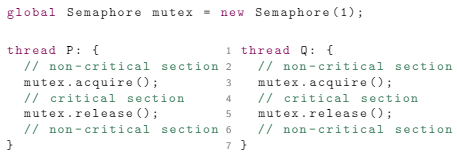
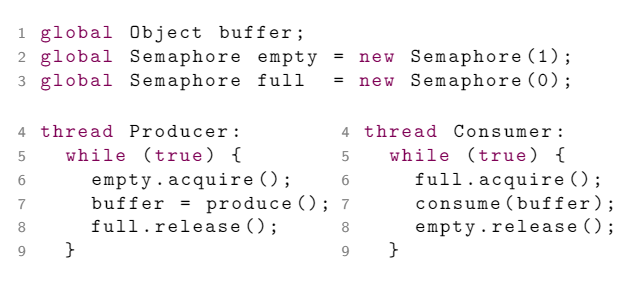
Semaphore  
A semaphore is an Abstract Data Type with the atomic operations:  
acquire (or wait) release (or signal)  
  
it has 2 data fields: permissions (>= 0 int)  
processes: set of processes  
  
aquire consumes permission, release frees one  
  
Binary Semaphore  
Acquire is the same  
Release now sets to 1 instead of incrementing, is undefined at permissions == 1

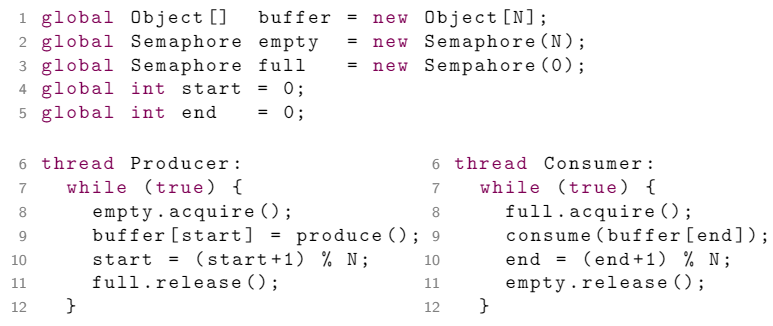
Mutex w/ Semaphores

Semaphore patterns

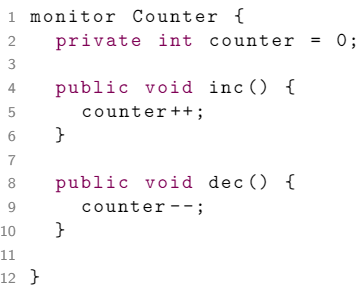
Split binary semaphores Producer/Consumer



N size buffer



MONITORS



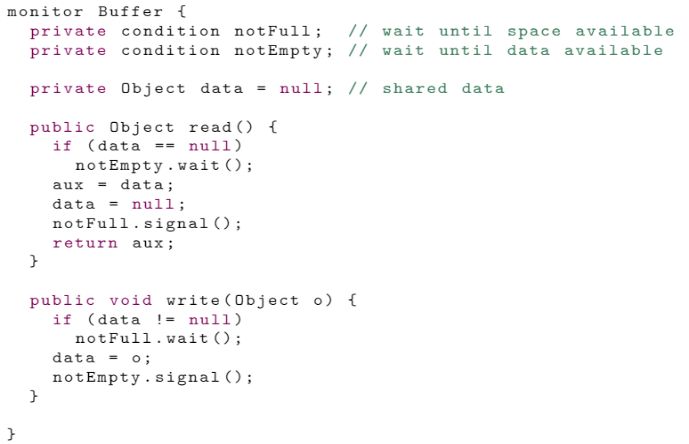
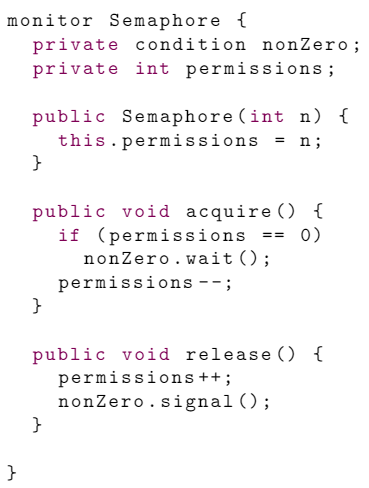
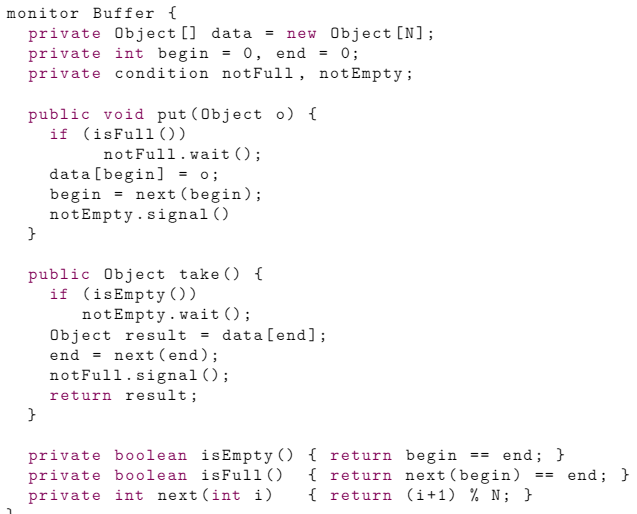
Also have condition variables initialized `condition name;` in pseudo

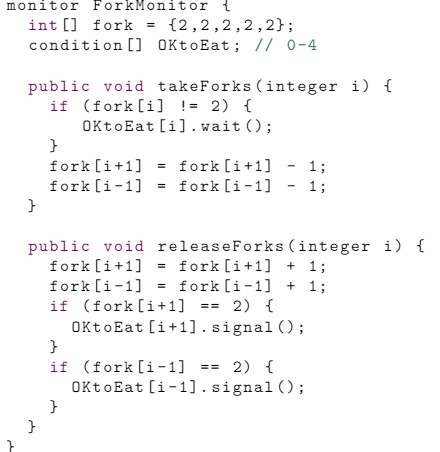
Have Cond.wait(), Cond.signal(), cond.empty()

Wait: puts into blocking queue

Signal: brings out oldest thing in blocking queue

Empty: checks if queue is empty



Fair Semaphore:

