EXCLUSÃO MÚTUA

DCE540 - Computação Paralela e Distribuída

Atualizado em: 10 de setembro de 2021

Iago Carvalho

Departamento de Ciência da Computação

ACESSO A RECURSOS COMPARTILHADOS

Em um sistema distribuído, existe um número de aplicações rodando simultaneamente

- Duas ou mais aplicações podem querer acessar um mesmo recurso simultaneamente

Este acesso simultâneo (concorrente) pode

- corromper o recurso
- gerar dados inconsistentes

Assim, é necessário criar maneiras de coordenar o acesso a recursos distribuídos

PROBLEMAS DE ACESSO COMPARTILHADO

Starvation: Um processo tenta continuamente acessar a um recurso compartilhado, mas nunca consegue acesso

- Muito comum quando processos de alta prioridade estão acessando o recurso
- Processo fica bloqueado até que o recurso seja liberado

Deadlock: Um processo A necessita de um recurso que está sendo acessado por outro processo B. Ao mesmo tempo, o processo B necessita de um recurso que está sendo acessado pelo processo A.

- Ambos os processos permanecem bloqueados
- Não é possível desfazer o *deadlock* sem mecanismos específicos

ALGORITMOS PARA COORDENAÇÃO DE ACESSO

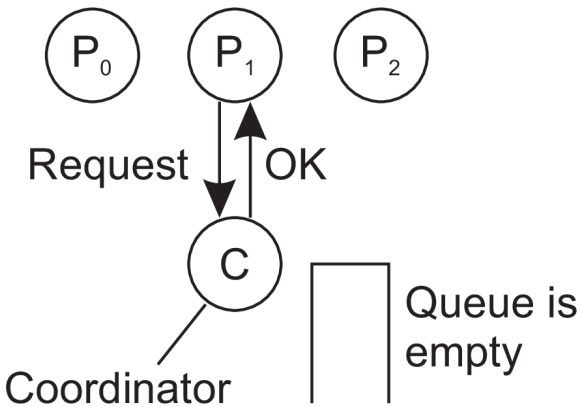
O *middleware* pode (e deve!) implementar algum tipo de algoritmo para coordenar o acesso a recursos compartilhados

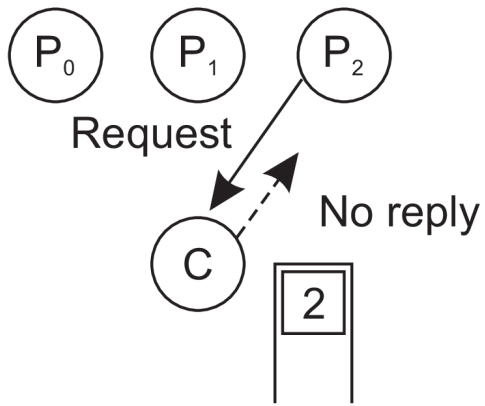
Existem diversos algoritmos diferentes para realizar esta coordenação

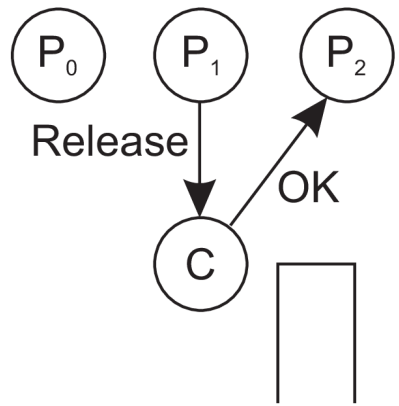
Todos os algoritmos devem garantir que

- Não ocorra deadlocks
- Não ocorra starvation
- Equidade e justiça
  - Todo processo deve ter chances claras (e parecidas) para acessar o recurso compartilhado

ALGORITMO CENTRALIZADO







ALGORITMO DISTRIBUÍDO

Baseado na ideia de *clocks* lógicos, como discutidos na aula anterior

- Baseia-se na ideia de ter uma ordem clara entre todos os eventos no sistema distribuído
- Estabelece uma ordem lógica sobre a ordem de acesso aos recursos compartilhados

Quando um processo quer acessar um recurso, ele envia uma mensagem para todos os processos do sistema distribuído (includindo ele mesmo)

- Nome do recurso compartilhado, número do processo, tempo (lógico)
- Assume-se que não existam erros na rede e todos os outros processos leiam a mensagem

ALGORITMO DISTRIBUÍDO

Quando um processo recebe uma mensagem de outro, ele pode tomar 3 diferentes ações

Caso ele não esteja acessando o recurso e nem queira acessar

- Responde *OK*

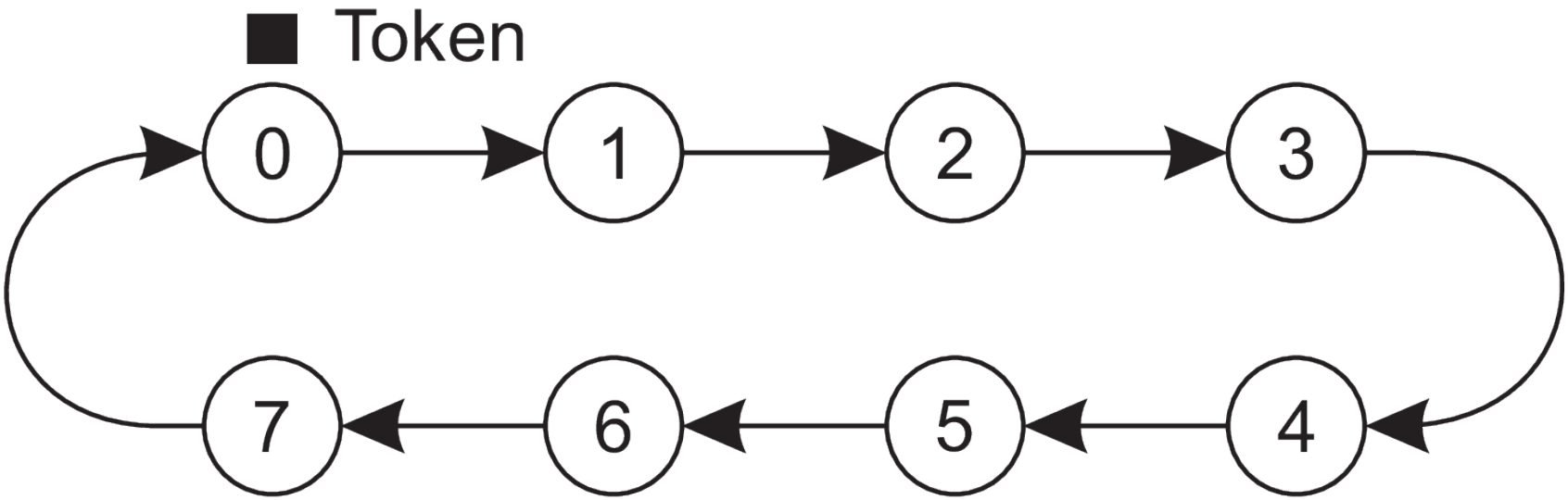
Caso ele esteja acessando o recurso

- Ele enfileira a mensagem e não responde nada
- Assim que ele liberar o recurso, ele responde OK

Caso ele também queira acessar o recurso

- O que possuir a mensagem com a menor data, vence

ALGORITMO *TOKEN-RING*



COMPARAÇÃO ENTRE OS ALGORITMOS

Algoritmo	Mensagens por entrada/saída	Delay
Centralizado	3	2
Distribuído	$3(N-1)$	$2(N-1)$
Token-ring	$1, \ldots, \infty$	$0, \ldots, N-1$