



Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Bacharelado em Ciência da Computação

Sistemas Distribuídos

Coordenação e Acordo

Prof. Rodrigo Campiolo

19/10/20

Tópicos

- Introdução
- Exclusão mútua distribuída
- Eleições
- Problemas de acordo
- Atividades

Introdução

- Algoritmos que possibilitem um conjunto de processos coordenarem suas ações ou concordarem com um ou mais valores.
- Evitar relações hierárquicas únicas para evitar pontos de falhas.
- Manter a consistência quando um servidor precisa assumir o controle.

Exclusão Mútua Distribuída

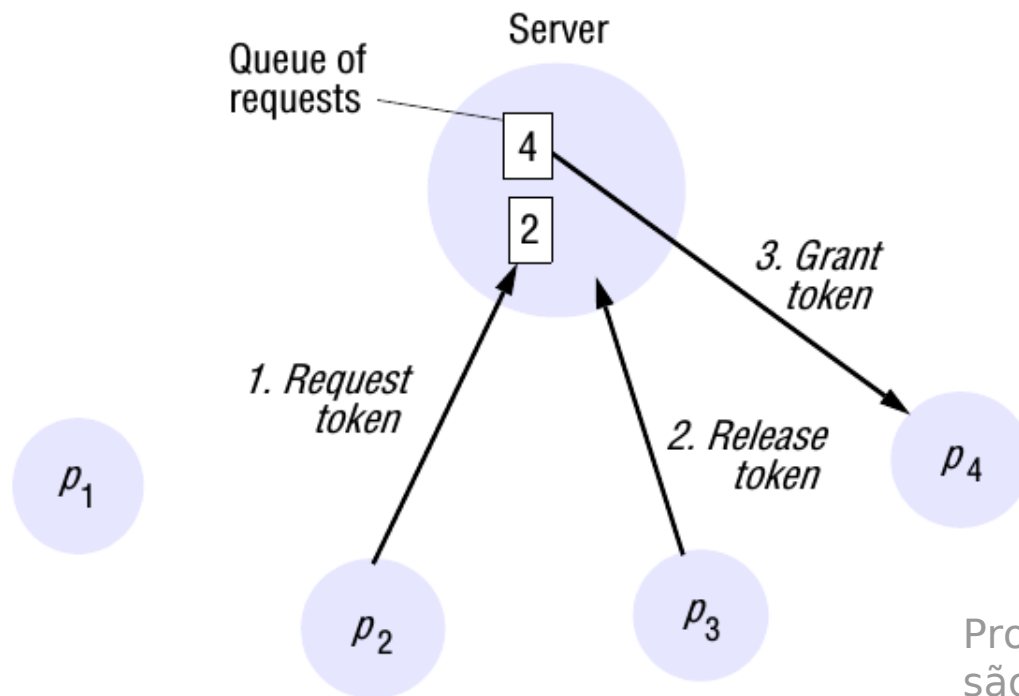
- Recursos compartilhados entre múltiplos processos.
- Uso de exclusão mútua.
- Em um SD, variáveis compartilhadas não podem ser usadas.
- A **exclusão mútua distribuída** deve ocorrer somente pela **troca de mensagens**.
- Exemplo: Estacionamento (vagas ou não vagas)

Algoritmos de Exclusão Mútua

- Definições
 - **N** processos p_i , $i = 1, 2, 3, \dots, N$
 - Processos acessam recursos compartilhados por meio de uma seção crítica.
- Requisitos
 - **EM1 (segurança)**: máximo um processo por vez na seção crítica.
 - **EM2 (subsistência)**: os pedidos para entrar e sair da seção crítica têm sucesso.
 - **EM3 (ordenação)**: se um pedido de entrar na seção crítica aconteceu antes de outro, então a ordem de entrada é garantida nessa ordem.

Algoritmos de Exclusão Mútua

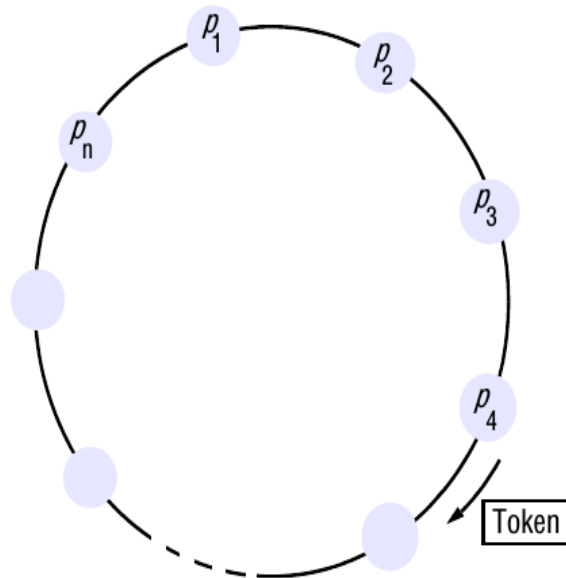
- Algoritmo do servidor central
 - Servidor que gerencia a entrada na seção crítica por meio da concessão de *tokens*.



