

5-14

1

已占有资源A 15 B 2 C 17

可用资源 A 2 B 3 C 3

P4 4 3 7

P3 8 3 12

P5 11 4 16

P1 13 5 18

P2 17 5 20

P4 P3 P5 P1 P2

2

Req_P4 (2 0 1) < Need_P4 (4 2 5)

Req_P4 (2 0 1) < Available(2 3 3)

假设分配给P4, Available(0 3 2) 可以满足P4 ,因此满足请求

Req_P4 (0 2 0) < Need_P1(5 5 9)

Req_P4 (0 2 0) < Available(2 3 3)

假设分配给P1, Available(2 1 3) 不能满足任何一个进程, 存在安全性问题, 拒绝请求

5-11

1

驳船长200米, 当驳船船头到达B时, 请求B吊起, 但如果此时B上面有车, 则驳船无法请求B吊起, 车也无法请求A落下, 造成死锁

2

吊桥AB的吊起和放下绑定, 只有吊桥AB上都没车时, 才能同时吊起, 等待驳船完全通过后, 一起放下

3

设置两个互斥信号量mutex a mutex b, 实现对A桥和B桥的互斥使用

设置一个共享变量count 记录当前占用A桥和B桥的汽车数量

设置互斥信号量mutex 实现汽车对count的互斥访问

```
1  Main(){
2      int mutex a,mutex b,mutex,count;
3      mutex a = 1;
4      mutex b = 1;
```

```
5     mutex = 1;
6     count = 0;
7     bargei;
8     carj;
9 }
10 bargei(){
11     P(mutexb);
12     P(mutexa);
13     吊起B;
14     吊起A;
15     驳船通过A;
16     放下A;
17     驳船通过B;
18     放下B;
19     V(mutexa);
20     V(mutexb);
21 }
22 carj(){
23     P(mutex);
24     count++;
25     if(count==1){
26         P(mutexb);
27         P(mutexa);
28     }
29     V(mutex);
30     汽车通过B;
31     汽车通过A;
32     P(mutex);
33     count--;
34     if(count == 0){
35         V(mutexb);
36         V(mutexa);
37     }
38     V(mutex);
39 }
```