

Final project report

Team 25

組員：資工三 B05902047 張庭逸

組員：資工三 B05902061 陳瑾瑋

Instructions on how to run server & clients

- Client端
 - 在project資料夾下，使用make client來產生執行所需檔案。
 - 使用make clean來移除生成的檔案。
 - 使用./client來執行client端。
 - 若要修改server的預設hostname和port，可以至client_data/config/server_info.config中修改。
 - 在執行client端時，可以改用./client [port]來指定連接的port。
 - 使用檔案傳輸功能下載的檔案會放在client_data/download中。
- Server端
 - 在project資料夾下，使用make server來產生執行所需檔案。
 - 使用make clean來移除生成的檔案。
 - 使用./server [port number]來執行server端並指定連結的port。
 - 使用者資料皆在server_data/[id]資料夾中，詳細架構後詳述之。

User & Operator Guide

- User Guide

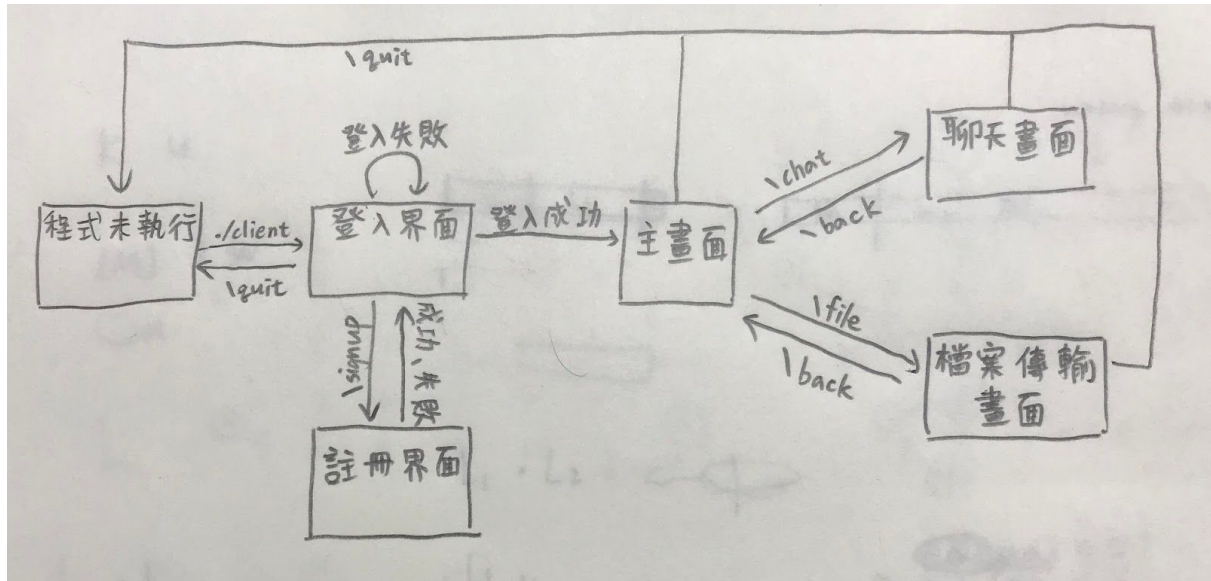
歡迎使用CNLine！若是新加入的朋友可以先輸入\signup指令來創立帳號，之後就可以用這組帳號登入進入主畫面了。CNLine目前有兩種功能：聊天以及檔案傳輸。輸入\chat [id]來和對方聊天吧！聊得盡興之後可以用\back回到主畫面，選擇其他朋友談天說地。主畫面中也可以輸入\file進入檔案傳輸系統：使用\send [id] [filename]來把珍藏的相片或各式各樣的檔案傳給對方吧！如果要接收檔案只要使用\download [filename]即可。別忘了，\back指令可以回到主畫面。太多指令記不住嗎？沒關係，只要記得使用\help即可立即查詢可用的指令。Team 25 希望您有個愉快的聊天體驗！
- Operator Guide

歡迎使用CNLine Server，只需簡單用./server [port number]執行server端就可以架設自己的server供朋友註冊，提供聊天和傳檔案功能囉！

