` (*basic calculator*) para fazer as operações em ponto-flutuante. Exemplo de uso do comando `bc`.

```
$ echo "22 / 7" | bc -l
3.14285714285714285714
```

Exemplo de uso do *shell script*.

```
$ ./calc
Usar: ./calc [Número] [Operador] [Número]
$ ./calc 22 / 7
3.14285714285714285714
$ ./calc 12 - 7.2
4.8
$ ./calc 5 % 4
%: operação inválida
1
#!/bin/bash

if [ $# != 3 ]; then
    echo "Usar: ./calc [Número] [Operador] [Número]"
    exit 1
fi

case "$2" in
    x)
        echo "$1 * $3" | bc -l
        ;;
    /)
        echo "$1 / $3" | bc -l
        ;;
    +)
        echo "$1 + $3" | bc -l
        ;;
    -)
        echo "$1 - $3" | bc -l
        ;;
    *)
        echo "$2: operação inválida"
        ;;
esac
```

; ;  
esac