



Sistema de Gestão do Espaço Aéreo

Relatório Sistemas Operativos II

Departamento Engenharia Informática e de Sistemas (DEIS)

Artur Dias - 2018018358
Pedro Correia - 2018020558

Índice

Lista de Figuras	3
Descrição	4
-> Mecanismos de Comunicação	4
-> Mecanismos de sincronização	4
-> Estruturas de dados	5
Manual de Instruções	7

Lista de Figuras

Figura 1 - Estrutura Aeroporto.....	5
Figura 2 - Estrutura Avião	5
Figura 3 - Estrutura Memoria	6
Figura 4 - Estrutura TDados.....	6
Figura 5 - Criação de aeroportos.....	7
Figura 6 - Criação avião	7
Figura 7 - Listar aeroportos e aviões	7
Figura 8 - Exemplo de comandos.....	8

Descrição

-> Mecanismos de Comunicação

A estrutura “Memoria” funciona como um mecanismo de comunicação, nomeadamente, memória partilhada. Tanto o programa “control” como “avião”, têm acesso.

É criada uma zona onde é conhecido o número máximo de aviões e de aeroportos. 2 zonas para os aviões (entrada avião e saída avião).

Por fim, há um array de aviões, basicamente, quando aparece ou desaparece um avião, essa zona é atualizada.

-> Mecanismos de sincronização

No corrente desenvolvimento do projeto, estão a ser utilizados semáforos e mutexes.

Na comunicação Control-aviao estão a ser usados semáforos. Sempre que há uma alteração do número de aviões, o control acede ao semáforo de escrita e liberta o de leitura. No fim de o control libertar o semáforo de leitura, atualiza o array assim que necessário.

Quando se pretende aceder a variáveis usadas em mais que uma thread, então utiliza-se os mutexes.

-> Estruturas de dados

Foi criada uma estrutura, “**Aeroporto**”, que é usada para guardar a posição (coordenadas) e o nome do aeroporto criado.

```
typedef struct {  
    TCHAR nome[BUFFER];  
    int x, y;  
} Aeroporto;
```

Figura 1 - Estrutura Aeroporto

Temos também a estrutura “**Aviao**”, representa cada avião que se “liga”, e onde é guardado:

- O ID;
- As coordenadas do avião;
- A velocidade;
- Uma função para o próprio terminar ou não;
- Aeroporto Inicial e o Destino;
- Um identificador de eventos, onde, 0 -> Alerta e 1 -> Chegada ao aeroporto.

```
typedef struct {  
    DWORD id;  
    int x;  
    int y;  
    int velocidade;  
    bool terminar;  
    Aeroporto inicial;  
    Aeroporto destino;  
    HANDLE eventos[2];  
} Aviao;
```

Figura 2 - Estrutura Avião

A estrutura “**Memoria**” que é tudo o que é partilhado entre o avião e o controlador, nomeadamente:

- O número máximo de aeroportos;
- O número máximo de aviões;
- Número de aeroportos;
- Número de aviões;
- “Entrada” e “saída” de aviões, referente ao modelo “Produtor-consumidor”;
- Condição de paragem do programa;
- Um array onde serão colocados os itens do programa;

```
typedef struct {  
    int maxaerportos;  
    int maxavioes;  
    int naerportos;  
    int navioes;  
    int entAviao; // c  
    int saiAviao; // c  
    bool terminar; // c  
    Aviao avioes[TAM];  
} Memoria;
```

Figura 3 - Estrutura Memoria

“**TDados**” que é usada para passagem de informação para as threads.

```
typedef struct {  
    Aviao self;  
    Memoria* ptr_memoria;  
    HANDLE sem_avioes; //  
    HANDLE sem_vazios; //  
    HANDLE sem_itens; // i  
    HANDLE mutex; // mutex  
    HKEY chaveAerportos;  
    bool suspend;  
} TDados;
```

Figura 4 - Estrutura TDados

Manual de Instruções

Primeiramente abre-se o control e de seguida cria-se dois aeroportos, para testes, um inicial e um de destino, ambos com nome e de seguida com as coordenadas.

```
Introduza o nome do aeroporto: Lisboa
Introduza as coordenadas do novo aeroporto (x, y): 30, 40

Introduza a opção do comando que pretende executar:
1. Criar aeroporto
2. Suspender/Ativar registo de aviões
3. Listar tudo

Opção:
Introduza a opção do comando que pretende executar:
1. Criar aeroporto
2. Suspender/Ativar registo de aviões
3. Listar tudo

Opção: 1
Introduza o nome do aeroporto: Porto
Introduza as coordenadas do novo aeroporto (x, y): 41, 41
```

Figura 5 - Criação de aeroportos.

Depois disto, noutras janela cmd, cria-se os aviões no aeroporto pretendido (tendo em atenção ao limite, pois se são criado a mais, esses ficam em espera).

Comando disponível: avião <aeroporto> <velocidade>.

```
>aviao Lisboa 10
Aeroporto inicial definido em Lisboa localizado em 30, 40.
Aviao 8572 30 40

O avião encontra-se no aeroporto Lisboa localizado em 30, 40.
Introduza o comando que pretende:
```

Figura 6 - Criação avião

Na janela do “control” é possível listar todos os aeroportos e aviões criados.

```
Introduza a opção do comando que pretende executar:
1. Criar aeroporto
2. Suspender/Ativar registo de aviões
3. Listar tudo

Opção: 3

Aeroporto Lisboa, localizado em 30, 40.

Aeroporto Porto, localizado em 41, 41.
3
Avião 8572 na posição 30, 40.
3
Avião 2688 na posição 30, 40.
3
Avião 15576 na posição 30, 40.
```

Figura 7 - Listar aeroportos e aviões

Depois de criado é pedido para introduzir um comando.
Comandos disponíveis: “proximo <aeroporto>”, “inicia””.

```
Introduza o comando que pretende:
proximo Porto

Aeroporto destino definido como Porto localizado em 41, 41.

O avião encontra-se no aeroporto Lisboa localizado em 30, 40.

Introduza o comando que pretende:
inicia

Introduza 'fim' para terminar a viagem.
Avião na posição 31 40
Avião na posição 32 40
```

Figura 8 - Exemplo de comandos