

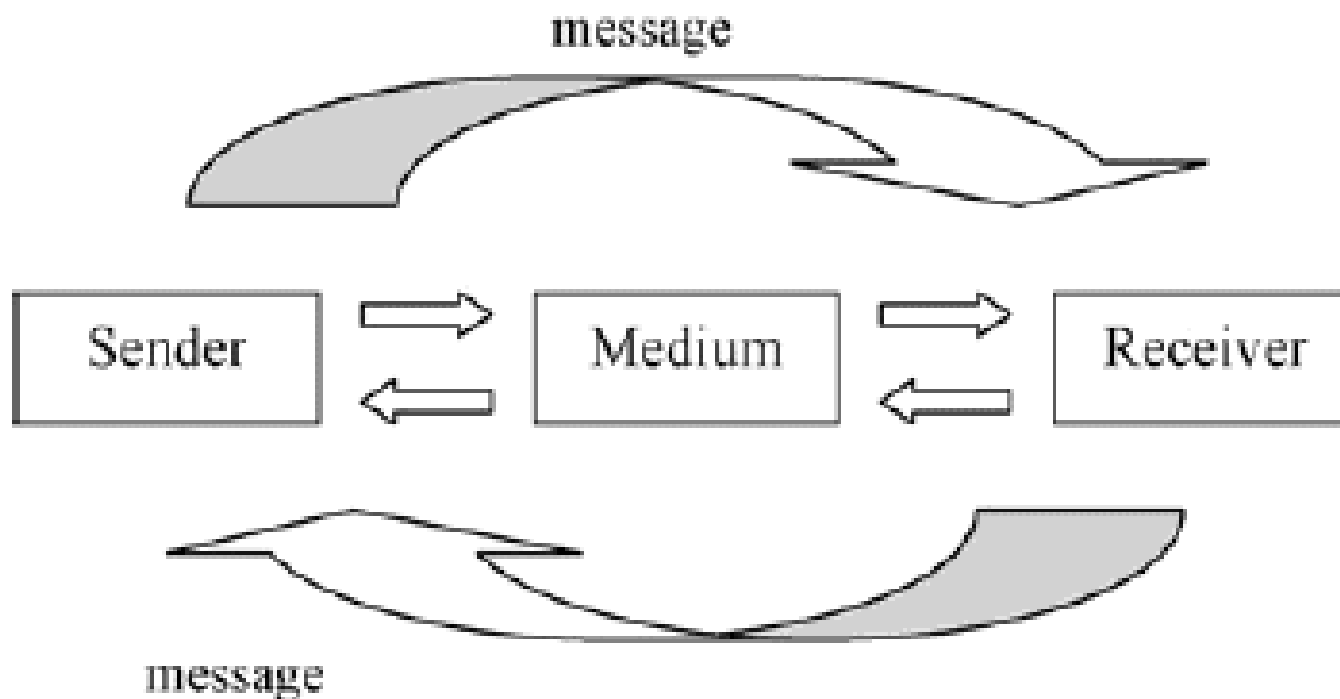
### 3.3 Liên lạc giữa các tiến trình (The Process Communication).

#### Mục đích :

Trao đổi thông tin.

