



# 计算机操作系统

## 6 并发程序设计 – 6.6 死锁

### 6.6.3 死锁的避免

理解死锁的避免

掌握银行家算法

理解银行家算法(例)

# 死锁的避免

- 当不能防止死锁的产生时，如果能掌握并发进程中与每个进程有关的资源申请情况，仍然可以避免死锁的发生
- 只需在为申请者分配资源前先测试系统状态，若把资源分配给申请者会产生死锁的话，则拒绝分配，否则接收申请，为它分配资源

# 银行家算法

- 银行家算法：借钱给有偿还能力的客户
  - 系统首先检查申请者对资源的最大需求量，如果现存的资源可以满足它的最大需求量时，就满足当前的申请
  - 换言之，仅仅在申请者可能无条件地归还它所申请的全部资源时，才分配资源给它

# 银行家算法(例)

- 为了进一步说明这种算法，考虑下面的例子。假设系统有三个进程P，Q，R，系统只有一类资源共10个，目前分配情况如下：

进程	已占资源	还需要申请数
P	4	4
Q	2	2
R	2	7

# 银行家算法(例)

进程	已占资源	还需要申请数
P	5	3
Q	2	2
R	2	7

对P执行分配  
剩余资源1

对R执行分配  
剩余资源1

进程	已占资源	还需要申请数
P	4	4
Q	2	2
R	3	6

- P或者R再申请资源时，不能分配，因为现在只剩下2个资源，不能满足它们的最大需求