

## Lista de exercícios 3 - Shell

Guilherme Oliveira Mussa Tavares

### 1 - Com suas palavras, explique o que é um shell

**R:** Shell é uma interface de linha de comando (CLI) que permite ao usuário interagir com o sistema operacional através da escrita de comandos os quais executam funções específicas, como, por exemplo, a execução de um determinado programa (processo) ou a manipulação de um arquivo.

### 2 - Quero executar esta sequência de comandos: `cat xx.txt`, `ls`, `sort` e guardar num arquivo `.txt`. Cite ao menos uma forma de fazer isso, de maneira que o arquivo final tenha o conteúdo do `ls` e do `cat`, não importando a ordem dos comandos.

**R:**

Uma maneira de executar a sequência de comandos requisitada é:

```
(ls; cat numeros.txt) | sort > saida.txt
```

### 3 - Qual a utilidade de usar a técnica de `shebang`?

**R:** Ao se utilizar a técnica de `shebang`, é possível definir um interpretador padrão para execução de um determinado shell script. Por exemplo, ao inserir o `shebang` `#!/bin/bash` na primeira linha de um shell script, é definido que o Bash deve ser utilizado como interpretador, não necessitando de invocar o Bash explicitamente quando quiser executar o script.

#### 4 - Abaixo, serão listadas várias ações. Execute uma por vez no terminal e escreva o comando que usou logo em seguida.

- Estrutura de diretórios sugerida

```
./ccuenf  
./exercicios
```

**R:**

<b>ação</b>	<b>comando</b>
Verifique em qual diretório você está	<code>pwd</code>
Crie a estrutura de diretórios sugerida logo acima na home do usuário	<code>mkdir ccuenf exercicios</code>
Crie um arquivo "numeros.txt" com o cat, não importa a quantidade de números	<code>cat &gt; numeros.txt</code>
Duplique "numeros.txt" para "numeros1.txt" e "numeros2.txt"	<code>cp numeros.txt numeros1.txt &amp;&amp; cp numeros.txt numeros2.txt</code>
Copie todos os arquivos com extensão .txt para a pasta "exercicios"	<code>cp *.txt exercicios</code>
Exiba os arquivos da pasta "exercicios" com seus detalhes (permissões, tamanho e etc) sem sair da pasta atual	<code>ls -l exercicios</code>
Apague a pasta "ccuenf" e "exercicios"	<code>rm -rf ccuenf exercicios</code>

#### 5 - Liste todos os arquivos da pasta /bin, salve em arquivo e ordene alfabeticamente. Quais foram os comandos utilizados? Não é obrigatório executar a sequência em apenas uma linha.

**R:**

O problema pode ser resolvido utilizando a seguinte linha de comando:

```
ls -a /bin > saida.txt && sort saida.txt
```

Neste caso, os comandos utilizados foram `ls` e `sort`, além do redirecionamento `>` e o operador lógico `&&`.

**6 - Crie um arquivo .txt com o conteúdo da execução de `man <comando>` - onde "comando" é qualquer comando linux conhecido e que possua um manual - e inverta a ordem das linhas, procure pela palavra "the" e conte o número de ocorrências. Coloque aqui seu resultado e qual comando usou para pegar o manual.**

**R:**

O comando utilizado para pegar o manual foi o `man`. Já a sequência de comandos solicitada ficou da seguinte forma:

```
man mkdir | tac > saida.txt && grep -c -w 'the' saida.txt
```

Primeiramente, obtemos um manual, no caso do comando acima, o manual do comando `mkdir`. Em seguida, é feito um *pipe* com o comando `tac`, para inverter a ordem das linhas do manual, e depois, a saída é gravada num arquivo `saida.txt`. Ao final, combinamos o comando anterior ao utilizar o comando `grep` com o parâmetro `-c` para obtermos o número de ocorrências da palavra "the" no arquivo de saída recém-criado.

**7 - Escolha um ou mais programas da lista e tente instalá-lo, execute-o e descreva o resultado.**

programas
sl
cmatrix
rig
nyancat

- Após terminar o exercício, remova o programa

**R:**

Ao instalar e executar o programa `sl`, aparece no terminal uma animação feita com caracteres ASCII de um trem "atravessando" a tela.

## 8 - Qual o resultado das seguintes expressões?

- Já que as colunas da tabela são formadas por |, não posso escrever o pipe dentro da tabela. Tem como escapar com o `\vert`, porém ele só é renderizado no `org-mode` do emacs :/
- `\vert == |`

**R:**

expressão	resultado
<code>false &amp;&amp; echo foo    echo bar</code>	bar
<code>true &amp;&amp; echo foo    echo bar</code>	foo
<code>true    echo foo &amp;&amp; echo bar</code>	bar
<code>false    echo foo &amp;&amp; echo bar</code>	foo bar