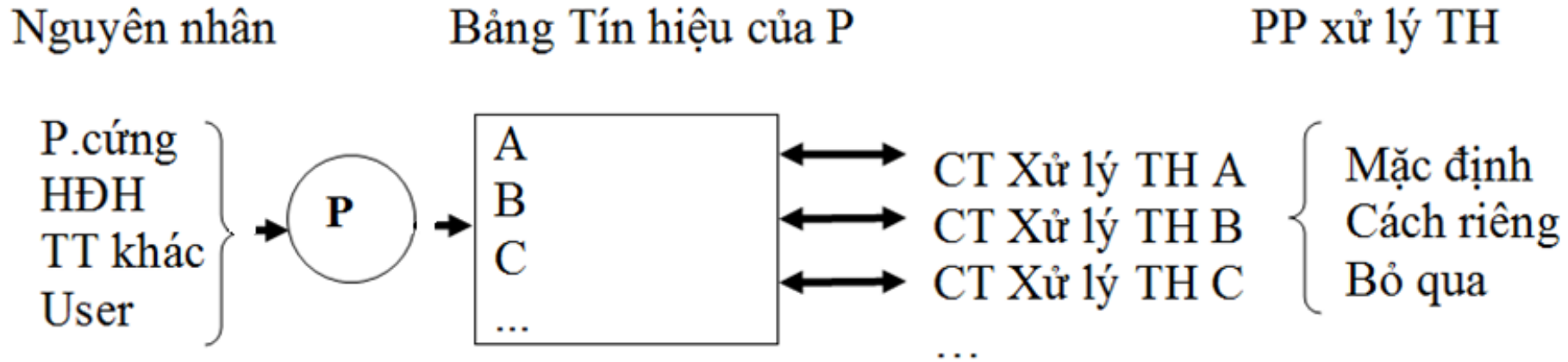
Hợp tác hoàn thành tác vụ.

Thông báo của hệ thống (phần cứng, HĐH).



