

## 計算機網路 final project

### 1. 組員：

b03902127 資工三 郭曉嵐

b03902125 資工三 林映廷

b03902129 資工三 陳鵬宇

### 2. 組員分工：

UI, code, debug 三人一同分擔

### 3. User guide and protocol introduction:

先用 make 去編譯 server 跟 client，並在 server 建立 socket 連線後並讓 client 去 connect()。

我們將登入成功的介面叫做首頁(/home)，以下說明我們有關於評分標準的實作方式：

(下圖為/home的截圖)

```
----- User's lists -----
Online          Chat          ID          Account
-----
*              *              1          corey
              2          gino
              3          melody
*              4          passenger
-----

Please input the number you want to do
1.Chat Room
2.Send/Receive files
3.Historical Messages
4.Off-line Messages
5.Exit
-----
```

#### (1) Registration:

首先，我們先開一個 member.txt，一開始就假設已經存入 corey、gino、melody 這三個已經 create 好的帳號，密碼分別為 corey1、gino1、melody1，並讓 server 去判斷 client 選擇下列圖的哪個指令後，去做正確的錯誤判別。

```
===== Welcome to group 21 for cn =====
Please input the number you want to do
1.Login
2.Create new account
3.Exit
=====

My selection:
```

當 client 選擇 1 的時候，要選擇登入的帳號跟密碼，其中若帳號不存在 member.txt 裡面或密碼錯誤的話，即回傳錯誤訊息然後回到登入帳號的介面。

當 client 選擇 2 的時候，會創造新帳號跟密碼，若想創造的新帳號跟 member.txt 有的帳號重複或者帳號密碼過長( $\geq 12$ )，則印出錯誤訊息然後回到一開始創帳號的介面。

Knocking:

當登入成功或創帳號成功後，即顯示有註冊帳號的那些人，現在有上線的人有誰(包含自己)，而這情況則是在首頁的時候。

| User's lists |      |    |           |
|--------------|------|----|-----------|
| Online       | Chat | ID | Account   |
| *            | *    | 1  | corey     |
|              |      | 2  | gino      |
|              |      | 3  | melody    |
| *            |      | 4  | passenger |

然後在傳送檔案時，可以判別哪些人 ready to receive files，此為 Knocking 的結果。

| User's lists |       |    |           |
|--------------|-------|----|-----------|
| Online       | Ready | ID | Account   |
| *            | *     | 1  | corey     |
|              |       | 2  | gino      |
|              |       | 3  | melody    |
| *            |       | 4  | passenger |

(2) Messaging:

我們寫在 chat room 裡面，建立一個 private message，下圖中的[toe toe talk]偷偷說，代表 corey 給 passenger 的私訊。在你想私訊給在線上的人的名字前面，加上/即可私訊。(ex: /passenger hello passenger)，這是代表傳給 corey 私訊給 passenger，messages 內容為 hello passenger)除了 sender 跟 receiver 之外的其他人，看不到她們兩者的對話內容。

```
/passenger hello passenger
Sat Jan 14 23:32:19 2017
```

```
[toe toe talk] corey: hello passenger
Sat Jan 14 23:32:19 2017
```

### (3) File transfer

當在首頁時輸入 2 即進入寄收檔案的 state，下圖為進入寄收檔案的 state 的截圖。

```
----- Send/Receive files -----
Please input the number you want to do
1.Send the file
2.Receive the file
Please input '/Home' if you want back
-----
```

首先有兩個人以上的話，都在首頁的時候，要有一個人先進入 receiving state，等待其他人 send 給她檔案。另一方有一人則進入 sending state，輸入正在 receiving state 的 ID 時就可以送檔案給她，再來輸入你想傳送的檔案名稱即可將此檔案傳送過去，而若在任何狀態輸入/Home 的話則都會跳回首頁。

想要 sending 的人：

```
----- Send/Receive files -----
Please input the number you want to do
1.Send the file
2.Receive the file
Please input '/Home' if you want back
-----
My selection:
1
-----
----- User's lists -----
-----
Online          Ready          ID          Account
-----
*              *              1          corey
              *              2          gino
              *              3          melody
*              *              4          passenger
-----
'You can only send to those who are online and ready!!'
Please input '/Home' if you want back
Please input the ID of user's lists that you want to send
ID:
4
-----
----- Sending -----
Receiver:passenger
Please input Files' names you want to send (no space)
Makefile
sending: Makefile
File: Makefile Tansfer Successful!!!
-----
Please input '/Home' if you want back
```

想要 receiving 的人:

```
----- Receiving -----
Please input '/Home' if you want back
Receiving:Makefile
From:corey
Receive File: Makefile Successful!!!
```

Chat room:

當在首頁時選 1 即可進去 chat room，並會顯示出現在 chat room 的人，且 在這裡輸入/Home 則會跳回首頁。

```
Please input the number you want to do
1.Chat Room
2.Send/Receive files
3.Historical Messages
4.Off-line Messages
5.Exit
-----
My selection:
1
----- User's lists -----
      Online          Chat          ID          Account
-----
      *                  1          corey
                        2          gino
                        3          melody
      *                  4          passenger
-----
Please input '/Home' if you want back
----- Chat Room -----
[system] passenger gets in the room.
```

在 chat room 裡面的人皆會看到其他 users 所說的話，且顯示說話的人是誰。

```
----- Chat Room -----
[system] gino gets in the room.

corey: hi
Sat Jan 14 23:43:17 2017

melody: hi
Sat Jan 14 23:43:40 2017
```

在這裡也可以用 Send/Receive messages 此功能，詳情請看(3)。

而紀錄時間的部分用 time\_t、struct tm 來求得，如下圖所示。

```
time_t curtime;  
struct tm *loctime;  
  
curtime = time(NULL);
```

(5) User interface:

用 printf 的方式直接印在螢幕上的方法，並藉此調整版面輸出，來去做 UI design。在每個不同的 UI 的情況，皆有說明現在要輸入什麼指令來跟 User 互動，並告知她現在該做什麼事。(如同前面截圖的結果)

(Bonus 1) Multiprocessing:

用 select 來讓 server 進行多個 client 連線，並做其他指令相對應的處理。

(Bonus 2) Historical messages:

在首頁輸入 3 即可以進入 historical messages 的 state。在 chat room 裡面的所有訊息跟傳訊息的時間，都會紀錄在 historical\_name.txt 裡面(每個人都有自己的 historical\_name.txt)，包含私訊跟被私訊的情況也會被紀錄，且輸入/Home 則會跳回首頁。

```
----- Historical Messages -----  
[toe toe talk] passenger: Hello corey  
Sat Jan 14 23:14:02 2017  
  
corey: hi  
Sat Jan 14 23:43:17 2017  
  
melody: hi  
Sat Jan 14 23:43:40 2017  
-----
```

### (Bonus 3) Off-line messages:

在首頁輸入 4 即可以進入 Off-line messages 的 state。離線的時候，會紀錄 所有訊息跟時間，但不包含私訊的內容，因為離線時無法被私訊。而 offline\_name.txt 用來紀錄這些資料。(每個人都有自己的 offline\_name.txt)，且輸入/Home 則會跳回首頁。

下圖為以 gino(第三人)角度所看到的 Off-line messages。

### 4.Process flow diagram:

每個 state 的狀態跟在哪個情況下會跳入哪個 state，請參考下列圖片。

(x,y)是 state 的情況，而 y 是 x 的 subtask。

箭頭所標示的方向為，當發生數字所描述的事情時，會跳入下個 state 或重覆進 去原本的 state。

