西安交通大学

软件学院

操作系统原理

Operating System Principle

田丽华

3-7　进程通信

Cooperating Processes
协同进程

- ☐ Independent process cannot affect or be affected by the execution of another process.
  独立进程不会影响另一个进程的执行或被另一个进程执行影响
- ☐ Cooperating process can affect or be affected by the execution of another process
  协同进程可能影响另一个进程的执行或被另一个进程执行影响
- ☐ Advantages of process cooperation
  进程协同的优点
  - Information sharing 信息共享
  - Computation speed-up 加速运算
  - Modularity 模块化
  - Convenience 方便

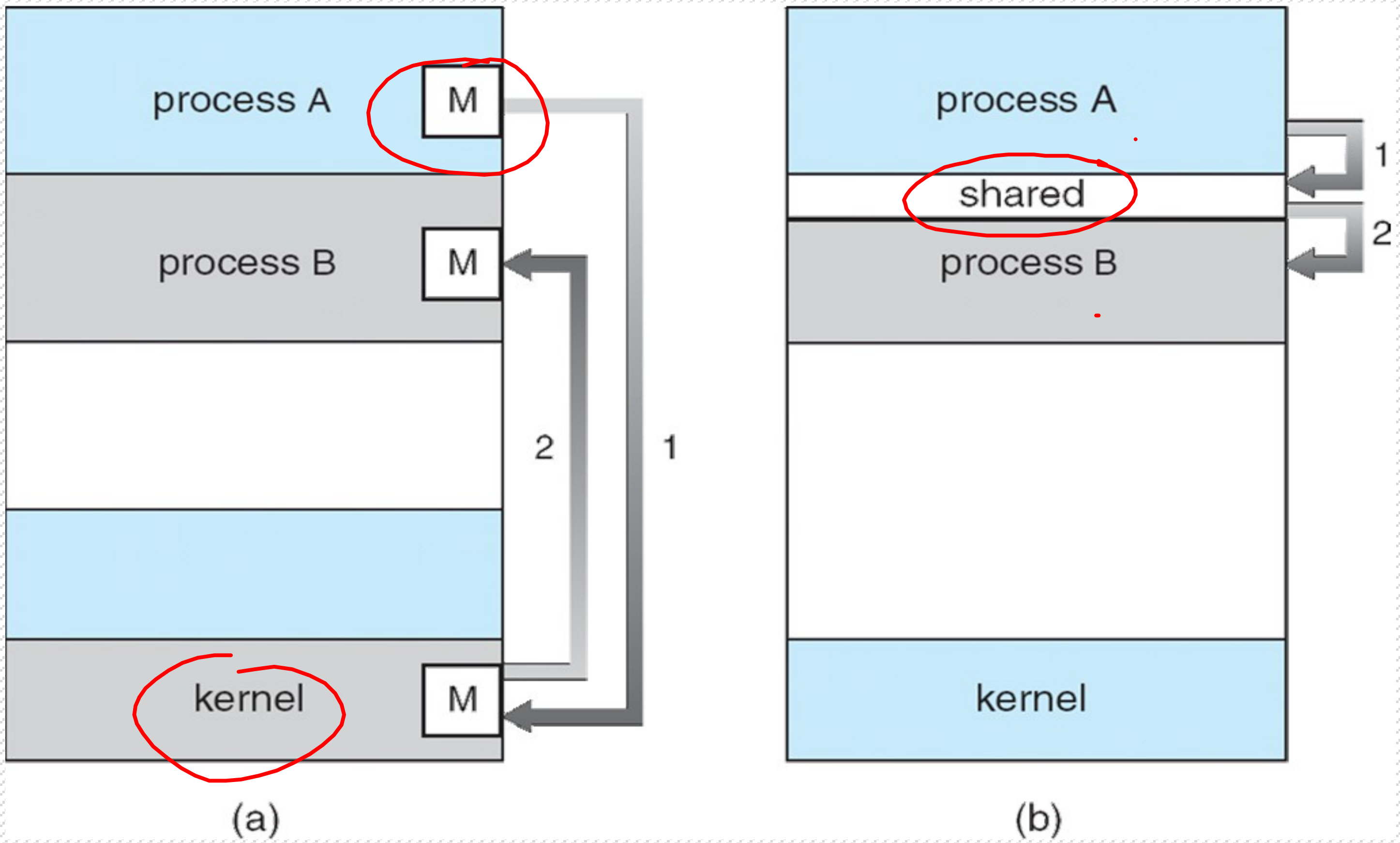
Interprocess Communication (IPC)
进程间通信

Cooperating processes need interprocess communication (IPC) to exchange data and information

Two models of IPC

- Shared memory
- Message passing

Communication Models
通信模型



Shared Memory systems
共享存储

01 Interprocess communication using Shared memory requires communicating processes to establish a shared memory
使用共享存储模型的进程间通信要建立共享存储区

02 Processes can exchange information by reading or writing data to the shared areas.
进程通过读写共享存储区来交换信息

03 The form of the exchanged data and location are determined by the communicating processes and are not under the OS's control
由通信进程来确定交换的数据和位置，不受操作系统的控制

