



Doutorado e Mestrado em Ciência da Computação

Instituto de Informática



Programação Paralela e Distribuída

Prof. Dr. Sérgio T. Carvalho

Algoritmos de Eleição

Marcos Alves Vieira

Goiânia, 07 de maio de 2019

Introdução

- ▶ Algoritmo de eleição
 - ▶ Algoritmo para escolher um único processo para desempenhar uma determinada função
- ▶ Aspectos importantes
 - ▶ Todos os processos devem concordar com a escolha
 - ▶ Se o processo coordenador falhar, outra eleição será realizada para escolher um substituto
 - ▶ Um determinado processo só pode convocar uma única eleição por vez
 - ▶ A escolha do processo eleito deve ser única, mesmo que vários processos convoquem eleições concorrentemente

Introdução

▶ Processo

▶ Identificador

- ▶ Pode ser qualquer valor, desde que os identificadores sejam exclusivos e totalmente ordenados

▶ Eleição

- ▶ Conterá o valor ***participante*** ou ***não participante***, caso o processo esteja ou não esteja participando de uma eleição

▶ Eleito

- ▶ Variável que conterá o identificador do processo coordenador
- ▶ Possui valor especial \perp , quando o processo participa pela primeira vez de uma eleição, indicando que seu valor ainda não está definido

Introdução

- ▶ Requisitos a serem cumpridos pelo algoritmo de eleição, ao final de sua execução
 - ▶ E1 (segurança)
 - ▶ Um processo participante tem eleito = \perp ou eleito = P, onde P é o processo não defeituoso com o maior identificador
 - ▶ E2 (subsistência)
 - ▶ Todos os processos participam e configuram eleito $\neq \perp$ ou falham

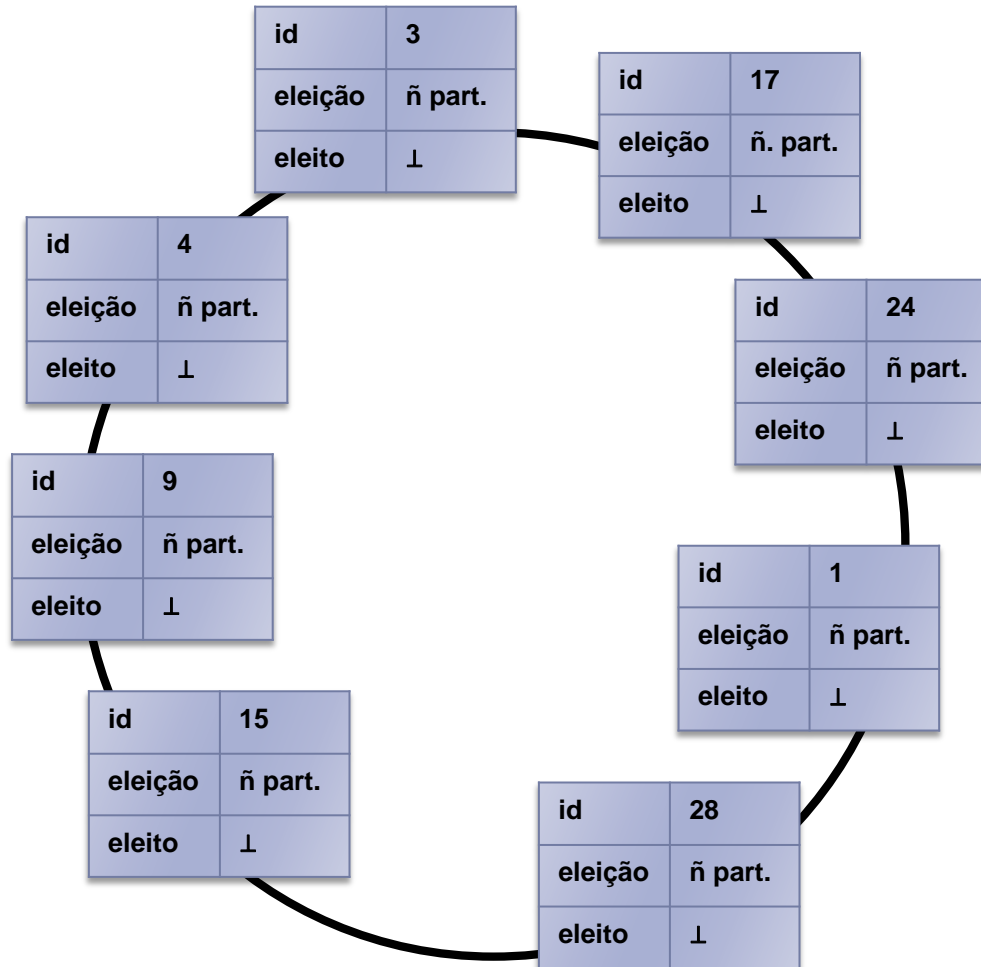
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

- ▶ Conjunto de processos organizados em um anel lógico
 - ▶ Cada processo tem um canal de comunicação para o processo seguinte no anel
 - ▶ Todas as mensagens são enviadas no sentido horário em torno do anel
 - ▶ Supõe-se que não ocorrem falhas e que o sistema é assíncrono
 - ▶ Possui dois tipos de mensagem: eleição e eleito
- ▶ Objetivo
 - ▶ Eleger um único processo, chamado de coordenador, que é aquele com o maior identificador

Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

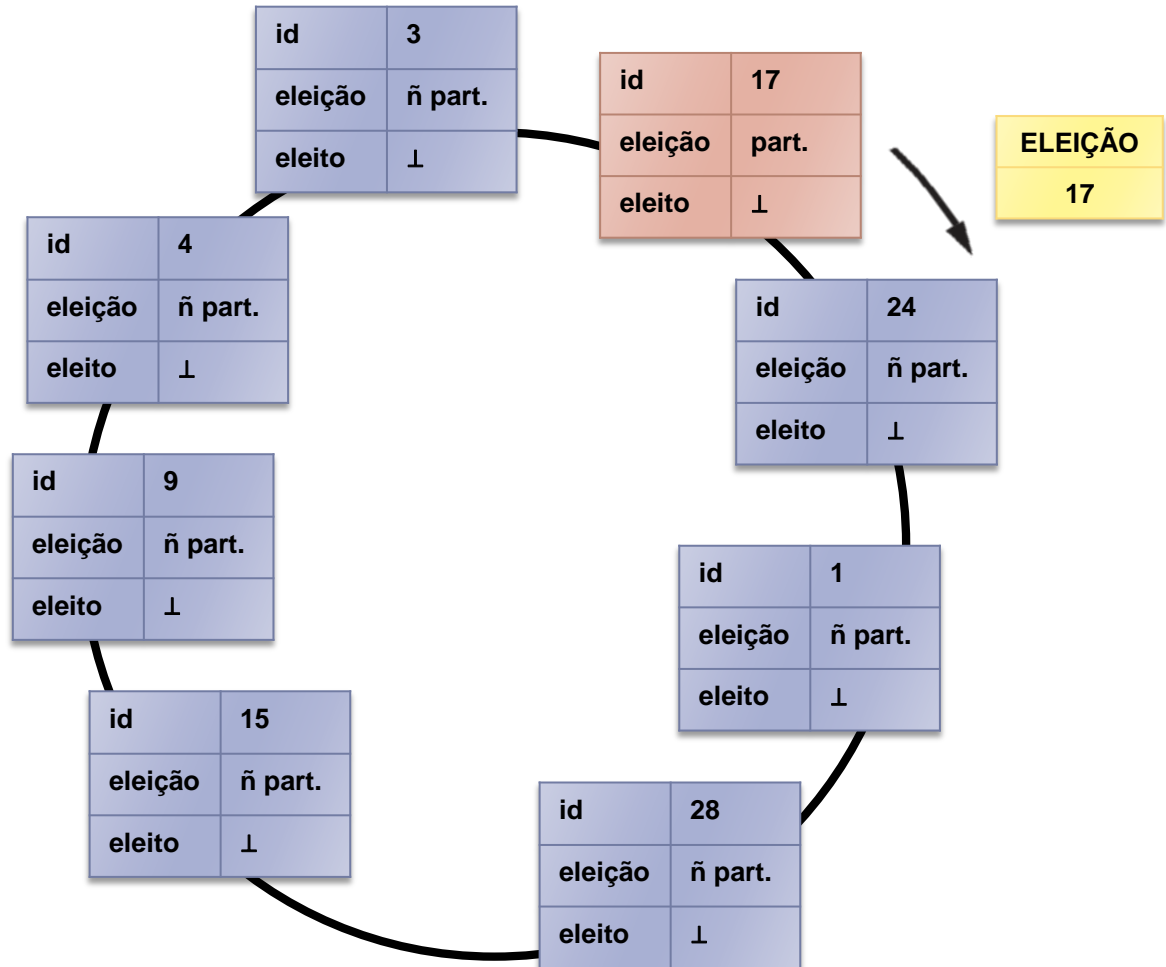


Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Iniciar uma eleição

1. O processo marca a si mesmo como **participante**
2. Coloca **seu identificador** em uma **mensagem de eleição**
3. Envia a mensagem para seu vizinho no sentido horário



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

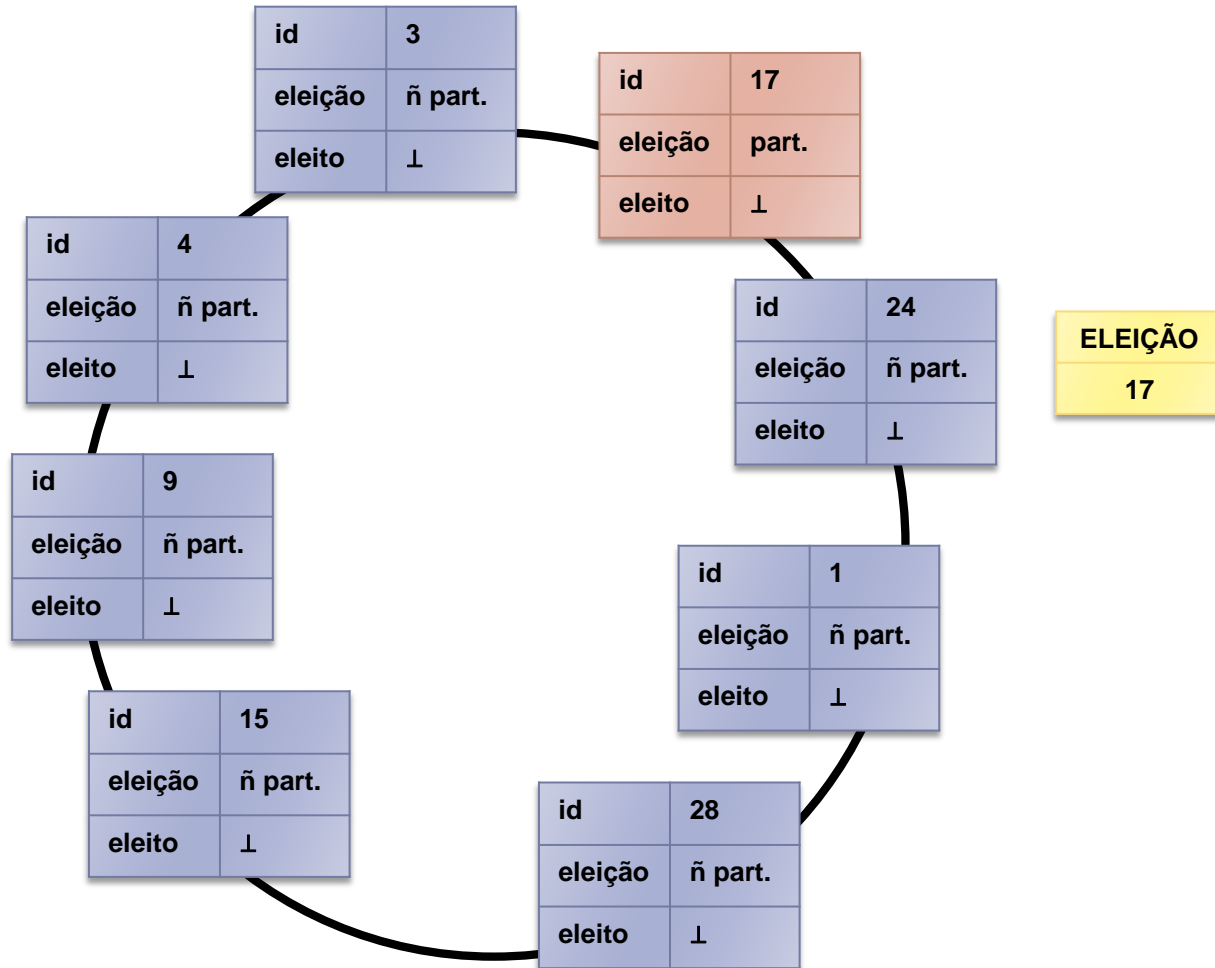
Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho

