复习纲要（6-7）

6. 进程同步

- 竞争条件
- 临界区问题（三个要求：互斥、前进、有限等待，理解）
  - 软件同步（Peterson算法，仅适用两个进程）
  - 硬件同步（testandset，适用多进程，使用复杂）
  - **信号量**（重点，须在理解的基础上做题）
- 若干经典同步问题（有限缓冲、读者-写者、哲学家就餐、习题课上的若干问题）

7. 死锁

- 死锁的4个必要条件
- 死锁预防
- 死锁避免
  - 安全状态（与死锁的关系）
  - 银行家算法（书上的例题必须会做）
- 死锁监测

进程同步:习题1

- 现有4个进程R1，R2，W1，W2，它们共享可以存放一个数的缓冲器B。
  - 进程R1每次把从键盘上输入的一个数存放到缓冲器B中，供进程W1打印输出；
  - 进程R2每次从磁盘上读一个数放到缓冲器B中，供进程W2打印输出。
  - 当一个进程把数据存放到缓冲器B后，在该数还没有被打印输出之前不准任何进程再向缓冲器中存数
  - 在缓冲器B中还没有存入一个新的数之前不允许任何进程加快从缓冲区中取出打印。
- 怎样才能使这四个进程并发执行时协调工作？

进程同步:习题1

- 这四个进程实际上是
  - 两个生产者 R1，R2，和
  - 两个消费者 W1，W2，
- 各自生成不同的产品给各自的消费对象。
- 他们共享一个的缓冲器。由于缓冲器只能存放一个数，所以，R1和R2在存放数时必须互斥。而R1和W1、R2和W2之间存在同步。

进程同步:习题1

- 为了协调它们的工作可定义三个信号量：
- S：表示能否把数存人缓冲器B，初始值为1.
- S1：表示R1是否已向缓冲器存入从键盘上读入的一个数，初始值为0.
- S2：表示R2是否已向缓冲器存入从磁盘上读入的一个数，初始值为0.

进程同步:习题1

```
semaphore S=1,S1=S2=0;
buffer B;
process R1() {
    int x;
    do {
        接收来自键盘的数;
        x=接收的数;
        wait(S);
        B=x;
        signal(S1);
     } while(1);
}
```

```
process R2() {
    int y;
    do {
        从磁盘上读一个数;
        y=接收的数;
        wait(S);
        B=y;
        signal(S2);
     } while(1);
}
```

进程同步:习题1

```
process W1() {
    int i;
    do {
        wait(S1);
        i=B;
        signal(S);
        打印i中数;
    } while(1);
}
```

```
process W2() {
    int j;
    do {
        wait(S2);
        j=B;
        signal(S);
        打印j中数;
    } while(1);
}
```

进程同步:习题2

- 某银行提供1个服务窗口和10个供顾客等待的座位。顾客到达银行时，若有空座位，则到取号机上领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下：

```
cobegin
        process顾客i
          {
          从取号机获得一个号码;
          等待叫号;
          获得服务;
          }

                                  process营业员
                                        {
                                        while(TRUE)
                                          {
                                          叫号;
                                          为顾客服务;
                                          }
                                        }
coend
```

- 请添加必要的信号量以及wait()、signal()操作，实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

```
Semaphore seats =10;//表示空余座位数量的资源信号量，初值为10
Semaphore mutex = 1; //管理取号机的互斥信号量，初值为1，表示取号机空闲
Semaphore custom = 0; //表示顾客数量的资源信号量，初值为0
   Process 顾客
      {
          wait(seats); //找个空座位
          wait(mutex); //在看看取号机是否空闲
            从取号机取号;
          signal(mutex) //放开那个取号机
          signal(custom); //取到号，告诉营业员有顾客
            等待叫号;
          signal(seats) //被叫号，离开座位
            接受服务;
      }

  Process 营业员
      {
         While(true)
        {
           wait(custom); //看看有没有等待的顾客
           叫号;
          为顾客服务;
        }
      }
```

```
Process 顾客 i：
 { wait (seats);
    wait(mutex);
    从取号机获取一个号码；
    signal(mutex);
    signal(customer);
    wait(service); //等待叫号;
    signal (empty);
    获得服务；
}
```

```
Process 营业员
  while(TRUE)
  {   wait(customer);
      signal(service);  // 叫号；
      为顾客提供服务；
   }
```

