**Project Report**

Team member: Li Jundong/Florian Prigent

Github address：https://github.com/Gavinaa123aah/socket

1. Abstract

This is a Socket(C Language) project based on TCP protocol. It implements the following functions:

1. Possibility to connect client/server in TCP
2. Possibility to send and receive message between server-side and client-side
3. Possibility to send and receive files between server-side and client-side
4. Possibility to Multi-threads many clients for a server
5. Implements login function based on sqlite database
6. Function Details Introduction（Screenshots）
7. The below screenshots shows some import files and directory in the project.

Server\_dir: this is server-side home directory

Client\_dir: this is client-sise home directory

Muti\_client.c: this is client-side source file

Muti\_server.c: this is server-side source file

Muti\_clien: this is client-side Executable file

Muti\_server: this is server-side Executable file



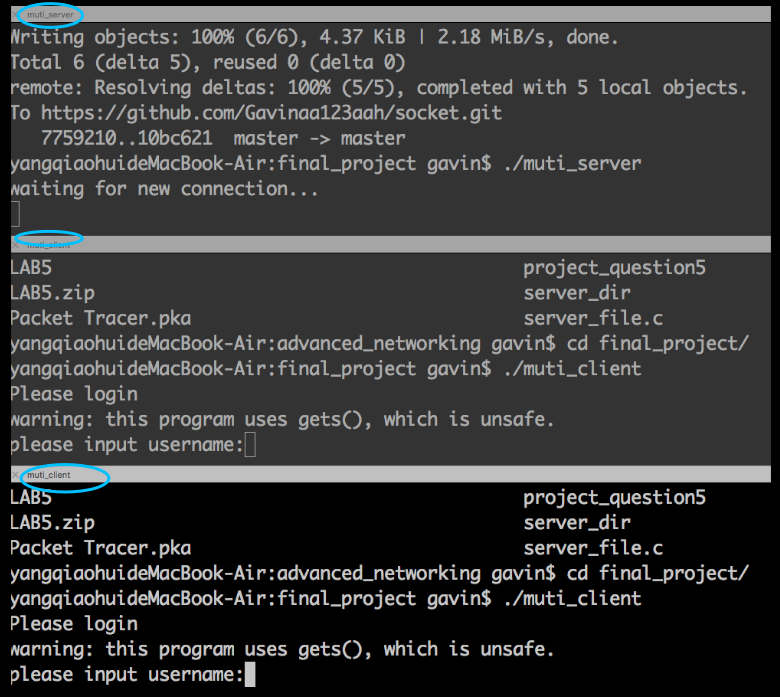
1. The below screenshots shows one server and two clients. these two clients connect the same server at the same time.

The server compile command:

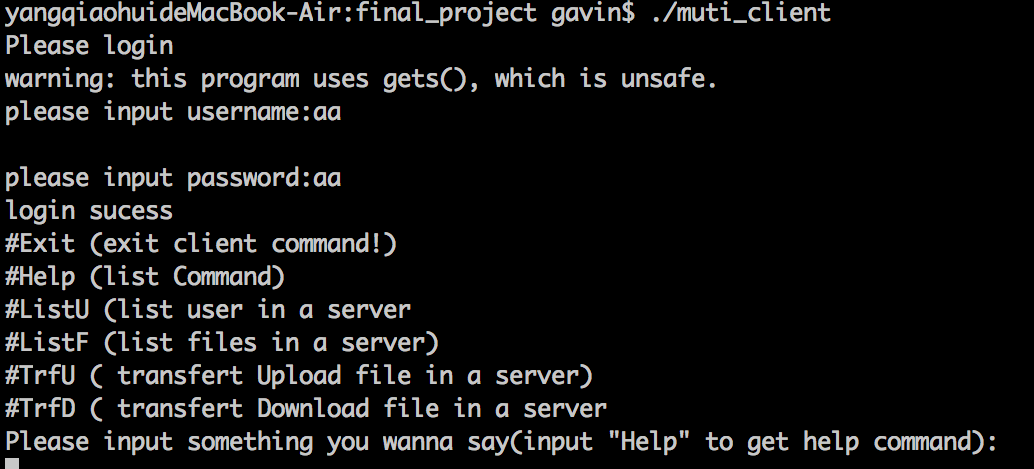


The client compile command:

