King Mongkut’s University of Technology Thonburi

Faculty of Engineering, Department of Computer Engineering

**CPE 326 Operating Systems, 1/2019**

**PROBLEM SESSION 7: Concurrency and Thread**

**Objective**

The objective of this problem session is to gain experience in writing multi-threaded programs and using a lock.

**Instructions**

Write a multi-threaded program that has a race condition. **Think of your own example** which is different from the example given in the lecture.

Compile and run the program to show that its behavior is not deterministic, i.e. each time you run the program, the results may be different. You may need to force an untimely context switch between threads by calling the yield or sleep function – see example below.

**Untimely Context Switch**

An example of forcing an untimely context switch is shown in Figure 1 for the code given in the lecture. A register is simulated for incrementing the counter variable and a context switch is forced between updating and storing the value of the register.

|  |
| --- |
| int reg = counter;  reg = reg + 1;  yield(); // force context switch  counter = reg; |

**Figure 1. Untimely Context Switch**

You may need to modify this for your own example.

**Submission**

Submit the program code and a report file of examples of your output, using LEB2. This should include the code with the race condition, and a short report (in pdf format) giving some explanation of the code and showing the results of running. Students should work in their existing groups of 3-4 members. Please identify the members of the group clearly in your submission. Please submit the file under the name PS7\_x.pdf where 'x' is the group number used to register your group in google sheet.