Chapter 3: Process management part II

- Semaphore Definition
- P and V operation
- Mutex using P and V operation
- Process Sychronization
- Synchronization using P and V operation
- Typical Problems

Semaphore

- It including two parts

  A integer ,its initial value >=0

  and a queue pointer

P and V operation on Semaphore

- P operation:

① sem-l

② If sem>=0 after (sem-1),then it return,the process continue executing

③ If sem<0 after(sem-1),then the process be blocked and put to the block queue.

- V operation

① sem+1;

② If sem>0 after (sem+1),the process continue executing

③ If sem<=0 after(sem+1),then awake a process in the blocking queue

How to implement P and V operation

Mutex using P and V operation

```
- P1                          P2
 P(sem)                    P(sem)
   <S>                       <S>
 V(sem)                    V(sem)
```

Synchronization

- An synchronization example
- Private and public Semaphore

Synchronization using P and V

- Three steps:

① Deciding how many semaphore should use

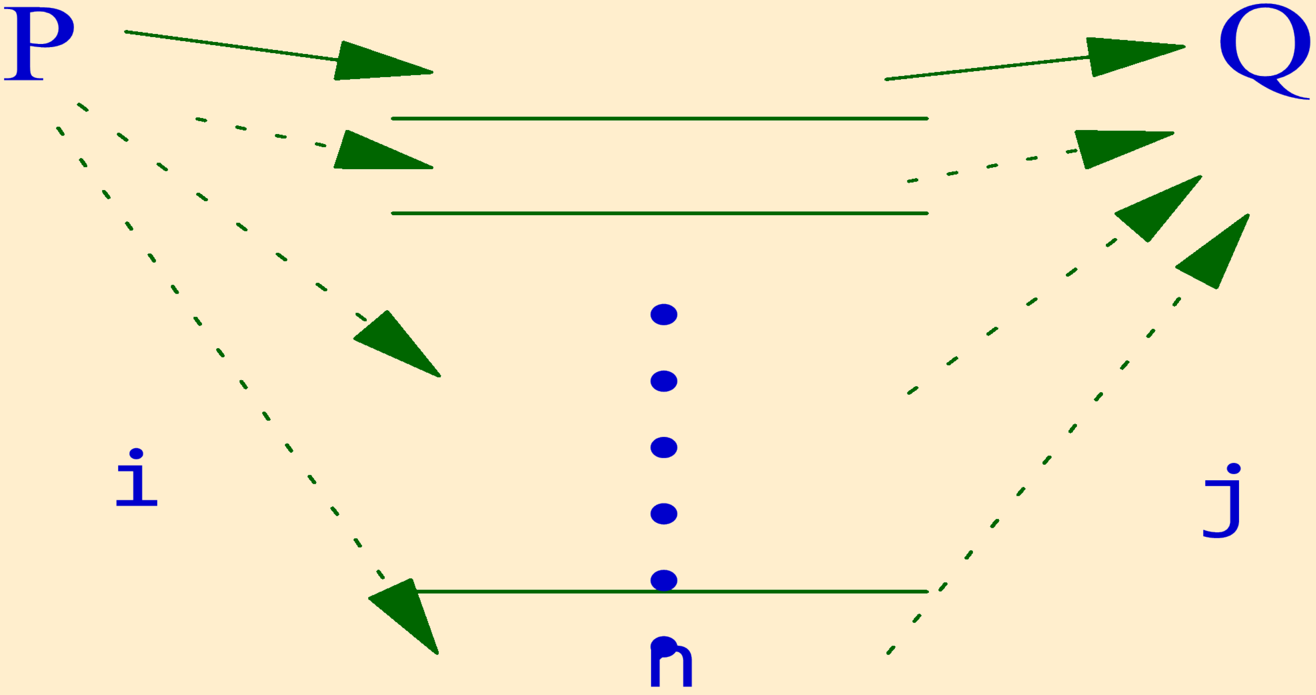
② Initialize the semaphore

③ Using P and V operation

Typical problems

- One producer and one consumer and one buffer
- Multi producers and multi consumers
- Barber shop problem
- Dining-Philosophers Problem

放消息　　　　　　　　取消息



n个缓冲区
(Buffer)

```
P：
  i = 0;
  while (true)  {
      producing;
      P($S_1$);
     put into Buffer [i];
     V($S_2$);
     i = (i+1) % n;
     };
```

```
Q：
     j =  0;
     while (true)  {
     P($S_2$);
     take from Buffer[j];
     V($S_1$);
     consuming;
     j = (j+1) % n;
     };
```

Multi producer and consumers

- Shared data

  semaphore full, empty, mutex;

  Initially:

  full = 0, empty = n, mutex = 1