

Name: Zach Scurlock
Section: 1
University ID: 852478643

Programming Project 2 Report

Summary (10pts):

While working on this project, I was able to gain a better understanding of threads and how to implement them in C. I was able to grow my knowledge of mutexes and how threading really works. I didn't have many problems other than a few segmentation faults and malloc errors, but after some trial and error, I got those ironed out. Another thing this project helped me get better at was using structs in C.

6.2:

(5pts) Average runtime for each program (use the "real" time)

Fine-grained average runtime: 1 minute 3.896s

Coarse-grained average runtime: 1 minute 3.795s

6.3:

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

Fine-grained locking was faster.

2. (3 pts) Why was this technique faster?

Fine-grained is faster since multiple threads are able to do TRANS and CHECK requests at the same time, while coarse-grained locks the whole bank every time, even when one account needs to be accessed.

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

No, because coarse-grained and fine-grained could be the same speed if every request is blocked by a mutex in fine-grained locking.

4. (3 pts) What would happen to the performance if a lock was used for every 10 accounts? Why?

It would most likely decrease the performance from fine-grained locking, but it would be faster than coarse-grained locking because you are still locking up more accounts than you really need to for each access.

5. (3 pts) What is the optimal locking granularity (fine, coarse or medium)?

Fine-grained locking is the optimal locking granularity.