U201914995-CS1903-罗虞阳

5-11

1. 驳船长 200 米大于两个吊桥之间的距离，所以当驳船通过A桥，其船头到达B桥，请求B桥吊起，而此时它的尾部占据A桥，若这个时候B桥及B桥到A桥之间的公路都被汽车占据，而汽车又要求通过A桥。这样驳船和汽车都无法前进，形成死锁的局面。
2. 可规定资源按序申请和分配。规定如果B桥的序号小于A桥的序号，驳船和汽车都必须先申请序号小的资源B桥，申请得到满足后，再申请序号大的资源A桥。
3. main{

int SA=1；//A桥的互斥信号量//

int SB=1；//B桥的互斥信号量//

cobegin

Auto1;Auto2;……Autoi;

Ship1; Ship2; ……Shipj;

coend

}

Autoi(){//汽车进程

车在公路上行驶；

P（SB）；

过B桥；

V（SB）；

过弯道；

P（SA）；

过A桥；

V（SA）；

车在公路上行驶；

}

Shipj(){//驳船进程

运河航行；

P（SB）；

P（SA）；

吊起过A桥；

运河航行；

吊起过B桥；

V（SA）；

V（SB）；

运河航行；

}

5-12

1. 死锁的四个必要条件是：

1. 互斥：每个路口只能在一个时间点通过一辆车
2. 占有并等待：一辆车在一个时间点通过一个路口的时候，其他车不允许同时通过
3. 非抢占：只有当一辆车完全通过一个路口的时候，下一辆车才能使用这个路口
4. 循环等待：左上路口的车在等待左下路口的车通过；左下路口的车在等待右下路口的车通过；右下路口的车在等待右上路口的车通过；右上路口的车在等待左上路口的车通过

2. 在四个路口均设立红绿信号灯，要求在前30秒仅允许横向车辆通过，后30秒仅允许纵向车辆通过，然后以此顺序进行轮转

3.

假设还未通过一个路口的车队为Ai，已通过一个路口的车队为Bi；

main(){

Int mutexi=1 // variable i from 0 to 3; mutexi控制四个路口

Int mutexBi=1 //variable I from 0 to 3;mutexBi控制四个路口与路口之间的车队

Int emptyBi=NBi,fullBi=0 //variable I from 0 to 3;表示路口间车队可用资源数目和已使用资源数目；

Cobegin

PiA,PiB// variale I from 0 to 3;

coend

}

PiA(){

While(1){

If(有一辆车到达Ai)

countAi ++

if(countAi > 0){

P(emptyBi)

P(mutexBi)

P(mutexi)

Ai中的一辆车通过路口i

V(mutexi)

V(mutexBi)

V(fullBi)

}

}

}

PiB(){

While(1){

P(fullBi)

P(mutexBi)

P(mutex((i+1)%4))

Bi中的一辆车通过路口(i+1)%4

V(mutex(i+1)%4)

V(mutexBi)

V(fullBi)

}

}

5-13

1. 有可能。可用资源为3，A、B、C三个所需的资源为4、7、1，如果A或者B占用了则会造成死锁。
2. 不会。先满足C，在满足A，最后满足B。

5-14

1. 是。P4 P3 P5 P1 P2
2. 任一进程均不能满足，都会发生死锁。