Message passing system
消息传递

1. Mechanism for processes to communicate and to synchronize their actions. 用于进程通信的机制，同步其间的活动

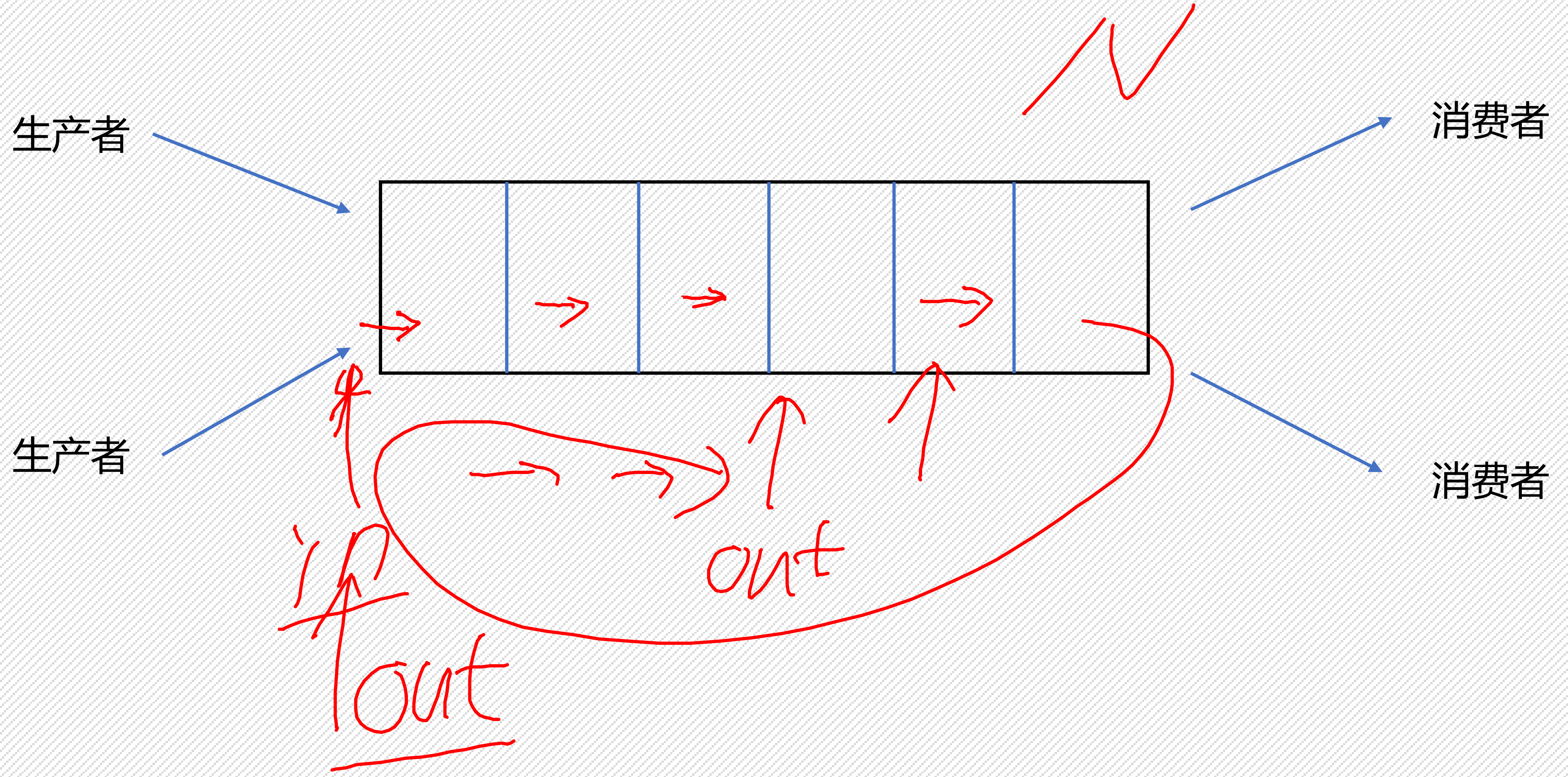
2. Message system – processes communicate with each other without resorting to shared variables. 消息系统 - 进程间通信无须再利用共享变量

3. IPC facility provides two operations IPC提供两个操作：
   - send(message) – message size fixed or variable 发送 - 固定或可变大小消息
   - receive(message)接收

Producer-Consumer Problem

生产者-消费者问题

Paradigm for cooperating processes, producer process produces information that is consumed by a consumer process. **生产者进程生产供消费者进程消费的信息**



Bounded-Buffer – Shared-Memory Solution

Shared data ❯

```
var n;
type item = … ;
var buffer. array [0..n–1] of item;
    in, out: 0..n–1;
```

Producer process ❯

```
repeat
    …
    produce an item in nextp
    …
    while in+1 mod n = out do no-op;
    buffer [in] :=nextp;
    in :=in+1 mod n;
until false;
```

Bounded-Buffer (Cont.) Shared-Memory Solution

Consumer process

```
repeat
    while in = out do no-op;
    nextc := buffer [out];
    out := out+1 mod n;
              …
    consume the item in nextc
              …
    until false;
```

Solution is correct, but can only fill up n–1 buffer.