

Projeto 1 - Sistema de Gerenciamento de Tráfego Aéreo

Lucca Magalhães Boselli Couto - 222011552

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)

CIC0202 - Programação Concorrente

16 de janeiro de 2025

1. Introdução

O controle do tráfego aéreo é uma área de extrema importância para a aviação, garantindo a segurança e a eficiência nas operações de pousos e decolagens. Com o aumento da demanda por voos comerciais, helicópteros de transporte e aeronaves de emergência, torna-se essencial o desenvolvimento de sistemas capazes de gerenciar o uso das pistas de forma dinâmica e priorizada.

Este projeto visa explorar a aplicação de técnicas de programação concorrente para simular um sistema de controle de tráfego aéreo eficiente. O objetivo é gerenciar o acesso de diferentes tipos de aeronaves a uma única pista, levando em consideração fatores como prioridades de emergência, condições climáticas e número de aeronaves em espera.

A simulação desenvolvida utiliza mecanismos de sincronização como mutexes e variáveis de condição, permitindo a modelagem precisa de interações e conflitos entre aviões e helicópteros. Este relatório apresenta a formalização do problema, a descrição do algoritmo proposto e os resultados obtidos, demonstrando a viabilidade e os desafios associados à implementação de sistemas de controle concorrente para cenários críticos.

2. Formalização do Problema

O problema abordado neste projeto consiste no gerenciamento eficiente e seguro do tráfego aéreo em um aeroporto com uma única pista. Esse cenário simula a necessidade de coordenar o acesso de diferentes tipos de aeronaves, como helicópteros e aviões, considerando as seguintes restrições e condições:

1. Prioridade de Emergência

Aeronaves em situação de emergência têm prioridade absoluta para utilização da pista. Esse requisito é essencial para garantir a segurança em casos críticos.

2. Condições Meteorológicas

As operações de pouso e decolagem só podem ocorrer sob condições meteorológicas favoráveis. Quando as condições climáticas se deterioram, a pista deve permanecer fechada, independente se há solicitações de pouso de emergência.

3. Prioridade entre Helicópteros e Aviões

Em situações normais, helicópteros possuem prioridade sobre aviões no acesso à pista. Caso existam helicópteros aguardando, eles devem ser atendidos antes dos aviões, desde que não haja emergências em andamento.

4. Conflito de Prioridade

Em casos em que não há emergências ou interferências meteorológicas, o acesso à pista é concedido de maneira ordenada, respeitando as filas de espera e as prioridades estabelecidas.

5. Finalização do Sistema

O aeroporto encerra suas operações quando não houver mais aeronaves aguardando o uso da pista, seja por ausência de novas demandas ou pela conclusão das operações pendentes.

O problema, portanto, consiste em projetar um sistema que satisfaça essas condições, permitindo o uso eficiente da pista, minimizando tempos de espera e priorizando emergências e helicópteros em situações regulares. Para isso, é necessário um modelo que implemente sincronização de threads, controle de acesso concorrente e tratamento dinâmico das condições climáticas e de prioridade.

3. Descrição do Algoritmo Desenvolvido para Solução do Problema

O algoritmo desenvolvido utiliza a biblioteca POSIX Pthreads para gerenciar a criação, execução e sincronização de threads no sistema. Ele é responsável por coordenar as ações de aviões e helicópteros que aguardam o uso de uma pista de pouso, garantindo que as operações sejam realizadas de maneira eficiente e segura, mesmo em cenários críticos, como emergências ou condições meteorológicas adversas.

3.1. Estruturas de Controle

O sistema utiliza variáveis compartilhadas e mecanismos de sincronização para gerenciar o acesso à pista:

- **Variáveis Compartilhadas:** Representam o estado do sistema, como a disponibilidade da pista (`pista_livre`), o número de aeronaves aguardando e a situação de emergência.
- **Mutex e Variáveis de Condição:** São usados para coordenar o acesso à pista, garantindo que as operações sejam realizadas de maneira segura e ordenada, evitando condições de corrida.

