



## Introdução à Ciência da Computação – Lista 6 Shell script – parte 3

Nome: Wagner Donizete Gonçalves RA: 2019.1.08.047

- 1) Crie um script chamado scriptaritmetico, com uma operação aritmética arbitrária usando pelo menos 4 variáveis, realizando uma operação de divisão cujo resultado não seja um número inteiro. Execute o script e mostre o resultado.

```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit scriptaritmetico.sh  
wagner@wagner-VirtualBox:~$ chmod a+x scriptaritmetico.sh  
wagner@wagner-VirtualBox:~$ ./scriptaritmetico.sh  
O resultado da operação é: 9.50  
wagner@wagner-VirtualBox:~$ gedit scriptaritmetico.sh
```

```
1 #!/bin/bash  
2 #atividade1  
3 a=15  
4 b=4  
5 c=2  
6 resultado=$(echo "scale=2; ($a+$b) / $c" | bc)  
7 echo "O resultado da operação é: $resultado"
```

Qual o recurso a ser utilizado caso você queira que o valor não inteiro apareça no resultado?

Comando | **bc**

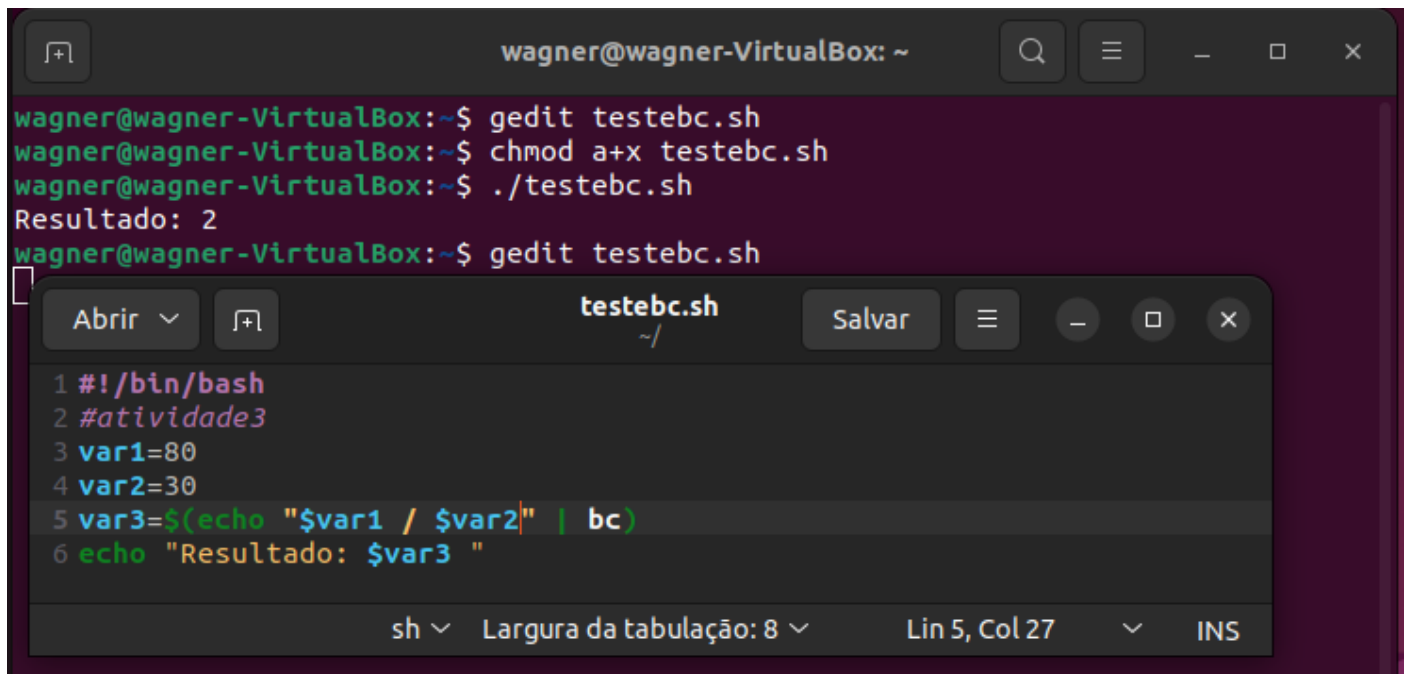
Qual variável eu uso para isso?

**Scale=2**

- 2) Ponha em execução a calculadora bc. Mostre o uso da variável scale, exibindo um resultado de operação aritmética com 6 casas decimais.

```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ bc -q  
scale=6  
7/2  
3.500000
```

- 3) Crie um script simples chamado testebc, em que você utilize a calculadora bc dentro dele, envolvendo o uso de algumas variáveis e a operação de divisão, com o direcionamento via pipe. Execute o script, mostrando o resultado.