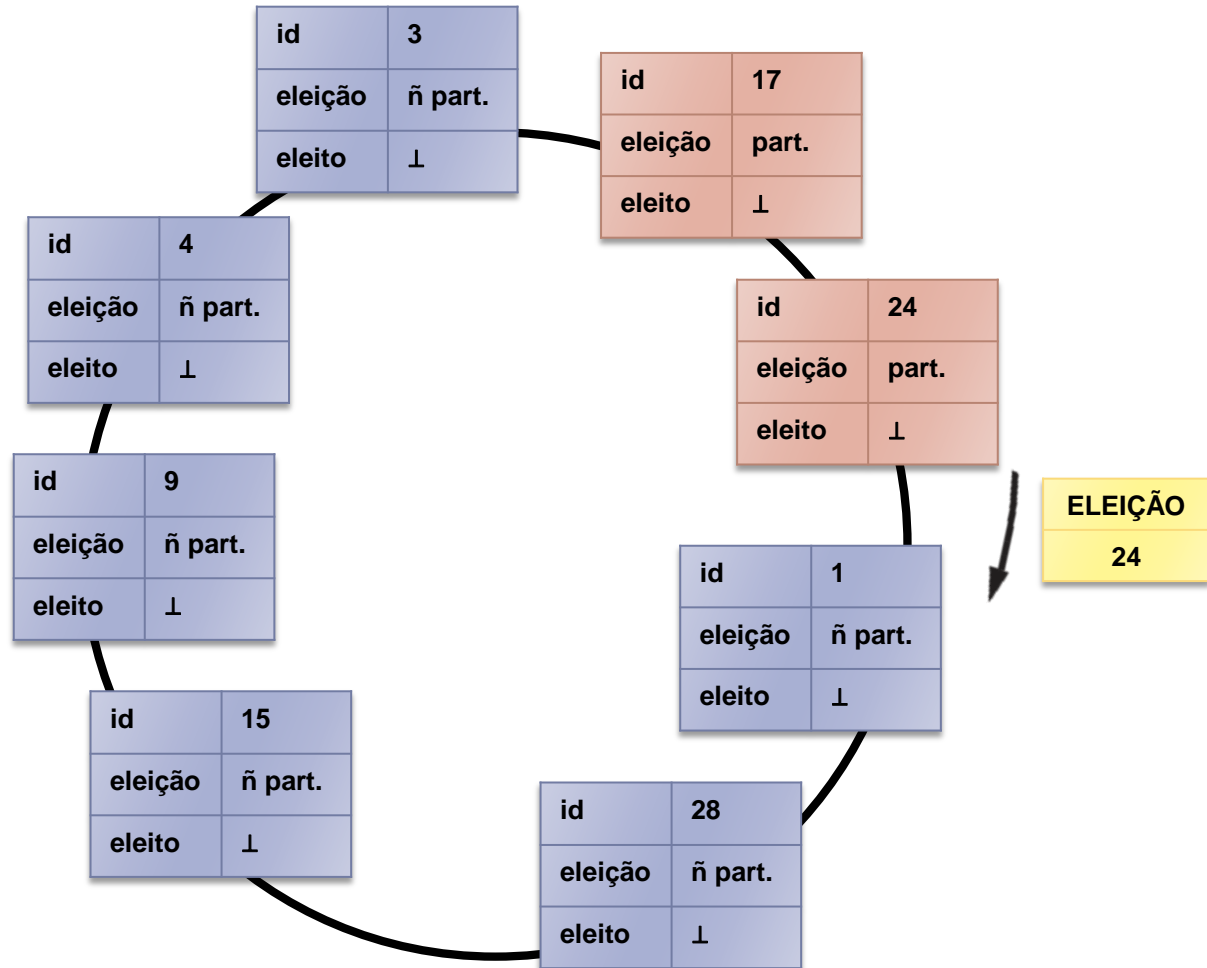
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha

- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



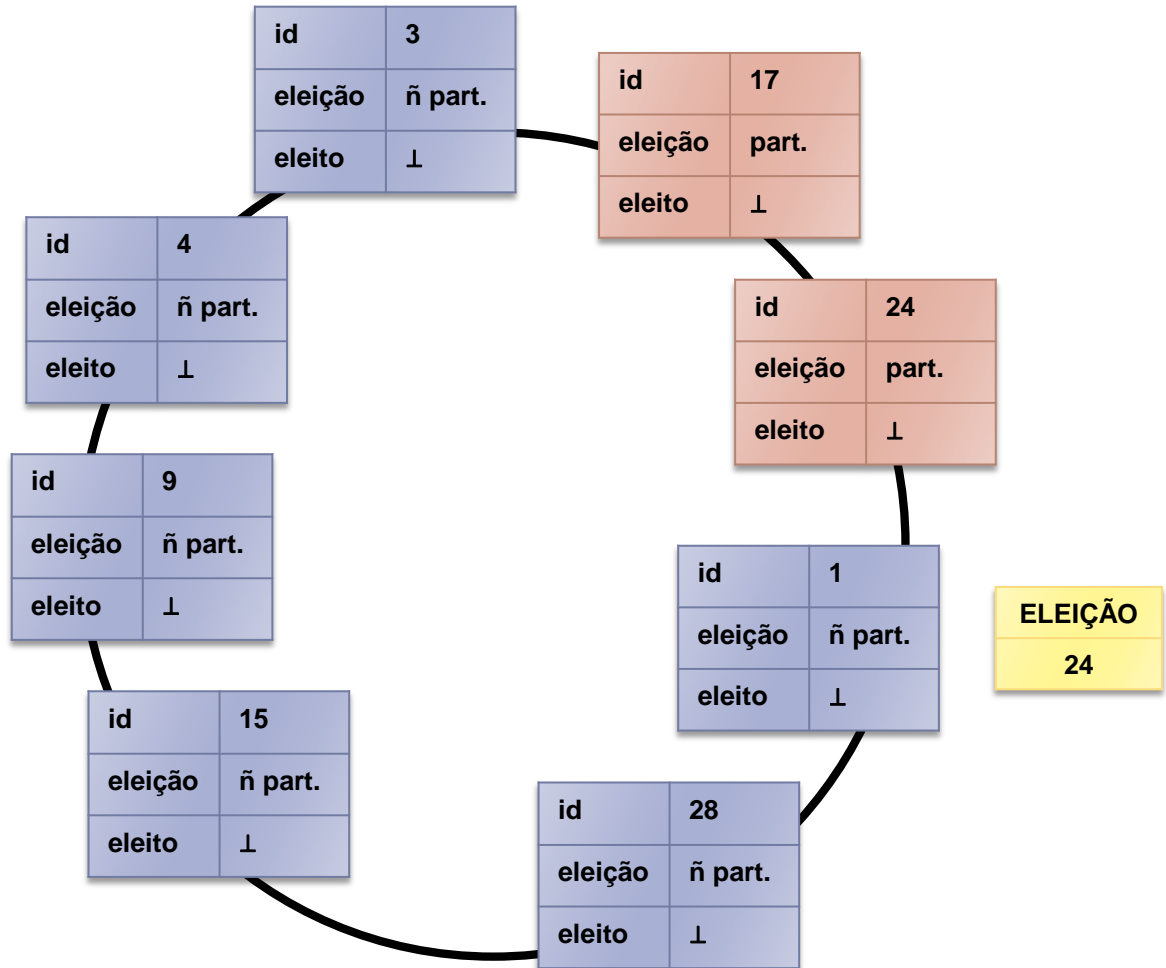
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

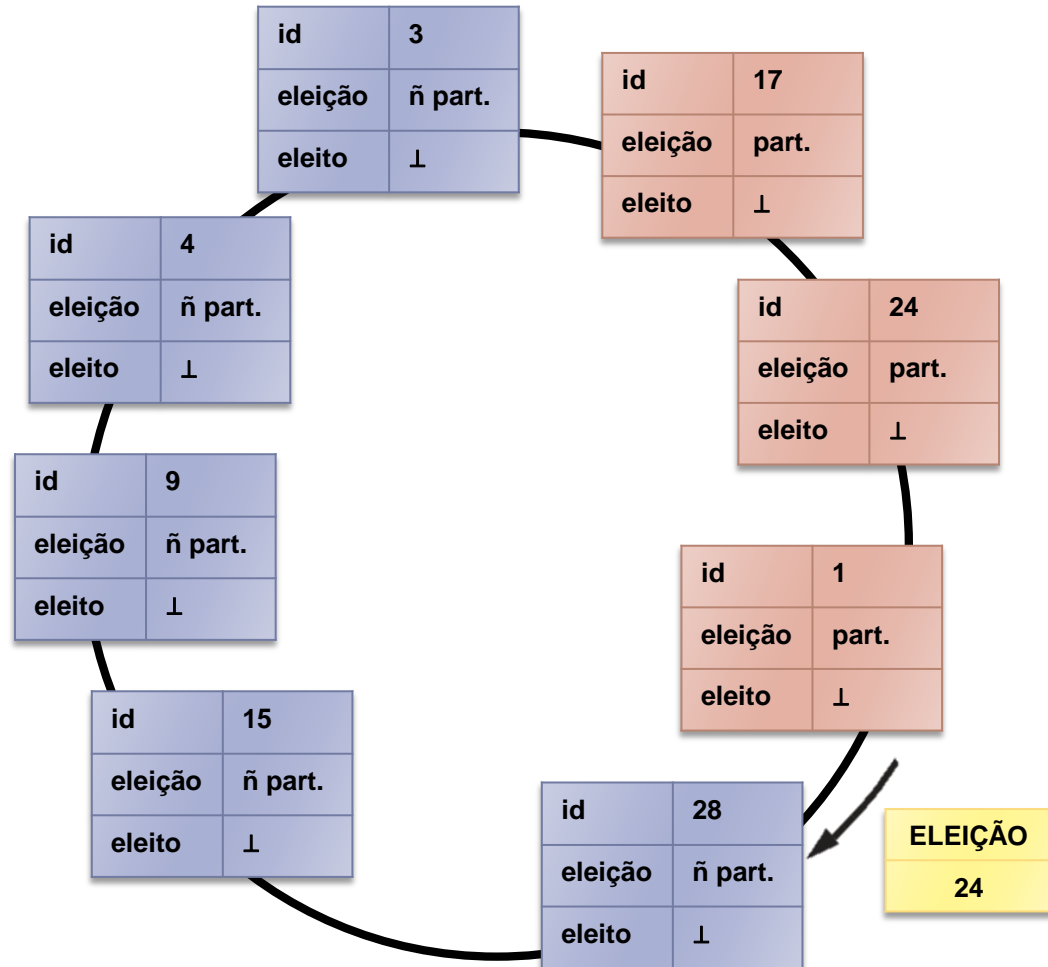
1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

