

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，主页搜索“苏曙光”即可进入MOOC课堂

《操作系统原理》

5.2 死锁起因

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，主页搜索“苏曙光”即可进入MOOC课堂

死锁的起因

■ 系统资源有限

- 资源数目不足，进程对资源的竞争而产生死锁。

■ 并发进程的推进顺序不当

- 进程请求资源和释放资源的顺序不当，导致死锁。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂
i和j是独占性的资源（例：键盘，打印机等）。

□ i或j没有被访问完之前（即被释放前）不能被其它进程访问。

进程A

```
1、.....  
2、i = 100 ;  
3、.....  
4、i = j + 800;  
5、i = j * 2 ;  
6、..... ;  
7、j = 200  
8、.....
```

i 被释放

j 被释放

进程B

```
1、.....  
2、j = 200 ;  
3、.....  
4、j = i + 600;  
5、j = i * 2;  
6、.....  
7、i = 100 ;  
8、.....
```

j 被释放

i 被释放

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂
i和j是独占性的资源（例：键盘，打印机等）。

 **i或j没有被访问完之前（即被释放前）不能被其它进程访问。**

进程A

- 1、.....
- 2、 $i = 100$;
- 3、.....
- 4、 $i = j + 800$;
- 5、 $i = j * 2$;
- 6、..... ;
- 7、 $j = 200$
- 8、.....

i 被释放

j 被释放

进程B

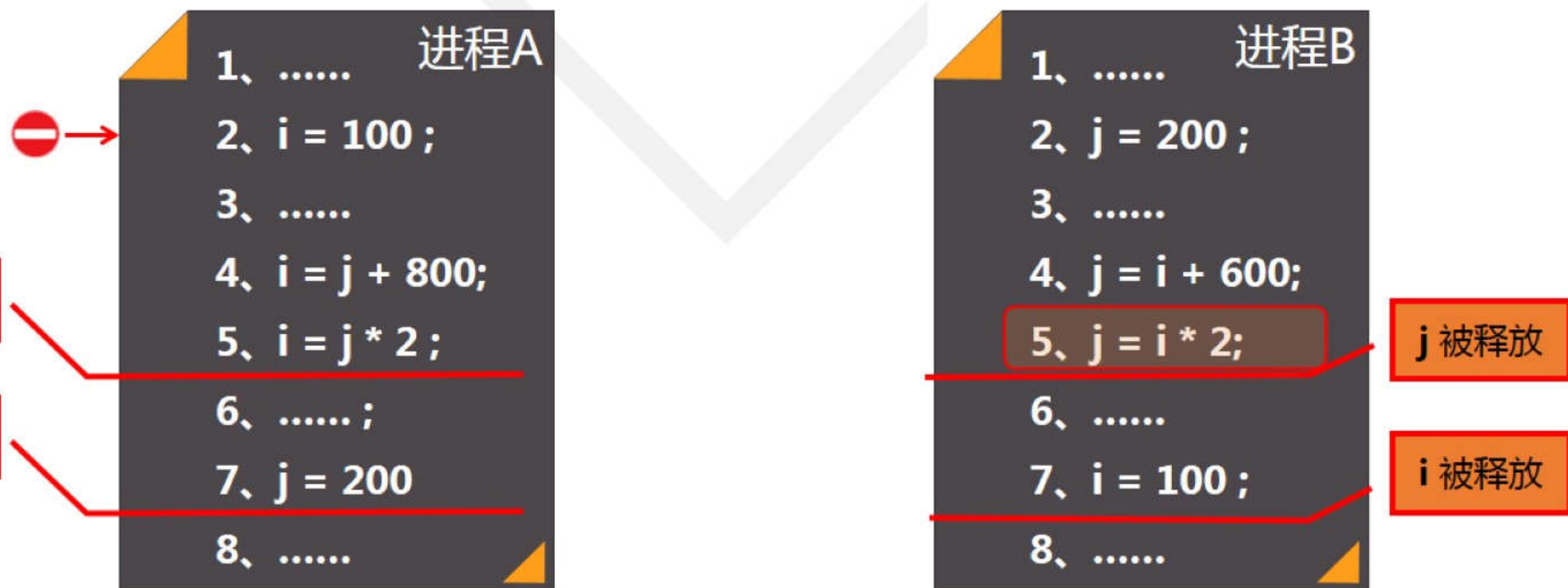
- 1、.....
- 2、 $j = 200$;
- 3、.....
- 4、 $j = i + 600$;
- 5、 $j = i * 2$;
- 6、.....
- 7、 $i = 100$;
- 8、.....

