

# Algoritmos de Eleição

Time 2

2023

# Sumário

- ▶ Introdução
- ▶ O que são algoritmos de eleição?
- ▶ Tipos de algoritmos de eleição
- ▶ Vantagens e Desvantagens
- ▶ Exemplos em prática
- ▶ Quando usar cada um dos tipos existentes





# Introdução



# Introdução

- Algoritmos de eleição são usados em sistemas distribuídos para selecionar um líder ou coordenador entre vários processos, favorecendo a ordem coordenação e tomada de decisões em ambientes descentralizados.
- Falaremos sobre as vantagens, o que são, tipos existentes e quando usar cada um deles e exemplos na prática.



# O que são Algoritmos de Eleição?

# O que são Algoritmos de Eleição?

- É um conjunto de regras computacionais usadas para selecionar um ou mais participantes de um grupo para posições específicas, como líderes ou representantes, de maneira justa e eficiente.
- Eles são comuns em sistemas distribuídos e redes, como em clusters de servidores ou dispositivos interconectados, onde a escolha de líderes ou coordenadores é necessária para garantir a coordenação e o funcionamento adequado do sistema.





# Tipos Existentes



# Tipos Existentes

- Algoritmos Centralizados (Como por exemplo o “Bully”): Um nó central (coordenador) é responsável por coletar informações dos outros nós e tomar a decisão de eleição com base em critérios específicos, como capacidade de processamento, disponibilidade, latência entre outros.
- Algoritmos Distribuídos (Como por exemplo o “Ring”): Todos os nós participam na tomada de decisão de forma colaborativa, trocando informações e utilizando critérios para escolher um líder ou coordenador.





# Vantagens



# Vantagens e Desvantagens

## Vantagens

- Escalabilidade: Permitem a adição ou remoção de nós do sistema sem interromper a operação global ou eleição de novos líderes.
- Balanceamento de carga: Algoritmos de eleição podem ajudar a distribuir a carga de trabalho entre os nós, evitando sobrecargas em um único nó.
- Coordenação: Melhora a eficiência e organização do sistema.
- Eficiência: Algoritmos bem projetados podem reduzir o tráfego de comunicação e minimizar o tempo necessário para eleger um novo líder.

## Desvantagens

- Latência: A eleição leva tempo, causando atraso em tomadas de decisões.
- Complexidade: Difíceis de projetar e manter.
- Segurança: Possível tentativa de falsificação influenciando na seleção do líder.



**Em prática**

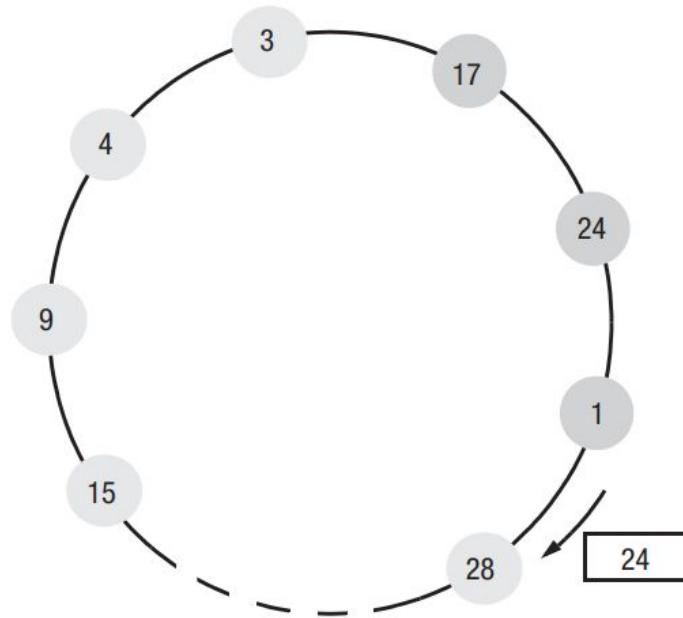


# Ring-based Election Algorithm

- Nesse algoritmo, os processos estão organizados em um anel, onde cada processo pode comunicar com o seguinte.
- Supomos que não ocorrem falhas e que o sistema é assíncrono.
- O objetivo desse algoritmo é eleger um único processo, chamado de coordenador, que é aquele com o maior identificador.



