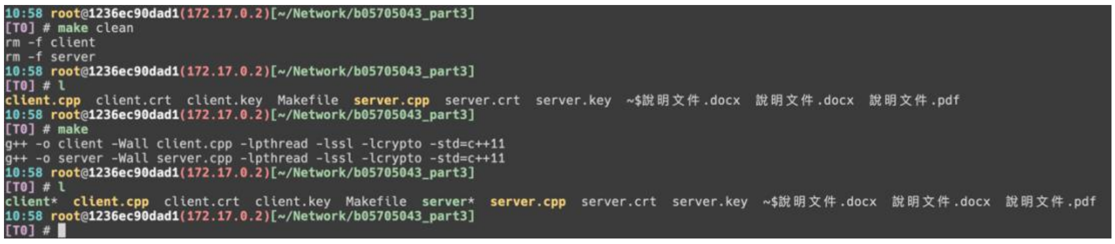
Socket Programming part 3 說明文件

資管三 唐瑋廷 b05705043

一、如何 compile

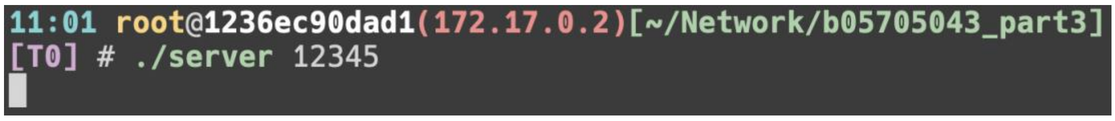
資料夾內包含 client.cpp, server.cpp 和 Makefile，直接下指令 make 即可編譯出執行檔 client, server。另外 make clean 會刪除 client, server。



```
10:58 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # make clean
rm -f client
rm -f server
10:58 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # l
client.cpp  client.crt  client.key  Makefile  server.cpp  server.crt  server.key  ~$說明文件.docx  說明文件.docx  說明文件.pdf
10:58 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # make
g++ -o client -Wall client.cpp -lpthread -lssl -lcrypto -std=c++11
g++ -o server -Wall server.cpp -lpthread -lssl -lcrypto -std=c++11
10:58 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # l
client*  client.cpp  client.crt  client.key  Makefile  server*  server.cpp  server.crt  server.key  ~$說明文件.docx  說明文件.docx  說明文件.pdf
10:58 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # 
```

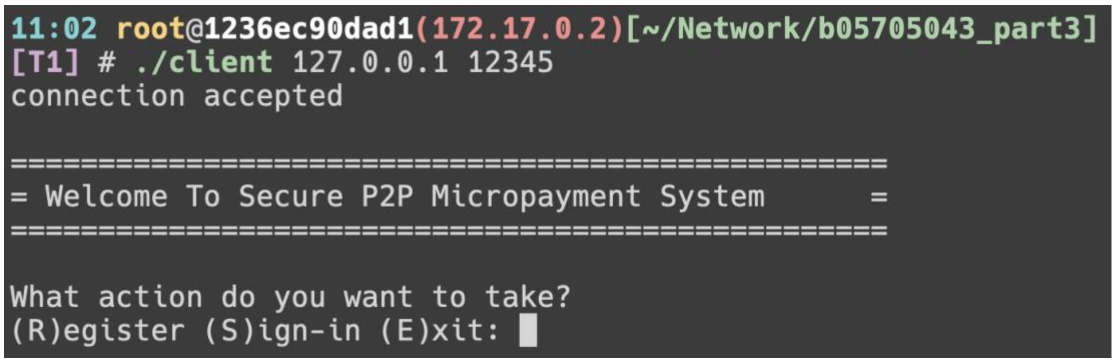
二、如何執行程式

./<serverName> <port>



```
11:01 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
```

./<clientName> <ip> <port>



```
11:02 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T1] # ./client 127.0.0.1 12345
connection accepted

====================================================
= Welcome To Secure P2P Micropayment System        =
====================================================

What action do you want to take?
(R)egister (S)ign-in (E)xit: 
```

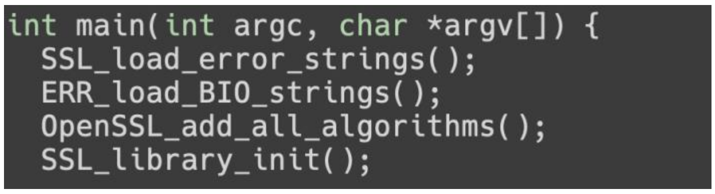
三、程式需求、執行需求

使用 Linux Socket Programming，可在 mac, linux 下編譯。附檔所附上的執行檔是在 Ubuntu 16.04 下編譯（gcc version 5.4.0），可在 Linux 上執行。需要支援 pthread 以及 ssl。

四、程式邏輯說明

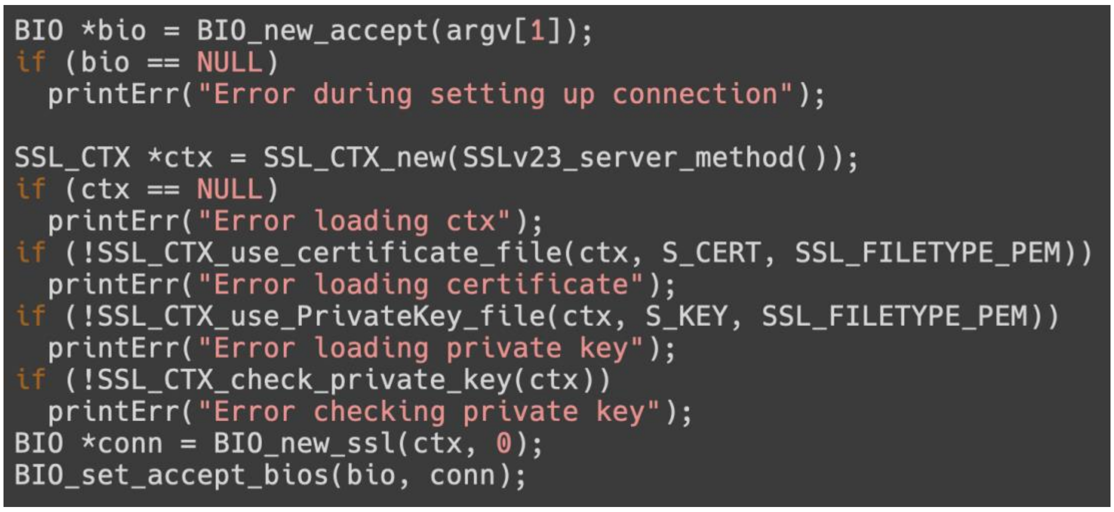
Server 端：(worker pool 和基本指令的部分和第二階段相同，故省略)

