

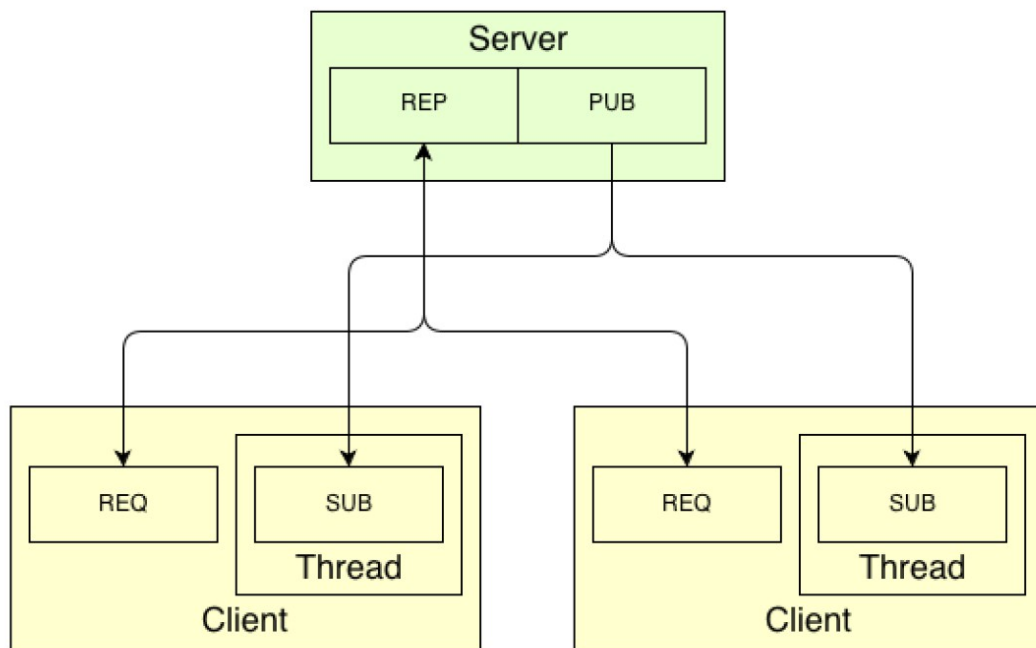
[2014 網路系統程式設計 Homework 9]

主旨

各位同學這幾個禮拜已經聽過幾位同學報告 ZMQ 的前面章節了，相信各位對於 ZMQ 已經有了初步的了解。接下來的幾個作業我們將會對於 ZMQ 做更進一步的練習，讓各位同學隨著撰寫程式碼的腳步，對 ZMQ 提供的各種 Pattern 有更進一步的了解。

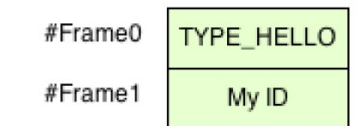
主題

利用 ZMQ 實作文字聊天室。
請參考以下程式架構，實作「群組聊天為主、指定對象聊天為輔」的文字聊天室。

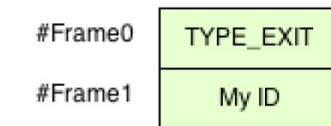


訊息傳送流程

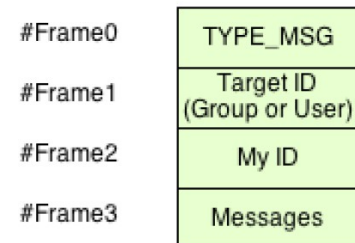
User 在 Client 輸入訊息將訊息交由 REQ 傳送給 Server 的 REP 收到訊息後進行判斷處理並用 REP 回傳處理結果；若為 NORMAL 訊息，請交由 PUB 將訊息發送給所有連接的 SUB 收到後顯示訊息。