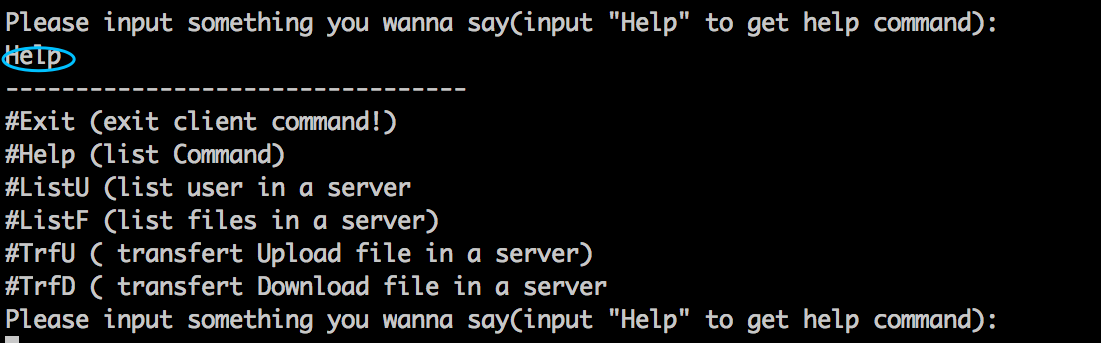




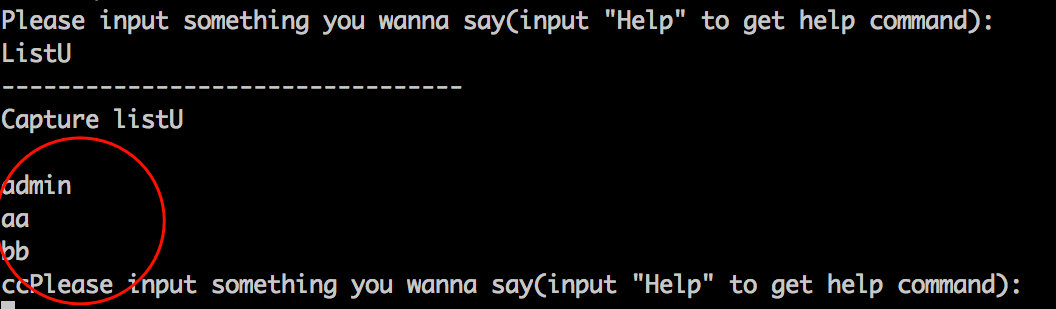
(3) The below screenshots shows that after running client-side, client need to input username and password to finish login(username: aa, password: aa)



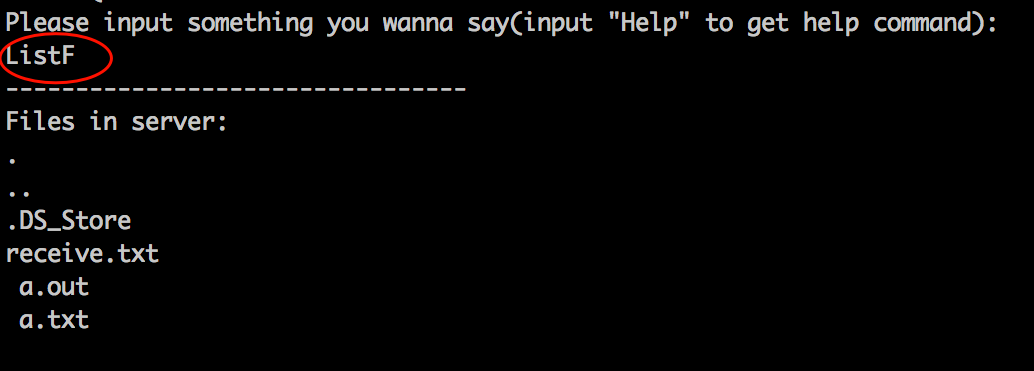
(4)The below screenshots shows that possibility to input “Help” to search related commands.



