



## Introdução à Ciência da Computação – Lista 8

### Shell script – parte 5

Nome: Wagner Donizete Gonçalves

RA: 2019.1.08.047

- 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.

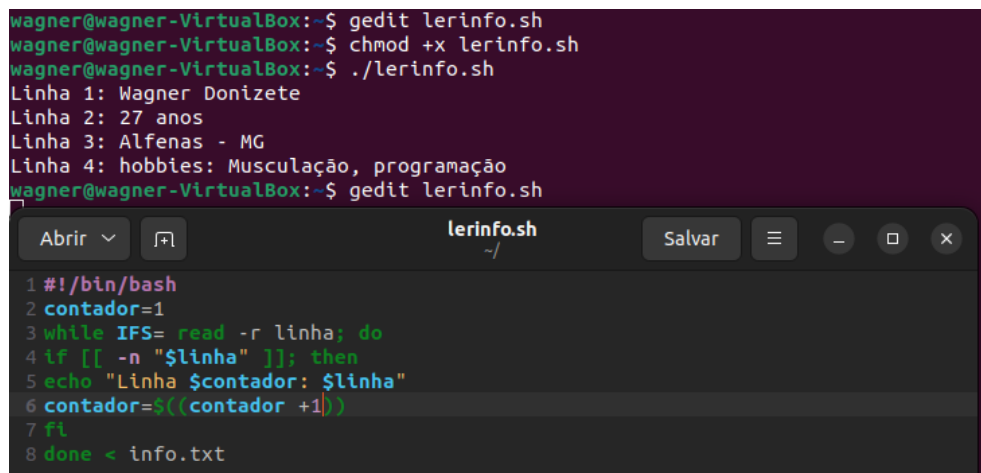
```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit test1.sh  
wagner@wagner-VirtualBox:~$ chmod +x test1.sh  
wagner@wagner-VirtualBox:~$ ./test1.sh  
Digite o mês atual:  
Junho  
o Mes atual é: Junho  
1 #!/bin/bash  
2 #usando timer  
3 echo "Digite o mês atual: "  
4 if read -t 8 mesatual  
5 then  
6     echo " o Mes atual é: $mesatual"  
7 else  
8     echo "já se passou 8 segundos..."  
9     echo "Tempo de espera concluído."  
10 fi  
11
```

```
wagner@wagner-VirtualBox:~$ ./test1.sh  
Digite o mês atual:  
já se passou 8 segundos...  
Tempo de espera concluído.
```

- 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.

