

Capítulo 3

Esses exercícios podem ser feitos em, no máximo, duas linhas. A finalidade deles é simplesmente de exercitar a sua lógica dentro da sintaxe do *Shell*.

===Exercício 1===

Fazer um programa para procurar, pelo sobrenome, pessoas no arquivo `telefones`.

```
$ cat exerc2-1.sh
cut -f2 -d' ' telefones | cut -f1
```

A saída do 1º `cut`, é a 2ª parte do arquivo com relação ao espaço, ou seja, `nome2<TAB>(DDD)telefone`, veja:

```
$ cut -f2 -d' ' telefones
Avellar      (514) 692-4322
Grippi       (021) 555-1234
Marcia       (021) 555-2112
Cardoso      (023) 232-3423
Duarte       (024) 622-2876
Carlos       (021) 767-2124
Biasoli      (051) 101-0101
Garrafas     (021) 988-3398
Gerhardt     (024) 543-4321
Duarte       (011) 449-0219
```

Esta saída que acabamos de ver, foi redirecionada via *pipe* (`|`) para a entrada do outro `cut`, que pegou o primeiro campo com relação ao `<TAB>`, que não foi especificado por ser o delimitador *default*

===Exercício 2===

Fazer um programa para listar todas as pessoas de um determinado DDD.

```
$ cat exerc2-2.sh
grep "($1)" telefones
```

Usamos como delimitadores os parênteses que envolvem o DDD para que o parâmetro informado não pudesse casar com um número de telefone.

===Exercício 3===

Como posso pegar o último parâmetro passado em uma lista de tamanho indeterminado?

Se temos `N` parâmetros, basta matar (`shift`) os `N-1` primeiros parâmetros e listar o único que sobrou. Não esqueça que `$#` tem a quantidade de parâmetros recebidos.

```
$ cat exerc2-3.sh
Mata=${($#-1)}
```

```
shift $Mata
echo $1
```

===Exercício 4===

Listar os usuários que estão “logados” há mais de um dia.

Para fazer este último exercício, veja a sintaxe do comando **date** e repare que você consegue colocar a data de hoje no mesmo formato da data de *login* do comando **who**, que é onde você encontra todos os usuários “logados”. A versão do **who** que estou usando, exibe a data como:

```
date +%Y-%m-%d
```

Sabendo disso, vamos ver o *script*:

```
$ cat exerc2-4.sh
Hoje=$(date +%Y-%m-%d)
who | grep -v $Hoje
```

O **grep -v** serve para deletar da saída do comando **who**, todos os registros que tiverem a data de hoje, sobrando somente os que se logaram há mais de 1 dia.

===Exercício 5===

Em se falando de passagens de parâmetros, as variáveis **\$0**, **\$#**, **\$1...\$n**, **\$*** e **\$@**, possuem respectivamente:

- A) Nome do programa, a quantidade de parâmetros, os parâmetros posicionais, todos os parâmetros e todos os parâmetros;
- B) Todos os parâmetros, a quantidade de parâmetros, nome do programa, os parâmetros posicionais e o último parâmetro;
- C) Nome do programa, todos os parâmetros, a quantidade de parâmetros, os parâmetros posicionais e o último parâmetro;
- D) A quantidade de parâmetros, todos os parâmetros, o nome do programa, os parâmetros posicionais e o último parâmetro;
- E) Nenhuma dessas aí

A resposta certa é a A

===Exercício 6===

Coloque nos parênteses ao lado das opções do **declare** a letra correspondente às descrições que estão à direita

Opção	Descrição do declare
-A (G)	A - Define que uma variável será só de maiúsculas
-a (H)	B - Define que uma variável será só de minúsculas

Opção	Descrição do declare
-c (E)	C - A variável será exportada para todos os subshells
-i (F)	D - Terá valor constante (seu valor não pode ser alterado)
-l (B)	E - A variável será capitalizada (só primeira letra em maiúscula)
-r (D)	F - Define que uma variável será só de números inteiros
-u (A)	G - A variável é um vetor associativo
-x (C)	H - A variável é um vetor indexado