# Socket Programming Part3

## 一、如何compile

1. 確認資料夾有以下四個檔案與一個資料夾（內含四個檔案）
    a. server.cpp
    b. client.cpp
    c. threadpool.h
    d. makefile
    e. ssl
        i. serverCert.pem
        ii. serverKey.pem
        iii. clientCert.pem
        iv. clientKey.pem
2. 打開makefile，將LDFLAGS、CPPFLAGS之路徑換成本機端openssl位置，例如：

```
LDFLAGS = -L/usr/local/opt/openssl/lib
CPPFLAGS = -I/usr/local/opt/openssl/include
```

   ps. 根據安裝方式不同，openssl位置可由以下指令獲得

```
apt show openssl
```

   或是

```
brew info openssl
```

3. 進行編譯，請輸入

```
make
```

## 二、如何執行程式

1.
執行server端

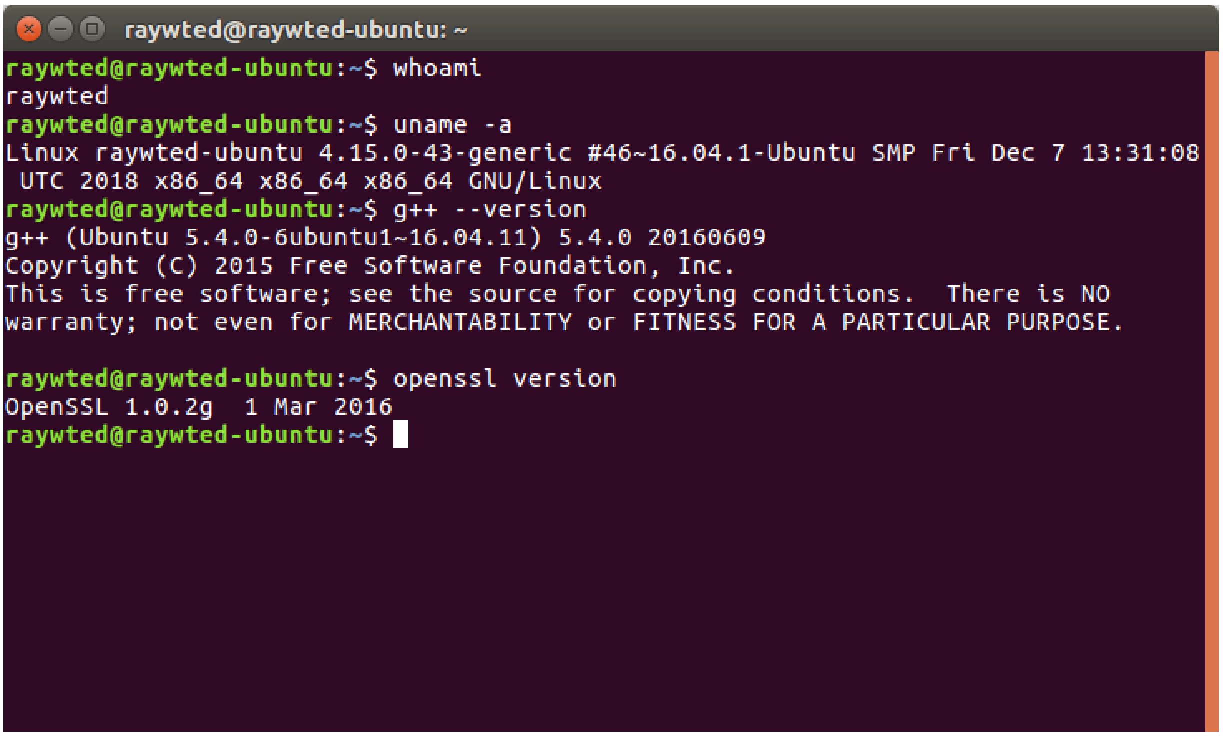
```
./server
```

2.
執行client端
使用自己的server，所以ip是127.0.0.1， port是8700

```
./client 127.0.0.1 8700
```

## 三、程式需求、執行需求

以下是我的ubuntu版本、g++版本、openssl版本



```
raywted@raywted-ubuntu:~$ whoami
raywted
raywted@raywted-ubuntu:~$ uname -a
Linux raywted-ubuntu 4.15.0-43-generic #46~16.04.1-Ubuntu SMP Fri Dec 7 13:31:08
 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
raywted@raywted-ubuntu:~$ g++ --version
g++ (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

raywted@raywted-ubuntu:~$ openssl version
OpenSSL 1.0.2g  1 Mar 2016
raywted@raywted-ubuntu:~$ 
```

## 四、程式邏輯說明

**Server端**

1. 初始化SSL [圖1]
2. 初始化Thread Pool
3. 初始化Socket
4. 進入While迴圈等待Client連線
    a. SSL加密連線設定 [圖2]
    b. 使用Worker Thread進行連線
    c. 根據Client請求（註冊、登入、上線清單、付款請求、付款確認、離開）進行對應回覆

**Client端**

1. 初始化SSL
2. 註冊\登入
    a. 初始化Socket
    b. SSL加密連線設定
    c. 進行連線
    d. 傳送註冊\登入訊息給Server
3. 產生一個Thread作為**Server of Client** [圖3]
4. 進入While迴圈等待Client指令
    a. 上線清單
    b. 付款請求

        i. 傳送付款對象給Server
        ii. 接收付款對象IP與Port
        iii. 產生一個Thread作為**Client of Client** [圖4]
    c. 離開

