

L6: Client/server in one computer; atomicity

Nickolai Zeldovich
6.033 Spring 2011

Bounded buffer send

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

Bounded buffer receive

```
receive(bb):  
    while True:  
        if bb.out  $\neq$  bb.in:  
            m  $\leftarrow$  bb.buffer[bb.out mod N]  
            bb.out  $\leftarrow$  bb.out + 1  
        return m
```

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

```
receive(bb):  
    while True:  
        if bb.out ≠ bb.in:  
            m ← bb.buffer[bb.out mod N]  
            bb.out ← bb.out + 1  
        return m
```

Send with locking

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

Does this send work?

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