

## 第五章 死锁

### 5.1死锁概念

### 5.2死锁起因

### 5.3死锁预防策略

网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 《操作系统原理》

# 5.1 死锁的概念

教师：苏曙光

华中科技大学软件学院

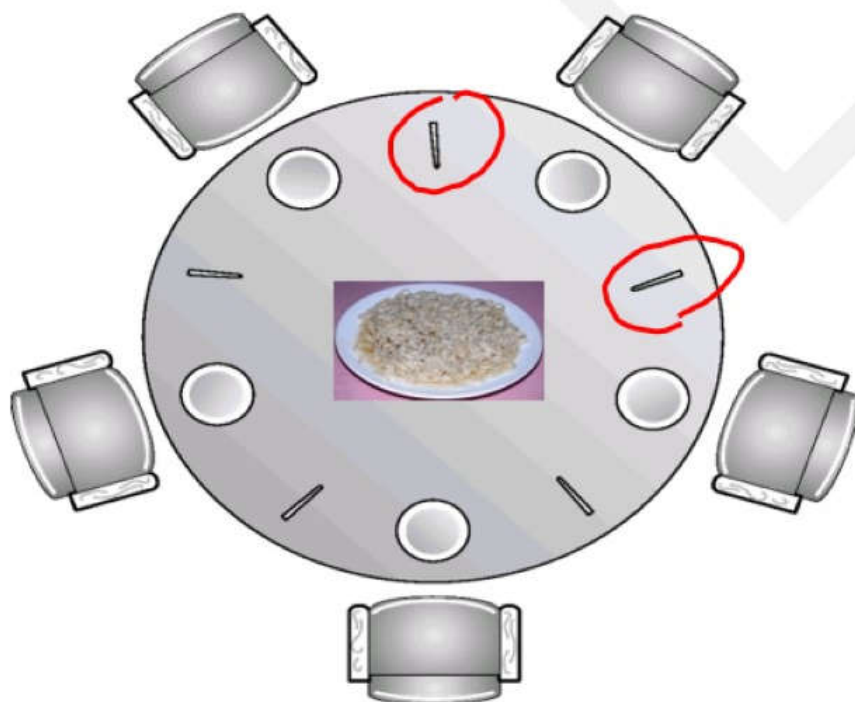


华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 哲学家就餐问题

五个哲学家围坐圆桌边，有5支筷子。哲学家：思考-休息-吃饭-.....



吃饭使用两只筷子：

- 拿一双筷子才能吃；
- 每次只取一支筷子；
- 只取身边的筷子；
- 吃完放下筷子。



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 用线程Philosopher实现哲学家的生活

```
UINT Philosopher(int i) // 线程函数，i是哲学家的编号
{
    while (TRUE)
    {
        思考；
        休息；
        取左手边的筷子
        取右手边的筷子
        吃饭； //正在用两只筷子...
        放下右手边的筷子
        放下左手边的筷子
    }
}
```



