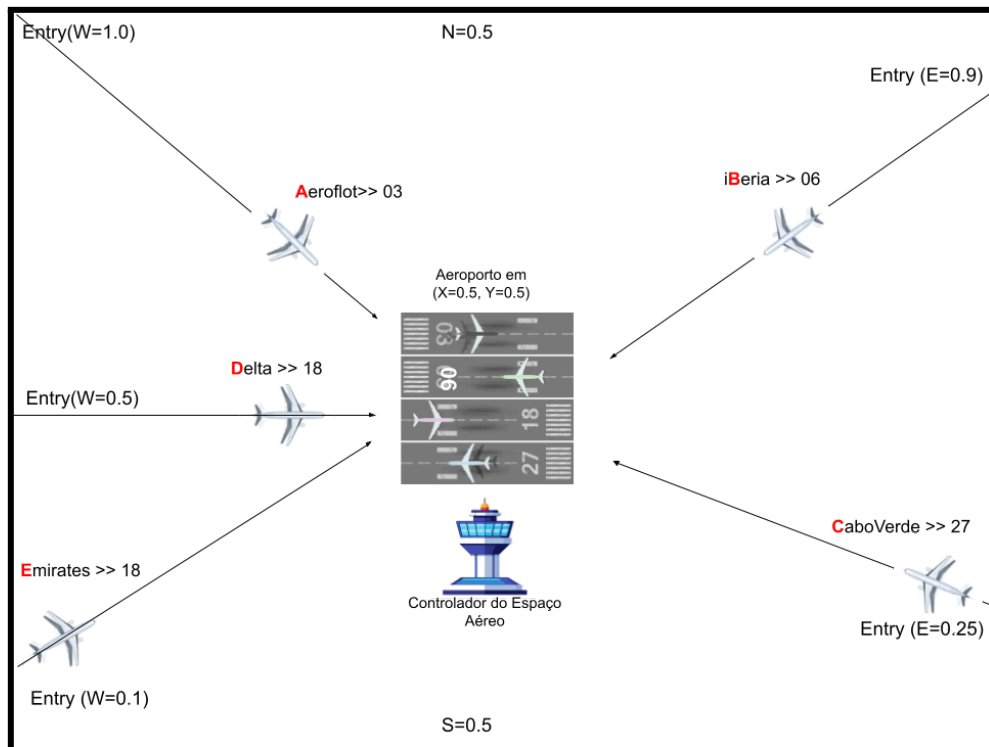


Trabalho 1 – INF1316 - Sistemas Operacionais -2025.1

Título: Simulação de um Sistema de Controle de Tráfego Aéreo



Controlador do Espaço Aéreo de um aeroporto de quatro pistas (#03, #06, #18, #27), e aproximação de cinco aeronaves (A,B,C,D,E) entrando no espaço em diferentes pontos (**Entry()**) dos lados W e E, em momentos diferentes e indicando uma pista de pouso de preferência (mas que pode ser trocada pelo Controlador do Espaço Aéreo). O objetivo é não deixar duas ou mais aeronaves se aproximarem por mais de 0.1 na coordenada Y (p.ex. aeronaves D e E parecem estar em rota de colisão).

Objetivo:

O objetivo é implementar um programa em linguagem C que simula um sistema de controle de tráfego aéreo em torno de um aeroporto, onde processos independentes (representando aeronaves) interagem entre si usando memória compartilhada e com o tratamento de sinais enviados pela controladora de espaço aéreo (processo pai), para fins de escalonar os pousos de N aeronaves (processos filho).

Descrição:

Imagine um cenário onde um controlador de tráfego aéreo precisa gerenciar várias aeronaves em um espaço aéreo limitado representado pelo quadrado acima de dimensões (1,0 x 1,0). As aeronaves entram no espaço aéreo ou da esquerda/oeste (W) ou da direita/leste (E) em determinada posição e voam com uma velocidade constante horizontal na direção das pistas do aeroporto, sendo duas para os aviões procedentes de W (pista 18 e 03) e outras duas pistas para aviões procedentes da direção E (pista 27 e 06). Ao entrar no espaço aéreo, cada aeronave indica a sua pista de preferência. A tarefa do controlador do espaço aéreo é evitar que duas ou mais aeronaves se aproximem demais (menos de 0.1 de distância na vertical, e 0.1 distância na horizontal) ao se moverem na direção da pista. Todos os aviões se movem a uma velocidade de 0.05/segundo no eixo horizontal. O progresso (de

movimento) de cada aeronave deve ser simulado através de um TimeSharing circular entre eles.

Além disso, cada aeronave (processo filho) deve ser capaz de se comunicar com a controladora (processo pai), além de obedecer a comandos de controle em tempo real. Resumindo, o seu programa deve ter um processo pai que cria N processos filho (um para cada aeronave) e faz-as se mover e ficar controlando na memória compartilhada se duas aeronaves estão se aproximando demais, e se planejam pousar na mesma pista. Para evitar colisões, o processo da controladora deve enviar sinais SIGUSR para as aeronaves potencialmente em rota de colisão. Aeronaves voando no mesmo sentido e que forem pousar em pistas diferentes não correm risco de colidir.