在 mian 裡面，首先會先將 ssl library 進行初始化，以便接下來使用。



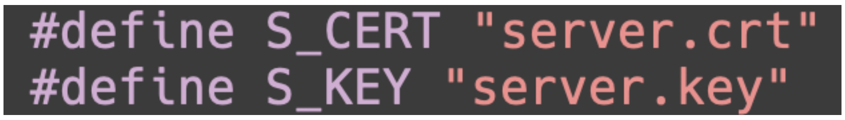
```
int main(int argc, char *argv[]) {
  SSL_load_error_strings();
  ERR_load_BIO_strings();
  OpenSSL_add_all_algorithms();
  SSL_library_init();
```

接著建立 CTX 會話通道和 BIO 通道，使用的是 SSLv23 server method，讀取 public key 和 private key，同時做一些 error control，並設置好監聽環境。



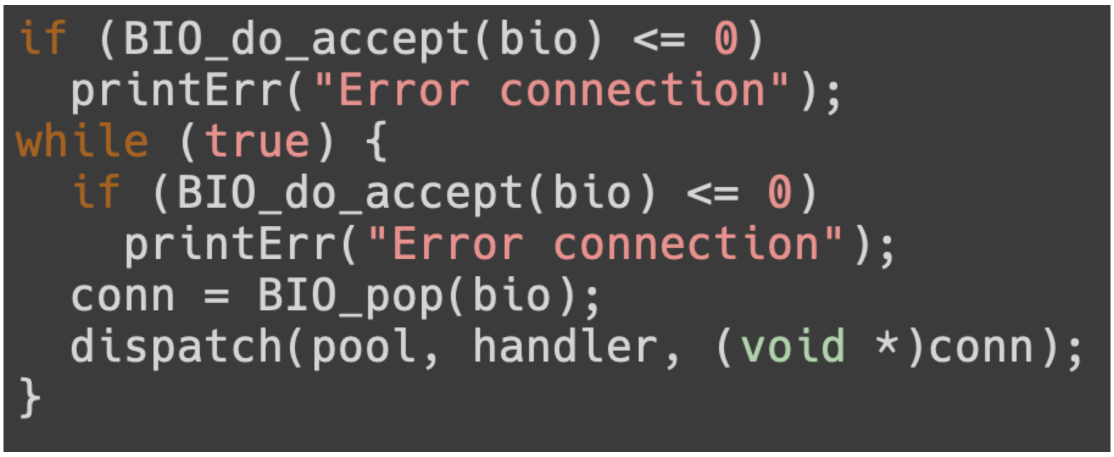
```
BIO *bio = BIO_new_accept(argv[1]);
if (bio == NULL)
  printErr("Error during setting up connection");

SSL_CTX *ctx = SSL_CTX_new(SSLv23_server_method());
if (ctx == NULL)
  printErr("Error loading ctx");
if (!SSL_CTX_use_certificate_file(ctx, S_CERT, SSL_FILETYPE_PEM))
  printErr("Error loading certificate");
if (!SSL_CTX_use_PrivateKey_file(ctx, S_KEY, SSL_FILETYPE_PEM))
  printErr("Error loading private key");
if (!SSL_CTX_check_private_key(ctx))
  printErr("Error checking private key");
BIO *conn = BIO_new_ssl(ctx, 0);
BIO_set_accept_bios(bio, conn);
```

其中 S_CERT(public key), S_KEY 分別是從資料夾中讀取的金鑰。



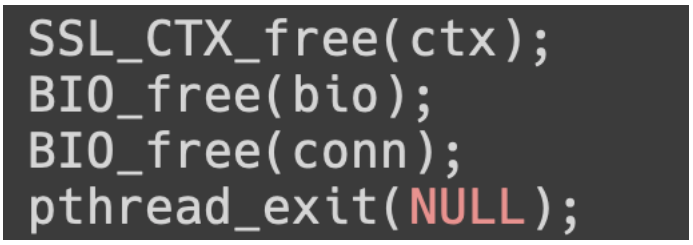
```
#define S_CERT "server.crt"
#define S_KEY "server.key"
```

接著開始接收 client 端的請求（BIO_do_accept 需要呼叫兩次來初始化，所以 while 外面多呼叫了一次），一旦 BIO 接收到請求，就把請求 dispatch 給 worker。而 worker 中做的事和上一個階段一樣。



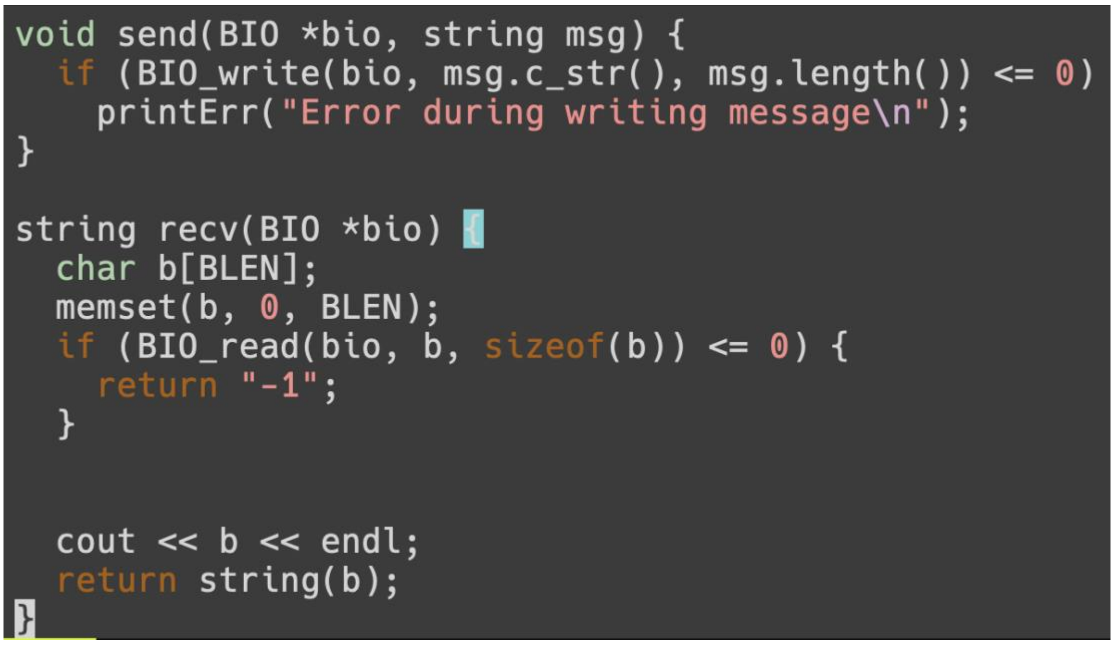
```
if (BIO_do_accept(bio) <= 0)
  printErr("Error connection");
while (true) {
  if (BIO_do_accept(bio) <= 0)
    printErr("Error connection");
  conn = BIO_pop(bio);
  dispatch(pool, handler, (void *)conn);
}
```