**Server of Client**

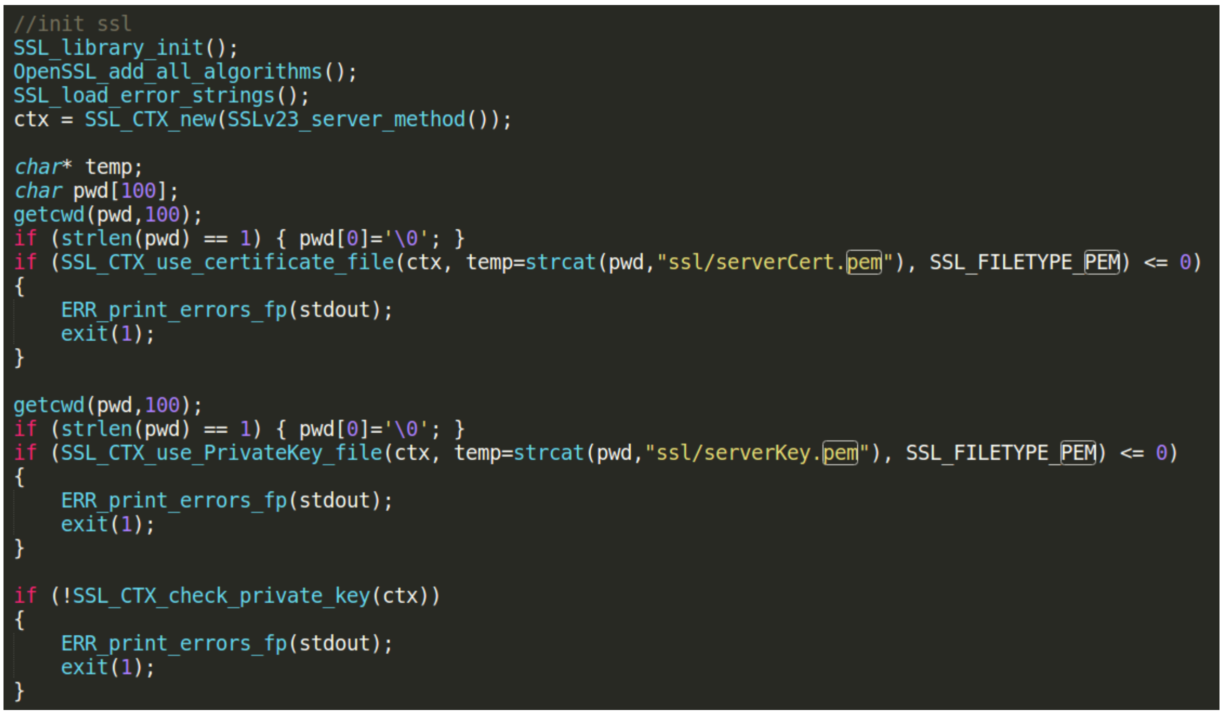
1. 初始化SSL
2. 初始化Socket
3. 進入While迴圈等待其他Client連線
    a. 收到付款請求
    b. 進行回覆

**Client of Client**

1. 初始化SSL
2. 初始化Socket
3. SSL加密連線至付款對象
4. 付款對象確認完付款資訊後，收到付款對象的回覆
    a. 同意
        i. 產生一個Thread與Server連線 [圖5]
        ii. 請求Server進行交易資訊處理
        iii. 接收交易資訊處理結果
    b. 不同意

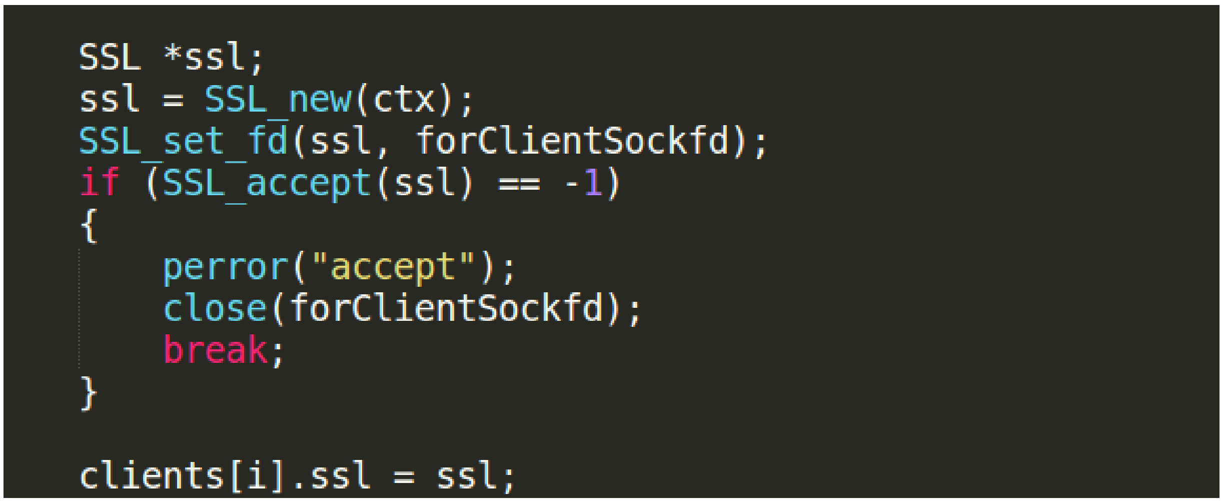
**PKI 傳輸流程加解密機制**

1. 初始化憑證與金鑰
2. 建立SSL連線，連線雙方得到對方的公鑰並與自己的私鑰形成一組鑰匙
3. 使用鑰匙對資料進行加密後再傳送
4. 收到資料後使用鑰匙對資料進行解密



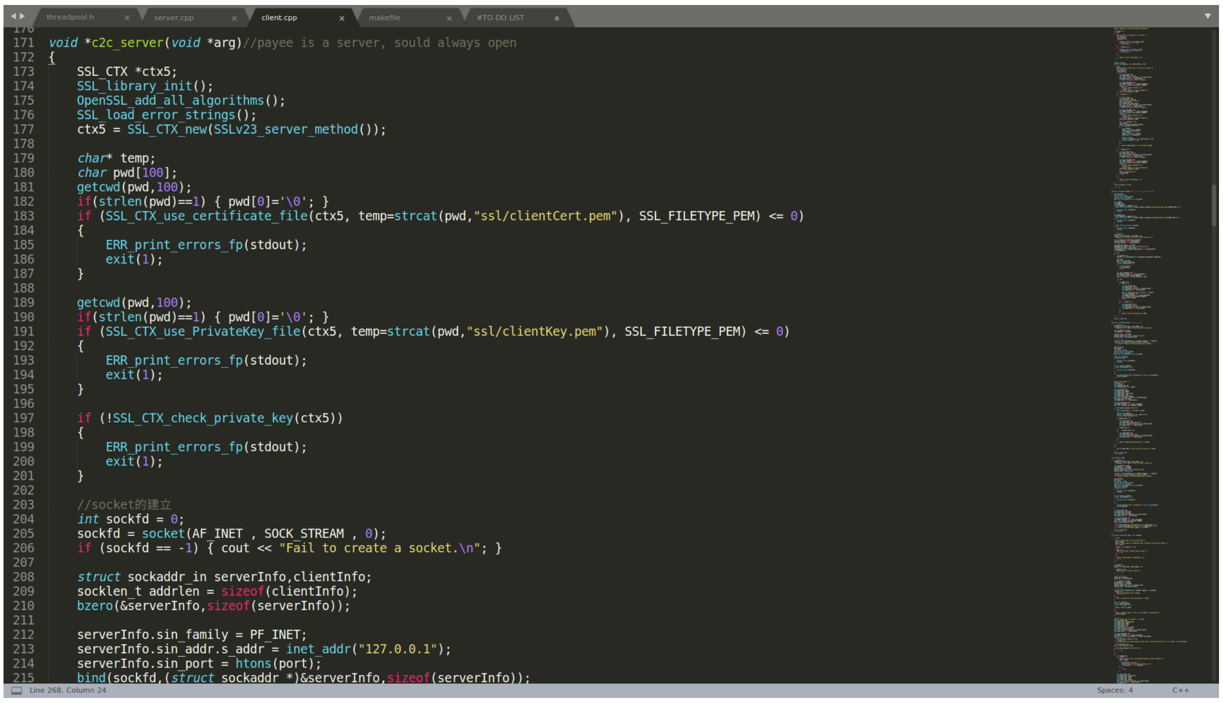
```c
//init ssl
SSL_library_init();
OpenSSL_add_all_algorithms();
SSL_load_error_strings();
ctx = SSL_CTX_new(SSLv23_server_method());

char* temp;
char pwd[100];
getcwd(pwd,100);
if (strlen(pwd) == 1) { pwd[0]='\0'; }
if (SSL_CTX_use_certificate_file(ctx, temp=strcat(pwd,"ssl/serverCert.pem"), SSL_FILETYPE_PEM) <= 0)
{
    ERR_print_errors_fp(stdout);
    exit(1);
}

getcwd(pwd,100);
if (strlen(pwd) == 1) { pwd[0]='\0'; }
if (SSL_CTX_use_PrivateKey_file(ctx, temp=strcat(pwd,"ssl/serverKey.pem"), SSL_FILETYPE_PEM) <= 0)
{
    ERR_print_errors_fp(stdout);
    exit(1);
}

if (!SSL_CTX_check_private_key(ctx))
{
    ERR_print_errors_fp(stdout);
    exit(1);
}
```

