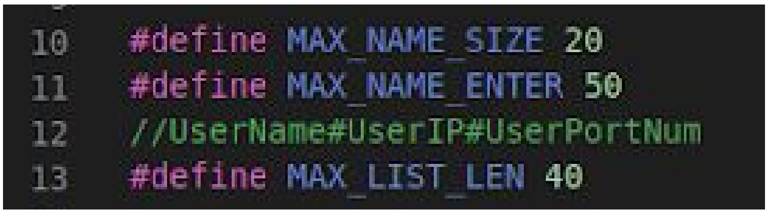
Computer Networks Programming Assignment 操作說明

圖資五 b05106010 黃冠文

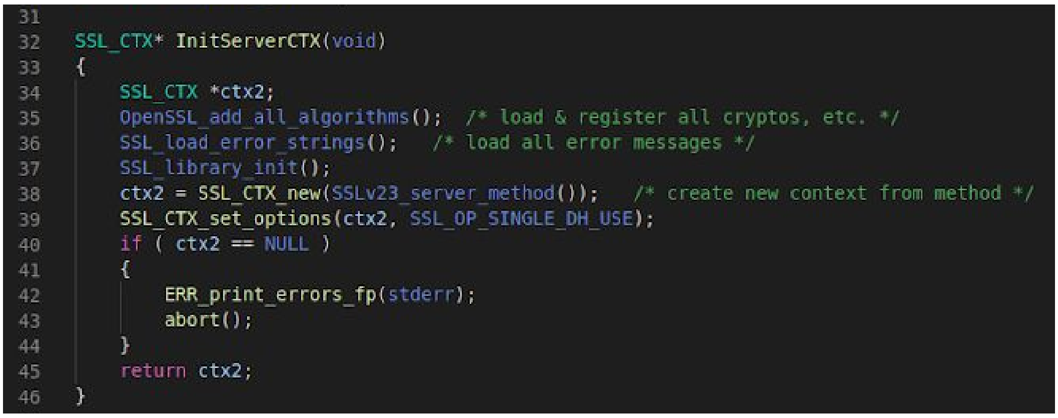
- **執行環境：**Ubuntu 20.04.1 LTS
- **程式語言：**C
- **server 處理邏輯說明：**

1. include 所需 libraries
    a. 基本需求 (建立 socket 與字串處理)：<stdio.h>、<stdlib.h>、<sys/socket.h>、<string.h>、<arpa/inet.h>、<unistd.h>
    b. threading： <pthread.h>
    c. 安全傳輸： <openssl/bio.h>、<openssl/ssl.h>、<openssl/err.h>、<errno.h>、<malloc.h>、<sys/types.h>、<netinet/in.h>、<resolv.h>
2. 定義各項最大值：



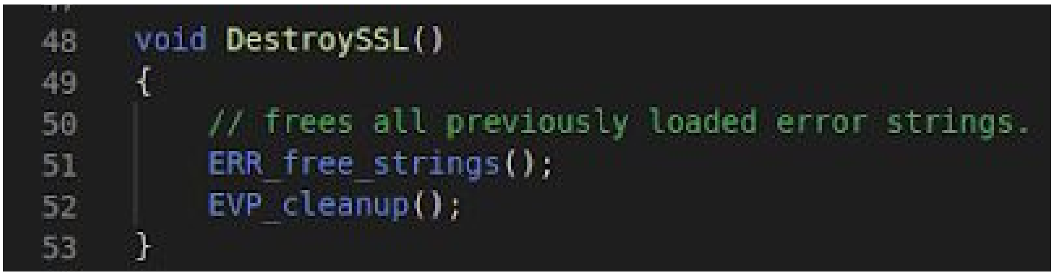
```c
#define MAX_NAME_SIZE 20
#define MAX_NAME_ENTER 50
//UserName#UserIP#UserPortNum
#define MAX_LIST_LEN 40
```

