

Aula 17 - Algoritmo de Ricart-Agrawala

Monday, May 9, 2016

13:41

Relembrando o algoritmo de Lamport

- Processos mantêm um relógio lógico global
- Canais de comunicação FIFO
- Cada processo i mantém uma *request queue* RQ_i com as requisições ordenadas pelo relógio lógico

Algoritmo de Ricarta-Agrawala

Requisição da RC

- Quando o processo i deseja entrar na RC, envia um REQUEST com timestamp para todos os outros processos
- Quando o processo j recebe o REQUEST do processo i , **apenas** envia um REPLY se (não está na RC) e (não tem requisição dele mesmo (j) com maior prioridade - timestamp menor)
- Caso j não envie REPLY para i , faz $pendentes_j[i] < -1$

Execução da PC

- Após receber reply de todos os processos, o processo i pode entrar na PC

Liberação da RC

- Após sair da RC, o processo i envia todos os REPLIES em $pendentes_i[]$, ou seja:
 - Se $pendentes_i[j] = 1$, então i envia REPLY para j .

Comparação

O algoritmo de Lamport exige

- Canais de comunicação FIFO unicast
- Mantém fila de requisições
- Usa mensagens de RELEASE

O algoritmo de RiCart-Agrawala não tem nada disso! Mas

- Mantém relógio lógico global
- Vetor de REPLIES pendentes

A ideia básica é simples: um processo só executa a RC se tiver recebido REPLIES, QUE SÃO PERMISSÕES de todos os demais

- A permissão só é enviada quando o processo não tem requisição com prioridade maior
 - o Definida pelo **relógio lógico** (processos mantêm um relógio lógico global)
- Cada processo mantém um vetor local pendentes, que indica quais processos fizeram requisições de menor prioridade, para os quais é necessário enviar replies.

Prova de correção

Teorema

O algoritmo Ricart-Agrawala garante a exclusão mútua

Prova por absurdo

Considere que o algoritmo permite que 2 processos i & j acessem a RC concorrentemente.

Sem perda de generalidade, considere que, usando o relógio lógico, a requisição de p_i tem maior prioridade que a requisição de p_j .

Nesse caso, a requisição do j só pode ter chegado em i depois deste ter feito sua própria requisição.

Entretanto, para j entrar na RC deve ter recebido REPLY do i . Impossível! Pois i só enviará o REPLY depois de executar e sair da RC.

Desempenho

mensagens para acessar a RC:

$$2(N-1)$$