图1



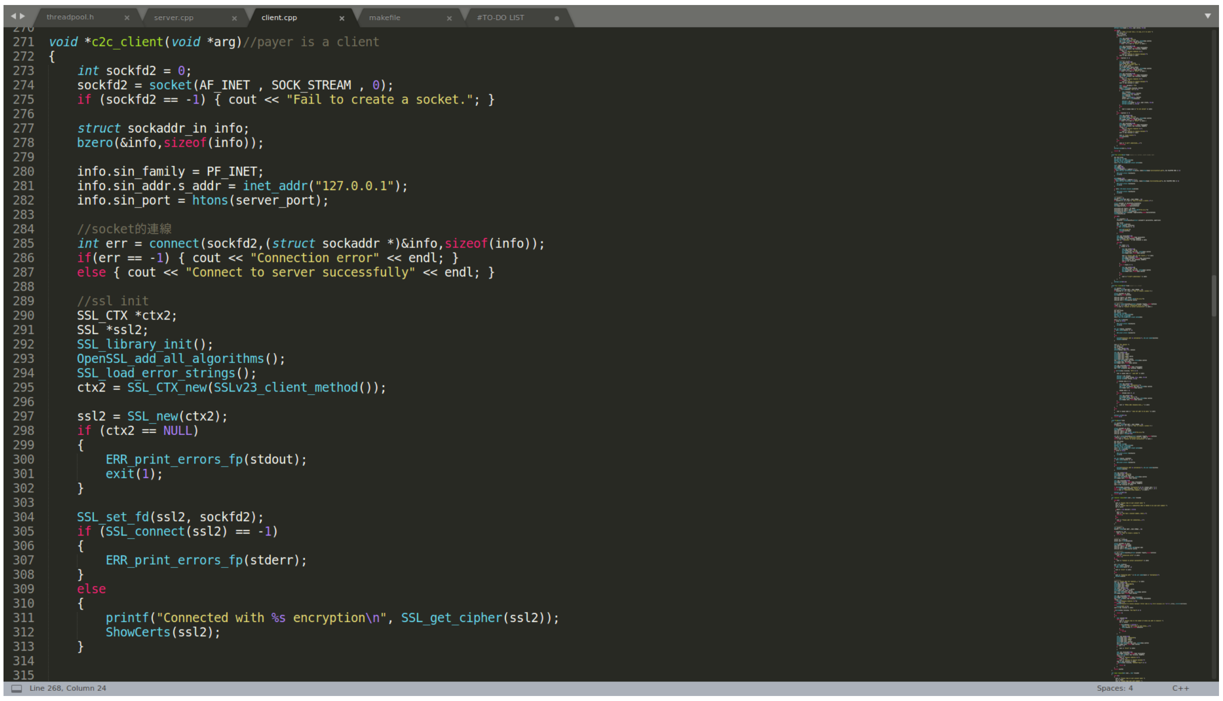
```c
SSL *ssl;
ssl = SSL_new(ctx);
SSL_set_fd(ssl, forClientSockfd);
if (SSL_accept(ssl) == -1)
{
    perror("accept");
    close(forClientSockfd);
    break;
}

clients[i].ssl = ssl;
```

图2



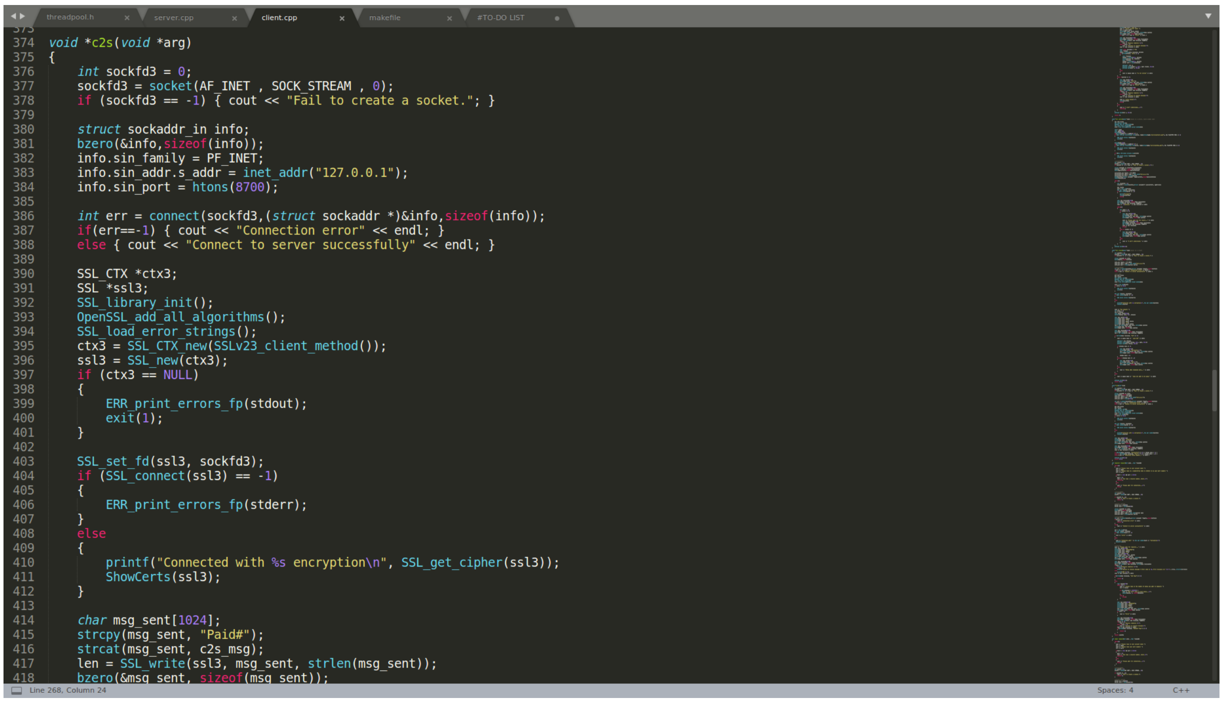
```cpp
void *c2c_server(void *arg)//payee is a server, sould always open
{
    SSL_CTX *ctx5;
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ctx5 = SSL_CTX_new(SSLv23_server_method());

    char* temp;
    char pwd[100];
    getcwd(pwd,100);
    if(strlen(pwd)==1) { pwd[0]='\0'; }
    if (SSL_CTX_use_certificate_file(ctx5, temp=strcat(pwd,"ssl/clientCert.pem"), SSL_FILETYPE_PEM) <= 0)
    {
        ERR_print_errors_fp(stdout);
        exit(1);
    }

    getcwd(pwd,100);
    if(strlen(pwd)==1) { pwd[0]='\0'; }
    if (SSL_CTX_use_PrivateKey_file(ctx5, temp=strcat(pwd,"ssl/clientKey.pem"), SSL_FILETYPE_PEM) <= 0)
    {
        ERR_print_errors_fp(stdout);
        exit(1);
    }

    if (!SSL_CTX_check_private_key(ctx5))
    {
        ERR_print_errors_fp(stdout);
        exit(1);
    }

    //socket的建立
    int sockfd = 0;
    sockfd = socket(AF_INET , SOCK_STREAM , 0);
    if (sockfd == -1) { cout << "Fail to create a socket.\n"; }

    struct sockaddr_in serverInfo,clientInfo;
    socklen_t addrlen = sizeof(clientInfo);
    bzero(&serverInfo,sizeof(serverInfo));

    serverInfo.sin_family = PF_INET;
    serverInfo.sin_addr.s_addr = inet_addr("127.0.0.1");
    serverInfo.sin_port = htons(port);
    bind(sockfd,(struct sockaddr *)&serverInfo,sizeof(serverInfo));
```