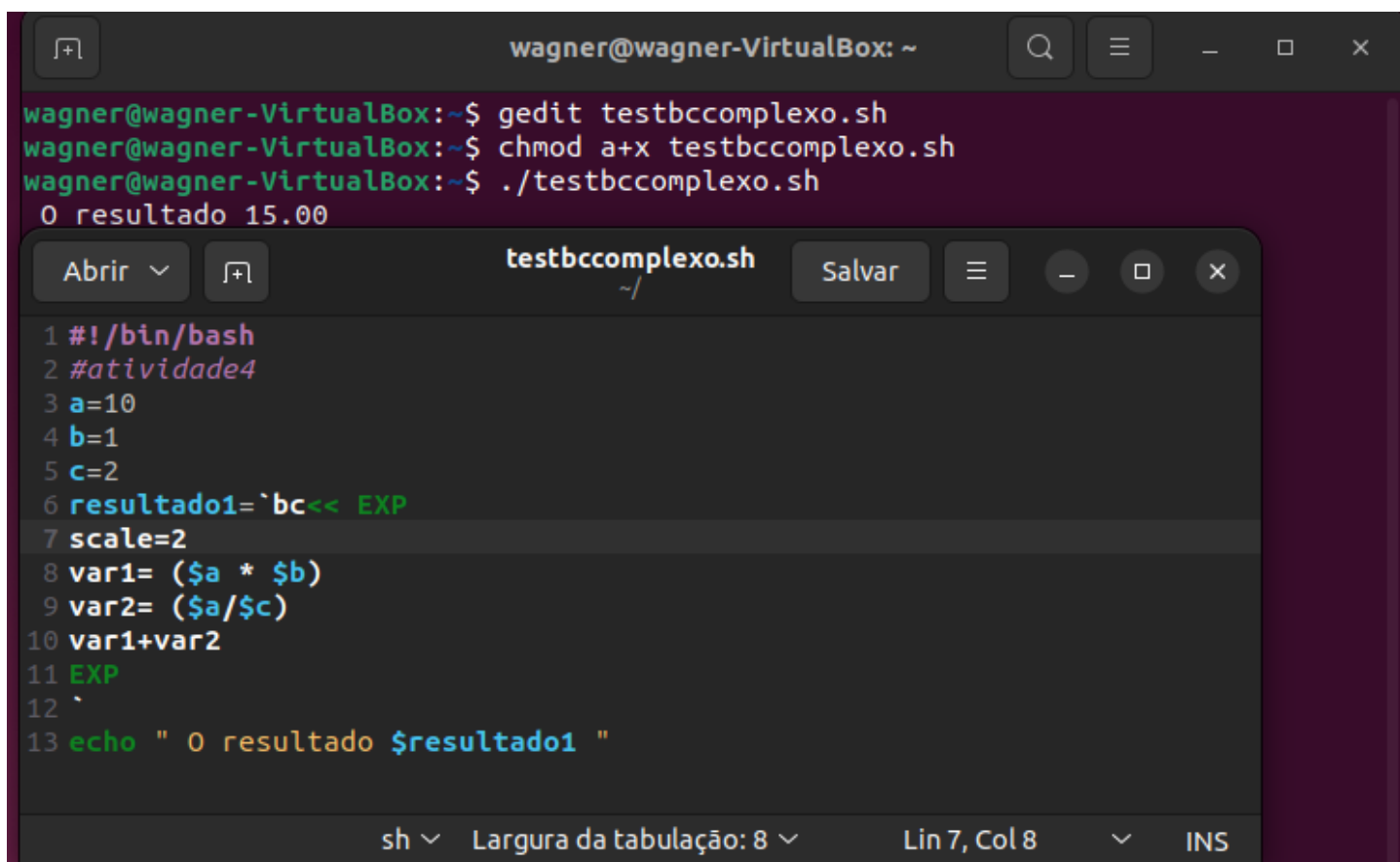
The image shows a terminal window and an editor window for a script named testebc.sh. The terminal window shows the following commands and output:

```
wagner@wagner-VirtualBox:~$ gedit testebc.sh
wagner@wagner-VirtualBox:~$ chmod a+x testebc.sh
wagner@wagner-VirtualBox:~$ ./testebc.sh
Resultado: 2
wagner@wagner-VirtualBox:~$ gedit testebc.sh
```

The editor window shows the content of testebc.sh:

```
1 #!/bin/bash
2 #atividade3
3 var1=80
4 var2=30
5 var3=$(echo "$var1 / $var2" | bc)
6 echo "Resultado: $var3 "
```

- 4) Crie um script chamado testbccomplexo, em que você utilize operações aritméticas diversas com a calculadora bc (pelo menos duas), armazenando os resultados em variáveis, como mostrado na aula. Neste caso, utilize a técnica de redirecionamento de entrada inline. Execute o script, mostrando o resultado.



The image shows a terminal window and an editor window for a script named testbccomplexo.sh. The terminal window shows the following commands and output:

```
wagner@wagner-VirtualBox:~$ gedit testbccomplexo.sh
wagner@wagner-VirtualBox:~$ chmod a+x testbccomplexo.sh
wagner@wagner-VirtualBox:~$ ./testbccomplexo.sh
0 resultado 15.00
```