# 1. Bảng tín hiệu.

Mỗi TT sở hữu 1 bảng tín hiệu. Liên lạc thông qua tín hiệu qui ước

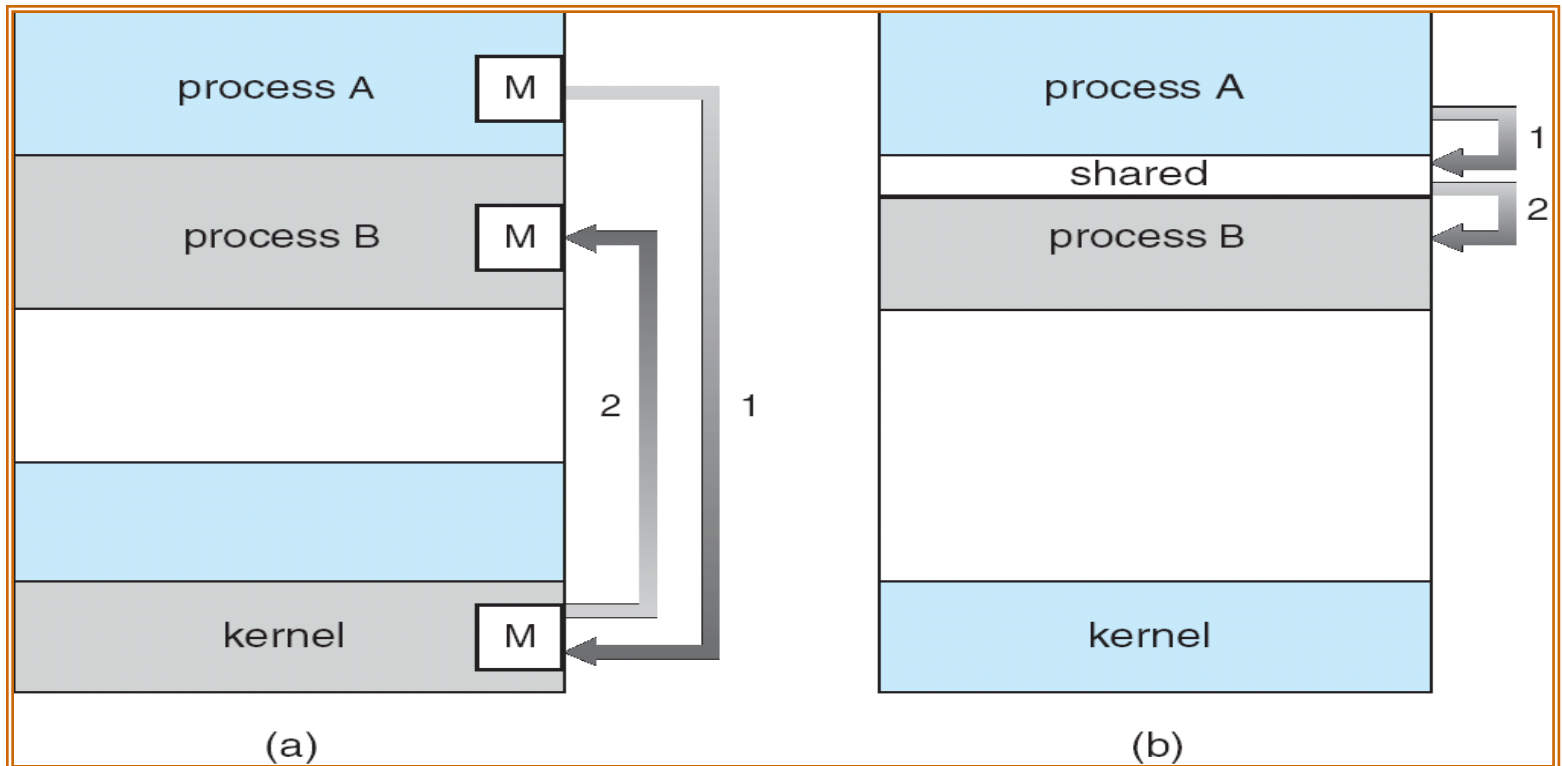


VD : các CT trong MS DOS thường kết thúc khi bấm tổ hợp phím Ctrl+Break.

## 2. Truyền thông điệp (IPC InterProcess Communication)

- Tiến trình truyền tin và đồng bộ hoạt động.
- Hệ thống thông báo (message) .
- Cung cấp 2 hoạt động :
  - Send(message).
  - Receive(message).
- Điều kiện để 2 TT truyền thông điệp.

Thiết lập 1 liên kết (bộ nhớ dùng chung, bus, thuộc tính logic, ...)  
Trao đổi thông báo trên cơ sở Send() và Receive()



# Các hình thức truyền thông điệp

## a. Trực tiếp

- **Xác định rõ TT trong gửi/nhận thông điệp.**  
Send(P1,message) – gửi cho P1.  
Receive(P2,message) – nhận từ P2.
- **Đặc điểm.**  
Liên kết giữa 2 TT được thiết lập tự động.  
Giữa 2 TT chỉ có duy nhất 1 liên kết.  
Liên kết có thể là gián tiếp nhưng thường là 2 chiều

## **b. Gián tiếp**

- **Thông báo nhận trực tiếp từ hộp thư (hoặc Port)**  
Hộp thư được định danh duy nhất.  
TT chỉ có thể truyền tin cho nhau nếu hộp thư là dùng chung.
- **Hoạt động**  
Tạo hộp thư.  
Hủy hộp thư.  
Gửi/nhận thông qua hộp thư.
- **Xác định rõ hộp thư trong gửi/nhận thông điệp.**  
send(A, message) – gửi thông báo cho hộp thư A  
receive(A, message) – nhận thông báo từ hộp thư A
- **Đặc điểm**  
Liên kết chỉ thiết lập khi TT có chung hộp thư.  
Một liên kết có thể kết hợp với nhiều TT.  
Một cặp TT có thể chia sẻ các liên kết truyền tin.  
Liên kết có thể là gián tiếp hoặc 2 chiều.
- **Vấn đề đối với hộp thư dùng chung (P1, P2, P3).**  
P1 gửi, P2 và P3 nhận được thông báo.  
Biện pháp giải quyết :  
Chỉ cho phép 1 TT nhận thông báo.  
Khi gửi thông báo có xác định TT nhận.  
Hệ thống độc quyền chọn TT nhận thông báo

# Các dạng truyền thông điệp

## a. Đồng bộ

**Thông báo dạng khối (chuẩn hóa)**

### **Hoạt động**

Blocking Send : gửi TB cho đến khi kết thúc.

Blocking Receive : nhận TB cho đến khi xong.

