Name: Kent Mark
Section: 5
University ID: 583972417

Project 2 Report

Summary:

10pts

Overall, this project helped give me a better understanding of threads. It also helped reiterate why threads are used in processes and how the use of threads can speed up code execution. This project also helped me get some more practice with structs and pointers. This project also gave me a neat overview of how bank systems work. Finally, I learned the difference between fine and coarse grained locking, which helped me enhance my understanding of mutex locks.

Part II:

6.2:

5pts Average processing times for TRANS and CHECK requests (from test script):

Average processing time for coarse program was 1:07.66.

Average processing time for fine program was 1:11.85

6.3:

3.2.1:

3pts Which technique was faster - coarse or fine grained locking?

The coarse grained locking was faster.

3pts Why was this technique faster?

Since the time to lock all the accounts for a transaction was omitted, more CPU time was given to the threads making the overall run-time faster.

3pts Are there any instances where the other technique would be faster?

Fine grained would be faster if there were many threads all trying to access the same memory location since it much more efficient accessing memory.

3pts What would happen to the performance if a lock was used for every 10 accounts? Why?

Unless all the locks are used constantly there would be a significant time delay and memory space would be wasted

3pts Discuss the probable "optimal" locking granularity (fine, coarse, or medium)?

Medium would be the optimal locking granularity because system delays for user account access and lock processing would be consistent across the board.