圖3



```cpp
void *c2c_client(void *arg)//payer is a client
{
    int sockfd2 = 0;
    sockfd2 = socket(AF_INET , SOCK_STREAM , 0);
    if (sockfd2 == -1) { cout << "Fail to create a socket."; }

    struct sockaddr_in info;
    bzero(&info,sizeof(info));

    info.sin_family = PF_INET;
    info.sin_addr.s_addr = inet_addr("127.0.0.1");
    info.sin_port = htons(server_port);

    //socket的連線
    int err = connect(sockfd2,(struct sockaddr *)&info,sizeof(info));
    if(err == -1) { cout << "Connection error" << endl; }
    else { cout << "Connect to server successfully" << endl; }

    //ssl init
    SSL_CTX *ctx2;
    SSL *ssl2;
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ctx2 = SSL_CTX_new(SSLv23_client_method());

    ssl2 = SSL_new(ctx2);
    if (ctx2 == NULL)
    {
        ERR_print_errors_fp(stdout);
        exit(1);
    }

    SSL_set_fd(ssl2, sockfd2);
    if (SSL_connect(ssl2) == -1)
    {
        ERR_print_errors_fp(stderr);
    }
    else
    {
        printf("Connected with %s encryption\n", SSL_get_cipher(ssl2));
        ShowCerts(ssl2);
    }

```

圖4

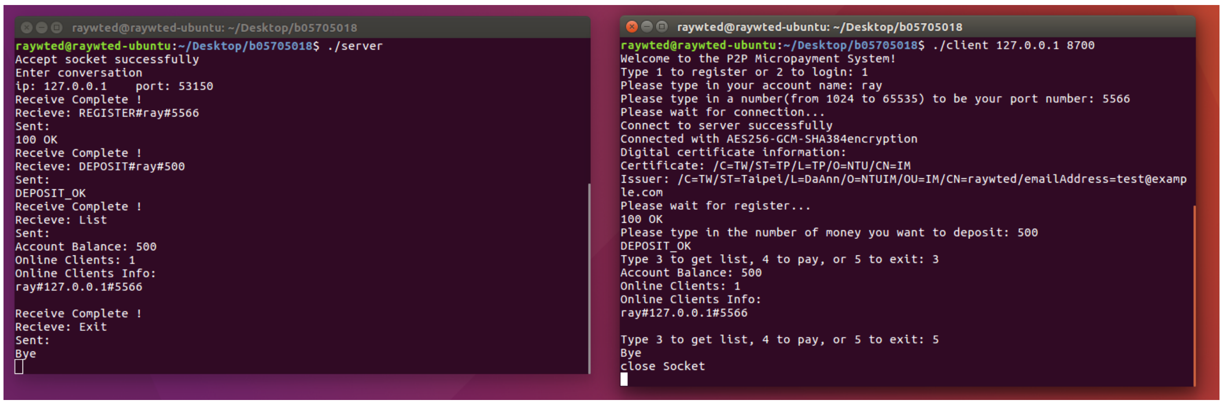


```cpp
void *c2s(void *arg)
{
    int sockfd3 = 0;
    sockfd3 = socket(AF_INET , SOCK_STREAM , 0);
    if (sockfd3 == -1) { cout << "Fail to create a socket."; }

    struct sockaddr_in info;
    bzero(&info,sizeof(info));
    info.sin_family = PF_INET;
    info.sin_addr.s_addr = inet_addr("127.0.0.1");
    info.sin_port = htons(8700);

    int err = connect(sockfd3,(struct sockaddr *)&info,sizeof(info));
    if(err==-1) { cout << "Connection error" << endl; }
    else { cout << "Connect to server successfully" << endl; }

    SSL_CTX *ctx3;
    SSL *ssl3;
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ctx3 = SSL_CTX_new(SSLv23_client_method());
    ssl3 = SSL_new(ctx3);
    if (ctx3 == NULL)
    {
        ERR_print_errors_fp(stdout);
        exit(1);
    }

    SSL_set_fd(ssl3, sockfd3);
    if (SSL_connect(ssl3) == -1)
    {
        ERR_print_errors_fp(stderr);
    }
    else
    {
        printf("Connected with %s encryption\n", SSL_get_cipher(ssl3));
        ShowCerts(ssl3);
    }

    char msg_sent[1024];
    strcpy(msg_sent, "Paid#");
    strcat(msg_sent, c2s_msg);
    len = SSL_write(ssl3, msg_sent, strlen(msg_sent));
    bzero(&msg_sent, sizeof(msg_sent));
```

