## Blocking mechanism

Associated with each semaphore is a queue of waiting threads

- → When P () is called by a thread:
  - · If semaphore is open, thread continue
  - · If semaphore is closed, thread blocks on queue
- → Then V () opens the semaphore
  - · If a thread is waiting on the queue, the thread is unblocked
  - If no threads are waiting on the queue, the signal is remembered for the next thread

## (Bad) Producer Consumer using a semaphore

```
sem_init(not_full, n)
sem_init(not_empty, 0)
```

```
void producer () {
  while(1) {
    item := produce()
    sem_wait(not_full)
    write(buffer, item)
    sem_signal(not_empty)
  }
}
```

```
void consumer () {
  while(1) {
    sem_wait(not_empty)
    item := read(buffer)
    sem_signal(not_full)
    consume(item)
}
```