

Computer Networks and Applications

Socket Programming 第二階段

B10703132 謝佳好

1. 執行環境：Ubuntu （使用 UTM 虛擬機）
2. 編譯 Server 端程式
 - (1) cd 至程式碼所在位置
 - (2) 使用 `g++ -o server server.cpp` 指令編譯程式碼。
3. 執行 Server 端程式
 - (1) 執行
 - i. cd 至程式碼所在位置，使用 `./server 8888` 指令執行編譯好的 Server 端程式。
 - (2) 資料結構
 - i. struct Client 儲存每次連線的資訊，包含 socket_fd、ip 和 port。
 - ii. class Account 儲存已註冊的帳戶資訊，包含 name、balance 和 client。
 - iii. 兩個 `vector<Account>` register_list 和 online_list 分別儲存所有註冊的帳戶和目前上線的帳戶。
 - (3) Multi-threaded 設計
 - i. 使用 multi-thread 和 worker pool 來處理多個 client 的請求。
 - ii. Main function 接收 client 連線後，將其存入共享的 queue，並透過條件變數喚醒等待中的 thread。Worker pool 中包含固定數量的 thread，負責從 queue 中提取任務並執行，處理 client 的註冊、登入、查詢上線清單、轉帳等功能。
 - iii. 執行緒透過 mutex 保護 queue 的存取，避免競爭條件。
 - (4) 處理 Client 的 request
 - i. REGISTER#<UserAccountName>：檢查該用戶名稱是否已註冊過。若註冊成功會傳送 100<space>OK<CRLF>，失敗會傳送 210<space>FAIL<CRLF> 訊息。
 - ii. <UserAccountName>#<portNum>：檢查該用戶名稱是否已註

冊過。若使用者有成功註冊，會回傳上線清單；若未註冊過會回傳 220<space>AUTH_FAIL<CRLF> 訊息。

- iii. List：回傳上線清單。
- iv. Exit：回傳 Bye<CRLF>，並關閉該 socket。
- v. <MyUserAccountName>#<payAmount>#<PayeeUserAccountName>：檢查 sender 跟 receiver 是否上線，並修改兩者的帳戶餘額。若成功會回傳 Transfer OK! <CRLF> 給 sender，失敗則會回傳 Transfer Fail! <CRLF>。

(5) Exception handling

- i. 在 socket 運作的過程中如果 api 的回傳值為負，都會印出相應階段的錯誤訊息。包含：socket(), connect(), send(), recv(), bind(), listen(), accept() 等。
- ii. Multi-thread 機制若無法成功創建 thread 等也會印出相對應的錯誤訊息。

4. 參考資料

- (1) <https://zake7749.github.io/2015/03/17/SocketProgramming/>
- (2) <https://github.com/davidleitw/socket>
- (3) <https://zake7749.github.io/2015/03/17/SocketProgramming/>