Sistemas Distribuídos

Rui Oliveira
Departamento de Informática
Universidade do Minho

Algoritmos Distribuídos

(Apontamentos baseados no livro Distributed Systems: Principles and Paradigms, A. Tanenbaum e M. Van Steen)

Programa Detalhado

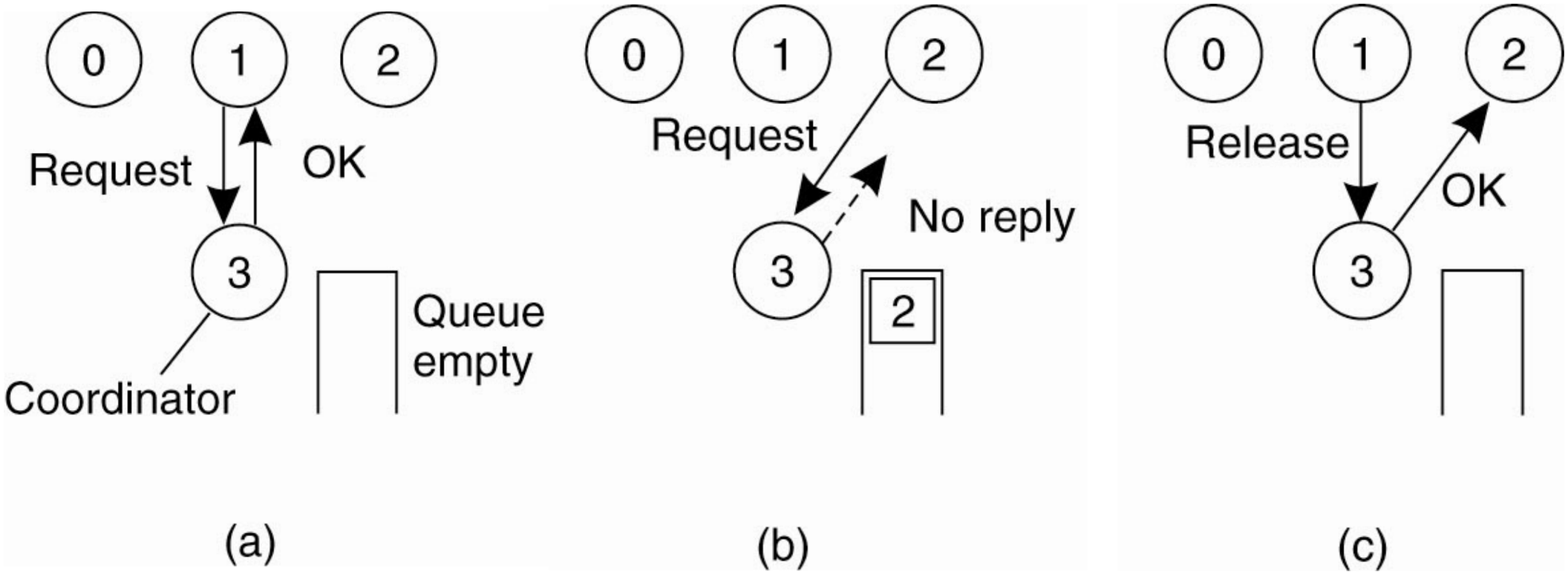
- Algoritmos distribuídos
  - Exclusão mútua
  - Eleição

- Assumpções
  - O sistema consiste de n processos; cada processo no seu processador.
  - Cada processo tem uma zona crítica que requer exclusão mútua.
- Requisitos
  - Se um processo se encontra a executar a sua zona crítica, então mais nenhum processo se encontra a executar a sua.

