
ALGORITMOS DISTRIBUÍDOS

Algoritmos de eleição

Algoritmos Distribuídos (eleição)

- Em sistemas distribuídos, diversos algoritmos necessitam que um processo funcione como coordenador, inicializador, sequenciador, enfim, ter um papel especial
- exemplos:
 - coordenador de exclusão mútua com controle centralizado
 - coordenador para detecção de deadlock distribuído
 - seqüenciador de eventos para ordenação consistente centralizada
 - etc.
- falha do coordenador compromete serviço para vários processos
- novo coordenador deve assumir - **eleição!**
 - **Objetivo: eleger *um* processo, entre os ativos, para desempenhar função especial**

Algoritmos Distribuídos (eleição)

- Existem algoritmos especiais para escolha de um processo que assumirá o papel de coordenador
- Veremos dois *algoritmos de eleição*:
 - algoritmo do valentão (*bully algorithm*)
 - algoritmo em anel (*ring algorithm*)

Algoritmos Distribuídos (eleição)

- Para ambos os algoritmos, assume-se que:
(*general assumptions*)
 - 1) Todo processo no sistema tem uma *prioridade única*
 - 2) Quando eleição acontece, o processo com maior prioridade *entre os processos ativos* é eleito como coordenador
 - 3) Na recuperação (volta à atividade), um processo falho pode tomar ações para juntar-se ao grupo de processos ativos

Algoritmos Distribuídos (eleição)

Bully algorithm

- Desenvolvido por Garcia-Molina em 1982
- Assume que um processo sabe a prioridade de todos outros processos no sistema

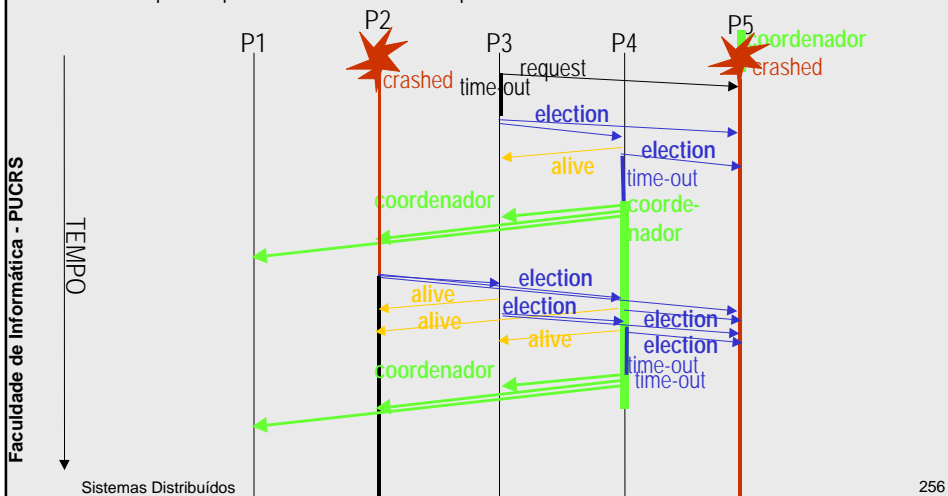
Algoritmo

- Quando um processo P_i detecta que o coordenador não está respondendo a um pedido de serviço, ele inicia uma eleição da seguinte forma:
 - P_i envia uma msg ELEIÇÃO para todos processos com prioridade maior que a sua
 - se nenhum processo responde
 - P_i vence a eleição e torna-se coordenador
 - // significa que não há processos ativos com maior prioridade que a sua
 - P_i manda uma msg COORDENADOR para os processos de menor prioridade informando que é o coordenador, deste momento em diante, ●
 - se processo P_j com prioridade maior que P_i responde (msg ALIVE)
 - P_i não faz mais nada, P_j assume o controle
 - P_j age como P_i nos passos a) e b)

Algoritmos Distribuídos (eleição)

Bully algorithm - exemplo [Pradeep, pg 334]

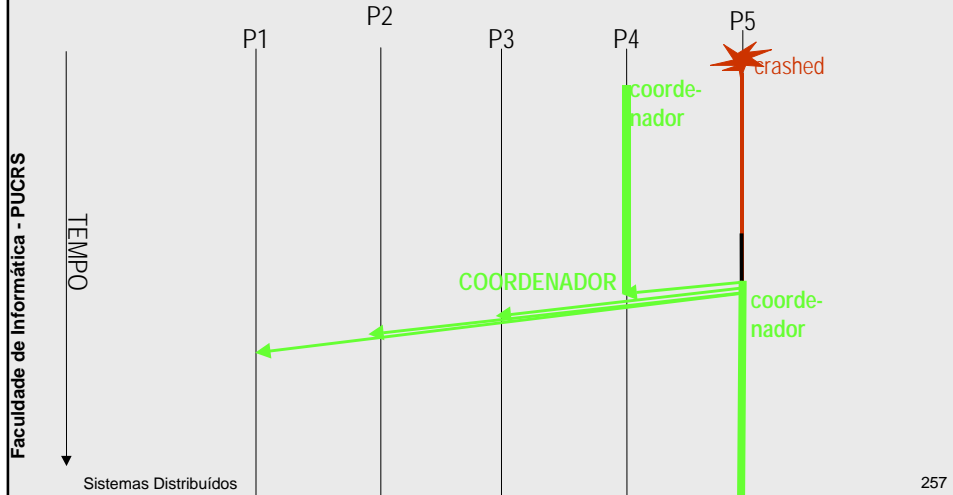
- suponha processos P1 a P5 com prioridade conforme seu identificador



Algoritmos Distribuídos (eleição)

Bully algorithm - exemplo [Pradeep, pg 334]

- suponha processos P1 a P5 com prioridade conforme seu identificador



Algoritmos Distribuídos (eleição)

Ring algorithm

- Baseado no uso de um anel lógico, sem uso de *token*
- Cada processo conhece o anel inteiro, mas manda mensagens somente para o próximo processo ativo na direção do anel
- Quando processo detecta que o coord. não está ativo, ele constrói uma msg ELEIÇÃO contendo seu id., e manda para o próximo do anel
- A cada passo, o processo que recebe a msg inclui seu id na msg e envia para o próximo do anel
- No final o processo que iniciou a eleição recebe a msg e escolhe aquele que tem maior id.
- Nova msg é enviada novamente através do anel para todos contendo o novo coordenador
- Uma vez que a msg passou por todos processos e chegou no originador, ela é retirada do sistema

Algoritmos Distribuídos (eleição)

■ Análise

- algoritmo do valentão (bully)
 - se processo com prioridade mais baixa detecta a falha do coordenador, em um sistema com n processos, então $n-1$ eleições acontecem (pradeep diz $n-2$)
 - cada eleição tem mensagens conforme número de processos - $O(n^2)$ mensagens - *no pior caso*
 - se processo que detecta falha é o ativo de maior prioridade precisa só de $n-2$ mensagens - *melhor caso*
- algoritmo do anel
 - eleição sempre precisa de $2(n-1)$ mensagens
 - $n-1$ mensagens para rotação da mensagem de eleição
 - $n-1$ mensagens para rotação da mensagem de coordenador