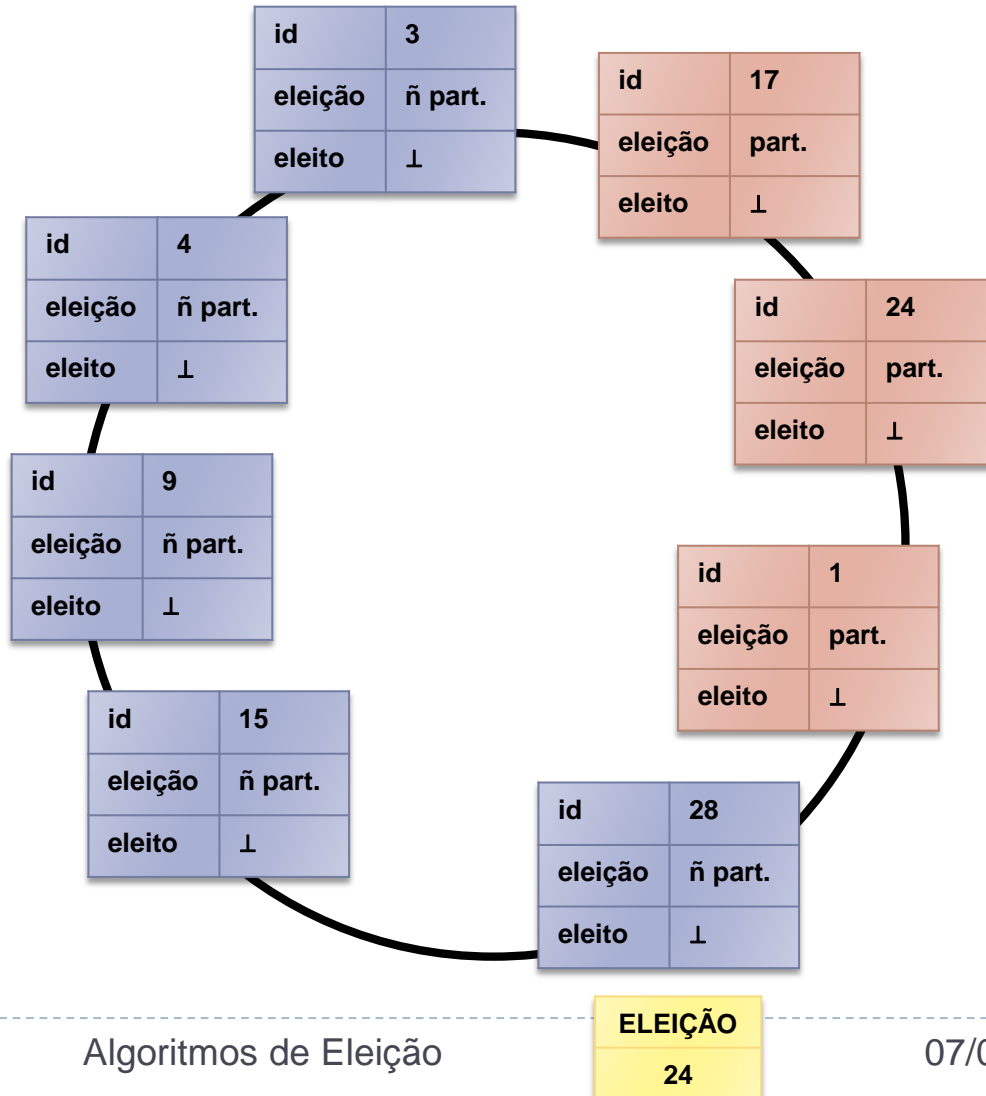


Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

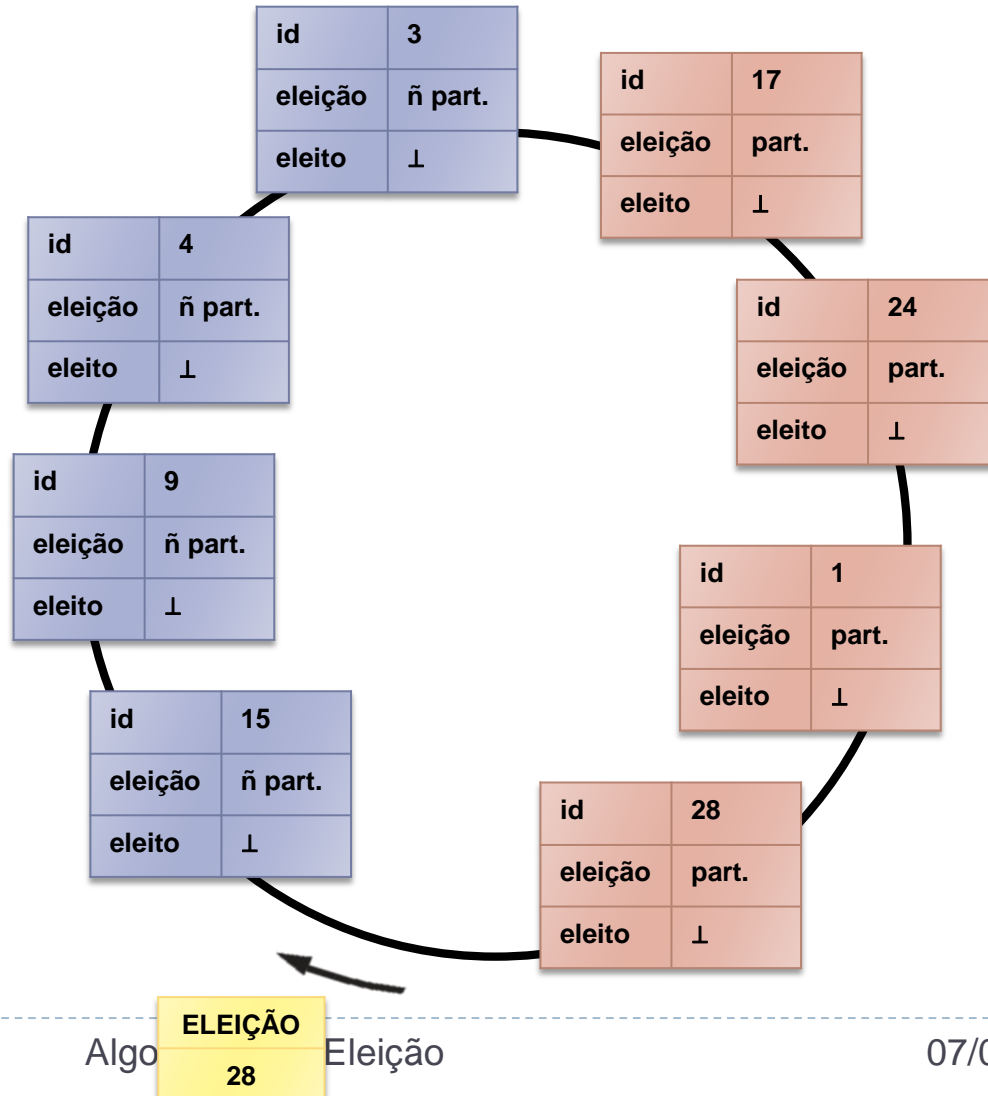
Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:
 - **id_msg > id_proc**
encaminha a mensagem para seu vizinho
 - **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
 - **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

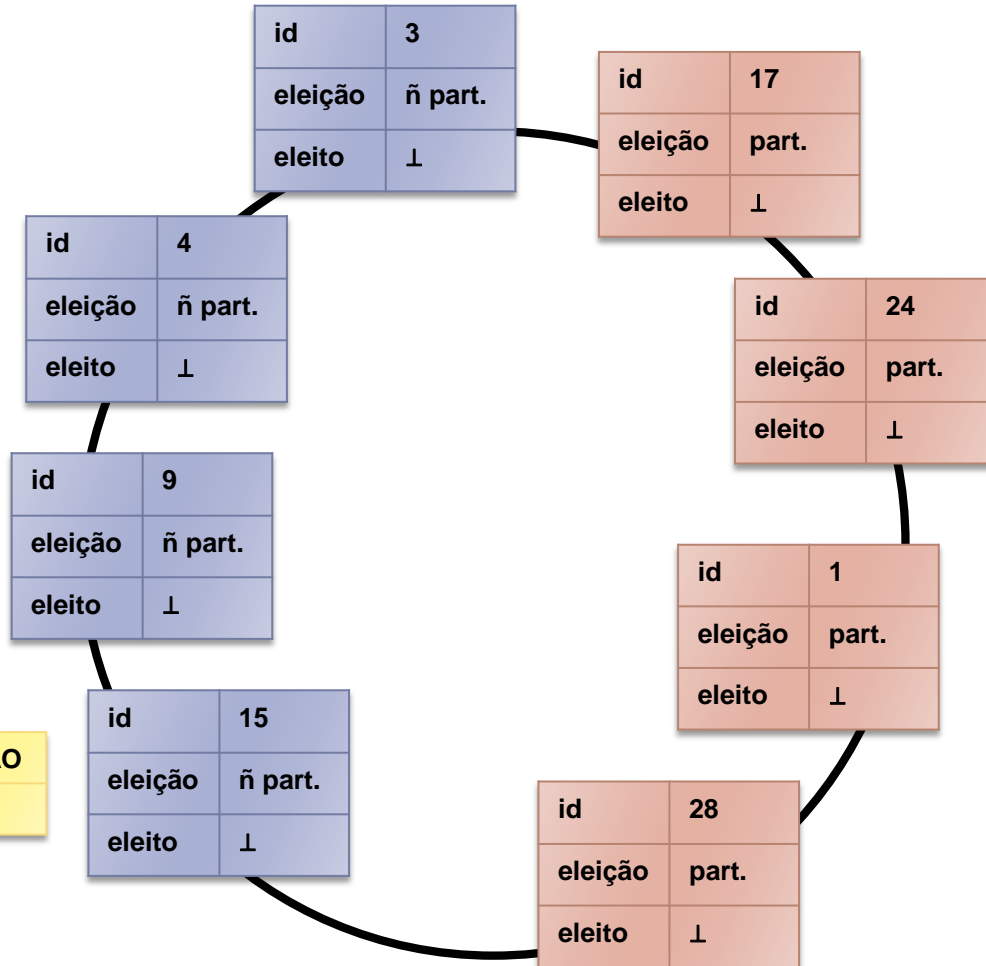
- **id_msg > id_proc**
encaminha a mensagem para seu vizinho

- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha

- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id

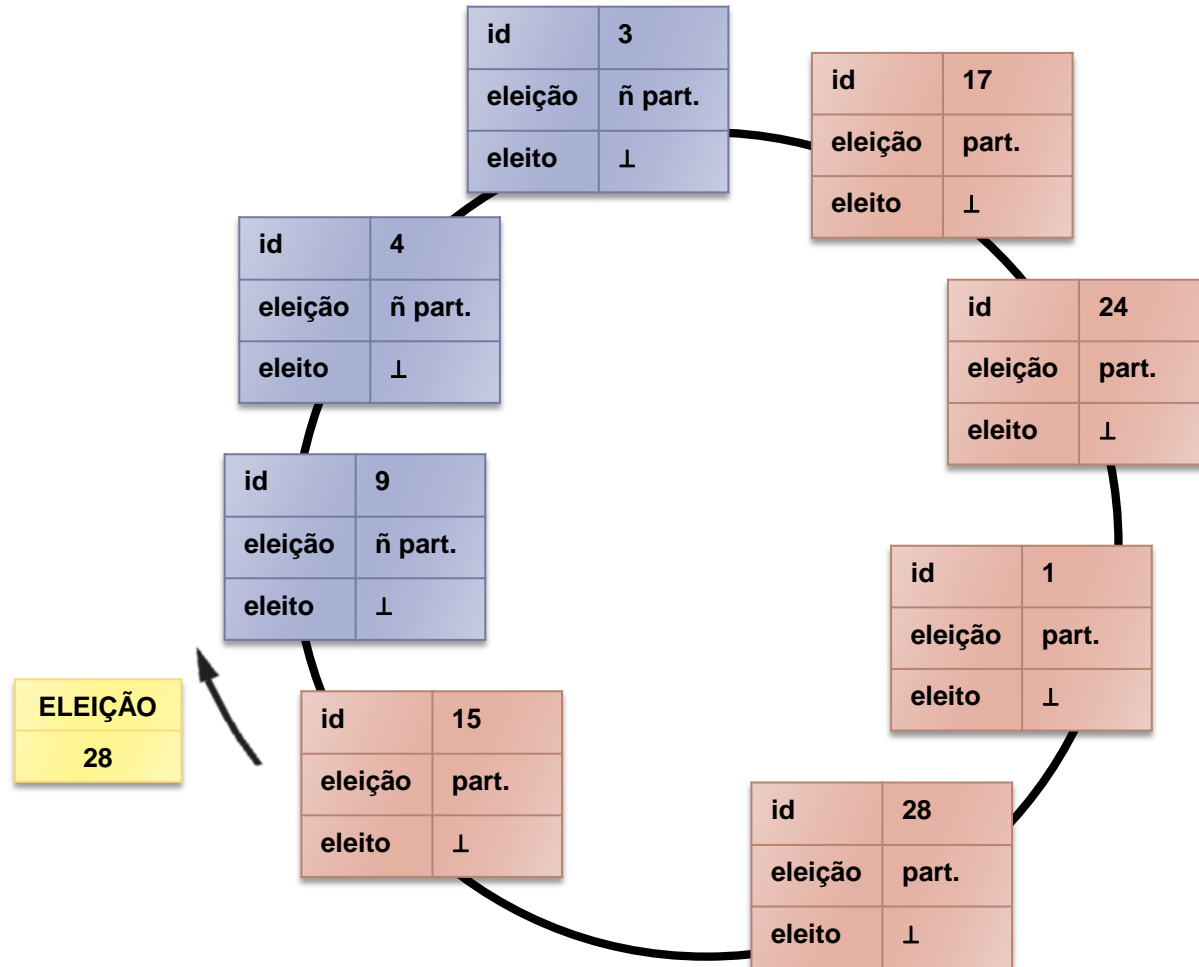
ELEIÇÃO

28



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

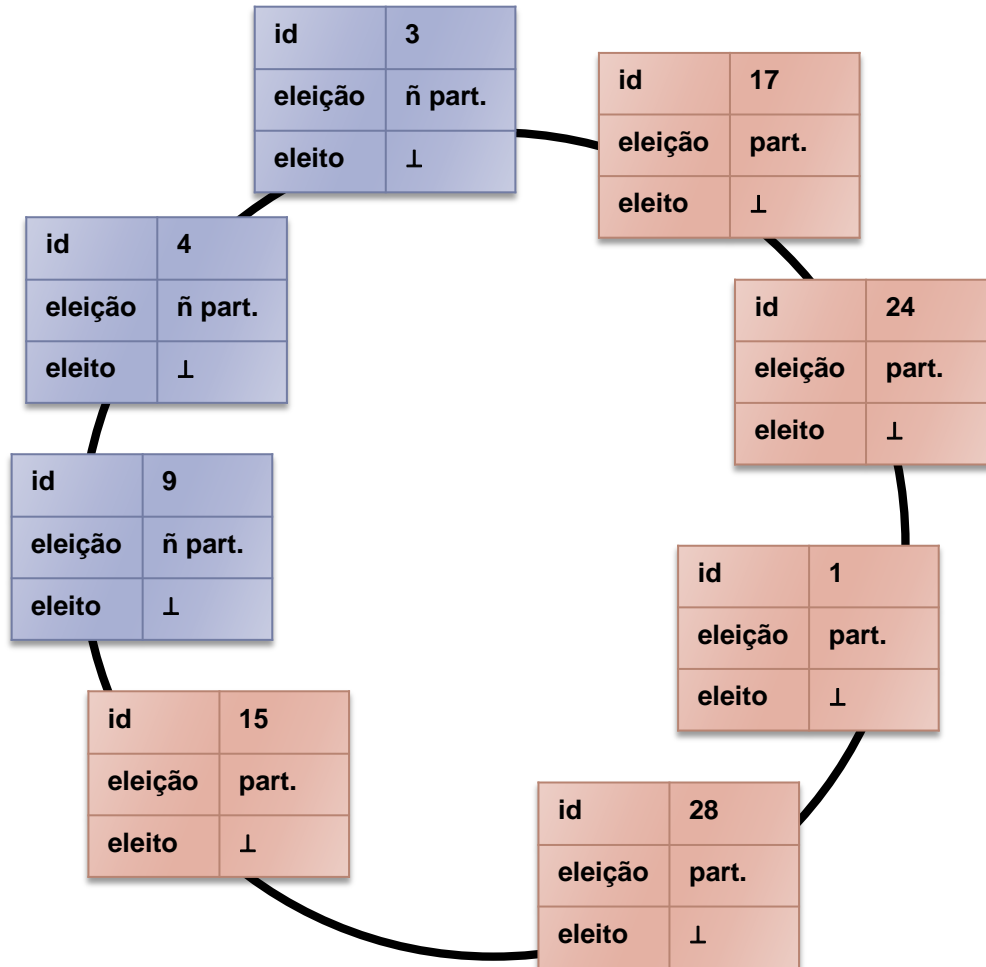
- **id_msg > id_proc**
encaminha a mensagem para seu vizinho

- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha

- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id

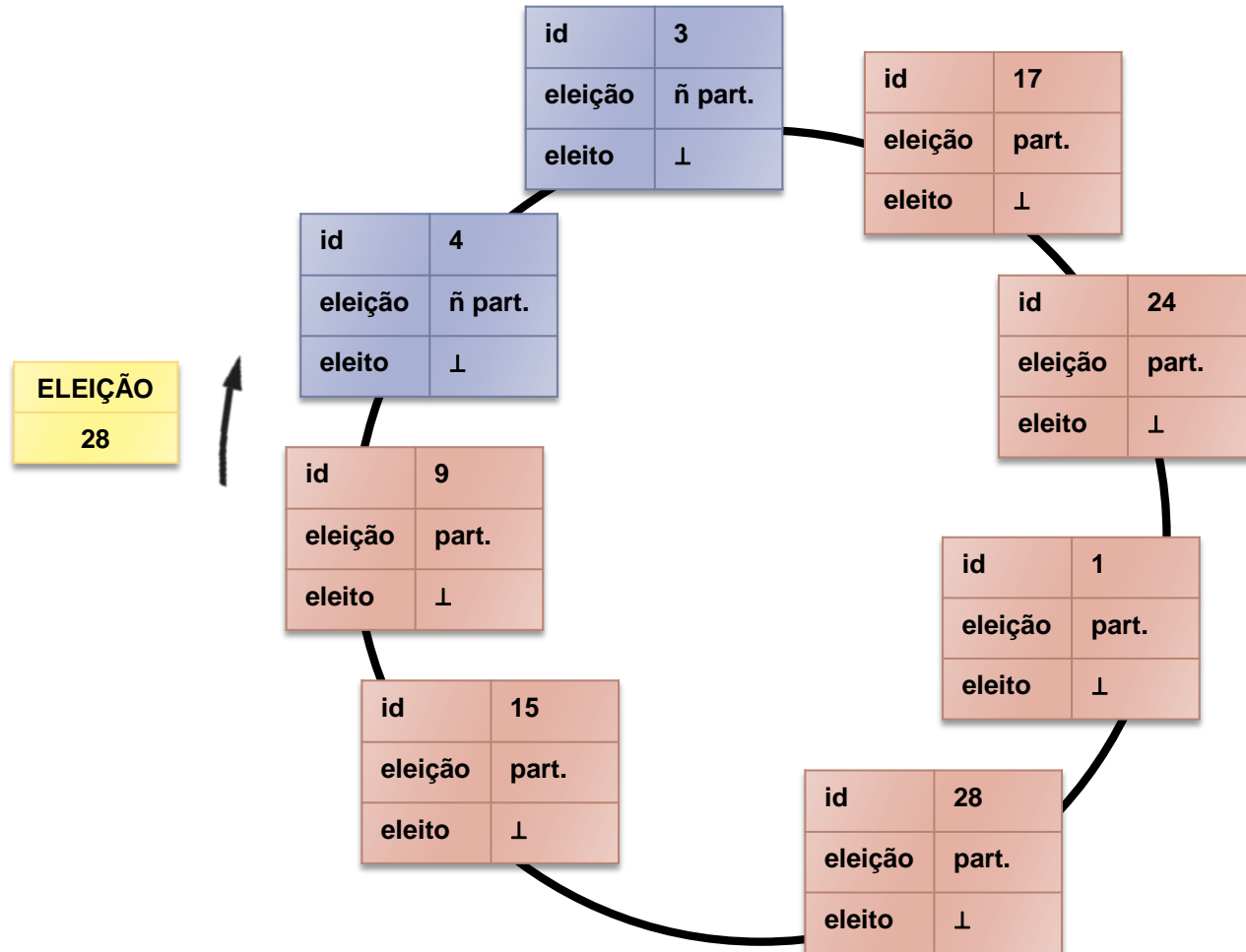
ELEIÇÃO

28



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



Algoritmo de eleição baseado em anel

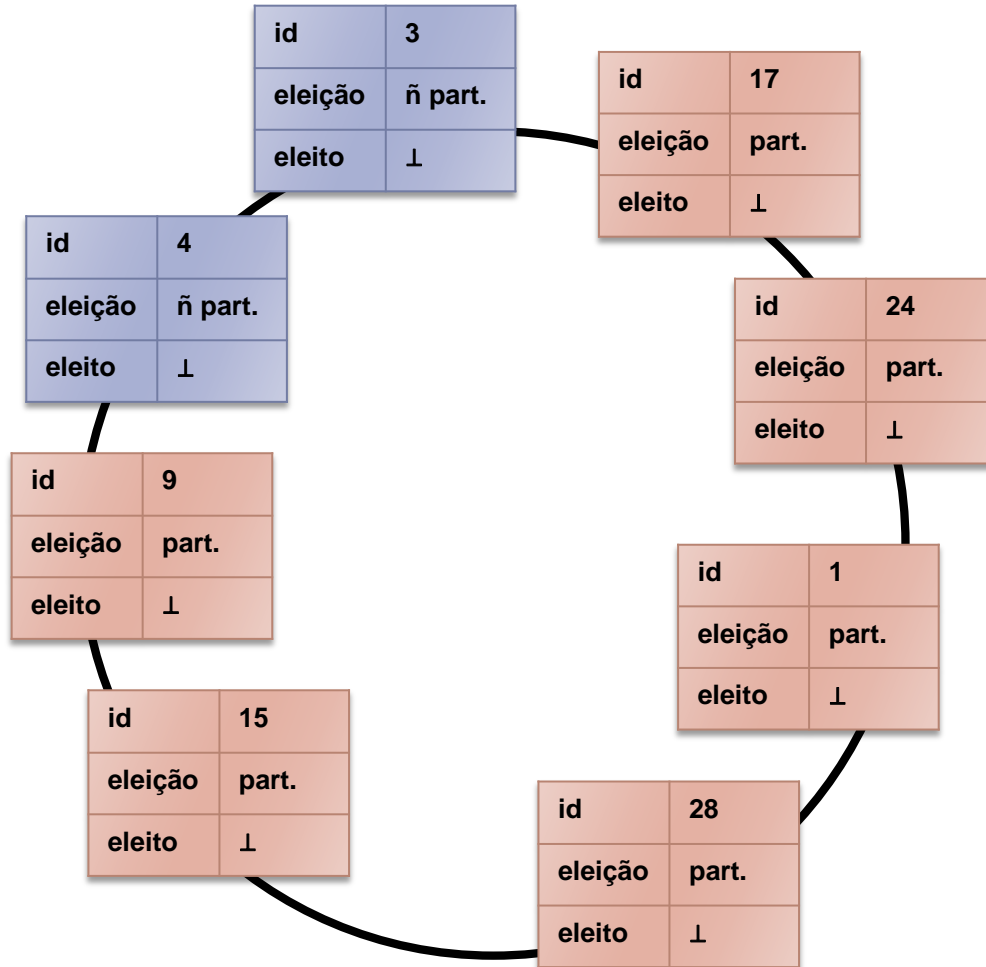
Chang e Roberts [1979]

Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu

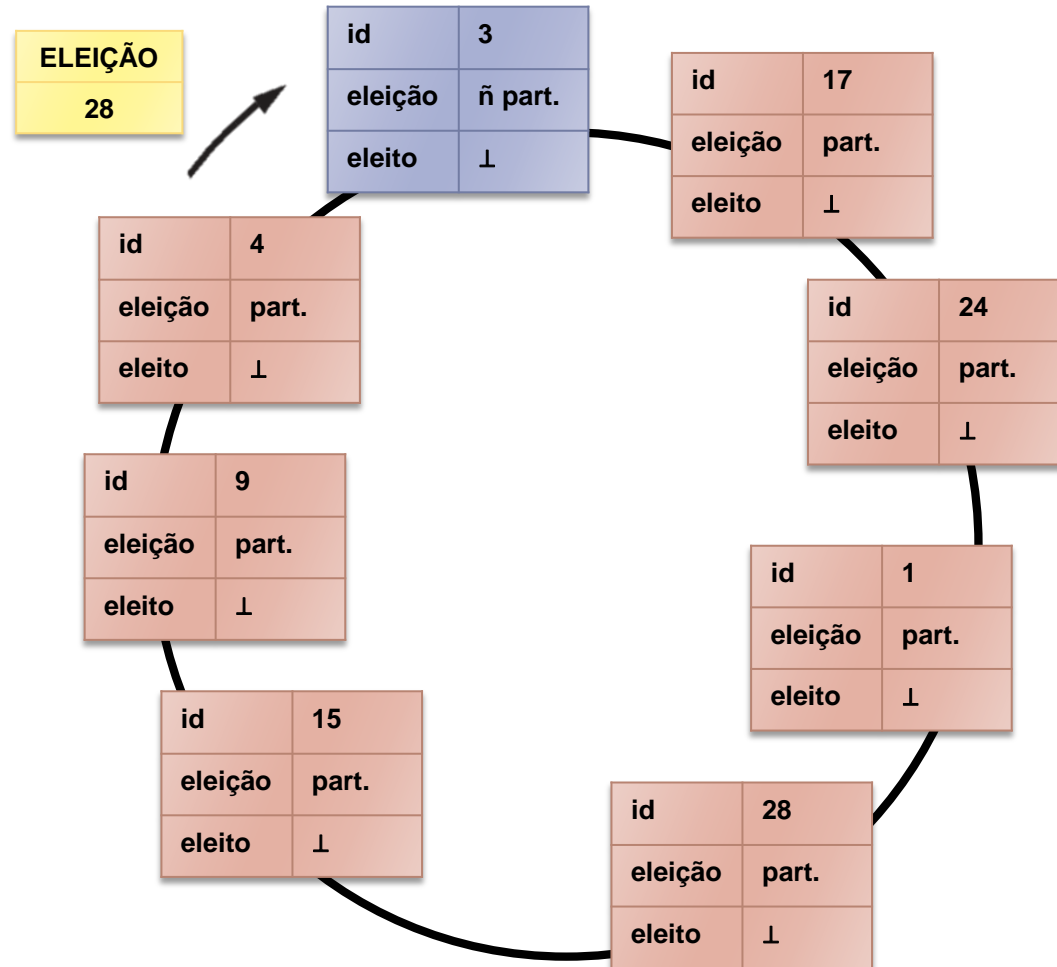
ELEIÇÃO
28

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



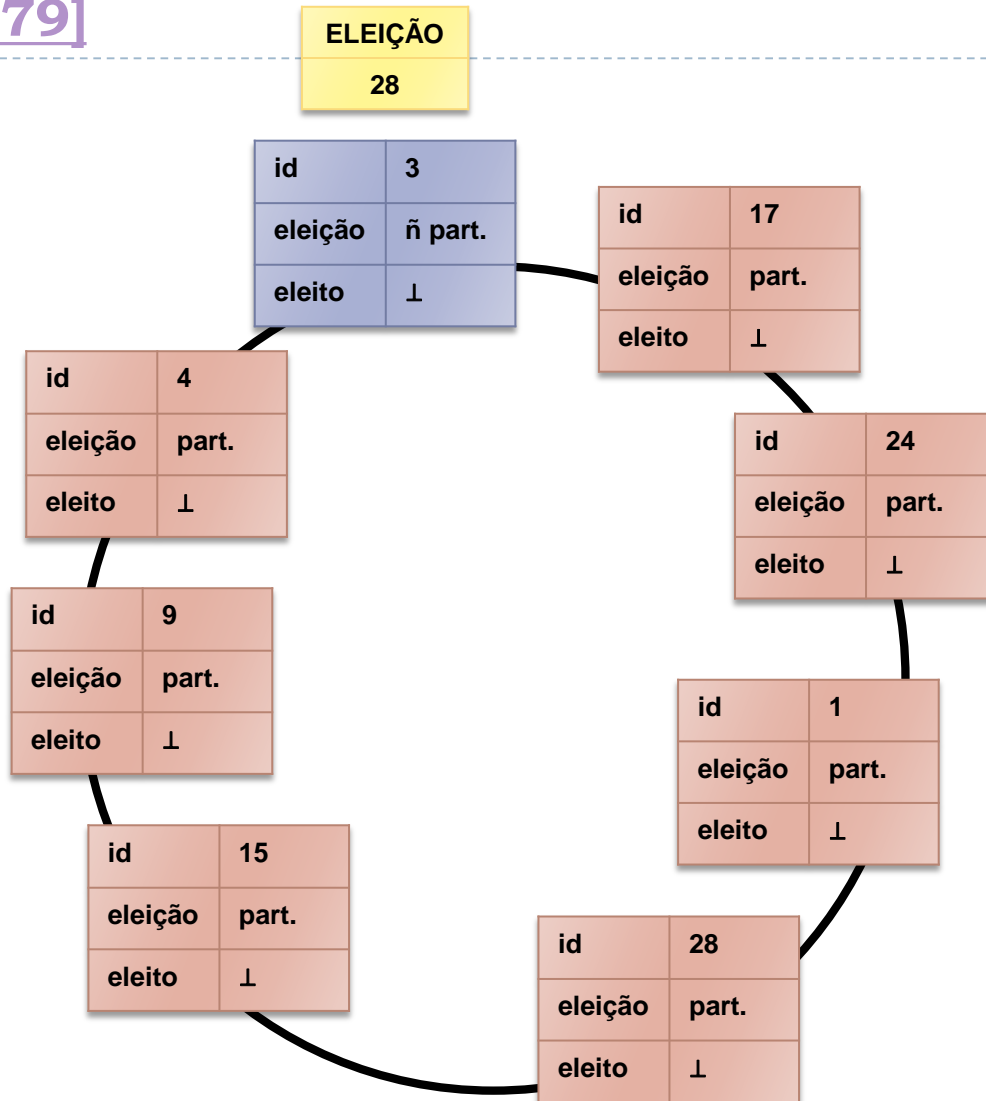
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

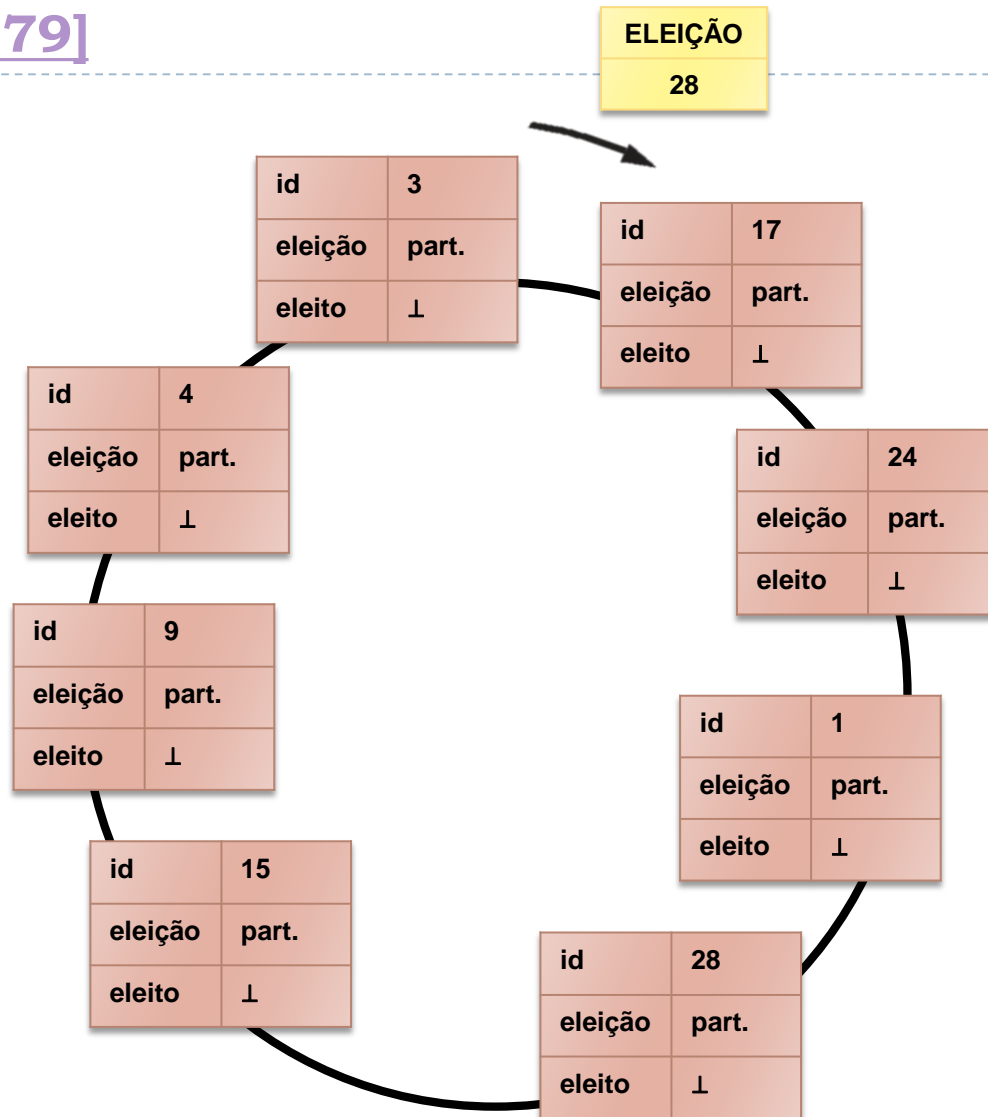
1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



