

Computer Network Final Project

b05902049 資工三 簡崇安 b05902109 資工三 柯上優

User & Operator Guide

- 這是一個可支援多人在線，只接受兩人一對一聊天的線上聊天室
- 提供功能
 - 註冊、登入/自動登入 (**bonus**「自動登入」)
 - 和「在線使用者」即時聊天
 - 內建貼圖傳送 (**bonus**「內建貼圖」)
 - 留言給「離線使用者」，當對方進到兩人的聊天室後，會立刻看到離線時錯過的資訊
 - 可以查看過去聊天紀錄，最大聊天紀錄筆數預設1000筆，可以從原始碼更改
 - 好友列表，提供查詢你的好友 (**bonus**「好友列表」)
 - 黑名單，不讓你討厭的人來和你聊天 (**bonus**「壞名單」)
- 保證
 - 密碼被加密傳遞 (**bonus**「加密傳遞密碼」)
 - 使用MD5加密，`#include <openssl/md5.h>`
 - 聊天到一半段線，當前的聊天紀錄保證被儲存

Instructions on how to run server & clients

- 編譯只需要輸入 'make' 即可
- 啟動 client 端前請務必先啟動 server 端，client 端單獨開啟是無法使用的
- server 的啟動方法

`$/server [port]` `#[port]`是server端綁定的port位置

- client 的啟動（與bonus「自動連線」）

```

$./client [IP:port]          #[IP:port]是server的位置與port number，以下同義
$./client [Host:port]
$./client [IP:port] -auto [client_name] [client_password] #自動登入模式
$./client [Host:port] -auto [client_name] [client_password]

```

- 接下來這段是關於註冊/手動登入的部分
 - 若自動登入成功，則省略這步驟
 - 若遇到已被註冊/已有他人使用這組帳號登入，則會在警告標語跳出後幾秒鐘內刷新

```
Please enter sign-up[s] or login[l] or quit[q]
s
username: a
password: a
```

- 以下將用兩位使用者為 a、b 作為範例
- 成功進到聊天室大廳後，會有以下畫面
 - 若想和其他使用這聊天，輸入 c，並輸入其使用者名稱
 - 若想安全離開，輸入 q

```
You are in the lobby now.
Enter specific command
- chat-with-friend[c]
- quit[q]
- good-friend show/add/delete[gf s/a/d]
- bad-friend show/add/delete[bf s/a/d]
c
Chat mate name: b
```

- 好友列表與黑名單，可以使用指令觀看/新增/刪除
- 只有有註冊的使用者可以被加入/刪除
- 黑名單的對象不能和你聊天

```
gf s

==== good-friend ====
a
=====

bf s

==== bad-friend ====
[empty]
=====
```

- 進入兩人聊天室後，會看到之前錯過的訊息
- 特殊指令：
 - 查看歷史紀錄
 - 停止聊天，對方會跳出你已離線的警告標語

```
You are in the chat room now.
Feel free to chat!
Enter specific command
- history[$ h]
- send-file[$ s [fileName]]
- stop-chat[$ sc]
==== un read message ====
b:q
=====
b: hello
b: byebye
user 'b' is not online, but you can still send the message.
user 'b' is online, now.
b: hello
```

b: again

- 寄送檔案，須確保檔案的存在，且要求該檔案位於執行位置的相同資料夾
- 經過測試，100MB大小檔案已可順利傳送

```
in the 'a' side
$ send a.txt

in the 'b' side
=> get file from 'a', fileName is 'a.txt'
```

- 斷線的方法，建議使用系統要求方式（ex: [q]、[\$ sc]）
- ctrl+C在開發者測試下也尚無大型錯誤，但仍然不建議。

Protocol Specification

- 每種封包的大小不一，根據其傳遞內容有所不同，以下使種類
- 每種封包都包含一個共同封包 Protocol_header，其包含以下資訊
 - option: 對應到上述封包類型，使接收端知道要recv多長的資訊

封包種類	Protocol_OPTION	包含資訊
Protocol_signup	PROTOCOL_OPTION_SIGNUP	帳號、密碼（加密）
Protocol_login	PROTOCOL_OPTION_LOGIN	帳號、密碼（加密）
Protocol_startchat	PROTOCOL_OPTION_STARTCHAT PROTOCOL_OPTION_GOODFRIEND PROTOCOL_OPTION_BADFRIEND	聊天對象、 離線時沒看到的內容/朋友 列表內容
Protocol_sendmsg	PROTOCOL_OPTION_SENDMSG	聊天內容
Protocol_stopchat	PROTOCOL_OPTION_STOPCHAT	（無）
Protocol_quit	PROTOCOL_OPTION_QUIT	（無）
Protocol_sendfile	PROTOCOL_OPTION_SENDFILE	傳遞檔案的檔名
Protocol_sendfilecontent	PROTOCOL_OPTION_SENDFILECONTENT PROTOCOL_OPTION_HISTORY（聊天紀錄也 用此類型傳遞）	傳遞檔案的內容、fragflag

- status: 此封包狀態，要求、接受、拒絕
 - PROTOCOL_STATUS_REQ
 - PROTOCOL_STATUS_SUCC
 - PROTOCOL_STATUS_FAIL
- datalen: 封包長度
- 每個使用者的狀態會在以下各項間轉移，使用者與伺服器都會同步紀錄狀態

狀態	解釋
CLIENT_STATUS_CONNECTONLY	連線剛建立，尚未註冊與登入
CLIENT_STATUS_WAIT	閒置狀態，可與他人聊天或被邀請
CLIENT_STATUS_CHAT_TOGETHER	兩人皆在線，可即時聊天
CLIENT_STATUS_CHAT_ALONE	僅一人在線，聊天紀錄會被保存，並在對方上線後傳遞
CLIENT_STATUS_SENDFILE	傳輸檔案狀態，使用thread處理，非main thread會進入的狀態

System & Program Design

- Server
 - 紀錄「在線者」的以下資訊
 - UserName、status、聊天對象的 UserName、聊天紀錄的備份檔案file descriptor
 - 對於每個新連線者，設為狀態 CLIENT_STATUS_CONNECTONLY
 - 登入成功（沒有密碼錯誤、雙重登入問題）者，設為狀態 CLIENT_STATUS_WAIT
 - 對於主動聊天、被邀請、離開雙人聊天室等狀況，做上面提到的三種狀態間的轉移
 - 每一則傳遞的訊息都會被寫入 Server 底下的資料夾 chatHistory，其中的對話紀錄檔
 - 明文紀錄
 - 檔名為「XXX_XXX.txt」，字母排序小的放前面
- Client
 - 狀態轉移與 Server 相同，因此在任何時刻兩邊「紀錄的狀態」都會相同
 - 有同時接受 STDIN 與 socket 的兩種接收方式，設計較複雜。
 - 值得一提的是傳送檔案的部分
 - 寄送方先確定檔案存在、位於當前資料夾
 - 寄送方開啟「子進程」，寄送封包 Protocol_sendfile
 - Server 傳送要求給接收方，接收方也開啟「子進程」
 - 接收方的「子進程」和 Server 建立 socket，建立完成後回傳ACK
 - Server 傳送回覆給寄送方，寄送方的「子進程」也與 Server 建立 socket
 - 接著透過兩邊的「子進程」，在不阻擋聊天室聊天的情況下背景傳送檔案
 - 接收方收到的檔案會建立在當前資料夾下
 - 將檔案分割傳送，一次4KB，有一個 fragflag 告知是否為最後一個檔案