3. 安全傳輸相關機制
    a. connection initiation：**SSL_CTX* InitServerCTX(void)**



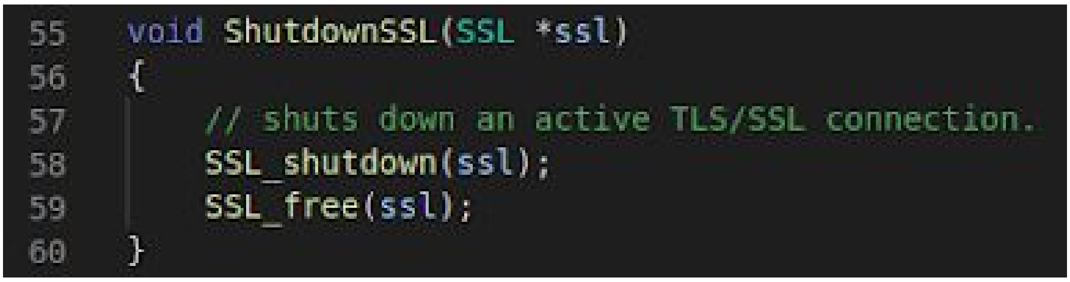
```c

SSL_CTX* InitServerCTX(void)
{
    SSL_CTX *ctx2;
    OpenSSL_add_all_algorithms();  /* load & register all cryptos, etc. */
    SSL_load_error_strings();   /* load all error messages */
    SSL_library_init();
    ctx2 = SSL_CTX_new(SSLv23_server_method());   /* create new context from method */
    SSL_CTX_set_options(ctx2, SSL_OP_SINGLE_DH_USE);
    if ( ctx2 == NULL )
    {
        ERR_print_errors_fp(stderr);
        abort();
    }
    return ctx2;
}
```

    b. error strings freeing：**void DestroySSL()**



```c
void DestroySSL()
{
    // frees all previously loaded error strings.
    ERR_free_strings();
    EVP_cleanup();
}
```

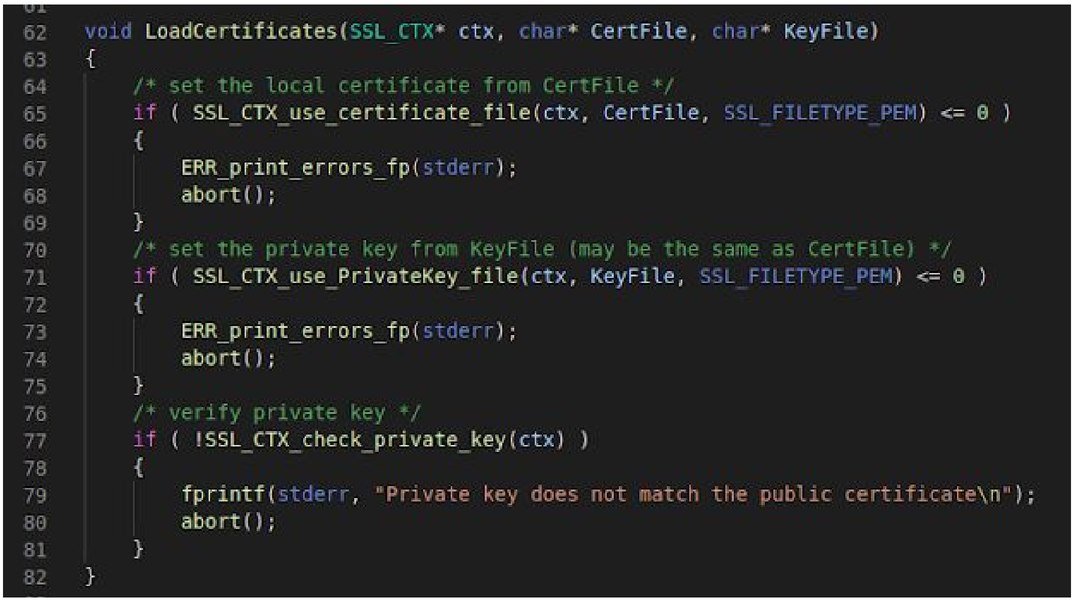
c. connection shutdown： **void ShutdownSSL(SSL *ssl)**



```c
void ShutdownSSL(SSL *ssl)
{
    // shuts down an active TLS/SSL connection.
    SSL_shutdown(ssl);
    SSL_free(ssl);
}
```