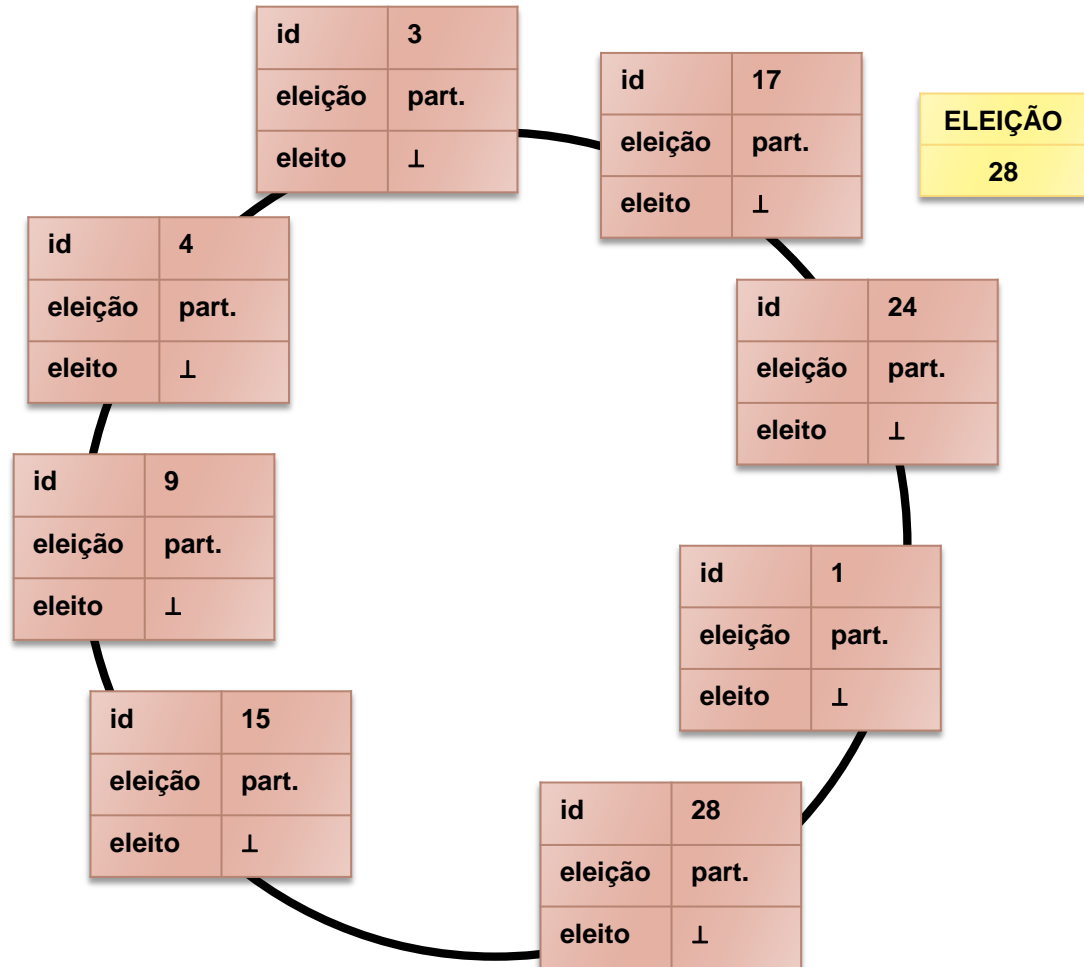
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

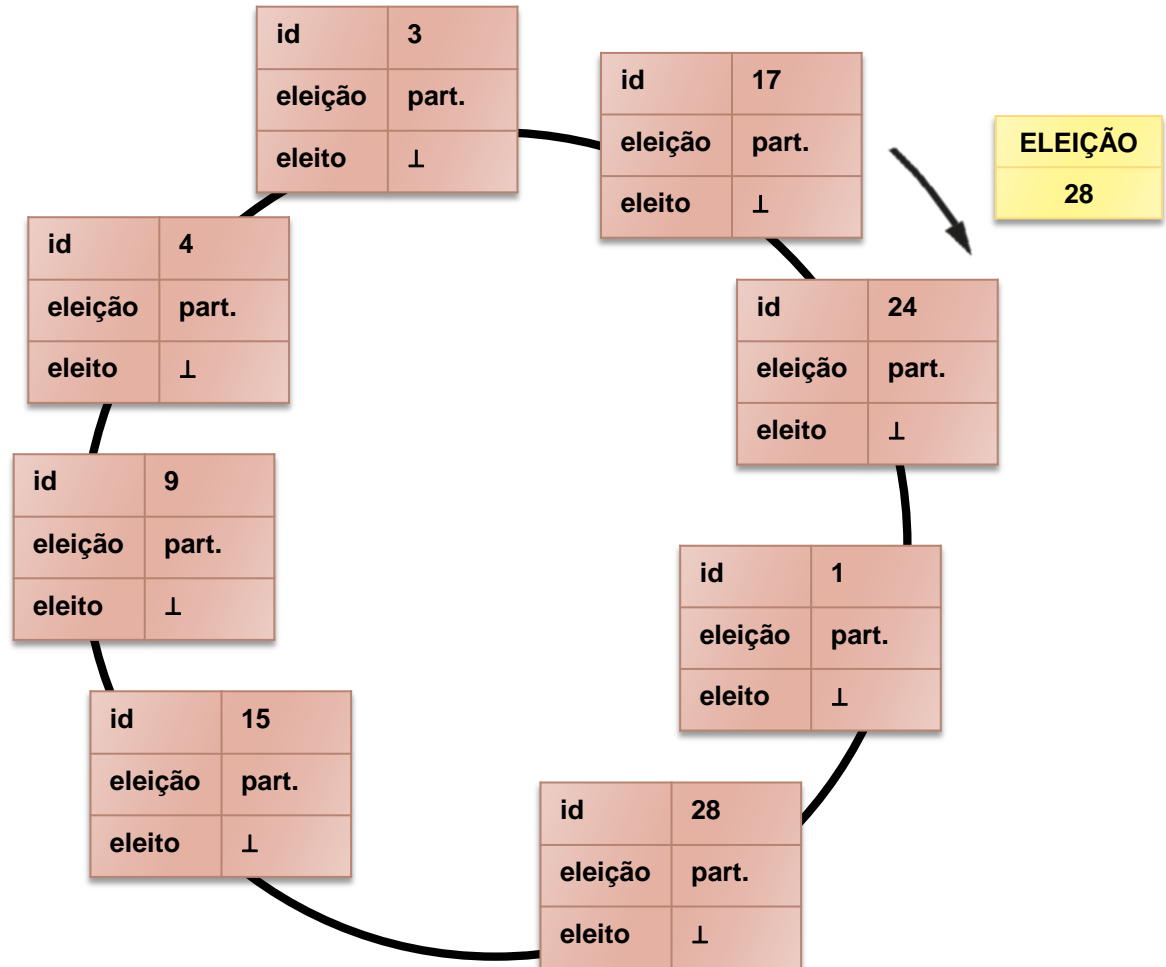
1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



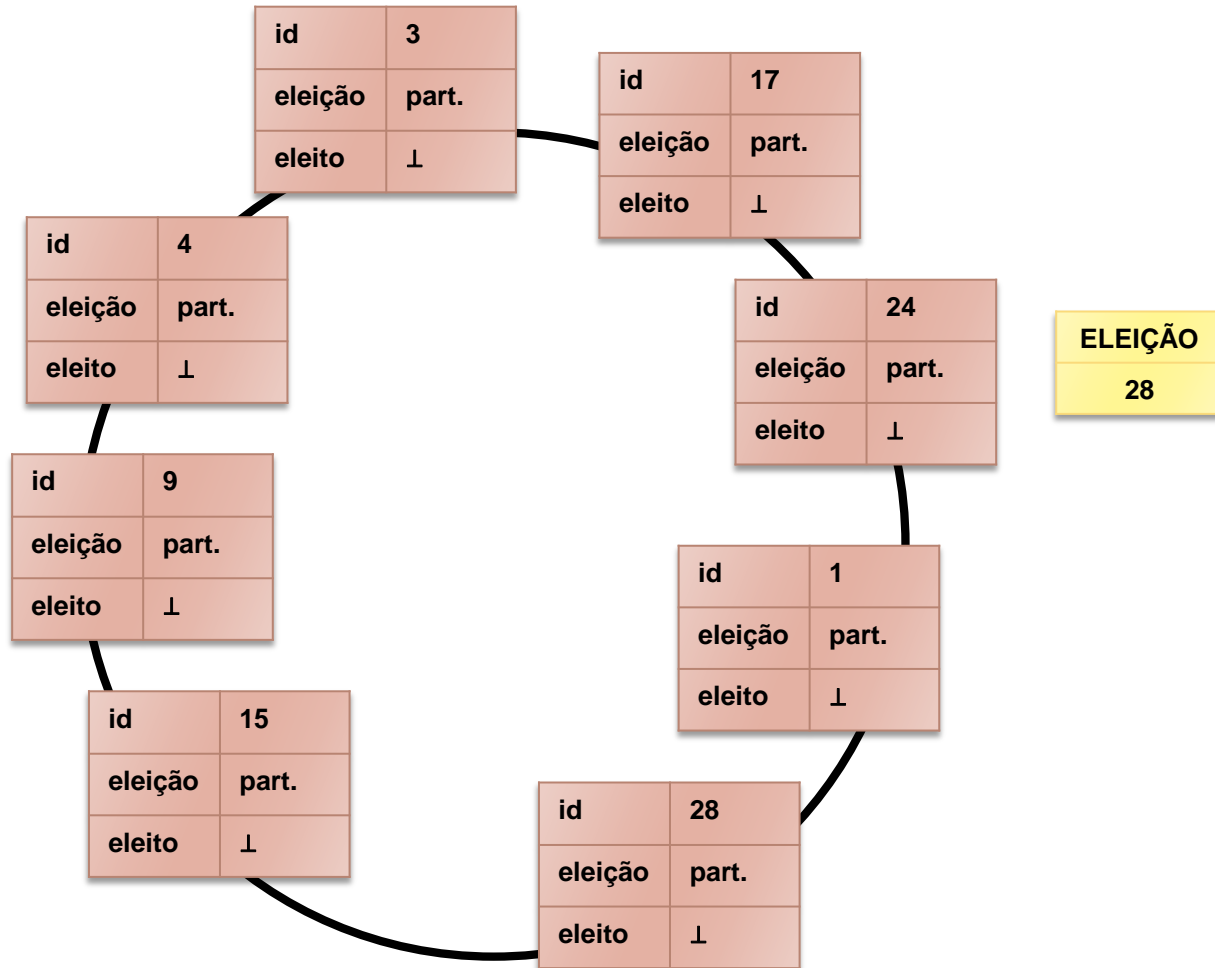
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

