

Michelle Chen mc1481
Emily Ng en140
PA5: ReadMe

Design:

For our server program, our session-acceptor and client-service algorithms run as separate processes. The initial parent process spawns a child process every time a successful connection is made with a client.

The bank consists of a database struct which holds 20 account structs which each contain an account name of up to 100 characters, a balance, and an in-session flag.

The bank is initialized in shared memory at the beginning of the program and the shared memory is cleaned up when the parent process terminates on SIGINT. To deal with deadlocks and race conditions, we used semaphores that functioned as mutexes. Each account holds a semaphore. To access and change the information for an account in the bank, a child process must attach to the bank shared memory and wait for the semaphore in `start()`. The process calls `sem_post()` and detaches from the shared memory when it is done. Our bank database also uses a semaphore so that new accounts cannot be opened while the server prints out account information every 20 seconds.

Our multithreaded client program attempts to connect to the server every 3 seconds and prompts the user for input once successfully connected. The input thread gets user commands while the output thread displays messages from the server.

If the server disconnects before the client, the client receives a message and terminates.

Idiosyncracies:

Account names are case sensitive so it is valid to have both accounts BKR and bkr.

The debit function checks that the debited amount is not larger than the current balance before allowing action so the user cannot go into debt.

IN SERVICE only prints when the account's in-session flag is 1. Also it prints in red for legibility.

Extra Credit:

For extra credit, we implemented a multiprocessed server that uses shared memory where each client connection is a new process. Server prints the `pid()` and `ppid()` for each fork to synchronize viewers.

Also, instead of trying to silently block while trying to start a customer session, our bank tries to lock the semaphore for the account every 2 seconds and sends a wait message to the client if the lock fails.