



死锁 (Deadlocks)

有关“死锁”的话题

- ◆ 死锁问题
- ◆ 关于死锁的系统模型
- ◆ 死锁状态的四大特征
- ◆ 处理死锁问题的思路（四个层次）
- ◆ 死锁预防, Deadlock Prevention
- ◆ 死锁避免, Deadlock Avoidance
- ◆ 死锁检测, Deadlock Detection
- ◆ 死锁恢复, Recovery from Deadlock

死锁问题，定义

◆ A **set** of blocked processes each **holding** a resource and **waiting to acquire** a resource held by another process in the set.

◆ **一组**被阻塞进程的**集合**；

☞ 集合中每个（进程）元素**既占有**一些资源，同时**又等待**另外一些资源；

☞ 进程所等待的资源恰恰被**集合中的**某个（等待）进程占用

死锁问题，定义

◆ 举例

∞ 计算机系统含 2 个硬盘。进程运行时需要 2 个硬盘。

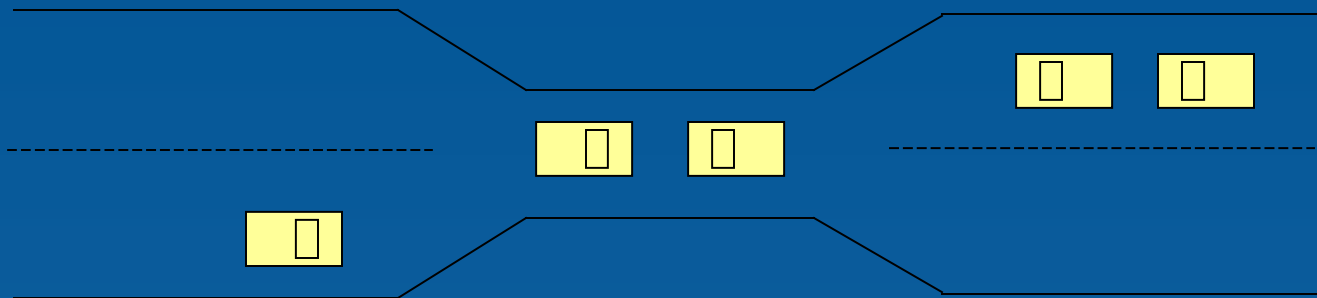
∞ 现状： P_1 进程和 P_2 进程各占用一个进程，同时又申请另外一个硬盘

◆ 举例：生产者 - 消费者问题中的情形

◆ 信号量 A、B 的初始值为 1

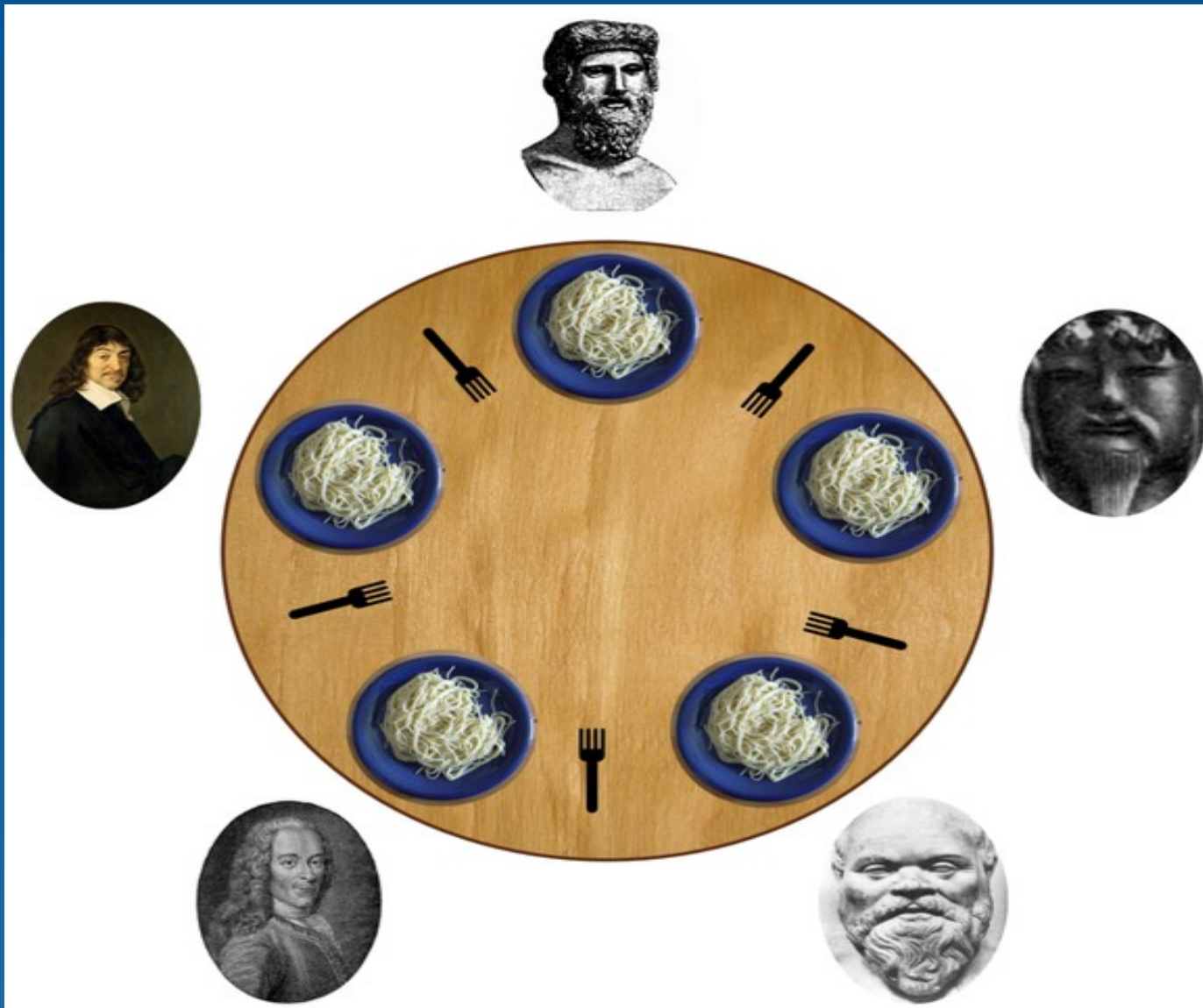
P_0	P_1
wait (A);	wait(B)
wait (B);	wait(A)

举例：车队过独木桥



- ◆双向都有车队
- ◆独木桥划分成 2 个单元位置，每个单元容纳 1 辆车
- ◆当出现如图的情形，陷入死锁
- ◆若想“解套”，其中一方必须后退。可能连累多辆车后退
- ◆可能伴随“饥饿” (Starvation)

举例：哲学家就餐问题



系统模型

- ◆ 一个进程的集合: P_1, P_2, \dots, P_n
- ◆ 各种类型的资源: R_1, R_2, \dots, R_m
- ◆ 进程可使用的 CPU 周期
- ◆ 内存空间
- ◆ 网络及其带宽
- ◆ I/O 设备
- ◆ R_i 类的资源共有 W_i 个实例 (Instances)

系统模型

◆ 进程 P_i 使用资源遵循这样的流程：

- ∞ 申请资源
- ∞ 申请没被批准，则等待
- ∞ 申请获批准，则使用该资源
- ∞ 使用完毕后，释放资源



End