进程同步：习题3

- a、b两点之间是一段东西向的单行车道，现要设计一个自动管理系统，管理规则如下：
  - 当ab之间有车辆在行驶时，同方向的车可以同时驶入ab段，但另一方向的车必须在ab段外等待。
  - 当ab之间无车辆在行驶时，到达a点（或b点）的车辆可以进入ab段，但不能从a点和b点同时驶入
  - 当某方向在ab段行驶的车辆驶出了ab段且暂无车辆进入ab段时，应当让另一方向等待的车辆进入ab段行驶。
- 请用信号量机制为工具，对ab段实现正确管理以保证行驶安全。

进程同步：习题3

- 这是读者－写者问题的一种变形。
- 我们设置两个共享变量和三个互斥信号量
  - 变量ab记录当前ab段上由a点进入的车辆的数量，
  - S1用于从a点进入的车互斥访问共享变量ab，
  - 变量ba记录当前ab段上由b点进入的车辆的数量，
  - S2用于从b点进入的车互斥访问共享变量ba，
  - Sab 用于a、b点的车辆互斥进入ab段。
- 两个共享变量ab和ba的初值分别为0、0。
- 三个信号量的初值分别为1、1和1。

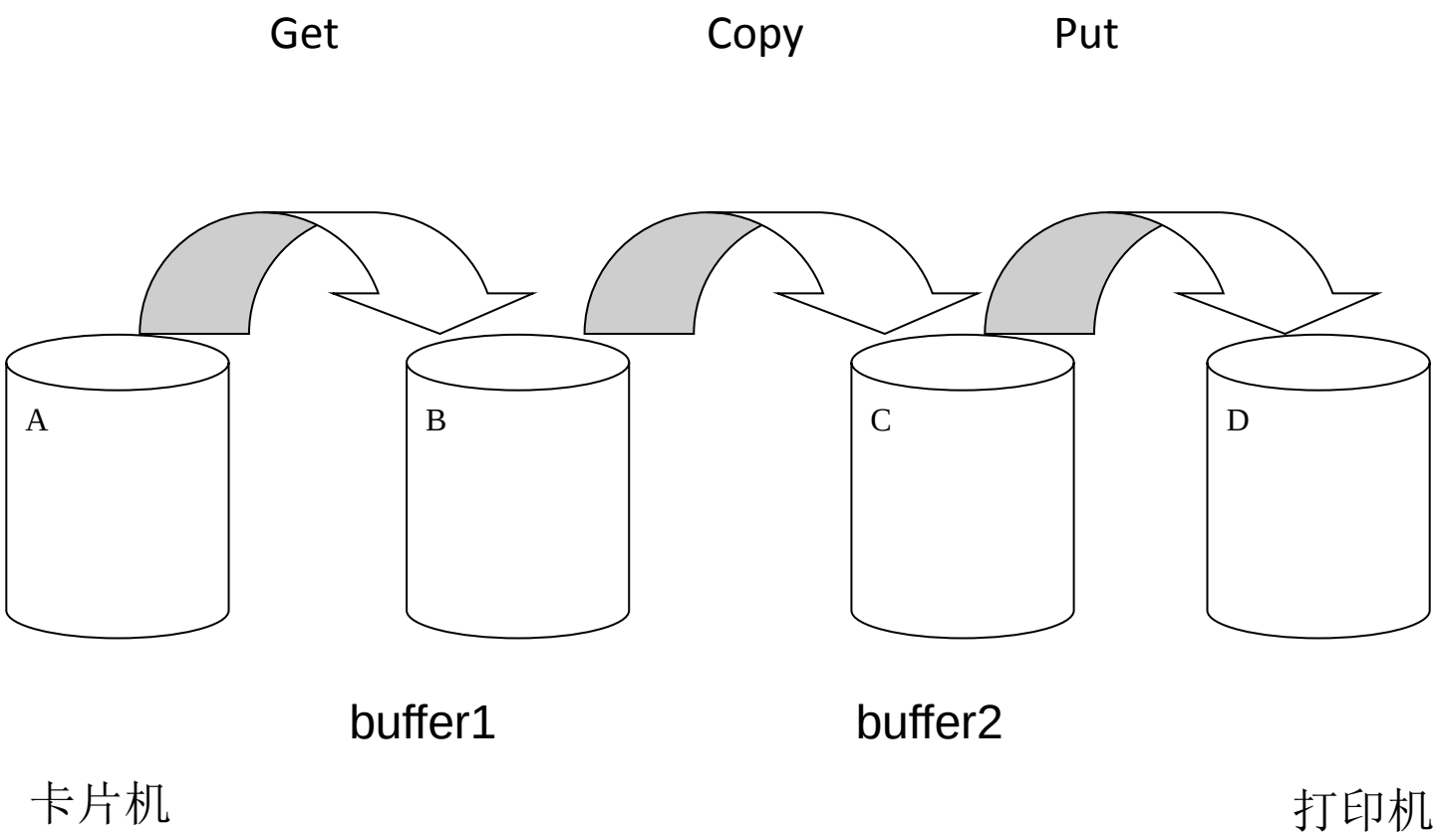
进程同步与互斥：习题3

```
semaphore S1=1,S2=1,Sab=1;
process Pab () {
   do {
      wait(S1);
      if(ab==0) wait(Sab);
      ab=ab+1;
      signal(S1);
      车辆从a点驶向b点;
      wait(S1);
      ab=ab-1;
      if(ab==0) signal(Sab);
      signal(S1);
   } while(1);
}
```

```
int ab=ba=0;
process Pba () {
   do {
      wait(S2);
      if(ba==0) wait(Sab);
      ba=ba+1;
      signal(S2);
      车辆从b点驶向a点;
      wait(S2);
      ba=ba-1;
      if(ba==0) signal(Sab);
      signal(S2);
   } while(1);
}
```