Exclusão mútua distribuída: alg centralizado

- Um processo é escolhido para coordenar o acesso à zona crítica.
- Um processo que queira executar a sua zona crítica envia um pedido ao coordenador.
- O coordenador decide que processo pode entrar na zona crítica e envia a esse processo uma resposta.
- Quando recebe a resposta do coordenador, o processo inicia a execução da sua zona crítica.
- Quando termina a execução da sua zona crítica, o processo envia uma mensagem a libertar a zona

Exclusão mútua distribuída: alg centralizado



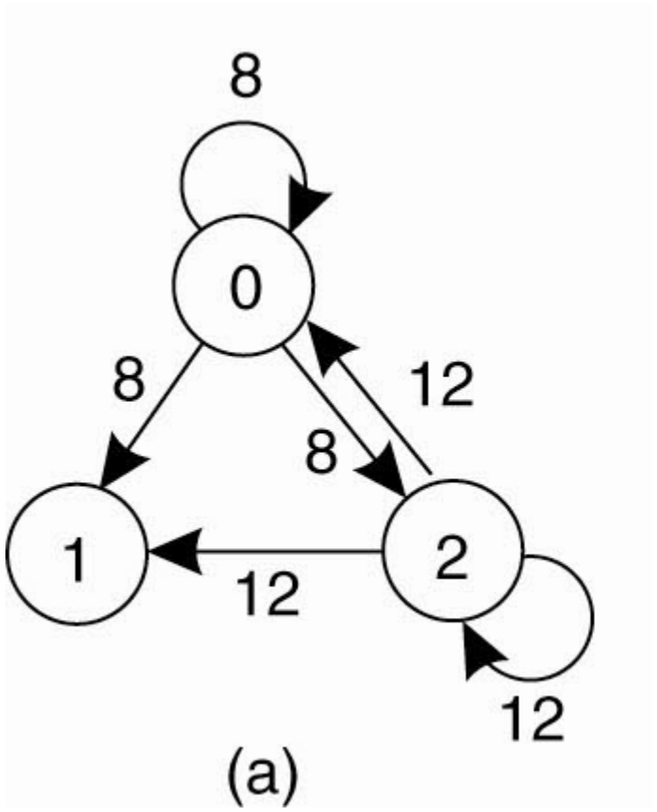
Exclusão mútua distribuída: algoritmo descentralizado

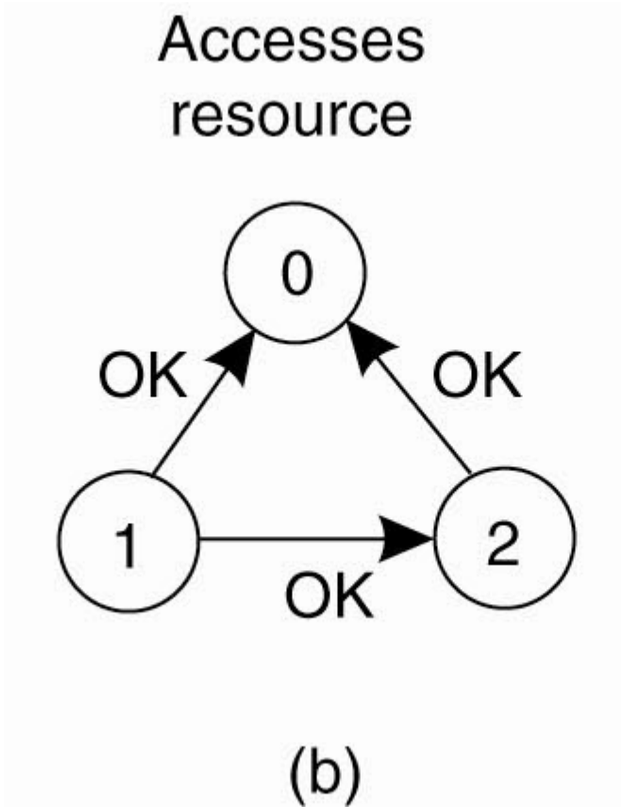
- Quando um processo Pi pretende aceder à sua zona crítica gera uma etiqueta temporal TS e envia um pedido (Pi, Tsi) a todos os processos.
- Quando um processo recebe um pedido pode responder logo ou adiar a sua resposta.
- Quando um processo recebe respostas de **todos** os processos no sistema pode então executar a sua zona crítica.
- Depois de terminar a execução da sua zona crítica, o processo responde a todos os pedidos aos quais adiou a resposta.

