Sistemas Distribuídos
Aula 15

Roteiro

- Coordenação dinâmica
- Eleição de líder
- Algoritmo do valentão
- Algoritmo em anel

Coordenação

- Muitos sistemas (e algoritmos) distribuídos necessitam de um (único) coordenador
  - algoritmo de Berkeley para sincronizar relógios
  - exclusão mútua centralizada
  - *tracker* no Bittorrent
- Em muitos casos, coordenador é determinado *a priori* e fixo
  - ponto único de falha, inflexibilidade

- Como construir sistemas distribuídos mais robustos que tenham um coordenador?

Coordenação Dinâmica

- **Ideia**: determinar coordenador dinamicamente
  - escolha faz parte do sistema distribuído
  - arquitetura com coordenador deixa de ser totalmente centralizada
  - papel do coordenador pode ser realizado por outros processos do sistema

- Como determinar o coordenador?
  - eleição de líder (coordenador)
  - algoritmo distribuído que detemina um único líder (coordenador) para o sistema
  - requer participação de todos processos

Eleição de Líder

- **Ideia**: escolher um único processo no sistema para ser o líder (coordenador)
  - todos os outros processos precisam concordar
- Assumir que os processos podem detectar a ausência (falha) do atual líder
- Assumir que todos processos possuem identificador único
- **Ideia**: Líder é o processo que possui o maior identificador

  - Mas como determinar o líder?

Eleição de Líder – Versão 1

- Algum processo detecta ausência do líder
  - envia mensagem de eleição com seu ID para todos
- Um processo ao receber a mensagem de eleição pela primeira vez ou detectar ausência do líder
  - envia mensagem de eleição com seu ID para todos
- Ao receber mensagem de eleição de todos
  - processo com maior ID envia mensagem de coordenador para todos
- **Problema:** Sistema com n processos demanda $n^2 + n$ mensagens para eleição

Podemos ser mais eficientes?

Algoritmo do Valentão (*Bully Algorithm*)

- Proposto em 1982 por Garcia-Molina
  - assumir rede confiável e FIFO
  - três tipos de mensagem: *eleição, OK, coordenador*
- Processo P ao detectar falha do líder inicia eleição (visando substituir o líder)

  1) Envia mensagem eleição com seu ID para todos os processos

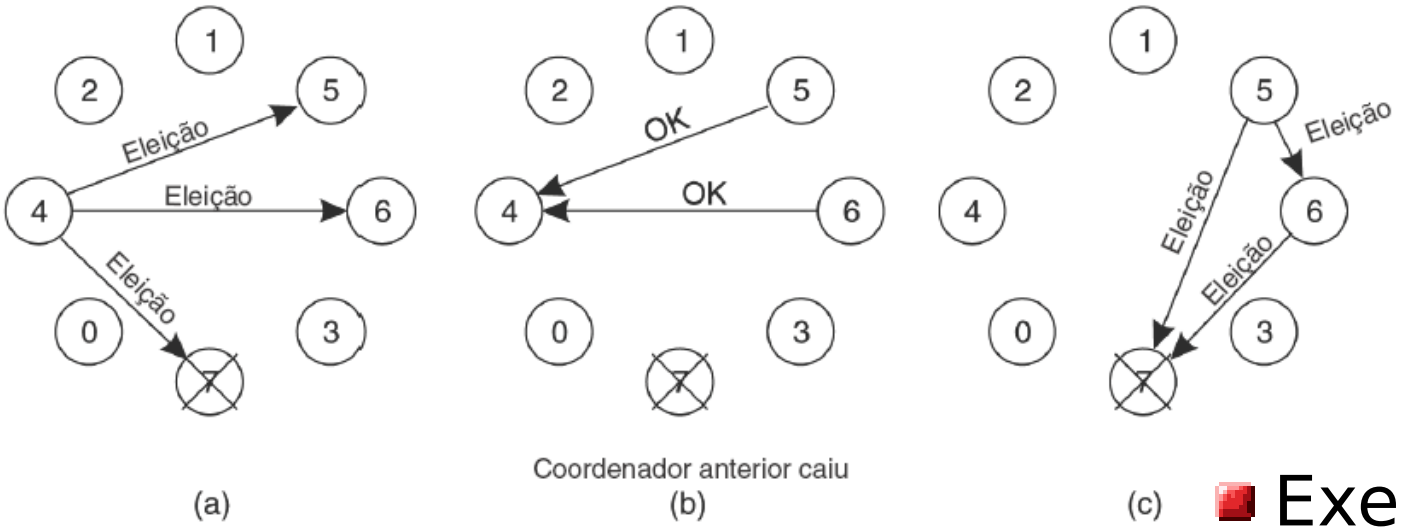
  2) Se não receber nenhuma resposta, P envia mensagem *coordenador* para todos os processos (se tornando o coordenador)

