**114 Project**

**Description**: in this project, you will implement a bounded buffer queue (BBQ) based on Fig. 5.8 in the textbook, where lock variable is used to synchronize the access of queues. 20 threads are created with 10 producers and 10 consumers. The size of the queue MAX should be a constant and can be changed (at of time of testing). Your program should display the following events on screen:

* Thread creation
* Item ID # produced by thread number #
* Item ID # consumed by thread number #
* Waiting to produce by thread number #
* Waiting to consume by thread number #

Your program should run with a command "project1 TP TC, where project1 is the executable program, TP represents the **"initial"** sleep time range limit for producing threads and TC represents the sleep time range limit for consuming threads.

**Note**: Producers and consumers can control their speed of producing/consuming by changing the sleeping time between two consecutive produce/consume operations. The sleeping interval *t* should be a random value within a range (0,T). For producers, it should dynamically change its speed of "producing" by changing its sleep time range TP as follows: it should gradually slow down the producing when the buffer is over 75% occupancy and stop producing (with condition variable) when buffer is 100% full. Similarly, it  should gradually accelerate the producing when the buffer is below 25% occupancy and reaches twice the initial average speed when buffer is empty. For consumers, it should maintain the same range TC and stop consuming (with condition variable) when the buffer is empty.

**Testing**: Your program should run forever until the user quits it using CTRL + C in the command line or kill the process. Different combinations of (TP,TC) will be used to evaluate the correctness of your output.

**Cygwin** compilation: "g++ -std=c++11 sample.cpp -lpthread"

**Project Report**: a max of 3-page brief description of the results on the comparison of number of times the buffer reaches full and empty status by varying:

* TC values
* TP values
* Buffer sizes (25, 50, 100)

Results should be summarized in excel charts or tables with brief justification.

**Submission**: One zip file containing the following should be submitted:

* All source code
* Project report (3-page max)
* README that "clearly" specify
  + Running environment: Cygwin, Linux, Mac
  + Compiler: C++11
  + commands that compile and run your program
  + Link of a brief video that demos the execution of your program