

必要条件

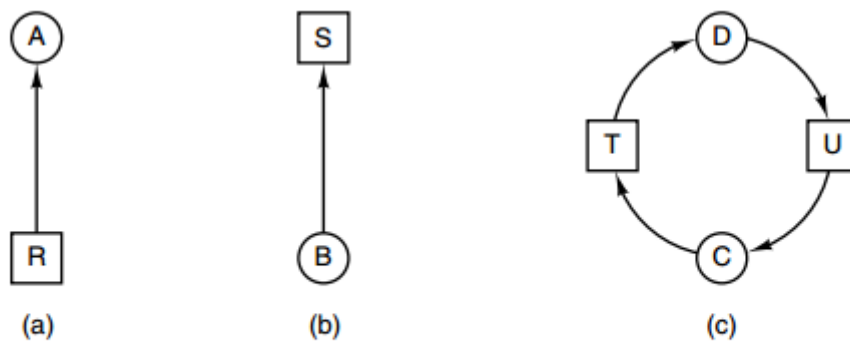


Figure 6-3. Resource allocation graphs. (a) Holding a resource. (b) Requesting a resource. (c) Deadlock.

1. 互斥：每个资源要么已经分配给一个进程，要么就是可以用的；
2. 占有和等待：已经得到了某个资源的进程可以再请求新的资源。
3. 不可抢占：已经分配给一个进程的资源不能强制性的被抢占，他只能被占有她的进程显示释放；
4. 环路等待：有两个或者两个以上的进程组成一条环路，该环路中的每一个进程都等待下一个进程所占有的资源。

处理方法

四种方法：**鸵鸟策略 死锁检测和死锁恢复 死锁预防 死锁避免**

鸵鸟策略

假装什么都没发生，解决死锁的代价很高，使用不采取任何措施的鸵鸟策略可以获得更高的性能；发生死锁的时候，不会对用户造成多大的影响。或者发生死锁的概率很低，可以采取鸵鸟策略；大多数操作系统，包括 Unix, Linux和Windows处理死锁的方式就是忽略他；

死锁检测和死锁恢复

不试图阻止死锁，发生的时候采取措施恢复。

1. 每种类型一个资源的死锁检测

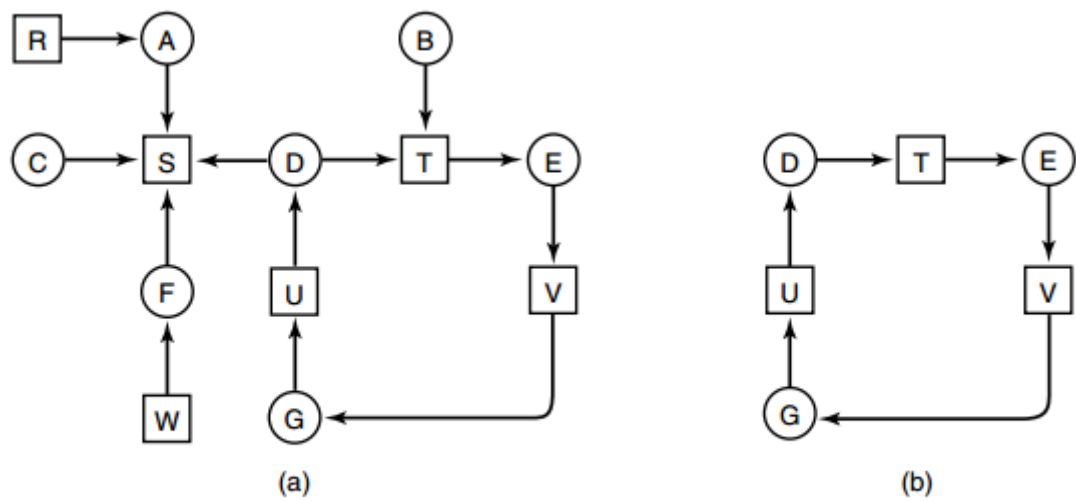


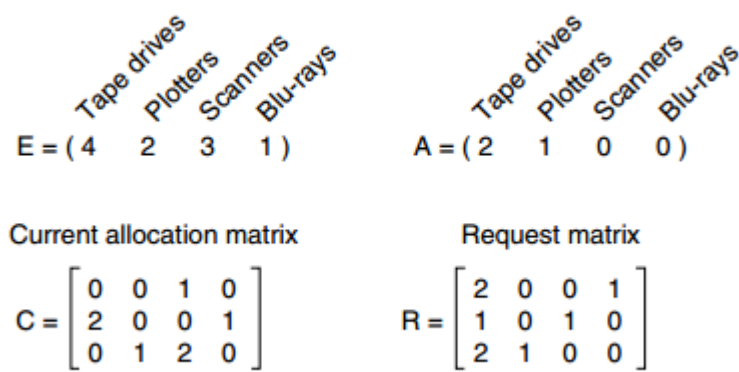
Figure 6-5. (a) A resource graph. (b) A cycle extracted from (a).

