

Projeto de Laboratório de Computadores 2019/20 - my_grep

O projeto a desenvolver em Laboratório de Computadores consiste no desenvolvimento de vários utilitários que poderão ser utilizados para o processamento, na linha de comandos, de dados guardados em ficheiros.

Nesta ficha de trabalho deverá implementar o comando `my_grep`.

Nota: este enunciado poderá ainda sofrer alterações.

Comando my_grep

1. Comando `my_grep`

Sinopse:

```
my_grep [OPÇÕES] word [FICHEIROS...]
```

Descrição:

Implemente, em C, o comando `my_grep` que procure, na lista de ficheiros, `[FICHEIROS]` por linhas que contenham a palavra `word`. Por omissão, essas linhas deverão ser impressas na saída padrão. Os ficheiros deverão ser processados pela ordem que aparecem na linha de comandos. Se o(s) ficheiro(s) for(em) apenas um `-` (ou não for indicado nenhum nome), o comando `my_grep` deverá ler da entrada padrão.

Valor de retorno:

O programa deverá retornar (função `exit()`):

- o valor **0**, caso termine normalmente e sejam encontradas linhas com a palavra a pesquisar;
- o valor **1**, caso termine normalmente e não seja encontrada nenhuma linha com a palavra a pesquisar;
- o valor **2**, caso ocorra algum erro.

Situações de erro:

Caso um ficheiro não consiga ser aberto, deverá ser impressa, a mensagem de erro (no `stderr`):

```
./my_grep: <file>: No such file or directory
```

Exemplos:

Considere o ficheiro `BaladaDaNeve.txt` com o seguinte conteúdo:

```
Batem leve, levemente,  
como quem chama por mim.  
Será chuva? Será gente?  
Gente não é, certamente  
e a chuva não bate assim.
```

e o ficheiro PedraFilosofal.txt com o seguinte conteúdo:

```
Eles não sabem que o sonho  
é uma constante da vida  
tão concreta e definida  
como outra coisa qualquer,
```

Exemplo 1: o comando

```
$ ./my_grep leve BaladaDaNeve.txt
```

escreverá na saída padrão

```
Batem leve, levemente,
```

Exemplo 2: o comando

```
$ ./my_grep "quem chama" BaladaDaNeve.txt
```

escreverá na saída padrão

```
como quem chama por mim.
```

Exemplo 3: o comando

```
$ ./my_grep chuva BaladaDaNeve.txt
```

escreverá na saída padrão

```
Será chuva? Será gente?  
e a chuva não bate assim.
```

Exemplo 4: o comando

```
$ ./my_grep casa BaladaDaNeve.txt
```

não escreverá nada da saída padrão.

Exemplo 5: o comando

```
$ ./my_grep não BaladaDaNeve.txt PedraFilosofal.txt
```

escreverá na saída padrão

```
BaladaDaNeve.txt:Gente não é, certamente  
BaladaDaNeve.txt:e a chuva não bate assim.  
PedraFilosofal.txt:Eles não sabem que o sonho
```

Exemplo 6: o comando

```
$ ./mygrep leve  
A bola do Pedro  
bate levemente no chão  
do campo.
```

escreverá na saída padrão

```
bate levemente no chão
```

Exemplo 7: o comando

```
$ ./mygrep leve - BaladaDaNeve.txt  
A bola do Pedro  
bate levemente no chão  
do campo.
```

escreverá na saída padrão

```
(standard input):bate levemente no chão  
BaladaDaNeve.txt:Batem leve, levemente,
```

2. Altere o programa `my_grep` para suportar as seguintes opções:

`-i` ignorar distinções de capitalização entre a palavra a procurar e o conteúdo dos ficheiros. Por exemplo, se a palavra a procurar for *portugal* e uma linha nos ficheiros contiver a palavra *PorTuGaL*, essa linha deverá ser considerada como uma pesquisa bem sucedida. `-v` inverter o sentido de correspondência na pesquisa, procurando-se por linhas que não contenham a palavra *word*. Por exemplo, se a palavra for *portugal* deverão ser consideradas como pesquisa bem sucedida as linhas que não contenham a palavra *portugal*. `-l` deverão **apenas ser impressos**, os nomes dos ficheiros

em que a procura teve sucesso. `-c` deverá **apenas ser impresso**, para cada ficheiro, o número de linhas em que a procura teve sucesso.

Sugestões

- Leia o manual do comando `grep` num terminal (`$ man grep`).
- Leia o manual das funções que considerar utilizar.
- Faça experiências com o comando `grep` num terminal. O seu programa deverá reproduzir o comportamento do comando `grep`.
- Para o processamento de opções, poderá recorrer à função `getopt()` das bibliotecas *unistd.h* e *getopt.h*. Poderá encontrar exemplos de utilização desta função nas ligações: [link1](#), [link2](#), e [link3](#).
- Outras funções deverá ponderar utilizar: `getc()` ou `fgets()`, `putchar()` ou `fputs()`, `tolower()`, `fprintf()`, `exit()`.