Program 2: CIS 427

Parashar Parikh

Ibrahim Ammar

Zeineb Moalla

11-20-2022

## Introduction

This README file servers as a way to show how to build my program. The program uses [Boost C++ Asio Library](https://www.boost.org/doc/libs/1_80_0/doc/html/boost_asio/overview/networking/bsd_sockets.html) which is used for my socket implementation. The libraryprovides low-level socket interface which are based of Berkeley sockets API. Now that we know a bit about boost, let me show ways you can test my program.

1. Testing on the University of Michigan servers. Since boost libraries are not in the C++ standard libraries yet, they will need to be build. It took me about 5 to 10 minutes to build the boost libraries. to get my program tested on the University servers.
2. Testing on my own linux server. Since this is my own machine, prior to me submitting the assignment have already installed and built the boost libraries on my server. I have also created a temprory user which you can use to login to build and test my program.

## Steps to Build Boost in Umich Servers

1. First we need to create a directory and download boost there.

mkdir boost  
cd boost/

1. The next step now is to download the boost boost\_1\_80\_0.tar.gz and extract it to that directory

wget https://boostorg.jfrog.io/artifactory/main/release/1.80.0/source/boost\_1\_80\_0.tar.gz  
tar -vxf boost\_1\_80\_0.tar.gzls

1. Now we will go into the the and start the building process. NOTE: It will take a while please be patient.

cd boost\_1\_80\_0  
./bootstrap.sh --without-libraries**=**python  
./b2

1. After you are done with building, you should see the following message. In the images directory

knitr::include\_graphics

1. Now add these paths into your .bash\_profile by following the steps below.

nano ~/.bash\_profile  
export BOOST\_ROOT**=<**Path\_to\_boost**>**/boost\_1\_80\_0  
export BOOST\_LIB**=<**Path\_to\_boost**>**/boost\_1\_80\_0/stage/lib

1. After pasting these two lines, press CTRL+X -> y -> ENTER to save the file. Now we just need to reload the .bash\_profile by running the command below.

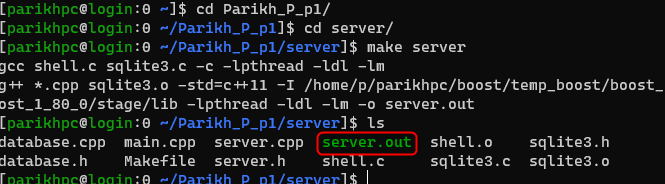
source ~**/**.bash\_profile

1. Assuming you have the program files uploaded to the server, you can build them using the Makefiles available.

## Steps to build client and server

1. To build the server,

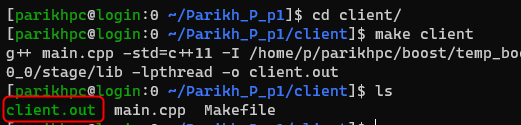
cd server  
make server



This will generate the server.o file in the current directory.

1. To build the client,

cd client  
make client



## Each Student’s Role

Student 1: Parashar Parikh

* Implementation of WHO command
* Implementation of LOGIN command
* Implementation of LOGOUT command

Student 2: Ibrahim Ammar

* Implementation of LOOKUP command
* Implementation of DEPOSIT command

Student 3: Zeineb Moalla

* Worked on Documentation
* Optimized code for new command functions implemented

As a group:

* Added new “if else” statements in “string database::process(string in, int& id)” and added the DEPOSIT, LOOKUP, LOGIN, LOGOUT, WHO commands set to token string.
* Added new “invalid argument” static string such as “crypto\_invalid” setting to “404 Your search did not match any records.”
* Added new public class members “string deposit(string command)” , “string lookup(string command)”, “string who(string command), “string login(string command)”, “ string logout(string command)”.
* Added user\_id to identify which user is logged in due to the list function and and lookup function.

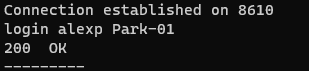
## Bugs in the Code

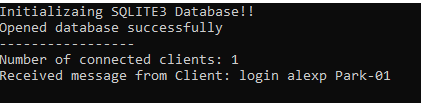
* There are no errors for the code we implemented for Programming Assignment 2.

