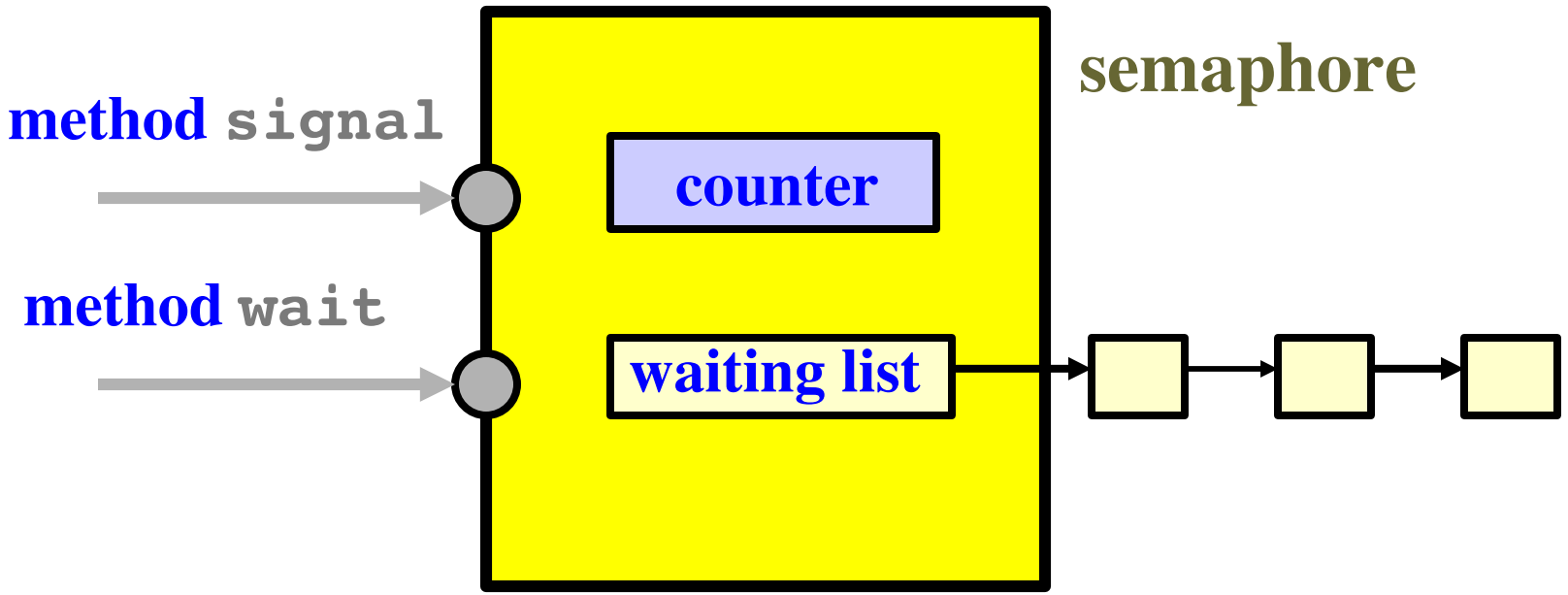
Semaphores

- A *semaphore* is an object that consists of a counter, a waiting list of processes and two methods (*e.g.*, functions): `signal` and `wait`.



Semaphore Method: wait

```
void wait(sem S)
{
    S.count--;
    if (S.count < 0) {
        add the caller to the waiting list;
        block();
    }
}
```

- After decreasing the counter by 1, if the counter value becomes negative, then
  - add the caller to the waiting list, and then
  - block itself.

Semaphore Method: signal

```
void signal(sem S)
{
    S.count++;
    if (S.count <= 0) {
        remove a process P from the waiting list;
        resume(P);
    }
}
```

- After increasing the counter by 1, if the new counter value is not positive, then
  - remove a process P from the waiting list,
  - resume the execution of process P, and return