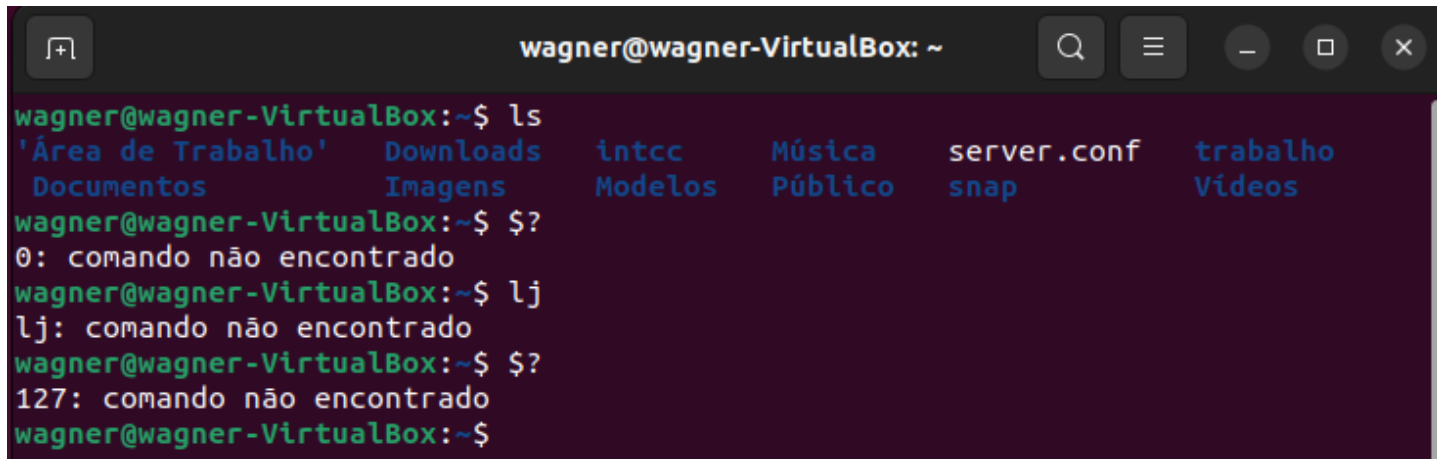
The editor window shows the content of testbccomplexo.sh:

```
1 #!/bin/bash
2 #atividade4
3 a=10
4 b=1
5 c=2
6 resultado1=`bc<< EXP
7 scale=2
8 var1= ($a * $b)
9 var2= ($a/$c)
10 var1+var2
11 EXP
12 `
13 echo " 0 resultado $resultado1 "
```

5) O que consiste o status de saída de um programa?

O status de saída de um programa, também conhecido como código de retorno ou código de saída, é um valor numérico que indica o resultado da execução desse programa. Zero quando é sucesso e um numero maior que zero para erro.

Mostre um exemplo de execução de dois comandos (um com sucesso e outro desconhecido) e verifique esse status. Mostre em tela.

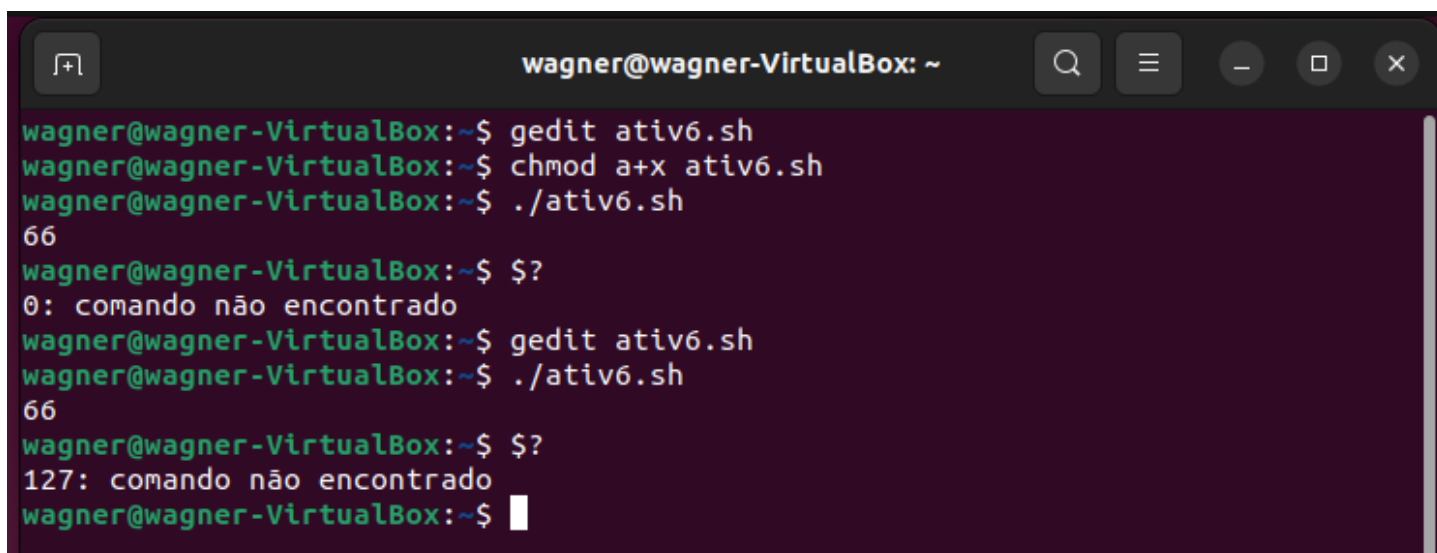


```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ ls  
'Área de Trabalho'  Downloads  intcc      Música     server.conf  trabalho  
Documentos          Imagens    Modelos    Público    snap         Vídeos  
wagner@wagner-VirtualBox:~$ $?  
0: comando não encontrado  
wagner@wagner-VirtualBox:~$ lj  
lj: comando não encontrado  
wagner@wagner-VirtualBox:~$ $?  
127: comando não encontrado  
wagner@wagner-VirtualBox:~$
```

6) Qual a função do comando exit?

