Name: Riley Lawson

Section:

University ID: 116555487

**Programming Project 2 Report**

**Summary (10pts):**

Throughout this project, I have learned a lot while alos struggling to understand specific concepts. This project really put me to the test to be able to understand mutex locks, deadlocks, structs, functions, how to output a file, and understanding how to use proper terminal syntax. However, throughout my time testing and trying to figure all this out, I couldn’t manage to a proper output. I put fflushes in and even tried to free memory with free. At this point, I am unsure on what I managed to do wrong and why the runtime is so disfunctional. I’m assuming I am not properly writing the information to the file, but am uncertain at this time. Otherwise, I have learned that deadlocks can be extremely frustrating and C as a language is extremely hard to debug.

S

**6.2:**

**(5pts) Average runtime for each program (use the “real” time)**

Did not receive a proper runtime output...

**6.3:**

1. **(3 pts) Which technique was faster – coarse or fine-grained locking?**

Course-grained

1. **(3 pts) Why was this technique faster?**

Because a single lock can be held through an entire operation faster than continuously locking and unlocking a mutex lock.

1. **(3 pts) Are there any instances where the other technique would be faster?**

If you were doing significantly larger programs fine-grained can be much faster because it would allow a lot tighter control of each individual lock being done.

1. **(3 pts) What would happen to the performance if a lock was used for every 10 accounts? Why?**

It would slow performance down, because it would have to check if the lock was unlocked and if the lock was locked everytime it would go to run through 10 more accounts.

1. **(3 pts) What is the optimal locking granularity (fine, coarse or medium)?**

It all depends on how long/complex your program is and whether or not you require numorous locks within certain places. For example, if you have a large program and use course locking, then certain threads won’t finish in time for another to unlock and might take signifantly longer elsewise. However if you use fine grain locking getting individual things done at a rapid pace it could leave you with a consistent output and a fast operation. So medium might be the best overall/optimal option, however there are circumstances that have better uses for either one or the other.