Main 的最後，把 ctx, bio, thread 給釋放掉。



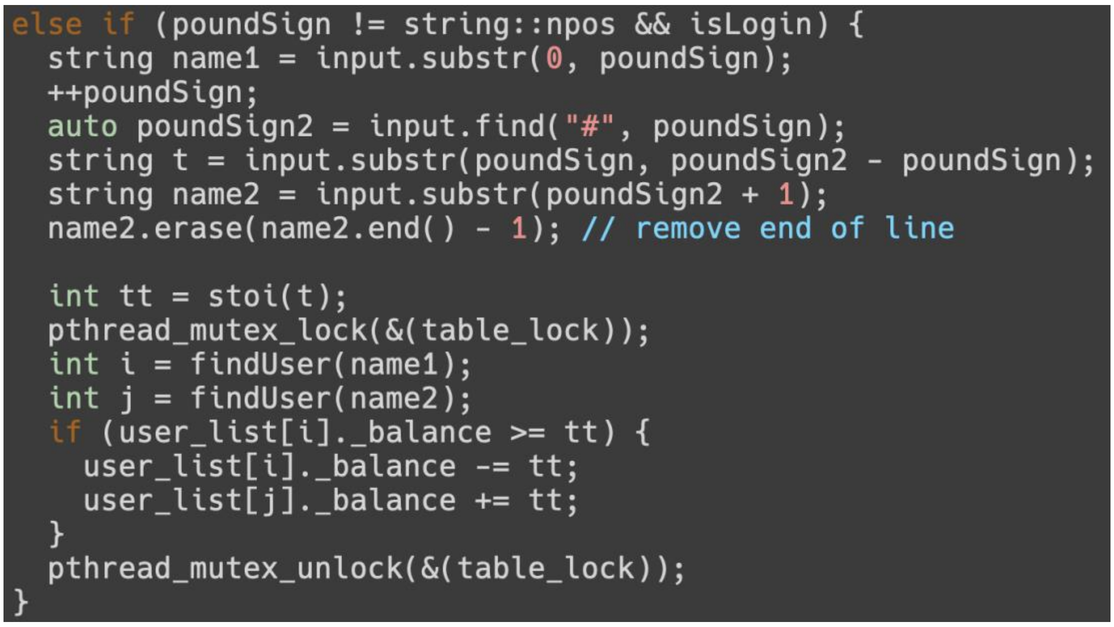
```
SSL_CTX_free(ctx);
BIO_free(bio);
BIO_free(conn);
pthread_exit(NULL);
```

另外，訊息收發的部分，overload 了 send, recv 函數（下面的 client 部分亦同），使用的是 BIO_write 和 BIO_read，在傳輸時，會根據一開始讀取的 private key 和對方的 public key 來進行加密傳輸。



```
void send(BIO *bio, string msg) {
  if (BIO_write(bio, msg.c_str(), msg.length()) <= 0)
    printErr("Error during writing message\n");
}

string recv(BIO *bio) {
  char b[BLEN];
  memset(b, 0, BLEN);
  if (BIO_read(bio, b, sizeof(b)) <= 0) {
    return "-1";
  }


  cout << b << endl;
  return string(b);
}
```

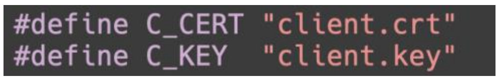
新指令處理：若有已經登入的 client 傳送<name1>#<amount>#<name2>的話，會直接進行餘額的加減。



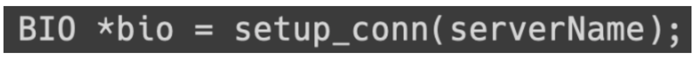
```
else if (poundSign != string::npos && isLogin) {
  string name1 = input.substr(0, poundSign);
  ++poundSign;
  auto poundSign2 = input.find("#", poundSign);
  string t = input.substr(poundSign, poundSign2 - poundSign);
  string name2 = input.substr(poundSign2 + 1);
  name2.erase(name2.end() - 1); // remove end of line

  int tt = stoi(t);
  pthread_mutex_lock(&(table_lock));
  int i = findUser(name1);
  int j = findUser(name2);
  if (user_list[i]._balance >= tt) {
    user_list[i]._balance -= tt;
    user_list[j]._balance += tt;
  }
  pthread_mutex_unlock(&(table_lock));
}
```

