



Introdução à Ciência da Computação – Lista 8 Shell script – parte 5

Nome: Isadora Resende Grandaux RA:

- 1) Qual variável de ambiente guarda o valor lido pelo comando read, caso não tenha nenhuma criada para guardar o valor? Faça um script utilizando essa variável de ambiente, onde a entrada será o nome do mês atual. Mostre a execução. Depois acrescente um timer de 8 segundos, como mostrado em aula e apresente a execução.

R:A variável de ambiente que guarda o valor lido pelo comando read, caso nenhuma variável seja especificada, é reply.

```
#!/bin/bash

# Parte 1: Leitura do mês e exibição

echo "Qual o nome do mês atual?"
read

echo "O mês atual é: $REPLY"

# Parte 2: Adição de um timer

echo "Aguardando 8 segundos..."
sleep 8
echo "Tempo esgotado!"

echo "Obrigado por utilizar o script!"
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano mes_atual.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 mes_atual.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./mes_atual.sh
Qual o nome do mês atual?
junho
O mês atual é: junho
Aguardando 8 segundos...
Tempo esgotado!
Obrigado por utilizar o script!
2025.1.08.012@suporte-OptiPlex-3050:~$
```

- 2) Crie um arquivo txt sobre você (nome, idade, cidade, hobbies, etc) e faça a leitura do arquivo com o comando read. Apresentando todo o conteúdo do arquivo no terminal.

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

echo "--- Conteúdo do arquivo 'sobre_mim.txt' ---"
echo ""

# Verifica se o arquivo existe
if [ ! -f "sobre_mim.txt" ]; then
    echo "Erro: O arquivo 'sobre_mim.txt' não foi encontrado no diretório atual."
    echo "Por favor, crie o arquivo com suas informações antes de executar este script."
    exit 1
fi

# Loop para ler cada linha do arquivo
while IFS= read -r linha; do
    echo "$linha"
done < sobre_mim.txt

echo ""
echo "--- Fim do conteúdo ---"
```

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano eu.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./eu.sh
--- Conteúdo do arquivo 'sobre_mim.txt' ---

Nome: Isadora
Idade:19 anos
Cidade: Alfenas
Hobbies: ler livros e histórias curtas em inglês ou espanhol para aprender.
Curiosidades: tenho duas galinhas de estimação

--- Fim do conteúdo ---
2025.1.08.012@suporte-OptiPlex-3050:~$
```

3) Desenvolva um script utilizando o comando case. Cada opção chama uma função e apresenta o resultado da função. Exemplos:

```
--- Operações Matemáticas ---
1. Subtrair 10
2. Multiplicar por 10
3. Dividir por 2
4. Sair
-----
Digite um número entre 1 e 4: 1
Digite um número: 20

Subtraindo 10: 10

Pressione Enter para continuar...
```

```
--- Operações Matemáticas ---
1. Subtrair 10
2. Multiplicar por 10
3. Dividir por 2
4. Sair
-----
Digite um número entre 1 e 4: 4

Saindo do script. Até mais!
2025.1.08.012@suporte-OptiPlex-3050:~$
```

4) Modifique o script abaixo para que apresente o dobro apenas dos números entre 20 e 30.

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

# Função para calcular o dobro de um número e validar o intervalo
function calcular_dobro() {
    # Pergunta ao usuário para digitar um número
    read -p "Digite um número entre 20 e 30: " numero

    # Verifica se a entrada é um número e se está no intervalo correto
    if [[ "$numero" =~ ^[0-9]+$ ]] && (( numero >= 20 && numero <= 30 )); then
        local dobro=$((numero * 2))
        echo "O dobro de $numero é: $dobro"
    else
        echo "Número inválido ou fora do intervalo [20, 30]."
```

Saindo do script. Até mais!

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano modifique.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 modifique.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./modifique.sh
Digite um número entre 20 e 30: 25
O dobro de 25 é: 50
2025.1.08.012@suporte-OptiPlex-3050:~$
```

5) Crie um script como whiptail, receba um número e mostre o dobro daquele número.

