Report - Home Banking System

Claúdia Martins, Joana Ferreira, João Matos
Operating Systems

Messages

We are passing the messages as byte streams, in the TLV format, as suggested by the specification.
We draw your attention to our reply messages, which implement variable length, something we wished to explore, along with serialization to readable text, but could not due to time constraints.

Synchronization

1. **Bank offices accessing accounts' array (several threads accessing the same data structure):**

   For this, we implemented a **mutex** for each account. The accounts array (global variable *srv_accounts*) is actually an array of *srv_account_t*, which 'inherits' *bank_account_t* and also has a mutex.

   When a thread performs an operation over *srv_accounts*, it should only lock the mutexes associated with the accounts involved, therefore allowing several threads to be working in the same array simultaneously, without overlapping.

2. **Bank offices receiving requests from main (Producer–consumer problem)**

   Producer → main
   Consumer → bank offices

   To solve this problem, we implemented a **queue** of requests (*srv_request_queue*) as a global variable with **semaphores** (*srv_request_queue_empty* and *srv_request_queue_full*), where *main* waits for the queue to have space before trying to push requests. Simultaneously, bank offices wait for the queue to have requests and after satisfying one, the queue is set to empty again.

Server shutdown

After the shutdown is requested, it is checked if it was requested by the admin. If so, the server FIFO closes, so there is not possible to send requests anymore. After that, all the threads created for the bank offices are terminated and the program ends.