



软件设计师

--操作系统基础知识

高级项目经理 任铄

QQ: 1530841586

第三章 操作系统基础知识

- 3.1 操作系统功能、类型和层次结构
- 3.2 处理机管理（进程管理）
- 3.3 存储管理
- 3.4 设备管理
- 3.5 文件管理
- 3.6 作业管理
- 3.7 嵌入式操作系统

一、死锁

是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种状态时，若无外力作用，它们都将无法再向前推进。

高校教师、高级项目经理 任铄

QQ: 1530841586

向上人生路!

1、产生死锁的原因

(1)竞争资源

当系统中供多个进程共享的资源如打印机、公用队列等，其数目不足以满足诸进程的需要时，会引起诸进程对资源的竞争而产生死锁。

(2)进程间推进顺序非法

进程在运行过程中，请求和释放资源的顺序不当，也同样会导致产生死锁。

(1)竞争资源引起进程死锁

高校教师、高级项目经理 任铎

QQ: 1530841586

可把系统中的资源分为两类：

a.可剥夺性资源：资源分配给进程后可以被高优先级的进程剥夺。如CPU、主存。

b.不可剥夺性资源：分配给进程后只能在进程用完后才释放的资源。如磁带机、打印机等。

2、产生死锁的必要条件

- (1)互斥条件
- (2)请求和保持条件
- (3)不剥夺条件
- (4)环路等待条件

高校教师、高级项目经理 任铄

QQ: 1530841586

(1)互斥条件

进程访问的是临界资源，即在一段时间内某资源只由一个进程占用。如果此时还有其他进程请求该资源，则请求者只能等待，直至占有该资源的进程用完释放。

(2)请求和保持条件

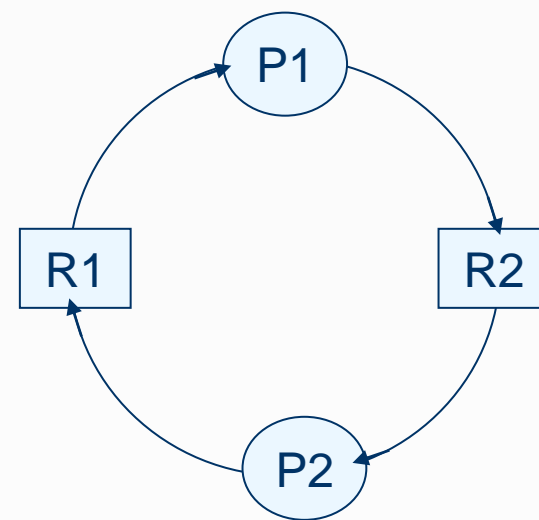
一进程在请求新的资源的同时，保持对已分配资源的占有。

(3)不剥夺条件

指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。

(4)环路等待条件

指在发生死锁时，必然存在一个进程--资源的环形链。即进程集合 $\{P_0, P_1, P_2, \dots, P_n\}$ 中的 P_0 正在等待一个 P_1 占用的资源； P_1 正在等待一个 P_2 占用的资源，……， P_n 正在等待一个已被 P_0 占用的资源。



例：

某系统中有3个并发进程，都需要同类资源4个，试问该系统不会发生死锁的最少资源数是（ B ）

A、 9 B、 10 C、 11 D、 12

高校教师、高级项目经理 任铎

QQ: 1530841586

向上人生路！

例：

假设某计算机系统中资源R的可用数为6，系统中有3个进程竞争R，且每个进程都需要 i 个R，该系统可能会发生死锁的最小 i 值是（C）。若信号量S的当前值为-2，则R的可用数和等待R的进程数分别为（D）。

A. 1 B. 2 C. 3 D. 4

A. 0、0 B. 0、1 C. 1、0 D. 0、2

高校教师、高级项目经理 任铄

QQ: 1530841586

向上人生路！

3、处理死锁的基本方法

在系统中已经出现死锁后，则应及时检测到死锁的发生，并采取适当措施来解除死锁。

目前处理死锁的方法可归结为四种：

- (1)预防死锁
- (2)避免死锁
- (3)检测死锁
- (4)解除死锁

高校教师、高级项目经理 任铎

QQ: 1530841586

(1)预防死锁

是一种较简单和直观的**事先预防**方法。该方法是通过设置某些限制条件，去破坏产生死锁的四个必要条件的一个或几个，来预防发生死锁。

- 摒弃“请求和保持”条件
- 摒弃“不剥夺”条件
- 摒弃“环路等待”条件

(2)避免死锁

该方法同样是属于**事先预防**的策略，这种方法不是预先加上各种限制条件以预防产生死锁的可能性，而是用某种方法去**防止系统进入不安全状态**，使死锁不致于最终发生。

- 银行家算法避免死锁

高校教师、高级项目经理 任铎

QQ: 1530841586

(3)检测死锁

这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区。

此方法允许系统在运行过程中发生死锁，但可通过系统所设置的检测机构，及时的检测出死锁的发生，并精确的确定与死锁有关的进程和资源；然后采取适当的措施，从系统中将已发生的死锁清除掉。

(4)解除死锁

是与死锁检测相配套的一种措施。当检测到系统中已发生死锁时，须将进程从死锁状态中解脱出来。

- 常用的实施方法是撤销或挂起一些进程，以便回收一些资源，再将 these 资源分配给已处于阻塞状态的进程，使之转为就绪状态，以继续运行。
- 死锁的检测与解除措施，有可能使系统获得较好的资源利用率和吞吐量，但在实现上难度也最大。

可以通过下列渠道沟通联系：

- 1、QQ:1530841586
- 2、QQ群：164955673

向上人生路！