

# Capítulo 8

## ===Exercício 1===

1. Fazer um programa (usando a instrução `eval`) que liste os parâmetros recebidos em ordem inversa.

```
$ cat exerc8-1.sh
#!/bin/bash
# Exercício 1 do capítulo 8
# Lista os parâmetros recebidos em ordem inversa
#+ Receberá:
#+ $1...$N - Parâmetros que serão
#+ listados na ordem inversa
function Erro
{
    echo "$*" >&2      # Esse $* é o da função, não é o do script
}
(($# < 2)) && {
    Erro Uso: $0 P1 P2 ... PN, que são os parâmetros a inverter
    exit 1
}
for ((Parm=$#; Parm>0; Parm--))
do
    eval echo O parâmetro $Parm é \$$Parm
    # Como o eval dá duas passadas no comando, na primeira
    #+ ele substitui $Parm pelo seu valor e tira a contrabarra
    #+ na segunda, ele escreve o valor do parâmetro.
done
```

## ===Exercício 2===

Escrever uma função que mandará uma mensagem de erro, que será recebida como parâmetro e será precedida pelo nome do programa.

A cadeia formada deverá ser redirecionada para *StdErr*.

Por exemplo: Se o programa, digamos, `exerc8-2.sh` tiver uma função `Erro` para fazer o solicitado, quando você chamar `Erro` passando:

**Este campo deveria ser numérico**

`Erro` deveria mandar para *StdErr* a seguinte mensagem:

**exerc8-2.sh: Este campo deveria ser numérico**

```
$ cat exerc8-2.sh
#!/bin/bash
# Exercício 2 do capítulo 8
#+ Cria uma função para dar uma mensagem de erro, formatada
```

```

#+ com o nome do programa, com a msg e enviada para StdErr
#+ Receberá:
#+      $* ou $@ - Mensagem
function Erro
{
    echo $0: "$*" >&2      # O mesmo que: echo $0: "$@" >&2
}

```

### ===Exercício 3===

Vou complicar um pouco o exercício anterior que até agora está muito moleza. Agora você vai mandar uma mensagem de erro para *StdOut*, recebendo os seguintes parâmetros:

- **\$1** - A mensagem de erro;
- **\$2** - A linha em que a mensagem será listada;
- **\$3** - A coluna a partir da qual a mensagem será listada. Esse parâmetro é opcional e caso não seja informado a mensagem deverá ser centrada na tela.

Sua função imprimirá a mensagem no local informado e ficará aguardando o usuário teclar algo para apagar o texto e prosseguir.

```

$ cat exerc8-2.sh
#!/bin/bash
# Exercício 3 do capítulo 8
#+ Cria uma função para dar uma mensagem.
#+ Receberá:
#+      $1 - Mensagem
#+      $2 - Linha na qual será dada a mensagem
#+      $3 - Coluna da mensagem (opcional)
function MandaErro
{
    local Centro=$((($tput cols) - ${#1}) / 2)
    local Col=${3:-$Centro} # Se $3 vazia, recebe valor de $Centro
    tput cup $2 $Col
    echo -n $1 >&2          # Manda mensagem para StdErr
    read -n1               # Espera qualquer caractere ser teclado
    tput cup $2 $Col       # Posiciona para apagar a msg
    tput el               # Erase Line apaga linha a partir deste ponto
}

```