

1) No shell, podemos executar uma sequência de comandos, separados por vírgula. Porém, mais vantajoso é encapsulá-los em um arquivo de script. Escolha um editor de texto e crie um arquivo de script, com os comandos `cd /` e `ls -a`, salvando-o como um arquivo com extensão `.sh`. Em seguida, execute-o. Mostre todos os passos para a saída final desse script.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ vim script.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x script.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./script.sh
.  .. bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run
sbin snap srv swapfile sys tmp usr var
```

2) No exercício anterior, altere o script para exibir uma mensagem “Meu primeiro script”, no final dele. Em seguida, adicione um comando na mesma linha, informando quem está logado no sistema.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ vim script.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x script.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./script.sh
.  .. bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run
sbin snap srv swapfile sys tmp usr var
Meu primeiro script
2021.1.08.049 tty2          2024-04-30 11:58 (tty2)
```

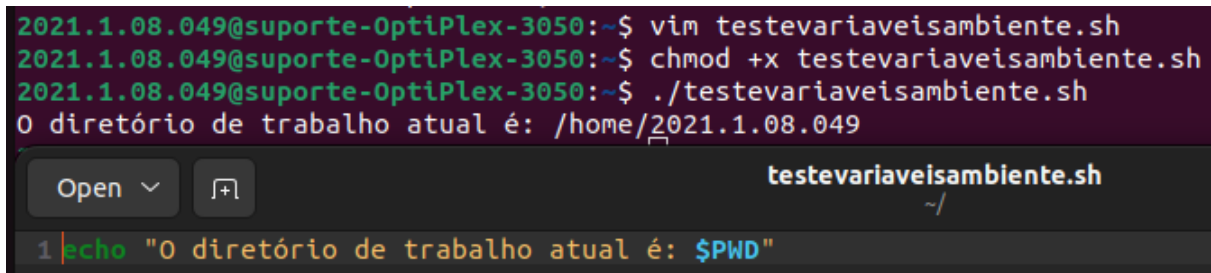
3) Crie um script chamado `testevariaveis`. Declare duas variáveis, uma numérica e uma string, com conteúdo que você desejar. Em seguida, seu script deve exibir uma frase, envolvendo o conteúdo das variáveis criadas. Execute o script.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ vim testevariaveis.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x testevariaveis.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./testevariaveis.sh
Gosto do número 42.
Esta é uma frase de exemplo: Olá, mundo!
```

4) Crie um script chamado `testevariaveisambiente`. Desenvolva o script de modo que você mostre em tela, em uma frase, o conteúdo de uma variável de ambiente à sua

escolha. Procure usar uma variável que não tenha sido mostrada na aula. Execute o script.

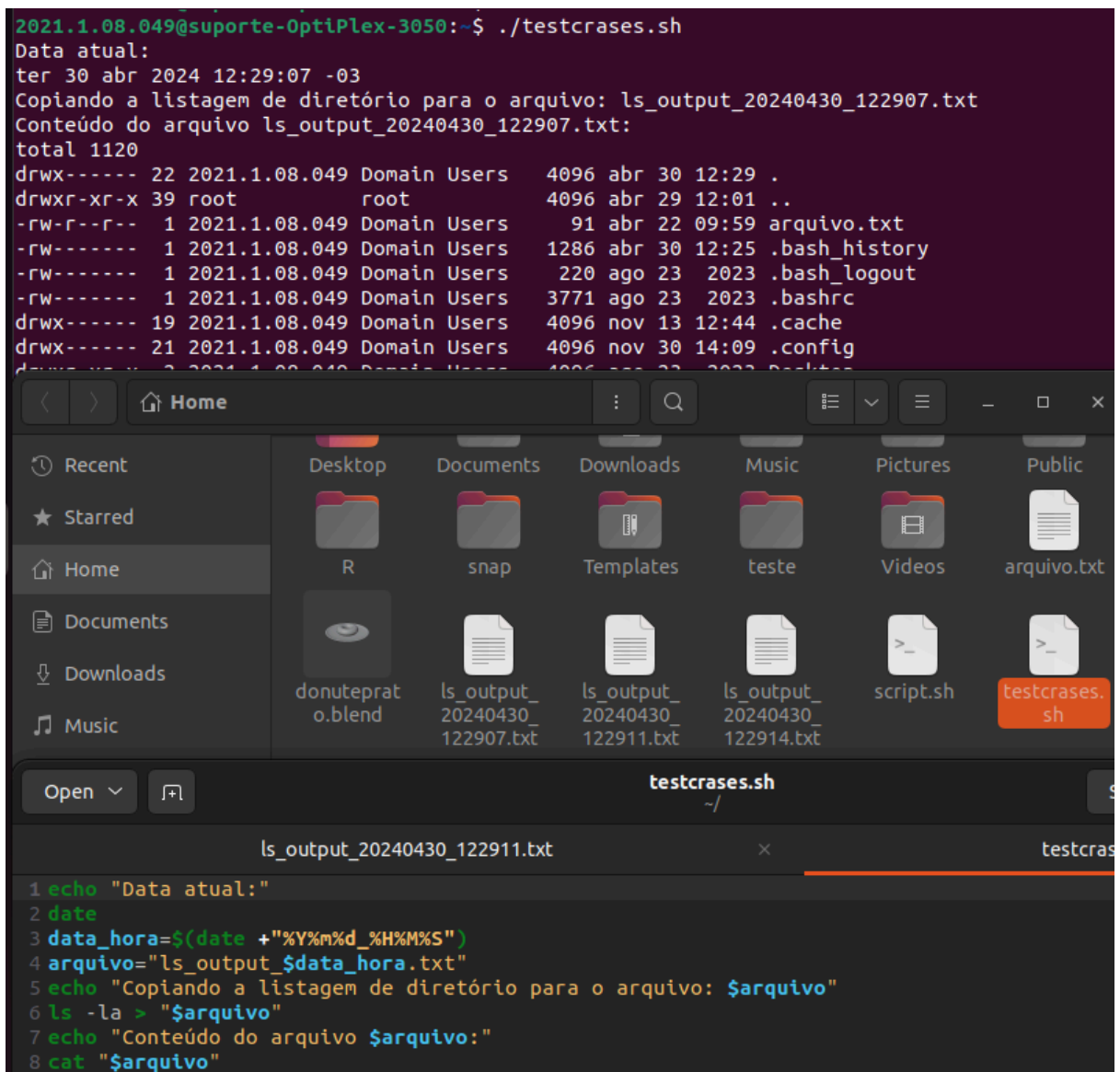
```
2021.1.08.049@suporte-OptiPlex-3050:~$ vim testevariaveisambiente.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x testevariaveisambiente.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./testevariaveisambiente.sh
O diretório de trabalho atual é: /home/2021.1.08.049
```



