

## CS4023 – Lab Exercise, Week 9

### Peterson's Algorithm

---

The goal of this exercise is apply the Peterson's algorithm for synchronizing two concurrently running threads.

---

**Step 1.** Download the file **example3.c** from the module's website (find it in the folder for week 9).

**Step 2.** Open **example3.c** (with emacs, for example). This file is a modified version of **example2.c** from the previous lab exercise. Two Linux threads are created, one is the *producer*, and the other is the *consumer*. The producer stores data in the shared array **buffer**, and the consumer reads from the array.

**Step 3.** Execute **gcc example3.cc** at the command line to build the executable **a.out**. Then run the executable, i.e. execute **./a.out** a few times (say 10 times one after another) at the command line.

**Q1.** What do you observe when you run **a.out** several times?

**Q2.** Identify the critical section in each of the two threads.

**Step 4.** Modify the code to employ the Peterson's algorithm for synchronizing the two threads. Rebuild the executable **a.out** and run it again.

**Q3.** What did you add to the code?

**Q4.** What is the result of executing the modified version?

**Q5.** Can you apply the Peterson's algorithm to the code from the previous lab exercise, i.e. **example2.c**? Why the code had to be modified to the version in **example3.c** before applying the Peterson's algorithm?

**IMPORTANT:** In your own time write a report that describes your answers to Q1-5. This will become part of your end-of-semester project.