进程同步与互斥：习题4

- 从读卡机上读进N张卡片，然后复制一份，要求复制出来的卡片与读进的卡片完全一样。该任务由三个进程get，copy和put及两个缓冲区buffer1和buffer2完成。进程get的功能是把一张卡片的信息从读卡机读入buffer1；进程copy是把buffer1中的信息复制到buffer2;进程put的功能是取出buffer2中的信息并从打印机上打印。
- 要求说明每个信号量是用于解决什么互斥、什么同步关系的。


Get
Copy
Put
A
B
C
D
buffer1
buffer2
卡片机
打印机

- 2. 互斥信号量 s1=1, s2=1;
- (因为只有一个get、copy和put进程，故可不定义互斥信号量）
- 　　同步信号量4个：
- full1=0, empty1=1; 为一对
- full2=0, empty2=1; 为一对

- get process:

```
while (1) {
    read data;
    wait (empty1);
    wait (s1)
       put data into buffer1
     //因为是单缓，故无in指针，可写为
    // buffer1=x;
    signal (s1)
    signal (full1)
}
```

- Copy process:

```
while (1) {
      wait (full1)
       wait (s1)
          copy data from buffer1
        //单缓,无out指针，可写y=buffer1
       signal (s1)
      signal (empty1)
      wait (empty2)
      wait (s2)
          put data into buffer2
        // buffer2=y
       signal(s2)
       signal (full2)
    }
```

- Put process:

```
while(1) {
        wait (full2)
        wait (s2)
            get data from buffer2
          // z=buffer2
        signal (s2)
        signal (empty2)
              print data on printer
}
```

进程同步与互斥：习题5

在一个箱子里混装有数量相等的黑色围棋子和白色围棋子，现要用自动分拣系统把黑子和白子分开，该系统由两个并发执行的进程组成，功能如下：

- 进程A专门拣黑子，进程B专门拣白子；
- 每个进程每次只拣一个子，当一个进程在拣子时不允许另一个进程去拣子；
- 当一个进程拣了一个棋子（黑子或白子）以后，必须让另一个进程拣一个棋子（黑子或白子），并要求A进程首先开始。

- 定义两个同步信号量s1,s2;
- 因为要求进程A先开始，故S1初值为1；s2初值为0;

- 进程A
```
while（1）{
        wait(s1);
          拣黑子;
        signal(s2);
    }
```

进程B：
```
while（1）{
        wait(s2);
          拣白子;
        signal(s1);
    }
```

死锁：习题1

设系统中有三类资源A、B和C，又设系统中有5个进程P1，P2，P3，P4和P5。A资源的数量为17，B资源的数量为5，C资源的数量为20。在T0时刻系统状态如下：

	最大需求量			已分配资源量			剩余资源		
	A	B	C	A	B	C	A	B	C
P1	5	5	9	2	1	2	2	3	3
P2	5	3	6	4	0	2			
P3	4	0	11	4	0	5			
P4	4	2	5	2	0	4			
P5	4	2	4	3	1	4			

根据银行家算法分析完成以下问题：

(1) T0时刻系统是否处于安全状态？如是，则给出进程安全序列。

(2) T0时刻如果进程P2申请0个资源类A、3个资源类B和4个资源类C，能否实施分配？为什么？

(3) 在(2)的基础上如果进程P4申请2个资源类A、0个资源类B和1个资源类C，能否实施分配？为什么？

(4) 在(3)的基础上如果进程P1申请0个资源类A、2个资源类B和0个资源类C，能否实施分配？为什么？