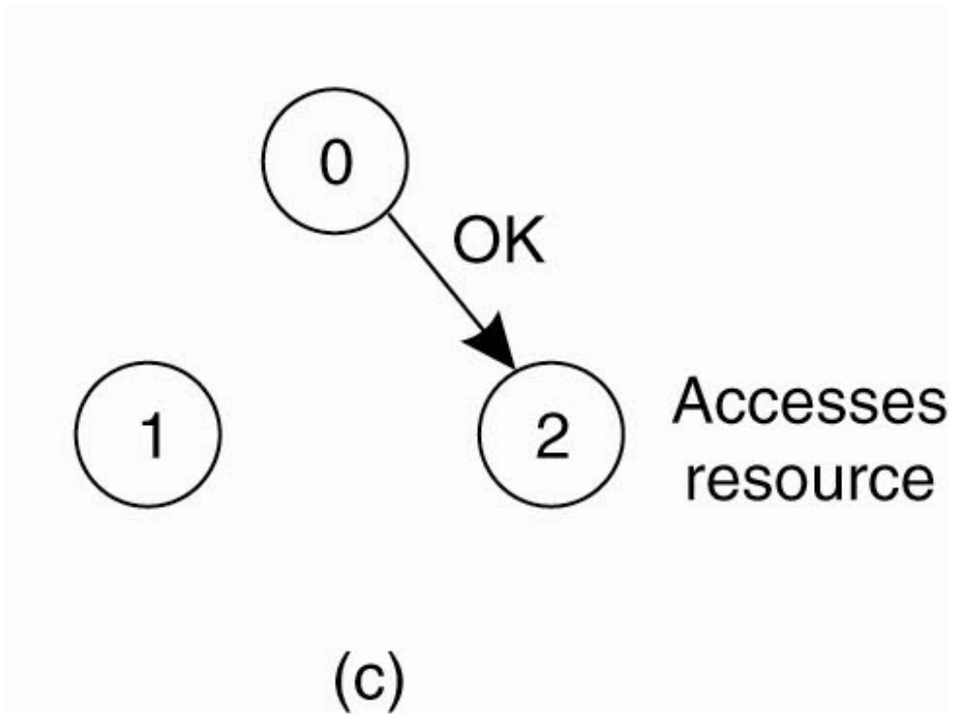
Exclusão mútua distribuída: algoritmo descentralizado

- A decisão de Pj responder logo a um pedido (Pi, TSi) ou adiar depende de três factores:
  - Se Pj estiver na sua zona crítica, adia.
  - Se Pj não pretender aceder à sua zona crítica, responde.
  - Se Pj pretender aceder à sua zona crítica (e já enviou também um pedido) então compara a etiqueta temporal do seu pedido TSj com TSi:
    - Se TSj > TSi então responde logo (Pi pediu primeiro)
    - Senão adia a resposta.

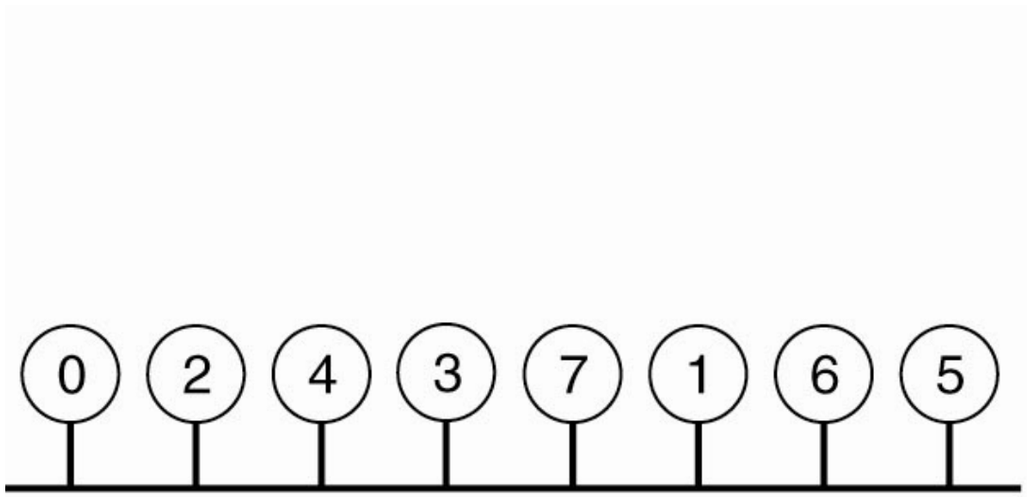
Exclusão mútua distribuída: algoritmo descentralizado