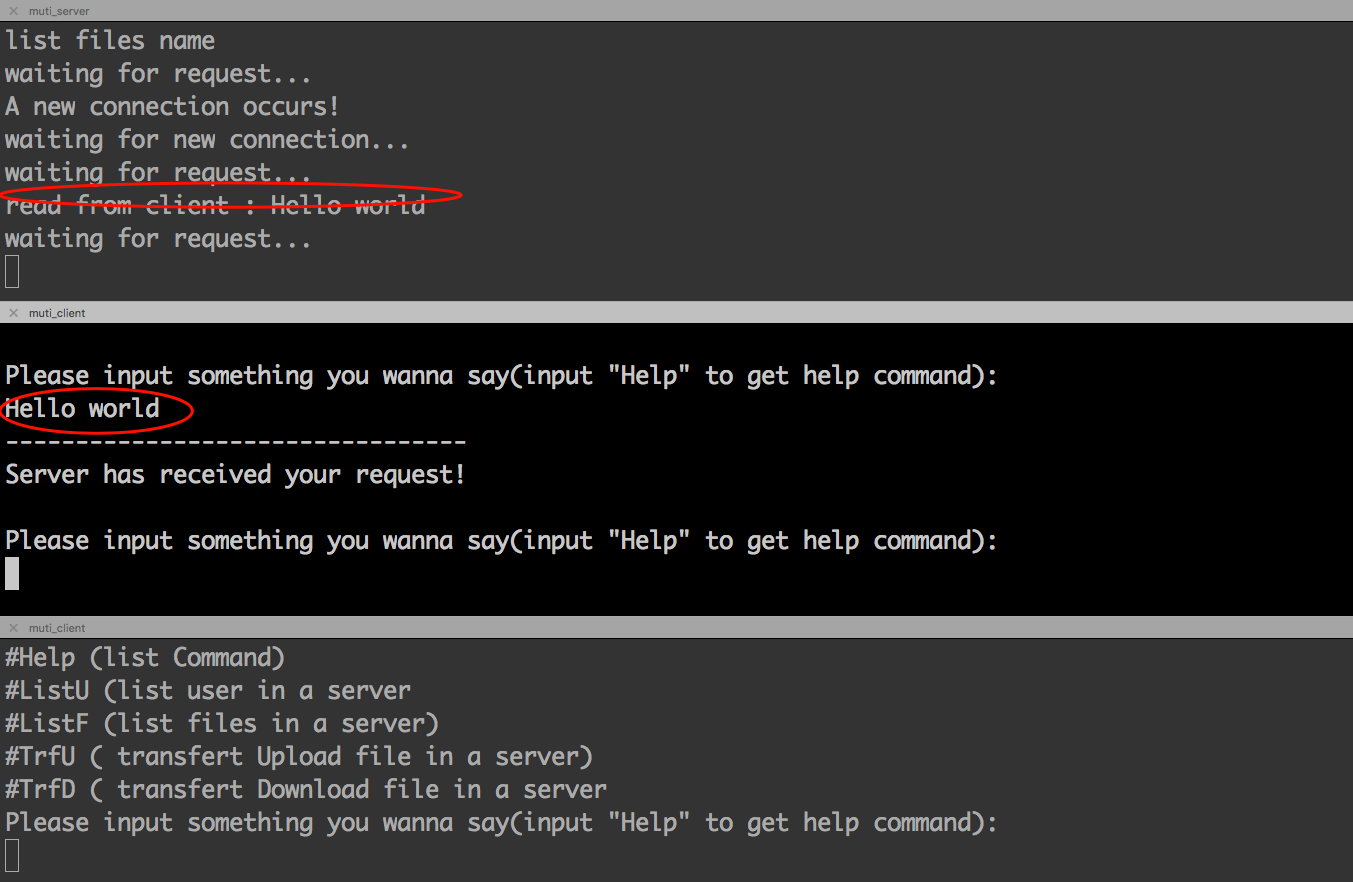
(5) The below screenshots shows that possibility to input “ListU”to view all users in data system.



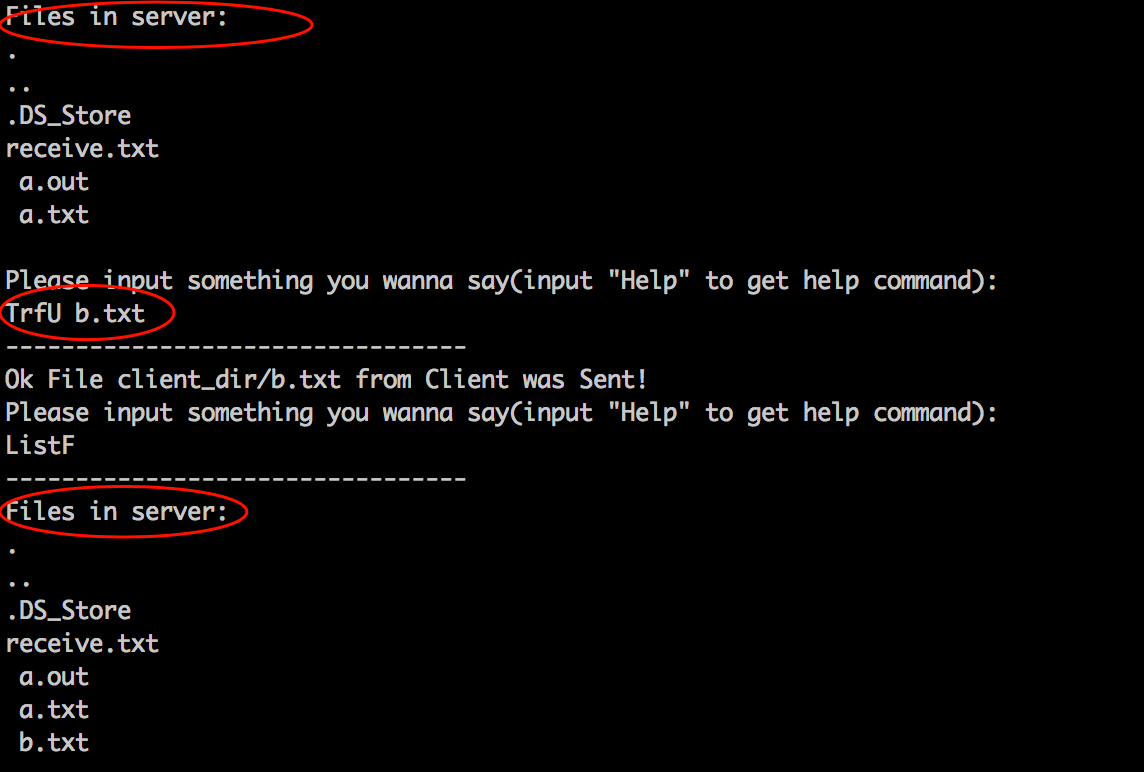
(6) The below screenshots shows that possibility to input “ListF”to all files in server directory