```
testevariaveisambiente.sh
1|echo "O diretório de trabalho atual é: $PWD"
```

5) Crie um script chamado testecrases. Desenvolva o script de modo que ele mostre inicialmente a data atual. Em seguida, o script deve copiar a listagem de diretório (ls -la) para arquivos únicos, diferenciados por data e hora. Execute o script três vezes e mostre o resultado em tela, dos diferentes arquivos.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data atual:
ter 30 abr 2024 12:29:07 -03
Copiando a listagem de diretório para o arquivo: ls_output_20240430_122907.txt
Conteúdo do arquivo ls_output_20240430_122907.txt:
total 1120
drwx----- 22 2021.1.08.049 Domain Users 4096 abr 30 12:29 .
drwxr-xr-x 39 root root 4096 abr 29 12:01 ..
-rw-r--r-- 1 2021.1.08.049 Domain Users 91 abr 22 09:59 arquivo.txt
-rw----- 1 2021.1.08.049 Domain Users 1286 abr 30 12:25 .bash_history
-rw----- 1 2021.1.08.049 Domain Users 220 ago 23 2023 .bash_logout
-rw----- 1 2021.1.08.049 Domain Users 3771 ago 23 2023 .bashrc
drwx----- 19 2021.1.08.049 Domain Users 4096 nov 13 12:44 .cache
drwx----- 21 2021.1.08.049 Domain Users 4096 nov 30 14:09 .config
drwxr-xr-x 3 2021.1.08.049 Domain Users 4096 ago 23 2023 .ssh
```



```
testecrases.sh
1|echo "Data atual:"
2|date
3|data_hora=$(date +"%Y%m%d_%H%M%S")
4|arquivo="ls_output_${data_hora}.txt"
5|echo "Copiando a listagem de diretório para o arquivo: $arquivo"
6|ls -la > "$arquivo"
7|echo "Conteúdo do arquivo $arquivo:"
8|cat "$arquivo"
```

6) Como é feito o redirecionamento do resultado de um comando para um arquivo, de modo que não se sobrescreva o conteúdo do arquivo? Dê um exemplo de como isso pode ser feito, mostrando os resultados. Procure fazer um exemplo diferente do visto na aula.

