

Bartholomeu Vitor Neto  
2024.1.08.043

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ nano meu_script.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x meu_script.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./meu_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
2024.1.08.043@suporte-OptiPlex-3050:~$ ls -a
.  .. 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
.  .bash_history .bash_logout .cache Desktop Downloads home .java meu_script.sh .netbeans Pictures .profile receitas .ssh Videos
.  .bashrc .conflic Documents .gnupg .local Music NetBeansProjects .pkg Public snap Templates .viminfo
2024.1.08.043@suporte-OptiPlex-3050:~$
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ nano meu_script.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./meu_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
.  .. 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. Quem está logado:
2024.1.08.043 tty2      2024-05-06 10:05 (tty2)
2024.1.08.043@suporte-OptiPlex-3050:~$
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ nano testevariaveis.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x testevariaveis.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testevariaveis.sh
bash: ./testevariaveis.sh: No such file or directory
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testevariaveis.sh
A variável 'numero' contém o valor 42.
A variável 'texto' contém o texto 'Ola,Mundo!'.
2024.1.08.043@suporte-OptiPlex-3050:~$
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ nano testevariaveisambiente.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x testevariaveisambiente.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testevariaveisambiente.sh
O diretório HOME do usuário é: /home/2024.1.08.043
2024.1.08.043@suporte-OptiPlex-3050:~$
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ nano testecrases.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ chmod +x testecrases.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data e hora atual: 2024-05-06_10-54-03
Listagem de diretório copiada para lista_2024-05-06_10-54-03.txt
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data e hora atual: 2024-05-06_10-54-06
Listagem de diretório copiada para lista_2024-05-06_10-54-06.txt
2024.1.08.043@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data e hora atual: 2024-05-06_10-54-08
Listagem de diretório copiada para lista_2024-05-06_10-54-08.txt
2024.1.08.043@suporte-OptiPlex-3050:~$ cat lista_2024-05-06_10-54-03.txt
total 140
lrwxrwxrwx. 1 root root 100 12 de 2024 10:54 .X11-unix-0
-rw-r--r--. 1 root root 100 12 de 2024 10:54 .X11-unix-0
```

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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ ls -a >> lista_de_arquivos.txt
2024.1.08.043@suporte-OptiPlex-3050:~$ ls -a
.          .bash_logout  Desktop      home          lista_2024-05-06_10-54-06.txt  meu_script.sh  .netbeans  .profile  .ssh  testecrases.sh  testecrases.sh
..         .bashrc       Documents   lncc          lista_2024-05-06_10-54-08.txt  .neu_script.sh.swp  NetBeansProjects  Public  templates  thunderbird
.          .bashrc       Downloads   java          lista_de_arquivos.txt         .nohup        Pictures    recentas  testecrases.sh  Videos
.bash_history  .config      gnuPG       lista_2024-05-06_10-54-03.txt  local1          Music        .pkc       snap      testevariaveisambiente.sh  .vininfo
2024.1.08.043@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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ cat home | grep "palavra"
cat: home: Is a directory
2024.1.08.043@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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ cat << EOF
Este é um texto de exemplo
que pode ocupar múltiplas linhas.
> 1
> 2
> 3
> EOF
Este é um texto de exemplo
que pode ocupar múltiplas linhas.
1
2
3
2024.1.08.043@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.

```
2024.1.08.043@suporte-OptiPlex-3050:~$ echo scriptaritmetico.sh
scriptaritmetico.sh
2024.1.08.043@suporte-OptiPlex-3050:~$ num1=10
num2=3
resultado=$(echo "scale=2; $num1 / $num2" | bc)
echo "O resultado da divisão de $num1 por $num2 é: $resultado"
O resultado da divisão de 10 por 3 é: 3.33
```