



Introdução à Ciência da Computação – Lista 5

Shell script – parte 2

Nome: Pedro Ferreira Prado RA: 2025.1.08.028

- 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.

```
1 #!/bin/bash
2 cd /
3 ls -a
4
5
```

```
2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x meuscript.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./meuscript.sh
bash: ./meuscript.sh: No such file or directory
2025.1.08.028@suporte-OptiPlex-3050:~$ ./meuscript.sh
bash: ./meuscript.sh: No such file or directory
2025.1.08.028@suporte-OptiPlex-3050:~$ ./meuscript.sh
.      boot  etc   lib32  lost+found  opt   run   srv      tmp
..     cdrom  home  lib64  media       proc  sbin  swapfile  usr
bin    dev    lib   libx32  mnt         root  snap  sys      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.

```
#!/bin/bash
cd /
ls -l
echo "Meu primeiro script"
echo -n "Usuário logado no momento é"
whoami
```

```
2025.1.08.028
2025.1.08.028@suporte-OptiPlex-3050:~$ nano meuscript.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./meuscript.sh
total 2097244
lrwxrwxrwx    1 root root          7 jul 13  2023 bin -> usr/bin
drwxr-xr-x    4 root root      4096 mai 14  06:50 boot
drwxrwxr-x    2 root root      4096 jul 13  2023 cdrom
drwxr-xr-x   21 root root      4800 mai 20  10:03 dev
drwxr-xr-x  160 root root     12288 mai 14  06:50 etc
drwxr-xr-x   71 root root      4096 mai 15  13:02 home
lrwxrwxrwx    1 root root          7 jul 13  2023 lib -> usr/lib
drwxr-xr-x    2 root root      4096 mar 18  17:46 lib32
lrwxrwxrwx    1 root root          9 jul 13  2023 lib64 -> usr/lib64
lrwxrwxrwx    1 root root         10 jul 13  2023 libx32 -> usr/libx32
drwx-----   2 root root     16384 jul 13  2023 lost+found
drwxr-xr-x    6 root root      4096 nov 27  07:53 media
drwxr-xr-x    2 root root      4096 fev 23  2023 mnt
drwxr-xr-x   12 root root      4096 mar 18  16:59 opt
dr-xr-xr-x  362 root root         0 mai 20  09:57 proc
drwx-----  13 root root      4096 jul 26  2023 root
drwxr-xr-x   45 root root      1300 mai 20  10:15 run
lrwxrwxrwx    1 root root          8 jul 13  2023/sbin -> usr/sbin
drwxr-xr-x   17 root root      4096 set 25  2024 snap
drwxr-xr-x    2 root root      4096 fev 23  2023 srv
-rw-----   1 root root 2147483648 jul 13  2023 swapfile
dr-xr-xr-x   13 root root         0 mai 20  09:57 sys
drwxrwxrwt   21 root root     12288 mai 20  10:39 tmp
drwxr-xr-x   15 root root      4096 mar 18  12:25 usr
drwxr-xr-x   15 root root      4096 jul 25  2023 var
Meu primeiro script
Usuário logado no momento é2025.1.08.028
```

- 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.

```
#!/bin/bash
numero=42
texto="Aprendendo Shell Script"
echo "O numero é $numero e a mensagem é: $texto"
```

```
2025.1.08.028@suporte-OptiPlex-3050:~$ nano testevariaveis.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x testevariaveis.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testevariaveis.sh
O numero é 42 e a mensagem é: Aprendendo Shell Script
```

- 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.

```
#!/bin/bash
echo "o Shell padrão deste usuário é: $SHELL"
```

```
2025.1.08.028@suporte-OptiPlex-3050:~$ nano testevariaveisambiente.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x testevariaveisambiente.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testevariaveisambiente
bash: ./testevariaveisambiente: No such file or directory
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testevariaveisambiente.sh
o Shell padrão deste usuário é: /bin/bash
```

- 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.

```
GNU nano 6.2                                     testecrases.sh
#!/bin/bash
data=$(date +"%Y-%m-%d_%H-%M-%S")
echo "Data atual: $data"
ls -la > "listagem_$data.txt"
```

```

2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x testecrases.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data atual: 2025-05-20_10-54-48
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data atual: 2025-05-20_10-54-49
2025.1.08.028@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data atual: 2025-05-20_10-54-50

```

- 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.

```

#!/bin/bash
data=$(date +"%Y-%m-%d_%H-%M-%S")
echo "Data atual: $data"
ls -la > "listagem_$data.txt"
echo "Registro em: $(date)" >> registro.txt

2025.1.08.028@suporte-OptiPlex-3050:~$ ./testecrases.sh
Data atual: 2025-05-20_10-58-17
2025.1.08.028@suporte-OptiPlex-3050:~$ cat registro.txt
Registro em: ter 20 mai 2025 10:58:17 -03
2025.1.08.028@suporte-OptiPlex-3050:~$ cat registro.txt
Registro em: ter 20 mai 2025 10:58:17 -03
2025.1.08.028@suporte-OptiPlex-3050:~$ cat registro.txt
Registro em: ter 20 mai 2025 10:58:17 -03

```

- 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.

```

GNU nano 6.2                                redirecionamento.sh
#!/bin/bash
cat < entrada.txt

```

```

2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x redirecionameto.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./redirecionameto.sh
nanoscript.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ nano redirecionameto.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ cat entrada.txt
nanoscript.sh

```

- 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.

```

#!/bin/bash
cat << FIM
Isso é um exemplo
Pedro Ferreira
FIM

```

```

2025.1.08.028@suporte-OptiPlex-3050:~$ nano inline.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ chmod +x inline.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./inline.sh
Isso é um exemplo
Pedro Ferreira

```

- 9) Crie um script chamado scriptaritimetrico, 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.

```

GNU nano 6.2 scripta.sh
#!/bin/bash
a=10
b=3
c=4
d=2

resultado=$(echo "($a + $c) / $b" | bc)
echo "Resultado da operação é: $resultado"

```

```

2025.1.08.028@suporte-OptiPlex-3050:~$ nano scripta.sh
2025.1.08.028@suporte-OptiPlex-3050:~$ ./scripta.sh
Resultado da operação é: 4

```