



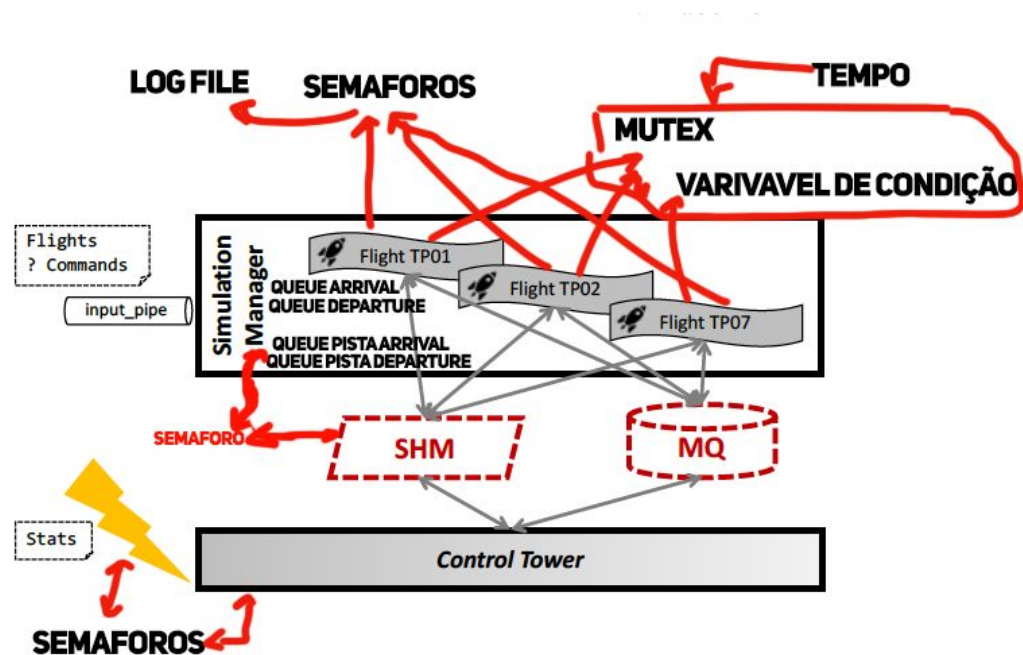
# SISTEMAS OPERATIVOS

## RELATÓRIO

Renato Santos n.º 2015237457 - Tempo total despendido: 100h

Simão Brito n.º 2015243394 - Tempo total despendido: 75h

-> ARQUITETURA



-> ESCALONAMENTO DOS VOOS:

Vamos comparando os tempos dos nós cabeçalhos das duas listas de espera para as pistas. O que tiver tempo menor, é escolhido primeira para a pista. No entanto se existirem duas chegadas ou duas partidas que possam ser feitas em lugar de um voo sozinho, opta-se pelos dois voos do mesmo tipo.

-> FUNCIONAMENTO GLOBAL DO PROGRAMA

A informação sobre os voos é recebida pelo pipe, devidamente confirmada e o voo é colocado numa lista inicial ordenada pelo "init". Depois aguarda-se até ao momento de "init" para iniciar a thread do respectivo voo e muda-se o voo para uma thread ordenada pelo momento de descolagem ou de aterragem (init + eta). Assim que chega o momento de aterragem a torre de controlo confirmar a disponibilidade da pista e decide colocar o voo em espera ou iniciar a ação de imediato caso existam pistas livres. É feita a descolagem e a thread é libertada e a pista é declarada como livre para receber mais voos.

-> SHARED MEMORY

A shared memory consiste em duas listas (voos de partida e de chegada) onde é armazenada toda a informação pertinente para a organização do voo por parte da torre de controlo e onde é dada a pista e a autorização para a aterragem ou descolagem dos voos.

-> NAMED PIPE

Recebemos os vôos no pipe, e vamos colocando os voos numa fila de espera. Quando chegar o INIT do voo, é criada a thread voo e é removido o nó cabeçalho da lista ligada.

-> MESSAGE QUEUE

Através da Message Queue, é possível trocar mensagens entre cada thread voo e a torre de controlo.

-> THREADS RELATIVAS AOS VOOS

Cada thread só é iniciada no momento de "init" dos voos, assim que é criada utiliza a message queue para informar a torre de controlo da sua existência e para saber qual a slot na shared memory que lhe foi atribuído. Depois disso aguarda pelo momento de iniciar a aterragem ou descolagem e caso tenha autorização da torre de controlo procede à aterragem ou descolagem. No final a thread é libertada e a pista utilizada é dada como livre.

-> MECANISMOS DE SINCRONIZAÇÃO

Utilizamos semáforos para controlar o acesso a processos, no caso de querermos inserir ou ir buscar algum elemento.

Para as threads voo esperarem que chegue o tempo desejado, temos uma variável de condição que verifica sempre o tempo atual com o tempo desejado, e depois procede para a operação do voo.