

Socket Programming Phase 2

公衛三 陳佳雯 b07801004

Environment

Ubuntu 20.04.1 LTS (on Windows 10 WSL)

Language

c++, c

Compile & Execute

Server

1. In terminal (Ubuntu \ windows Powershell wsl), type in `cd /[file's name]` and then type in `cd myserver` to change direction to file directory.
2. Type in `make` command, and then `server` binary file will appear.
3. After successful compilation, type in `./server` command.
4. Follow the instructions on the screen.

Feature

1. Client will wait in queue when all the threads are occupied.
2. If any threads are available, clients in the waiting queue can go into thread pool and be served.

System design

Const.h

- Define some common constants

```

#ifndef _CONST_H_
#define _CONST_H_

#define MAX_STRING_LENGTH 200
#define LISTEN_BACKLOG 10

#define CSTRING_MAX_SIZE 500

#endif

```

ThreadPool.h

- Also called worker pool
- Have multiple threads to handle multiple clients

```

class ThreadPool{
private:
    int threadNu
    pthread_mutex_t mutex;
    pthread_cond_t condition_var;
    pthread_t* threads;
    queue<Client> waitingQueue;
    static void* work_static(void* thread);
    void work(void* thread);
    void workFunctions(Client aClient);
    UserList userMap;
public:
    ThreadPool();
    void connect(Client &c);
    void createThreads();
};

```

User & UserList.h

User:

- The class stores user's information

UserList:

- A list to store registered users
- Provide functions to deal with "Register", "Query", "List", "Microtransaction"

```

class User{
public:
    User();

```

```

    User(string name, int money, string IP, string port, bool isOnline);
    string getName();
    string getIP();
    string getPort();
    int getMoney();
    bool getIsOnline();

    void setName(const string name);
    void setIP(const string IP);
    void setPort(const string port);
    void setMoney(const int money);
    void setIsOnline(const bool online);
    friend class UserList;
private:
    string port;
    string IP;
    string name;
    int money;
    bool isOnline;
};

class UserList{
private:
    vector<User*> usersListPtr;
public:
    int getOnlineNum();
    UserList();
    User* findUser(string name);
    bool registerUpdate(string clientMessage);
    User* loginUpdate(string clientMessage, Client aClient);
    string list(User* aUser);
    bool exit(User* aUser);
    void transact(string clientMessage);
};

```

ClientClass.h

- Store client's information

```

class Client{
private:
    string IP;
    SSL* clientSSL;
public:
    Client();
    Client(string, int, SSL*);
    int sockfd;
    string getIP();
};

```

Server's Reply Message

Register

- **100<space>OK<LF>** : Register Success
- **210<space>NAME_EXIST<LF>** : User name existed

Login

- **<accountBalance><LF>**
<number of accounts online><LF>
<userAccount1>#<userAccount1_IPAddr>#<userAccount1_portNum>
<LF>
<userAccount2>#<userAccount2_IPAddr>#<userAccount2_portNum>
<LF> ... : Login Success
- **220<space>AUTH_FAIL<LF>** : Account haven't registered
- **230<space>HAVE_LOGGED_SRY<LF>** : Account has been logged in

List

- **<accountBalance><LF>**
<number of accounts online><LF>
<userAccount1>#<userAccount1_IPAddr>#<userAccount1_portNum>
<LF>
<userAccount2>#<userAccount2_IPAddr>#<userAccount2_portNum>
<LF> ... : List Format

Logout

- **Bye<LF>** : Logout Success

Exception Handling

- The server will automatically logout user if the connection terminates unexpectedly

References

Jacob Sorber Tutorials:

- Multi-thread Server : https://www.youtube.com/watch?v=Pg_4Jz8ZIH4&ab_channel=JacobSorber
- Thread Pool, Mutex: https://www.youtube.com/watch?v=FMNnusHqjpw&t=156s&ab_channel=JacobSorber
- Conditional Variables: https://www.youtube.com/watch?v=P6Z5K8zmEmc&t=284s&ab_channel=JacobSorber
- How to get ip address from sock structure in c?
<https://stackoverflow.com/questions/3060950/how-to-get-ip-address-from-sock-structure-in-c>
- pthread
<https://stackoverflow.com/questions/1151582/pthread-function-from-a-class>