圖5

## 五、所實作的各功能截圖

1. 註冊功能
    a. 可以輸入account name、port number與存款金額
    b. 取得清單、付款（第4點）、離開

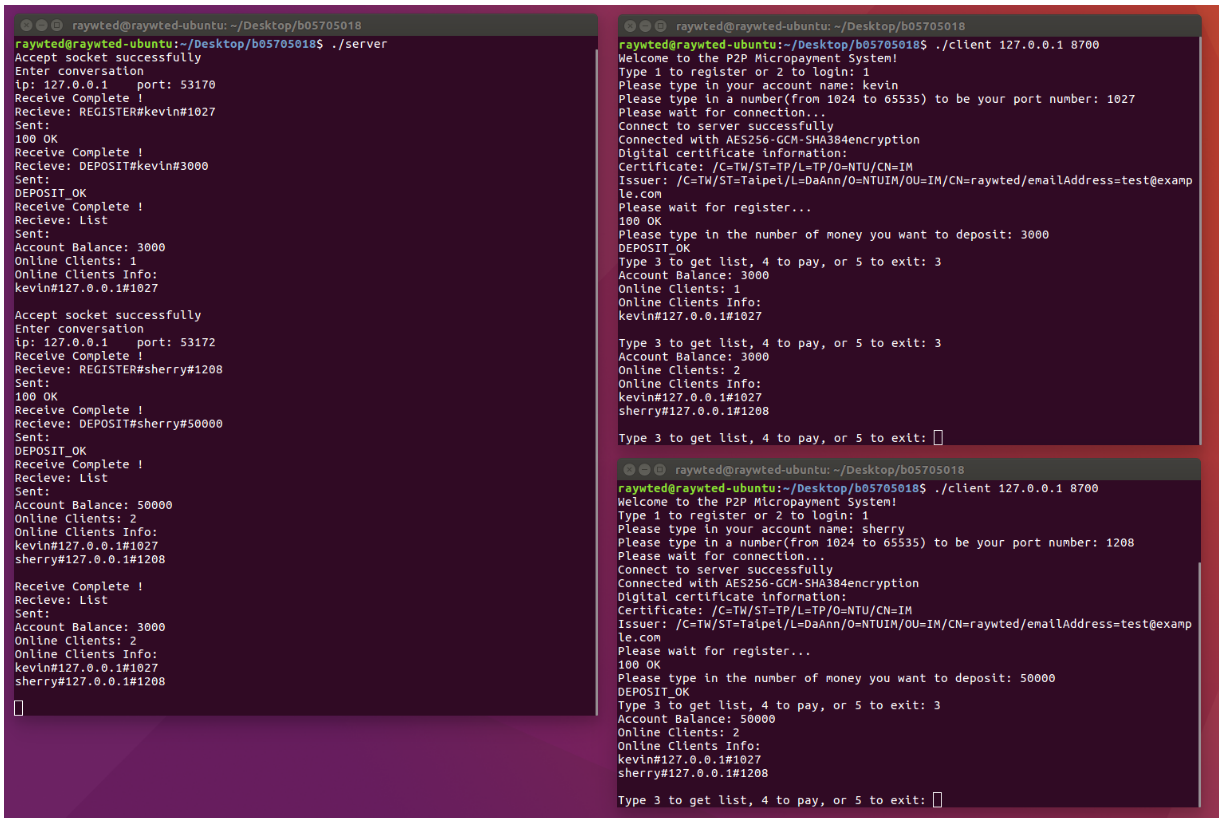


raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu: ~/Desktop/b05705018

```
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./server
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 53150
Receive Complete !
Recieve: REGISTER#ray#5566
Sent:
100 OK
Receive Complete !
Recieve: DEPOSIT#ray#500
Sent:
DEPOSIT_OK
Receive Complete !
Recieve: List
Sent:
Account Balance: 500
Online Clients: 1
Online Clients Info:
ray#127.0.0.1#5566

Receive Complete !
Recieve: Exit
Sent:
Bye
```

```
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: ray
Please type in a number(from 1024 to 65535) to be your port number: 5566
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 500
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 500
Online Clients: 1
Online Clients Info:
ray#127.0.0.1#5566

Type 3 to get list, 4 to pay, or 5 to exit: 5
Bye
close Socket
```

2. 多人連線
    - 第一個client（右上）先註冊，取得清單發現只有自己一個人
    - 再讓第二個client（右下）註冊，此時兩者都能正確取得上線清單為兩人
    - 最後，第一個client離開，第二個client再次取得清單發現只剩自己一人。

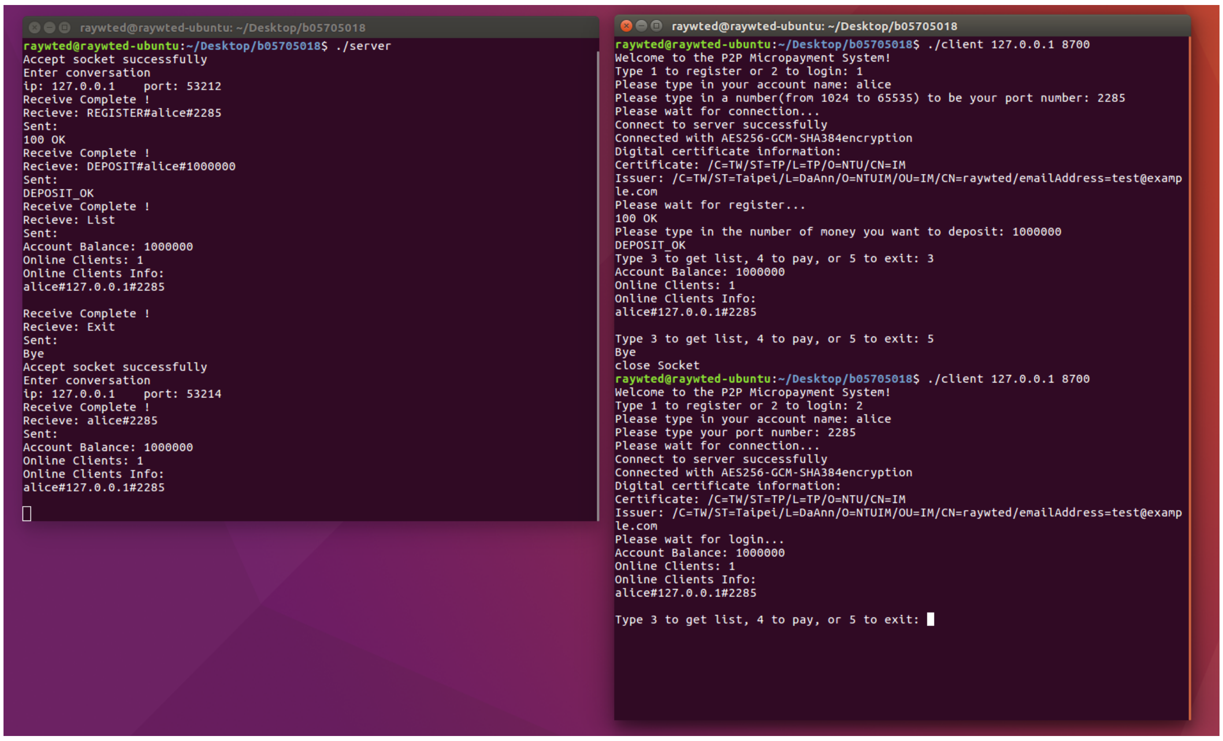


```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./server
Accept socket successfully
Enter conversation
ip: 127.0.0.1     port: 53170
Receive Complete !
Recieve: REGISTER#kevin#1027
Sent:
100 OK
Receive Complete !
Recieve: DEPOSIT#kevin#3000
Sent:
DEPOSIT_OK
Receive Complete !
Recieve: List
Sent:
Account Balance: 3000
Online Clients: 1
Online Clients Info:
kevin#127.0.0.1#1027

Accept socket successfully
Enter conversation
ip: 127.0.0.1     port: 53172
Receive Complete !
Recieve: REGISTER#sherry#1208
Sent:
100 OK
Receive Complete !
Recieve: DEPOSIT#sherry#50000
Sent:
DEPOSIT_OK
Receive Complete !
Recieve: List
Sent:
Account Balance: 50000
Online Clients: 2
Online Clients Info:
kevin#127.0.0.1#1027
sherry#127.0.0.1#1208

Receive Complete !
Recieve: List
Sent:
Account Balance: 3000
Online Clients: 2
Online Clients Info:
kevin#127.0.0.1#1027
sherry#127.0.0.1#1208
```

```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: kevin
Please type in a number(from 1024 to 65535) to be your port number: 1027
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 3000
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 3000
Online Clients: 1
Online Clients Info:
kevin#127.0.0.1#1027

Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 3000
Online Clients: 2
Online Clients Info:
kevin#127.0.0.1#1027
sherry#127.0.0.1#1208

Type 3 to get list, 4 to pay, or 5 to exit:
```

```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: sherry
Please type in a number(from 1024 to 65535) to be your port number: 1208
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 50000
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 50000
Online Clients: 2
Online Clients Info:
kevin#127.0.0.1#1027
sherry#127.0.0.1#1208

Type 3 to get list, 4 to pay, or 5 to exit:
```

