

30-769

Sistemas Distribuídos

MSc. Fernando Schubert

EXCLUSÃO MÚTUA

- A exclusão mútua é uma propriedade de controle de concorrência que é introduzida para prevenir condições de corrida.
- É a exigência de que um processo não possa entrar em sua seção crítica enquanto outro processo concorrente está presente ou executando em sua seção crítica, ou seja, apenas um processo é permitido executar a seção crítica em qualquer instante de tempo.

EXCLUSÃO MÚTUA

- Exclusão mútua em sistema de computador único vs. sistema distribuído:
 - **Em um sistema de computador único**, a memória e outros recursos são compartilhados entre diferentes processos.
 - O status dos recursos compartilhados e o status dos usuários estão facilmente disponíveis na memória compartilhada, então, com a ajuda de variáveis compartilhadas (por exemplo: Semáforos), o problema de exclusão mútua pode ser facilmente resolvido.
 - **Em sistemas distribuídos**, não temos memória compartilhada nem um relógio físico comum e, portanto, não podemos resolver o problema de exclusão mútua usando variáveis compartilhadas.
 - Para eliminar o problema de exclusão mútua em um sistema distribuído, é utilizada uma abordagem baseada em passagem de mensagens.
 - Um site em um sistema distribuído não tem informações completas sobre o estado do sistema devido à falta de memória compartilhada e de um relógio físico comum.

EXCLUSÃO MÚTUA

- Uma questão fundamental em SD é a concorrência e a colaboração entre vários processos
- Como garantir que o acesso concorrente de recursos não gere situações de inconsistência de dados?
- São necessárias soluções que garantam o acesso mutualmente exclusivo pelos processos
- Algoritmos distribuídos de exclusão mútua podem ser classificados em duas categorias:
 - Baseados em ficha (token)
 - Baseados em permissão

EXCLUSÃO MÚTUA

Exclusão Mútua Baseada em Token com Números de Sequência

1. **Token Único:** Há um único token compartilhado entre todos os sites no sistema distribuído. A posse deste token concede ao site o direito de entrar na seção crítica.
2. **Números de Sequência:** Cada solicitação para a seção crítica é acompanhada por um número de sequência. Este número de sequência ajuda a distinguir entre solicitações antigas e atuais, garantindo que a solicitação mais recente seja tratada corretamente.
3. **Ordenação de Solicitações:** Os números de sequência são usados para ordenar as solicitações. Um número de sequência menor indica uma solicitação mais antiga. Quando um site solicita entrada na seção crítica, ele gera um novo número de sequência, que é maior do que qualquer número usado anteriormente.

EXCLUSÃO MÚTUA

Exclusão Mútua Baseada em Token com Números de Sequência

4. Transmissão de Solicitações e Fila de Solicitações de Token: Quando um site deseja entrar na seção crítica, ele transmite sua solicitação junto com o número de sequência para todos os outros sites. Cada site mantém uma fila de solicitações, ordenada pelos números de sequência.
5. Passagem de Token:
 - a. Se um site possui o token e recebe uma solicitação com um número de sequência maior do que o seu próprio, ele coloca a solicitação na fila.
 - b. Quando um site termina de executar na sua seção crítica, ele passa o token para o próximo site na fila, com base nos números de sequência
6. Exclusão Mútua: A exclusão mútua é garantida porque apenas o site que possui o token pode entrar na seção crítica. O token é único, garantindo que nenhum dois sites possam entrar na seção crítica simultaneamente.

EXCLUSÃO MÚTUA

Exclusão Mútua Baseada em Token com Números de Sequência

- **Benefícios**

- Exclusão Mútua: A singularidade do token garante que apenas um site possa estar na seção crítica a qualquer momento.
- Ordenação: Os números de sequência garantem que as solicitações sejam tratadas na ordem correta, impedindo que solicitações antigas sejam processadas após solicitações mais recentes.
- Descentralização: Esta abordagem pode funcionar de maneira descentralizada, sem um único ponto de falha, pois o token pode ser passado entre os sites.

