



Sistemas Operativos II

1.ª Meta



João Almas | 2021138417 | P5

Paulo Sá | 2021142819 | P5

Índice

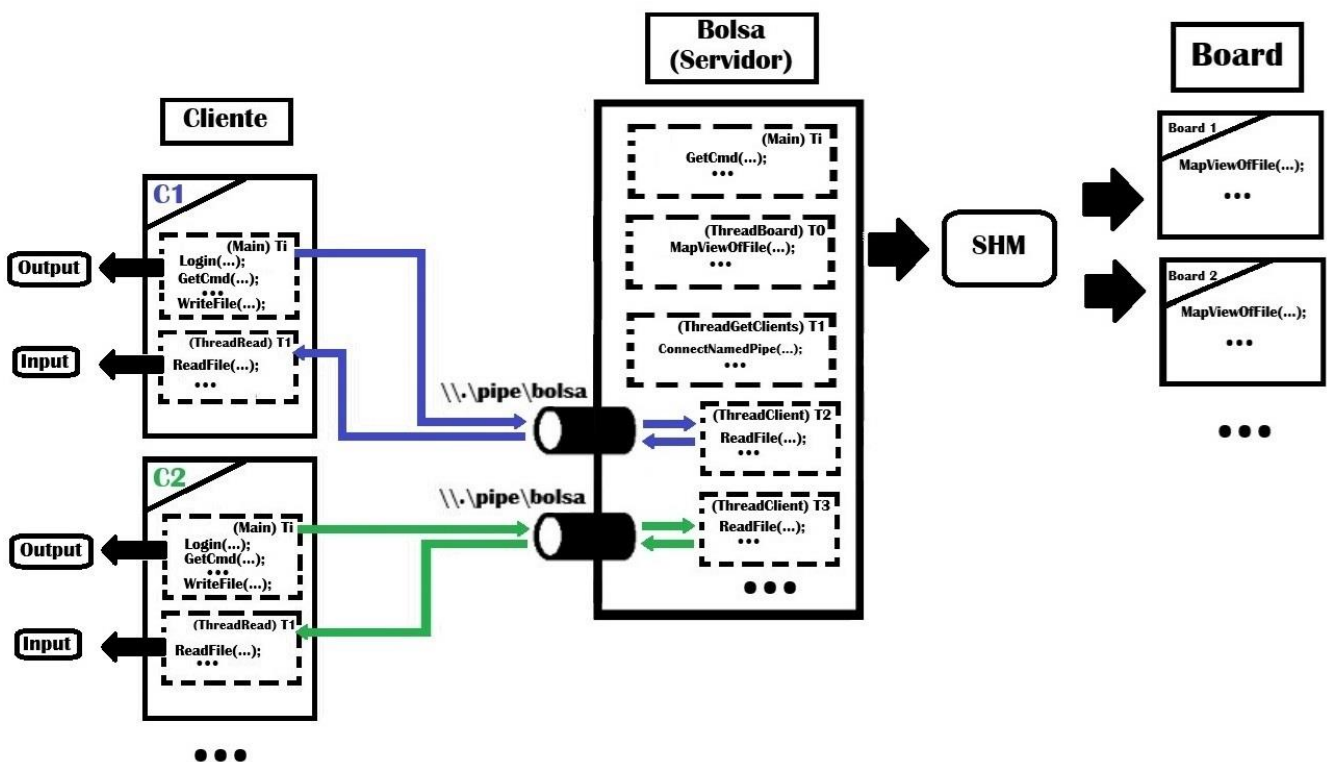
1.	Introdução	2
2.	Programas	2
2.1	Bolsa.....	3
2.2	Cliente	4
2.3	Board.....	5
3.	Comunicação entre Processos	5
3.1	Cliente ↔ Bolsa	5
3.2	Bolsa ↔ Board.....	6
4.	Estruturas de Dados Utilizadas	7

1. Introdução

No âmbito da unidade curricular de Sistemas Operativos II, foi proposto o desenvolvimento de uma aplicação que simula uma bolsa de valores online. Este projeto consiste na implementação de diversos programas que operam em conjunto para criar uma plataforma de transação de ações entre utilizadores. Ao longo deste relatório, serão detalhados os principais aspetos técnicos e funcionais do sistema desenvolvido.

Neste relatório, serão abordados detalhes técnicos da implementação, incluindo a arquitetura do sistema, as estruturas de dados utilizadas, os mecanismos de comunicação e de sincronização, bem como outras decisões de design e implementação. O objetivo é fornecer uma visão abrangente do funcionamento do sistema e das soluções adotadas para alcançar seus objetivos.