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有



网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

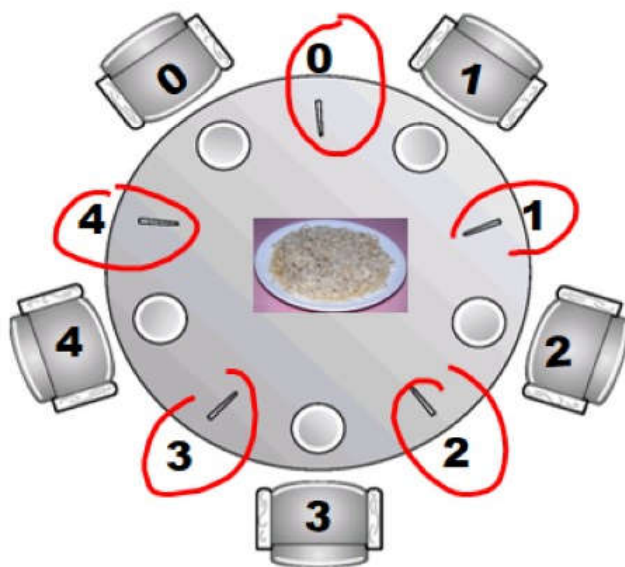
## 用线程Philosopher实现哲学家的生活

//筷子编号：0 - 4（哲学家左手边筷子编号与哲学家编号相同）

int S[5] = { 1, 1, 1, 1, 1 }; // 信号量：i号筷子是否可用：0不可用，1可用

UINT Philosopher (int i) // 线程函数，i是哲学家的编号

```
{  
    while (TRUE)  
    {  
        思考；  
        休息；  
        P(S[i]); //取左手边的筷子  
        P(S[(i+4) % 5]); //取右手边的筷子  
        吃饭； //正在用两只筷子...  
        V(S[(i+4) % 5]); //放下右手边的筷子  
        V(S[i]); //放下左手边的筷子  
    }  
}
```



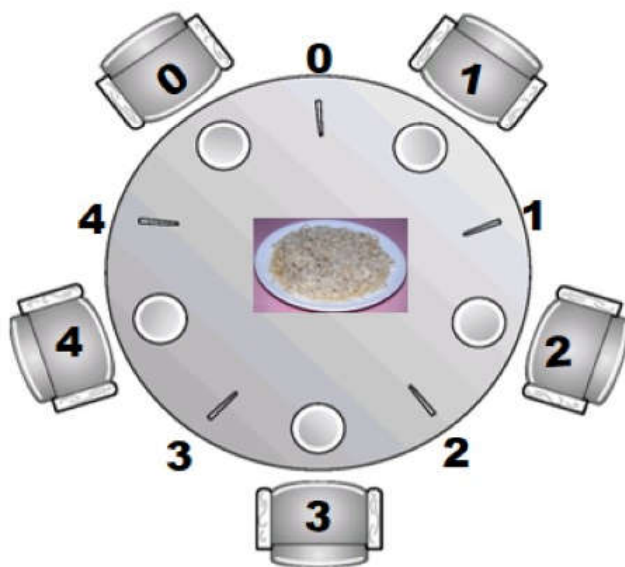
华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 用线程Philosopher实现哲学家的生活

```
//筷子编号：0 - 4（哲学家左手边筷子编号与哲学家编号相同）
int S[5] = { 1, 1, 1, 1, 1 }; // 信号量：i号筷子是否可用：0不可用，1可用
UINT Philosopher (int i) // 线程函数，i是哲学家的编号
{
    while (TRUE)
    {
        思考；
        休息；
        P(S[i]); //取左手边的筷子
        P(S[(i+4) % 5]); //取右手边的筷子
        吃饭； //正在用两只筷子...
        V(S[(i+4) % 5]); //放下右手边的筷子
        V(S[i]); //放下左手边的筷子
    }
}
```

全部哲学家就绪



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有



网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 用线程Philosopher实现哲学家的生活

//筷子编号：0 - 4（哲学家左手边筷子编号与哲学家编号相同）

int S[5] = { 1, 1, 1, 1, 1 }; // 信号量：i号筷子是否可用：0不可用，1可用

UINT Philosopher (int i) // 线程函数，i是哲学家的编号

{

while (TRUE)

{

思考；

休息；

P(S[i]); //取左手边的筷子

P(S[(i+4) % 5]); //取右手边的筷子

吃饭； //正在用两只筷子...