方框代表资源，圆圈代表进程。进程只想资源代表请求资源，资源只想进程代表资源已经分配给了进程。

a图可以抽象出b图的环，所以存在环路等待条件，可能会发生死锁。

从一个节点出发深度优先搜索，如果发现已经标记过的节点，表示存在有向图存在环；

2.每种类型多个资源的死锁检测



三个进程四个资源，图中 E: 每种资源的总量 A: 每种资源的剩余量 C: 每个进程拥有的资源 R: 每个进程请求的资源

比较请求资源量R矩阵和剩余资源A，进程3可以被满足，先执行进程3，进程3释放之后， $A=\{2, 2, 2, 0\}$ ；继续比较R矩阵和A，进程2可以满足，执行进程2，最后执行进程1；如果在这样的过程中，R矩阵中的每一行都比剩余资源大，就出现了死锁；

3.死锁恢复

利用抢占恢复 利用回滚恢复 利用杀死进程恢复

死锁预防

破坏死锁的四个必要条件，就可以避免死锁：

- 1. **破坏互斥条件** 假脱机打印机技术允许若干个进程同时输出，唯一真正请求物理打印机的进程是打印机守护进程。

- 2. **破坏占有和等待条件** 一种实现方式是规定所有进程在开始执行前请求所有需要的全部资源。
- 3. **破坏不可以抢占条件**
- 4. **破坏环路等待** 给资源统一编号，进程只能按照编号顺序来请求资源。

死锁避免

在程序运行的时候避免发生死锁

1.安全状态

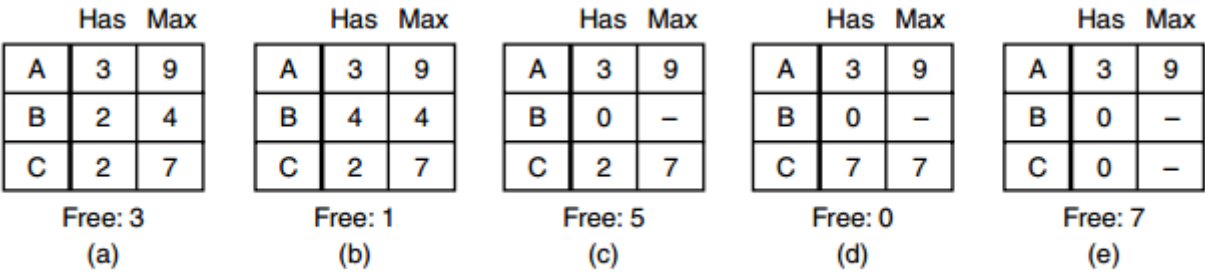


Figure 6-9. Demonstration that the state in (a) is safe.

图中

has表示已经拥有的资源数，max代表需要最大的资源数，Free代表还剩可以使用的资源数。从图a开始，先满足B所需所有资源，运行结束之后，剩余资源变成了5，满足C所需所有资源，最后满足A；

如果所有进程突然请求对资源的最大需求，也仍然存再某种调度次序能够使得每一个进程运行完毕，则称该状态是安全的。

安全状态的检测和死锁的检测是类似的。银行家算法与死锁检测很类似；

2.单个银行家算法

一个小城镇的银行家，他向一群客户分别承诺了一定的贷款额度，算法要做的是判断对请求的满足是否会进入不安全状态，如果是，就拒绝请求；否则予以分配。

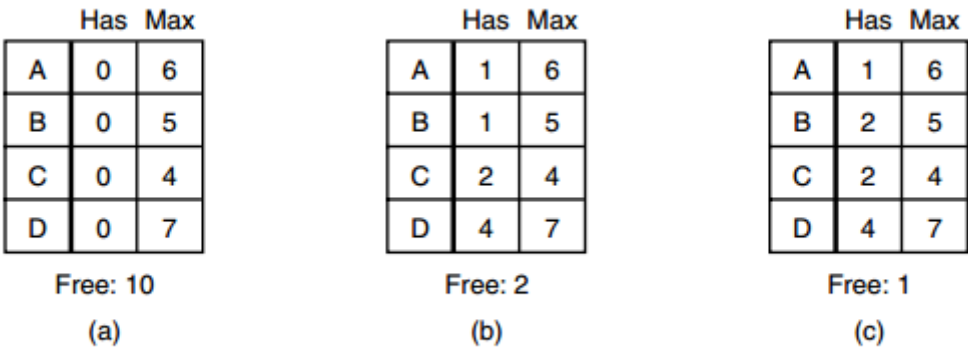


Figure 6-11. Three resource allocation states: (a) Safe. (b) Safe. (c) Unsafe.

c图是不安全的

3.多个资源的银行家算法

	Process	Tape drives	Plotters	Printers	Blu-rays
A	3	0	1	1	
B	0	1	0	0	
C	1	1	1	0	
D	1	1	0	1	
E	0	0	0	0	

Resources assigned

	Process	Tape drives	Plotters	Printers	Blu-rays
A	1	1	0	0	
B	0	1	1	2	
C	3	1	0	0	
D	0	0	1	0	
E	2	1	1	0	

Resources still assigned

$E = (6342)$
 $P = (5322)$
 $A = (1020)$

Figure 6-12. The banker's algorithm with multiple resources.

五个进程，四个资源。左边的图表示已经分配的资源，右边的图表示还需要分配的资源。最右边的 E、P 以及 A 分别表示：总资源、已分配资源以及可用资源，注意这三个为向量，而不是具体数值，例如 $A=(1020)$ ，表示 4 个资源分别还剩下 1/0/2/0。

在右边的矩阵中找到小于等于向量 A 的行，如果没有就是不安全，如果有，就标记终止，继续重复寻找这样的行，直到所有的行都标记终止，状态就是安全的。否则不安全；