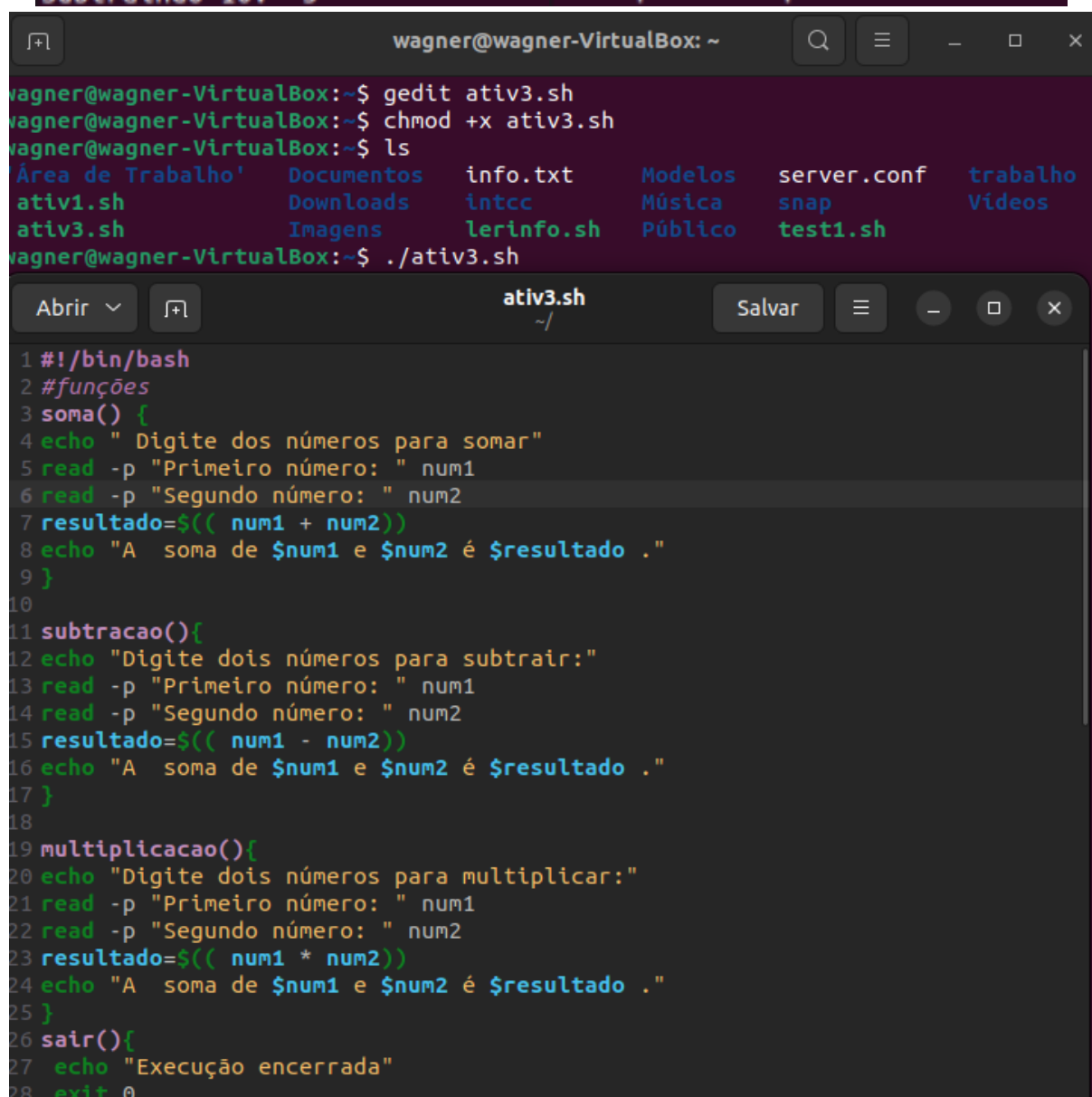
```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit info.txt  
1 Wagner Donizete  
2 27 anos  
3 Alfenas - MG  
4 hobbies: Musculação, programação
```

```
wagner@wagner-VirtualBox:~$ gedit lerinfo.sh
wagner@wagner-VirtualBox:~$ chmod +x lerinfo.sh
wagner@wagner-VirtualBox:~$ ./lerinfo.sh
Linha 1: Wagner Donizete
Linha 2: 27 anos
Linha 3: Alfenas - MG
Linha 4: hobbies: Musculação, programação
wagner@wagner-VirtualBox:~$ gedit lerinfo.sh
```



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

```
Digite um número entre 1 e 4: 4 Digite um número entre 1 e 4: 1
Digite um número: 5           Digite um número: 10
Subtraindo 10: -5             Multiplicando por 10: 100
```



```
wagner@wagner-VirtualBox:~$ gedit ativ3.sh
wagner@wagner-VirtualBox:~$ chmod +x ativ3.sh
wagner@wagner-VirtualBox:~$ ls
'Área de Trabalho'  Documentos  info.txt    Modelos     server.conf  trabalho
ativ1.sh            Downloads  intcc       Música      snap         Videos
ativ3.sh            Imagens    lerinfo.sh  Público     test1.sh
wagner@wagner-VirtualBox:~$ ./ativ3.sh
```

```
1 #!/bin/bash
2 #funções
3 soma() {
4 echo " Digite dos números para somar"
5 read -p "Primeiro número: " num1
6 read -p "Segundo número: " num2
7 resultado=$(( num1 + num2))
8 echo "A soma de $num1 e $num2 é $resultado ."
9 }
10
11 subtracao(){
12 echo "Digite dois números para subtrair:"
13 read -p "Primeiro número: " num1
14 read -p "Segundo número: " num2
15 resultado=$(( num1 - num2))
16 echo "A soma de $num1 e $num2 é $resultado ."
17 }
18
19 multiplicacao(){
20 echo "Digite dois números para multiplicar:"
21 read -p "Primeiro número: " num1
22 read -p "Segundo número: " num2
23 resultado=$(( num1 * num2))
24 echo "A soma de $num1 e $num2 é $resultado ."
25 }
26 sair(){
27 echo "Execução encerrada"
28 exit 0
```

```

31 echo "Escolha uma opção:"
32 echo "1) soma"
33 echo "2) Subtração"
34 echo "3) Multiplicação"
35 echo "4) sair"
36
37 read -p "Digite o número da opção desejada: " opcao
38
39 #comando case
40 case $opcao in
41 1)
42     soma
43     ;;
44
45 2)
46     subtracao
47     ;;
48 3)
49     multiplicacao
50     ;;
51 4)
52     sair
53     ;;
54 *)
55     echo "Opção inválida! Por favor, escolha uma opção válida."
56     ;;
57 esac
58
59