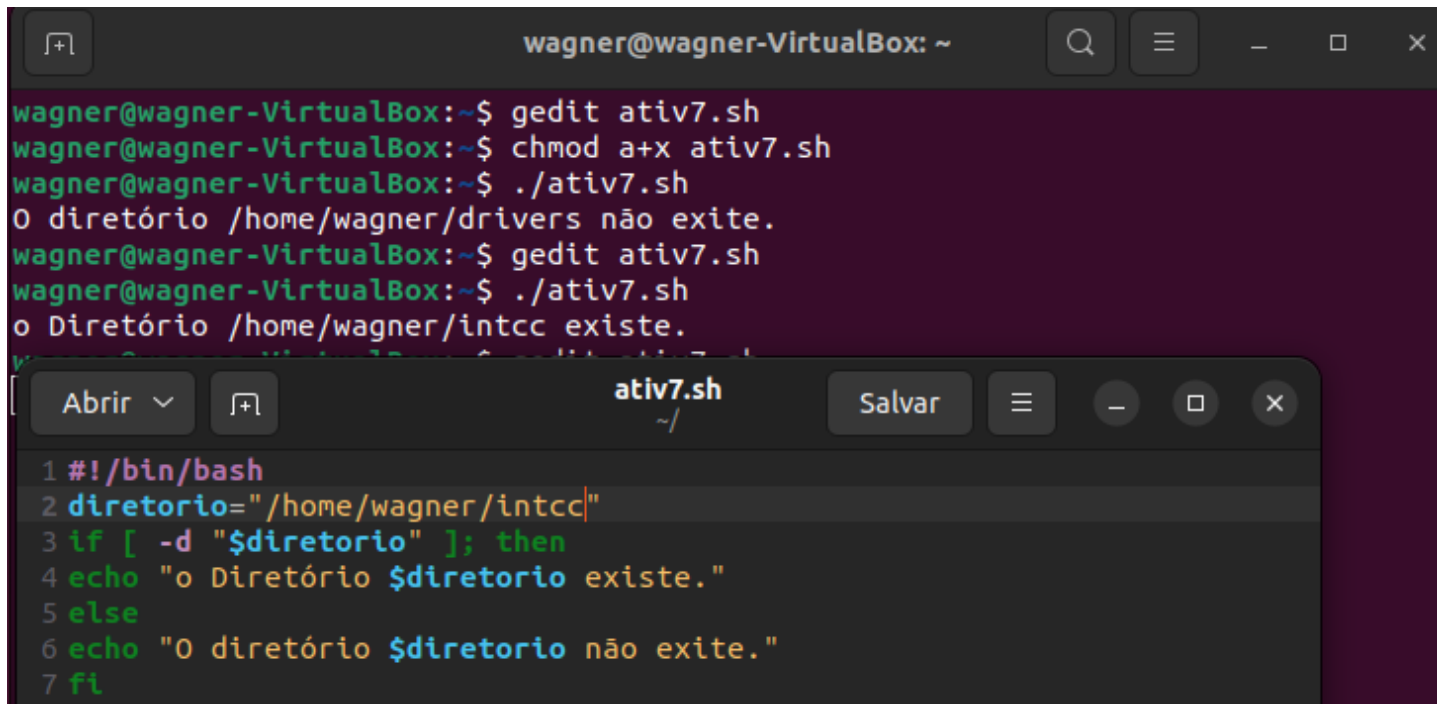
Por padrão, seu shell script finaliza com o status de saída do último comando executado no script. Utilizado para finalizar a execução de um script ou de um shell, retornando um código de saída específico.

Mostre um exemplo do uso do comando exit dentro de um script, mudando o valor padrão do status de saída. Mostre tanto o uso do exit exibindo um número qualquer até 255, quanto o valor de uma variável que você utilize no script. Execute o script e mostre o valor do status de saída em cada caso.



```
wagner@wagner-VirtualBox: ~  
wagner@wagner-VirtualBox:~$ gedit ativ6.sh  
wagner@wagner-VirtualBox:~$ chmod a+x ativ6.sh  
wagner@wagner-VirtualBox:~$ ./ativ6.sh  
66  
wagner@wagner-VirtualBox:~$ $?  
0: comando não encontrado  
wagner@wagner-VirtualBox:~$ gedit ativ6.sh  
wagner@wagner-VirtualBox:~$ ./ativ6.sh  
66  
wagner@wagner-VirtualBox:~$ $?  
127: comando não encontrado  
wagner@wagner-VirtualBox:~$
```

- 7) Crie um script simples envolvendo comandos condicionais if then else, para verificar a existência de um diretório específico no seu home. Primeiro procure um diretório inexistente, depois um diretório existente e exiba as mensagens específicas de acordo com o resultado. Execute o script e mostre em tela. ~