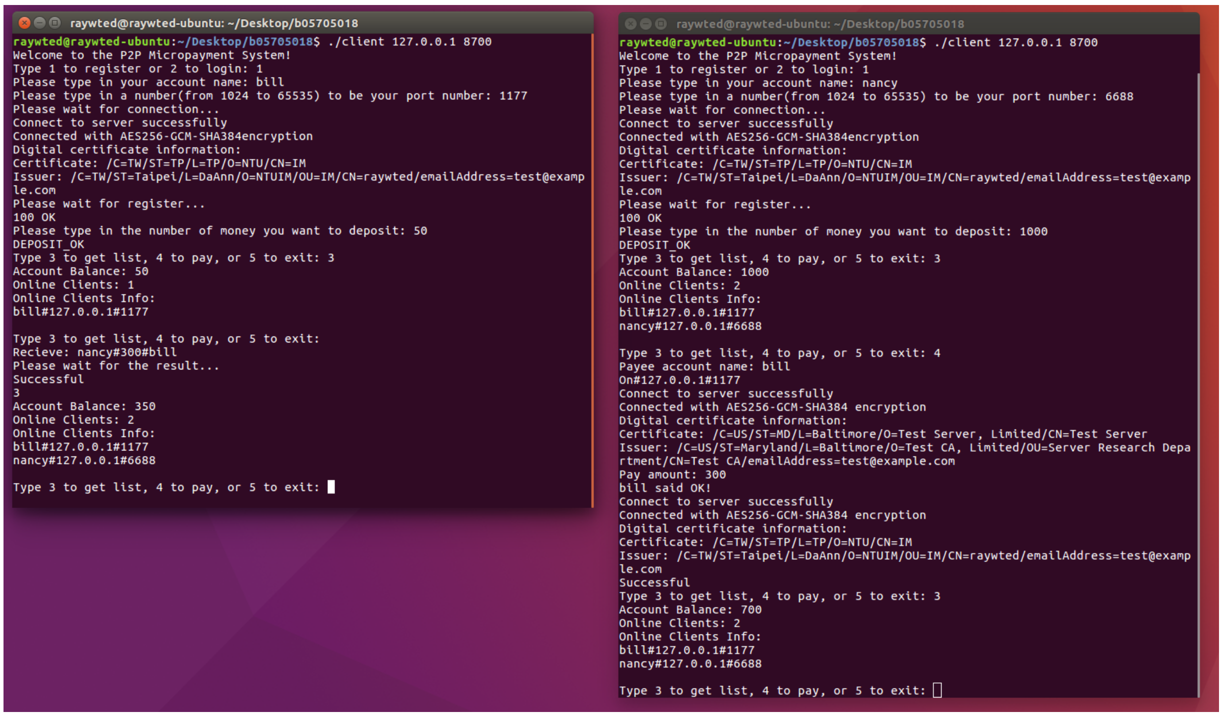
3. 登入功能
   - 首先讓client正常註冊，成功後離開
   - 再次連線，可以使用已註冊過的account name與port number直接登入



```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./server
Accept socket successfully
Enter conversation
ip: 127.0.0.1     port: 53212
Receive Complete !
Recieve: REGISTER#alice#2285
Sent:
100 OK
Receive Complete !
Recieve: DEPOSIT#alice#1000000
Sent:
DEPOSIT_OK
Receive Complete !
Recieve: List
Sent:
Account Balance: 1000000
Online Clients: 1
Online Clients Info:
alice#127.0.0.1#2285

Receive Complete !
Recieve: Exit
Sent:
Bye
Accept socket successfully
Enter conversation
ip: 127.0.0.1     port: 53214
Receive Complete !
Recieve: alice#2285
Sent:
Account Balance: 1000000
Online Clients: 1
Online Clients Info:
alice#127.0.0.1#2285
```

```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: alice
Please type in a number(from 1024 to 65535) to be your port number: 2285
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 1000000
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 1000000
Online Clients: 1
Online Clients Info:
alice#127.0.0.1#2285

Type 3 to get list, 4 to pay, or 5 to exit: 5
Bye
close Socket
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 2
Please type in your account name: alice
Please type your port number: 2285
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for login...
Account Balance: 1000000
Online Clients: 1
Online Clients Info:
alice#127.0.0.1#2285

Type 3 to get list, 4 to pay, or 5 to exit:
```

