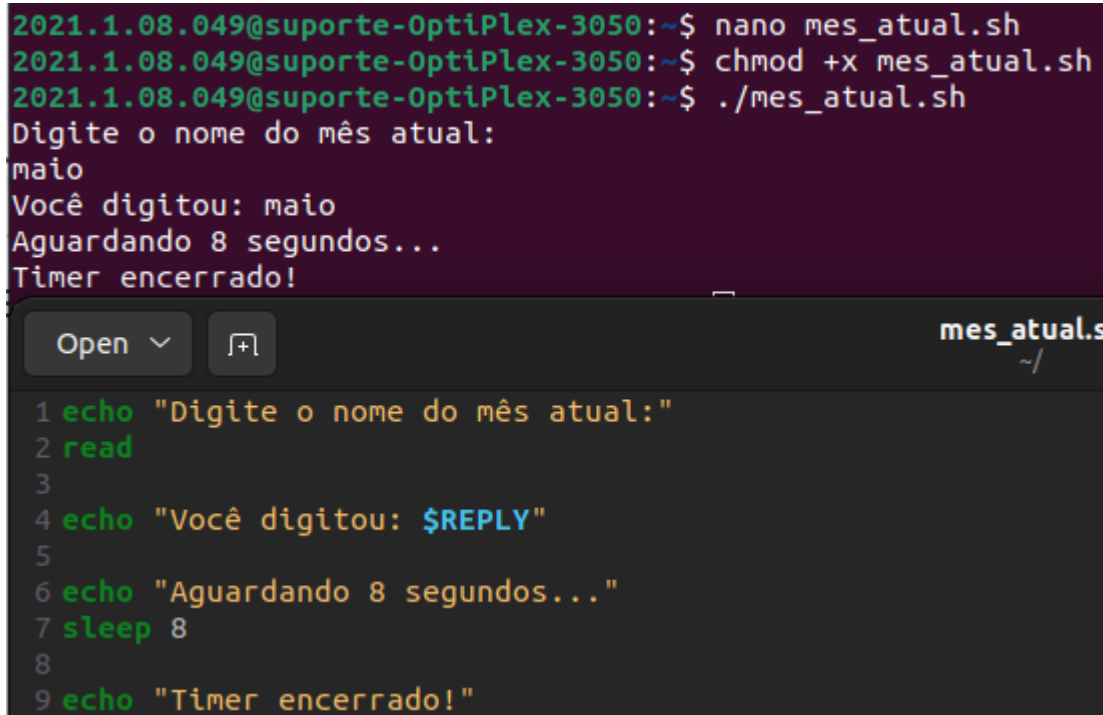


1) Qual variável de ambiente guarda o valor lido pelo comando read, caso não tenha nenhuma criada para guardar o valor? Faça um script utilizando essa variável de ambiente, onde a entrada será o nome do mês atual. Mostre a execução. Depois acrescente um timer de 8 segundos, como mostrado em aula e apresente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

Se comando read for usado sem atribuir o valor lido a uma variável específica, o valor é armazenado na variável de ambiente \$REPLY.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ nano mes_atual.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x mes_atual.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./mes_atual.sh
Digite o nome do mês atual:
maio
Você digitou: maio
Aguardando 8 segundos...
Timer encerrado!
```



```
1 echo "Digite o nome do mês atual:"
2 read
3
4 echo "Você digitou: $REPLY"
5
6 echo "Aguardando 8 segundos..."
7 sleep 8
8
9 echo "Timer encerrado!"
```

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.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ echo "Nome: Felipe Correa" > sobremim.txt
echo "Idade: 21 anos" >> sobremim.txt
echo "Cidade: Porto Ferreira, São Paulo" >> sobremim.txt
echo "Hobbies: cozinhar, ler, fazer exercicios fisicos" >> sobremim.txt

2021.1.08.049@suporte-OptiPlex-3050:~$ while IFS= read -r line; do echo "$line"; done < sobremim.txt
Nome: Felipe Correa
Idade: 21 anos
Cidade: Porto Ferreira, São Paulo
Hobbies: cozinhar, ler, fazer exercicios fisicos
```

