



# LOẠI TRỪ TƯƠNG HỒ

**ThS. LÊ NGỌC BẢO**

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG CƠ SỞ TPHCM

# Loại trừ tương hỗ

Sự tương tranh và cộng tác giữa các tiến trình. Có thể xảy ra trường hợp cùng một lúc có nhiều tiến trình cùng truy nhập đến một tài nguyên dẫn đến sự xung đột hoặc một thao tác nào đó không được thực thi trọn vẹn.

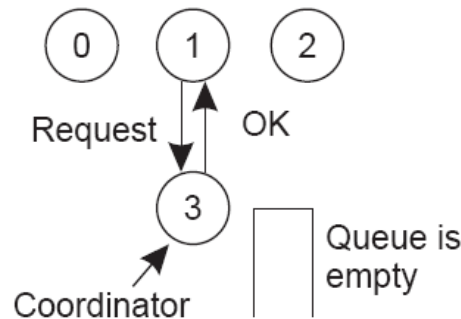
Hai phương pháp khác nhau: sử dụng thẻ bài và cấp quyền sử dụng.

- + Phương pháp thẻ bài là duy trì một thông điệp duy nhất (gọi là thẻ bài) di chuyển trong một nhóm các thành viên của hệ thống. Thành viên nào nắm giữ thẻ bài thì có quyền truy nhập tài nguyên, nếu không có nhu cầu truy nhập tài nguyên thì sẽ chuyển cho thành viên kế tiếp.

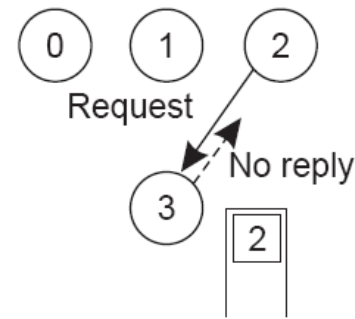
- + Phương pháp cấp quyền sử dụng hoạt động theo hai bước: Gửi yêu cầu đề nghị được cấp quyền sử dụng, sau khi được cấp quyền mới chuyển sang bước truy nhập tài nguyên.

# PP 1: Giải thuật tập trung

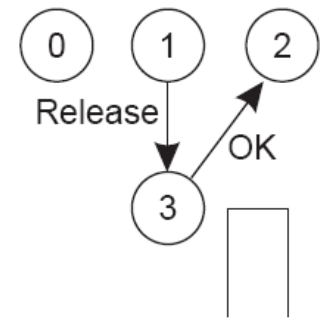
- Cách đơn giản nhất:
  - Các yêu cầu vào và ra các critical section được gửi tới một server khóa
  - Tiến trình được quyền vào khi nhận được một token
  - Khi ra khỏi critical section, token được trả lại cho server



(a)



(b)



(c)

- Đặc điểm:
  - Dễ thực hiện
  - Khó mở rộng
  - Server trung tâm có thể thất bại

# Giải thuật không tập trung

•

Giải thuật tập trung chỉ có một thành phần điều phối, các thành viên chỉ có duy nhất một lựa chọn.

- Mỗi tài nguyên sẽ được nhân bản N lần và có tiến trình điều phối riêng để kiểm soát truy nhập.
- Một tiến trình muốn truy nhập tài nguyên thì phải gửi yêu cầu đến N tiến trình điều phối đó và sẽ được cấp quyền truy nhập khi và chỉ khi nhận được số phiếu đồng ý lớn hơn N/2.

Giải thuật này ít bị lỗi hơn, gọi p là xác suất bị lỗi của mỗi tiến trình điều phối, xác suất k trong m tiến trình điều phối bị lỗi sẽ là:

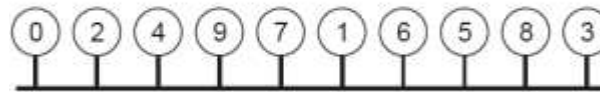
$$P[k] = \binom{m}{k} p^k (1-p)^{m-k}$$

# PP 2: Token ring – Giải thuật thẻ bài

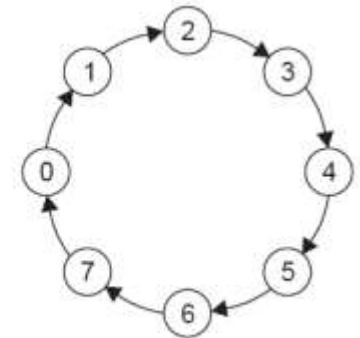
- Các tiến trình được xếp trong một cấu trúc vòng (lôgic)
- Một thông điệp token được chuyển dần cho nhau quanh vòng
- Trước khi vào critical section, tiến trình phải đợi token đến
- Phải giữ token cho đến khi ra khỏi critical section

- Tính chất:

- Độ trễ trung bình  $N/2$  chặng
- Token chiếm băng thông
- Vòng vỡ khi nút hỏng hoặc kênh hỏng



(a)



(b)