1. Criação de Processos:

- O programa deve criar múltiplos processos filhos, cada um representando uma aeronave que está em rota. Cada aeronave deve executar uma tarefa simples, como "voar" (simular uma contagem de distância percorrida) e comunicar sua posição ao controlador através da memória compartilhada. Quando cada processo representando uma aeronave é criada, escolhe-se randomicamente:

- o lado de entrada (E, W)
- a coordenada Y do ponto de entrada
- sorteio de um atraso (dentre 0 e 2 segundos)) até que a aeronave de fato comece a entrar no espaço aéreo (por exemplo aeronaves A e D entraram antes do que aeronaves C e E)
- qual é a pista de pouso preferencia, de acordo com o lado de entrada

2. Uso de Sinais:

- O controlador (processo pai) deve ser capaz de enviar sinais para as aeronaves, utilizando `SIGCONT` e `SIGSTOP` (para implementar o Timesharing em modo RoundRobin), e SIGUSR1 para suspender temporariamente o avanço de uma aeronave (simulando assim uma redução de velocidade), e SIGUSR2 para direcionando de uma aeronave à uma pista de pouso alternativa à que ele tinha solicitado. Tanto SIGUSR1 e SIGUSR2 devem funcionar como "toggle" (flipflop), que permite alternar entre dois valores (e.g. Pista 6 <-> Pista 27)
- Seu programa também deve fazer com que a controladora do tráfego aéreo possa verificar se uma vez enviado SIGUSR1/ SIGUSR2 se a aeronave alvo de fato está processando corretamente o sinal, e está alterando o seu estado de acordo.
- Quando uma aeronave chega no aeroporto (coordenadas [0.5,0.5]) e pouso instantaneamente (não há demora por taxiamento), o processo correspondente deve terminar com status normal, avisando a controladora.
- E caso uma colisão seja inevitável (p.ex. duas aeronaves a menos de [0.1, 0.1] de distância) e sem a possibilidade de reduzir suficientemente a sua velocidade, então o controlador deve ordenar que uma das aeronaves remeta o pouso (enviando um SIGKILL)

3. Memória Compartilhada:

Será necessário você implementar comunicação interprocesso através de memória compartilhada, onde cada aeronave armazena e divulga sua posição e status atuais. Este

status inclui a posição no espaço aéreo, (coordenada bidimensional, por exemplo, [0.7, 0.35] onde a altura de voo será desprezível, e a velocidade. Além disso, deve indicar o número de pista de preferência. (O controlador deve acessar e exibir essas informações em tempo real, permitindo monitorar o tráfego no espaço aéreo).

4. Escalonamento de Processos:

O seu programa deve implementar um algoritmo de escalonamento que assegure que as aeronaves possam "voar" no espaço aéreo sem colisões, priorizando aviões em situações críticas e garantindo que todos tenham a oportunidade de se mover (pode-se usar um algoritmo de round-robin ou prioridades baseadas na altitude).

5. Interface do Usuário:

O programa deve ter uma interface simples de linha de comando onde o usuário pode iniciar, pausar, retomar e finalizar o movimento das aeronaves, além de visualizar a posição e a direção de voo corrente de todas as aeronaves sendo controladas

Requisitos:

- O código deve ser bem documentado, com comentários explicativos sobre as funções e a lógica utilizada., assim como dando exemplos dos parâmetros da linha de comando (p.exemplo, quantos aviões irão surgir).
- Você deve mostrar que o seu controlador de espaço aéreo consegue escalonar $N=5$, $N=10$ e $N=20$ voos mais ou menos simultâneos.
- O programa deve ser testado e rodar em um ambiente Linux.
- Mantendo a conformidade com tudo que foi especificado acima, você tem liberdade para fazer algumas variações que tornam o problema de controle mais complexo, como por exemplo:
 - as aeronaves podem iniciar com velocidades horizontais diferentes, ou
 - levar em conta altura de voo (coordenada Z) e permitir que um avião insista em pousar uma pista, tendo assim que retardar vários voos ou mudar de altura de voo para outro pouse e que não haja colisões.

Entrega:

Deve ser entregue o código fonte completo, e o relatório explicativo sobre a o uso (parâmetros de chamada), implementação e os testes realizados

Data de Entrega: **6 de maio de 2025, até as 23:59**

Apresentação dos grupos: **13 de maio, 11-13 hs**

ATENÇÃO: é terminantemente proibido usar qualquer ferramenta de LLM/GenAI para a programação. Se eu perceber que você entendeu apenas mais ou menos o que o seu programa faz, a nota será zero.

Boa sorte e mãos à obra!