



Due: Midnight, March 24, 2020

For each of the following problems I am asking for a small “C” program to be uploaded as part of the usual compressed archive. Include a makefile to build the programs and a readme.md describing your programs.

1. Broken Code

In the `Broken.c` file there are two (at least) problems with how concurrency is implemented:

- a) What is one problem?
- b) What is the second problem?
- c) How would you fix the code? Upload a corrected file (`Unbroken.c`) that has been corrected with comments describing your correction.

2. Double Linked List

In class (and the book) there is an example of a thread-safe linked list. The example is for a singly linked list. Produce a new version of the example that uses a doubly linked list (`Double.c`). I’ve provided a starter `Double.c` file with the definitions of the list (minus any locks, control variables or semaphores required). The starter file also includes skeletons for the functions you will need to write as well as providing some sort of main function to test your list. Upload the completed `Double.c` file.

3. Multiple Wait.

It’s frequently necessary to wait on more than a single event and be able to tell which event was the one that unlocked the thread. Using a control variable, construct a program that:

- a) Starts up two threads
- b) Each thread will signal the main task that it’s done something.
- c) The main program will wait for the signals. After each is received, it will print out which thread had signaled.

See `Waiting.c` in the `ControlVariables.zip` file from the March 3 Daily Slides module for an example of waiting for a single event. Upload your solution as `MultipleWait.c`. The code should compile but I do NOT expect a fully functioning example, it doesn’t have to run, just compile. That said, I would recommend testing it anyway.