Client 端：

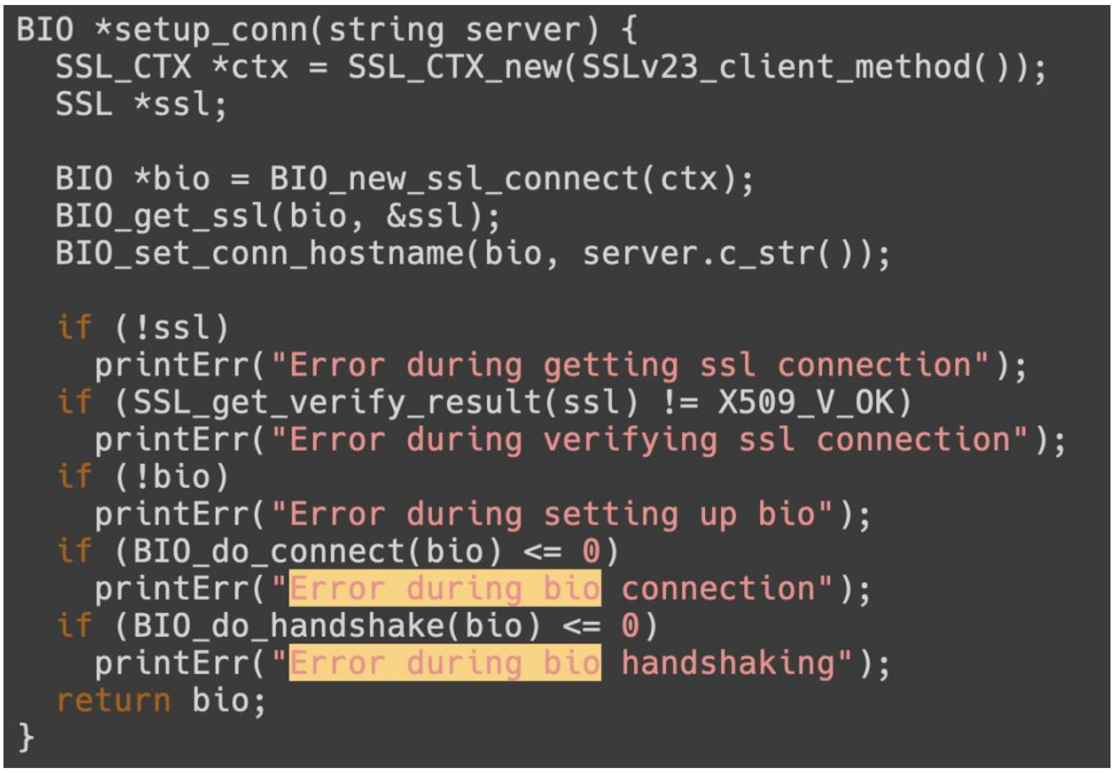
一開始一樣進行初始化，和 server 相同的部分不再贅述，不同的是的 key 變成 C-CERT 和 C-KEY，以及使用的是 SSLv23 client method，



```
#define C_CERT "client.crt"
#define C_KEY  "client.key"
```

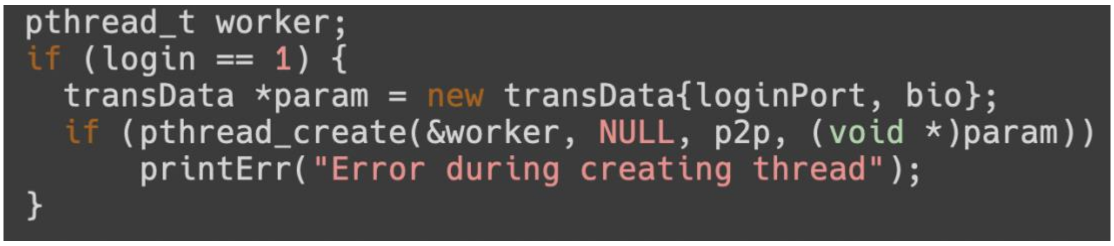


```
BIO *bio = setup_conn(serverName);
```

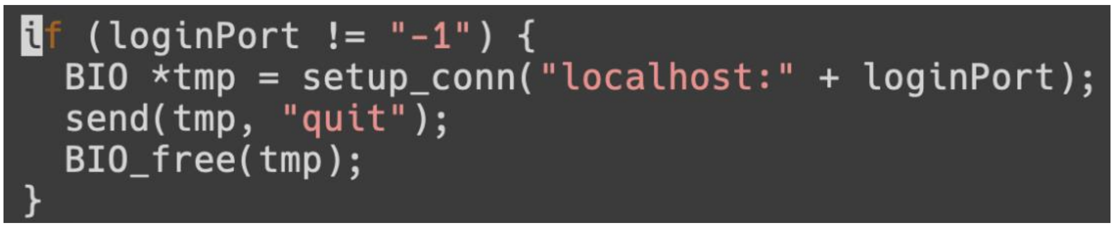


```
BIO *setup_conn(string server) {
  SSL_CTX *ctx = SSL_CTX_new(SSLv23_client_method());
  SSL *ssl;

  BIO *bio = BIO_new_ssl_connect(ctx);
  BIO_get_ssl(bio, &ssl);
  BIO_set_conn_hostname(bio, server.c_str());

  if (!ssl)
    printErr("Error during getting ssl connection");
  if (SSL_get_verify_result(ssl) != X509_V_OK)
    printErr("Error during verifying ssl connection");
  if (!bio)
    printErr("Error during setting up bio");
  if (BIO_do_connect(bio) <= 0)
    printErr("Error during bio connection");
  if (BIO_do_handshake(bio) <= 0)
    printErr("Error during bio handshaking");
  return bio;
}
```

使用者在輸入後，會建立一個 p2p 的 thread，用來接收別人的轉帳請求。
並在程式結束的時候，自己 connect 上去把它關掉。

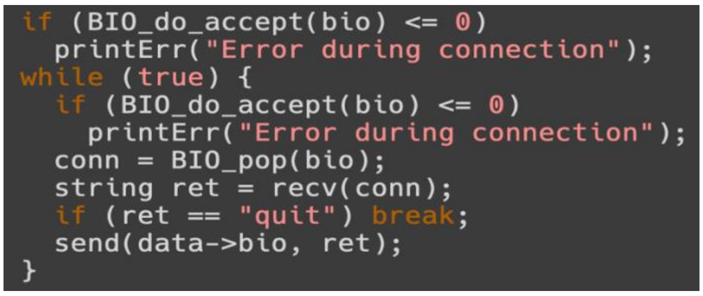


```
pthread_t worker;
if (login == 1) {
  transData *param = new transData{loginPort, bio};
  if (pthread_create(&worker, NULL, p2p, (void *)param))
      printErr("Error during creating thread");
}
```



```
if (loginPort != "-1") {
  BIO *tmp = setup_conn("localhost:" + loginPort);
  send(tmp, "quit");
  BIO_free(tmp);
}
```

