

2.4.4 死锁检测和恢复

1. 资源分配图算法
2. 资源矩阵法
3. 死锁的解除与系统恢复

恢复死锁常用的方法有如下几种：

- (1) 资源剥夺法：挂起某些死锁进程，并抢占它的资源。
- (2) 进程撤销法：通过撤销占有资源多的进程或代价量小的进程，以恢复死锁。
- (3) 进程回退法：设置还原点，让一个或多个进程回退到足以解除死锁的地步。
- (4) 重新启动系统：代价最大，一切从头开始。我们要尽量避免采用此方法。

2.4.5 死锁避免

1. 安全与不安全状态

某一时刻，系统能按某种顺序为每个进程分配其所需资源，使每个进程都能顺利地地完成，则称此时系统处于安全状态。反之，称之为不安全状态。

2. 银行家算法

银行家算法问题描述是：一个银行家把他的固定资金借给若干顾客，使这些顾客能满足对资金的要求又能完成其交易，也使银行家可以收回全部的现金。只要不出现一个顾客借走所有资金后还不够、还需要借贷，则银行家的资金应是安全的。

表 2-1 资源分配表

进 程	已分配资源			尚需资源			可用资源		
	R 1	R 2	R 3	R 1	R 2	R 3	R 1	R 2	R 3
P ₁	2	0	0	0	0	1	0	2	1
P ₂	1	2	0	1	3	2			
P ₃	0	1	1	1	3	1			
P ₄	0	0	1	2	0	0			

经过分析。

表2-2 用银行家算法进行资源分配

进 程	已分配资源			尚需资源			可用资源		
	R 1	R 2	R 3	R 1	R 2	R 3	R 1	R 2	R 3
P	2	0	0	0	0	1	0	2	1

1									
P ₄	0	0	1	2	0	0	2	2	1
P ₂	1	2	0	1	3	2	2	2	2
P ₃	0	1	1	1	3	1			

此时的安全序列不存在，故处于不安全状态。

2.4.6 死锁预防

所谓死锁预防，就是采用某种策略，限制并发进程对资源的请求，使系统在任何时刻都不满足死锁的四个必要条件。死锁预防主要是针对破坏四个必要条件进行的。

破坏互斥条件：某些设备可以通过 SPOOLING 系统将独享设备改造成为共享设备，以此可以解决互斥问题，例如打印机。

破坏非剥夺条件：资源暂时释放策略，申请新的资源得不到满足则暂时释放已有的资源。

破坏占用并请求条件：一次性申请全部资源。

破坏循环等待条件：资源有序申请，给资源编号，使用时按升序进行。