Three Typical Uses of Semaphores

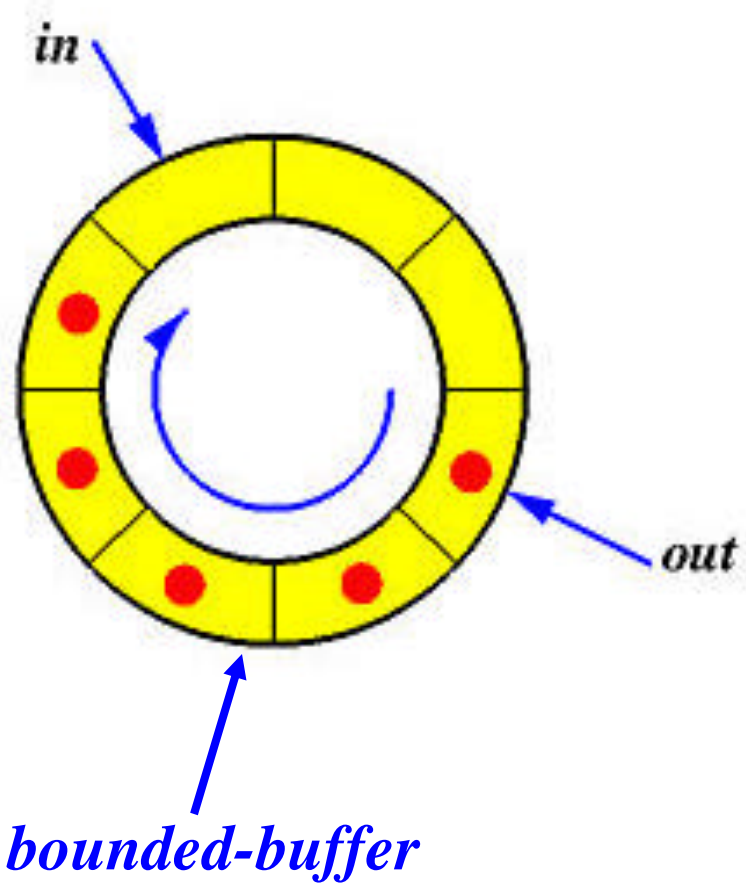
- **There are three typical uses of semaphores:**
  - **mutual exclusion:**

    Mutex (*i.e.*, *Mut*ual *Ex*clusion) locks
  - **count-down lock:**

    Keep in mind that semaphores have a counter.
  - **notification:**

    Indicate an event has occurred.

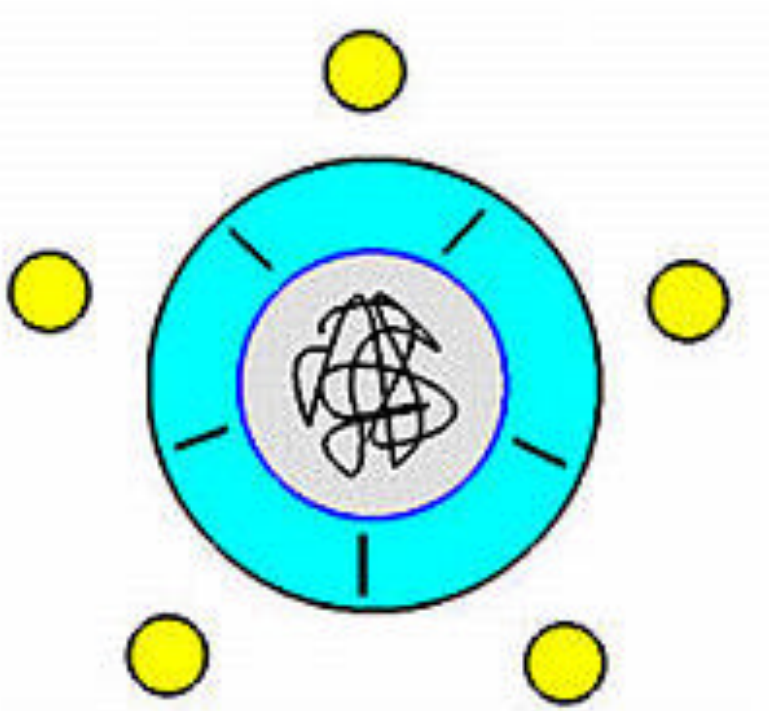
The Producer/Consumer Problem



- Suppose we have a circular buffer of *n* slots.
- Pointers *in* (*resp.*, *out*) points to the first empty (*resp.*, filled) slot.
- Producer processes keep adding info into the buffer
- Consumer processes keep retrieving info from the buffer.

Lock Example: Dining Philosophers

- Five philosophers are in a thinking - eating cycle.
- When a philosopher gets hungry, he sits down, picks up *two nearest* chopsticks, and eats.
- A philosopher can eat only if he has *both* chopsticks.
- After eating, he puts down both chopsticks and thinks.
- This cycle continues.



The Readers/Writers Problem

- Two groups of processes, readers and writers, are accessing a shared resource by the following rules:
  - Readers can read simultaneously.
  - Only one writer can write at any time.
  - When a writer is writing, no reader can read.
  - If there is any reader reading, all incoming writers must wait. Thus, readers have higher priority.