

Projeto de Laboratório de Computadores 2019/20 - my_uniq

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_uniq`.

Nota: este enunciado poderá ainda sofrer alterações.

Comando my_uniq

Sinopse:

```
my_uniq [OPÇÕES]... [INPUT]
```

Descrição:

Implemente, em C o comando `my_uniq` que leia o conteúdo de ficheiro `INPUT` de forma sequencial, escrevendo na saída padrão. Apenas uma cópia de **linhas adjacentes duplicadas** deverá ser escrita, eliminando-se assim repetições (em linhas adjacentes!!). Se o ficheiro `INPUT` for apenas um "-" (ou não for indicado nenhum nome), o comando `my_uniq` deverá ler da entrada padrão.

Em particular, deverá implementar o comando `my_uniq` ****com processamento de opções**** da linha de comandos, nomeadamente:

- `-c` - preceder cada linha com o número de repetições, seguido de um espaço.
- `-d` - imprimir apenas linhas duplicadas.
- `-i` - comparar linhas ignorando a capitalização.

Valor de retorno:

O programa deverá retornar (função `exit()`) o valor **0**, caso termine normalmente, ou um valor **>0**, caso ocorra algum erro.

Situações de erro:

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

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

Exemplos de utilização

Considere o ficheiro *file1* com o seguinte conteúdo:

```
Eles não sabem que o sonho
Eles não sabem que o sonho
Eles não sabem que o sonho
```

```
é uma constante da vida
```

```
tão concreta e definida
tão concreta e definida
```

```
como outra coisa qualquer,
```

Exemplo 1:

o comando

```
$/my_uniq file1
```

escreverá na saída padrão:

```
Eles não sabem que o sonho
```

```
é uma constante da vida
```

```
tão concreta e definida
```

```
como outra coisa qualquer,
```

Exemplo 2:

Com a opção `-c`, o comando `my_uniq` deverá indicar o número de vezes que uma linha foi repetida.

Considerando o ficheiro *file1* apresentado acima, o comando

```
$/my_uniq -c file
```

escreverá na saída padrão:

```
3 Eles não sabem que o sonho
```

```
1
```

```
1 é uma constante da vida
```

```
1
```

```
2 tão concreta e definida
```

```
1
```

```
1 como outra coisa qualquer,
```

Exemplo 3:

Com a opção `-d`, o comando `my_uniq` deverá imprimir apenas as linhas repetidas.

Considerando o ficheiro *file1* apresentado acima, o comando

```
$/my_uniq -d file
```

escreverá na saída padrão:

```
Eles não sabem que o sonho  
tão concreta e definida
```

Considere o ficheiro *file2* com o seguinte conteúdo:

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

Exemplo 4:

O comando

```
$/my_uniq file
```

escreverá na saída padrão:

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

visto o comando `my_uniq` ser *case sensitive*.

No entanto, com a opção `-i`, o comando `my_uniq` deverá ser *case insensitive*, ignorando, portanto, a capitalização.

Logo, considerando o ficheiro *file2* apresentado acima, o comando

```
$/my_uniq -i file2
```

escreverá na saída padrão:

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

Sugestões:

- Leia o manual do comando `uniq` num terminal (`$ man uniq`).
 - Faça experiências com o comando `uniq` num terminal. O seu programa deverá reproduzir o comportamento do comando `uniq`.
 - Leia o manual das funções que considerar utilizar.
 - 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](#).
-