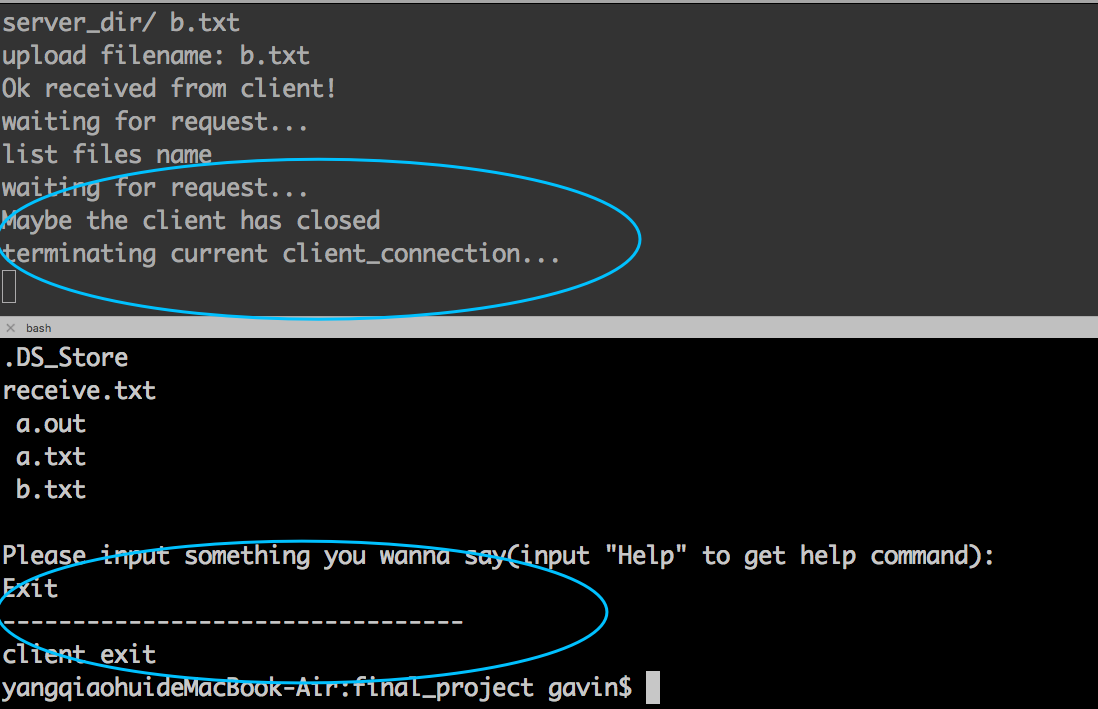
(7) The below screenshots shows that server-side can receive “Hello world” message when one of client-side send “hello world”message to server-side. And client-side can receive message “server has receive your request!”which comes from server-side.



(8) The below screenshots shows that when client-side input command “TrfU a.txt”,the server-side can receive a.txt file from client-side. Now, we can see that a.txt file has been saved into the server-side home directory (we can view the all files in server-side home directory by taping command ‘ListF’).



(9) The below screenshots shows that we can exit the client-side when we input command ‘Exit’. And the server-side can display some information which is related to client-side exit.



