

Prova Prática #2 – Enunciado F

2019.06.06/21h00

Prova com Consulta

Duração: 75 minutos

Nome Completo: _____

N.º de estudante: _____ Regime: ☐ Diurno ☐ Pós-laboral

IMPORTANTE

É expressamente proibido o recurso à Internet durante a prova. Qualquer utilização não autorizada da Internet leva à anulação da prova e ao reportar da situação às autoridades competentes. O mesmo sucede com outros tipos de tentativa de fraude.

• Antes de iniciar a prova:

- Execute os seguintes comandos:

```
cd; mkdir -p ~/SOProva2/R_NUMERO/
```

(em que **R** deve ser substituído pela letra **D** se for do regime diurno e **N** se for aluno do regime pós-laboral e **NUMERO** deve ser substituído pelo seu número ESTG);

- Para garantir que o seu diretório de trabalho seja o correto, faça:

```
cd ~/SOProva2/R_NUMERO/
```

• Após ter terminado a prova:

- Deverá proceder à criação de um arquivo TAR, fazendo uso do seguinte comando:

```
cd ~/SOProva2/R_NUMERO/; tar cvf SOProva2_YYYYMMDD_R_NUMERO.tar *
```

(em que YYYYMMDD corresponde à data corrente (e.g., 20190606) e R_NUMERO obedece ao formato acima indicado);

- Verifique que o arquivo “.tar” que criou não está vazio, através da execução de:

```
tar tvf SOProva2_YYYYMMDD_R_NUMERO.tar
```

- Entregue o arquivo “.tar” através da plataforma *moodle*, no espaço reservado para o efeito. Em caso de dúvidas, pergunte ao professor;
- Informe o professor para este validar a receção dos seus ficheiros.

Pergunta 1 [20 valores]

(Escreva as suas respostas a esta pergunta no diretório "`~/SOProva2/R_NUMERO/Pergunta1`". Deve indicar o seu nome completo e número IPLeia numa das 5 primeiras linhas de todos os ficheiros de texto que entregar).

NOTA 1: não é permitida a chamada a comandos externos através da função *system* ou de outra com funcionalidade similar.

NOTA 2: a solução deve ser implementada com recurso aos ficheiros do *template makefile* empregue na UC.

NOTA 3: código entregue que **não compile** através do utilitário *make* e do respetivo *makefile* leva à atribuição da classificação de **0 (zero) valores** à resposta.

Recorrendo à linguagem C, elabore a aplicação **invert_file** cujo propósito é o de guardar, de forma inversa, o conteúdo binário de um ficheiro. O nome do ficheiro a inverter é indicado pela linha de comando.

A aplicação deve receber o seguinte parâmetro da linha de comandos:

-f/--file <ficheiro>: nome do ficheiro a inverter. Parâmetro obrigatório.

Caso não seja possível abrir o ficheiro, indicado no parâmetro **-f/--file**, deve ser escrita uma mensagem de erro adequada no canal de erros, bem como terminado o programa com o código de erro 1.

Relativamente ao ficheiro invertido, o seu nome deve conter a *string* “_inverted” depois do nome do ficheiro a processar e antes da sua extensão. Por exemplo, para o ficheiro com o nome “hello.jpg” o nome do ficheiro invertido será “hello_inverted.jpg”.

Tenha ainda em atenção que, o 1º byte do ficheiro a processar deve ser o último byte do ficheiro invertido, 2º byte do ficheiro a processar deve ser o penúltimo byte do ficheiro invertido, e assim sucessivamente. O último byte do ficheiro a processar é o 1º byte do ficheiro invertido.

Para ambos os ficheiros deve ser mostrado, na saída padrão da aplicação, o seu tamanho em *bytes*.

Considere os seguintes exemplos de execução:

Exemplo 1

```
./invert_file -f example.txt
File to invert: example.txt (1892 bytes)
Inverted file: example_inverted.txt (1892 bytes)
```

Exemplo 2

```
./invert_file -f image.png
File to invert: image.png (7589 bytes)
Inverted file: image_inverted.png (7589 bytes)
```

Exemplo 3

```
./invert_file -f ficheiro_inexistente
ERROR: Can't open file 'ficheiro_inexistente': No such file or directory
```