西安交通大学
软件学院

操作系统原理

Operating System Principle

田丽华

7-2 死锁的必要条件

死锁的原因

竞争资源
引起死锁

进程推进顺序
不当引起死锁

Deadlock Necessary Conditions
死锁的必要条件

Deadlock can arise if four conditions hold simultaneously.

（四个条件同时出现，死锁将会发生）

- **Mutual exclusion**: only one process at a time can use a resource.（互斥：一次只有一个进程可以使用一个资源）
- **Hold and wait**: a process holding at least one resource is waiting to acquire additional resources held by other processes.

（占有并等待：一个至少持有一个资源的进程等待获得额外的由其他进程所持有的资源）(请求与保持)

Deadlock Necessary Conditions
死锁的必要条件

- **No preemption**: a resource can be released only voluntarily by the process holding it, after that process has completed its task.
  (不可抢占：一个资源只有当持有它的进程完成任务后，自由的释放)
  (非剥夺)
- **Circular wait**: there exists a set $\{P_0, P_1, \ldots, P_0\}$ of waiting processes such that P_0 is waiting for a resource that is held by P_1, P_1 is waiting for a resource that is held by P_2, …, P_{n-1} is waiting for a resource that is held by P_n, and P_0 is waiting for a resource that is held by P_0.
  (循环等待：等待资源的进程之间存在环)

System Model
系统模型

- Resource types （资源类型）$R_1, R_2, \ldots, R_m$

  CPU cycles, memory space, I/O devices

  (CPU周期，内存空间，I/O设备)

- Each resource type R_i has W_i instances.

  (每一种资源R_i有W_i种实例)

- Process Pi （进程Pi）

Resource-Allocation Graph
资源分配图

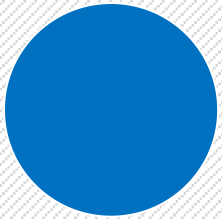
A set of vertices V and a set of edges E.（一组顶点的集合V和边的集合E）

- **V is partitioned into two types:（V被分为两个部分）**
  - $P = \{P_1, P_2, \ldots, P_n\}$, **the set consisting of all the processes in the system.**（**P：**含有系统中全部的进程）
  - $R = \{R_1, R_2, \ldots, R_m\}$, **the set consisting of all resource types in the system.**（**R：**含有系统中全部的资源）
- **request edge – directed edge** $P_i \rightarrow R_j$（请求边：直接$P_i \rightarrow R_j$）
- **assignment edge – directed edge** $R_j \rightarrow P_i$ 分配边： $R_j \rightarrow P_i$ ）

