西安交通大学
软件学院

# 操作系统原理

## Operating System Principle

田丽华

# 3-8 消息传递

# Message passing system
# 消息传递

## If P and Q wish to communicate, they need to 若P与Q要通信，需要:

establish a *communication link* between them 建立通信连接

exchange messages via send/receive 通过send/receive交换消息

## Implementation of communication link 通信连接的实现

physical (e.g., shared memory, hardware bus)物理的
（如，共享存储，硬件总线）

logical (e.g., logical properties)逻辑的（如，逻辑特性）

# Implementation Questions
# 实现中的问题

1. How are links established?
连接如何建立?

2. Can a link be associated with more than two processes?
连接可同多于两个的进程相关吗?

3. How many links can there be between every pair of communicating processes?
每对通信进程有多少连接?

# Implementation Questions
# 实现中的问题

4. at is the capacity of a link?
一个连接的容量是多少?

5. Is the size of a message that the link can accommodate fixed or variable? 连接可使用的固定或可变消息的大小?

6. Is a link unidirectional or bi-directional? 连接是无向的还是双向的?

# Direct Communication
# 直接通信

1

Processes must name each other explicitly:
进程必须显式的命名

- **send** (*P, message*) – send a message to process P向进程P发消息
- **receive**(*Q, message*) – receive a message from process Q从进程Q收消息

2

Properties of communication link 通信连接的特性

- Links are established automatically. 连接自动建立
- A link is associated with exactly one pair of communicating processes.
  连接精确的与一对在通信的进程相关
- Between each pair there exists exactly one link.
  在每一对之间就存在一个连接
- The link may be unidirectional, but is usually bi-directional.
  连接可以无向，但通常是双向的

# Direct Communication
# 直接通信

01 asymmetric communication

02 Sender names the recipient, the recipient is not required to name the sender.

03 Disadvantage: the limited modularity

send (P, message) – send a message to process P向进程P发消息

receive(id, message) – receive a message from process Q从进程Q收消息

# Indirect Communication
# 间接通信

1. Messages are directed and received from mailboxes (also referred to as ports).消息导向至信箱并从信箱接收（被视作端口）

   - Each mailbox has a unique id. 每一个信箱有一个唯一的id
   - Processes can communicate only if they share a mailbox. 仅当共享一个信箱时进程才能通信

2. Primitives are defined as

   send(A, message) – send a message to mailbox A
   receive(A, message) – receive a message from mailbox A

# Indirect Communication
# 间接通信

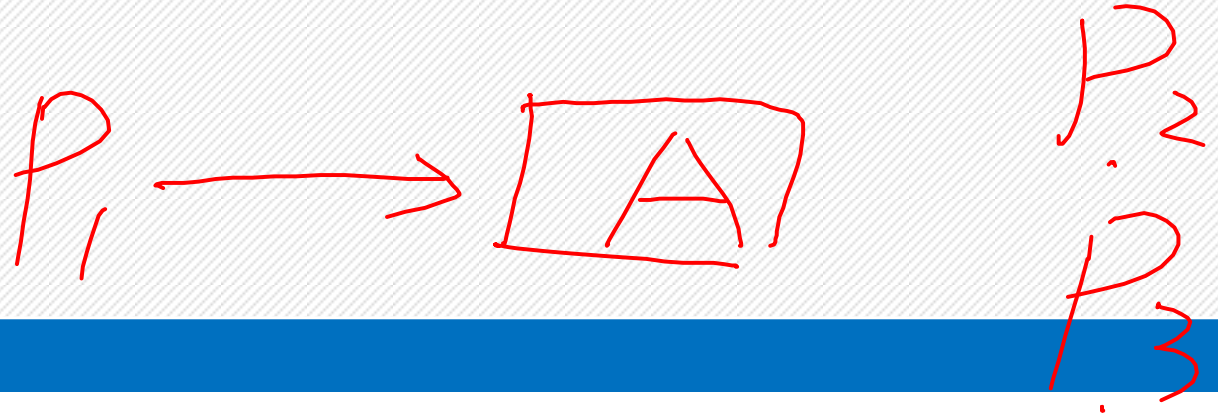
## 3. Properties of communication link 通信连接的特性

- Link established only if processes share a common mailbox
  仅当进程共有一个信箱时连接才能建立
- A link may be associated with many processes.连接可同多个进程相关
- Each pair of processes may share several communication links.
  每一对进程可共享多个通信连接
- Link may be unidirectional or bi-directional.连接可是无向或双向的

## 4. Operations操作

- create a new mailbox 创建新的信箱
- send and receive messages through mailbox 通过信箱发送和接收消息
- destroy a mailbox 销毁信箱

# Indirect Communication
# 间接通信




## Mailbox sharing 信箱共享

- P1, P2, and P3 share mailbox A. P1, P2与P3共享信箱A
- P1, sends; P2 and P3 receive. P1发送； P2与P3接受
- Who gets the message?谁得到消息?

# Indirect Communication
# 间接通信

Solutions 解决方案

1. Allow a link to be associated with at most two processes.
允许一个连接最多同2个进程相关

2. Allow only one process at a time to execute a receive operation.
只允许一个时刻有一个进程执行接受操作

3. Allow the system to select arbitrarily the receiver. Sender is notified who the receiver was.
允许系统任意选择接收者。发送者被通知谁是接收者。

# Synchronization
# 同步

1

**Message passing may be either blocking or non-blocking**

2

**Blocking is considered synchronous**

- Blocking send has the sender block until the message is received
- Blocking receive has the receiver block until a message is available

3

**Non-blocking is considered asynchronous**

- Non-blocking send has the sender send the message and continue
- Non-blocking receive has the receiver receive a valid message or null

# Buffering
# 缓冲

Queue of messages attached to the link; implemented in one of three ways.

消息队列附加在连接上；有以下三种实现方案

01 Zero capacity – 0 messages零容量 - 0 消息

Sender must wait for receiver (rendezvous).发送者必须等待接收者

02 Bounded capacity – finite length of *n* messages有界容量 - *n*个消息有限长度

Sender must wait if link full.若连接满了发送者必须等待

03 Unbounded capacity – infinite length无界容量 - 无限长度

Sender never waits.发送者从不等待