而 p2p 就是在 client 建立一個監聽的 server，做法和 server 一樣，故不贅述。
其中，當 p2p 接收到請求時，會轉送給 server，讓 server 更新帳戶，而收到 quit 時，則會 close 掉。



```
if (BIO_do_accept(bio) <= 0)
  printErr("Error during connection");
while (true) {
  if (BIO_do_accept(bio) <= 0)
    printErr("Error during connection");
  conn = BIO_pop(bio);
  string ret = recv(conn);
  if (ret == "quit") break;
  send(data->bio, ret);
}
```

處理 transaction 的部分，會先強制作一次 list 來更新上線名單。

```
case 't':
case 'T': {
  // update list before transaction
  send(bio, "List\n");
  string ret = recv(bio);
  // update user list
  userList = listParser(ret, bal, userNum);
  printList(userList, bal, userNum);
```

接著會請使用者輸入要轉帳對象以及餘額，然後連上該使用者的 p2p server，傳送轉帳請求

```
int u;
int t;
cout << "Please enter the user number(1~" << userList.size() <<  "): ";
cin >> u;
```

```
cout << "Please enter the transaction amount: ";
cin >> t;
```

```
--u;
BIO *p2pb = setup_conn(userList[u]._ip + ":" + userList[u]._port);
send(p2pb, loginName + "#" + to_string(t) + "#" + userList[u]._name + "\n");
BIO_free(p2pb);
break;
```

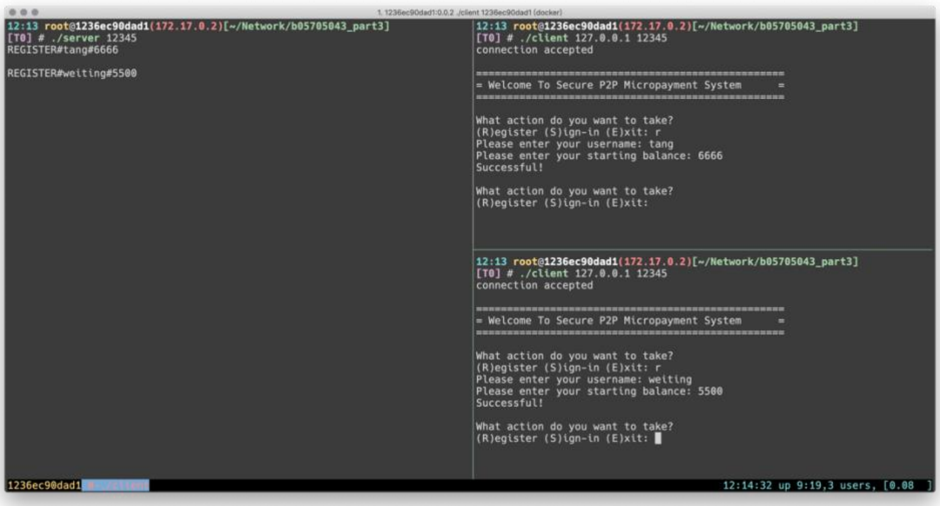
金鑰產生方式：

用指令：openssl req -x509 -out server.crt -new -newkey rsa:1024 -nodes -keyout server.key 以及

openssl req -x509 -out client.crt -new -newkey rsa:1024 -nodes -keyout client.key 來產生 server 和 client 的公私鑰，格式為 x509，長度 1024。

五、所實作的各功能截圖

註冊：



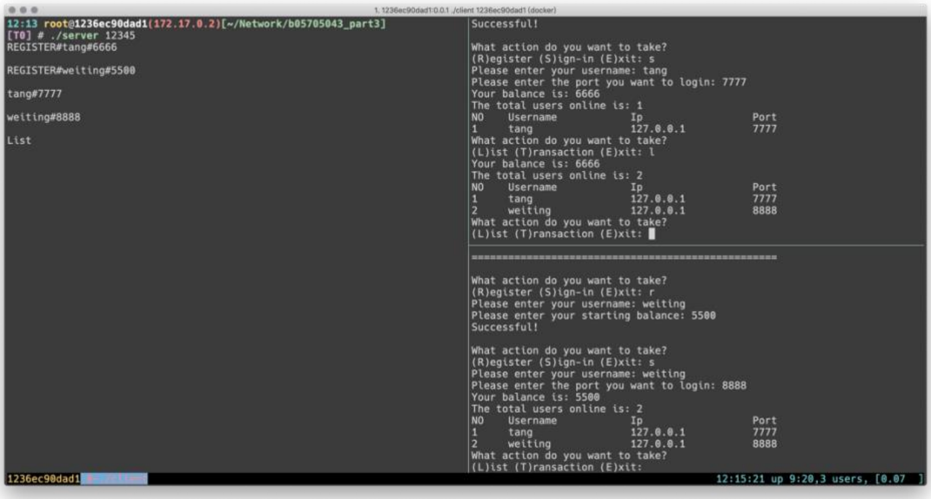
```
1. 1236ec90dad1:0.0.2 ./client 1236ec90dad1 (docker)

12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
REGISTER#tang#6666

REGISTER#weiting#5500
```

```
12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./client 127.0.0.1 12345
connection accepted

==================================================
= Welcome To Secure P2P Micropayment System      =
==================================================

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: tang
Please enter your starting balance: 6666
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit:
```

```
12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./client 127.0.0.1 12345
connection accepted

==================================================
= Welcome To Secure P2P Micropayment System      =
==================================================

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: weiting
Please enter your starting balance: 5500
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit: 
```

```
1236ec90dad1 0:./client                                12:14:32 up 9:19,3 users, [0.08  ]
```