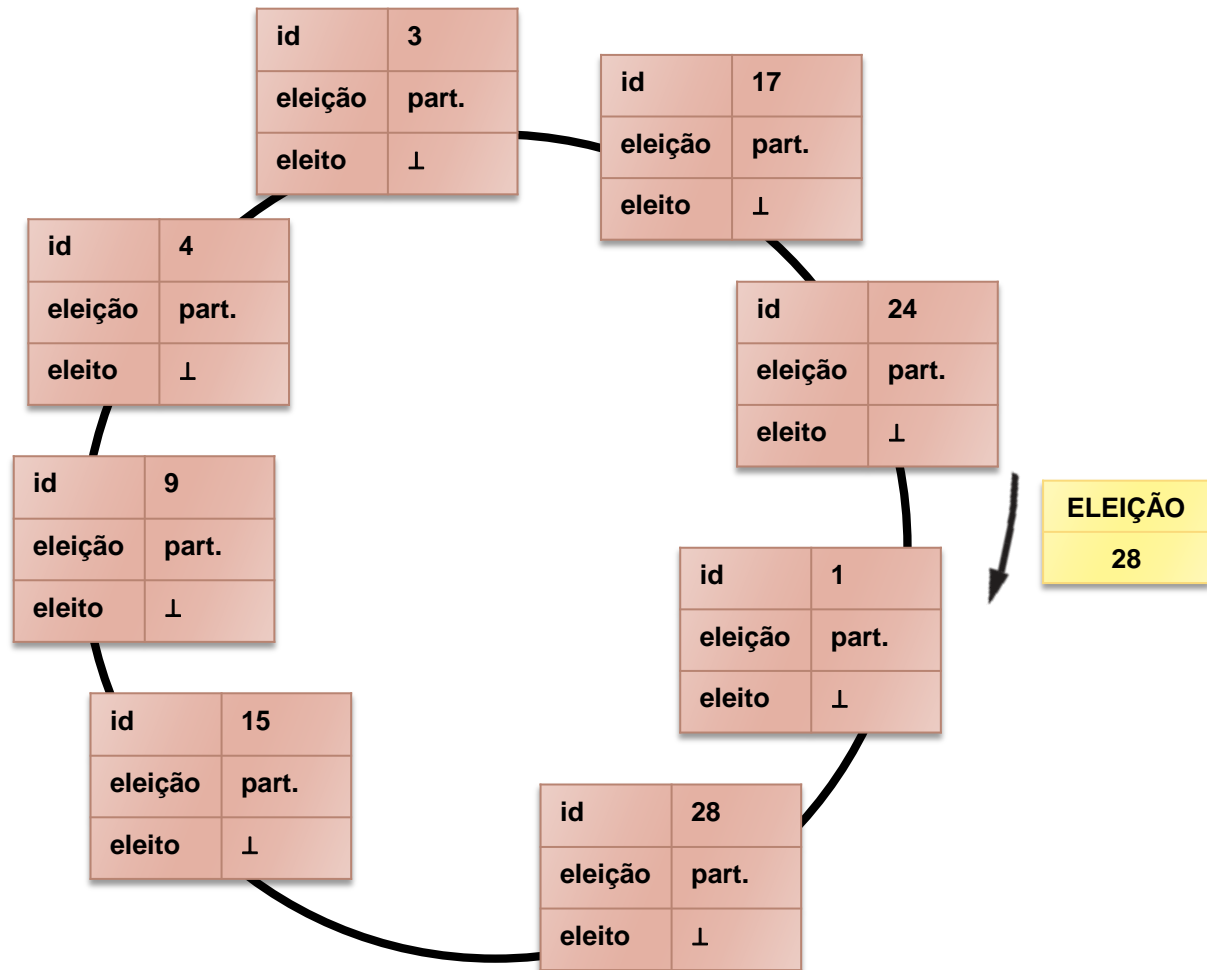
1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



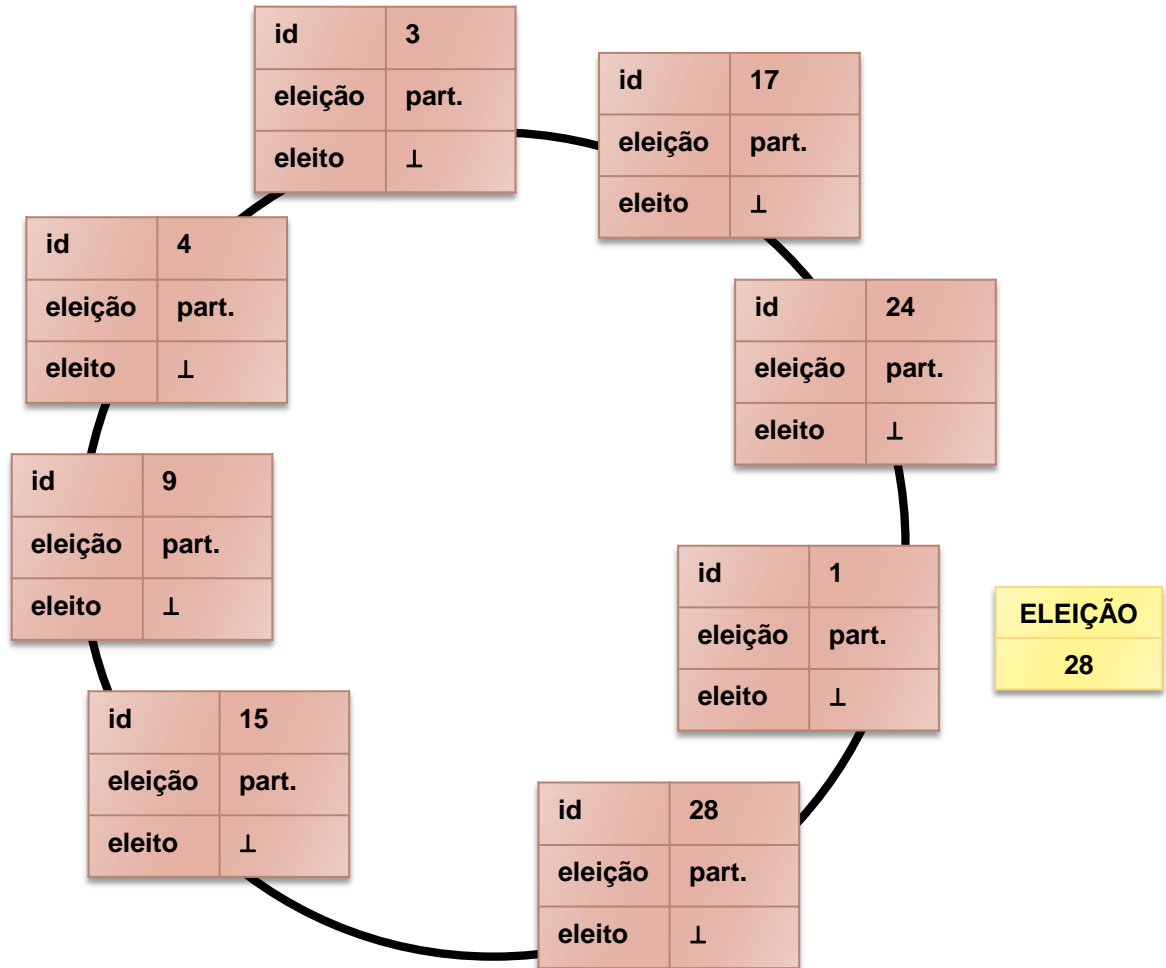
Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

Recebimento de mensagem de eleição

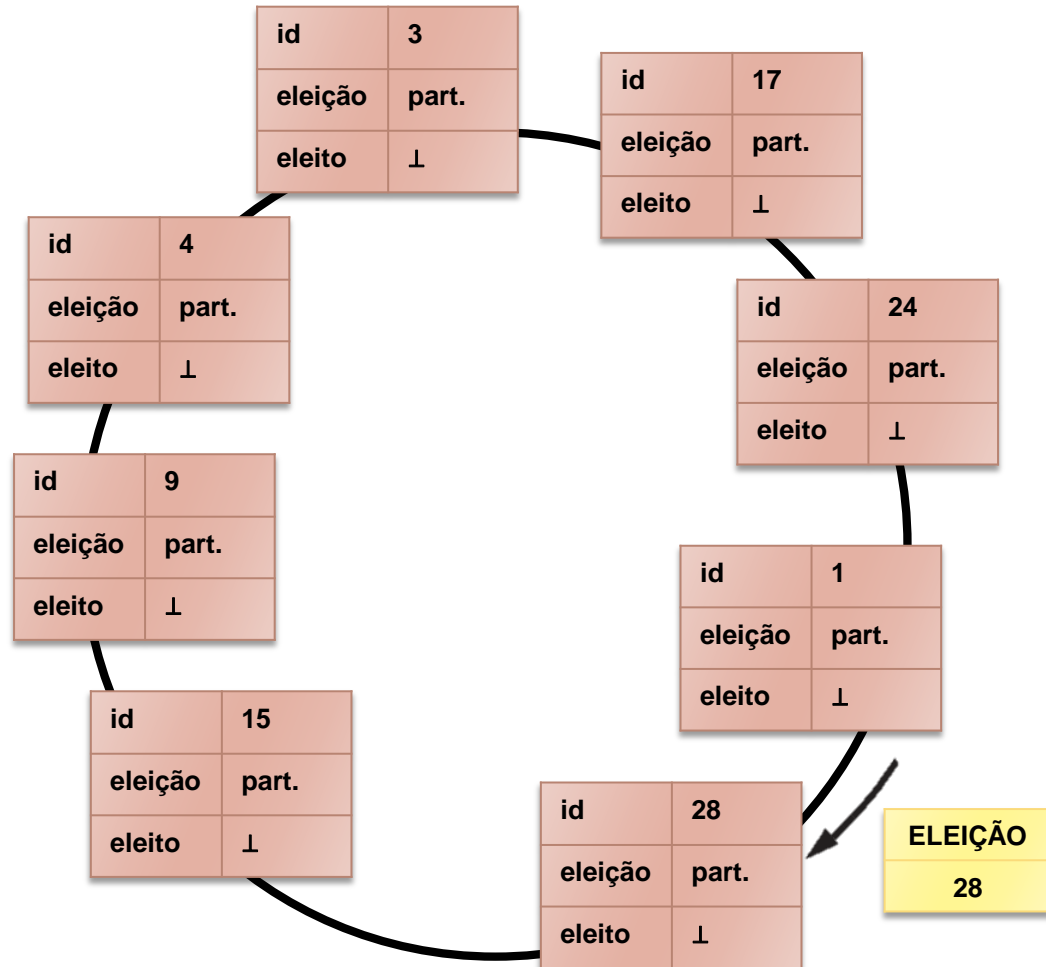
1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:

- **id_msg > id_proc**
encaminha a mensagem para seu vizinho
- **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
- **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