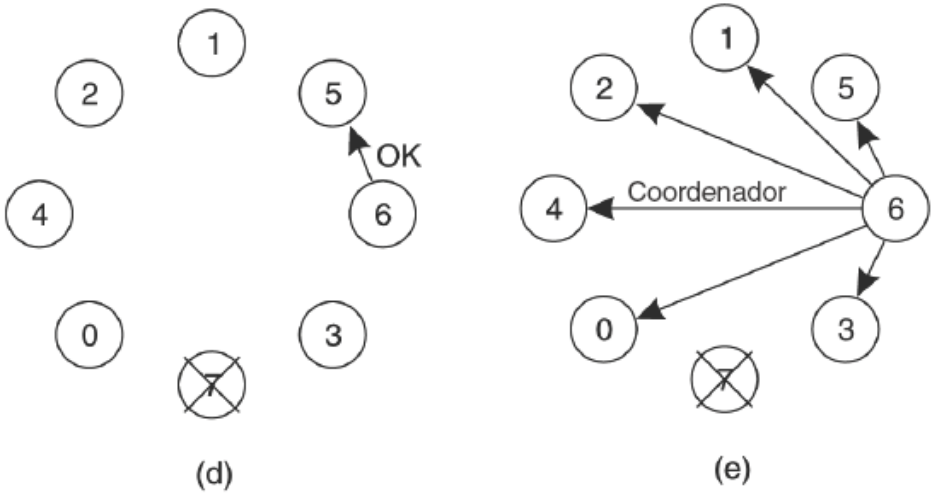
  3) Se receber mensagem *OK* de algum processo, P desiste de ser coordenador

Algoritmo do Valentão (*Bully Algorithm*)

- Processo Q que recebe mensagem de *eleição*
  - 1) Se ID local for menor, não faz nada
  - 2) Se ID local for maior
  - 2.1) Envia mensagem *OK* para P
  - 2.2) Inicia uma eleição (faz o papel de P)
- Eleição eventualmente termina, pois algum processo possui maior ID
  - neste caso todos desistem ou não respondem
  - processo com maior ID vence eleição e se torna o coordenador (*valentão*)

Exemplo do Valentão





- Exemplo ilustrativo não mostra todas as mensagens enviadas!
- Mensagem de eleição é enviada para todos
- Processo 4 inicia eleição (detectou primeiro falha do lider)

Complexidade do Valentão

- Quantas mensagems são enviadas?
- Depende de quem inicia a eleição
- Pior caso: menor ID inicia eleição
  - $\Theta(n^2)$ mensagens
- Melhor caso: maior ID inicia eleição
  - $\Theta(n)$ mensagens

- Como tentar evitar o pior caso?
- Retardar início de nova eleição em processos com ID pequenos
  - após detectar falha do líder, ou após receber ID ainda menor