```

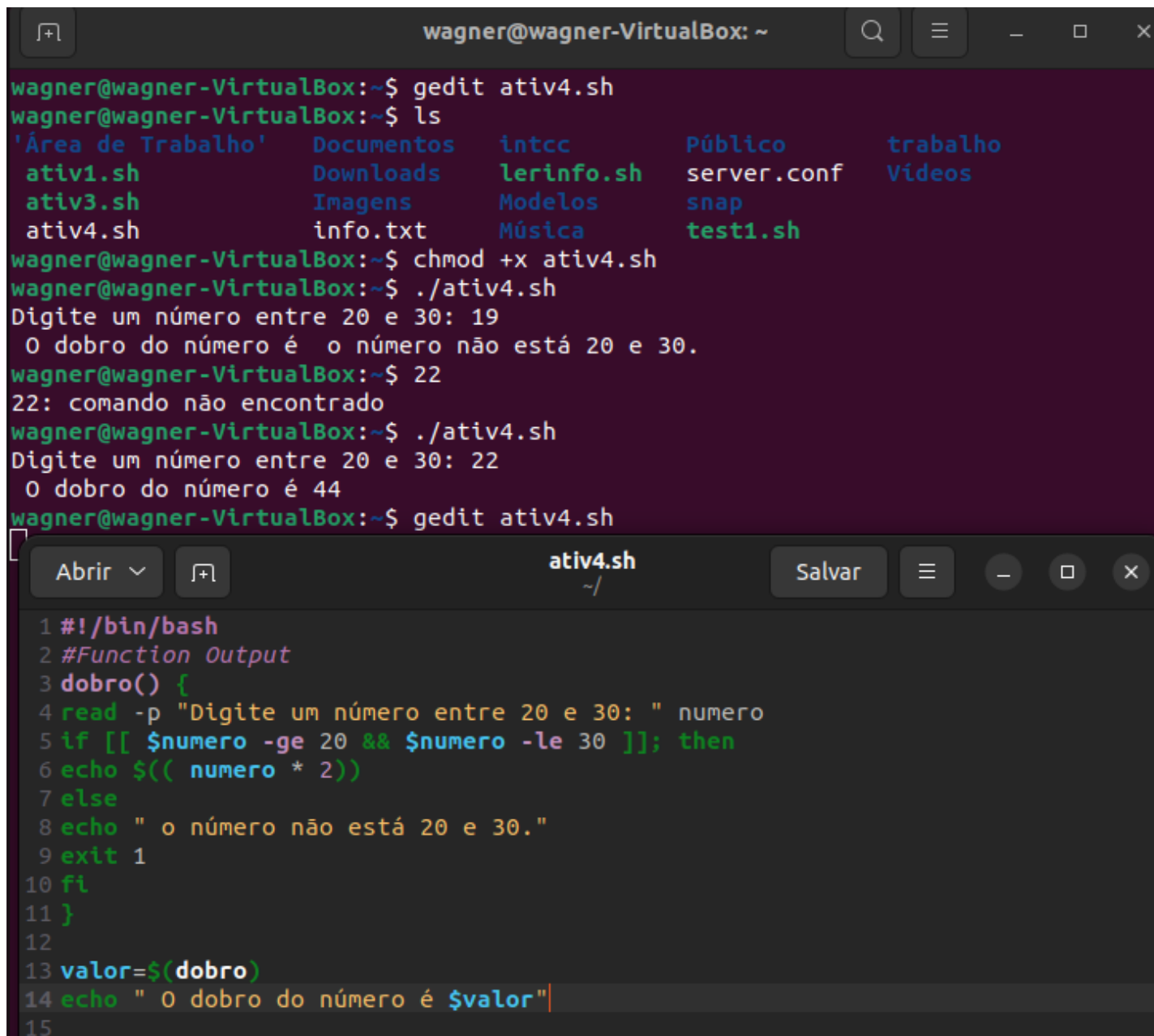
```

wagner@wagner-VirtualBox:~$ ./ativ3.sh
Escolha uma opção:
1) soma
2) Subtração
3) Multiplicação
4) sair
Digite o número da opção desejada: 1
  Digite dos números para somar
Primeiro número: 99
Segundo número: 11
A soma de 99 e 11 é 110 .
wagner@wagner-VirtualBox:~$ 2
2: comando não encontrado
wagner@wagner-VirtualBox:~$ ./ativ3.sh
Escolha uma opção:
1) soma
2) Subtração
3) Multiplicação
4) sair
Digite o número da opção desejada: 2
  Digite dois números para subtrair:
Primeiro número: 33
Segundo número: 40
A soma de 33 e 40 é -7 .
wagner@wagner-VirtualBox:~$ ./ativ3.sh
Escolha uma opção:
1) soma
2) Subtração
3) Multiplicação
4) sair
Digite o número da opção desejada: 3
  Digite dois números para multiplicar:
Primeiro número: 3
Segundo número: 3
A soma de 3 e 3 é 9 .