Propriedades EM1 e EM2
são satisfeitas.

Algoritmos de Exclusão Mútua

- Algoritmo baseado no anel
 - Uso de um anel lógico formado pelos processos. O *token* é passado em um único sentido. Se o processo estiver de posse do *token* pode acessar a seção crítica.



Fonte: Coulouris

Propriedades EM1 e EM2
são satisfeitas.

Algoritmos de Exclusão Mútua

- Algoritmo de Ricart e Agrawala (1981)
 - Um processo que deseja entrar na seção crítica faz um **multicast** com uma mensagem de pedido e só pode entrar após a resposta de todos os outros processos.
 - Cada processo mantém um **relógio lógico**.
 - Variável de estado de cada processo:
 - RELEASED: fora da seção crítica.
 - WANTED: deseja entrar na seção crítica.
 - HELD: está na seção crítica.

Algoritmos de Exclusão Mútua

▪ Algoritmo de Ricart e Agrawala

On initialization

state := RELEASED;

To enter the section

state := WANTED;

Multicast *request* to all processes;

T := request's timestamp;

Wait until (number of replies received = (*N* − 1));

state := HELD;

} *Request processing deferred here*

On receipt of a request $\langle T_i, p_i \rangle$ *at* p_j ($i \neq j$)

if (*state* = HELD or (*state* = WANTED and $(T, p_j) < (T_i, p_i)$))

then

 queue *request* from p_i without replying;

else

 reply immediately to p_i ;

end if

To exit the critical section

state := RELEASED;

reply to any queued requests;

Algoritmos de Exclusão Mútua

- Algoritmo de votação de Maekawa (1985)
 - Processos só precisam obter permissão de uma parte de seus pares para acessar a seção crítica, desde que os subconjuntos de dois processos se sobreponham.
 - Processos votam um no outro para entrar na seção crítica.
 - Um processo deve reunir votos suficientes para entrar.
 - Algoritmo pode resultar em impasse (**deadlock**).

Eleições

- Algoritmo para eleger um único processo para desempenhar uma função específica.
- Exemplo: selecionar o servidor entre um conjunto de processos.
- Todos os processos devem concordar.
- O processo eleito deve ser aquele com o maior **identificador**.
- O identificador pode ser definido por um valor constante ou variável.

Eleições

- Todo processo p_i ($i=1,2,\dots,N$) tem uma variável **elected_i**.
- Requisitos:
 - **E1 (segurança)**: p_i tem $\text{elected}_i = \perp$ ou $\text{elected}_i = P$ onde P é escolhido com o processo não defeituoso com maior identificador.
 - **E2 (subsistência)**: todos os processos p_i participam e configuram $\text{elected}_i \neq \perp$