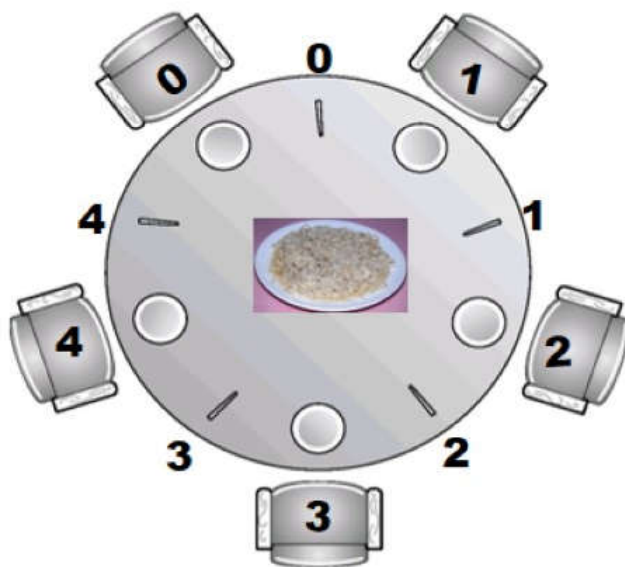
V(S[(i+4) % 5]); //放下右手边的筷子

V(S[i]); //放下左手边的筷子

}

}

全部哲学家  
阻塞



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

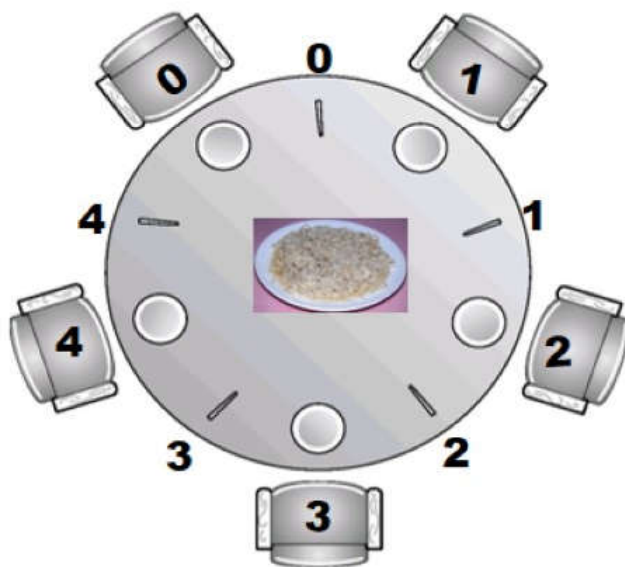
网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 用线程Philosopher实现哲学家的生活

```
//筷子编号：0 - 4（哲学家左手边筷子编号与哲学家编号相同）
int S[5] = { 1, 1, 1, 1, 1 }; // 信号量：i号筷子是否可用：0不可用，1可用
UINT Philosopher (int i) // 线程函数，i是哲学家的编号
{
    while (TRUE)
    {
        思考；
        休息；
        P(S[i]); //取左手边的筷子
        P(S[(i+4) % 5]); //取右手边的筷子
        吃饭； //正在用两只筷子...
        V(S[(i+4) % 5]); //放下右手边的筷子
        V(S[i]); //放下左手边的筷子
    }
}
```