Basta utilizar o operador de redirecionamento >>, que anexará a saída do comando ao final do arquivo especificado, sem apagar o conteúdo anterior.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ echo "Conteúdo inicial do arquivo" > arquivo.txt
2021.1.08.049@suporte-OptiPlex-3050:~$ cat arquivo.txt
Conteúdo inicial do arquivo
2021.1.08.049@suporte-OptiPlex-3050:~$ date >> arquivo.txt
2021.1.08.049@suporte-OptiPlex-3050:~$ cat arquivo.txt
Conteúdo inicial do arquivo
ter 30 abr 2024 12:33:58 -03
2021.1.08.049@suporte-OptiPlex-3050:~$
```

7) Como é feito o redirecionamento do conteúdo de um arquivo para ser usado em um comando? Dê um exemplo de como isso pode ser feito, mostrando os resultados. Procure fazer um exemplo diferente do visto na aula.

O redirecionamento de conteúdo de um arquivo para ser usado em um comando é feito usando o operador de redirecionamento de entrada <, permite que seja fornecido o conteúdo de um arquivo como entrada para um comando.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ cat numeros.txt
5
10
15
20
2021.1.08.049@suporte-OptiPlex-3050:~$ awk '{soma += $1} END {print soma}' < numeros.txt
50
2021.1.08.049@suporte-OptiPlex-3050:~$
```

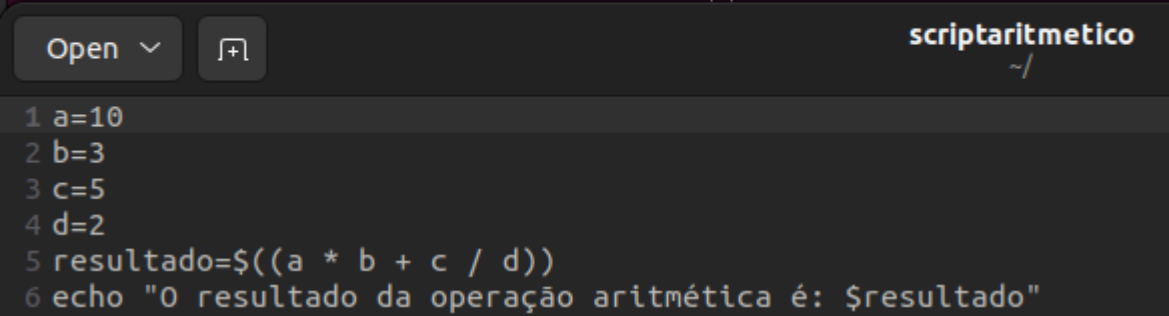
8) O que consiste o redirecionamento de entrada inline? Dê um exemplo de como isso pode ser feito, mostrando os resultados. Procure fazer um exemplo diferente do visto na aula.

É uma técnica em que é possível fornecer uma entrada diretamente no comando em vez de direcioná-la de um arquivo externo.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ wc -w << END
Este é um exemplo de redirecionamento de entrada inline.
É útil quando você precisa fornecer entrada diretamente no terminal.
END
19
2021.1.08.049@suporte-OptiPlex-3050:~$
```

9) 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.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ vim scriptaritmetico
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico
2021.1.08.049@suporte-OptiPlex-3050:~$ ./scriptaritmetico
O resultado da operação aritmética é: 32
```



The screenshot shows a terminal window with a dark background. At the top, there are three lines of command history: `vim scriptaritmetico`, `chmod +x scriptaritmetico`, and `./scriptaritmetico`. Below this, the output of the script is displayed: `O resultado da operação aritmética é: 32`. The number 32 is underlined. Below the terminal output, there is a visual representation of the script editor's interface. It shows a header bar with an 'Open' button, a file icon, and the filename 'scriptaritmetico' with a tilde indicating the current directory. Below the header, the script's content is listed line by line: `1 a=10`, `2 b=3`, `3 c=5`, `4 d=2`, `5 resultado=$((a * b + c / d))`, and `6 echo "O resultado da operação aritmética é: $resultado"`.

```
Open  [icon] scriptaritmetico ~/
1 a=10
2 b=3
3 c=5
4 d=2
5 resultado=$((a * b + c / d))
6 echo "O resultado da operação aritmética é: $resultado"
```