

A Manual Test of the Assignment 2 concurrency requirement

The tests in the assignment 2 test driver only test for race conditions and deadlock. We were not able to come up with a reliable automatic test of the concurrency requirement. You will need to decide if your scheme meets of the concurrency requirement.

To help guide you in judging your concurrency, here's a manual thought test you can perform against your implementation.

Assume you have six accounts (Xb1, Yb1, Zb1, Ab1, Xb2, and Yb2) that are all different from each other. The accounts ending in b1 are in branch 1 and those ending in b2 are in branch 2.

The first requirement from the assignment document was:

1. *Branches operate independently from one another so worker threads should be able to operate concurrently on accounts in different branches. Similarly, the branch managers should be able to concurrently compute their branch's balance.*

In order for two commands to proceed concurrently, they can not run through the same critical section. This means that commands running concurrently can not access the same lock. If you examine the locks that are accessed under a withdrawal from account Xb1 command and the locks that are accessed under a withdrawal from Xb2 command, you should find no lock in common between the two commands. Make sure this is true for the withdrawal, deposit, and transfer commands. For example, a transfer command between Xb1 and Yb1 should use a different set of locks than a transfer between Xb2 and Yb2.

A branch balance command on branch 1 should not use the same lock as a branch balance on branch 2.

2. *Transfers between different accounts in the same branch should be able to proceed concurrently.*

The commands "Transfer from Xb1 to Yb1" and "Transfer from Zb1 and Ab1" should acquire no lock in common.

3. *Although computing the bank and branch balances when multiple workers are operating will produce different results depending on the exact interleaving of the balance command with respect to the other commands in the system, your answer to these balance requests must correspond to some possible interleaving of the commands. In other words, the balance command results won't necessarily match the single worker run but should match some single worker run with possibly the commands reordered. To achieve this property the balance commands can not include the results of a halfway completed command by some other worker.*

We talked about this in the discussion section. The balance command shouldn't include a partial update of a command. For example, if you do a transfer between Xb1 and Xb2, you shouldn't get a bank balance that doesn't include the in flight money between Xb1 and Xb2.

If your approach passes this manual test and doesn't have races or deadlock problems, congratulations you are done. If not, think about the locking you will need to pass the test.