

## LAB Assignment Week #7

Topic: Semaphore and dining philosopher problem

Name/ID of participant(s)

### **Instructions:**

- You will be working in a group of two. Choosing your own team.
- You are allowed to discuss freely within your group. Avoiding seeking solution from other groups.
- Two computers are provided per group.
- Turn in the result by the end of class period.

### **Activity 1:** Binary semaphore

- Run lab6\_1.cpp (Linux) and lab6\_2.cpp (Windows) to learn how to implement binary semaphore for the problems that require interlocking mechanism
- Compare the results of the programs against the results from using spinlock, mutex lock, or CRITICALSECTION from previous lab.

### **Activity 2:** Bounded buffer problem

- Run lab6\_5.cpp (Linux) and Lab6\_6.cpp (Windows) to learn how to implement the solution to the bounded buffer problem for Linux and Windows platforms

### **Activity 3:** Reader-writer problem

- Run lab6\_7.cpp (Linux) and Lab6\_8.cpp (Windows)
- Then run the modified version of lab6\_7; lab6\_72.cpp and observe the result of the implementation of the first reader-writer problem algorithm in the situation where readers always going in and out of critical section
- Run lab6\_73.cpp which second reader-writer algorithm was implemented and observe the result

### **Activity 4:** Dining-philosopher problem

- Run lab6\_9.cpp (Linux) and Lab6\_10.cpp (Windows) and observe the result
- Insert statement: `randomDelay();` between lines `sem_wait(&chopstick[no]);` and `sem_wait(&chopstick[(no+1)%CHAIRNUM]);` to introduce hold-and-wait situation to the program. Compile the source code and run it again, see what happens and explain why it is so.
- Implement the solution by modifying the source code without removing `randomDelay();` using one of the solution as follows:

- a) Allow one empty seat
- b) Use asymmetric algorithm, by forcing the odd-numbered philosopher to pick the chopstick on the left before on the right, and even-numbered philosopher to pick the chopstick on the right side before the left.
- c) Force each philosopher to pick both chopsticks at a given time. (Once a philosopher picks a chopstick, everybody else has to wait until another chopstick is also picked)