Eleição em Anel

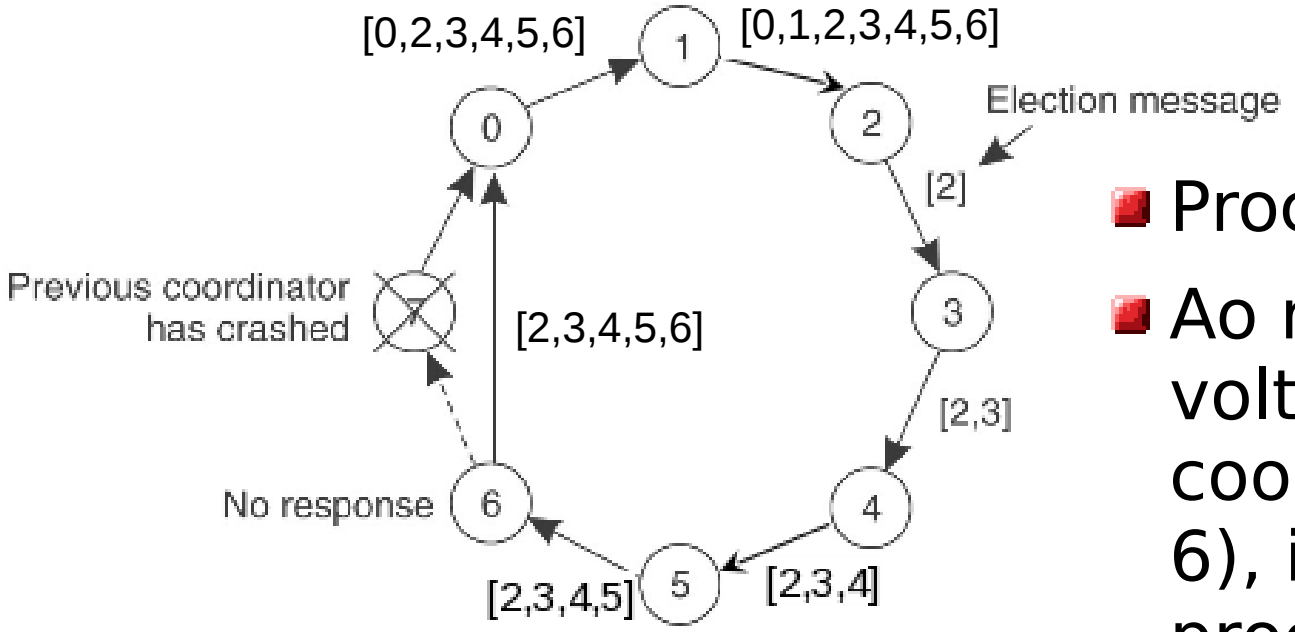
- Organizar processos em rede formando um anel lógico (já vimos isto antes!)
  - cada processo conectado a sucessor e predecessor
  - anel lógico é para fins de eleição (não é a arquitetura do sistema distribuído)
- Dois tipos de mensagem: *eleição*, *coordenador*
- Processo P que detecta falha do coordenador inicia uma eleição

  1) envia mensagem *eleição* com seu ID ao próximo processo operacional no anel

  2) ao receber sua mensagem que circulou o anel, determina o coordenador (maior ID recebido), envia mensagem *coordenador* pelo anel

Eleição em Anel

- Processo Q que recebe mensagem *eleição*
  1) adiciona seu ID na mensagem
  2) envia mensagem ao próximo processo operacional no anel
- Exemplo



- Processo 2 inicia eleição
- Ao receber mensagem de volta, determina o coordenador (nó com ID 6), informa a todos os processos

Complexidade

- Depende de quantos processos iniciam a eleição
- Melhor caso: apenas um processo inicia
  - 2 voltas no anel (eleição + coordenador) = 2n mensagens
- Pior caso: todos os processos iniciam
  - n vezes o caso anterior = $2n^2$ mensagens
- Ideias para evitar o pior caso
  - não iniciar uma eleição ao participar de uma; atrasar o início de uma nova eleição
- Tamanho da mensagem de eleição não é constante
  - cresce a cada salto no anel, com total de n identificadores ao final

Propriedades

- O que acontece se dois ou mais processos iniciam uma eleição?
  - eleições rodam de forma independente
  - produzem o mesmo resultado
- O que acontece se a rede não é FIFO
  - troca de ordem não afeta corretude do algoritmo
- O que acontece se processo que iniciou a eleição falhar?
  - eventualmente uma nova eleição vai ser iniciada, pois coordenador não será substituído
- O que acontece se processo que ganhou a eleição falhar?

Robustez do algoritmo!