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

# Verifica se o whiptail está instalado
if ! command -v whiptail &> /dev/null
then
    echo "Erro: O comando 'whiptail' não foi encontrado."
    echo "Por favor, instale-o. Em sistemas baseados em Debian/Ubuntu, use: sudo apt install whiptail"
    echo "Em sistemas baseados em Red Hat/CentOS, use: sudo yum install newt ou sudo dnf install newt"
    exit 1
fi

# Variáveis para a caixa de diálogo
TITLE="Cálculo do Dobro"
BACKTITLE="Script com Whiptail"
HEIGHT=15
WIDTH=50

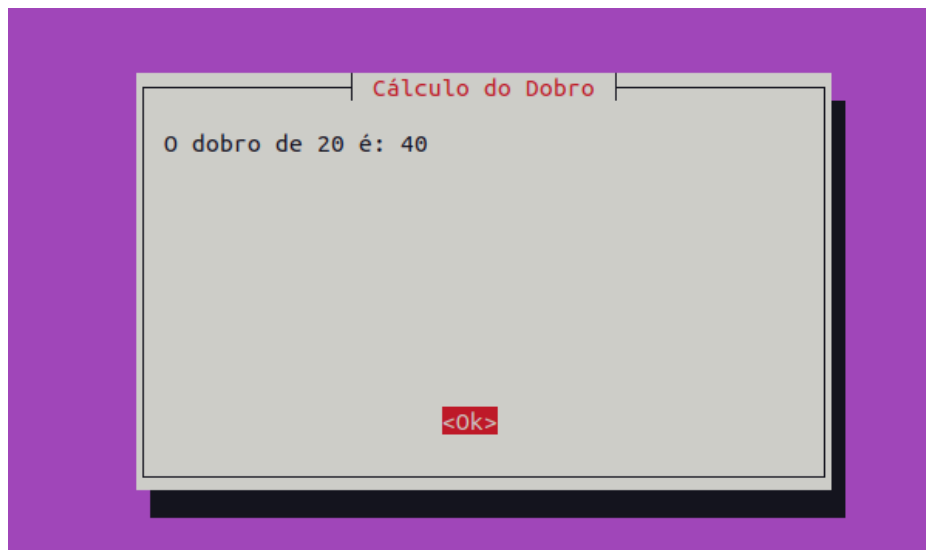
# --- Passo 1: Pedir o número ao usuário ---
# Usando 'inputbox' para receber a entrada de texto
input_value=$(whiptail --title "$TITLE" --backtitle "$BACKTITLE" \
    --inputbox "Digite um número inteiro:" \
    $HEIGHT $WIDTH "" 3>&1 1>&2 2>&3)

# Verifica se o usuário cancelou (pressionou Esc ou Cancelar)
exitstatus=$?
if [ $exitstatus != 0 ]; then
    whiptail --title "Saída" --msgbox "Operação cancelada pelo usuário." $HEIGHT $WIDTH
    exit 0
fi

# --- Passo 2: Validar e calcular o dobro ---
# Verifica se a entrada é um número (aceita números negativos)
if [[ "$input_value" =~ ^-?[0-9]+$ ]]; then
    dobro=$((input_value * 2))
    mensagem="O dobro de $input_value é: $dobro"
else
    mensagem="Entrada inválida. Por favor, digite apenas números inteiros."
fi

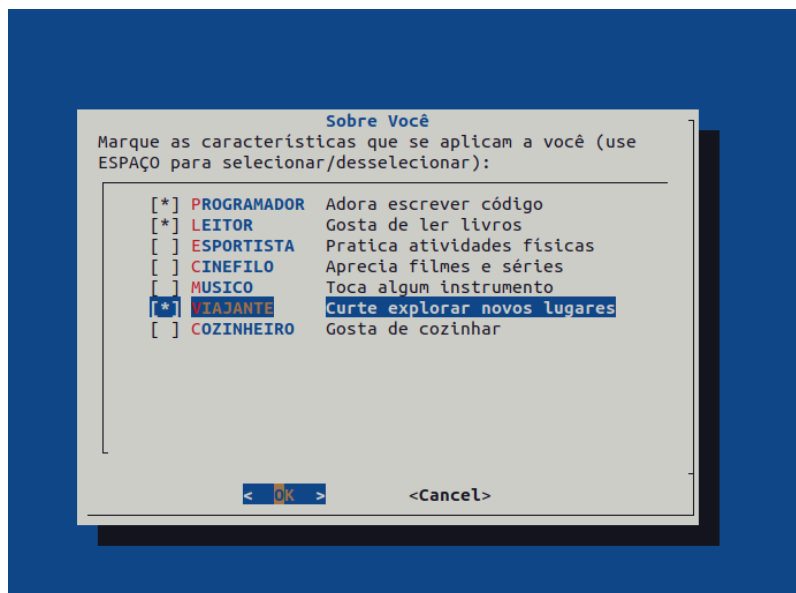
# --- Passo 3: Mostrar o resultado em uma caixa de mensagem ---
# Usando 'msgbox' para exibir a saída
whiptail --title "$TITLE" --backtitle "$BACKTITLE" \
    --msgbox "$mensagem" \
    $HEIGHT $WIDTH