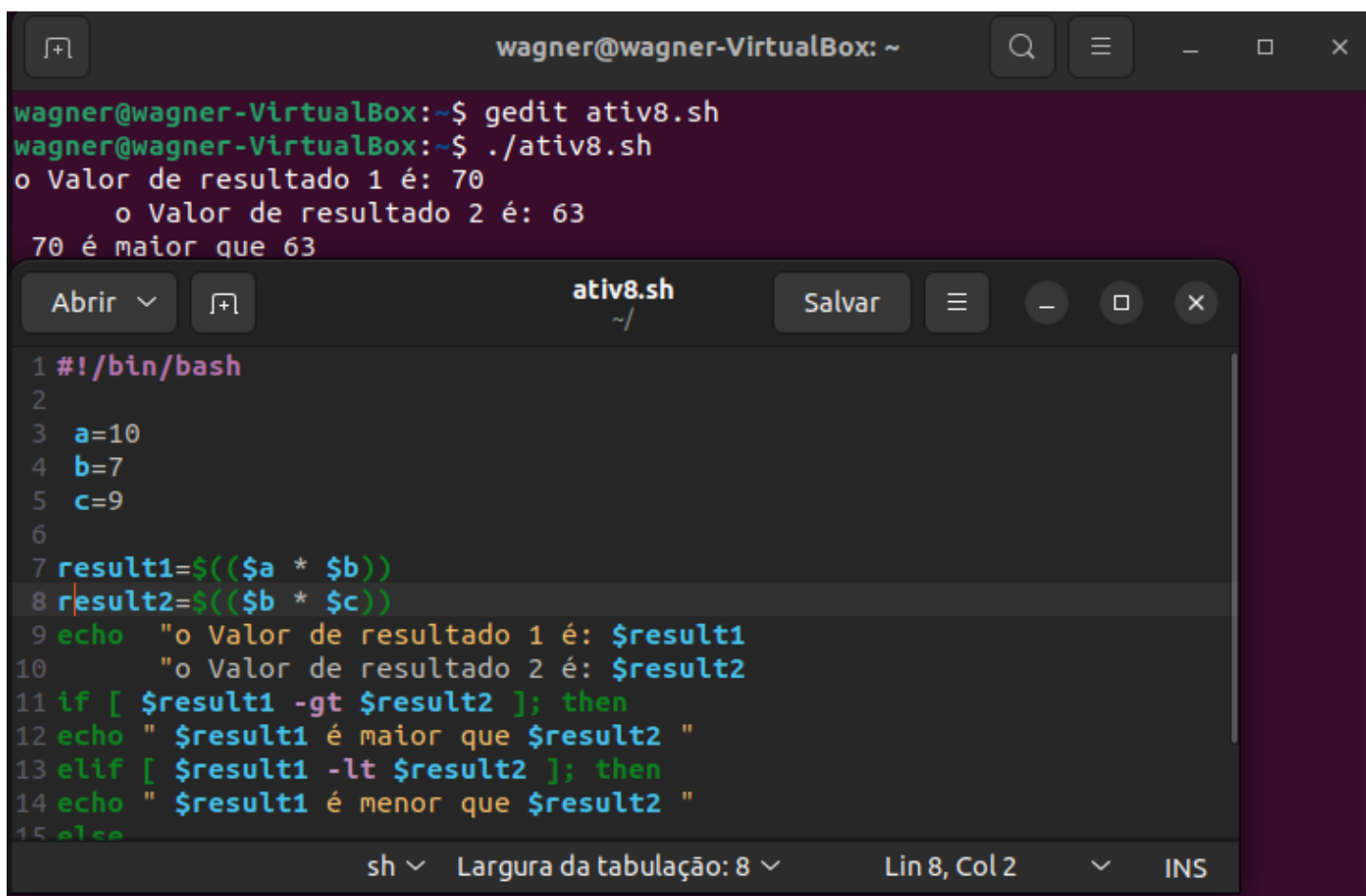
The image shows a terminal window and a gedit editor window. The terminal window, titled 'wagner@wagner-VirtualBox: ~', shows the following commands and output:

```
wagner@wagner-VirtualBox:~$ gedit ativ7.sh
wagner@wagner-VirtualBox:~$ chmod a+x ativ7.sh
wagner@wagner-VirtualBox:~$ ./ativ7.sh
O diretório /home/wagner/drivers não existe.
wagner@wagner-VirtualBox:~$ gedit ativ7.sh
wagner@wagner-VirtualBox:~$ ./ativ7.sh
o Diretório /home/wagner/intcc existe.
```

The gedit editor window, titled 'ativ7.sh', shows the following script:

```
1 #!/bin/bash
2 diretorio="/home/wagner/intcc"
3 if [ -d "$diretorio" ]; then
4 echo "o Diretório $diretorio existe."
5 else
6 echo "O diretório $diretorio não existe."
7 fi
```

- 8) Crie um script envolvendo várias condicionais usando a estrutura if then elif else, fazendo duas operações aritméticas arbitrárias, verificando o valor das variáveis que armazenam essa operação, checando se o valor da primeira é maior, menor ou igual ao valor da segunda. Execute o script e mostre o resultado em tela.



The image shows a terminal window and a gedit editor window. The terminal window, titled 'wagner@wagner-VirtualBox: ~', shows the following commands and output:

```
wagner@wagner-VirtualBox:~$ gedit ativ8.sh
wagner@wagner-VirtualBox:~$ ./ativ8.sh
o Valor de resultado 1 é: 70
    o Valor de resultado 2 é: 63
70 é maior que 63
```

