Project 1b Report

Group: Kelly Wang, Jonathan Day, Ajit Vijayakumar

Work Distribution:

- Jonathan and Ajit worked on task 1 & 2 (with debugging). Also helped Kelly with debugging task 3.
- Kelly worked on task 3
- We felt this work distribution was fair because task 1 was relatively simple and task 2 required actual implementation. The combination of task 1 and 2 felt equally challenging as task 3. While we cannot really divide the work 50/50 straightforwardly, team members showed an eagerness to help one another. In the end, we feel that we all equally understand each task–which is the most important part.

Design Explanation:

- **Task 1 :** Used the declared term in the provided hashchain.h file to revise the #ifdef blocks in hashchain.cc as well as initializing each element of the array for locks, rwlocks, and sem. There wasn't much of a design element, rather it was mostly fill-in the blanks.
- **Task 2:** Implemented the functions for Lock class in synch.cc. We looked at the private fields of the Lock class to get an idea of what we had to work with as well as what we had to initialize in the constructor. Then for Acquire(), if value == 1 then some thread has the lock and it would do the same thing as P() in semaphores since we need to tell the threads to wait. Similarly, if the lock is released, it's similar to the semaphore signaling that it's OK to resume.
- **Task 3:** We designed our condition variables to work around the lock so implementation is quite similar. Then for RWLock.h/.cc we just swapped out the pthread functions from project 1a to use the nacho threads we had implemented in task 1 and 2.
- **Overview:** Most of the design was just built around what would "logically" happen for each scenario along with reading some other provided files to get a scope of what we had to work with.