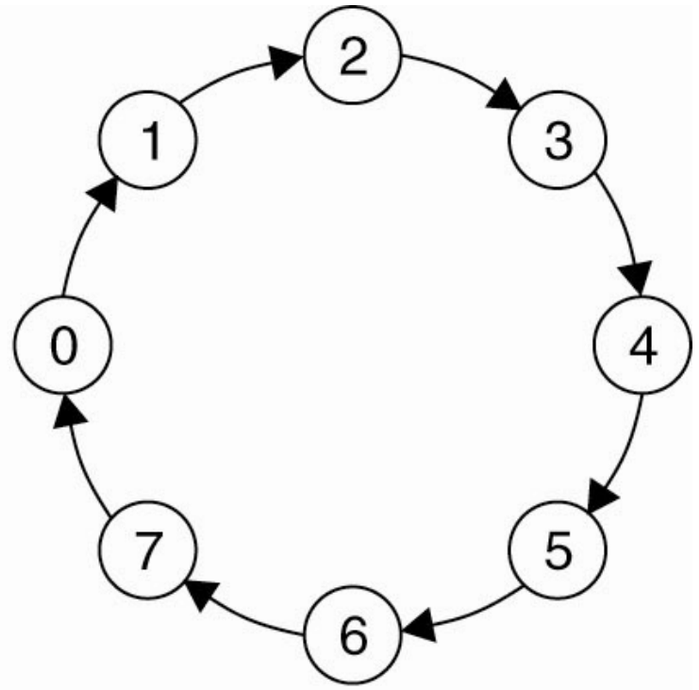




Exclusão mútua distribuída: algoritmo descentralizado em anel



(a)



(b)

Exclusão mútua distribuída: comparação dos 3 algoritmos

Algoritmo	#Mensagens por entrada/saída	#Mínimo de mensagens para entrar	Problemas
Centralizado	3	2	Falha do Coordenador
Descentralizado	2(n-1)	2(n-1)	Falha de qualquer processo
Anel	1 a Infinito	0 a n-1	Perda do Token Falha de qualquer processo

Eleição

- Determinar quando e onde um novo coordenador é necessário.
- Assuma-se que cada processo tem uma prioridade única e distinta.
- O coordenador é sempre o processo com mais alta prioridade.

- Algoritmo Bully
  - O algoritmo é despoletado por um processo Pi que julga que o coordenador falhou.
  - Pi envia uma mensagem de eleição a todos os processos com maior prioridade que Pi.
  - Se num intervalo T Pi não receber qq resposta, então auto-elege-se coordenador.
  - Se receber alguma resposta então Pi espera durante T' por uma mensagem do novo coordenador.
  - Se não receber nenhuma mensagem então reinicia o algoritmo.
  - Se Pi não é o coordenador então, a qualquer momento pode receber uma mensagem...

- Algoritmo em anel
  - Aplicável a sistemas organizados fisica or logicamente em anel
  - Assume que os canais de comunicação são unidireccionais
  - Cada processo mantem uma lista de activos que consiste nas prioridades de todos os processos activos quando este algoritmo terminar.
  - Quando um processo Pi suspeita a falha do coordenador, Pi cria uma lista de activos vazia, envia uma mensagem de eleição m(i) ao seu vizinho e insere i na lista de activos.

- Algoritmo em anel
  - Se Pi recebe uma mensagem de eleição m(j) responde de uma de três formas:
    - Se foi a primeira mensagem de eleição que viu, então cria uma lista de activos com i e j. Envia as mensagens de eleição m(i) e m(j) (nesta ordem) ao vizinho.
    - Se i≠j então junta j à sua lista de activos e reenvia a mensagem ao vizinho.
    - Se i=j então a sua lista de activos já contem todos os processos activos no sistema e Pi pode determinar o coordenador.