在等待用戶與接受到任何用戶傳送的資料時皆有提示訊息可做參考。

System & Program Design

- 系統狀態介紹：



- 登入介面：使用者輸入帳號密碼的地方。
 - 註冊介面：註冊新帳號密碼的地方。
 - 主畫面：連接聊天系統和檔案傳輸系統的大廳。若是在線狀態可以於大廳收到聊天通知。
 - 聊天畫面：與另外一個帳戶聊天。每次進入會印出所有歷史訊息。
 - 檔案傳輸畫面：檔案的收發信箱。可以看到信箱中有什麼檔案並且下載，也可以指定其他人傳送檔案到他的信箱。支援對方離線時傳送。
- project資料夾架構
 - Makefile
 - src/
 - client_template/ (make時會複製一份並改名為client_data)
 - server_template/ (make時會複製一份並改名為server_data)
 - 其他source code
 - server_data/
 - [id]
 - [id].txt (存password, 未來可以放黑名單、好友名單、其他自定義資料)
 - 格式：
hash: [password]
 - message
 - [dest_id] (跟dest_id對話紀錄)
 - [dest_id2]
 - .
 - file
 - [id_file]
 - [id_file]
 - .
 - [id2]

- .
 - client_data/
 - config/
 - server_info.config (用於儲存server連線的初始設定)
 - download/ (擺放下載檔案的地方)
- Client 程式架構
 - 建構Config物件，此物件會讀檔、載入並儲存所有初始設定值，例如預設的server hostname和port等等。
 - 建構Connection物件，此物件會管理所有和連線有關的資料和方法，例如socket的file descriptor和包含自定義protocol的send、recv的方法等等。
 - 建構Page的物件，此物件會處理狀態的記憶和轉移，以及呼叫每個狀態該做什麼事的handler。
 - 開始執行迴圈直到Page物件的finish flag被設置為止。
 - 各個state共通指令：
 - 若收到\help指令則印出當前可用指令。
 - 若收到\quit指令則關閉程式。
 - 若state為登入界面：
 - 接收使用者輸入的帳號密碼，並傳給server來嘗試登入。登入成功則跳到主畫面，失敗則維持在登入界面。
 - 若收到使用者輸入註冊指令則跳至註冊畫面。
 - 若state為註冊界面：
 - 接收使用者輸入的新帳號密碼，並傳給server來嘗試註冊。成功與否皆會印出提示並轉回登入界面。
 - 若state為主畫面：
 - 將累積的通知印出，可以知道誰傳了訊息給你，形如：
[Tony] sent you a message.
 - 反覆查看server傳送來的訊息與使用者鍵入的指令。
 - 若server傳來聊天訊息，將通知立刻印出。
 - 若用戶要求聊天，進入聊天畫面。
 - 若用戶要求傳檔，進入檔案傳輸畫面。
 - 若state為聊天畫面：
 - 對server傳送聊天對象，server會寄回聊天歷史紀錄。
 - 反覆查看server傳送來的訊息與使用者鍵入的指令。
 - 若server傳來當前對象聊天訊息，將訊息立刻印出。
 - 若server傳來非當前對象聊天訊息，轉換成通知並存起來，等候用戶回到主畫面一併通知。
 - 若使用者輸入\back指令，回到主畫面。
 - 若使用者並非輸入指令，即聊天，則傳送給server聊天內容。
 - 若state為檔案傳輸畫面：
 - 通知server進入了檔案傳輸畫面，server會把信箱中的待收檔案檔名清單傳回。
 - 若用戶選擇下載某個檔名的檔案，server會將指定檔案傳來。
 - 若用戶指定對象和檔案，欲傳送檔案，檔案會傳上server並存在對象的信箱中等候對方下載。對方在離線狀態依然可以傳送檔案。