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

```
1
2 funcao_boas_vindas() {
3     echo "Bem-vindo ao script!"
4 }
5
6 funcao_mostrar_data() {
7     data_atual=$(date +"%A, %d de %B de %Y")
8     echo "Hoje é $data_atual"
9 }
10
11 funcao_despedida() {
12     echo "Obrigado por usar o script. Até mais!"
13 }
14
15 funcao_opcao_invalida() {
16     echo "Opção inválida. Por favor, escolha uma opção válida."
17 }
18
19 principal() {
20     echo "Escolha uma opção:"
21     echo "1 - Mostrar mensagem de boas-vindas"
22     echo "2 - Mostrar a data atual"
23     echo "3 - Sair"
24
25     read opcao
26
27     case $opcao in
28         1) funcao_boas_vindas ;;
29         2) funcao_mostrar_data ;;
30         3) funcao_despedida; exit ;;
31         *) funcao_opcao_invalida ;;
32     esac
33 }
34
35 principal
```

2021.1.08.049@suporte-OptiPlex-3050:~\$ nano script\_case.sh  
2021.1.08.049@suporte-OptiPlex-3050:~\$ chmod +x script\_case.sh  
2021.1.08.049@suporte-OptiPlex-3050:~\$ ./script\_case.sh  
Escolha uma opção:  
1 - Mostrar mensagem de boas-vindas  
2 - Mostrar a data atual  
3 - Sair  
1  
Bem-vindo ao script!  
2021.1.08.049@suporte-OptiPlex-3050:~\$

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

```
2021.1.08.049@suporte-OptiPlex-3050:~$ function dobro {
    read -p "Digite um número entre 20 e 30: " numero

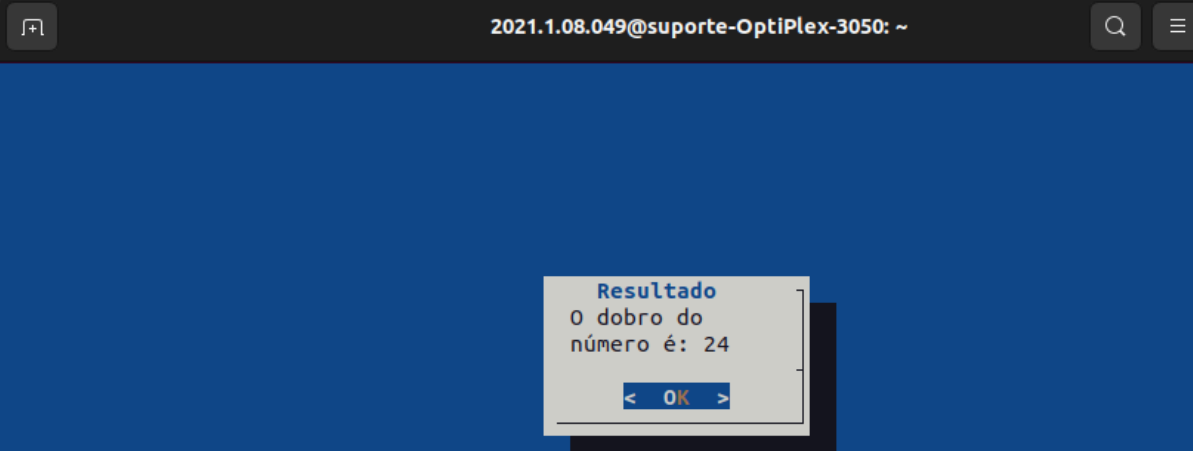
    if (( numero >= 20 && numero <= 30 )); then
        echo $((numero * 2))
    else
        echo "O número não está dentro do intervalo de 20 a 30."
    fi
}

# Chamando a função e atribuindo seu valor à variável do shell "valor":
valor=$(dobro)

# Mostrando o valor:
echo "O dobro do número é $valor"
Digite um número entre 20 e 30: 22
O dobro do número é 44
2021.1.08.049@suporte-OptiPlex-3050:~$
```