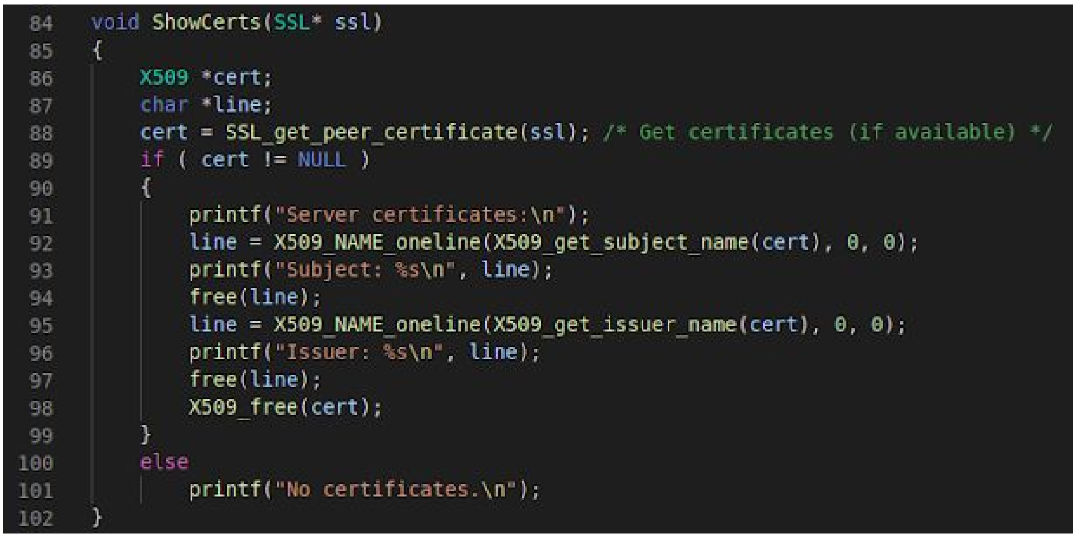
d. certificate loading：

void LoadCertificates(SSL_CTX* ctx, char* CertFile, char* KeyFile)



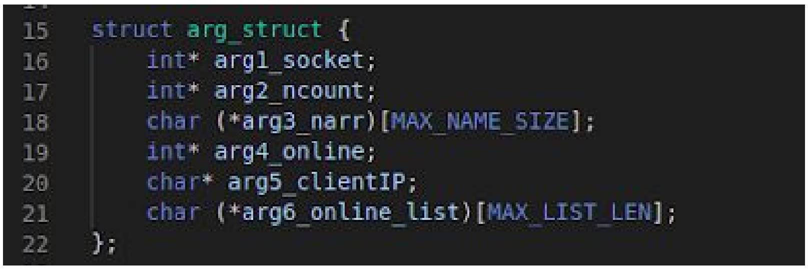
```c
void LoadCertificates(SSL_CTX* ctx, char* CertFile, char* KeyFile)
{
    /* set the local certificate from CertFile */
    if ( SSL_CTX_use_certificate_file(ctx, CertFile, SSL_FILETYPE_PEM) <= 0 )
    {
        ERR_print_errors_fp(stderr);
        abort();
    }
    /* set the private key from KeyFile (may be the same as CertFile) */
    if ( SSL_CTX_use_PrivateKey_file(ctx, KeyFile, SSL_FILETYPE_PEM) <= 0 )
    {
        ERR_print_errors_fp(stderr);
        abort();
    }
    /* verify private key */
    if ( !SSL_CTX_check_private_key(ctx) )
    {
        fprintf(stderr, "Private key does not match the public certificate\n");
        abort();
    }
}
```