### **Đặc điểm**

TB gửi/nhận hoàn chỉnh.

## **b. Bất đồng bộ**

**Thông báo không dạng khối (không chuẩn hóa)**

### **Hoạt động**

Non Blocking Send : gửi TB và tiếp tục hoạt động.

Non Blocking Receive : nhận TB và tiếp tục hoạt động.

### **Đặc điểm**

TB gửi/nhận không hoàn chỉnh hoặc rỗng.

# Vấn đề Buffering

**Các TT liên kết thông qua Buffer để gửi/nhận TB.**

**Các dạng gửi/nhận**

**Buffer 0** : gửi nhận trực tiếp.

**Buffer hữu hạn** :

Send phải chờ khi Buffer rỗng.

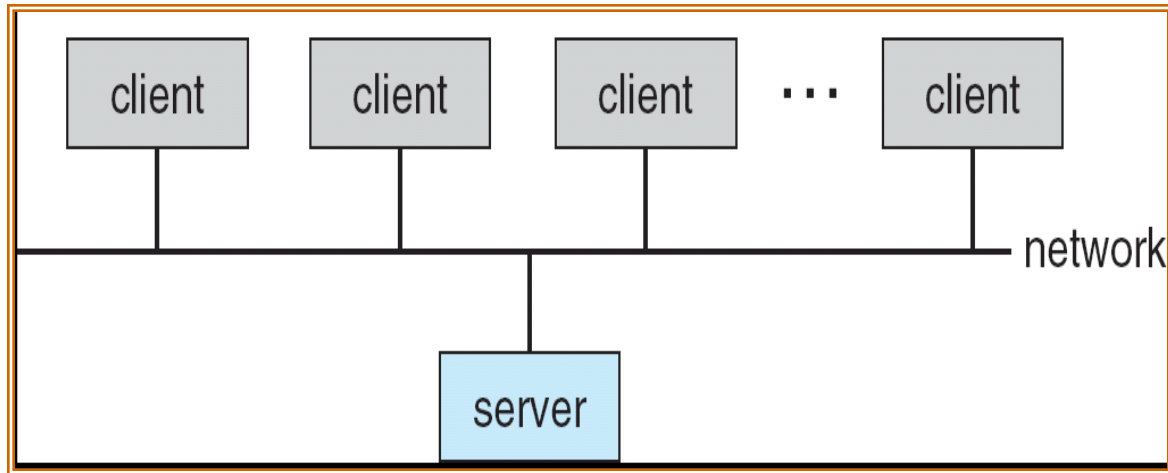
Receive phải chờ khi Buffer đầy.