exit 0
```



- 6) Escolha um widget do comando dialog e mostre sua execução. Escolha um diferente dos que foram mostrados em aula.

```
GNU nano 6.2                                2025.1.08.012@suporte-OptiPlex-3050: ~
2>&1 >/dev/tty) # Redireciona a saída para a variável
# Captura o status de saída (0 para OK, 1 para Cancelar/Esc)
exitstatus=$?
# Limpa a tela após a interação com o dialog
clear
# Processa a escolha do usuário
if [ $exitstatus = 0 ]; then
    if [ -z "$CHOICES" ]; then # Verifica se a string CHOICES está vazia
        echo "Você não selecionou nenhuma característica."
    else
        echo "Suas características selecionadas são:"
        # O IFS (Internal Field Separator) é importante aqui para separar as opções selecionadas por nova linha
        IFS=$'\n' # Define o IFS como nova linha
        for choice in $CHOICES; do
            echo "- $choice"
        done
        IFS=$' \t\n' # Restaura o IFS para o padrão
    fi
else
    echo "Você cancelou a operação."
fi
exit 0
```



```
Suas características selecionadas são:
- PROGRAMADOR LEITOR VIAJANTE
2025.1.08.012@suporte-OptiPlex-3050:~$
```

7) Desenvolva um ckecklist, utilize sua criatividade. Pode ser whiptail ou dialog.

Checklist de Viagem para a Praia

Marque os itens que você já tem ou precisa levar:

<input checked="" type="checkbox"/>	ROUPAS_BANHO	Roupas de banho (biquini, sunga)
<input checked="" type="checkbox"/>	TOALHAS	Toalhas de praia
<input checked="" type="checkbox"/>	PROTETOR_SOLAR	Protetor solar (indispensável!)
<input checked="" type="checkbox"/>	OCULOS_SOL	Óculos de sol
<input type="checkbox"/>	CHAPEU	Chapéu ou boné
<input checked="" type="checkbox"/>	CHINELO	Chinelo/sandália
<input checked="" type="checkbox"/>	ROUPAS_LEVES	Roupas leves (shorts, camisetas)
<input type="checkbox"/>	SAIDA_PRAIA	Saídas de praia
<input checked="" type="checkbox"/>	CARREGADORES	Carregadores de celular/dispositivos
<input type="checkbox"/>	FONES_OUVIDO	Fones de ouvido
<input type="checkbox"/>	LIVRO	Livro ou revista para relaxar
<input type="checkbox"/>	CAIXA_SOM	Caixa de som portátil
<input type="checkbox"/>	LANCHES	Lanches e petiscos
<input checked="" type="checkbox"/>	AGUA_GARRAFA	Garrafas de água reutilizáveis
<input checked="" type="checkbox"/>	SACO_LIXO	Sacos de lixo (para manter a praia limpa)

<Ok>

<Cancel>

```
2025.1.08.012@suporte-OptiPlex-3050:~$ nano praia.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ chmod 755 praia.sh
2025.1.08.012@suporte-OptiPlex-3050:~$ ./praia.sh
--- Seus Itens Selecionados para a Praia: ---
## Essenciais:

## Extras e Lazer:

## Saúde e Segurança:
-----
Pronto para o sol, areia e mar! Tenha uma excelente viagem!
2025.1.08.012@suporte-OptiPlex-3050:~$
```