e. certificate showing： **void ShowCerts(SSL* ssl)**



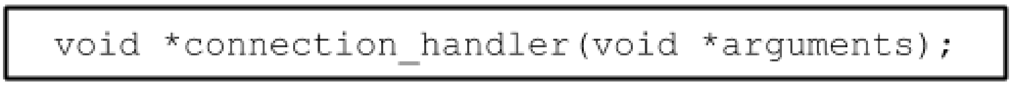
```c
void ShowCerts(SSL* ssl)
{
    X509 *cert;
    char *line;
    cert = SSL_get_peer_certificate(ssl); /* Get certificates (if available) */
    if ( cert != NULL )
    {
        printf("Server certificates:\n");
        line = X509_NAME_oneline(X509_get_subject_name(cert), 0, 0);
        printf("Subject: %s\n", line);
        free(line);
        line = X509_NAME_oneline(X509_get_issuer_name(cert), 0, 0);
        printf("Issuer: %s\n", line);
        free(line);
        X509_free(cert);
    }
    else
        printf("No certificates.\n");
}
```

4. 建立一個名為 arg_struct 的 struct 存取各項所需資訊 (例如socket的記憶體位置與數量、在線人員名單與ip位址等）



```
struct arg_struct {
    int* arg1_socket;
    int* arg2_ncount;
    char (*arg3_narr)[MAX_NAME_SIZE];
    int* arg4_online;
    char* arg5_clientIP;
    char (*arg6_online_list)[MAX_LIST_LEN];
};
```

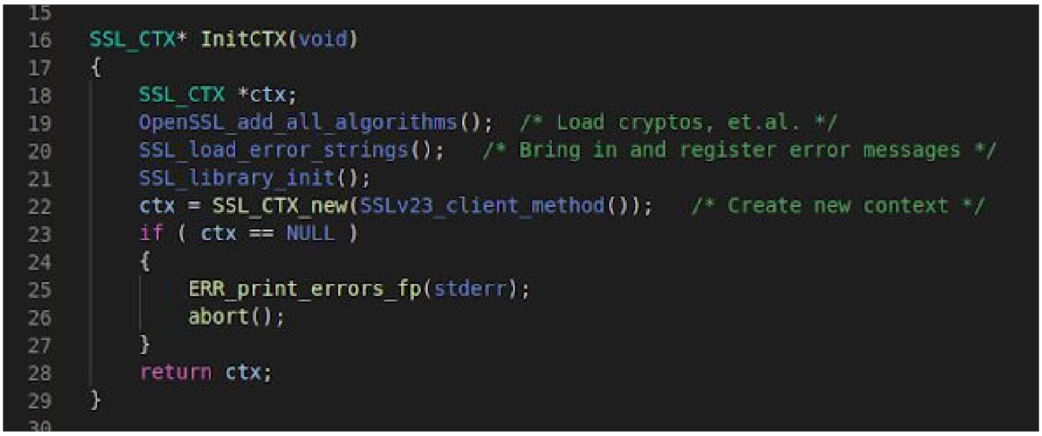
5. 建立一個名為 connection_handler 的 function 處理與回覆 client 發出的各項請求，包含註冊、登入、請求在線清單、client 之間小額交易以及離開，一共五項功能：



```
void *connection_handler(void *arguments);
```

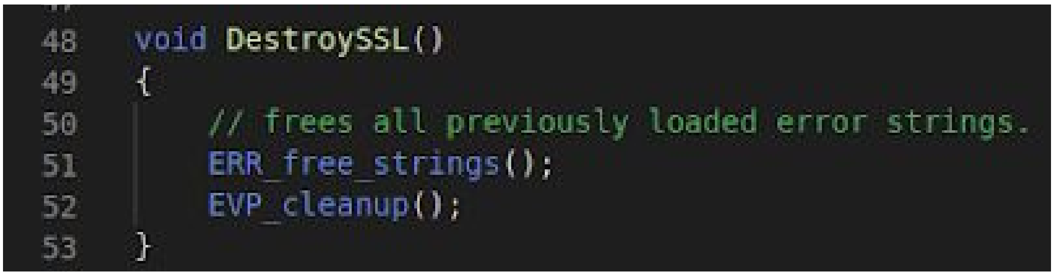
6. 先建立一個 socket，並設定為手動輸入port number
7. bind&listen
8. client 驗證 server load 好的 certificate，進行三方握手連線
9. 接著 server 利用 pthread 處理多位使用者同時連線，assign 給每位使用者一人一個connection_handler，並利用 arg_struct 存取各使用者相關資訊以轉換成清單