j 被释放

i 被释放

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂
i和j是独占性的资源（例：键盘，打印机等）。

□ i或j没有被访问完之前（即被释放前）不能被其它进程访问。



网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

i和j是独占性的资源（例：键盘，打印机等）。

□ i或j没有被访问完之前（即被释放前）不能被其它进程访问。

进程A

- 1、.....
- 2、 $i = 100$;
- 3、.....
- 4、 $i = j + 800$;
- 5、 $i = j * 2$;
- 6、..... ;
- 7、 $j = 200$
- 8、.....

i 被释放

j 被释放

进程B

- 1、.....
- 2、 $j = 200$;
- 3、.....
- 4、 $j = i + 600$;
- 5、 $j = i * 2$;
- 6、.....
- 7、 $i = 100$;
- 8、.....

j 被释放

i 被释放

□ A,B尽管并发争用ij，且发生了阻塞，但是没有死锁。

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

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

i和j是独占性的资源（例：键盘，打印机等）。

□ i或j没有被访问完之前（即被释放前）不能被其它进程访问。



□ A,B分别在第4行阻塞，等待对方释放j和i：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

不正确的P-V
操作也可能会
带来死锁

生产者和消费
者问题

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

```
int full = 0;    /* 信号量：缓冲区中的数据个数，初值0 */
int empty = 5;  /* 信号量：缓冲区中的空位个数，初值5 */
int mutex = 1;  /* 信号量：缓冲区互斥使用，初值1，可用 */
```

```
producer_i ( ) // i = 1 .. m
{
    while( TRUE )
    {
        生产1个数据；
        P(empty);
        P(mutex);
        存1个数据到缓冲区；
        V(mutex);
        V(full);
    }
}
```

```
consumer_j ( ) // j = 1 .. k
{
    while( TRUE )
    {
        P(full);
        P(mutex);
        从缓冲区取1个数据；
        V(mutex);
        V(empty);
        消费一个数据；
    }
}
```

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

不正确的P-V
操作也可能会
带来死锁

生产者和消费
者问题

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

```
int full = 0;    /* 信号量：缓冲区中的数据个数，初值0 */
int empty = 5;   /* 信号量：缓冲区中的空位个数，初值5 */
int mutex = 1;   /* 信号量：缓冲区互斥使用，初值1，可用 */
```

```
producer_i ( ) // i = 1 .. m
{
    while( TRUE )
    {
        生产1个数据；
        → P(mutex);
        → P(empty);
        存1个数据到缓冲区；
        V(mutex);
        V(full);
    }
}
```

```
consumer_j ( ) // j = 1 .. k
{
    while( TRUE )
    {
        P(full);
        P(mutex);
        从缓冲区取1个数据；
        V(mutex);
        V(empty);
        消费一个数据；
    }
}
```

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

不正确的P-V
操作也可能会
带来死锁

生产者和消费
者问题

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

```
int full = 0; /* 信号量：缓冲区中的数据个数，初值0 */
int empty = 5; /* 信号量：缓冲区中的空位个数，初值5 */
int mutex = 1; /* 信号量：缓冲区互斥使用，初值1，可用 */
```

```
producer_i ( ) // i = 1 .. m
{
    while( TRUE )
    {
        生产1个数据; 6
        P(mutex);
        P(empty);
        存1个数据到缓冲区;
        V(mutex);
        V(full);
    }
}
```

假定：某个时刻
FULL = 5
EMPTY = 0
生产者继续生产
.....

死锁！

```
consumer_j ( ) // j = 1 .. k
{
    while( TRUE )
    {
        P(full); 4
        P(mutex); 0-1
        从缓冲区取1个数据;
        V(mutex);
        V(empty);
        消费一个数据;
    }
}
```

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，主页搜索“苏曙光”即可进入MOOC课堂

关于死锁的一些结论

- 参与死锁的进程至少是2个
 - ◆ 两个或以上进程才会出现死锁
- 参与死锁的进程至少有2个已经占有资源 ✓
- 参与死锁的所有进程都在等待资源 ✓
- 参与死锁的进程是当前系统中所有进程的子集 ✓
- 死锁会浪费大量系统资源，甚至导致系统崩溃