- Server程式架構
 - 建構socket，並把socket的fd加入rdfs，並開始迴圈select是否有新request。
當select到server有新連線時，加入新client。
當select到client送request來時，進入Recv_Mes處理。
在Recv_Mes裡，根據User log和client送來的op和data來處理request。
注意不論在任何階段若收到data為空，則代表client斷線，立刻回到迴圈並關閉client fd。
 - struct User_log：用來儲存已連接的client的狀態
 - fd
 - id
 - status
 - 存目前client在登入/註冊/聊天/檔案介面
 - 收到client的資料時，根據client所在的介面去處理
 - L: login, R: register, M: main(lobby), C: chat, F:file
 - dest_id
 - 若status為C，存正在聊天的client的id
 - 若status不為C，id留空
 - 若client在登入介面：
 - 接受client傳來的帳號密碼，嘗試登入，成功則轉至主畫面，若失敗則發送錯誤訊息。
 - 若收到註冊指令，將client轉至註冊介面。
 - 若client在註冊介面：
 - 接受client傳來的帳號密碼，嘗試註冊，若失敗則發送錯誤訊息。
 - 若client在主畫面：
 - 若收到聊天指令，若使用者存在，將client轉至聊天界面，並發送聊天記錄；若否，發送錯誤訊息。
 - 若收到檔案指令，將client轉至檔案傳輸畫面，並發送所有別的用戶傳送的檔案名。
 - 若client在聊天介面：
 - 接收client傳來的聊天訊息，寫進歷史紀錄，並確認訊息對象的狀態，若是上線中，則傳送訊息給訊息對象。
 - 若收到返回指令，返回主畫面。
 - 若client在檔案傳輸畫面：
 - 若收到下載指令，嘗試下載目標檔案，若檔案不存在則發送錯誤訊息，否則發送檔案內容。
 - 若收到傳送指令，嘗試下載檔案至傳檔對象的檔案儲存資料夾，若傳檔對象不存在則發送錯誤指令，否則在下載結束後發送傳送完畢指令。
 - 若收到返回指令，返回主畫面。
 - 處理完request後，回到迴圈等待下個request。

Protocol Specification

- SIGNUP流程
- client send(APP_SIGNUP, data_len, "I want to sign up.")

server send(APP_SIGNUP, data_len, "Sign up ID:")

client send(APP_SIGNUP, data_len, "[id]")

if ID exists

server send(APP_ERROR, data_len, "ID already exist.\nSign up ID:")

if ID does not exist

server send(APP_SIGNUP, data_len, "Please enter password.")

client send(APP_SIGNUP, data_len, "passpass")

if correct

server send(APP_SIGNUP, data_len, "Sign up success!")

- LOGIN流程

client send(APP_LOGIN, data_len, "[id]")

if ID exists

server send(APP_LOGIN, data_len, "Please enter password.")

if ID does not exist

server send(APP_ERROR, data_len, "User doesn't exist.")

client send(APP_LOGIN, data_len, "[password]")

if correct

server send(APP_MAIN, data_len, "Welcome.")

if wrong

server send(APP_ERROR, data_len, "Wrong Password, please login again.")

- CHAT流程

//if alan want to chat with tony

alan enter: \chat tony

alan: send(APP_CHAT, data_len, "tony")

if id:tony does not exist

server: send to alan(APP_ERROR, data_len, "user does not exist")

else

server: send to client1(APP_CHAT, data_len, "log [all_history_log]")

alan client: send(APP_CHAT, data_len, "[message]")

if tony is chatting with alan:

write into log, send to tony(APP_CHAT, data_len, "alan [message]").

if tony is online, but not chatting with alan:

write into log, send to tony(APP_CHAT, data_len, "alan [message]").

if tony is offline:

write into log.

alan enter \back

alan client: send(APP_MAIN, data_len, "\back")

server: send(APP_MAIN, data_len, "\back")

- FILE流程

//if tony already got file1 from tony, file2 from sam

client enter: \file

client: send(APP_FILE, data_len, "anyways")

server: send(APP_FILE, data_len, "Available: file tony_file1\nsam_file2\n...")

alan enter: \download tony_file1

alan client: send(APP_FILE, data_len "download")

server: send(APP_FILE, data_len "Which file?")

alan client: send(APP_FILE, data_len "tony_file1")

if file exists

server: send to alan(APP_FILE, data_len, binary_file_context)

if file does not exist

server: send to alan(APP_ERROR, data_len, "No such file.")

//if alan want to send file3 to tony

alan enter: \send tony file3

alan client: send(APP_FILE, data_len, "upload")

server: send(APP_FILE, data_len "To who?")

alan client: send(APP_FILE, data_len, "tony file3") //alan_filename

if correct

server: send(APP_FILE, data_len "Ready.")

if ID does not exist

server: send(APP_ERROR, data_len "User doesn't exist.")

//server store alan_file3 in tony's file directory

alan client: send(APP_FILE, data_len, binary_file_content)

server: send to alan(APP_FILE, data_len, "Done")

alan enter \back

alan client: send(APP_MAIN, data_len, "\back")

server: send(APP_MAIN, data_len, "\back")

Bounus

- 在線狀態時，若有其他使用者傳訊息給你，可以在主畫面中收到 [id] sent you a message 的通知。
- 支援離線傳檔，接收檔案方不在線仍能傳送檔案。
- 支援多次下載檔案。