- **client 處理邏輯說明：**
  1. include 所需 libraries
     a. 基本需求 (建立 socket 與字串處理)：<stdio.h>、<stdlib.h>、<sys/socket.h>、<string.h>、<arpa/inet.h>、<unistd.h>
     b. 安全傳輸：<resolv.h>、<netdb.h>、<openssl/ssl.h>、<openssl/err.h>
  2. 安全傳輸相關機制
     a. connection initiation：**SSL_CTX* InitCTX(void)**



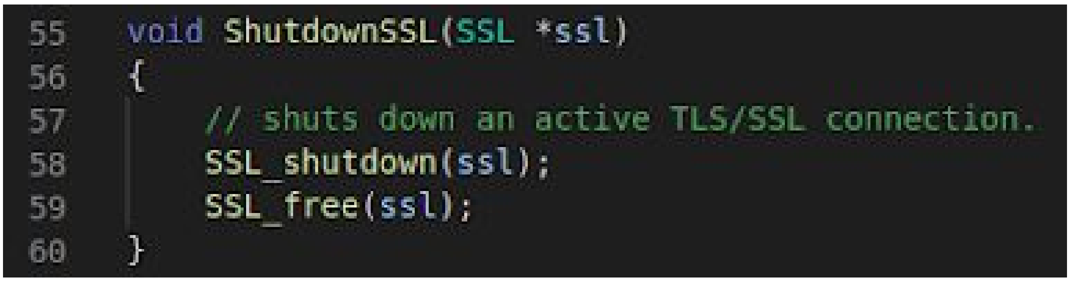
```c
SSL_CTX* InitCTX(void)
{
    SSL_CTX *ctx;
    OpenSSL_add_all_algorithms();  /* Load cryptos, et.al. */
    SSL_load_error_strings();   /* Bring in and register error messages */
    SSL_library_init();
    ctx = SSL_CTX_new(SSLv23_client_method());   /* Create new context */
    if ( ctx == NULL )
    {
        ERR_print_errors_fp(stderr);
        abort();
    }
    return ctx;
}
```

b. error strings freeing：**void DestroySSL()**



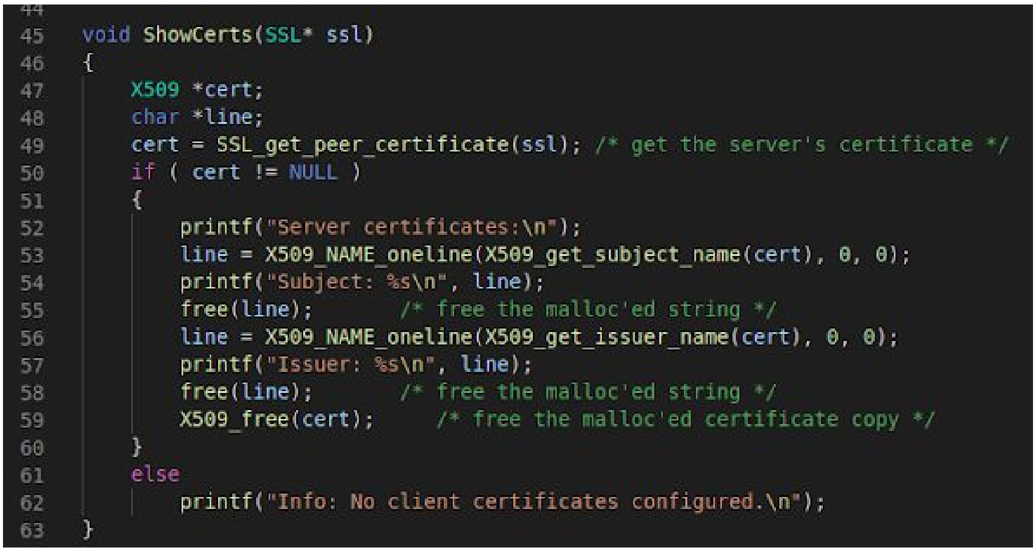
```c
void DestroySSL()
{
    // frees all previously loaded error strings.
    ERR_free_strings();
    EVP_cleanup();
}
```