## PP 3: Giải thuật phân tán

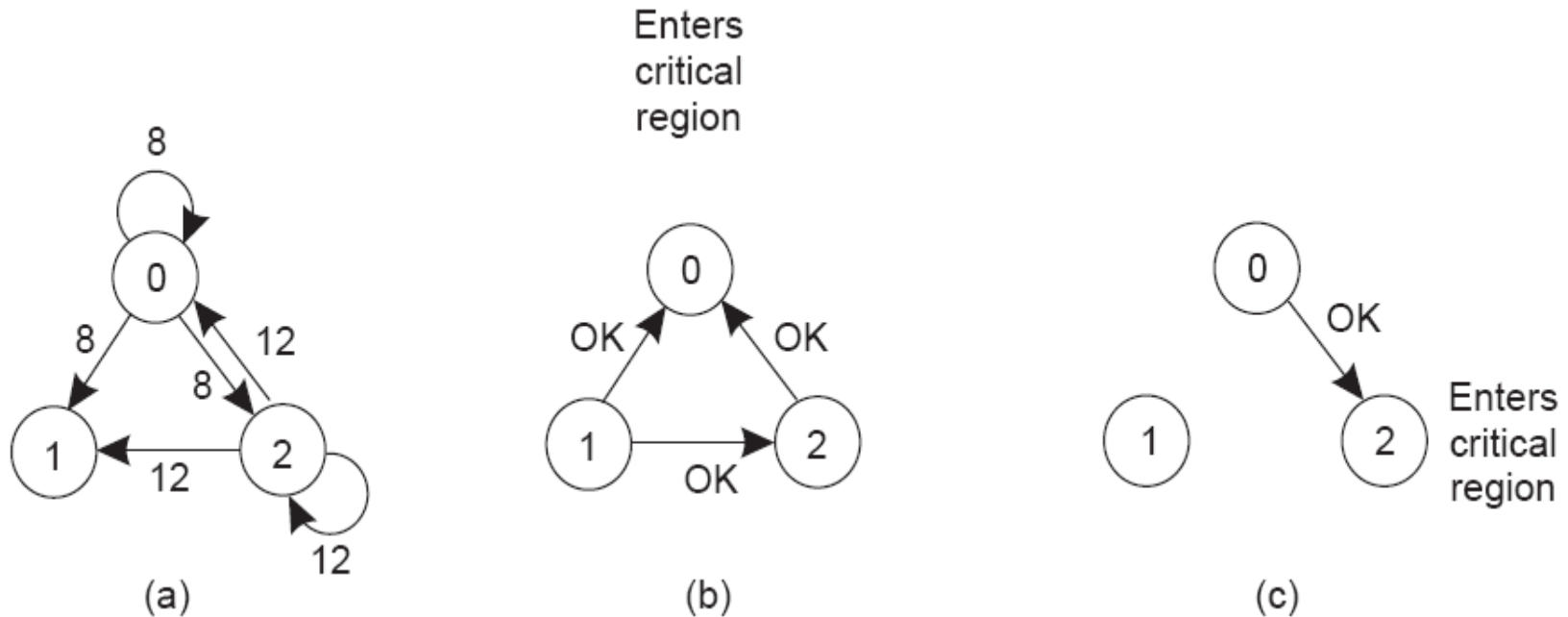
- Thuật toán của Ricart & Agrawala:
- Mỗi tiến trình giữ một đồng hồ Lamport và có thể liên lạc theo từng cặp
- Tiến trình ở một trong ba trạng thái:
  1. **Thả**: ở ngoài critical section
  2. **Muốn**: đợi vào critical section
  3. **Giữ**: đang ở trong critical section

## PP 3: Giải thuật phân tán

### ➤ Ứng xử của tiến trình:

1. Nếu một tiến trình  $p_i$  muốn vào critical process, nó
  - multicast thông điệp  $(L_i, p_i)$  và
  - đợi cho đến khi nhận được trả lời từ tất cả các tiến trình khác
2. Nếu một tiến trình ở trạng thái *Thả*, nó lập tức trả lời tất cả các yêu cầu vào critical section
3. Nếu một tiến trình ở trạng thái *Giữ*, nó không trả lời cho đến khi hoàn tất critical section
4. Nếu một tiến trình ở trạng thái *Muốn*, nó lập tức trả lời yêu cầu nếu timestamp của yêu cầu đó **nhỏ hơn** timestamp của yêu cầu của chính nó

# PP 3: Giải thuật phân tán



## ➤ Tính chất:

- Multicast dẫn đến chi phí cao về đường truyền
- Dễ thất bại do lỗi



# Các giải thuật bầu chọn

- Xác định một điều phối viên (coordinator)
  - Một số thuật toán dựa vào một tiến trình điều phối riêng biệt
  - Điều phối viên phải được cùng quyết định
  - Cũng có thể thay điều phối viên trong khi đang chạy
  - Giả sử mỗi nút đều có ID khác nhau
  - Bầu cử: thống nhất về tiến trình đang chạy có ID lớn nhất
- Đòi hỏi:
  - một tiến trình hoặc không biết điều phối viên hoặc biết ID của tiến trình có ID lớn nhất
  - một tiến trình rồi cũng sẽ hoặc treo hoặc biết điều phối viên là tiến trình nào

