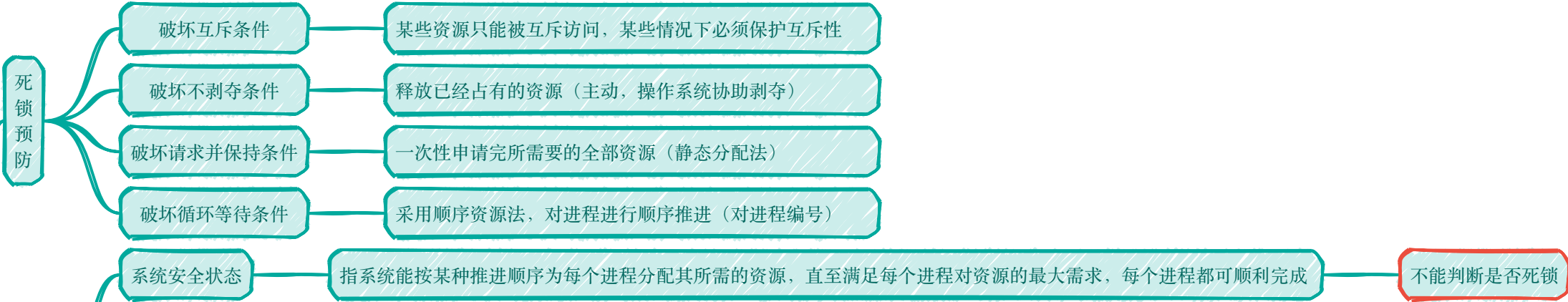
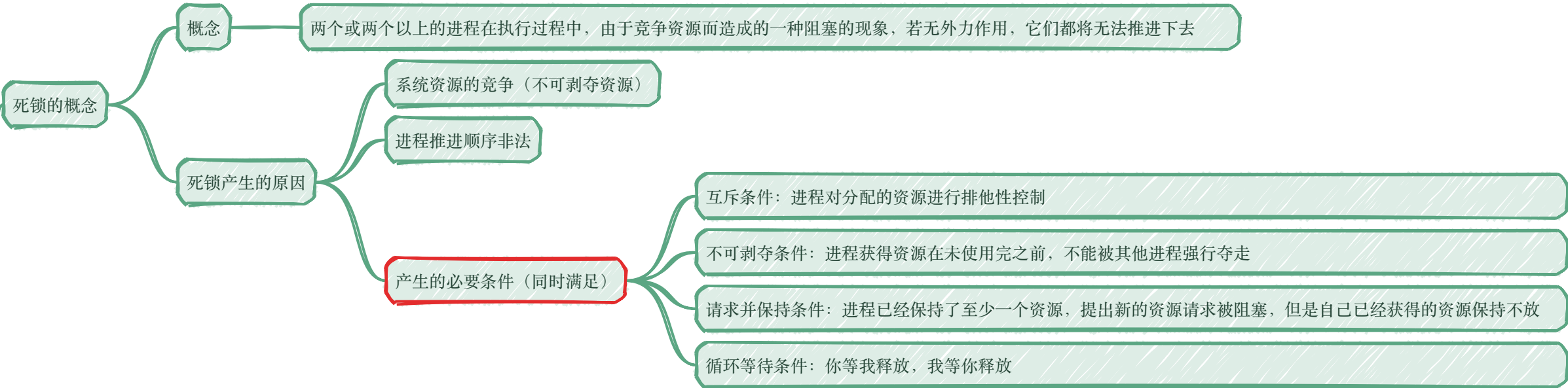


第二章
进程与线程

死锁

死锁的处理策略



死锁避免

银行家算法

进程	最大需求	已分配	最多还需要
P0	(7, 5, 3)	(0, 1, 0)	(7, 4, 3)
P1	(3, 2, 2)	(2, 0, 0)	(1, 2, 2)
P2	(9, 0, 2)	(3, 0, 2)	(6, 0, 0)
P3	(2, 2, 2)	(2, 1, 1)	(0, 1, 1)
P4	(4, 3, 3)	(0, 0, 2)	(4, 3, 1)

资源总数 (10, 5, 7)，剩余可用资源 (3, 3, 2)

概念（三个矩阵图片中从左到右）

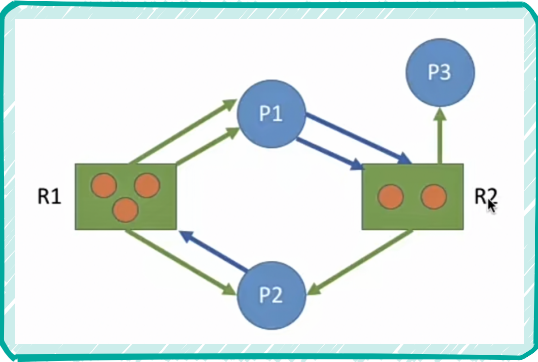
- 可利用资源向量Available：m个元素的数组，每个元素代表一类可用资源数目
- 最大需求矩阵Max：n个进程中每个进程对m类资源最大需求
- 分配矩阵Allocation：系统中每类资源已经分配给每个进程的数量
- 需求矩阵Need：表示接下来每个进程最多还要多少资源
- 矩阵关系：Need=Max-Allocation

执行

判断剩余资源是否能满足需求矩阵中进程，若满足回收已分配资源，继续判断，直到所有资源都被分配完，若无一满足则回溯

死锁的检测和解除

资源分配图



- 绿色箭头：资源分配边
- 蓝色箭头：申请资源边
- 圈圈为进程、框为一类资源（框内圆点为拥有的资源）

死锁定理

- 在资源分配图中找到分配满足的进程，然后消去请求边与分配边
- 若所有边都可以被消去，则不存在死锁，反之存在死锁

死锁解除

- 资源剥夺法
 - 挂起某些死锁进程，抢占资源，将这些资源分配给其他死锁进程，但需要防止挂起时间过长
 - 撤销进程法
 - 强制撤销部分甚至全部死锁进程，剥夺他们的资源
 - 进程回退法
 - 让一个或者多个进程回退到足以回避死锁的地步，进程回退时自愿释放资源而非被剥夺（要求系统保持进程历史信息，设置还原点）
- 选择原则：优先级、已执行时间、还要多久完成、已用资源数、交互式/批处理式

通常不采用从非死锁进程处抢夺资源

死锁、饥饿、死循环的区别

- 死锁：各进程相互等待对方手里的资源，导致各进程都阻塞，无法向前推进的现象（操作系统解决）
- 饥饿：由于长期得不到想要的资源，某进程无法向前推进的现象（操作系统解决）
- 死循环：某进程执行过程中一直跳不出某个循环的现象（程序员解决）