

Distributed Systems - Mutual Exclusion

O que é Exclusão Mútua?

Mecanismo que garante que apenas um processo por vez pode acessar um recurso compartilhado (ex: banco de dados, arquivo, seção crítica).

Em sistemas distribuídos, os processos estão em máquinas diferentes e a comunicação é feita via rede.

Objetivo principal:

Evitar condições de corrida (race conditions), inconsistência de dados e conflitos de acesso.

Tipos:

centralized algorithm

distributed algorithm

token-ring algorithm

decentralized algorithm

Token-ring algorithm

Algoritmo de exclusão mútua distribuída.

Cria um anel lógico entre os processos.

Cada processo sabe quem é o próximo no anel.

Um único token circula entre os processos.

Como o Token Ring Funciona

Inicialização: Um processo (ex: Processo 0) recebe o token

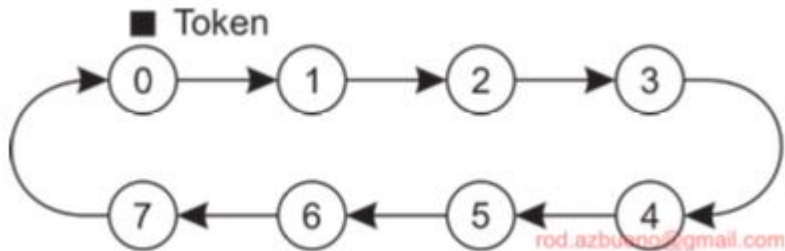
O token circula de processo em processo, na ordem do anel

Quando um processo recebe o token:

- Se quiser acessar a seção crítica → Entra → Executa → Sai → Passa o token.

- Se não quiser → Passa o token imediatamente.

Ninguém pode entrar duas vezes seguidas com o mesmo token.



Garantias Oferecidas

- Exclusão mútua garantida: Só o processo com o token pode acessar o recurso.
- Sem Starvation: Todos os processos recebem o token em sua vez.
 - a. Starvation = Fome / Inanição:
Quando um processo fica esperando indefinidamente e nunca consegue acessar o recurso.
 - b. O Token Ring evita isso, porque o token circula por todos.
- Ordem definida: O token segue sempre a mesma sequência.

Problema de Falhas no Token Ring

- Se um processo **cai (crash)** ou **fica offline**, o token pode ficar "preso" tentando passar para ele.
- Se o token for perdido (ex: processo que tinha o token falha), o sistema precisa de uma forma de recuperação.

Recuperação de Falhas - Abordagem Simples

- O processo tenta passar o token para o próximo.
- Se falhar várias vezes seguidas (ex: 10 tentativas), o processo considera o vizinho como "morto".
- O token é então passado para o próximo processo da lista.
- A lista de participantes fica localmente atualizada em cada nó.

Casos de Uso Reais

- Jogos online baseados em turnos.
- Sistemas de controle de acesso a recursos compartilhados.

Show me the code:

<https://github.com/rodolfobueno/distributed-system-mutual-exclusion>