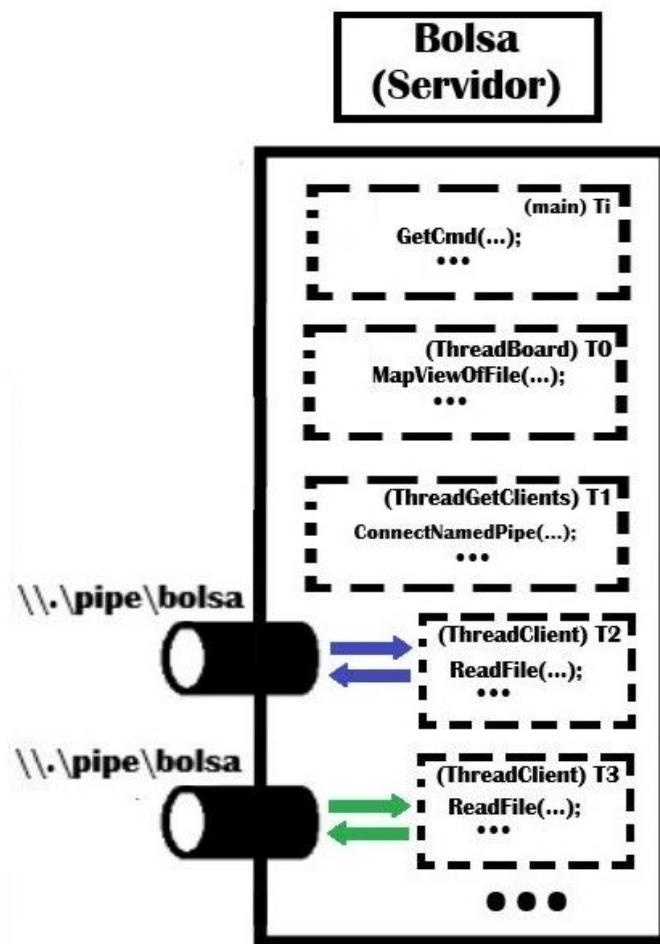
2. Programas



2.1 Bolsa

O programa da bolsa será composto por $3 + n$ *threads* sendo n o número de processos Cliente em execução no momento:

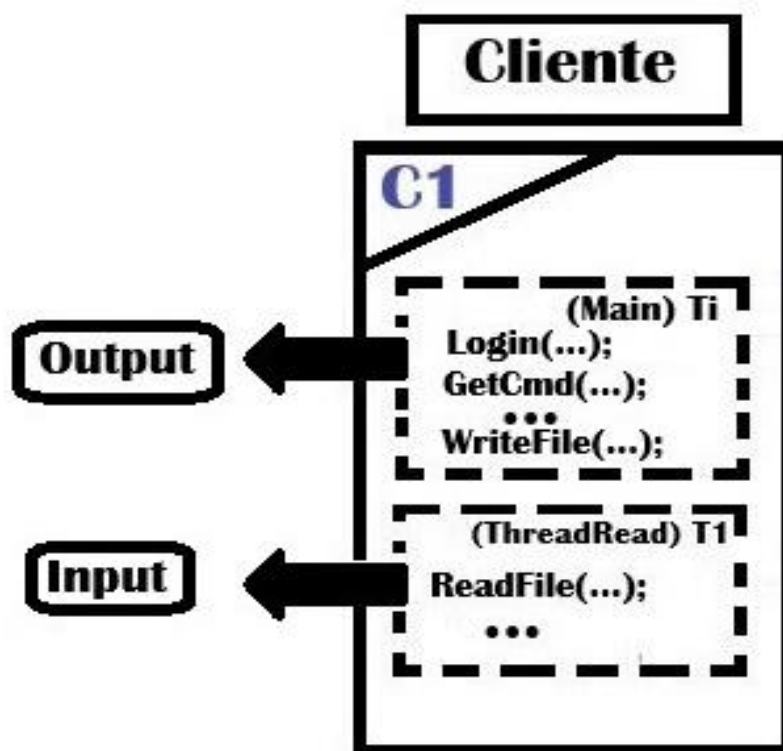
- **Main:** Após lançar as restantes *threads* irá aguardar pela introdução de comandos por parte do utilizador validando e executando os mesmos de seguida.
- **ThreadBoard:** Esta thread terá como única função atualizar a memória partilhada utilizada pelo(s) processo(s) *board* e *boardGUI* com as informações relevantes sobre as empresas que estão no top.
- **ThreadGetClients:** Aguarda que um cliente se conecte, cria uma instância do *named pipe* “bolsa” e lança uma *thread* para gerir a comunicação entre o cliente e a bolsa.
- **ThreadClient:** Esta *thread* terá a função de implementar um mecanismo simples de pergunta e resposta entre os clientes e a bolsa. Será responsável por receber as mensagens dos clientes, processá-las e enviar as respostas apropriadas.



2.2 Cliente

No programa cliente, será implementado um sistema com duas threads:

- **Main:** Irá aguardar pela introdução de comandos por parte do utilizador validando e executando os mesmos de seguida enviando informação para o processo bolsa através do *named pipe* se assim for necessário.
- **ThreadRead:** Assume a tarefa de receber as mensagens enviadas pelo processo Bolsa.



2.3 Board

O programa Board consiste em apenas uma *thread* que estará encarregue de ler a informação contida na memória partilhada com o processo Bolsa (SHM).