1. Key Library Function and coding details:
2. socket creation（server）

AF\_INET:IPVE

SOCK\_STREAM:TCP



1. Bind address and port(server)

Sockfd\_server: socket which was created just now

S\_addr\_in:struct which include ip address and port

Sizeof(S\_addr\_in):struct length



1. Listen port(server)

Sockfd\_server: socket which was created just now

MAX\_CONN\_LIMIT: max length of queue of connections



1. Accept connection from client-side(server)

Sockfd\_server: socket which was created just now

S\_addr\_client:client-in structure

Client\_length:length



1. Client-side request connection(client)

Sockfd: socket which was created just now

S\_addr\_in:socket client structure

Sizeof(s\_addr\_in):length of socket client structure



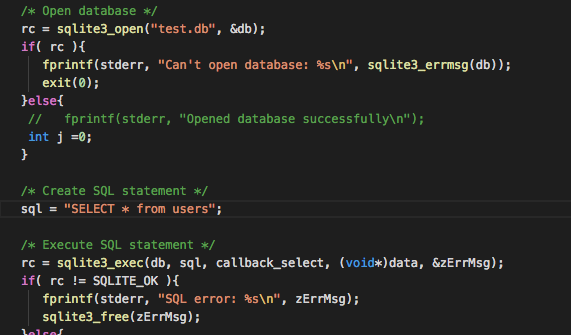
1. Create multi-thread(server)

thread\_id:id address of thread

Data\_handle:sub-threading function



1. Sqlite related coding(client)

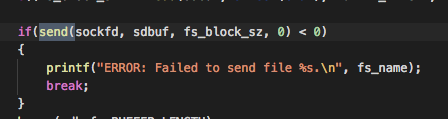


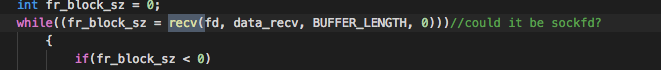
1. Read/write (read and write buffer from socket)



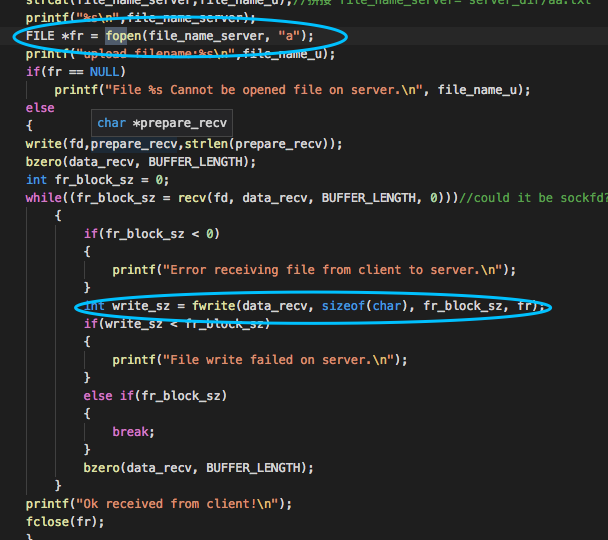


1. Recv/send(send buffer to socket and receive buffer from socket)

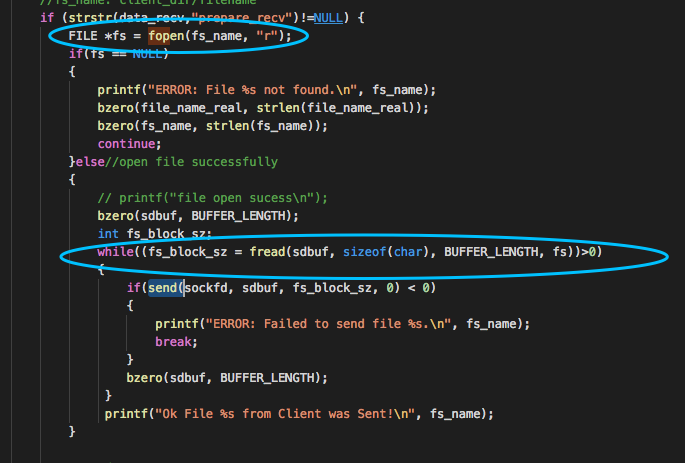




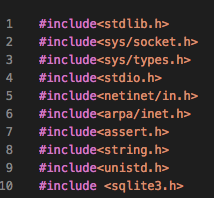
1. Write file



1. Read file



1. .h file reference



1. Conclusion

Through this project, I have mastered the knowledge of soket transmission, TCP protocol, muti-thread use, file reading and writing, database operation, etc., and have gained a deeper understanding of network transmission. Thank you for your hard teaching and patient guidance. I will never forget the short exchange study in France in my future life.