登入：



```
1. 1236ec90dad1:0.0.1 ./client 1236ec90dad1 (docker)
```

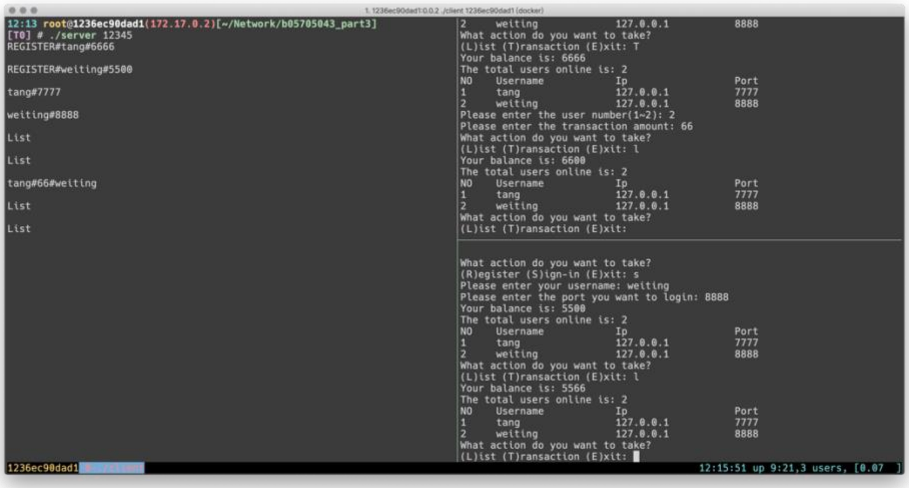
```
12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
REGISTER#tang#6666

REGISTER#weiting#5500

tang#7777

weiting#8888

List
```

```
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit: s
Please enter your username: tang
Please enter the port you want to login: 7777
Your balance is: 6666
The total users online is: 1
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
What action do you want to take?
(L)ist (T)ransaction (E)xit: l
Your balance is: 6666
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: █
```

```
==================================================

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: weiting
Please enter your starting balance: 5500
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit: s
Please enter your username: weiting
Please enter the port you want to login: 8888
Your balance is: 5500
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit:
```

```
1236ec90dad1                                    12:15:21 up 9:20,3 users, [0.07  ]
```

List & Transaction：



```
1. 1236ec90dad1:0.0.2 ./client 1236ec90dad1 (docker)
```

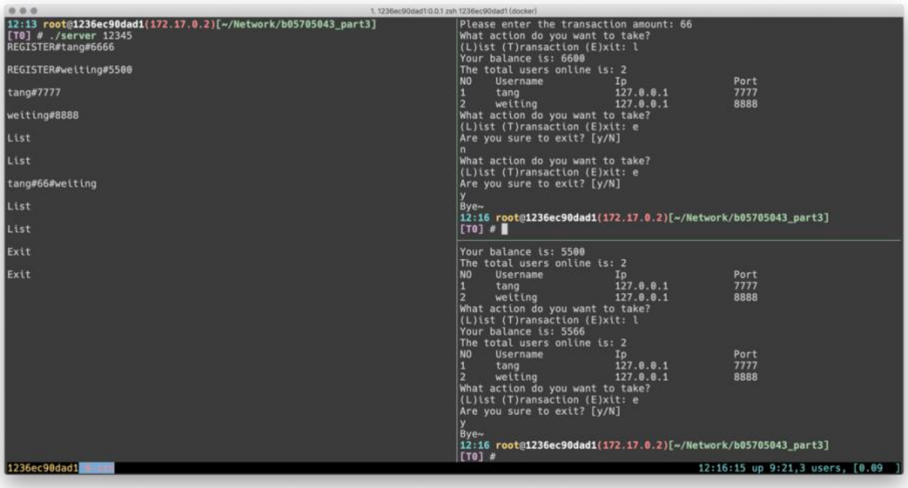
```
12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
REGISTER#tang#6666

REGISTER#weiting#5500

tang#7777

weiting#8888

List

List

tang#66#weiting

List

List
```

```
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: T
Your balance is: 6666
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
Please enter the user number(1~2): 2
Please enter the transaction amount: 66
What action do you want to take?
(L)ist (T)ransaction (E)xit: l
Your balance is: 6600
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit:
```

```

What action do you want to take?
(R)egister (S)ign-in (E)xit: s
Please enter your username: weiting
Please enter the port you want to login: 8888
Your balance is: 5500
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: l
Your balance is: 5566
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: █
```

```
1236ec90dad1                                    12:15:51 up 9:21,3 users, [0.07  ]
```

Exit：



```
1. 1236ec90dad1:0.0.1 zsh 1236ec90dad1 (docker)
```

```
12:13 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
REGISTER#tang#6666

REGISTER#weiting#5500

tang#7777

weiting#8888

List

List

tang#66#weiting

List

List

Exit

Exit
```

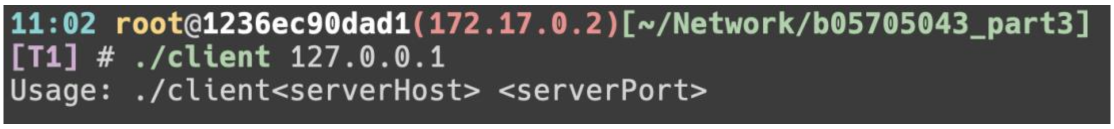
```
Please enter the transaction amount: 66
What action do you want to take?
(L)ist (T)ransaction (E)xit: l
Your balance is: 6600
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: e
Are you sure to exit? [y/N]
n
What action do you want to take?
(L)ist (T)ransaction (E)xit: e
Are you sure to exit? [y/N]
y
Bye~
12:16 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # █
```

```
Your balance is: 5500
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: l
Your balance is: 5566
The total users online is: 2
NO    Username             Ip                  Port
1     tang                 127.0.0.1           7777
2     weiting              127.0.0.1           8888
What action do you want to take?
(L)ist (T)ransaction (E)xit: e
Are you sure to exit? [y/N]
y
Bye~
12:16 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] #
```

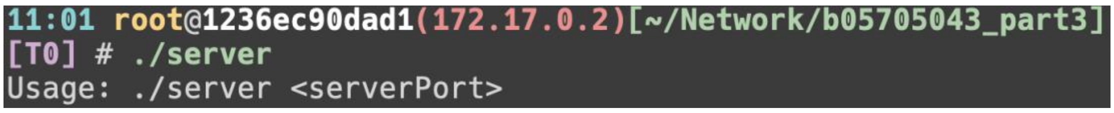
```
1236ec90dad1                                    12:16:15 up 9:21,3 users, [0.09  ]
```

六、Bonus 截圖與展示

1. 輸入的參數數量錯誤

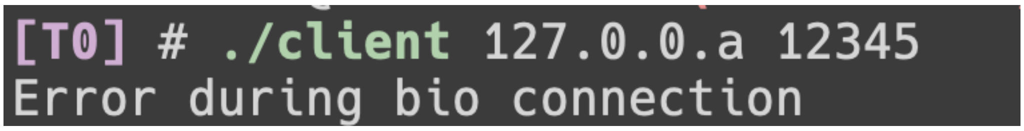


```
11:02 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T1] # ./client 127.0.0.1
Usage: ./client<serverHost> <serverPort>
```