c. connection shutdown：**void ShutdownSSL(SSL *ssl)**



```c
void ShutdownSSL(SSL *ssl)
{
    // shuts down an active TLS/SSL connection.
    SSL_shutdown(ssl);
    SSL_free(ssl);
}
```

d. certificate showing：**void ShowCerts(SSL* ssl)**



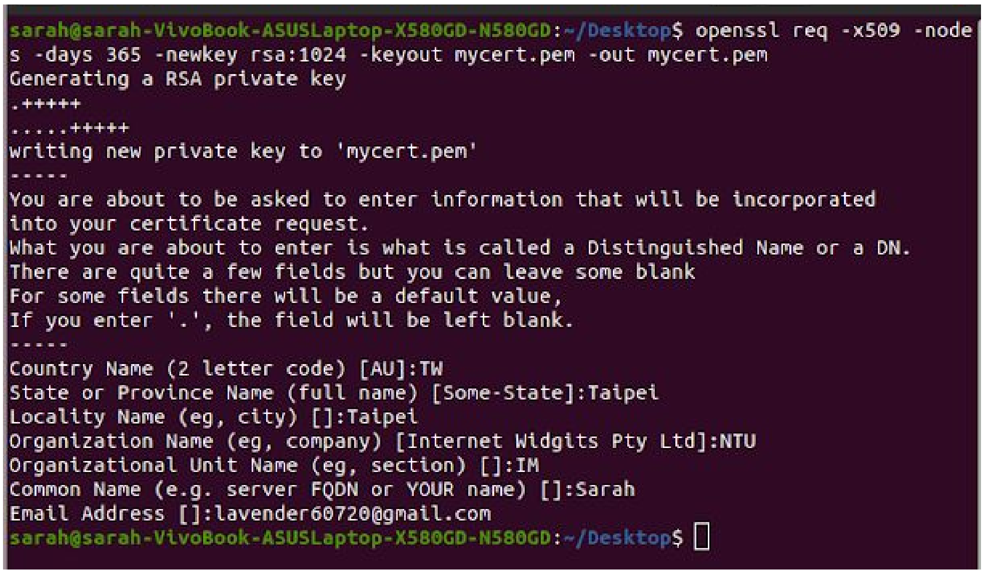
```c
void ShowCerts(SSL* ssl)
{
    X509 *cert;
    char *line;
    cert = SSL_get_peer_certificate(ssl); /* get the server's certificate */
    if ( cert != NULL )
    {
        printf("Server certificates:\n");
        line = X509_NAME_oneline(X509_get_subject_name(cert), 0, 0);
        printf("Subject: %s\n", line);
        free(line);       /* free the malloc'ed string */
        line = X509_NAME_oneline(X509_get_issuer_name(cert), 0, 0);
        printf("Issuer: %s\n", line);
        free(line);       /* free the malloc'ed string */
        X509_free(cert);     /* free the malloc'ed certificate copy */
    }
    else
        printf("Info: No client certificates configured.\n");
}
```

3. 建立要跟 server 建立連線的 socket
4. 分別以 port_num 以及 server_ip 存取欲連線之 server 端的 port number 還有 ip address
5. 驗證 server 的 certificate， 進行三方握手連線
6. 向 server 傳送要求 (如註冊、登入、查看清單、小額交易以及離開)，並收取 server 端的回覆

- **關於程式編譯與執行：**

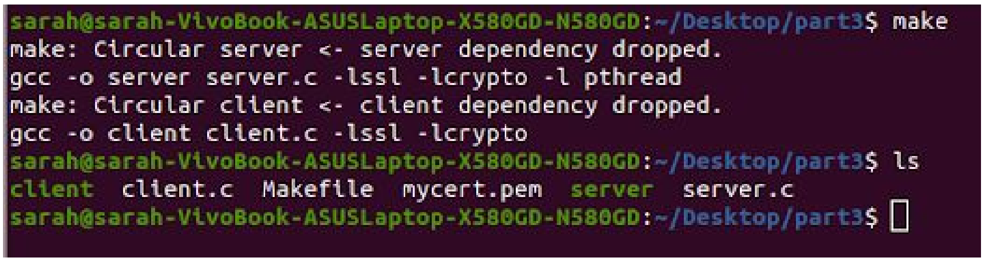
1. 開啟 terminal，利用以下 command 產生一份自己的 certificate (檔名為 mycert.pem)，將其與 server 以及 client 的程式放在同一資料夾：

   openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem



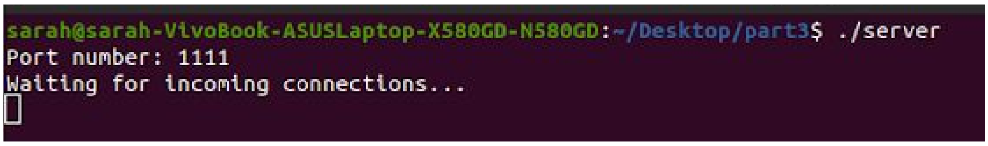
```
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop$ openssl req -x509 -node
s -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem
Generating a RSA private key
.+++++
.....+++++
writing new private key to 'mycert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taipei
Locality Name (eg, city) []:Taipei
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NTU
Organizational Unit Name (eg, section) []:IM
Common Name (e.g. server FQDN or YOUR name) []:Sarah
Email Address []:lavender60720@gmail.com
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop$ 
```

2. 將 server.c 以及 Makefile 放在同個資料夾，打開 terminal 並定位至該資料夾後，輸入make即可編譯程式：



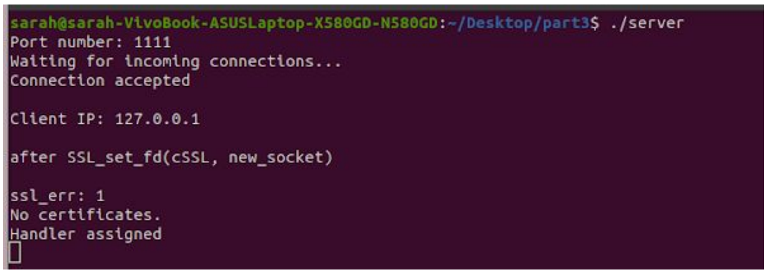
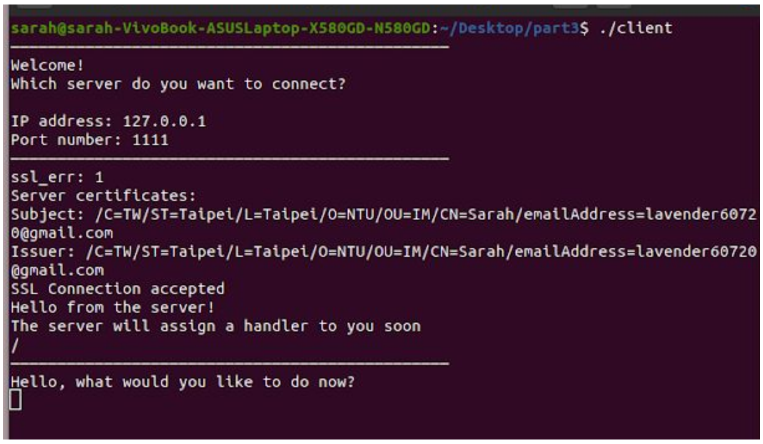
```
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ make
make: Circular server <- server dependency dropped.
gcc -o server server.c -lssl -lcrypto -l pthread
make: Circular client <- client dependency dropped.
gcc -o client client.c -lssl -lcrypto
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ ls
client  client.c  Makefile  mycert.pem  server  server.c
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ 
```