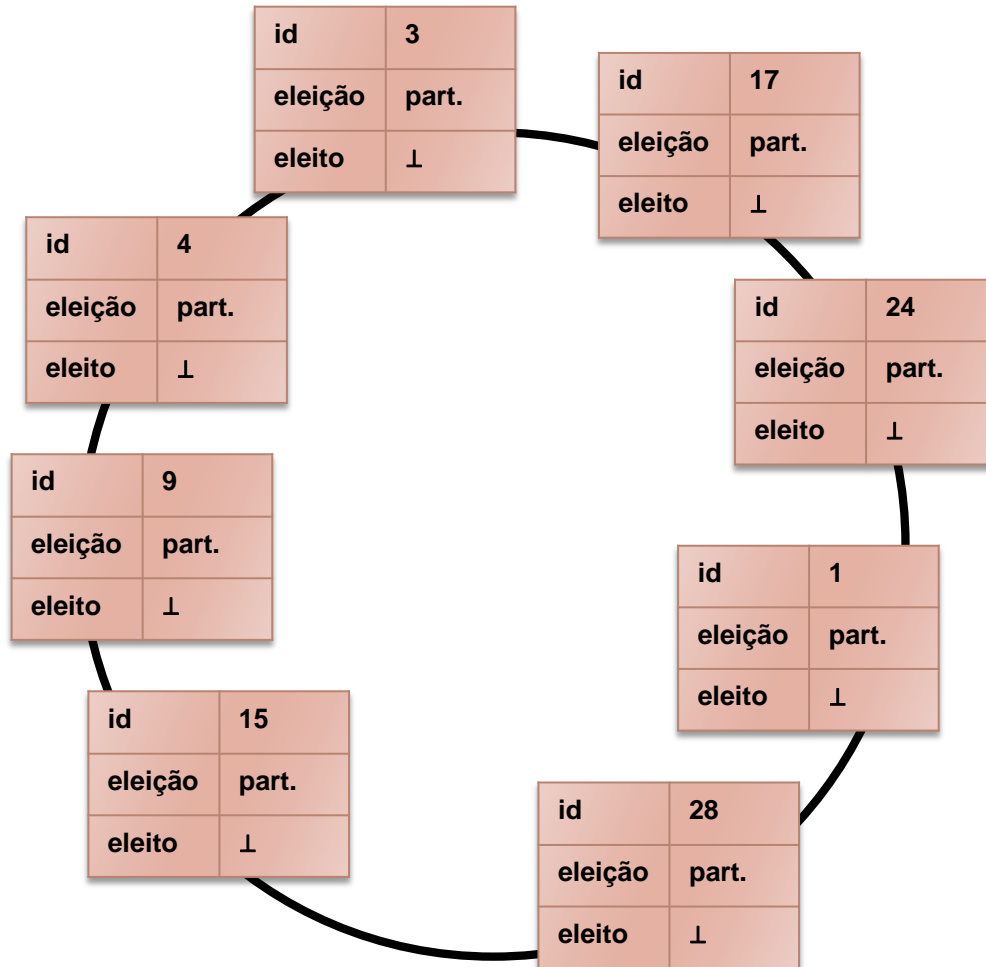


Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

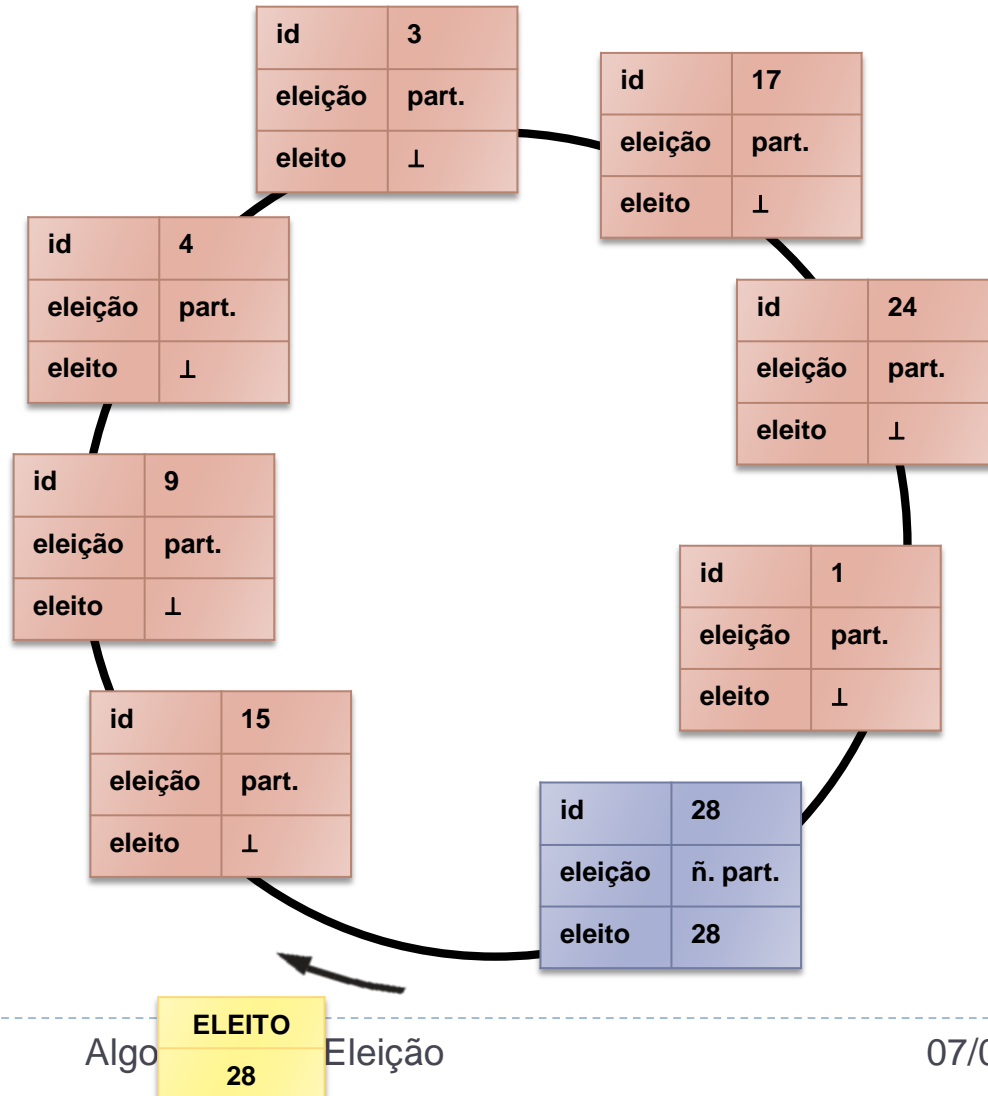
Recebimento de mensagem de eleição

1. O processo marca a si mesmo como **participante**
2. Compara o identificador da mensagem com o seu próprio:
 - **id_msg > id_proc**
encaminha a mensagem para seu vizinho
 - **id_msg < id_proc**
substitui o identificador da mensagem pelo seu próprio identificador e a encaminha
 - **id_msg == id_proc**
O processo marca a si mesmo como **não participante** e envia uma **mensagem de eleito** com seu id



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]



Algoritmo de eleição baseado em anel

Chang e Roberts [1979]

- ▶ Considerações sobre o algoritmo
 - ▶ Algoritmo útil, pois atende às propriedades de eleição
- ▶ É limitado, pois não tolera falhas
 - ▶ Com um detector de falhas confiável, é possível reconstruir o anel quando um processo falha

Algoritmo valentão (*bully*)

Garcia-Molina [1982]

- ▶ Permite que seus processos falhem durante uma eleição
- ▶ Presume que a distribuição de mensagens entre processos é confiável e síncrona
 - ▶ Usa tempos limites para detectar uma falha de processo
- ▶ Presume que cada processo sabe quais processos têm identificadores mais altos e que pode se comunicar com todos esses processos
- ▶ Possui três tipos de mensagem
 - ▶ Eleição: convoca uma eleição
 - ▶ Resposta: resposta à uma mensagem de eleição
 - ▶ Coordenador: anuncia a identidade de um coordenador eleito

Algoritmo valentão (*bully*)

Garcia-Molina [1982]

- ▶ Início do processo de eleição
 - ▶ Um processo inicia o a eleição quando percebe que o coordenador atual falhou, com base nos tempos limites

$$T = 2T_t + T_p$$

- ▶ Por ser um sistema síncrono, vários processos podem perceber a falha do coordenador simultaneamente

Algoritmo valentão (*bully*)

Garcia-Molina [1982]

- ▶ Processo de eleição
 - ▶ O processo que sabe que possui o **identificador mais alto** pode eleger a si mesmo como coordenador
 - ▶ Envia uma **mensagem de coordenador** para todos os processos com identificadores mais baixos

Algoritmo valentão (*bully*)

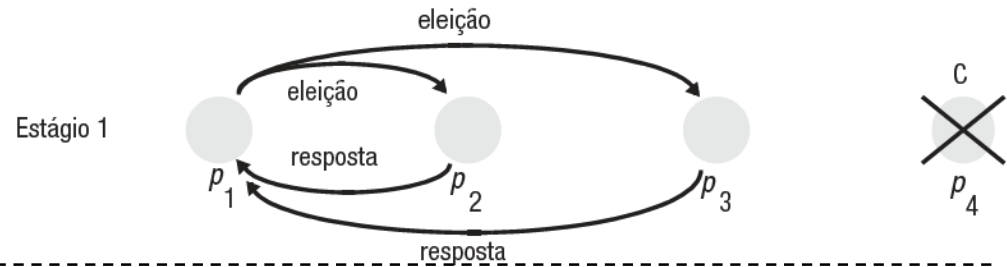
Garcia-Molina [1982]

- ▶ Processo de eleição
 - ▶ Um processo com um **identificador mais baixo** inicia uma eleição:
 - ▶ Enviando uma **mensagem de eleição** para os processos que têm identificador mais alto e esperando uma mensagem de resposta
 - Se nenhuma **mensagem de resposta** chegar dentro do tempo T
 - O processo se considerará o coordenador e enviará uma **mensagem de coordenador** para todos os processos com identificadores mais baixos
 - Se chegar uma **mensagem de resposta**, o processo esperará, por mais um período T , que chegue uma **mensagem de coordenador** do novo coordenador
 - Se a mensagem de coordenador não chegar, ele iniciará outra eleição
 - Se a mensagem de coordenador chegar, ele configura sua variável eleito com o identificador do coordenador contido na mensagem

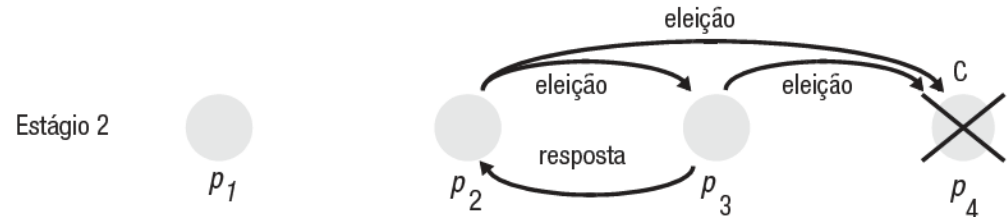
Algoritmo valentão (*bully*)

Garcia-Molina [1982]

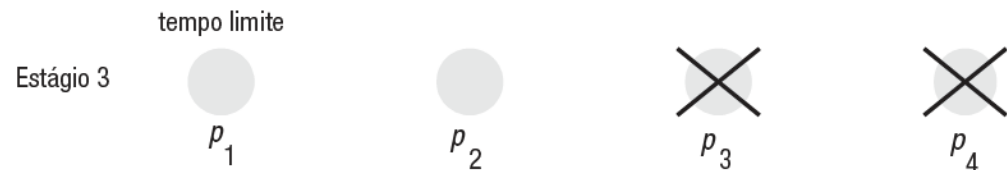
- O processo p_1 detecta a falha do coordenador p_4 e anuncia uma eleição
- Os processos p_2 e p_3 enviam mensagens de *resposta* para p_1 e iniciam suas próprias eleições



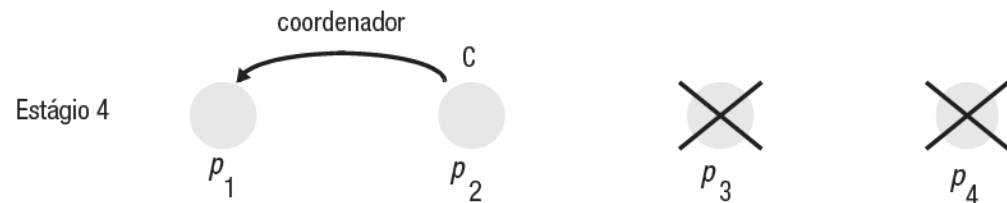
- p_3 envia uma mensagem de resposta para p_2 , mas p_3 não recebe nenhuma mensagem de resposta do processo falho p_4



- p_3 decide que é o coordenador, mas antes que possa enviar a mensagem de coordenador, ele também falha



- Quando o período do tempo limite de p_1 expira, ele deduz a ausência de uma mensagem de coordenador e inicia outra eleição.
- Finalmente, p_2 é eleito coordenador



Algoritmo valentão (*bully*)

Garcia-Molina [1982]

- ▶ Origem do termo “valentão”
 - ▶ Quando um processo para substituir um processo falho é iniciado, ele inicia uma eleição
 - ▶ Se tiver o identificador de processo mais alto que os demais, decidirá que é o coordenador e anunciará isso para os outros processos
 - ▶ Ele se tornará o coordenador, mesmo que o coordenador corrente esteja funcionando

Algoritmo valentão (*bully*)

Garcia-Molina [1982]

- ▶ Considerações sobre o algoritmo
 - ▶ Condição de subsistência (E2)
 - ▶ Satisfaz, pois supõe o envio de mensagem confiável
 - ▶ Condição de segurança (E1)
 - ▶ Satisfaz, caso nenhum processo seja substituído com mesmo id
 - ▶ É impossível dois processos decidirem simultaneamente que são o coordenador
 - O processo com o identificador mais baixo descobrirá que o outro existe e o acatará
 - ▶ Não é garantido que o algoritmo satisfaça, caso processos que tenham falhado sejam substituídos por processos com o mesmo id
 - ▶ Um processo que substitui um processo falho p pode decidir que tem o identificador mais alto, assim como outro processo (que detectou a falha de p) decidiu que possui o identificador mais alto
 - Os dois processos se anunciarão como coordenadores, simultaneamente

Dúvidas?



Obrigado!

Referências

COULOURIS, George et al. **Sistemas Distribuídos: Conceitos e Projeto.** Bookman Editora, 2013.