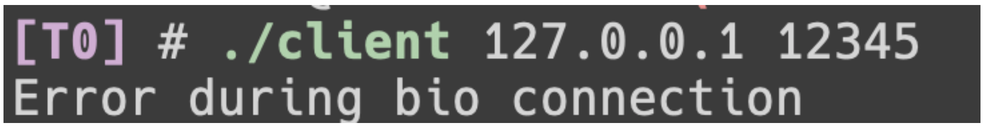
```
11:01 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server
Usage: ./server <serverPort>
```

2. 輸入的 port 不是數字



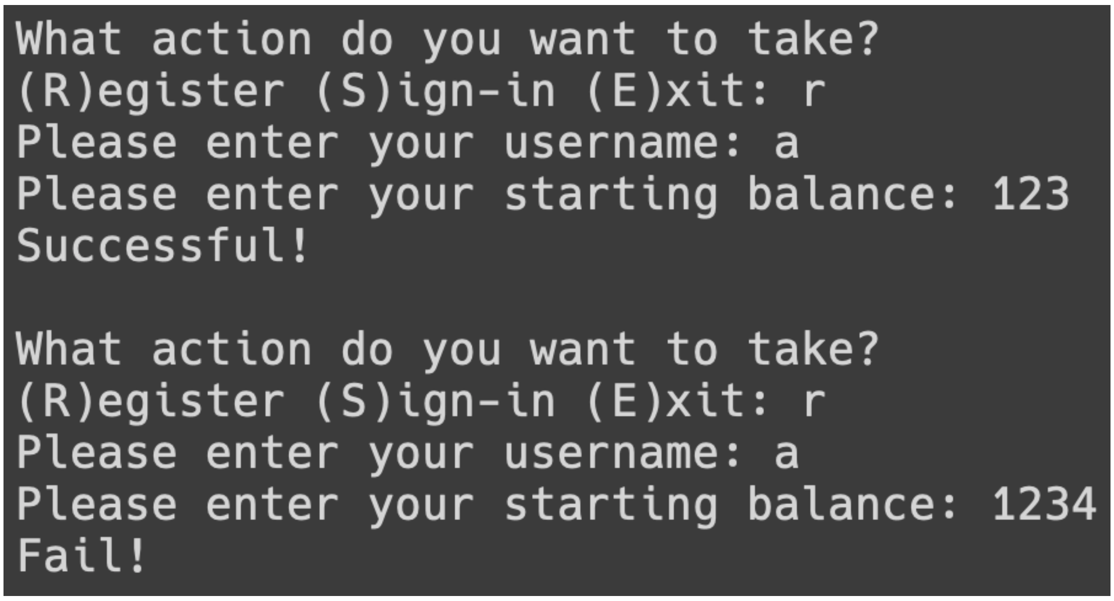
```
[T0] # ./client 127.0.0.a 12345
Error during bio connection
```

3. Server 尚未啟動



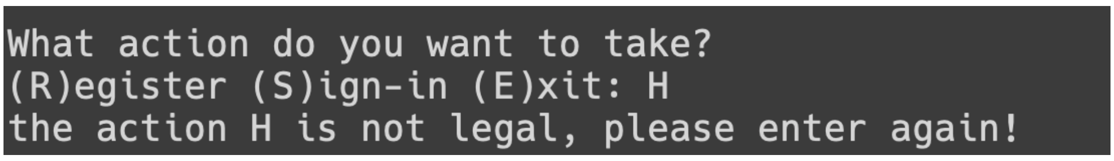
```
[T0] # ./client 127.0.0.1 12345
Error during bio connection
```

4. 重複註冊



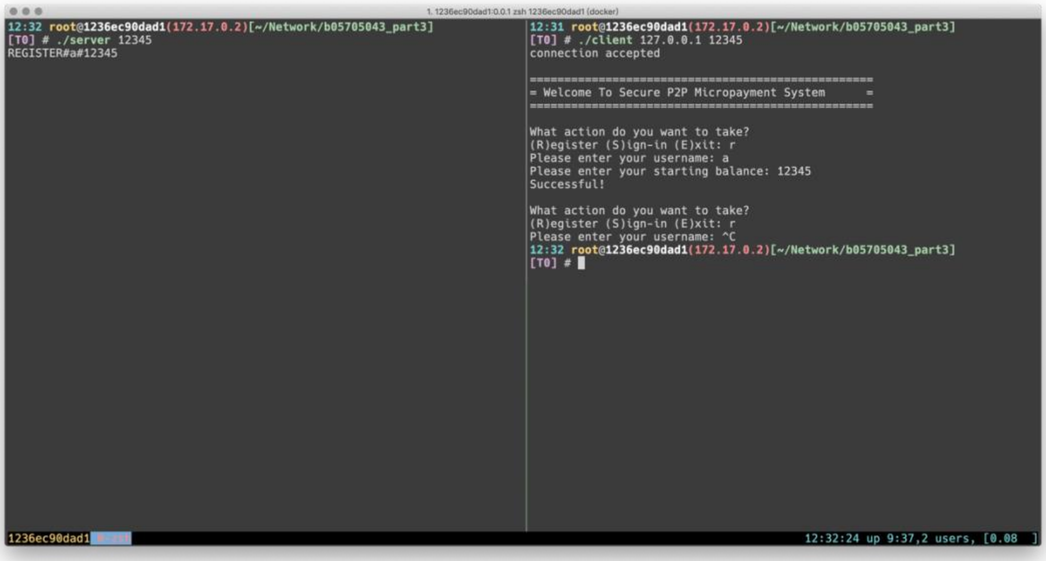
```
What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: a
Please enter your starting balance: 123
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: a
Please enter your starting balance: 1234
Fail!
```

5. 錯誤指令輸入



```
What action do you want to take?
(R)egister (S)ign-in (E)xit: H
the action H is not legal, please enter again!
```