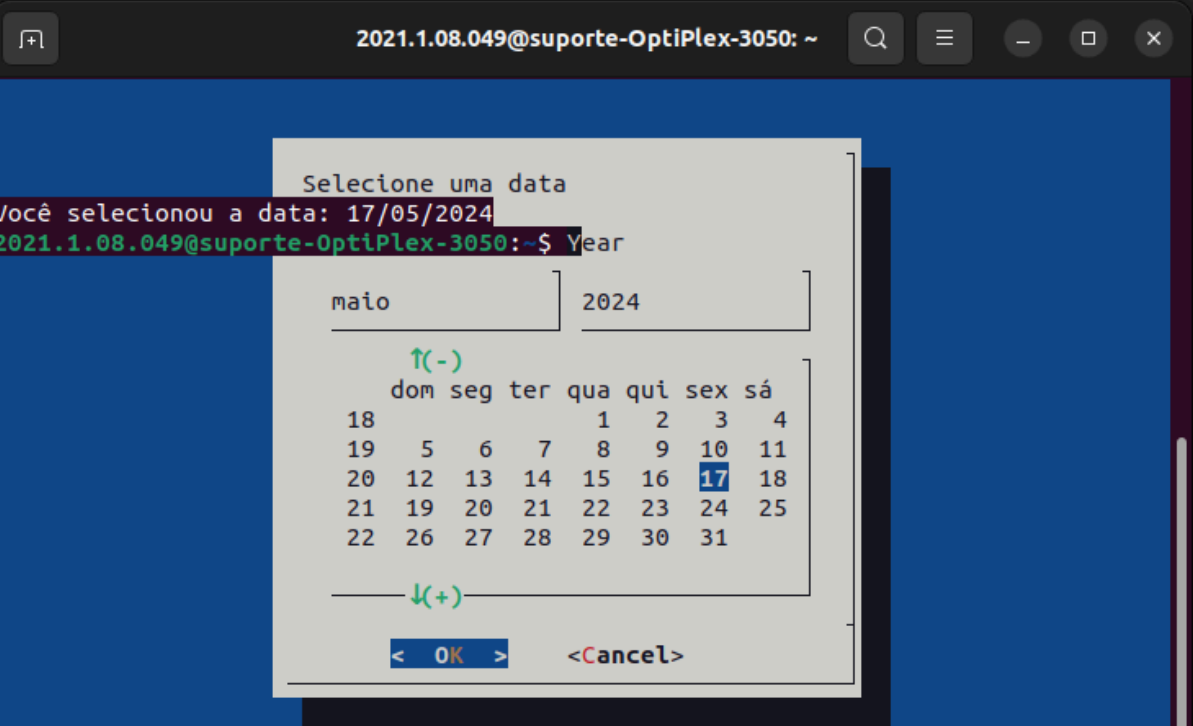
5) Crie um script como whiptail, receba um número e mostre o dobro daquele número.

```
2 numero=$(dialog --stdout --title "Insira um número" --inputbox "Digite um número:" 0 0)
3
4 if [ $? -ne 0 ]; then
5     echo "Operação cancelada."
6     exit 1
7 fi
8
9 if ! [[ "$numero" =~ ^[0-9]+$ ]]; then
10     dialog --title "Erro" --msgbox "Por favor, insira um número válido." 0 0
11     exit 1
12 fi
13
14 dobro=$((numero * 2))
15
16 dialog --title "Resultado" --msgbox "O dobro do número é: $dobro" 0 0
```



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

```
1
2 data=$(dialog --stdout --calendar "Selecione uma data" 0 0)
3
4 if [ $? -ne 0 ]; then
5     echo "Operação cancelada."
6     exit 1
7 fi
8
9 echo "Você selecionou a data: $data"
```



The screenshot shows a terminal window with a dark background. The terminal output displays the execution of a script that uses the `dialog` command to select a date. The dialog box is titled "Selecione uma data" and shows a calendar for May 2024. The date 17 is selected. The terminal output shows the selected date as 17/05/2024.

2021.1.08.049@suporte-OptiPlex-3050: ~

Você selecionou a data: 17/05/2024

2021.1.08.049@suporte-OptiPlex-3050:~\$ Year

maio 2024

↑(-)

	dom	seg	ter	qua	qui	sex	sá
18				1	2	3	4
19	5	6	7	8	9	10	11
20	12	13	14	15	16	17	18
21	19	20	21	22	23	24	25
22	26	27	28	29	30	31	

↓(+)

< OK > <Cancel>

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

```
2021.1.08.049@suporte-OptiPlex-3050:~$ ./script_checklist.sh

script_case.sh × script_dialog.sh × script_calendar.sh × script_checklist.sh ×

1
2 itens=(
3     "Limpar a casa"
4     "Fazer compras de supermercado"
5     "Lavar roupa"
6     "Pagar contas"
7     "Cuidar das plantas"
8 )
9
10 selecionados=()
11
12 selecionados=$(dialog --stdout --separate-output --checklist "Lista de
13     Tarefas" 0 0 0 \
14     "${itens[@]}")
15 if [ $? -ne 0 ]; then
16     echo "Operação cancelada."
17     exit 1
18 fi
19
20 echo "Itens selecionados:"
21 for item in "${selecionados[@]"; do
22     echo "- $item"
23 done
```