

Aluno: Samuel Rodrigues Viana de Faria

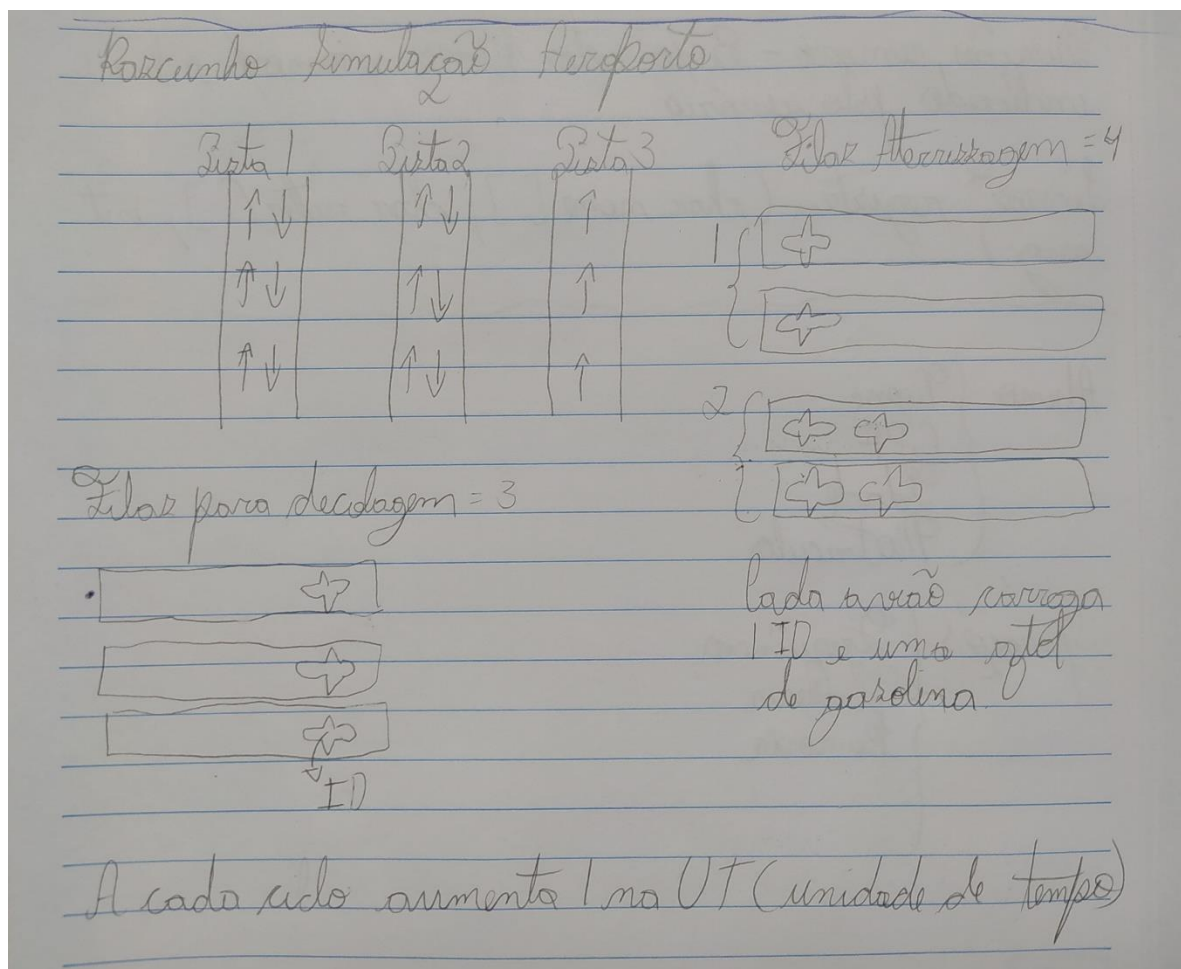
Simulador Aeroporto

1 – Resumo

Este trabalho prático teve como objetivo a criação de um simulador de aeroporto, onde o processo de aterrissagens e decolagens é controlado através de filas simplesmente encadeadas que acumulam os aviões até serem desenfileirados de acordo com a disponibilidade em três diferentes pistas. Este documento tem a finalidade de explicar o funcionamento do aeroporto, para um melhor entendimento.