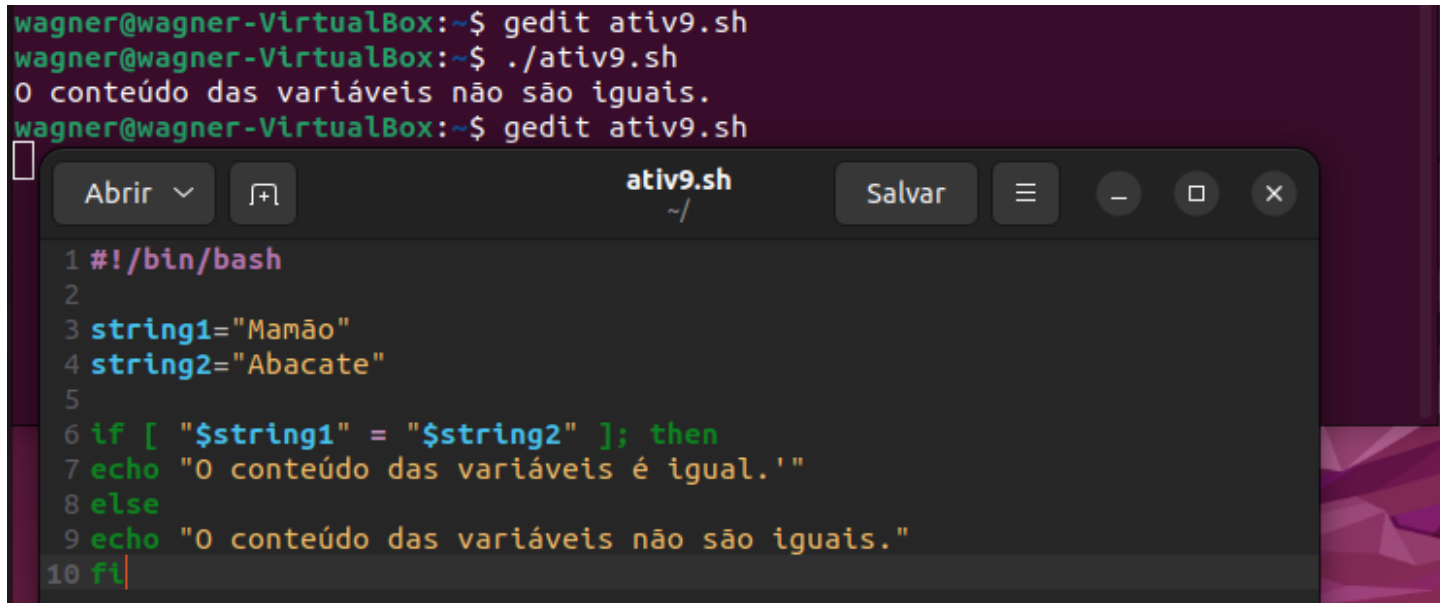
The gedit editor window, titled 'ativ8.sh', shows the following script:

```
1 #!/bin/bash
2
3 a=10
4 b=7
5 c=9
6
7 result1=$(( $a * $b ))
8 result2=$(( $b * $c ))
9 echo "o Valor de resultado 1 é: $result1"
10 echo "o Valor de resultado 2 é: $result2"
11 if [ $result1 -gt $result2 ]; then
12 echo " $result1 é maior que $result2 "
13 elif [ $result1 -lt $result2 ]; then
14 echo " $result1 é menor que $result2 "
15 else
```

The status bar at the bottom of the gedit window shows 'sh', 'Largura da tabulação: 8', 'Lin 8, Col 2', and 'INS'.

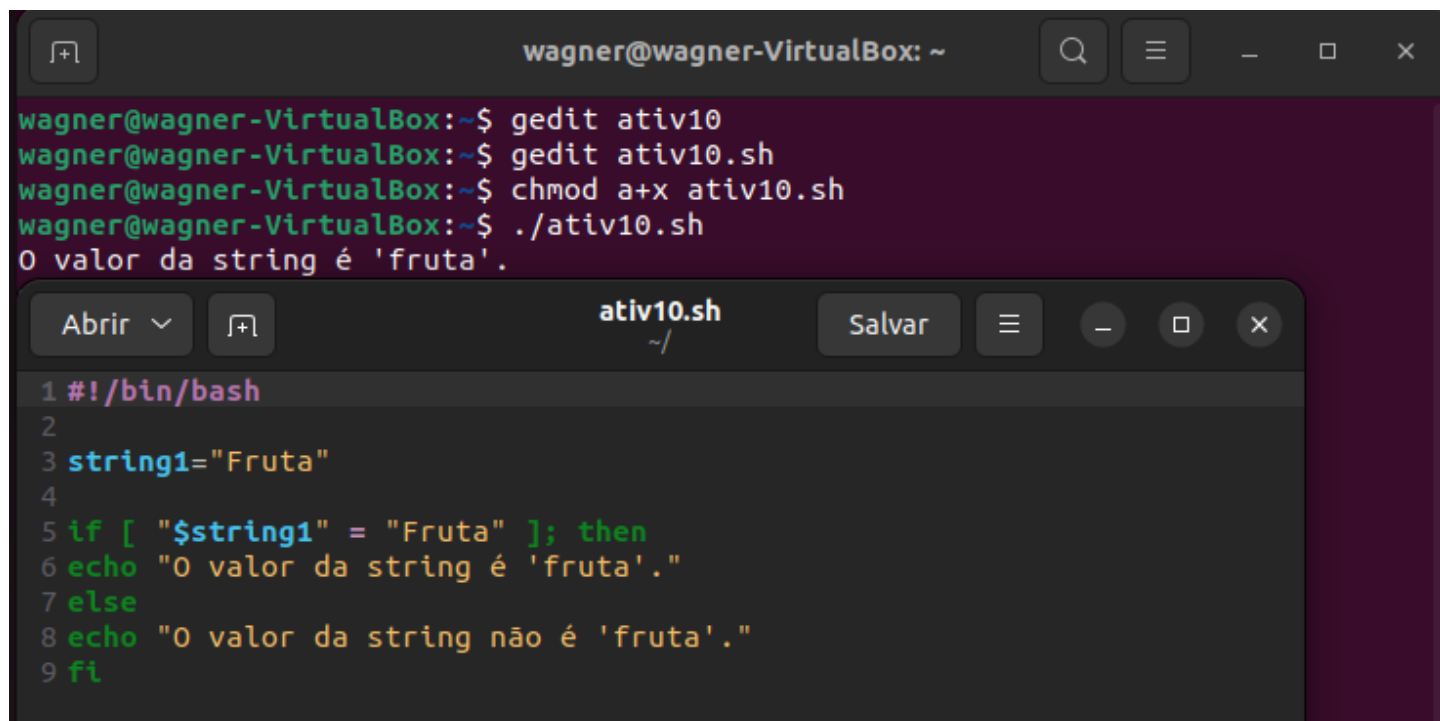
- 9) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas variáveis string arbitrárias e verificando seus valores, checando se o conteúdo das variáveis é igual. Execute o script e mostre o resultado em tela.

```
wagner@wagner-VirtualBox:~$ gedit ativ9.sh
wagner@wagner-VirtualBox:~$ ./ativ9.sh
O conteúdo das variáveis não são iguais.
wagner@wagner-VirtualBox:~$ gedit ativ9.sh
```



```
1 #!/bin/bash
2
3 string1="Mamão"
4 string2="Abacate"
5
6 if [ "$string1" = "$string2" ]; then
7 echo "O conteúdo das variáveis é igual."
8 else
9 echo "O conteúdo das variáveis não são iguais."
10 fi
```