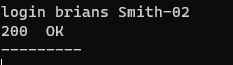
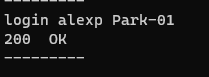
## Table Showing Test Cases with Details

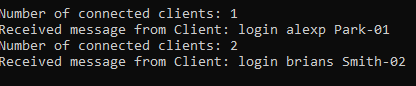
| **Test #** | **Description of test** | **Input Value** | **Expected Output** | **Actual Output** | **Test Pass / Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | LOGIN: test correct user/password | LOGIN alexp Park-01 | 200 OK | 200 OK | Pass |
| 2 | LOGIN: test invalid input | LOGIN root password | 403 message format error  Missing arguments for Login! | 403 message format error  Missing arguments for Login! | Pass |
| 3 | LOGOUT: test valid input of command | LOGOUT | 200 OK | 200 OK | Pass |
| 4 | LOGOUT: test input if user not logged in | LOGOUT | No user currently logged in! | No user currently logged in! | Pass |
| 5 | LIST: test valid input of command | LIST | 200 OK  The list of records in the crypto database for Alex:  1 Bitcoin 19483.23  2 Ethereum 1330.4  3 Tether 1.0 | 200 OK  The list of records in the crypto database for Alex:  1 Bitcoin 19483.23  2 Ethereum 1330.4  3 Tether 1.0 | Pass |
| 6 | LIST: test input if user not logged in | LIST | User not logged in | User not logged in | Pass |
| 7 | WHO: test valid input of command when multiple users arelogged in | WHO | 200 OK  The list of the active users:  root 127.0.0.1  Alex 127.0.0.1  Brian 127.0.0.1 | 200 OK  The list of the active users:  root 127.0.0.1  Alex 127.0.0.1  Brian 127.0.0.1 | Pass |
| 8 | WHO: test valid input of command when user is not logged in | WHO | 400 Invalid Command  Denied: Non-root user detected! | 400 Invalid Command  Denied: Non-root user detected! | Pass |
| 9 | LOOKUP: test valid input of command and crypto name | LOOKUP bin | 200 OK  Found 2 Match!  Binance 282.48  BinanceUSD 1.0 | 200 OK  Found 2 Match!  Binance 282.48  BinanceUSD 1.0 | Pass |
| 10 | LOOKUP: test invalid input of lookup command and crypto name | LOOKUP NOCOIN | 404 Your search did not match any records | 404 Your search did not match any records | Pass |
| 11 | DEPOSIT: test valid input of deposit command | DEPOSIT 150.0 | Deposit Successfully. New Balance $150.00 | Deposit Successfully. New Balance $150.00 | Pass |
| 12 | DEPOSIT: test invalid input, such as negative amount | DEPOSIT -150.0 | 400 Invalid Command!  Insufficient Funds to Deposit! | 400 Invalid Command!  Insufficient Funds to Deposit! | Pass |
| 13 | BUY AND/OR SELL: User can only buy and sell their own crypto | LIST  The list of records in the crypto database for Alex:  1 Binance 282.48  2 BinanceUSD 1.0  3 Solano 33.86  4 Dogecoin 0.06  buy cardano 2 2 | 403 Message Format Error!  The crypto cardano does not exist for the current user! | 403 Message Format Error!  The crypto cardano does not exist for the current user! | Pass |
| 14 | LOGIN: Try to login as the same user for two clent connections | Client 1:  login alexp Park-01  Client 2:  login alexp Park-01 | Client 1:  200 OK  Client 2:  400 invalid command.  DENIDED: You are already logged in somewhere else!! | Client 1:  200 OK  Client 2:  400 invalid command.  DENIDED: You are already logged in somewhere else!! | Pass |
| 15 | LIST: List all the cryptos when the root user is logged in | list | ALL CRYPTO ENTRIES | ALL CRYPTO ENTRIES | Pass |

Test #1 Screenshot (Sever and Client 1):

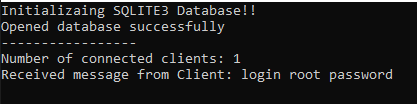
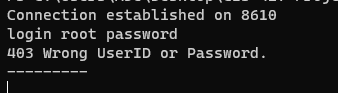




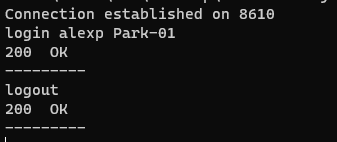


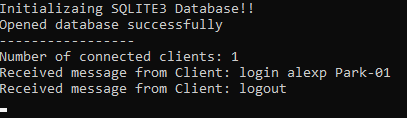


Test #2 Screenshot (Sever and Client 1 and 2):

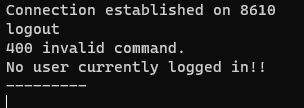


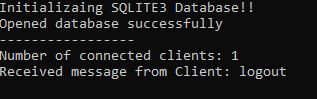
Test #3 Screenshot (Sever and Client 1):





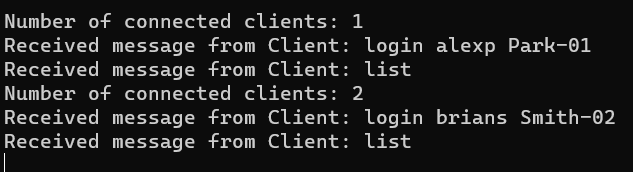
Test #4 Screenshot (Sever and Client 1):



