

Prova Prática #2 – Enunciado E

2019.06.05/16h00

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., 20190605) 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 IPLeiria 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.

ELF é um formato padrão para representar ficheiros executáveis, código objeto, bibliotecas partilhadas e ficheiros *core*. Recorrendo à linguagem C elabore a aplicação **elf-parser** que recebendo um ou mais nomes de ficheiros através da linha de comando deve avaliar se cada um dos ficheiros é ou não um ficheiro do tipo ELF. Um ficheiro válido ELF contém como primeiros 4 octetos os valores (hexadecimal): **7F 45 4C 46**; o 5º octeto indica se o ficheiro representa um ELF de 32-bits (valor **1**) ou 64-bits (valor **2**); e o 17º octeto indica o tipo de ficheiro, de acordo com a tabela seguinte:

0x00	No file type
0x01	Object file
0x02	Executable
0x03	Shared object file
0x04	Core file

Para cada ficheiro válido, a aplicação deve mostrar a arquitetura (32/64 bits) e o tipo de ficheiro ELF. Caso o ficheiro seja inválido a aplicação deve mostrar a string "NOT ELF". Considere o seguinte exemplo:

```
./elf-parser main.o elf-parser makefile core
```

```
file.o: 32 bits; Object file
```

```
elf-parser: 32 bits; Shared object file
```

```
makefile: NOT ELF
```

```
core: 32 bits; Core file
```

A aplicação deve emitir uma apropriada mensagem no canal de erro padrão quando é detetado um nome de ficheiro que não existe, que não é um ficheiro ou para o qual não se tenha acesso. Considere os seguintes exemplos:

```
./elf-parser
```

```
Invalid number of arguments
```

```
Usage: ./elf-parser <file>...
```

```
./elf-parser not-found no-permissions
```

```
not-found: No such file or directory
```

```
no-permissions: Permission denied
```