```

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

```
1#!/bin/bash
2#Function Output
3
4function quadrado {
5    read -p "Digite um número entre 20 e 30: " numero
6    echo $[ $numero * $numero ]
7}
8
9#Atribuindo o valor da função à variável do shell valor:
10 valor=`quadrado`
11
12#Mostrando o valor:
13 echo "O quadrado do número é $valor"
```



The screenshot shows a terminal window titled 'wagner@wagner-VirtualBox: ~'. The user runs 'gedit ativ4.sh' to open a file. Then they run 'ls' showing a directory listing with files like 'ativ1.sh', 'ativ3.sh', 'ativ4.sh', 'Downloads', 'Imagens', 'info.txt', 'intcc', 'lerinfo.sh', 'Modelos', 'Música', 'Público', 'server.conf', 'snap', 'test1.sh', 'trabalho', and 'Vídeos'. They then run 'chmod +x ativ4.sh' and './ativ4.sh'. The script prompts for a number between 20 and 30. When the user enters '19', the script outputs 'O dobro do número é o número não está 20 e 30.'. When the user enters '22', the script outputs '22: comando não encontrado'. They then run './ativ4.sh' again, enter '22', and the script outputs 'O dobro do número é 44'. Finally, they run 'gedit ativ4.sh' to open the file in a text editor. The editor shows the modified script content.

```
1#!/bin/bash
2#Function Output
3dobro() {
4    read -p "Digite um número entre 20 e 30: " numero
5    if [[ $numero -ge 20 && $numero -le 30 ]]; then
6        echo $(( numero * 2))
7    else
8        echo " o número não está 20 e 30."
9        exit 1
10    fi
11}
12
13 valor=$(dobro)
14 echo " O dobro do número é $valor"
15
```