2 – Aeroporto

Inicialmente, foi criado pelo aluno um rascunho bem simples com o pensamento inicial sobre o que realizar para o funcionamento do simulador de aeroporto.



Através do esboço inicial, foi decidido que haveria 4 filas para aterrissagem e 3 para decolagem de aviões, e somente 3 pistas para o desenfileiramento dos mesmos, onde uma delas

seria exclusiva para decolagens em situações normais e as outras duas para aterrissagem e decolagens.

2.1 TADs/Structs

Todos os TADs e Structs foram definidos no arquivo de cabeçalho `operacoes.h`. Para o controle dos aviões foi criado um TAD que contém a seguinte struct:

Struct `Avioes` {

- ID do avião
 - Quantidade de gasolina
 - Quantidade de ciclos na fila
- }

Para o controle das filas foi criado um TAD que contém as seguintes structs:

Struct `Celula` {

- Avião contido na fila.
 - Apontador para o próximo avião
- }

Struct `Fila` {

- Apontador para o primeiro e último avião da fila.
 - Tamanho da fila
- }

Para exibir todas as estatísticas foi criada a seguinte struct:

Struct `estat` {

- Quantidade parcial e total de decolagens e aterrissagens.
 - Quantidade parcial e total de quedas e aterrissagens de emergência.
 - Quantidade parcial e total de ciclos dos aviões que são retirados da fila
- }

2.2 Função main e sub-rotinas

int main(int argc, char argv):** Na função main ocorre a criação de várias variáveis que auxiliam no controle do aeroporto, dentre elas estão as variáveis responsáveis pelas pistas e os IDs de aterrissagem e decolagem que possuem apontadores para serem usados em diferentes funções. Também é feito a alocação/criação de todas as filas e de uma variável responsável pelas estatísticas, todas as variáveis que funcionam como contadores são iniciadas em 0. Por seguinte, todas as filas são iniciadas e assim é pedido uma interação com o usuário onde o mesmo escolhe a quantidade de ciclos da simulação do aeroporto. Em seguida se inicia um “do” que funciona enquanto todos os ciclos terminarem. Dentro de cada ciclo são geradas as filas com aviões de forma aleatória e ocorre a mostragem da mesma na tela, em seguida ocorre todo o processo de aterrissagem e decolagem e por final de cada ciclo são mostradas as estatísticas parciais e os contadores parciais são zerados. No final são mostradas as estatísticas finais após todos os ciclos.

As sub-rotinas estão presentes no arquivo `operacoes.c` e suas definições estão no arquivo `operacoes.h`.

Sub-rotinas

void FFVazia(Fila *fila): Responsável por “iniciar” a fila, aloca a primeira posição e iguala a última posição com a primeira, seta a próxima posição como NULL e iguala o tamanho da fila a 0.

void Enfileira(Fila *fila, int *ID, int gastemp): Responsável por inserir um avião na última posição da fila. O ID, a gasolina inicial e a fila em questão são recebidas como parâmetro. Primeiramente ocorre uma verificação de fila vazia, se não, ocorre a inserção do avião na última posição da fila e seu ID e gasolina são igualados aos recebidos por parâmetro. Em seguida, a quantidade de ciclos do avião é iniciada em 0, o tamanho da fila é aumentado em 1, a próxima posição da fila é alocada e o ID é somado em 2 para manter os mesmos em números pares.

void Imprime(Fila *fila): Recebe a fila em questão por parâmetro e percorre toda ela com a ajuda de uma variável auxiliar a fim de imprimir todos os dados dos aviões presentes na fila.

int Vazia(Fila *fila): Responsável por verificar se uma fila está vazia, simplesmente comparando sua primeira e última posição.

void DesenfileiraA(Fila *fila, estat *Estatisticas): Responsável pelo desenfileiramento de filas de aterrissagem. Recebe como parâmetro a fila em questão e a variável de estatísticas. Primeiramente é feita a verificação de existência da fila, caso exista, o primeiro avião é retirado da fila, em seguida é somado a um contador parcial e total a quantidade de ciclos que o avião permaneceu na fila. Também ocorre uma verificação da quantidade de gasolina, caso ela esteja em 0 ou abaixo é contabilizado em +1 nas estatísticas de queda de aviões parcial e total. Por fim é diminuído em 1 o tamanho da fila e somado 1 nas estatísticas de aterrissagem parcial e total.