4. P2P付款功能
   - client1（左）為收款者
   - client2（右）為付款者



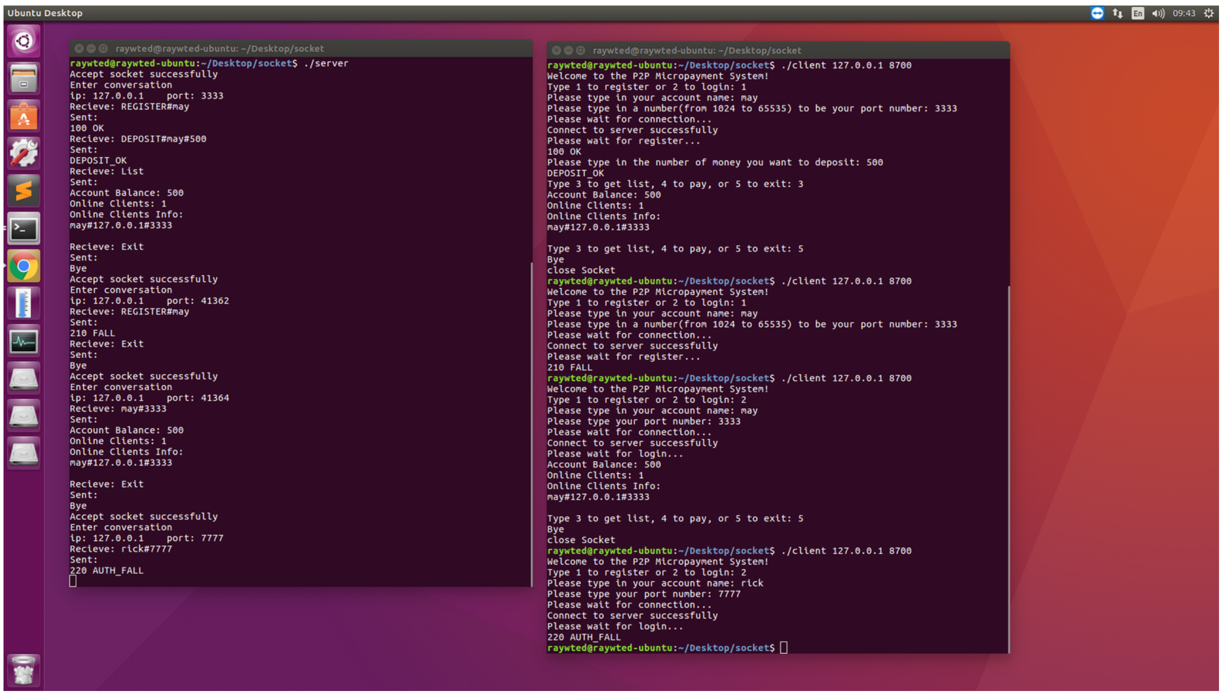
```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: bill
Please type in a number(from 1024 to 65535) to be your port number: 1177
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 50
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 50
Online Clients: 1
Online Clients Info:
bill#127.0.0.1#1177

Type 3 to get list, 4 to pay, or 5 to exit:
Recieve: nancy#300#bill
Please wait for the result...
Successful
3
Account Balance: 350
Online Clients: 2
Online Clients Info:
bill#127.0.0.1#1177
nancy#127.0.0.1#6688

Type 3 to get list, 4 to pay, or 5 to exit: 
```

```
raywted@raywted-ubuntu: ~/Desktop/b05705018
raywted@raywted-ubuntu:~/Desktop/b05705018$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: nancy
Please type in a number(from 1024 to 65535) to be your port number: 6688
Please wait for connection...
Connect to server successfully
Connected with AES256-GCM-SHA384encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 1000
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 1000
Online Clients: 2
Online Clients Info:
bill#127.0.0.1#1177
nancy#127.0.0.1#6688

Type 3 to get list, 4 to pay, or 5 to exit: 4
Payee account name: bill
On#127.0.0.1#1177
Connect to server successfully
Connected with AES256-GCM-SHA384 encryption
Digital certificate information:
Certificate: /C=US/ST=MD/L=Baltimore/O=Test Server, Limited/CN=Test Server
Issuer: /C=US/ST=Maryland/L=Baltimore/O=Test CA, Limited/OU=Server Research Depa
rtment/CN=Test CA/emailAddress=test@example.com
Pay amount: 300
bill said OK!
Connect to server successfully
Connected with AES256-GCM-SHA384 encryption
Digital certificate information:
Certificate: /C=TW/ST=TP/L=TP/O=NTU/CN=IM
Issuer: /C=TW/ST=Taipei/L=DaAnn/O=NTUIM/OU=IM/CN=raywted/emailAddress=test@examp
le.com
Successful
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 700
Online Clients: 2
Online Clients Info:
bill#127.0.0.1#1177
nancy#127.0.0.1#6688

Type 3 to get list, 4 to pay, or 5 to exit: 
```