EXCLUSÃO MÚTUA

Exclusão Mútua Baseada em Token com Números de Sequência

- **Desafios**

- Perda de Token: Se o token for perdido (por exemplo, devido a uma falha no site), um mecanismo de recuperação deve estar em vigor para regenerar o token.
- Sobrecarga de Token: Passar o token entre os sites introduz uma sobrecarga de comunicação, que pode ser significativa em sistemas distribuídos grandes.
- Gerenciamento da Fila de Solicitações: Cada site precisa manter uma fila de solicitações e garantir que a fila seja atualizada corretamente à medida que novas solicitações chegam e o token é passado.

EXCLUSÃO MÚTUA

Exclusão Mútua Não Baseada em Token

1. Comunicação entre Sites: Um site se comunica com outros sites para determinar qual site deve executar a seção crítica a seguir. Isso requer a troca de duas ou mais rodadas sucessivas de mensagens entre os sites.
2. Timestamps em vez de Números de Sequência: Esta abordagem usa timestamps em vez de números de sequência para ordenar as solicitações para a seção crítica.
3. Solicitação de Seção Crítica com Timestamp: Sempre que um site faz uma solicitação para a seção crítica, ele recebe um timestamp. O timestamp também é usado para resolver qualquer conflito entre solicitações de seção crítica.
4. Relógio Lógico: Todos os algoritmos que seguem a abordagem não baseada em token mantêm um relógio lógico. Os relógios lógicos são atualizados de acordo com o esquema de Lamport.

EXCLUSÃO MÚTUA

Exclusão Mútua Não Baseada em Token

- Benefícios
 - Sem Necessidade de Token: A abordagem elimina a necessidade de um token único, evitando problemas relacionados à perda do token ou à sobrecarga de comunicação associada à passagem do token.
 - Resolução de Conflitos com Timestamps: O uso de timestamps garante uma ordenação precisa das solicitações, ajudando a resolver conflitos de maneira eficiente.

EXCLUSÃO MÚTUA

Exclusão Mútua Não Baseada em Token

Desafios

- Troca de Mensagens: A necessidade de múltiplas rodadas de troca de mensagens entre os sites pode introduzir latência, especialmente em sistemas grandes e distribuídos.
- Gerenciamento de Relógios Lógicos: Manter e atualizar os relógios lógicos de acordo com o esquema de Lamport pode ser complexo, especialmente em ambientes de rede não confiáveis ou sujeitos a atrasos.

EXCLUSÃO MÚTUA

- Desejável que algoritmo de exclusão mútua ofereça
 - Baixo overhead de mensagens
 - Não possuir gargalos (ponto único de falha)
 - Tolerar mensagens fora de ordem
 - Tolerar entrada e saída de processos
 - Tolerar falha de processos
 - Tolerar perda de mensagens
 - Sem deadlocks
 - Sem starvation

EXCLUSÃO MÚTUA - ALGORITMO CENTRALIZADO

- Ideias para um algoritmo centralizado?
 - Coordenador: processo responsável por coordenar acesso a região crítica
 - comunicação com mensagens
 - utiliza fila para armazenar pedidos
 - Processos: solicitam entrada na região crítica ao coordenador;
 - liberam região crítica avisando coordenador

EXCLUSÃO MÚTUA - ALGORITMO CENTRALIZADO

- Simula o que é feito em um sistema monoprocessador:
 - Um processo é eleito como coordenador
 - Sempre que um processo quiser acessar determinado recurso, é necessário pedir permissão ao coordenador, através de uma mensagem
 - O coordenador permite acesso ao recurso através de uma mensagem de concessão, desde que nenhum outro processo esteja acessando o recurso naquele momento
 - Para usar uma região crítica: REQUISIÇÃO, CONCESSÃO e LIBERAÇÃO

