

Multithreaded_Bank

A multithreaded bank application

EXTRA CREDIT

Extra credit can be found in server2.c. It forks upon client connection and uses mmap and shared memory.

Client

This is a command line program that takes 1 argument To correctly connect to the server, type the address of the machine it's running on.

By default, no argument will attempt to connect to "localhost" This was done to make life easier for testing, but is also helpful because the majority of the time the client and server will be on the same machine.

If no server is found, the client will attempt to reconnect every 2 seconds.

The client has two threads, one for receiving and one for sending. Throttling is done client side. There is a two second delay after sending a command to simulate a large amount of users. Throttling is done after the input is sent, so any commands will be sent instantly to the server.

CAUTION: Improper closing of the server can sometimes cause the client to not find the server. To fix it, close your terminals and restart the client and server. This should not occur with proper start and shutdown. The server can be properly closed and shutdown using control c or simply closing the terminal.

Server

This is a command line program that takes no arguments. Simply run it, and then keep it in the background Connections to the server will generate a new client thread. Initial start of the server will generate a thread for printing accounts.

Server2

Code between Server and Server 2 are nearly identical. Server 2 though forks on a client connection. The account list is mmaped in shared memory so that the server forks can read and write to it. The number of accounts is also mmaped. Client connections will cause the server to print the PID of the new process that was generated.

open

Starts a new account. Account name must be under 100 characters. You cannot have more than 20 accounts at any one time Accounts get deleted on server closing. Accounts can only be added one at a time.

You can either type

open accountname together as one command

or

open

accountname as two commands

start

Starts an account session for a client. Can only start one account session per client per account This locks the inuse variable. The locking of the variable can only be done by one thread at a time.

You can either type

start accountname together as one command

or

start

accountname as two commands

debit

Removes money from an account. Cannot remove more than what exists

credit

Credits money to an account

exit

Finishes the session cleanly and exits all threads. Better than closing the terminal though there is a signal handler that should handle closing a terminal unexpectedly and other things like Control C

finish

Exits the current account, allowing for a new account to be opened.