3. server 和 client 即為程式執行檔，輸入 server 的 port number 後即可得知是否成功建立 socket：

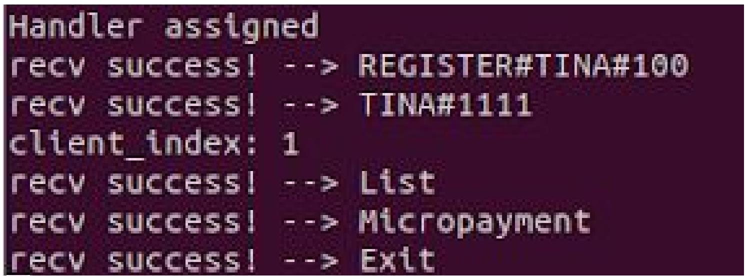


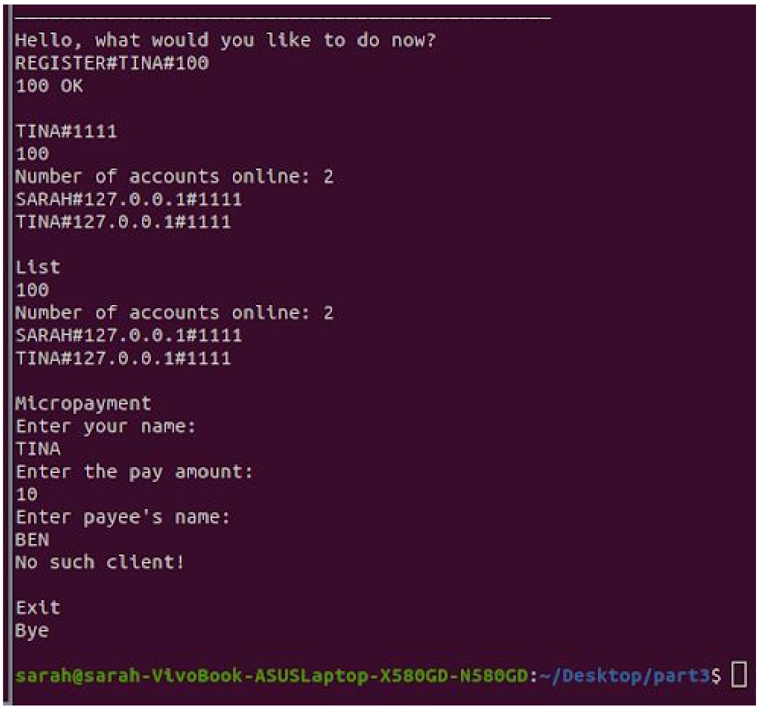
```
sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ ./server
Port number: 1111
Waiting for incoming connections...
```

4. 開啟另一個 terminal window，執行同一資料夾中的 client，輸入 ip（以主機 ip 為例）以及剛剛設定的 server port number 進行安全連線：

server 端顯示	 sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ ./server Port number: 1111 Waiting for incoming connections... Connection accepted Client IP: 127.0.0.1 after SSL_set_fd(cSSL, new_socket) ssl_err: 1 No certificates. Handler assigned
client 端顯示	 sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ ./client -- Welcome! Which server do you want to connect? IP address: 127.0.0.1 Port number: 1111 -- ssl_err: 1 Server certificates: Subject: /C=TW/ST=Taipei/L=Taipei/O=NTU/OU=IM/CN=Sarah/emailAddress=lavender6072 0@gmail.com Issuer: /C=TW/ST=Taipei/L=Taipei/O=NTU/OU=IM/CN=Sarah/emailAddress=lavender60720 @gmail.com SSL Connection accepted Hello from the server! The server will assign a handler to you soon / -- Hello, what would you like to do now?

5. 成功連線後即可使用該五項功能：

server 端顯示	 Handler assigned recv success! --> REGISTER#TINA#100 recv success! --> TINA#1111 client_index: 1 recv success! --> List recv success! --> Micropayment recv success! --> Exit

client 端顯示	

```
______________________________________________
Hello, what would you like to do now?
REGISTER#TINA#100
100 OK

TINA#1111
100
Number of accounts online: 2
SARAH#127.0.0.1#1111
TINA#127.0.0.1#1111

List
100
Number of accounts online: 2
SARAH#127.0.0.1#1111
TINA#127.0.0.1#1111

Micropayment
Enter your name:
TINA
Enter the pay amount:
10
Enter payee's name:
BEN
No such client!

Exit
Bye

sarah@sarah-VivoBook-ASUSLaptop-X580GD-N580GD:~/Desktop/part3$ 
```