1. cp和iop的多缓冲区问题

semaphore mutex = 1; //互斥临界资源

semaphore S = 1; //互斥缓冲区资源

int num = 0; //已使用的缓冲区资源

void cp()

{

while(计算未完成)

{

得到一个计算结果;

P(mutex);

if(num < N)

{

P(S);

将数送到缓冲区中;

V(S);

num++;

}

V(mutex);

}

}

void iop()

{

while(打印工作未完成)

{

P(mutex);

if(num > 0)

{

P(S);

从缓冲区中取数;

V(S);

num--;

}

V(mutex);

从打印机上输出;

}

}

void main()

{

cobegin

cp();

iop();

coend

}

2、水果问题

该问题属于同步互斥问题，应设置三个信号量S、So、Sa，信号量S表示盘子是否为空，其初值为l；信号量So表示盘中是否有桔子，其初值为0；信号量Sa表示盘中是否有苹果，其初值为0。这里，生产者放入缓冲区的产品有两类，消费者也有两类，每类消费者只消费其中固定的一类产品。同步描述如下：

int S＝1;

int Sa＝0;

int So＝0;

main()

{

cobegin

father(); //父亲进程

son(); //儿子进程

daughter(); //女儿进程

coend

｝

father()

{

while(1)

{

P(S);

将水果放入盘中;

if（放入的是桔子）V(So);

else V(Sa);

}

}

son()

{

while(1)

{

P(So);

从盘中取出桔子;

V(S);

吃桔子;

｝

}

daughter()

{

while(1)

{

P(Sa);

从盘中取出苹果;

V(S);

吃苹果;

｝

｝