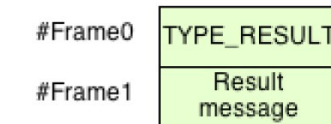
HELLO message format



EXIT Message format



NORMAL message format



RESULT message format

Client 端設計重點

Usage: ./chartroom <GroupID> <UserID>

- MSG Type:
 - “TYPE_HELLO”、“TYPE_EXIT”、“TYPE_MSG”、“TYPE_RESULT”。
 - 判斷完畢後，即可捨棄 MSG Type 資訊。
- 使用者執行 client 端時，需指定欲加入群組聊天室編號 (<GroupID>)，以及使用者自己的 ID (<UserID>)，未指定情況下，請提示並結束程式。
- 啟動 Client 時，首先發送 HELLO message 通知 Server。
- 結束 Client 時，發送 EXIT message 通知 Server。
- Client 只能收到所處 GroupID 的訊息，或是指定傳給自己 (UserID) 的訊息。
- 請針對 REP 回傳的訊息進行判斷 (Success or Fail)。
- 請實作 thread，使得 SUB 可以隨時接收訊息。主程序結束，呼叫 `zmq_term()` 時會讓所有被 `zmq_recv()` 或 `s_recv()` block 住的 ZMQ socket 回傳 -1 或 NULL，請以判斷此回傳值並結束 thread。
- 若使用者輸入以下特殊指令，Client 端需要執行特定動作：
 - @<TargetID> <Messages>：傳送私密訊息 <Messages> 給名為 <TargetID> 的人。
 - @exit：離開並關閉程式。
- UserID 不能與特殊指令相同 (Example: exit)。

Server 端設計重點

Usage: ./chatroomd

- Server 擁有一個 Linked list，動態儲存所有目前在線上的使用者 ID。
 - 收到 Hello 訊息時，將該 ID 加入 list 中，並回傳 Success。
 - 收到 Exit 訊息時，從 list 中移除，並回傳 Success。
- 若訊息發送對象為特定 ID (Example: John) 而非群組，請檢查該 ID 是否

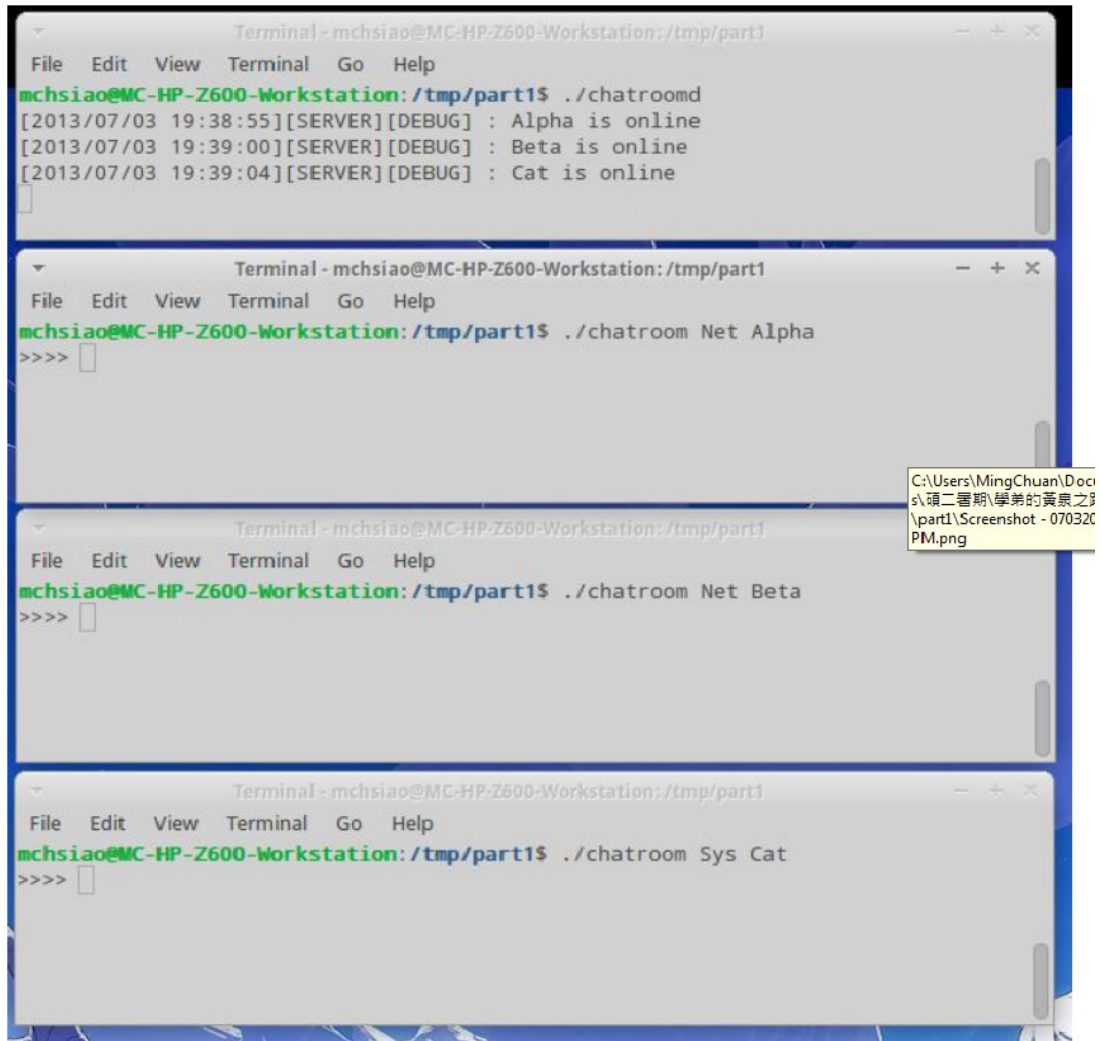
在線上，若在線上則回覆 Success；否則回覆 Fail。群組訊息一律回傳 Success。

