西安交通大学
软件学院

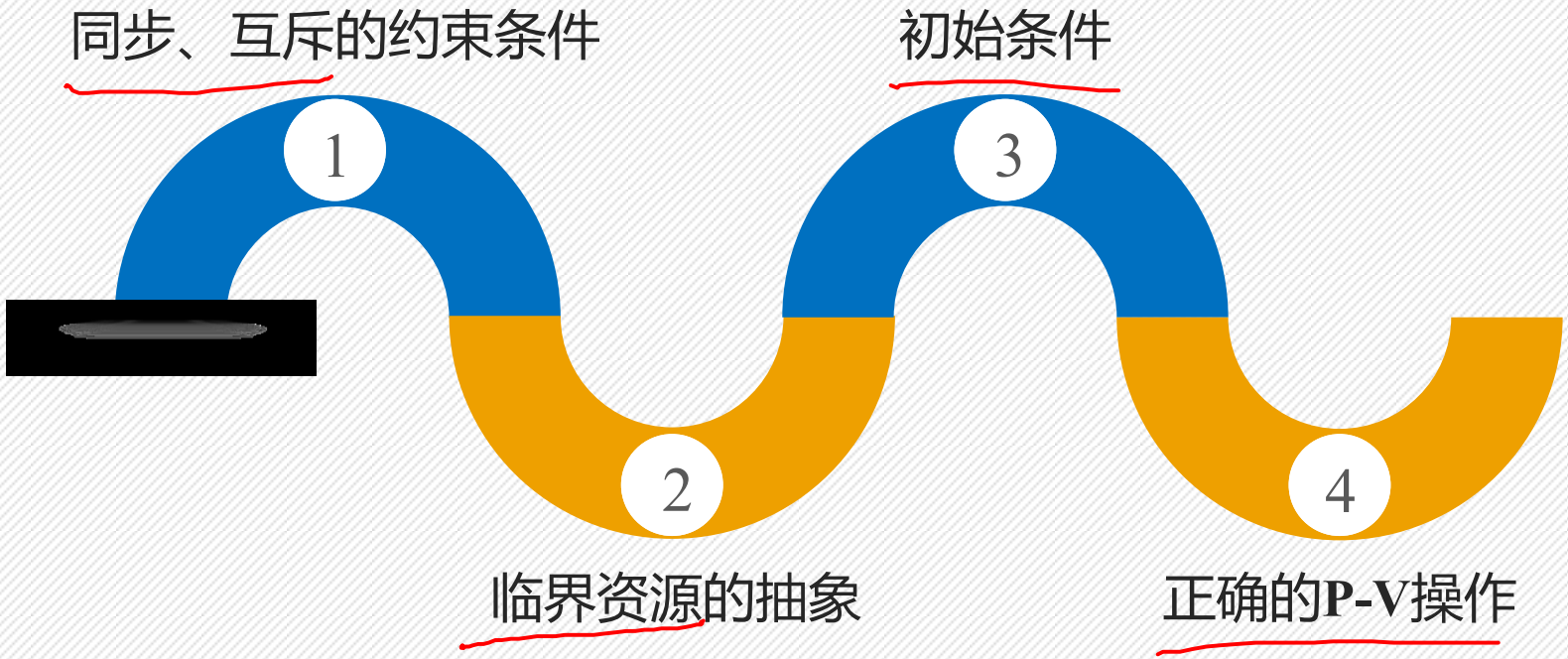
操作系统原理

Operating System Principle

田丽华

6-4 哲学家问题

Semaphore
信号量



Classical Problems of Synchronization

Dining-Philosophers Problem

(哲学家就餐问题)

Bounded-Buffer Problem

(有限缓冲区问题)

Readers and Writers Problem

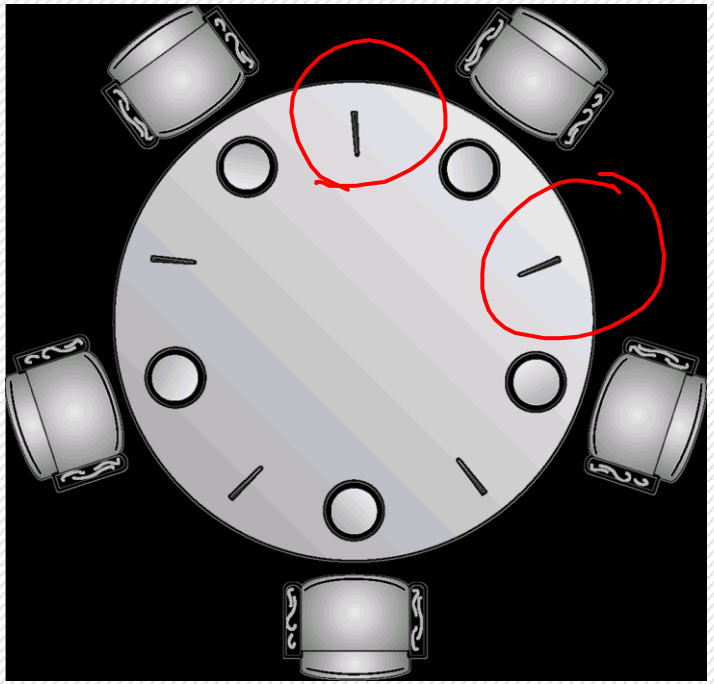
(读者写者问题)

哲学家进餐问题
(the dining philosophers problem)

问题描述：

（由Dijkstra首先提出并解决）5个哲学家围绕一张圆桌而坐，桌子上放着5支筷子，每两个哲学家之间放一支；哲学家的动作包括思考和进餐，进餐时需要同时拿起他左边和右边的两支筷子，思考时则同时将两支筷子放回原处。如何保证哲学家们的动作有序进行？如：不出现相邻者同时进餐；

Dining-Philosophers Problem



- Shared data
- Semaphore chopStick[] = new Semaphore[5];

Philosopher(i)

Repeat

思考;

取**chopStick[i]**;

取**chopStick[(i+1) mod 5]**;

进餐;

放**chopStick[i]**;

放**chopStick[(i+1) mod 5]**;

Until false;

Dining-Philosophers Problem (Cont.)

Philosopher i:

```
while (true) {
    // get left chopstick
    chopStick[i].P();
    // get right chopstick
    chopStick[(i + 1) % 5].P();

    // eat for awhile

    //return left chopstick
    chopStick[i].V();
    // return right chopstick
    chopStick[(i + 1) % 5].V();

    // think for awhile
}
```

讨论

可能会出现死锁，五个哲学家每人拿起了他左边的筷子

01 最多允许四个哲学家同时就坐

02 同时拿起两根筷子

03 非对称解决