EXCLUSÃO MÚTUA - ALGORITMO CENTRALIZADO

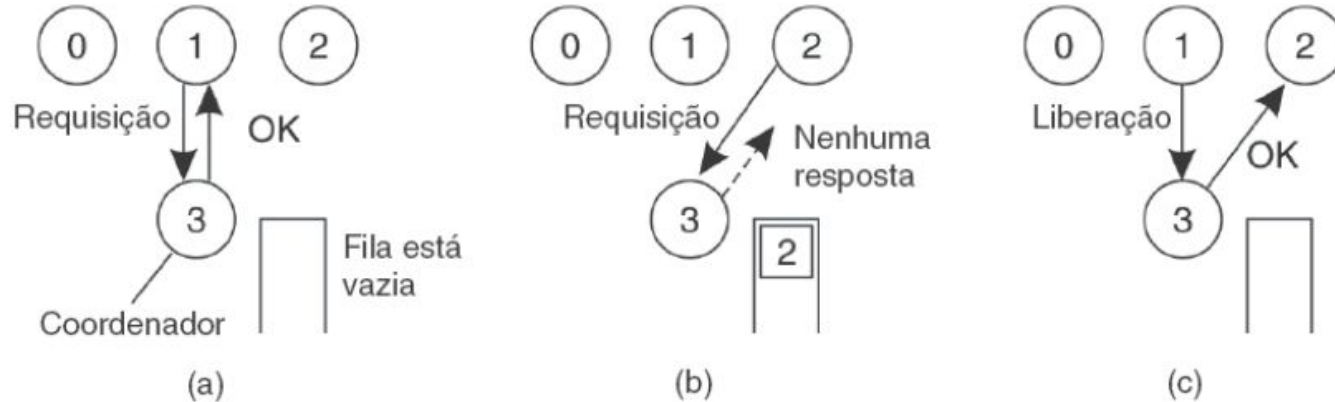


Figura 6.14 (a) O processo 1 solicita ao coordenador permissão para acessar um recurso compartilhado. A permissão é concedida. (b) Depois, o processo 2 solicita permissão para acessar o mesmo recurso. O coordenador não responde. (c) Quando o processo 1 libera o recurso, informa ao coordenador, que então responde a 2.

EXCLUSÃO MÚTUA - ALGORITMO CENTRALIZADO

- Prós:
 - Simples, fácil de entender e de implementar
 - É justo (segue o FCFS)
 - Como no FCFS, não há inanição (starvation)

- Contras:
 - Como todo sistema centralizado um ponto de erro e um ponto de gargalo
 - Processos não conseguem distinguir coordenador inativo de permissão negada, caso se o recurso não está disponível nenhuma mensagem é enviada

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

- Pesquisadores procuraram algoritmos distribuídos determinísticos de exclusão mútua.
- Lamport apresentou o primeiro em 1978
- Ricart e Agrawala o tornaram mais eficiente em 1981

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

- Requer ordenação total de todos os eventos no sistema
 - Para isso será usado... Algoritmo de Lamport!
- Funcionamento:
 - Quando processo deseja acessar um recurso compartilhado, monta uma mensagem que contém o nome do recurso, seu número de processo e a hora corrente (lógica).
 - Envia mensagem para todos processos, inclusive ele mesmo

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

- Funcionamento (continuação):
 - Quando um processo recebe uma mensagem de requisição de outro, executa uma ação de acordo com seu próprio estado em relação ao recurso:
 - 1. Se o receptor não estiver acessando o recurso nem quer acessá-lo, devolve “OK” ao remetente
 - 2. Se já tem acesso ao recurso, não responde e coloca requisição em uma fila
 - 3. Se receptor também quer acessar o recurso, mas ainda não possui a permissão, compara a marca de tempo da mensagem que chegou com a marca de tempo da mensagem que enviou a todos. Se a que recebeu é mais baixa envia OK, senão coloca requisição na fila e não manda nada

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

- Funcionamento (continuação):
 - Após enviar requisições que peçam permissão, um processo aguarda recebimento de todas as respostas
 - Quando houver permissão de todos, processo acessa o recurso
 - Processo libera o recurso enviando um “OK” a todos os processos que estão em sua fila