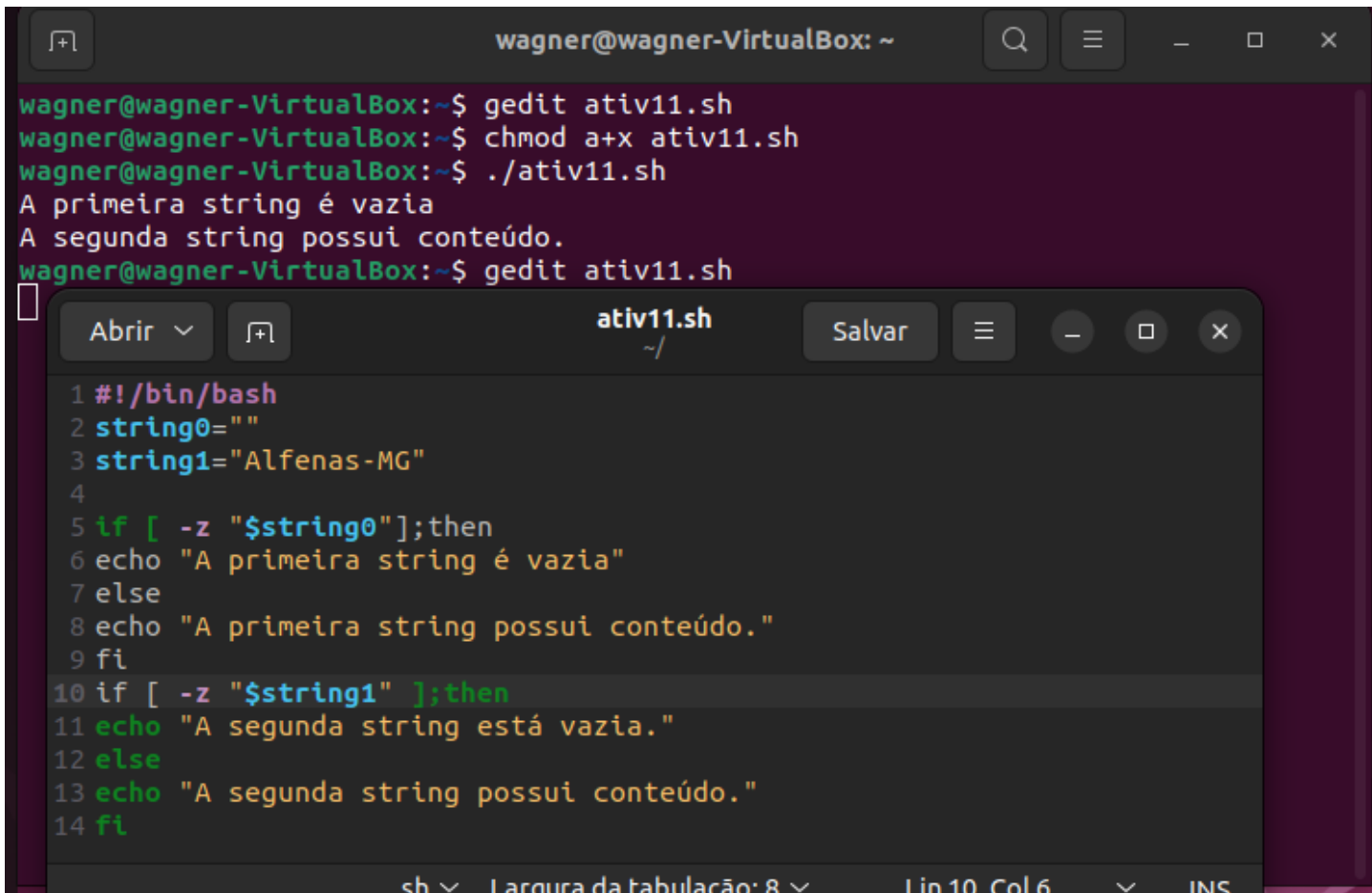
- 10) Crie um script envolvendo condicionais usando a estrutura if then else, criando uma string com um conteúdo, verificando se seu valor é "fruta". Execute o script e mostre o resultado em tela.



```
wagner@wagner-VirtualBox:~$ gedit ativ10
wagner@wagner-VirtualBox:~$ gedit ativ10.sh
wagner@wagner-VirtualBox:~$ chmod a+x ativ10.sh
wagner@wagner-VirtualBox:~$ ./ativ10.sh
O valor da string é 'fruta'.
```

```
1 #!/bin/bash
2
3 string1="Fruta"
4
5 if [ "$string1" = "Fruta" ]; then
6 echo "O valor da string é 'fruta'."
7 else
8 echo "O valor da string não é 'fruta'."
9 fi
```

- 11) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas strings, uma vazia, outra com conteúdo e verificando estes resultados (se tem conteúdo em ambos os casos).