參考結果

本案例共有 3 個 User。

Alpha，加入 Net 群組。Beta，加入 Net 群組。Cat，加入 Sys 群組。

1. 3 個 user 執行聊天室。



The image displays four terminal windows stacked vertically, showing the execution of a chatroom program. Each window has a title bar 'Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1' and a menu bar 'File Edit View Terminal Go Help'. The first window shows the output of './chatroomd', displaying three debug messages: '[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online', '[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online', and '[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online'. The second window shows the output of './chatroom Net Alpha', with the prompt '>>>>' followed by a cursor. The third window shows the output of './chatroom Net Beta', with the prompt '>>>>' followed by a cursor. The fourth window shows the output of './chatroom Sys Cat', with the prompt '>>>>' followed by a cursor. A yellow tooltip on the right side of the image contains the file path 'C:\Users\MingChuan\Documents\碩二暑期\學弟的黃晨之\part1\Screenshot - 07032013\PM.png'.

```
Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchshiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroomd
[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online
[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online
[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online
>>>>

Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchshiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroom Net Alpha
>>>>

Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchshiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroom Net Beta
>>>>

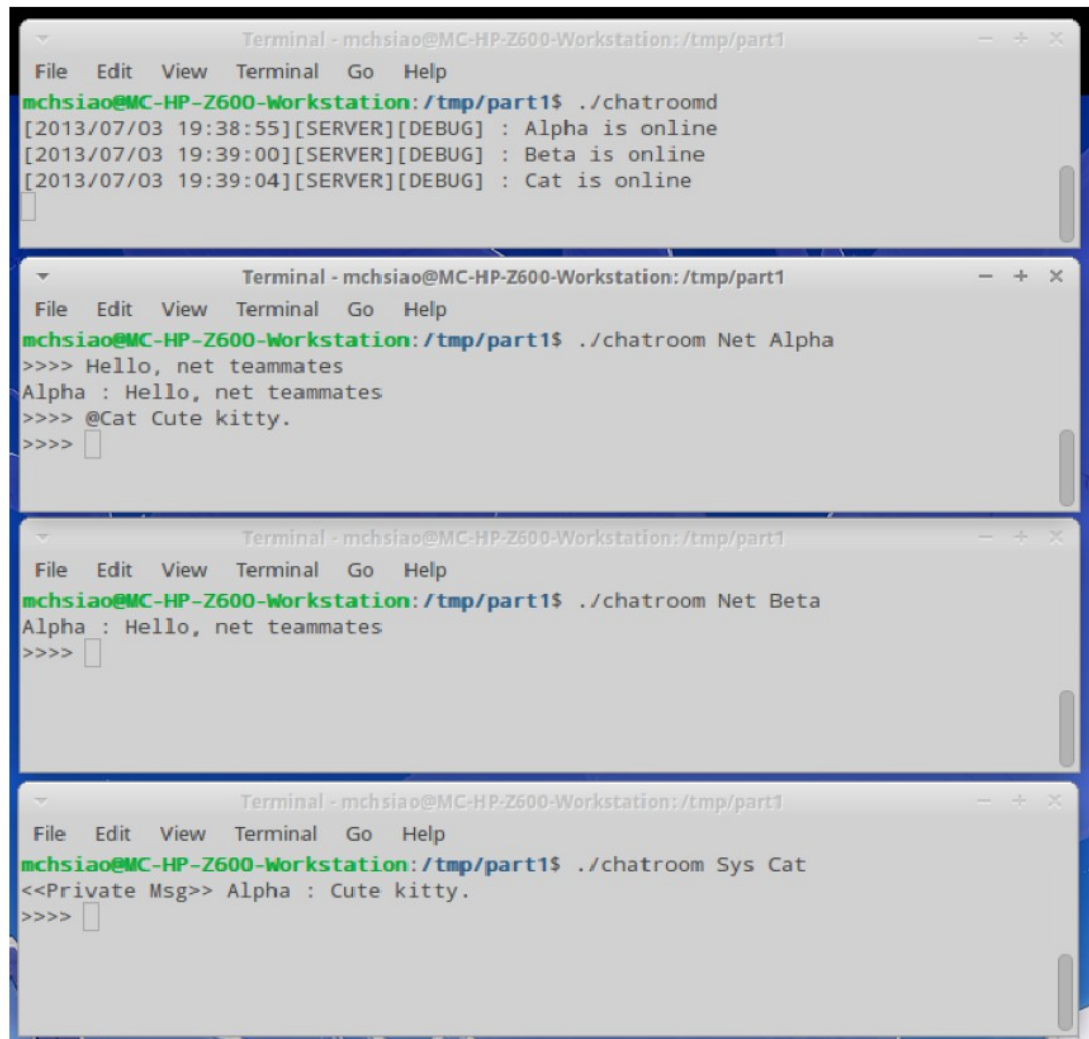
Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchshiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroom Sys Cat
>>>>
```

2. Alpha 發送訊息給 Net 群組。不同群組的會看不到訊息。

The image displays four terminal windows stacked vertically, all titled "Terminal - mchshiao@MC-HP-Z600-Workstation:/tmp/part1". Each window has a menu bar with "File", "Edit", "View", "Terminal", "Go", and "Help".

- Top window:** Shows the execution of `./chatroomd`. It outputs three lines of server debug messages: `[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online`, `[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online`, and `[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online`. A cursor is visible on the line following the last message.
- Second window:** Shows the execution of `./chatroom Net Alpha`. It outputs `>>>> Hello, net teammates` followed by `Alpha : Hello, net teammates`. A cursor is on the line after the second message.
- Third window:** Shows the execution of `./chatroom Net Beta`. It outputs `Alpha : Hello, net teammates`. A cursor is on the line after the message.
- Bottom window:** Shows the execution of `./chatroom Sys Cat`. It outputs a single line `>>>>` followed by a cursor.