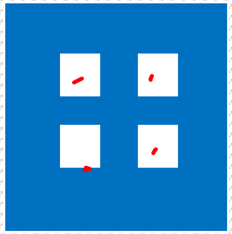
Resource-Allocation Graph

资源分配图

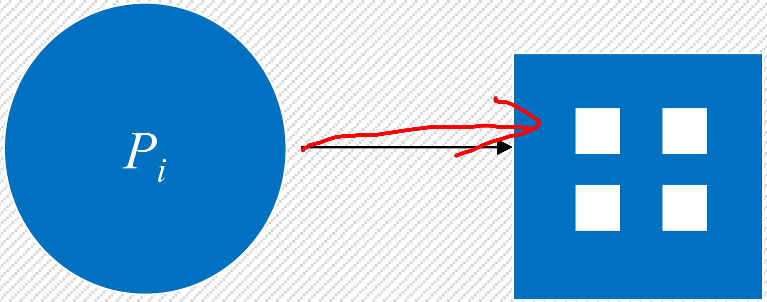
- **Process进程**



- **Resource Type with 4 instances有四个实例的资源类型**

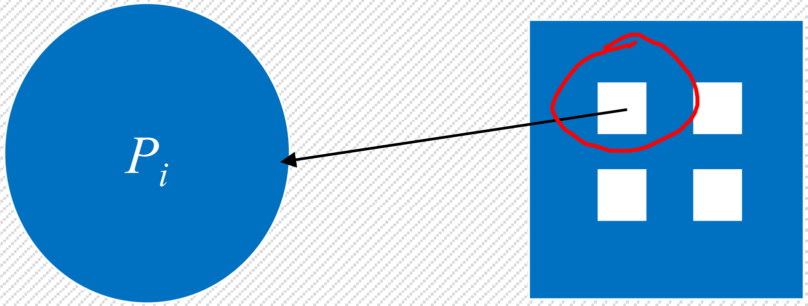


***P*$_i$ requests instance of *R*$_j$ （*P*$_i$ 请求一个*R*$_j$的实例）**



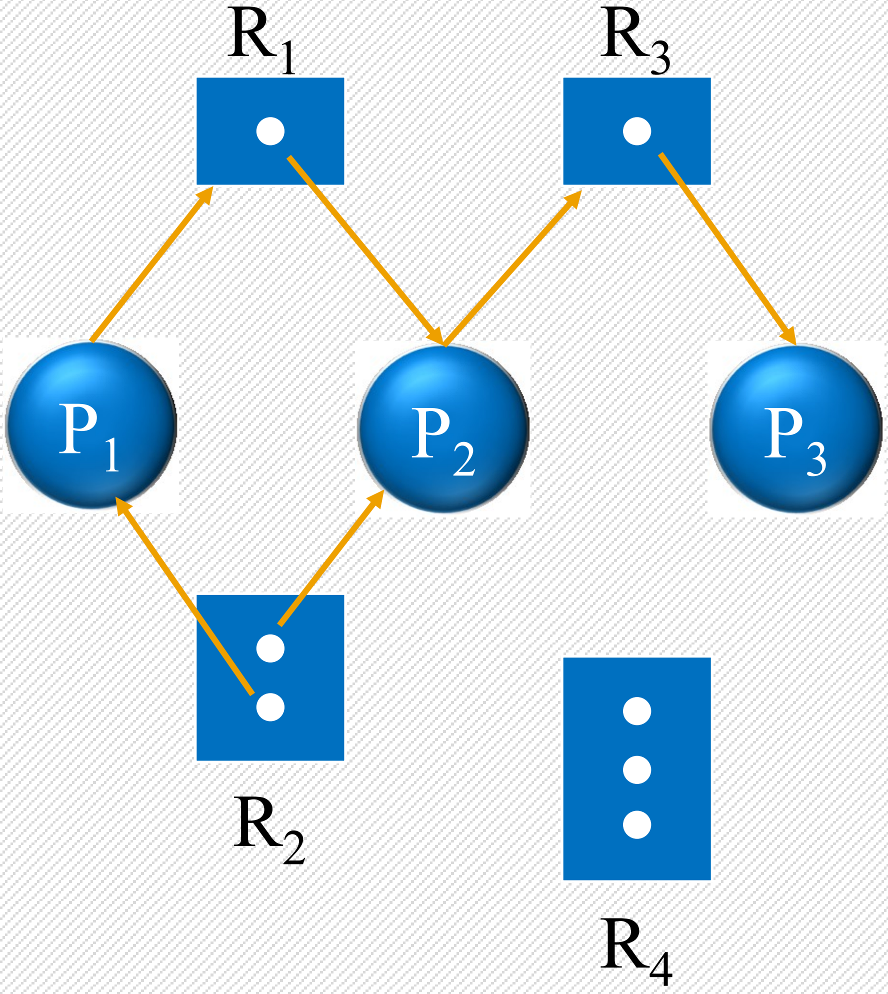
R_j

***P*$_i$ is holding an instance of *R*$_j$（*P*$_i$ 持有一个*R*$_j$的实例）**



R_j

Example of a Resource Allocation Graph
资源分配图的例子



Basic Facts
基本事实

如果

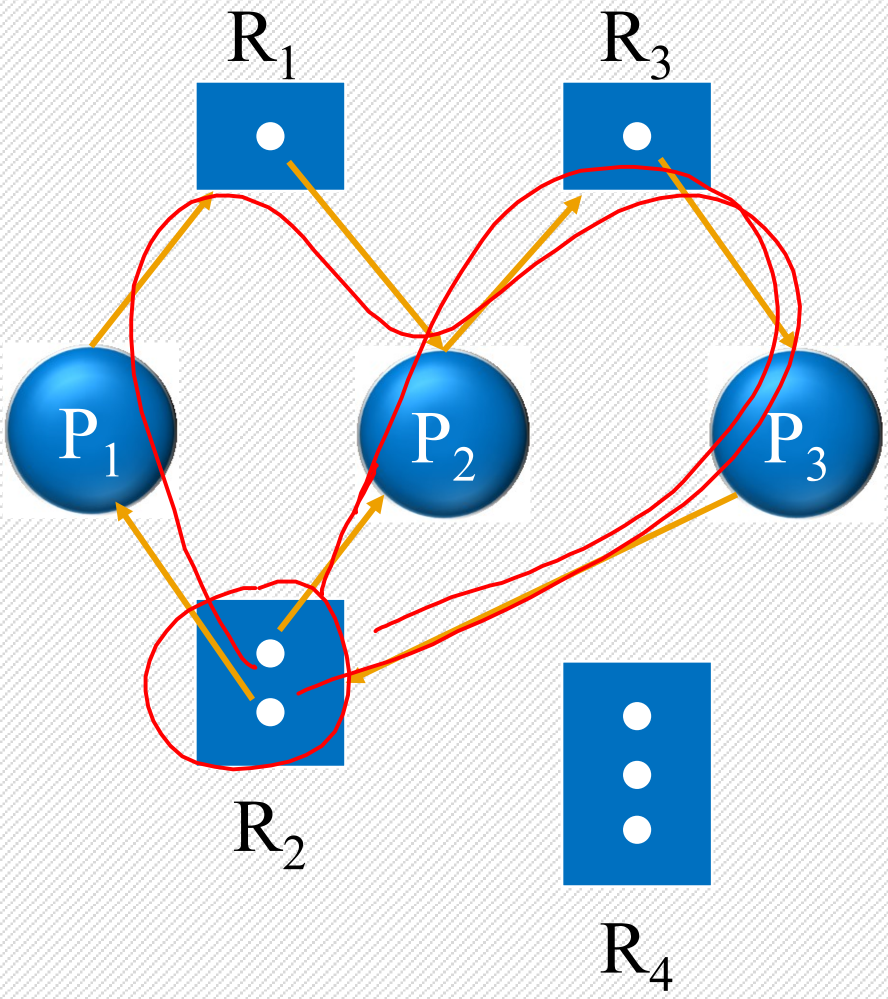
If graph contains no cycles ⇒ no deadlock.
（如果图没有环，那么不会有死锁）

如果

If graph contains a cycle ⇒ （如果图有环）

- if only one instance per resource type, then deadlock.
  (如果每一种资源类型只有一个实例，那么死锁发生)
- if several instances per resource type, possibility of deadlock.
  (如果每种资源类型有多个实例，可能死锁)

Resource Allocation Graph With A Deadlock



Resource Allocation Graph With A Cycle But No Deadlock