

## Teste Prático – Enunciado B

2023.11.04 / 11h15'

Prova com consulta

Duração: 90 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 ~/ProvaP/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 ~/ProvaP/R_NUMERO/
```

- **Após ter terminado a prova:**

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

```
cd ~/ProvaP/R_NUMERO/; tar cvf ProvaP_YYYYMMDD_R_NUMERO.tar *
```

(em que YYYYMMDD corresponde à data corrente, e.g., 20211218, e R\_NUMERO obedece ao formato acima indicado);

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

```
tar tvf ProvaP_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 [20 valores]

(Escreva as suas respostas a esta pergunta no diretório "`~/ProvaP/R_NUMERO/Pergunta`". Deve indicar o seu nome completo e número de estudante IPLeiria no ficheiro **README.txt** a ser criado no diretório)

**NOTA 1:** não é permitida a chamada a comandos externos através da função `system` ou de outra com funcionalidade similar (e.g., `/bin/sh`).

**NOTA 2:** a solução deve ser implementada com recurso aos ficheiros/makefile do arquivo EmptyProject-Templatev3.05.zip

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

Recorrendo à linguagem C, implemente a aplicação `regional_elections`, cujo propósito é o de simular um processo eleitoral em várias regiões de um país recorrendo a processos e a *threads*. O processo pai deve criar um processo filho por cada região. Cada processo filho deve gerir o processo eleitoral em cada região. Cada *thread* representa um eleitor. Em cada região só pode votar um eleitor de cada vez (mas pode haver eleitores de regiões distintas a votar em simultâneo). Cada eleitor realiza uma operação muito simples: escolhe aleatoriamente um dos partidos candidatos e incrementa o número de votos desse partido na sua região. No final, o programa deve mostrar o número de votos que cada partido obteve em cada região.

O número de partidos candidatos é passado através dos parâmetros da linha de comandos. Para efeitos de teste, assuma os seguintes valores constantes:

- Número de regiões: 2

- Número máximo de eleitores em cada região: 5

O número de eleitores de cada região é definido aleatoriamente.

Considere os seguintes exemplos de execução.

<b>Exemplo 1</b> <code>./regional_elections -c 2</code>  Number of candidates: 2 Num voters in region 2: 2 Num voters in region 1: 4 Votes of candidate 1 in region 2: 2 Votes of candidate 2 in region 2: 0 Votes of candidate 1 in region 1: 1 Votes of candidate 2 in region 1: 3 The elections are now closed.	<b>Exemplo 2</b> <code>./regional_elections -c 2</code>  Number of candidates: 2 Num voters in region 1: 4 Num voters in region 2: 5 Votes of candidate 1 in region 1: 2 Votes of candidate 2 in region 1: 2 Votes of candidate 1 in region 2: 1 Votes of candidate 2 in region 2: 4 The elections are now closed.
<b>Exemplo 3</b> <code>./regional_elections -c 1</code>  Number of candidates: 1 Num voters in region 1: 2 Num voters in region 2: 1 Votes of candidate 1 in region 1: 2 Votes of candidate 1 in region 2: 1 The elections are now closed.	<b>Exemplo 4</b> <code>./regional_elections -c 0</code>  Parameter candidates should be larger than 1

**Nota:** Poderá usar a função `rand_r` para gerar números aleatórios diferentes como é exemplificado no código abaixo.

### Fora das threads:

```
unsigned int rand_state = time(NULL) ^ getpid();  
int value1 = rand_r(&rand_state); // gera um valor aleatório  
int value2 = rand_r(&rand_state); // gera outro valor aleatório
```

### Nas threads:

```
unsigned int rand_state = time(NULL) ^ getpid() ^ pthread_self();  
int value1 = rand_r(&rand_state); // gera um valor aleatório  
int value2 = rand_r(&rand_state); // gera outro valor aleatório
```