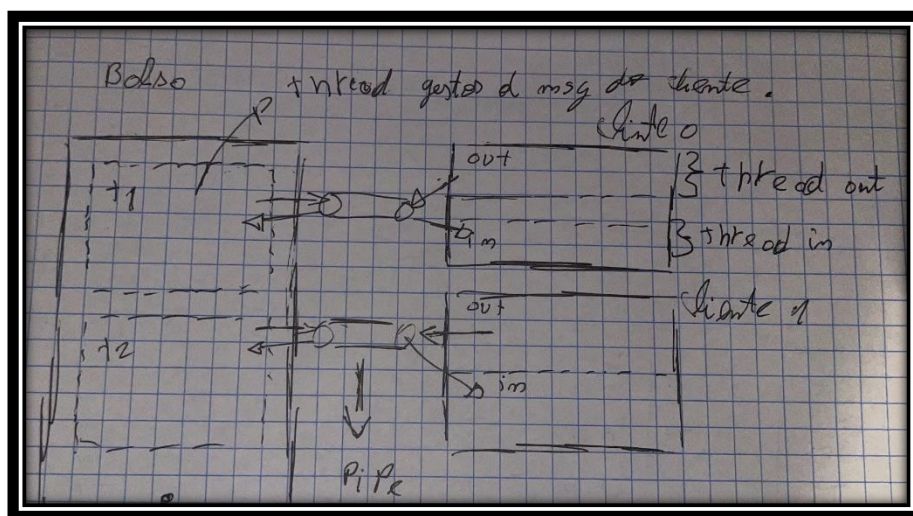
3. Comunicação entre Processos

3.1 Cliente ↔ Bolsa

A comunicação entre o Cliente e a Bolsa será feita através de *named pipes*.

No lado da Bolsa existirá uma thread responsável por aguardar a conexão de um cliente, após a conexão ser estabelecida será lançada uma thread dedicada a esse mesmo cliente. Esta thread será responsável por receber pedidos, processar informações e enviar as respetivas respostas.

No lado do cliente, uma thread será designada para receber a saída do servidor, enquanto outra será responsável por enviar as solicitações para o servidor



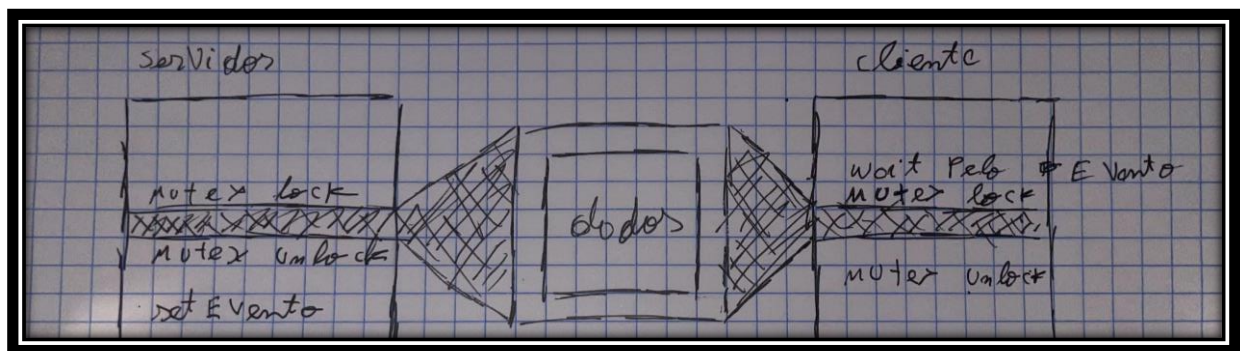
3.2 Bolsa ↔ Board

A comunicação entre a Bolsa e a Board será feita através de memória partilhada. O programa Bolsa terá a responsabilidade de inicializar e gerir a memória partilhada começando por criar um arquivo para a memória, de seguida, estabelecerá um mapeamento do mesmo (*file mapping*), onde a visualização (*view*) terá o tamanho exato de uma em bites de uma estrutura “empresa”.

Do lado da Board, será também criada uma visualização (*view*) para copiar as informações contidas na memória partilhada.

Existirá ainda um Evento que será acionado pela Bolsa sempre que as informações na memória partilhada forem atualizadas de maneira que o processo Board saiba quando voltar a recolher informação.

Para garantir a integridade dos dados, será utilizado um mecanismo de exclusão mútua (*mutex*) para proteger o acesso à memória partilhada.



```
typedef struct {
    TCHAR nome[TAM];
    DWORD cotacao;
    DWORD numDeAcao;
} empresa;
```

4. Estruturas de Dados Utilizadas

<i>Struct</i>	Bolsa	Cliente	<i>Board</i>
<pre>typedef struct { TCHAR nome[TAM]; DWORD cotacao; DWORD numDeAcao; } empresa;</pre>	X	X	X
<pre>typedef struct { TCHAR nome; TCHAR empresaNome[NUM_empresas]; DWORD numDeAcoes[NUM_empresas]; } carteiraDeAcoes;</pre>	X	X	
<pre>typedef struct { TCHAR username[TAM]; TCHAR password[TAM]; carteiraDeAcoes carteira; } useres;</pre>	X		