

Computer Networks and Applications

Socket Programming 第一階段

B10703132 謝佳好

1. 執行環境：Ubuntu （使用 UTM 虛擬機）
2. 編譯 Client 端程式
 - (1) cd 至程式碼所在位置
 - (2) 使用 `gcc -o client client.c` 指令編譯程式碼。
3. 執行 Client 端程式
 - (1) 執行
 - i. cd 至程式碼所在位置，先使用 `./server 8888 -a` 指令執行助教提供的 Server 端程式，再用 `./client 127.0.0.1 8888` 指令執行編譯好的 Client 端程式。
 - (2) 與 Server 溝通
 - i. 使用 `REGISTER#<UserAccountName>` 指令向 Server 註冊，若成功會印出 `100<space>OK<CRLF>`，失敗會印出 `210<space>FAIL<CRLF>` 訊息。
 - ii. 使用 `<UserAccountName>#<portNum>` 指令登入，若使用者有成功註冊，Server 會回傳 Client 端的上線清單；若未註冊過會回傳 `220<space>AUTH_FAIL<CRLF>` 訊息。
 - iii. 使用 `List` 指令向 Server 要求上線清單。
 - iv. 使用 `Exit` 指令主動告知 Server 端程式準備離線，同時結束程式。
 - (3) P2P
 - i. 假設已執行兩個 Client 端程式，且 `client1` 和 `client2` 皆已向 Server 註冊並登入。
 - ii. `client1` 透過
`<MyUserAccountName>#<payAmount>#<PayeeUserAccountName>` 指令轉帳給 `client2`。（或 `client2` 轉帳給 `client1`）
 - (4) 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. 實作 `client` 之間互相轉帳時需要 `create thread`，若過程中找不到目標 `client`、無法成功創建 `thread` 等，都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>