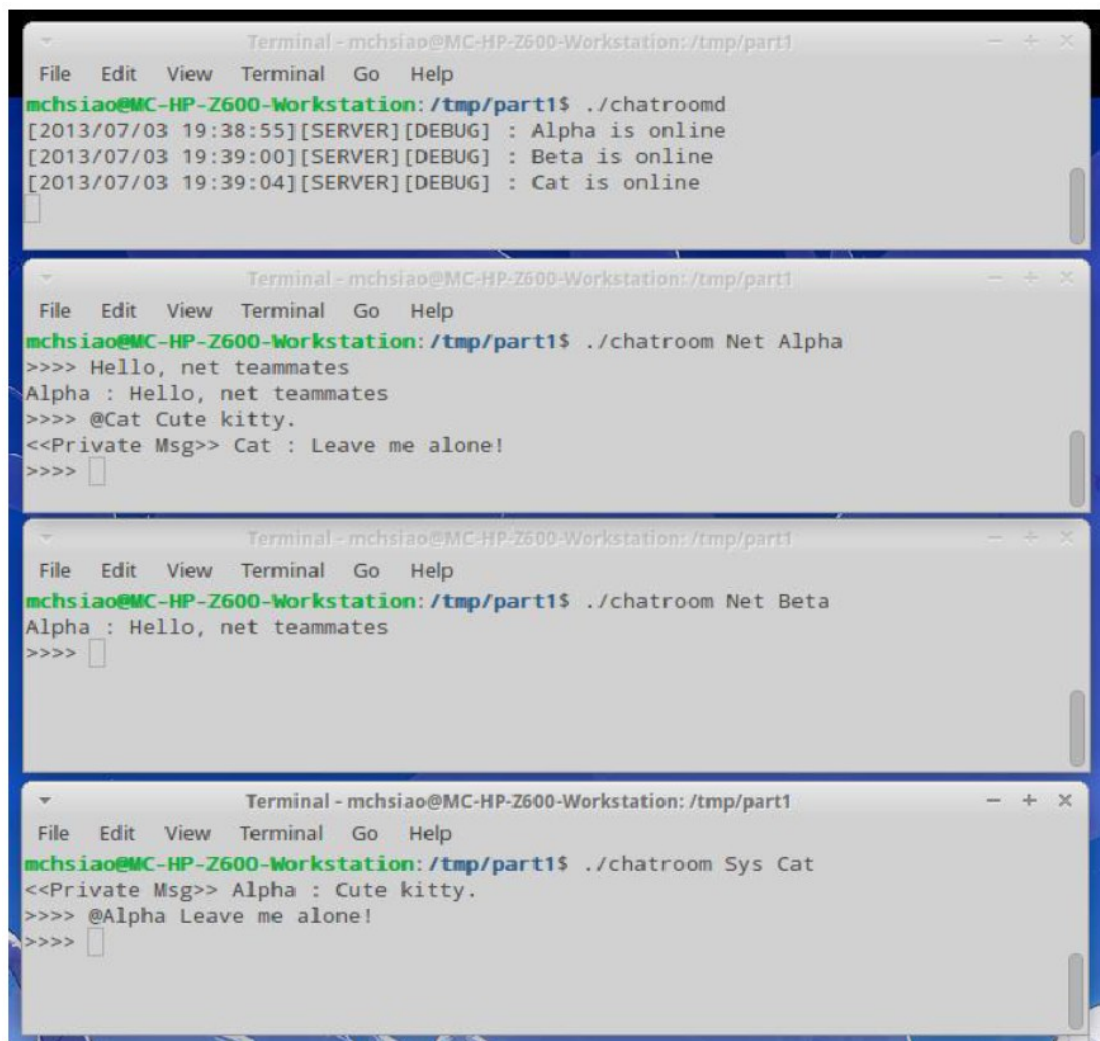
3. Alpha 傳送私密訊息給 Cat。



The image displays four terminal windows stacked vertically, all titled "Terminal - mchshiao@MC-HP-Z600-Workstation:/tmp/part1".