6. 若 client 端中斷，則復原狀態，server 繼續執行。

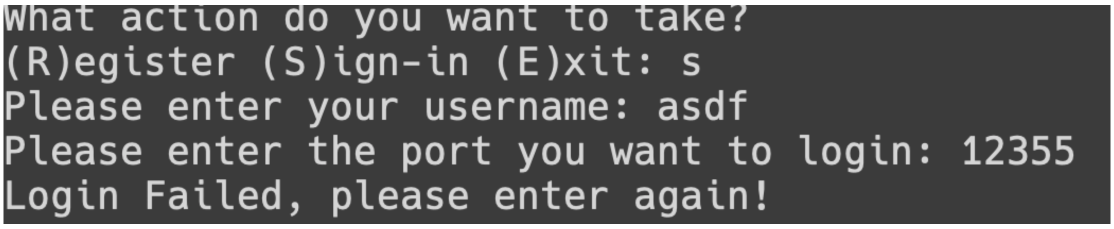


```
12:32 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./server 12345
REGISTER#a#12345
```

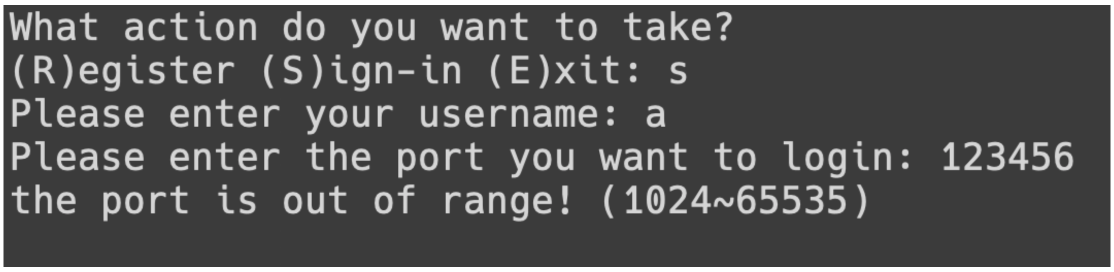
```
12:31 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] # ./client 127.0.0.1 12345
connection accepted

=================================================
= Welcome To Secure P2P Micropayment System      =
=================================================

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: a
Please enter your starting balance: 12345
Successful!

What action do you want to take?
(R)egister (S)ign-in (E)xit: r
Please enter your username: ^C
12:32 root@1236ec90dad1(172.17.0.2)[~/Network/b05705043_part3]
[T0] #
```

```
1236ec90dad1                                       12:32:24 up 9:37,2 users, [0.08  ]
```

7. 登入錯誤

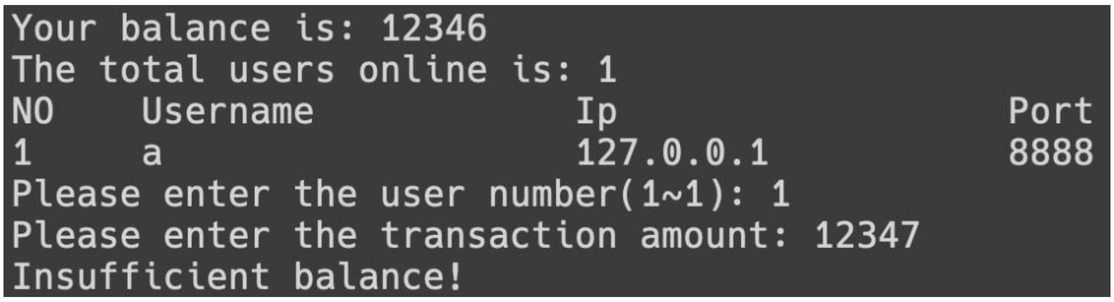


```
What action do you want to take?
(R)egister (S)ign-in (E)xit: s
Please enter your username: asdf
Please enter the port you want to login: 12355
Login Failed, please enter again!
```



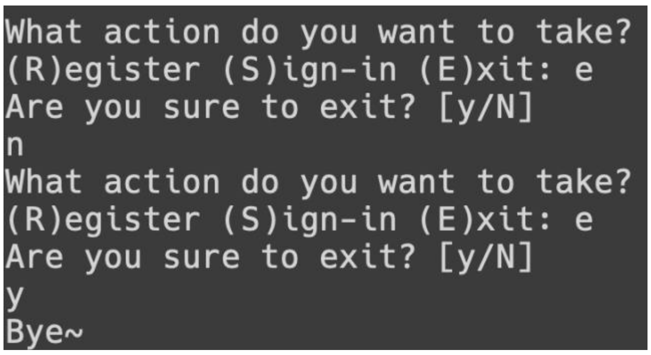
```
What action do you want to take?
(R)egister (S)ign-in (E)xit: s
Please enter your username: a
Please enter the port you want to login: 123456
the port is out of range! (1024~65535)
```

8. 餘額不足



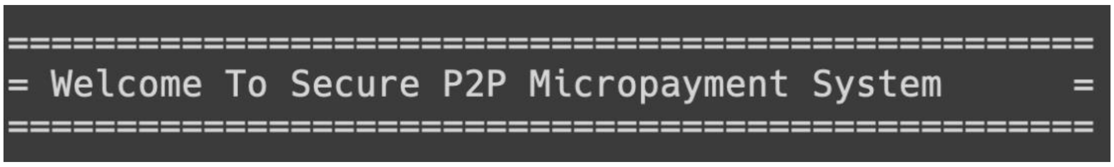
```
Your balance is: 12346
The total users online is: 1
NO    Username                 Ip                      Port
1     a                        127.0.0.1               8888
Please enter the user number(1~1): 1
Please enter the transaction amount: 12347
Insufficient balance!
```

9. Exit control

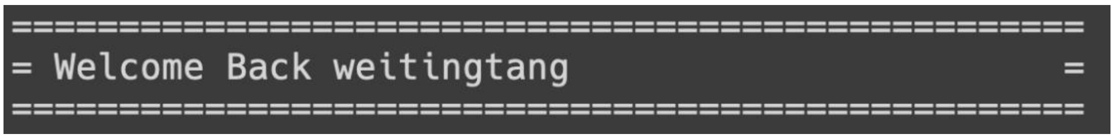


```
What action do you want to take?
(R)egister (S)ign-in (E)xit: e
Are you sure to exit? [y/N]
n
What action do you want to take?
(R)egister (S)ign-in (E)xit: e
Are you sure to exit? [y/N]
y
Bye~
```

10. Welcome 畫面



```
==================================================
= Welcome To Secure P2P Micropayment System      =
==================================================
```



```
==================================================
= Welcome Back weitingtang                       =
==================================================
```