

6.033 Spring 2007

Lecture 6

Client / Server within a Computer
and Concurrency

Bounded Buffer Send

```
send(p, m):  
  while true:  
    if p.in - p.out < N:  
      p.buffer[p.in mod N] ← m  
      p.in ← p.in + 1  
      return
```

Bounded Buffer Receive

```
receive(p):  
  while true:  
    if p.out < p.in:  
      m ← p.buffer[p.out mod N]  
      p.out ← p.out + 1  
      return m
```

```
send(p, m):  
  while true:  
    if p.in - p.out < N:  
      p.buffer[p.in mod N] ← m  
      p.in ← p.in + 1  
      return
```

```
receive(p):  
  while true:  
    if p.out < p.in:  
      m ← p.buffer[p.out mod N]  
      p.out ← p.out + 1  
      return m
```

Send with Locking

```
send(p, m):  
  while true:  
    acquire(p.lock)  
    if p.in - p.out < N:  
      p.buffer[p.in mod N] ← m  
      p.in ← p.in + 1  
      release(p.lock)  
      return  
    release(p.lock)
```

Does this work?

```
send(p, m):  
  while true:  
    acquire(p.lock)  
    if p.in - p.out < N:  
      acquire(p.lock)  
      p.buffer[p.in mod N] ← m  
      p.in ← p.in + 1  
      release(p.lock)  
      return  
    release(p.lock)
```