



90 minutos

Nome _____

nº _____

Regras

Qualquer tentativa de fraude conduzirá à anulação da prova para todos os intervenientes.

Consulta: Apenas em papel. Não são permitidos meios electrónicos (calculadora, telemóvel, ...).

As respostas devem ser obrigatoriamente efectuadas na folha fornecida.

1 Questão (85%)

Implemente um processo que leia números inteiros de dois *named pipes* e que apresente no ecrã a estatística dessas leituras a cada 5 segundos. A estatística é constituída pelo número de inteiros lidos e pela sua soma total. Deverá ainda ter em conta as seguintes funcionalidades e restrições:

- Os *named pipes* devem ser criados pelo processo principal no início da sua execução e terão o nome “PIPE1” e “PIPE2”.
- A leitura de dados dos *named pipes* é feita através de 1 *thread* específica (apenas uma!).
- Os dados recebidos pelos *named pipes* são inteiros.
- Após a leitura de um novo inteiro, a *thread* deverá somar esse valor a uma variável de nome *soma_total* que conterà a cada momento a soma total dos valores lidos de todos os *named pipes*.
- De 5 em 5 segundos uma *thread* responsável pelas estatísticas irá escrever no ecrã o número de leituras realizadas e o valor da variável *soma_total*.
- A *thread* que lê dados dos *named pipes* termina quando tiver lido o valor -1 de qualquer um dos *named pipes*. Depois disso, a *thread* que apresenta as estatísticas apresentará uma última vez as estatísticas e também sairá. Por fim, todos os recursos em utilização devem ser destruídos (incluindo o *named pipe*).
- Declare todas as variáveis que precisa no início do código e use todos os mecanismos de sincronização necessários a um bom funcionamento do programa.
- Não deve existir espera activa.

Complete o código que se segue, escrevendo nos locais a cinzento.

```
(... includes ...)
```

```
#define PIPE_NAME1 "PIPE1"  
#define PIPE_NAME2 "PIPE2"
```

```
// variáveis globais  
int fd_named_pipe[2];  
int soma_total=0;
```

```
// Outras funções
```

```
// Função a ser executada pelas thread que lê dos named pipes
```

```
// Função a ser executada pelas thread que lê e escreve a estatística no ecrã
```

Nome:_____

Número:_____

```
int main(void) {
```

```
// criar pipes
```

```
// criar threads
```

```
// Esperar pelo fim das threads e remover recursos
```

```
}
```

2 Questão (15%)

Dois dos mecanismos de sincronização de *threads* são os Mutexes e os Semáforos. Qual a diferença entre estes dois mecanismos e em que situações deve utilizar cada um?