# COMP3500: Classical Synchronization Problems

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

**🟊🟊🟊🟊 Exercise 1 (Plickers):** Which one of the following statement about item “L” in the semaphore structure is **incorrect**?

A. It is a waiting queue.

B. It doesn’t waste CPU time.

C. It avoids busy waiting for a signal.

D. It doesn’t wait for a signal

typedef struct {

int value;  
 struct process \*L;  
} semaphore;

**Exercise 2:** What semaphores do you need to control the shared pool in the bounded buffer problem? (1 minute plus 1-minute group discussion).

**🟊🟊🟊🟊🟊 Exercise 2.1 (Plickers):** (see slides)

**🟊🟊 Exercise 2.2 (Plickers):** (see slides)

**🟊 Exercise 2.3 (Plickers):** (see slides)

**Exercise 3:** Where to place the mutex semaphore?

do {

...  
 /\* produce an item in next\_produced \*/

/\* add next produced to the buffer \*/

...

} while (true);

**Exercise 4:** Can you design a consumer process using the three semaphores? **Hint:** Consider semaphores followed by a mutex.

Do {

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

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

...  
 /\* remove an item from buffer tonext\_consumed \*/

...

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

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

...  
 /\* consume the item in next consumed \*/

...  
 } while (true);

**Exercise 5:** What semaphores do we need for the reader-writer problem? What are their initial values?

**Exercise 6:** Which semaphores do we use in the writer process?

do {

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

...  
 /\* writing is performed \*/

...

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

} while (true);

**Exercise 7:** Explain the Reader Process.

do {  
 wait(mutex);  
 read\_count++;  
 if (read\_count == 1)

wait(rw\_mutex);

signal(mutex);

...  
 /\* reading is performed \*/

...

wait(mutex);  
 read count--;  
 if (read\_count == 0)

signal(rw\_mutex);

signal(mutex);

} while (true);