- Terminal 1:** Shows the execution of `./chatroomd`. It outputs three status messages: `[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online`, `[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online`, and `[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online`. A cursor is visible on the line following the last message.
- Terminal 2:** Shows the execution of `./chatroom Net Alpha`. It outputs `>>>> Hello, net teammates` and `Alpha : Hello, net teammates`. A second prompt `>>>>` is shown with the input `@Cat Cute kitty.` and a cursor on the next line.
- Terminal 3:** Shows the execution of `./chatroom Net Beta`. It outputs `Alpha : Hello, net teammates`. A prompt `>>>>` is shown with a cursor on the next line.
- Terminal 4:** Shows the execution of `./chatroom Sys Cat`. It outputs `<<Private Msg>> Alpha : Cute kitty.`. A prompt `>>>>` is shown with a cursor on the next line.

4. Cat 傳送私密訊息給 Alpha。



The image displays four terminal windows stacked vertically, showing the execution of a chatroom application. Each window has a title bar 'Terminal - mchshiao@MC-HP-Z600-Workstation: /tmp/part1' and a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Go', and 'Help'.

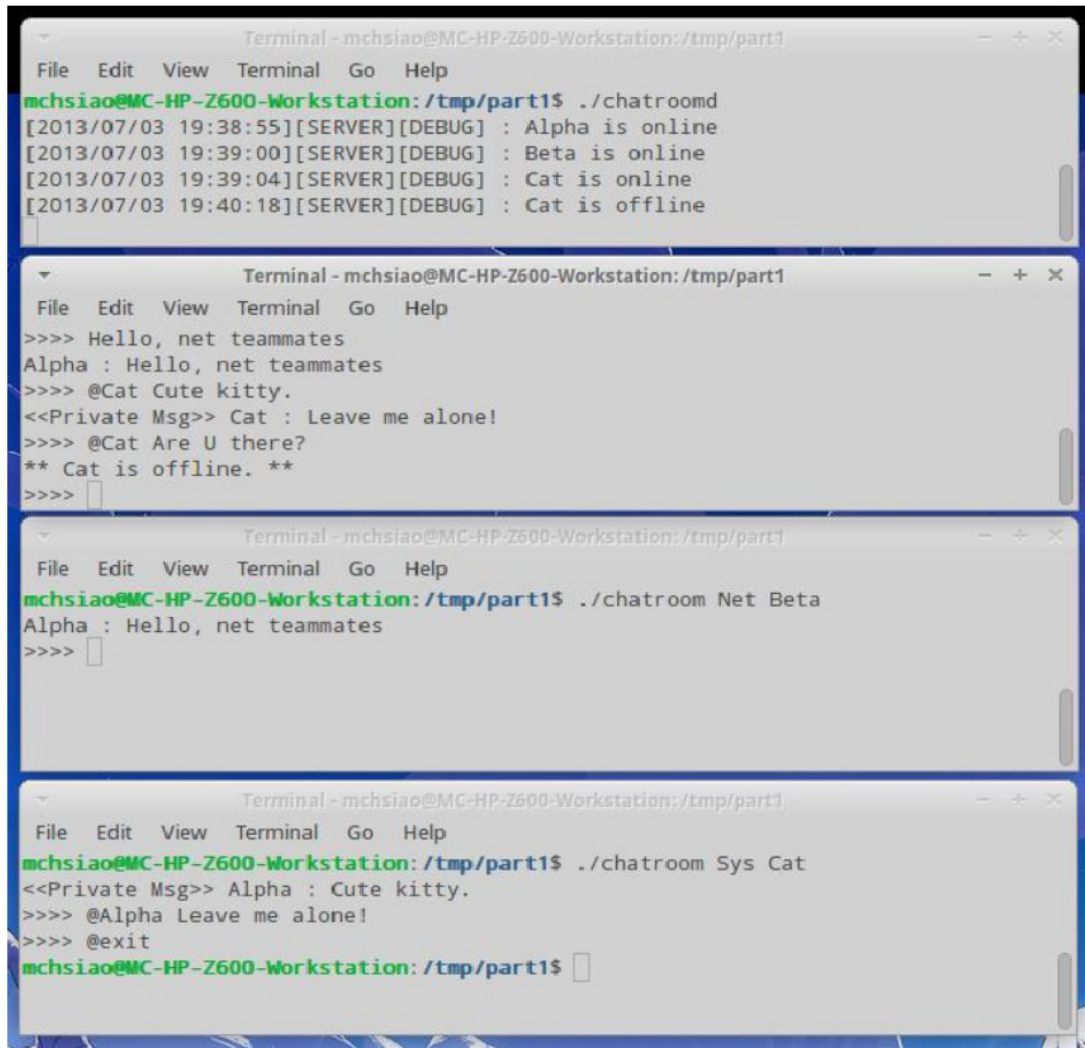
Terminal 1: Shows the startup of the chatroomd service. The prompt is `mchshiao@MC-HP-Z600-Workstation: /tmp/part1$./chatroomd`. The output consists of three lines of debug messages: `[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online`, `[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online`, and `[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online`.

