

Exclusão mútua distribuída

Paulo Augusto Gomes Kataki

paulogkataki@hotmail.com

Instituto de Informática
Universidade Federal de Goiás

13 de Julho de 2019

Exclusão mútua distribuída

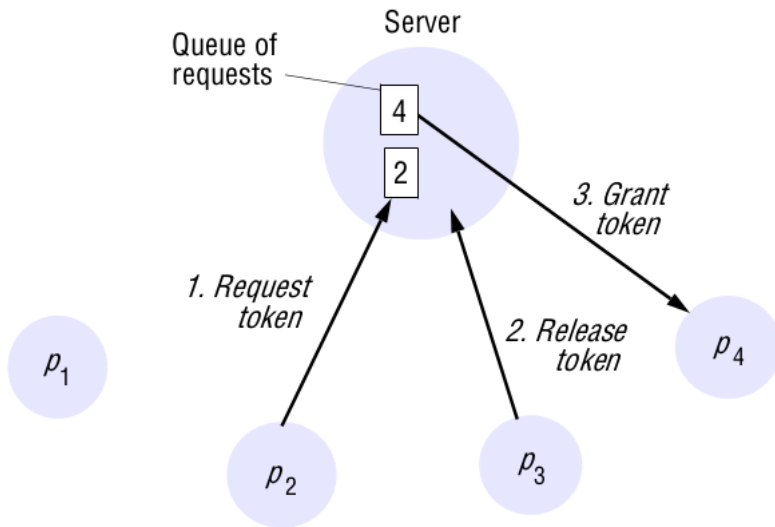
Requisitos essenciais para exclusão mútua

- 1 No máximo um processo pode acessar a região crítica por vez;
- 2 Pedidos para entrar e sair da região crítica eventualmente são sucedidos;
- 3 A ordem cronológica dos pedidos de entrada na região crítica deve ser mantida;

Algoritmo do servidor central

Algoritmo do servidor central

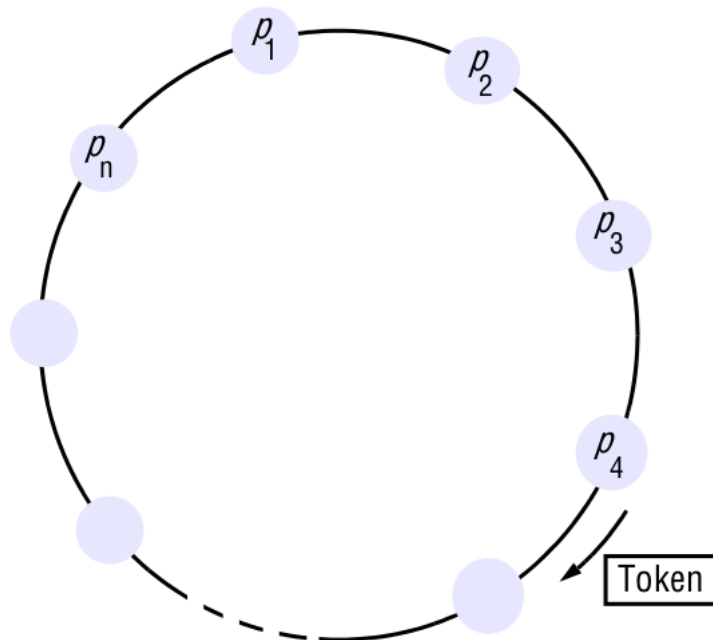
- 1 Forma mais simples de obter a exclusão mútua distribuída;
- 2 Ter um servidor central que controla a exclusão mútua;
- 3 Servidor utiliza um token para controlar o acesso a região crítica;
- 4 Há uma fila no servidor central para armazenar as requisições;



Algoritmo do anel

Algoritmo do anel

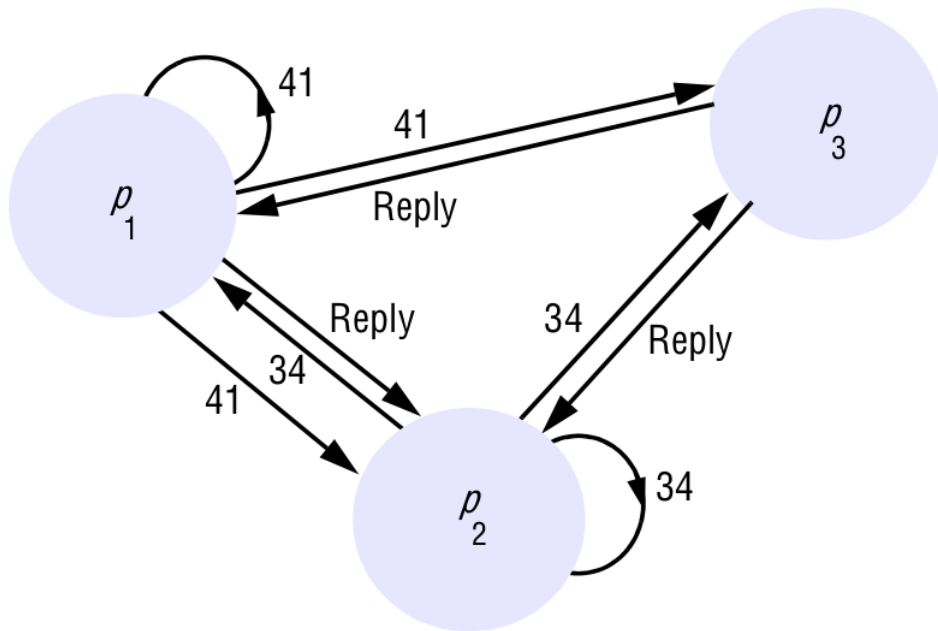
- 1 Organiza os processos em forma de anel;
- 2 Cada processo comunica somente com o próximo processo, ou seja, p_i comunica com $p_{(i+1) \bmod N}$.
- 3 Utiliza a ideia de token;
- 4 O processo que tiver o token pode entrar na região crítica;
- 5 Assim que sair da região crítica o processo passa o token para o próximo processo.



Algoritmo de Ricart e Agrawala

Algoritmo de Ricart e Agrawala

- Utiliza somente troca de mensagem entre os processos;
- Utiliza multicast e relógios lógicos de Lamport;
- Garante todas as 3 restrições;
- Cada processo necessita de $2(N - 1)$ mensagens para entrar na seção crítica;
- Tempo de sincronização dos processos é somente na transmissão das mensagens.



Ricart and Agrawala's algorithm

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;

Exclusão mútua distribuída

Paulo Augusto Gomes Kataki

paulogkataki@hotmail.com

Instituto de Informática
Universidade Federal de Goiás

13 de Julho de 2019