1. 产品A,B入库过程

Int mutex=1 ; /\*互斥信号量\*/

Int sa=M-1-(pa-pb); /\*pa,pb为产品A,B的库存数量\*/

Int sb=N-1-(pb-pa);

Main()

{ cobegin

inputA();

inputB();

coend;

}

inputA();

{

While(true)

{

P(sa);

P(mutex);

产品A入库；

V(mutex);

V(sb);

}

inputB();

{

P(sb);

P(mutex);

产品B入库；

V(mutex);

V(sa);

}

2. 在这道经典题目中，每一把叉子都是必须互斥使用的，因此，应为每把叉子设置一个互斥信号量Si (i=0,1,2,3,4)，初值均为1。当一个哲学家吃通心面之前必须获得自己左边和右边的两把叉子，即执行两个P 操作，吃完通心面后必须放下叉子，即执行两个V 操作。程序如下：

main()

{

int fork[0...4]={1,1,1,1,1,1};

cobegin

p1;p2;p3;p4;p5;

coend

}

process Pi /\*i=0,1,2,3,4

{

while()

{

思考;

P(fork[i]);

P(fork[i+1] mod 5);

吃通心面;

V(fork[i]);

V(fork[i+1] mod 5);

}

}

1. 司机-售票员同步问题

Int s1=0; /\* 0表示停车，1表示启动 \*/

Int s2=0; /\* 0表示不开门，1表示开门 \*/

Main()

{cobegin

Driver();

Busman();

Coend；

}

Driver()

{

While（true）

{ p(s1)；

启动车辆；

正常行车；

到站停车；

V(s2);

}

}

Busman()

{

While(ture)

{

关车门；

V(s1);

售票；

P(s2);

开车门；

上下乘客；

}

}

1. 苹果-香蕉问题

Int disk=1;

Int apple=0;

Int banana=0;

Main()

{

Cobegin

Father();

Mother();

Son();

Daughter();

Coend;

}

Father()

{

While(ture)

{p(dish);

将苹果放入盘中；

V(apple);

}

}

Mother()

{

While()

{p(dish);

将香蕉放入盘中；

V(banana);

}}

Son()

{while()

{

P(banana);

从盘中取香蕉；

V(dish);

吃香蕉；

}

}

Daughter（）

{

While（ture）

{

P(apple);

从盘中取出苹果；

V(dish);

吃苹果；

}

}