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

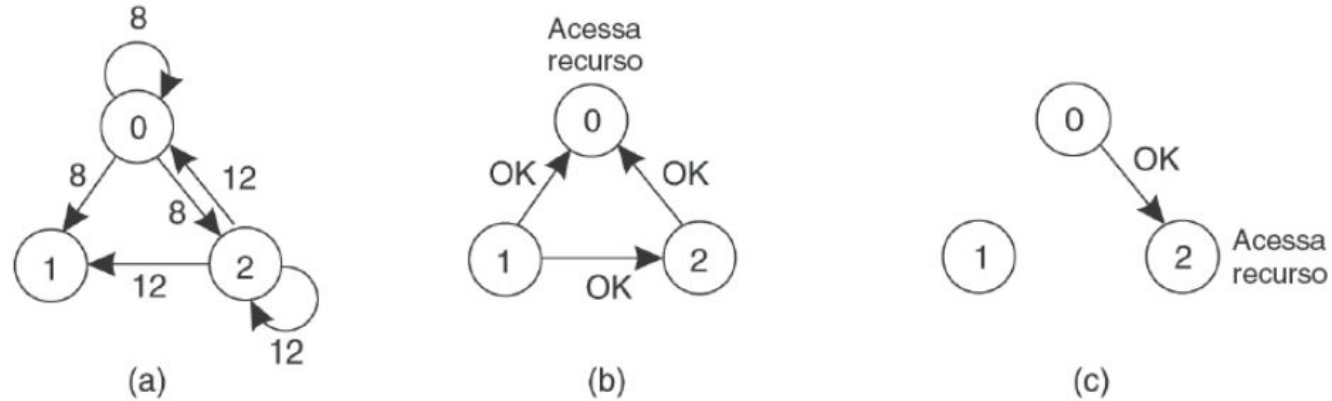


Figura 6.15 (a) Dois processos querem acessar um recurso compartilhado no mesmo momento.
 (b) O processo 0 tem a marca de tempo mais baixa, portanto vence. (c) Quando o processo 0
 conclui, também envia uma mensagem OK, portanto, agora, 2 pode seguir adiante.

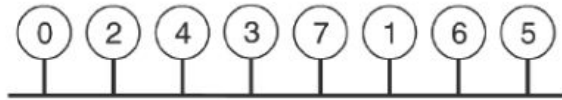
P0 envia requisição com marcador de hora igual a 8 e P2 com marcador igual a 12. P1 envia OK para todos. P0 e P2 veem o conflito e comparam Marcadores. P2 perdeu e envia OK para todos. P0 coloca pedido de P2 na Fila e acessa o recurso. Quando acaba envia OK para P2.

EXCLUSÃO MÚTUA - ALGORITMO DISTRIBUÍDO

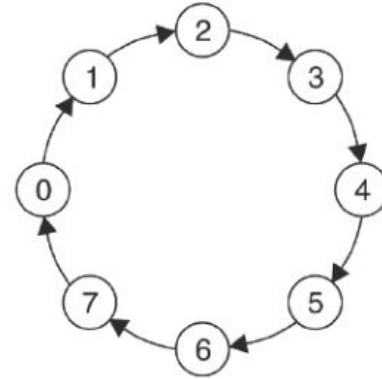
- Vantagens:
 - Exclusão mútua é garantida
 - Não há deadlock nem starvation
- Desvantagens:
 - Ponto de falha único foi substituído por n pontos de falha
 - Trocou 1 gargalo por n gargalos
 - Pode melhorar enviando sempre uma resposta.
 - Se não chegar resposta quem pediu pode concluir que o nó morreu
 - Tem que ter uma comunicação de grupo e gerenciamento

EXCLUSÃO MÚTUA – ALGORITMO TOKEN RING

- Algoritmo baseado em ficha que circula ao redor do anel lógico
- Se o processo quer acessar recurso e tem a ficha, usa recurso e quando acaba passa a ficha
- Se não quer usar recurso, passa a ficha



(a)



(b)

EXCLUSÃO MÚTUA – ALGORITMO TOKEN RING

- **Corretude:** token está em apenas em um processo em cada instante
- **Justiça:** acesso garantido antes de um mesmo processo acessar RC novamente (token circula)
- Limitações se um processo falhar? se o token se perder?

EXCLUSÃO MÚTUA – ALGORITMO TOKEN RING

- Garante a exclusão mútua
- Não há deadlock nem starvation
- A dificuldade é recuperar uma ficha perdida
- Como saber o tempo razoável para um token não aparecer?
- Se um processo morreu, um vizinho que tenta enviar o token para ele pode detectar, e o anel pode ser reconfigurado

EXCLUSÃO MÚTUA – ALGORITMO TOKEN RING

Tabela Comparativa dos Algoritmos de Exclusão Mútua

Algoritmo	Mensagens por Entrada/Saída	Problemas
Centralizado	3	Queda do coordenador
Descentralizado	$3mk$	Starvation, baixa eficiência
Distribuído	$2(n-1)$	Queda de qualquer processo
Token Ring	$1 a \infty$	Ficha perdida