**Buffer vô hạn** : Send/Receive không cần phải đợi.



### 3. Liên lạc trong hệ thống Client - Server

Cấu trúc mạng Client - Server



Các hình thức

**Socket**

**RPC (Remote Procedure Call)**

**RMI (Remote Method Invocation)**

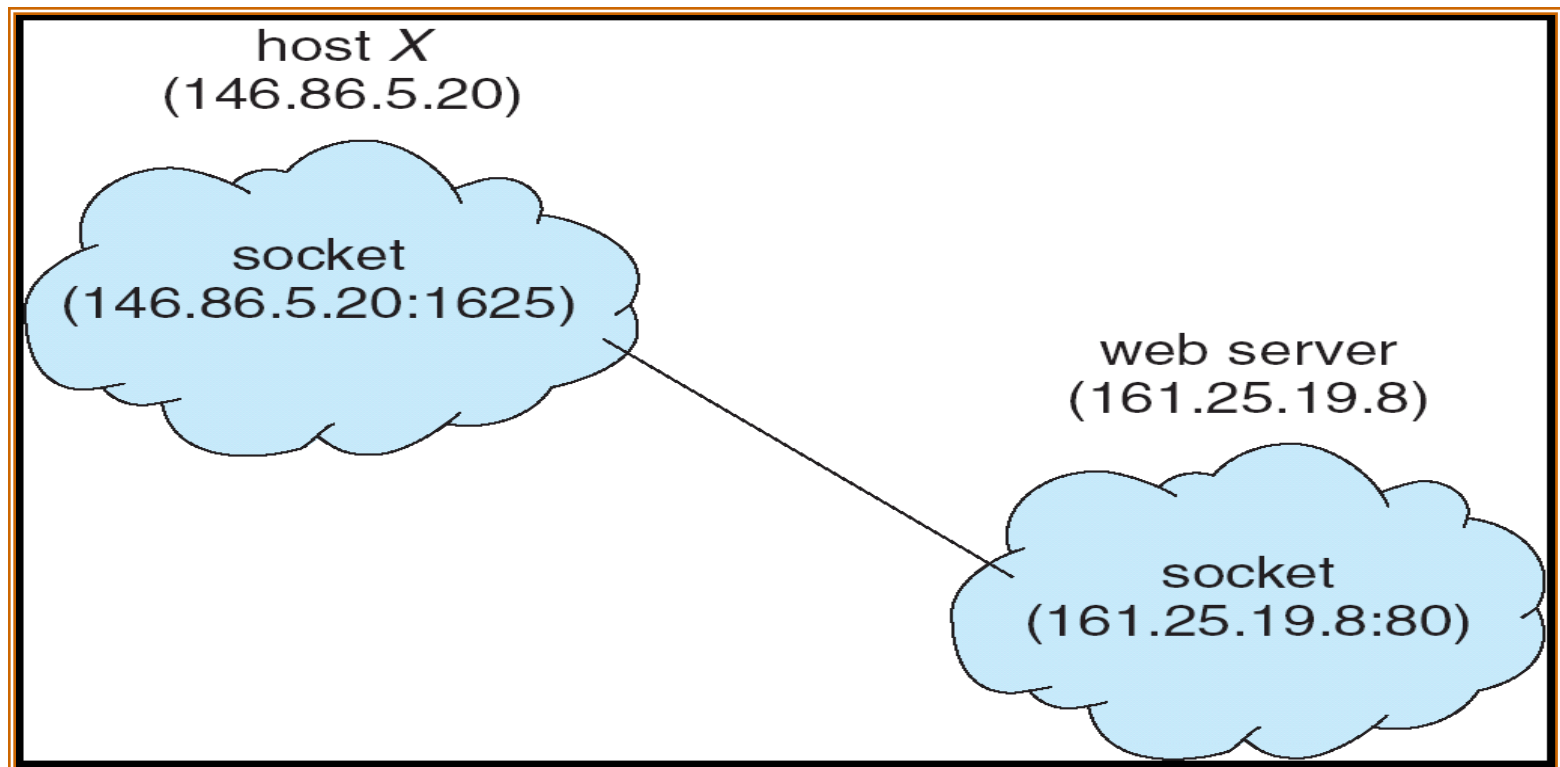
# Socket

Cơ chế phần mềm (lập trình)

Điểm cuối truyền tin sử dụng TCP/IP

Cấu trúc Socket : IP:Port (16 bit)

