# COMP 7500/7506 Lecture 16

# Project 3-3 AUbatch – Data Structures and Condition Variables

**🟊: >85%, 🟊🟊: 70-85%, 🟊🟊🟊: 55-70%, 🟊🟊🟊🟊: 40-55%, 🟊🟊🟊🟊🟊: < 40%**

**🟊🟊🟊 Exercise 1:** Please design data structures for the parameters in the above function prototypes.

**🟊🟊 Exercise 2:** Can you explain this code?Which lock\_acqure() and lock\_release() are a pair? Can we improve this code using wait and signal?

char consumer() {

char c;

lock\_acquire(mutex);

while (count == 0) {

lock\_release(mutex);

lock\_acquire(mutex);

}

count--;

c = buffer[tail];

tail++;

if (tail == SIZE) {

tail = 0;

}

lock\_release(mutex);

return c;

}

void producer(char c) {

lock\_acquire(mutex);

while (count == SIZE) {

lock\_release(mutex);

lock\_acquire(mutex);

}

count++;

buffer[head] = c;

head++;

if (head == SIZE) {

head = 0;

}

lock\_release(mutex);

}

**🟊🟊🟊 Exercise 3:** What’s the problem with the above “lock” solution? How can you improve the above code using the wait-and-signal solution?

**🟊🟊 Exercise 4:** Please complete the consumer code using condition variables.

char consumer() {

char c;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

while (count == 0) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

count--;

c = buffer[tail];

tail++;

if (tail == SIZE) {

tail = 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

return c;

}

**🟊🟊🟊 Exercise 5:** How to implement cv\_wait()?

void cv\_wait(struct cv \*cv, struct lock \*lock) {

use assert to check input cv and lock;

turn off interrupts;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

sleep the thread until someone signals cv;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

turn on interrupts to the previous level;

}

**🟊 Exercise 6:** How to implement cv\_signal()?

void cv\_signal(struct cv \*cv, struct lock \*lock) {

use assert to check cv and lock;

turn off \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

/\* Question: How to implement the following IF \*/

if (this thread does not hold lock)

panic("cv\_signal error: cv %s at %p, lock %s at

%p.\n", cv->name, cv, lock->name, lock);

/\* see also how to wakeup a thread Slide 15 \*/

wakeup one thread using indicator “cv”;

turn on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}