void DesenfileiraD(Fila *fila, estat *Estatisticas): Responsável pelo desenfileiramento de filas de decolagem. Recebe como parâmetro a fila em questão e a variável de estatísticas. Primeiramente é feita a verificação de existência da fila, caso exista, o primeiro avião é retirado da fila, em seguida é somado a um contador parcial e total a quantidade de ciclos que o avião permaneceu na fila. Por fim é diminuído em 1 o tamanho da fila e somado 1 nas estatísticas de decolagem parcial e total.

void RemoveGas(Fila *filaA1, Fila *filaA2, Fila *filaA3, Fila *filaA4, int UT): Responsável por diminuir o contador de gasolina em 1 de cada avião por ciclo. Recebe como parâmetro todas as filas de aterrissagem e a variável UT (Unidade de tempo). Primeiramente é realizado uma verificação onde só acontecerá a remoção de gasolina após o primeiro ciclo. Em seguida todas as filas são percorridas com a ajuda de um auxiliar a fim de decrementar em 1 a gasolina de todos os aviões presentes nas filas.

void VerificarEmergencia(Fila *filaA1, Fila *filaA2, Fila *filaA3, Fila *filaA4, int *p1, int *p2, int *p3, estat *Estatisticas): Responsável por realizar aterrissagens de emergência onde o contador de gasolina do avião está em 1. Recebe como parâmetro todas as filas de aterrissagem, pistas e a variável de estatísticas. Primeiramente ocorre a verificação da existência da filaA1 e em seguida com a ajuda de um auxiliar verifica-se se o avião que está na primeira posição da fila está em emergência, caso esteja é realizado o processo de retirada do avião com a função de desenfileirar e a ocupação prioritária da pista 3, caso não esteja é realizada a verificação de emergência do segundo avião da fila, se a condição for satisfeita é realizado o pouso do primeiro avião e em seguida do segundo nas pistas disponíveis. Esse processo é repetido nos restantes das filas e sempre que ocorre alguma emergência é somado em 1 nas estatísticas parciais e totais de quedas.

void Aterrissagem(Fila *filaA1, Fila *filaA2, Fila *filaA3, Fila *filaA4, int *p1, int *p2, int *p3, estat *Estatisticas): Responsável pelo processo de aterrissagem dos aviões presentes nas filas de aterrissagem. Recebe como parâmetro as filas de aterrissagem, as pistas e a variável de estatísticas. Primeiramente ocorre uma verificação geral de existência das filas, em seguida são priorizadas as situações de emergência e a situação das pistas é mostrada após o processo. Após isso ocorre a verificação de situação da pista 1, caso ela esteja disponível é realizada uma comparação entre os tamanhos das filas A1 e A2 e a retirada de avião é realizada na que for maior. Em seguida ocorre o mesmo processo, onde a pista 2 vai ser ocupada por um avião da filaA3 ou um da filaA4.

void Decolagem(Fila *filaD1, Fila *filaD2, Fila *filaD3, int *p1, int *p2, int *p3, estat *Estatisticas): Responsável pelo processo de decolagem dos aviões presentes nas filas de decolagem. Recebe como parâmetro as filas de decolagem, as pistas e a variável de estatísticas. Primeiramente ocorre uma verificação geral de existência das filas, em seguida ocorre a verificação de situação da pista 1, caso ela esteja disponível é realizado desenfileiramento de um avião da filaD1, o mesmo processo se repete com as pistas 2 e 3. Após a ocupação ou não de todas as pistas são feitas mais condições a fim de obter um melhor aproveitamento das pistas e aumentar a vazão de aviões.

void ContCiclos(Fila *filaA1, Fila *filaA2, Fila *filaA3, Fila *filaA4, Fila *filaD1, Fila *filaD2, Fila *filaD3): Responsável por aumentar em 1 o contador de ciclos de cada avião por ciclo. Recebe como parâmetro todas as filas de aterrissagem e decolagem. Antes de qualquer processo é verificado a existência da fila. Por fim todas as filas são percorridas com a ajuda de um auxiliar a fim de aumentar em 1 o contador de ciclos de todos os aviões.