buyer_seller testplan

WO1 Clayton E. Williams

December 2023

1 Purpose

buyer_seller is a two-part project, consisting of a buyer (client) and seller (server). The client reads a data file of 50000 entries, consisting of an account number and order or payment, and sends it to the server through the use of Unix Domain Sockets. The server receives these transactions, and through the use of multi-threading, consolidates data from all 10 clients into a single account structure. The purpose of this test plan is to provide the commands for executing automated tests to test against file validation, and to provide a framework and sequence of manual tests in the form of bash commands to test the various scenarios the program could run and execute under, to ensure the program exhibits desired behavior, has no memory leaks, and does not crash on unexpected or invalid input.

2 Components

buyer_seller contains both automated and manual testing.

2.1 Automated Tests - Test Suite

Automation of unit testing is provided through the make tool and check library, and can be run from the command line as:

```
<project_root>$ make check
```

There is only one test suite for buyer_seller, test_client_helper, which tests the validate_file function against valid and invalid files, directories, and files of size 0.

2.2 Manual Testing - Valgrind

Manual testing of the program is to ensure that there are no memory leaks reported by valgrind, regardless of program termination status.
There are 6 scenarios in which to test with valgrind.

- client no args
  This scenario should test that running the client without supporting file argument causes the program to exit normally.

  ```
  <project_root>$ valgrind ./client
  ```

- client invalid file arg
  This scenario should test that running the client with invalid file arguments causeses the program to exit normally.
  A list of invalid files can be found in test_client_helper.

  ```
  <project_root>$ valgrind ./client <INVALID_FILE>
  ```

- client no server
  With the server not running, running the client with a valid file argument should result in normal termination due to failure to connect to server.

  ```
  <project_root>$ valgrind ./client <VALID_FILE>
  ```

- server with interrupt The server is designed to accept a maximum of 10 client connections, and will continue to listen until 10 connections have been accepted. To test this scenario, run the server, and without establishing a client connection, send an interrupt signal (ctl+C) to the server while it is waiting on additional clients.

  ```
  <project_root>$ valgrind ./server
  <project_root>$ <ctl+c>
  ```

  Additionally, test against signal interrupt after any number of connections less than 10 have been established. To do this, run the server with valgrind as before, and initialize any number of connections (less than 10) and sent signal interrupt while the server is still listening.

- client and server run to completion
  This test tests the scenario of a server initialized, with 10 client connections that run to completion. To run the program, first run the server:

  ```
  <project_root>$ valgrind ./server
  ```

  Once the server is running, the bash command:

  ```
  <project_root>$ for i in {1..10}; do ./client data/seller$i.dat & do
  ```

  Will execute 10 client connections in the background, each with one of the provided data files in data/. The client has a built-in sleep function after each 10000 entries to ensure that each client is run concurrently in its own threaded process.