# Ring-based Election Algorithm



*Nota: A eleição foi iniciada pelo processo 17. O identificador de processo mais alto encontrado até aqui é 24. Os processos participantes aparecem com fundo cinza mais escuro.*



# Ring-based Election Algorithm

## Initialization

Participant<sub>*i*</sub> := FALSE;  
Elected<sub>*i*</sub> := NIL

## P<sub>*i*</sub> starts an election

Participant<sub>*i*</sub> := TRUE;  
Send the message <election, p<sub>*i*</sub>> to its neighbor

## Receipt of a message <elected, p<sub>*j*</sub>> at p<sub>*i*</sub>

Participant<sub>*i*</sub> := FALSE;  
If p<sub>*i*</sub> ≠ p<sub>*j*</sub>  
Then Send the message <elected, p<sub>*j*</sub>> to its neighbor



# Ring-based Election Algorithm

Receipt of the election's message  $\langle \text{election}, p_i \rangle$  at  $p_j$

If  $p_i > p_j$

Then Send the message  $\langle \text{election}, p_i \rangle$  to its neighbor  
Participant<sub>j</sub> := TRUE;

Else If  $p_i < p_j$  AND Participant<sub>j</sub> = FALSE

Then Send the message  $\langle \text{election}, p_j \rangle$  to its neighbor  
Participant<sub>j</sub> := TRUE;

Else If  $p_i = p_j$

Then Elected<sub>j</sub> := TRUE;  
Participant<sub>j</sub> := FALSE;  
Send the message  $\langle \text{elected}, p_j \rangle$  to its neighbor

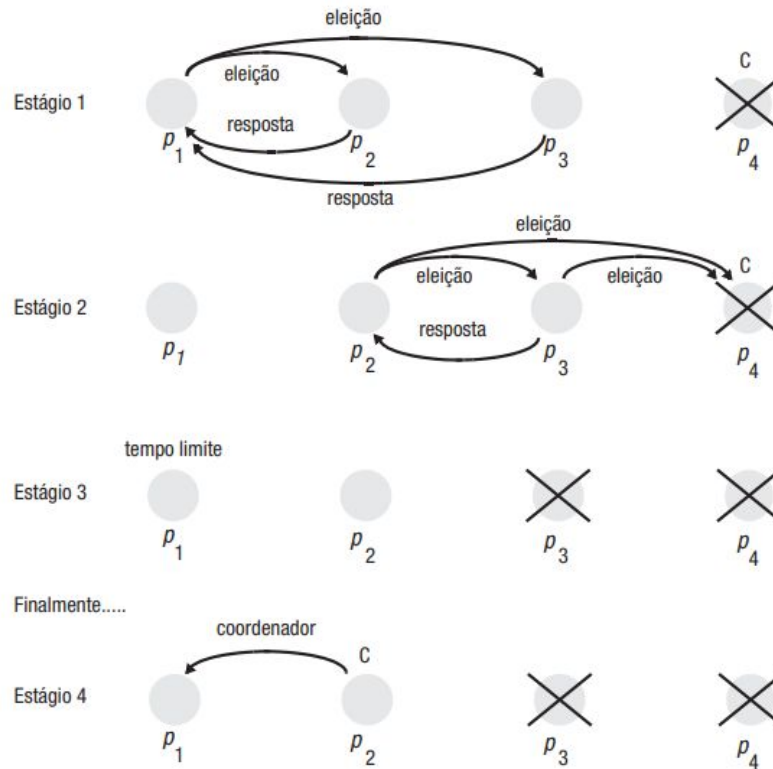


# Bully Algorithm

- Esse algoritmo permite falhas nos processos, embora presuma que a comunicação entre eles é confiável
- Ao contrário do algoritmo baseado em anel, este algoritmo presume que o sistema é síncrono: ele usa tempos limites para detectar uma falha de processo
- O algoritmo valentão presume que cada processo sabe quais processos têm identificadores mais altos e que pode se comunicar com todos esses processos.



# Bully Algorithm



A eleição do coordenador  $p_2$ , após a falha de  $p_4$  e, depois, de  $p_3$ .





# Bully Algorithm

## Initialization

$\text{Elected}_i := \text{NIL}$

## $p_i$ starts the election

Send the message (*Election*,  $p_i$ ) to  $p_j$ , i.e.,  $p_j > p_i$

Waits until all messages (*OK*,  $p_j$ ) from  $p_j$  are received;

If no message (*OK*,  $p_j$ ) arrives during  $T$

Then  $\text{Elected} := p_i$ ;

Send the message (*Coordinator*,  $p_i$ ) to  $p_j$ , i.e.,  $p_j < p_i$

Else waits until receipt of the message (*coordinator*)

(if it doesn't arrive during another timeout  $T'$ , it begins another election)



# Bully Algorithm

Receipt of the message (*Coordinator*,  $p_j$ )

$\text{Elected} := p_j;$

Receipt of the message (*Election*,  $p_j$ ) at  $p_i$

Send the message (*OK*,  $p_i$ ) to  $p_j$

Start the election unless it has begun one already

# Obrigado!

