Chapter 6

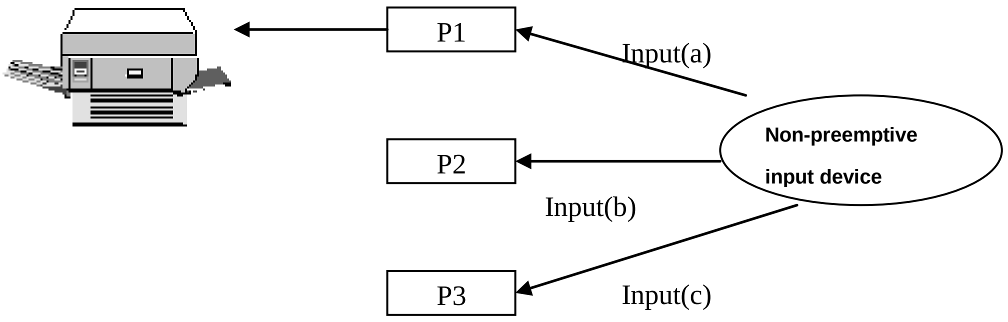
6.0 There are three cooperative processes P1, P2, and P3. They read data , denoted as a、 b and c,from the same input device， as shown in figure below. The input device is an exclusive access device.The first datum(a) must be read by process P1, the second datum(b) is read by process P2, and the third datum(c) by process P3. These three processes perform the following calculations:

P1: x = a + b

P2: y = a * b

P3: z = y + c - a

P1, which is linked to the printer, should print the results of x,y, and z. Write the pseudo code for the three processes, using semaphores to synchronize their activities.



Answer:

```
Semaphore :s1,s2,s3,s4,s5:=0;
P1:              P2:              P3:
{                {                  {
input (a)          wait(s1)          wait(s3)
signal(s1)         input (b)         input (c)
wait(s2)           signal(s2)        wait(s4)
x=a+b              signal(s3)        z=y+c-a
wait(s5)           y=a*b             signal(s5)
Print (x,y,z)      signal(s4)        }
}                }
```

6.01 In a system, there are multiple producer processes which produce numbers to a buffer and multiple consumer processes which consume numbers from the buffer, where the buffer is shared among all producers and consumers. The following variables are shared among all processes:

```
int nextc=0, nextp=0, buf[10];
semaphore full; empty; mutex;
```

Producer and consumer processes are given in the following C++-like pseudo programs

```
   Producer Process:
   int itemp;
   while(1){
1     itemp = rand(); // Generate a number
2     P(empty);
3     P(mutex);
4     buf[nextp]=itemp;
5     nextp=(nextp+1)%10;
6     V(mutex);
7     V(full);
   }
```

```
Consumer Process:
     int itemc;
     while(1){
1     P(full);
2     P(mutex);
3     itemc=buf[nextc];
4     nextc=(nextc+1)%10;
5     V(mutex);
6     V(empty);
7     cout << itemc << endl;
      }
```

(1) What are the critical sections in the given producer and consumer processes? *(4 marks)*

(2) How should the semaphores **full, empty**, and **mutex** be initialized? *(3 marks)*

(3) If we switch the order of 2 and 3 in the producer process and the order of 1 and 2 in the consumer process, would the system still work properly? Justify your answer. *(3 marks)*

```
Producer Process
…
1 itemp = rand(); // Generate a number
2 P(mutex);
3 P(empty);
…
```

```
Consumer Process
…
1 P(mutex);
2 P(full);
3 itemc=buf[nextc];
…
```

(1) Producer: Lines 4 and 5.
    Consumer: Lines 3 and 4.

(2) empty = 10, mutex = 1, and full = 0.

(3) No, the system may be deadlocked. For example, if a producer gets mutex semaphore but there is no more empty item, no consumers can continue and the system is deadlocked.