

Lista de Tarefas: mini-curso *bash* SDC-CI 2018

Obtendo o diretório padrão

- Baixar o arquivo `.zip` do curso do [Github](#)

Exercícios

Parte 1

- Verificar versão do `bash` :

```
1 | $ echo $BASH_VERSION
```

Bash

- Acessar o diretório padrão:

```
1 | $ cd ~/sdc-mc
```

Bash

- Listar o conteúdo de todo o diretório do curso de modo recursivo em lista e com cores:

```
1 | $ ls -lRG
```

Bash

- Criar novo diretório `data` com dois subdiretórios `pb` e `rn`

```
1 | $ mkdir -p data/pb data/rn
```

Bash

- Copiar arquivo `csv/pop-x-2017.csv` para o diretório `data/x` , `x=pb,rn`

```
1 | $ cp csv/pop-pb-2017.csv data/pb;  
2 | $ cp csv/pop-rn-2017.csv data/rn;
```

Bash

- Renomear arquivo `data/pb/pop-pb-2017.csv` para `pop-pb-2017-2sem.csv`

```
1 | $ mv data/pb/pop-pb-2017.csv data/pb/pop-pb-2017-2sem.csv
```

Bash

- Acessar diretório `data/pb` e verificar tipo de arquivo

```
1 | $ cd data/pb
2 | $ file pop-pb-2017-2sem.csv
3 | # arquivo deve ser Non-ISO extended-ASCII text, with CR line terminators
```

Bash

Obs.: o arquivo original possui terminação com *carriage return* (CR), do DOS. Caso se queira convertê-lo em tipo UNIX, devemos usar algum artifício. Um caminho mais simples é instalar `dos2unix` por `apt-get` ou por `brew` no macOS.

- Abrir o arquivo `.csv` com o programa padrão apenas para demonstração (interface gráfica) e depois fechar janela (na interface).

```
1 | $ open pop-pb-2017-2sem.csv
2 | # ou use open -a <nome_programa>
```

Bash

- Criar uma variável de ambiente temporária para o nosso diretório padrão

```
1 | $ cd ~/sdc-mc;
2 | # pwd imprime o diretório atual
3 | $ pwd
4 | # este comando cria a variável 'sdc' com o valor
5 | # do diretório atual
6 | $ export sdc=$PWD;
7 | # o comando 'echo' imprime na tela
8 | $ echo $sdc
9 | $ /Users/gustavo/sdc-mc
```

Bash

Obs.: A partir da definição da variável temporária `sdc` como o caminho para nosso diretório padrão, podemos usá-la na sessão de login atual para ir diretamente para nosso diretório de qualquer lugar do computador.

- Visitando o `root` e voltando para o nosso diretório:

```
1 | $ cd /; cd $sdc
```

Bash

- Visualizando o nosso histórico de comandos

Bash

```
1 | $ history
2 | # repetir último comando digitado
3 | $ !!
4 | # repetir 2o. evento para trás
5 | $ !-2
```

- Criar um diretório novo `img` um nível abaixo de `sdc-mc`

Bash

```
1 | $ mkdir -p $sdc/img
```

- Fazer download da imagem do pinguim do Linux da Wikipedia e salvar em `img` como um `.png` :

Bash

```
1 | $ cd $sdc/img
2 | # no macOS
3 | $ curl https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/2000px-Tux.svg
4 | # no Linux (verificar!)
5 | $ wget https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/2000px-Tux.svg
```

- Criar uma cópia da imagem em `.tiff` :

Bash

```
1 | $ sips -s format png pinguim.png --out pinguim2.tiff
```

- Pedindo ajuda sobre o comando `sips` no manual

Bash

```
1 | $ man sips
```

- Deletar a imagem `.tiff` com pergunta de segurança:

Bash

```
1 | $ rm -i pinguim2.tiff
2 | # y: yes n: no
3 | $ remove pinguim2.tiff? y
4 | # para remover na força bruta, use rm -f (COM CUIDADO!)
```

Parte 2

Parte 3

- Executar editor de texto, inserir o nome do aluno, salvar, fechar o editor, retornar ao `bash` e imprimir o conteúdo na tela.

Como fazer:

Bash

```
1 | # Depois de executar 'nano', realizar comandos com o teclado e sair
2 | # no macOS: 'Ctrl+O' (escrever); 'Ctrl+M' (formato); Ctrl+X (sair)
3 | $ cd ~/sdc-mc
4 | $ nano eusou.txt
5 | $ cat eusou.txt
6 | $ Gustavo
```