Assignment 2

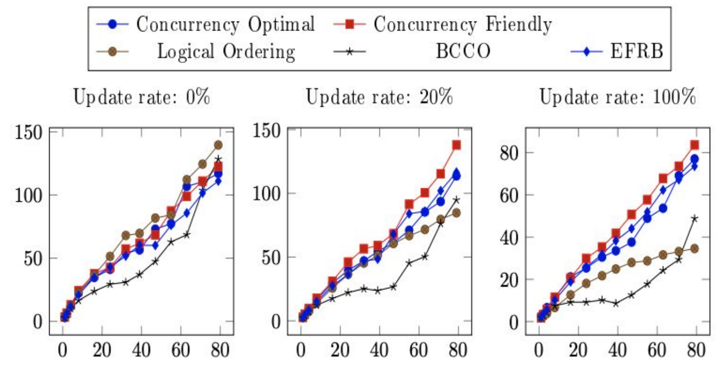
Instructions:

- You may put your doubts here
- You have a provide a makefile for each question
- The entire code should be reproducible. A single make command should be able to generate the final submission.
- You could dump all the data in CSV and use Python Matploltib for the plots. However, this should also be a part of the Makefile. Alternatively, you could use a tool like GNUPlot too or any other C plotting library.
- The code should be well commented with docstrings for each method.
- The code should be modular

1. Develop a trie (refer:https://en.wikipedia.org/wiki/Trie) in C.
    a. Verify that autocomplete and basic functionalities work correctly. [1 marks]
    b. Now, make this a thread-safe data structure. Compare the performance of lockings techniques like single locking [1 marks], Readers–writer locking [1 marks] and hand-over-hand locking [2 marks] against these parameters [2 marks for plot],
    - Number of concurrent threads.
    - Different types of workload,
      1] Write intensive workload
      2] Read intensive workload
      3] Mixed Read / Write workload (50%-50%)

Resources:
Chapter 29: Lock-based Concurrent Data Structures :
http://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks-usage.pdf
Different locking techniques for linked list :
http://www.cs.technion.ac.il/~erez/courses/seminar/talks/05.pdf

Reference for plot : Page 9 of this paper.



2. Implement approx. LRU in C [1.5 marks]
    a. Compare its performance on various loads (80-20, looping, random) with random, FIFO, LRU exact (all implemented in C) [3.5 marks]
    - The Python implementation of the above is here. You will implement in C.
    - You will need to compare the performance of each policy. Particularly, state the time complexity and space complexity with respect to the data structure you would use to implement them. For example, FIFO requires a queue data structure (or a linked list). Does the random policy require O(1) time? Why or why not?
    - Generate Figure 22.6 (for no locality workload), Figure 22.7 (80-20 workload), Figure 22.8 (looping sequential) and Figure 22.9 (Approx LRU compared with others). These figures are here.