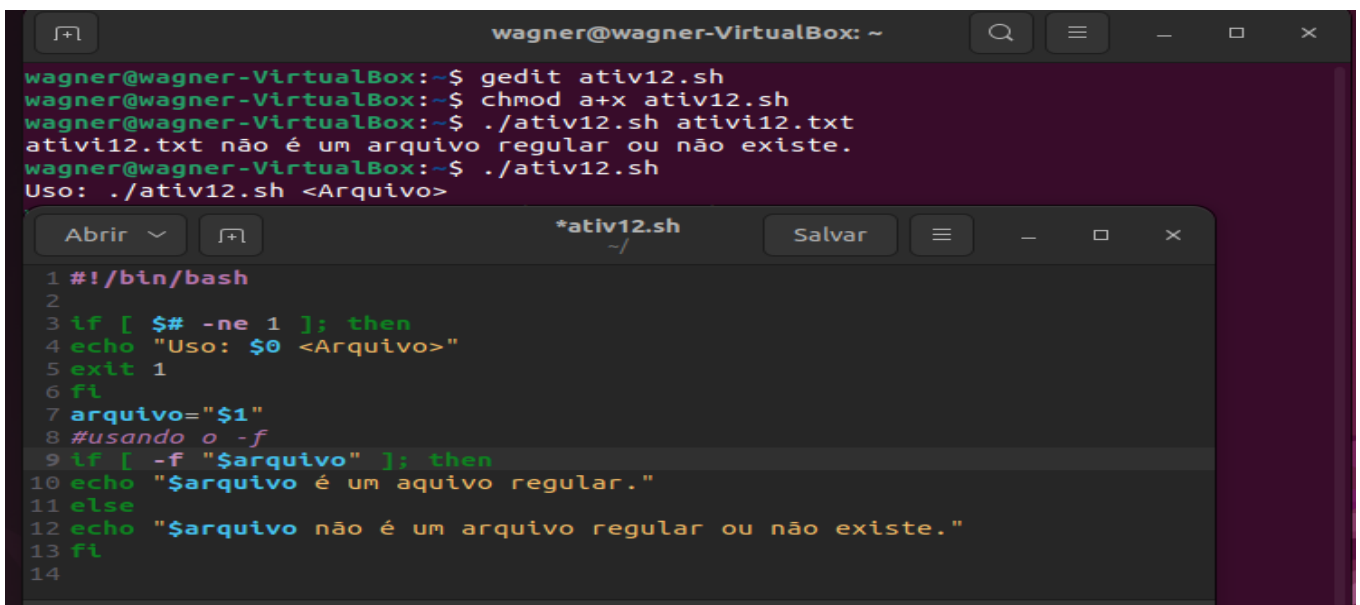


```
wagner@wagner-VirtualBox: ~$ gedit ativ11.sh
wagner@wagner-VirtualBox:~$ chmod a+x ativ11.sh
wagner@wagner-VirtualBox:~$ ./ativ11.sh
A primeira string é vazia
A segunda string possui conteúdo.
wagner@wagner-VirtualBox:~$ gedit ativ11.sh
```

```
ativ11.sh
#!/bin/bash
string0=""
string1="Alfenas-MG"
if [ -z "$string0" ]; then
echo "A primeira string é vazia"
else
echo "A primeira string possui conteúdo."
fi
if [ -z "$string1" ]; then
echo "A segunda string está vazia."
else
echo "A segunda string possui conteúdo."
fi
```

- 12) Cite 5 opções de comparações envolvendo arquivos. Escolha uma das opções e crie um script envolvendo essa opção.

1. `-e arquivo`: Verifica se o arquivo existe.
2. `-f arquivo`: Verifica se o arquivo é um arquivo regular (não um diretório ou dispositivo especial).
3. `-d arquivo`: Verifica se o arquivo é um diretório.
4. `-r arquivo`: Verifica se o arquivo tem permissão de leitura.
5. `-s arquivo`: Verifica se o arquivo tem tamanho maior que zero (ou seja, se contém dados).



```
wagner@wagner-VirtualBox:~$ gedit ativ12.sh
wagner@wagner-VirtualBox:~$ chmod a+x ativ12.sh
wagner@wagner-VirtualBox:~$ ./ativ12.sh ativ12.txt
ativ12.txt não é um arquivo regular ou não existe.
wagner@wagner-VirtualBox:~$ ./ativ12.sh
Uso: ./ativ12.sh <Arquivo>
```

```
*ativ12.sh
#!/bin/bash
if [ $# -ne 1 ]; then
echo "Uso: $0 <Arquivo>"
exit 1
fi
arquivo="$1"
#usando o -f
if [ -f "$arquivo" ]; then
echo "$arquivo é um arquivo regular."
else
echo "$arquivo não é um arquivo regular ou não existe."
fi
```