发生死锁

全部哲学家阻塞



华中科技大学苏曙光老师.《操作系统原理》MOOC课程组版权所有



网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

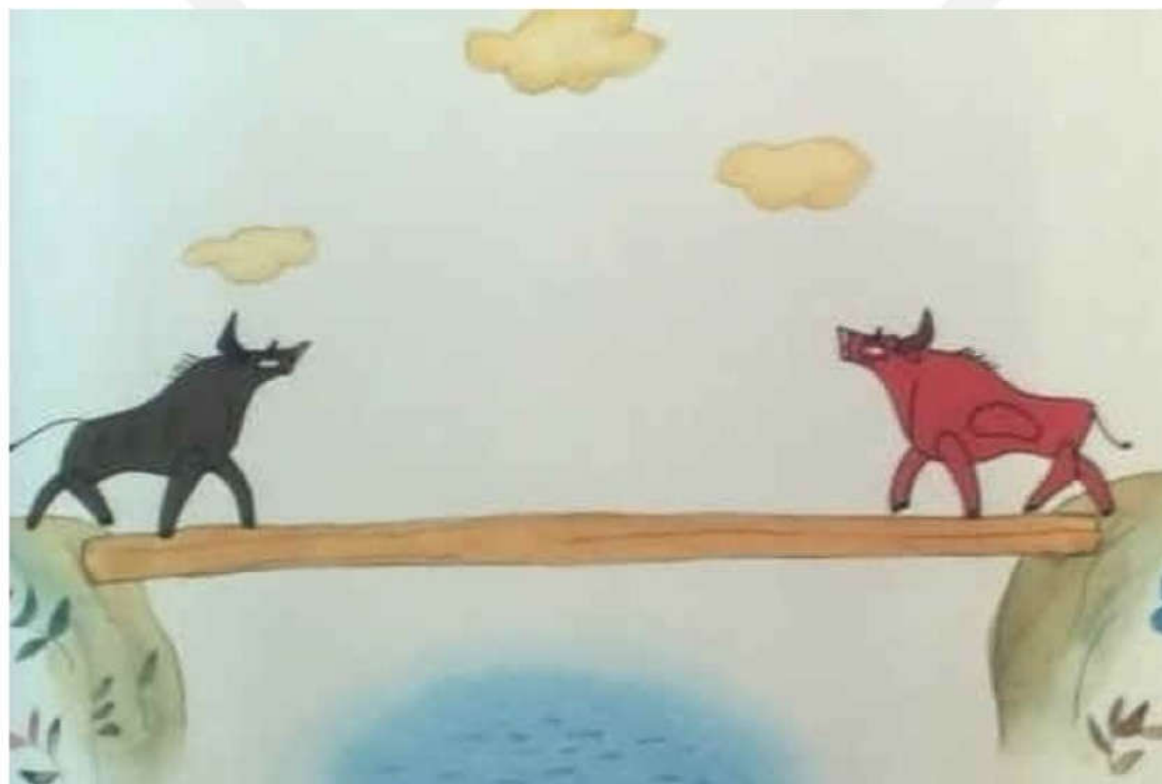
## 死锁概念



华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 死锁概念



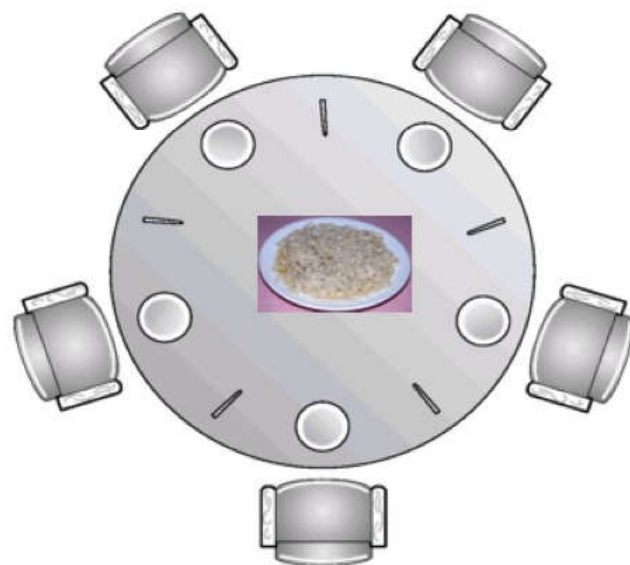
华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

## 死锁的定义

两个或多个进程无限期地等待永远不会发生的条件的一种系统状态。

【结果：每个进程都永远阻塞】

- ◆ 每个哲学家都无限期地等待邻座放下筷子！
- ◆ 而邻座没有吃完之前不会放下筷子！
- ◆ 而邻座缺一只筷子永远都无法吃完！

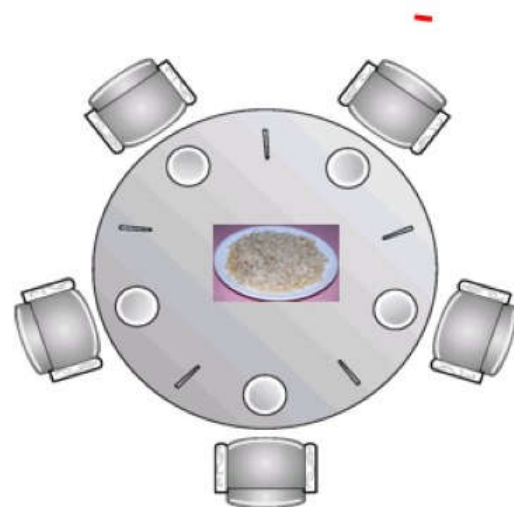




网址：[www.icourses.cn](http://www.icourses.cn)，主页搜索“苏曙光”即可进入MOOC课堂

## 死锁的另一个定义

- 在两个或多个进程中，每个进程都持有某种资源，但又继续申请其它进程已持有的某种资源。此时每个进程都拥有其运行所需的一部分资源，但是又都不够，从而每个进程都不能向前推进，陷于阻塞状态。这种状态称死锁。



华中科技大学苏曙光老师.《操作系统原理》MOOC课程组版权所有