



## Guião de aula Programação em Shell I

---

### 1. Introdução

Utilizando o interpretador de comandos *bash* ou *shell*, podemos criar scripts no sistema Linux. Esses scripts facilitam muito a vida do programador ao economizar tempo, pois permitem automatizar tarefas de rotina e também tornam mais simples ações como abrir programas, escalonar tarefas e/ou esvaziar diretórios temporários e lixeira.

O primeiro passo na criação de um script, é fazer um ficheiro de texto com o script. Para dizer à shell, que o ficheiro de texto é um script, devemos escrever nos dois primeiros caracteres da 1ª linha o código `#!/`, que é o magic code referente a script. A seguir a este código deve-se indicar o interpretador para o script, no nosso caso `/bin/bash`, seguido dos parâmetros de configuração da shell (ex. `-noprofile`, para não carregar os ficheiros de profile). Os scripts perl e tcl/tk, etc. também utilizam este código, sendo somente alterado o interpretador.

A partir deste momento (se a flag *x* estiver ativa), podemos executar o script diretamente na linha de comando (`./nome_do_script`). A definição do interpretador do script é importante, pois se correremos dentro de uma shell *csh* um script feito para *bash* sem a indicação da shell a utilizar, irão ocorrer erros devidos à diferente sintaxe destas shells. O carácter `#` é também um carácter especial, que indica à shell que tudo o que está até ao fim da linha é comentário.

Antes de iniciar a programação, verifique se entrou no programa como utilizador comum ou root. Se estiver no modo root, mude para utilizador comum. Porque necessita mudar? Como root qualquer ação descuidada do programador será aceita pelo sistema operativo e poderá causar danos na sua execução.

### 2. Exercícios propostos

Todos os exercícios propostos são para ser realizados através da linha de comando existente no Linux.

#### 2.1. Executar o seguinte *shell-script*

```
#!/bin/bash
if [ $1 -lt 0 ]
then echo "$1 e' um numero inteiro negativo"
elif [ $1 -eq 0 ]
then echo "$1 e' o numero zero"
else echo "$1 e' um numero inteiro positivo"
fi
```

- 2.2. Alterar este *shell script* de modo que seja escrita uma mensagem de erro “Erro: falta um argumento” se ele for invocado sem argumentos.
- 2.3. Alterar este *shell script* de modo que possa ser invocado com 1 ou mais números inteiros na linha de comandos.
- 2.4. Efetuar um comando que receba 1 ou mais números inteiros através da linha de comando e que apresente o número de positivos, negativos e zeros que foram lidos.
- 2.5. Efetuar um comando que calcula a soma de todos os números inteiros lidos através da linha de comandos.
- 2.6. Efetuar um comando que calcula a soma de um conjunto de números inteiros lidos de *stdin* (um por linha).
- 2.7. Efetuar um comando que calcula quantos ficheiros e quantas diretorias existem na diretoria corrente.

### 3. Exercícios extras

- 3.1. Alterar o *shell script* da pergunta 2.2 de modo que seja escrita uma mensagem de erro se ele for invocado com argumentos que não sejam números inteiros.
- 3.2. Efetuar um comando que calcula a soma de um conjunto de números inteiros lidos de um ficheiro (um por linha) cujo nome é definido através da linha de comandos.
- 3.3. Efetuar um comando que calcula a soma de um conjunto de números inteiros lidos de um ficheiro cujo nome é definido através da linha de comandos. Cada linha do ficheiro tem o seguinte formato: <palavra> <inteiro>.
- 3.4. Efetuar um comando que receba um conjunto de diretorias na linha de comandos e que indique quais destas diretorias não possuem a subdiretoria TRAB1.