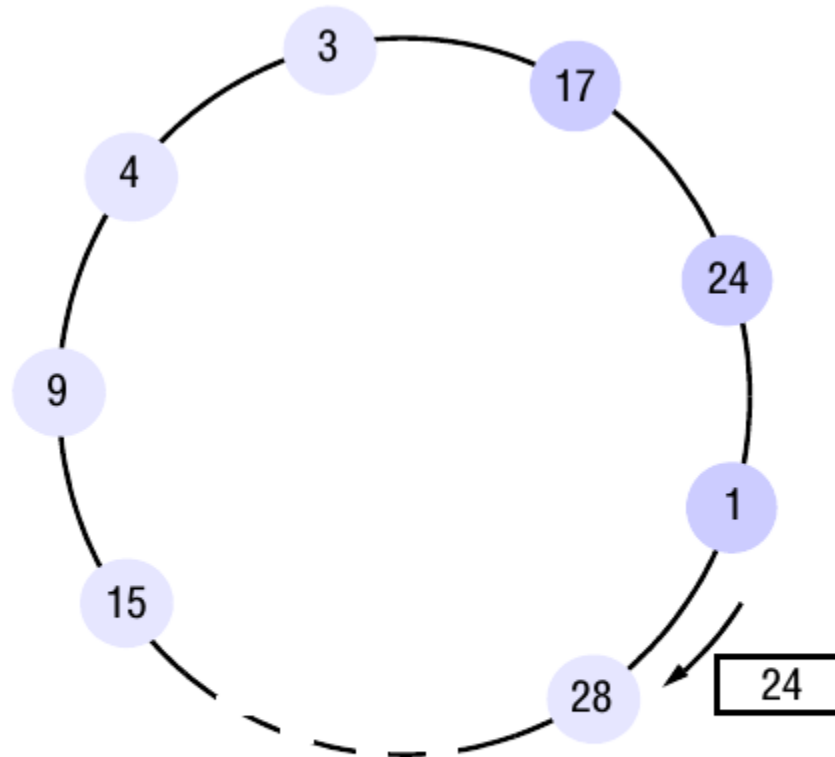
Eleições

▪ Algoritmo de eleição baseado em anel

- coordenador é o processo com maior identificador.
- inicialmente todos são **não participantes**.
- processo começa a eleição, cria uma mensagem com **seu identificador**, marca a si próprio como **participante** e encaminha mensagem para o próximo processo.
- se o processo que recebeu tem identificador menor, encaminha para o próximo e marca como participante.
- se o processo que recebeu **tem identificador maior** e é **não participante**, substitui o identificador, marca como participante e encaminha a mensagem.
- se o processo que recebeu verifica que é seu **próprio identificador**, marca-se como **não participante** e anuncia a **eleição** via mensagem *elected* com seu identificador.
- todos os processos que são participantes e recebem a mensagem *elected*, marcam a si próprios como não participantes, configuram a *variável elected* para o identificador recebido.

Eleições

- Algoritmo de eleição baseado em anel



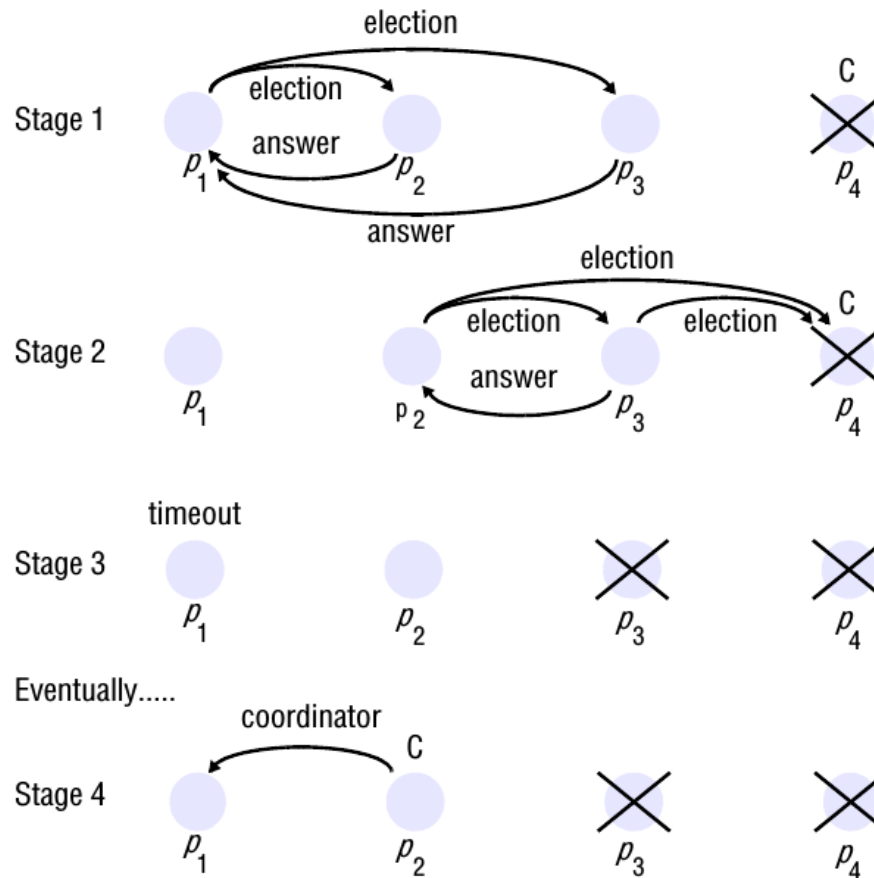
Note: The election was started by process 17. The highest process identifier encountered so far is 24. Participant processes are shown in a darker tint. Fonte: Coulouris

Eleições

- Algoritmo valentão (bully)
 - Suporta a falha de um processo durante a eleição.
 - Supõe que o sistema é síncrono para detectar falhas de um processo.
 - Supõe que os processos sabem quem tem os identificadores mais altos.
 - Mensagens:
 - `election`: solicitação de eleição.
 - `answer`: resposta a solicitação de eleição.
 - `coordinator`: anúncio de coordenador.
 - Ao receber uma mensagem `election` um processo inicia uma eleição, exceto se já tenha começado.

Eleições

- Algoritmo valentão



The election of coordinator p_2 , after the failure of p_4 and then p_3 Fonte: Coulouris

Problemas de Acordo

- O problema está relacionado ao fato de os processos concordarem com um valor, após um ou mais dos processos terem proposto qual deve ser esse valor.
- Garantir o acordo na presença de falhas.
- Problemas relacionados:
 - Problema de consenso.
 - Problemas relacionados aos generais bizantinos.
 - Problemas relacionados a consistência interativa.

Atividades

- Fazer uma pesquisa sobre os problemas: consenso, generais bizantinos e consistência interativa.

Referências

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.