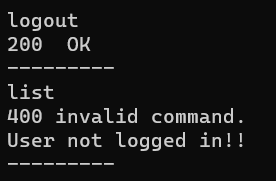


Test #5 Screenshot (Sever and Client 1):



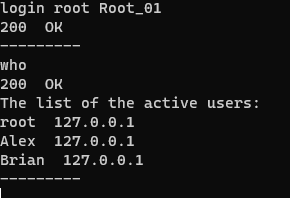


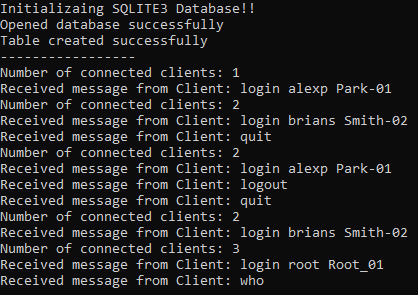
Test #6 Screenshot (Sever and Client 1):



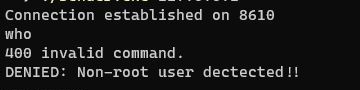


Test #7 Screenshot (Sever and Client 1):



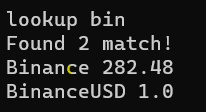


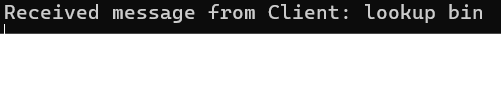
Test #8 Screenshot (Sever and Client 1):

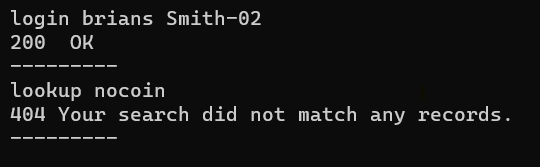
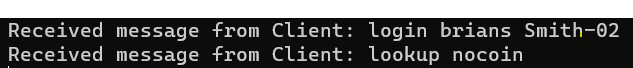




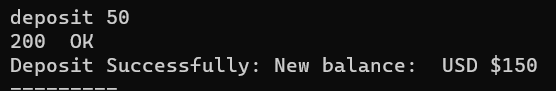
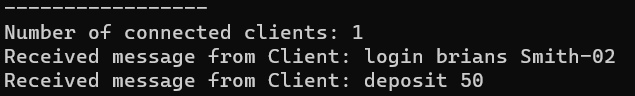
Test #9 Screenshot (Sever and Client 1):



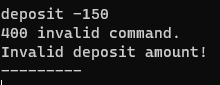


Test #10 Screenshot (Sever and Client 1):  


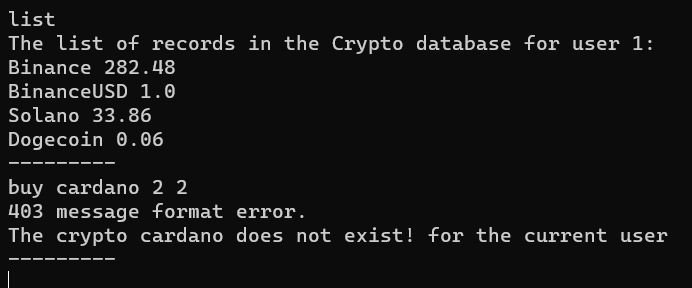
Test #11 Screenshot (Sever and Client 1):

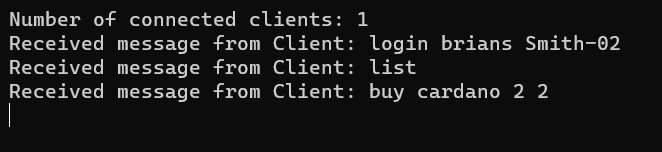
  


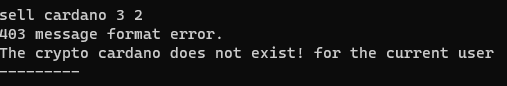
Test #12 Screenshot (Sever and Client 1):



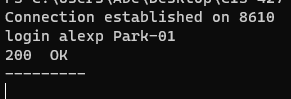
Test #13 Screenshot (Sever and Client 1):

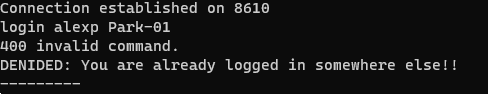
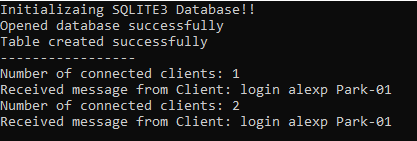






Test #14 Screenshot (Sever and Client 1):





Test #15 Screenshot (Sever and Client 1):