# Thuật toán Lớn trị bé – Bully algorithm

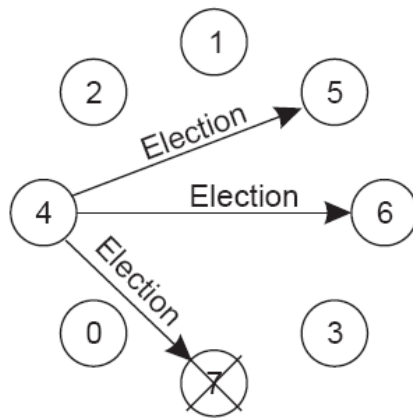
## ➤ Ba loại thông điệp:

- **Election**: phát động bầu cử
- **Answer**: hồi đáp thông báo bầu cử
- **Coordinator**: công bố điều phối viên đã được bầu

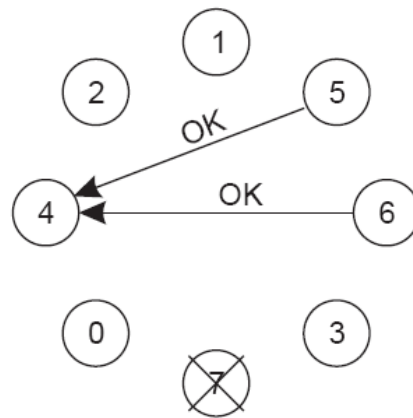
## ➤ Thuật toán:

- Một tiến trình phát động bầu cử khi nó
  - thấy điều phối viên có vẻ đã hỏng (timeout), hoặc
  - nhận được một thông điệp Election.
- Khi nhận được Election, tiến trình trả lời bằng Answer
- Tiến trình phát động bầu cử bằng cách gửi Election tới các tiến trình **CÓ ID cao hơn** mình rồi đợi nhận Answer từ các tiến trình đó
- Nếu không nhận được Answer, tiến trình trở thành điều phối viên mới và gửi Coordinator tới tất cả các tiến trình khác để thông báo kết quả
- Nếu nhận được Answer, tiến trình đợi Coordinator để biết kết quả bầu điều phối viên mới

# Thuật toán Lớn trị bé

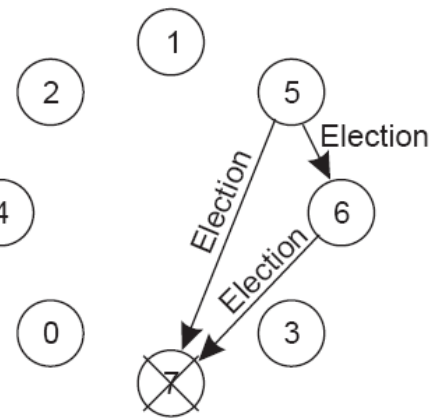


(a)

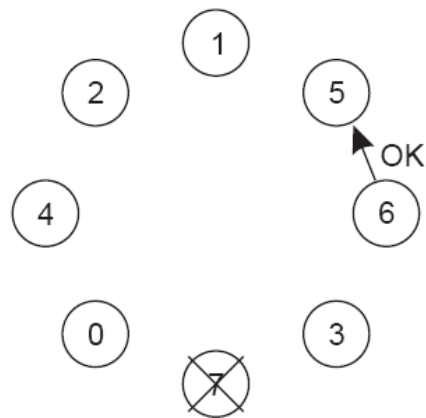


Previous coordinator  
has crashed

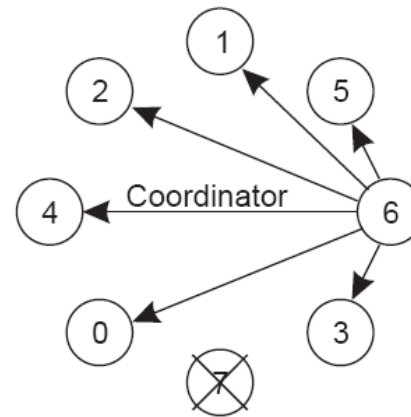
(b)



(c)



(d)



(e)

# Thuật toán vòng

- Các nút sắp xếp thành vòng (logic)
- Một nút phát động bầu cử bằng cách gửi Election tới nút tiếp theo. Nếu nút đó đang hỏng, thông điệp được chuyển cho nút tiếp theo sau đó
- Tiến trình gắn ID của mình vào thông điệp trước khi gửi đi. Khi thông điệp quay lại tiến trình phát động bầu cử, tất cả các nút đều đã tham gia bầu cử.
- Nút phát động gửi thông điệp Coordinator chứa danh sách các tiến trình đang hoạt động quanh vòng. Tiến trình có ID cao nhất trong đó được chọn làm điều phối viên