3.2. Priorização e Lógica de Espera

O algoritmo define regras claras de prioridade:

1. **Emergências:** Aeronaves em emergência têm prioridade absoluta. Isso é implementado monitorando uma variável específica (`emergencia_ativa`) que bloqueia todas as outras operações enquanto a emergência está em andamento. É importante frisar que, mesmo em condições de emergência, helicópteros ainda possuem prioridade de pouso em relação aos aviões.
2. **Helicópteros:** Em situações normais, helicópteros têm prioridade sobre aviões. Isso é gerenciado verificando se há helicópteros aguardando antes de liberar a pista para aviões.
3. **Condições Meteorológicas:** Nenhuma aeronave pode utilizar a pista enquanto as condições meteorológicas forem ruins, de modo que a pista só é liberada quando as condições climáticas estiverem boas.

3.3. Lógica de Liberação da Pista

Após uma aeronave utilizar a pista:

- O sistema libera a pista marcando-a como disponível.

- Dependendo da situação, uma das variáveis de condição é sinalizada para liberar as threads correspondentes:
 - Aeronaves em emergência.
 - Helicópteros aguardando.
 - Aviões aguardando.

3.4. Paralelismo e Controle Dinâmico

- **Aeronaves:** Cada aeronave é representada por uma thread separada, que entra em uma fila de espera e aguarda sua vez para utilizar a pista, respeitando as prioridades.
- **Clima:** Uma thread dedicada simula mudanças climáticas, alterando as condições meteorológicas de maneira dinâmica e notificando as aeronaves quando as condições melhoram.
- **Monitoramento:** Uma thread adicional exibe o status do sistema periodicamente, mostrando a quantidade de aeronaves aguardando, o estado da pista e as condições meteorológicas.

3.5. Encerramento do Sistema

O sistema verifica continuamente o número de aeronaves aguardando. Quando não houver helicópteros ou aviões na fila, o aeroporto é fechado, e o programa encerra sua execução.

3.6. Fluxo Geral

1. Uma aeronave solicita o uso da pista.
2. Verifica-se a condição de emergência, a prioridade do tipo de aeronave e as condições meteorológicas.
3. A aeronave entra em espera, caso necessário, ou utiliza a pista.
4. Após o uso, a pista é liberada, e o próximo processo na fila é sinalizado.

Segue vídeo do sistema funcionando: [Vídeo de Explicação do Projeto](#)

Segue o link para o repositório do projeto contendo o código fonte: [Repositório do Projeto no GitHub](#).

4. Conclusão

A análise do sistema de controle de tráfego aéreo proposta neste projeto demonstra uma abordagem robusta e eficiente para gerenciar a utilização de uma única pista, levando em consideração diversos fatores críticos, como a prioridade de aeronaves em emergência, as condições meteorológicas e a diferenciação de prioridades entre helicópteros e aviões. O uso de técnicas de programação concorrente, como mutexes e variáveis de condição, assegura que as interações entre as aeronaves e a pista sejam realizadas de maneira segura e ordenada, evitando problemas de sincronização e condições de corrida.

O algoritmo desenvolvido é eficaz na gestão das diferentes situações que podem surgir, como a emergência de aeronaves, a variação das condições climáticas e a organização de filas de espera de acordo com as prioridades preestabelecidas. A implementação de threads dedicadas para a simulação das mudanças climáticas e para o monitoramento do status do sistema garante a flexibilidade e o controle dinâmico necessários para um ambiente de tráfego aéreo realista.

Além disso, o sistema de encerramento da operação do aeroporto quando não houver mais aeronaves aguardando para utilizar a pista mostra uma boa prática na modelagem de sistemas que simulam o ciclo de vida de um processo, minimizando o uso de recursos quando não é necessário. O projeto se destaca por sua capacidade de integrar diferentes aspectos de controle e sincronização em um único modelo de tráfego aéreo, oferecendo uma solução eficiente e escalável para problemas reais de gestão de tráfego em aeroportos com pistas limitadas.

Portanto, o desenvolvimento desse sistema não apenas resolve o problema de coordenação de aeronaves em uma pista única, mas também pode servir de base para aprimoramentos futuros, considerando cenários mais complexos ou aumentando a robustez do sistema para lidar com uma maior variedade de condições e tipos de aeronaves.

Referências

[Tutorial POSIX Threads](#)

[O que é Condição de Corrida?](#)

[Documentação da Linguagem C](#)