

# CS 3214 "signalling"

mutual exclusion ← protect state from multiple threads  
↳ synch primitive

↳ preserving order

- pthread\_join()

condition variable

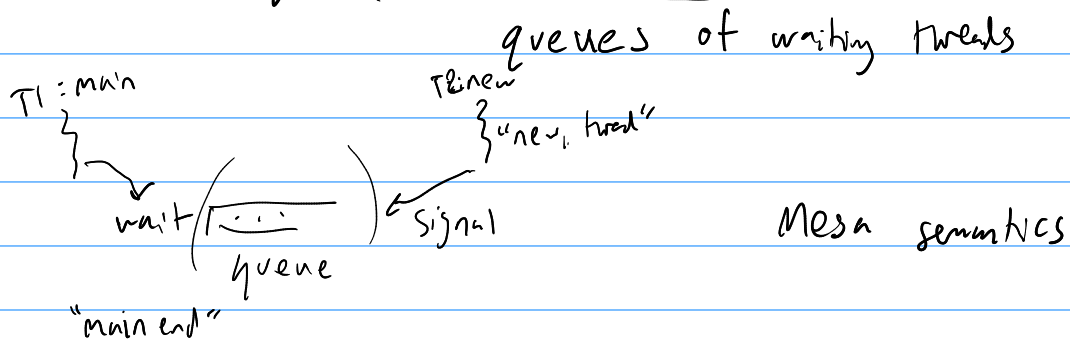
- wait + signal

Solutions to pthread\_join:

\* sleep (not correct, not efficient)

- state variable + loop

- wait + signal: condition variables



3 things:

- state variable "what am I waiting for"
- mutex - protect your state variables
- condition variable: somewhere for threads to wait

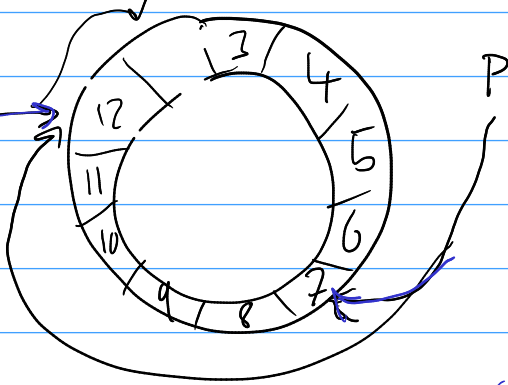
monitor pattern: public synchronized foo() {  
take mutex at start of method  
release mutex at end

"producer/consumer" = "bounded buffer"

unix pipe

wait for  
(item available)  
read C

signal  
the consumer  
has more  
space)



wait (for  
space)  
write element

signal  
(item available)

3 things

- state
- mutex
- condition variable