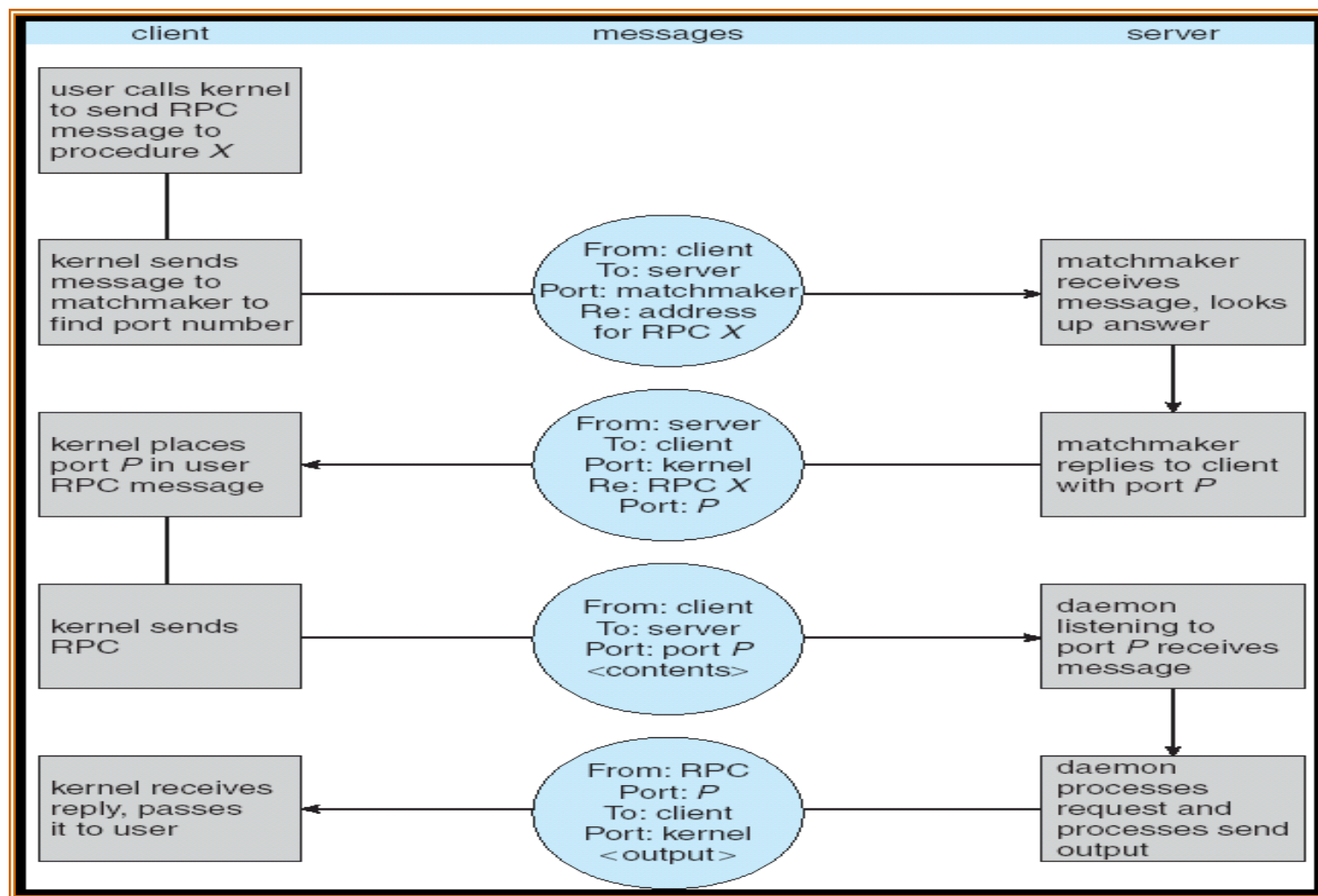
TT liên lạc thông qua 1 cặp Socket



# RPC

RPC Remote Procedure Call : gọi thủ tục từ xa, thực hiện giữa các TT trên mạng sử dụng TCP/IP

Cách liên lạc :



# RPC

VD :

Gs TT Client yêu cầu thực hiện Procedure X (RPC X) trên Server.

## Client

Yêu cầu tạo liên kết cho RPC X

Gởi tham số cho RPC X

Nhận KQ và chuyển cho TT Client

## Server

Nhận yêu cầu và tạo liên kết

Nhận tham số

Thực thi Procedure X

Gởi kết quả cho RPC X

VD : hoạt động của virus Sars tắt MT sau 60s (svchost.exe)

# RMI

RMI Remote Method Invocation (Java)

Hoạt động tương tự như RPC trong Windows XP

Cho phép CT Java gọi 1 phương thức (method) từ

1 Remote Object (đối tượng từ xa - ở máy khác)