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

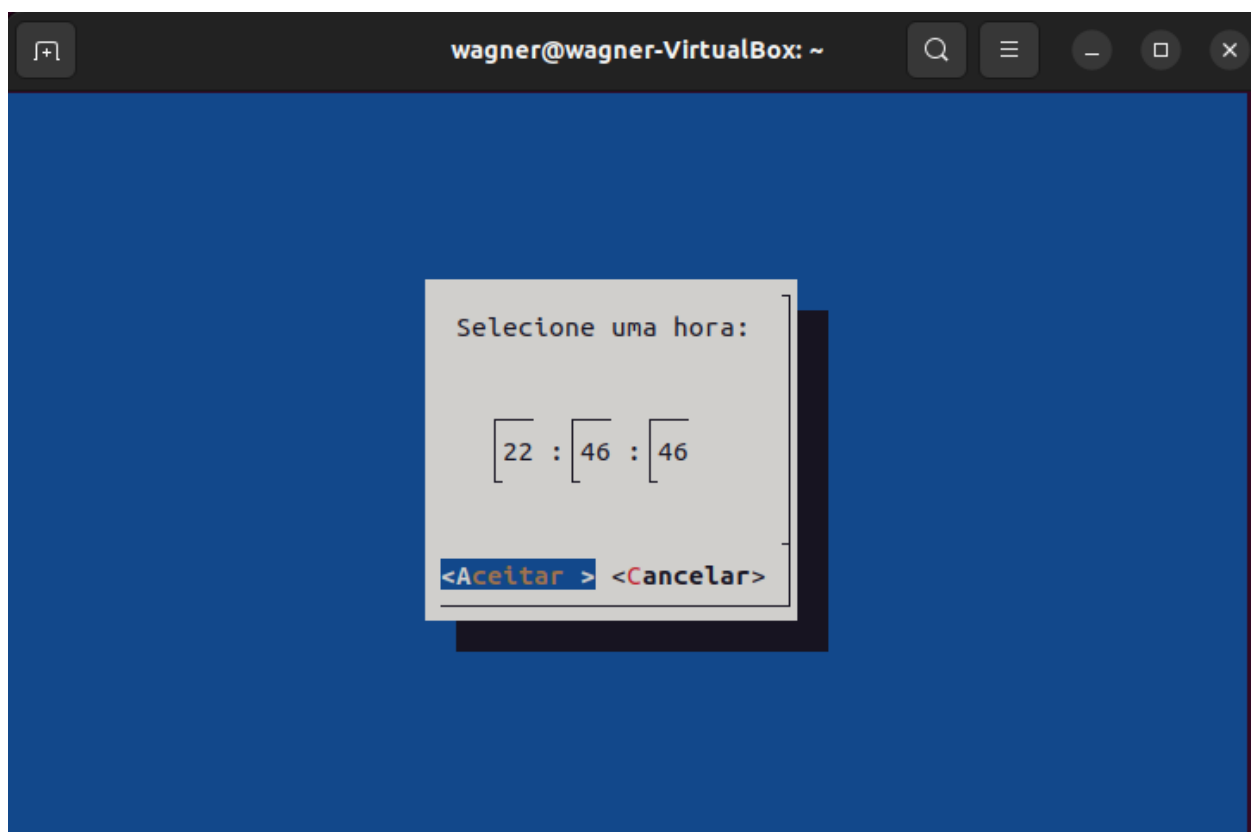
```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit ativ5.sh  
wagner@wagner-VirtualBox:~$ chmod +x ativ5.sh  
wagner@wagner-VirtualBox:~$ ./ativ5.sh  
wagner@wagner-VirtualBox:~$ gedit ativ5.sh
```

```
ativ5.sh  
#!/bin/bash  
# Utiliza o whiptail para solicitar um número ao usuário  
numero=$(whiptail --inputbox "Digite um número:" 10 40 --title  
"Calculadora" 3>&1 1>&2 2>&3)  
# Verifica se o usuário cancelou a entrada  
if [ $? -ne 0 ]; then  
    whiptail --msgbox "Nenhum número foi fornecido. Cancelando." 10 40  
    exit 1  
fi  
# Verifica se o valor fornecido é um número  
if ! [[ $numero =~ ^[0-9]+$ ]]; then  
    whiptail --msgbox "Por favor, digite apenas números inteiros." 10 40  
    exit 1  
fi  
# Calcula o dobro do número  
dobro=$((numero * 2))  
# Exibe o resultado ao usuário  
whiptail --msgbox "O dobro de $numero é $dobro." 10 40
```



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

```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit ativ6.sh  
wagner@wagner-VirtualBox:~$ chmod +x ativ6.sh  
ativ6.sh  
1 #!/bin/bash  
2  
3 # Utiliza o comando dialog para solicitar uma hora  
4 dialog --timebox "Selecione uma hora:" 0 0  
5
```



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

```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit ativ7.sh  
wagner@wagner-VirtualBox:~$ chmod +x ativ7.sh  
wagner@wagner-VirtualBox:~$ ./ativ7.sh  
Frutas selecionadas:  
wagner@wagner-VirtualBox:~$ ./ativ7.sh  
Frutas selecionadas:  
- "Maçã"  
- "Morango"  
- "Abacaxi"
```

```
Abrir  ▾  ativ7.sh  Salvar  ≡  _  □  ×
~/
1 #!/bin/bash
2 # Função para exibir o checklist de frutas
3 exibir_checklist_frutas() {
4     whiptail --title "Checklist de Frutas" --checklist \
5     "Selecione suas frutas favoritas:" 15 50 6 \
6     "Maçã" "" off \
7     "Banana" "" off \
8     "Morango" "" off \
9     "Uva" "" off \
10    "Abacaxi" "" off \
11    "Melancia" "" off \
12    3>&1 1>&2 2>&3
13 }
14 # Exibe o checklist de frutas e armazena a seleção do usuário
15 frutas_selecionadas=$(exibir_checklist_frutas)
16 # Verifica se o usuário cancelou a seleção
17 if [ $? -ne 0 ]; then
18     echo "Seleção cancelada."
19     exit 1
20 fi
21 # Exibe as frutas selecionadas
22 echo "Frutas selecionadas:"
23 for fruta in $frutas_selecionadas; do
24     echo "- $fruta"
25 done
26
```