## 六、Bonus截圖與展示

### 例外處理

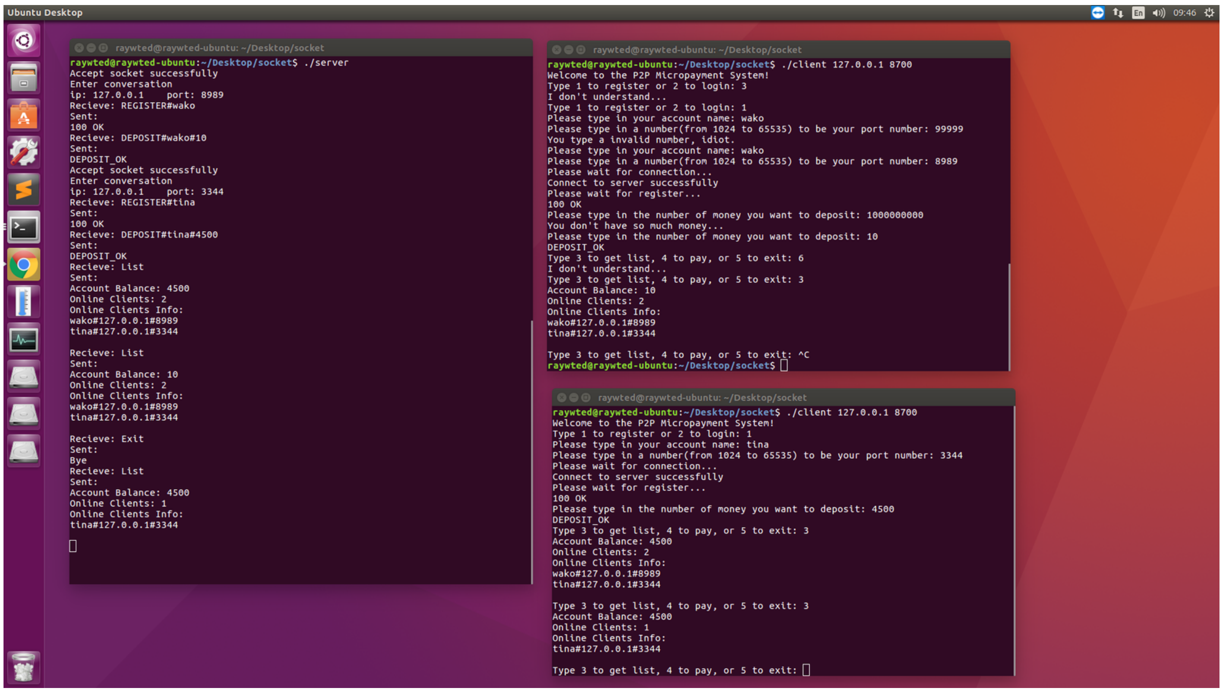
1. 註冊、登入例外處理
   - 首先正常註冊一個帳號
   - 使用同樣account name與port number再次註冊，會失敗（收到”210 FALL”）
   - 使用同樣account name與port number可以成功登入
   - 使用未註冊過的account name與port number登入，會失敗（收到”220 AUTH_FALL”）



```
raywted@raywted-ubuntu: ~/Desktop/socket
raywted@raywted-ubuntu:~/Desktop/socket$ ./server
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 3333
Recieve: REGISTER#may
Sent:
100 OK
Recieve: DEPOSIT#may#500
Sent:
DEPOSIT_OK
Recieve: List
Sent:
Account Balance: 500
Online Clients: 1
Online Clients Info:
may#127.0.0.1#3333

Recieve: Exit
Sent:
Bye
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 41362
Recieve: REGISTER#may
Sent:
210 FALL
Recieve: Exit
Sent:
Bye
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 41364
Recieve: may#3333
Sent:
Account Balance: 500
Online Clients: 1
Online Clients Info:
may#127.0.0.1#3333

Recieve: Exit
Sent:
Bye
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 7777
Recieve: rick#7777
Sent:
220 AUTH_FALL
```

```
raywted@raywted-ubuntu: ~/Desktop/socket
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: may
Please type in a number(from 1024 to 65535) to be your port number: 3333
Please wait for connection...
Connect to server successfully
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 500
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 500
Online Clients: 1
Online Clients Info:
may#127.0.0.1#3333

Type 3 to get list, 4 to pay, or 5 to exit: 5
Bye
close Socket
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: may
Please type in a number(from 1024 to 65535) to be your port number: 3333
Please wait for connection...
Connect to server successfully
Please wait for register...
210 FALL
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 2
Please type in your account name: may
Please type your port number: 3333
Please wait for connection...
Connect to server successfully
Please wait for login...
Account Balance: 500
Online Clients: 1
Online Clients Info:
may#127.0.0.1#3333

Type 3 to get list, 4 to pay, or 5 to exit: 5
Bye
close Socket
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 2
Please type in your account name: rick
Please type your port number: 7777
Please wait for connection...
Connect to server successfully
Please wait for login...
220 AUTH_FALL
raywted@raywted-ubuntu:~/Desktop/socket$ 
```

2. 輸入例外處理
    - 在註冊登入階段輸入不正確指令
    - 輸入不正確port number
    - 輸入過大存款金額
    - 在指令操作階段輸入不正確指令
    - 意外斷線（ctrl+c），server與其他client不受影響，server會當作client傳了Exit過來，正常讓該client離線



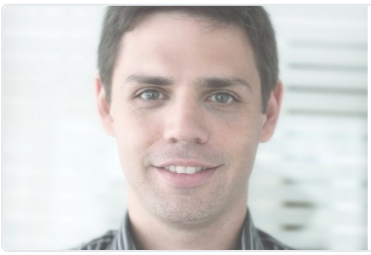
```
raywted@raywted-ubuntu:~/Desktop/socket$ ./server
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 8989
Recieve: REGISTER#wako
Sent:
100 OK
Recieve: DEPOSIT#wako#10
Sent:
DEPOSIT_OK
Accept socket successfully
Enter conversation
ip: 127.0.0.1    port: 3344
Recieve: REGISTER#tina
Sent:
100 OK
Recieve: DEPOSIT#tina#4500
Sent:
DEPOSIT_OK
Recieve: List
Sent:
Account Balance: 4500
Online Clients: 2
Online Clients Info:
wako#127.0.0.1#8989
tina#127.0.0.1#3344

Recieve: List
Sent:
Account Balance: 10
Online Clients: 2
Online Clients Info:
wako#127.0.0.1#8989
tina#127.0.0.1#3344

Recieve: Exit
Sent:
Bye
Recieve: List
Sent:
Account Balance: 4500
Online Clients: 1
Online Clients Info:
tina#127.0.0.1#3344
```

```
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 3
I don't understand...
Type 1 to register or 2 to login: 1
Please type in your account name: wako
Please type in a number(from 1024 to 65535) to be your port number: 99999
You type a invalid number, idiot.
Please type in your account name: wako
Please type in a number(from 1024 to 65535) to be your port number: 8989
Please wait for connection...
Connect to server successfully
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 1000000000
You don't have so much money...
Please type in the number of money you want to deposit: 10
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 6
I don't understand...
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 10
Online Clients: 2
Online Clients Info:
wako#127.0.0.1#8989
tina#127.0.0.1#3344

Type 3 to get list, 4 to pay, or 5 to exit: ^C
raywted@raywted-ubuntu:~/Desktop/socket$
```

```
raywted@raywted-ubuntu:~/Desktop/socket$ ./client 127.0.0.1 8700
Welcome to the P2P Micropayment System!
Type 1 to register or 2 to login: 1
Please type in your account name: tina
Please type in a number(from 1024 to 65535) to be your port number: 3344
Please wait for connection...
Connect to server successfully
Please wait for register...
100 OK
Please type in the number of money you want to deposit: 4500
DEPOSIT_OK
Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 4500
Online Clients: 2
Online Clients Info:
wako#127.0.0.1#8989
tina#127.0.0.1#3344

Type 3 to get list, 4 to pay, or 5 to exit: 3
Account Balance: 4500
Online Clients: 1
Online Clients Info:
tina#127.0.0.1#3344

Type 3 to get list, 4 to pay, or 5 to exit:
```

## 七、參考資料

Thread Pool



mbrossard/threadpool

A simple C Thread pool implementation. Contribute to mbrossard/threadpool development by creating an account on...

github.com

mbrossard/threadpool • github.com

Openssl

1.

How do you sign a Certificate Signing Request wi...

During my search, I found several ways of signing a SSL Certificate Signing Request: Using the x509 module: openssl x509 -req -days...

stackoverflow.com

How do you sign a Certificate Signing Request with your Certification Authority? • stackoverflow.com

2.
https://blog.csdn.net/sjin_1314/article/details/21043613?fbclid=IwAR2kxMAXsH1YPWZNNEauIXR9I-GaeBKE-Q6YdB8TI-4zbN0LHzfUdLWW7C0