Terminal 2: Shows a user joining the chatroom. The prompt is `mchshiao@MC-HP-Z600-Workstation: /tmp/part1$./chatroom Net Alpha`. The output shows a greeting: `>>>> Hello, net teammates`, followed by a message from Alpha: `Alpha : Hello, net teammates`, and a private message from Cat: `<<Private Msg>> Cat : Leave me alone!`.

Terminal 3: Shows a user joining the chatroom. The prompt is `mchshiao@MC-HP-Z600-Workstation: /tmp/part1$./chatroom Net Beta`. The output shows a greeting: `>>>> Hello, net teammates`.

Terminal 4: Shows a user joining the chatroom. The prompt is `mchshiao@MC-HP-Z600-Workstation: /tmp/part1$./chatroom Sys Cat`. The output shows a private message from Alpha: `<<Private Msg>> Alpha : Cute kitty.`, followed by a message from Alpha: `>>>> @Alpha Leave me alone!`.

5. Cat 離開聊天室，此時 Alpha 發送私密訊息給 Cat 時，Server 提示 Cat 不在線上。



```
Terminal - mchsiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchsiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroomd
[2013/07/03 19:38:55][SERVER][DEBUG] : Alpha is online
[2013/07/03 19:39:00][SERVER][DEBUG] : Beta is online
[2013/07/03 19:39:04][SERVER][DEBUG] : Cat is online
[2013/07/03 19:40:18][SERVER][DEBUG] : Cat is offline

Terminal - mchsiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
>>>> Hello, net teammates
Alpha : Hello, net teammates
>>>> @Cat Cute kitty.
<<Private Msg>> Cat : Leave me alone!
>>>> @Cat Are U there?
** Cat is offline. **
>>>>

Terminal - mchsiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchsiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroom Net Beta
Alpha : Hello, net teammates
>>>>

Terminal - mchsiao@MC-HP-Z600-Workstation: /tmp/part1
File Edit View Terminal Go Help
mchsiao@MC-HP-Z600-Workstation: /tmp/part1$ ./chatroom Sys Cat
<<Private Msg>> Alpha : Cute kitty.
>>>> @Alpha Leave me alone!
>>>> @exit
mchsiao@MC-HP-Z600-Workstation: /tmp/part1$
```

其他注意事項

1. gcc compile 時請記得加上 `-lpthread -lzmq` 參數。
2. 會檢查各個 ZMQ Socket 收到的封包內容，尤其是 SUB 收到的訊息。
目的是讓各位了解如何在 SUB 上設定 `filter`。
3. 可參考並使用 Code Connected Volume 1(直接 google) 內的 Example code。
4. 非常推薦使用 zhelpers.h 內的函式，簡化字串傳輸處理。
5. **逾期繳交以零分計算，不接受補交**，有問題請事先告知，Demo 時間會另

行通知

6. 對題目有問題可以寄信問助教(a66766676@gmail.com)或是到實驗室(F5018)直接詢問，但不幫忙 debug。

限制

1. 請在 **Ubuntu 14.04** 上使用 C 語言進行開發，並使用 ZMQ v3.2.0 版。
2. 請附上可以編譯你的程式的 **Makefile**，未寫扣分。
3. **嚴禁抄襲**其他同學作業, **參與者(抄襲與被抄襲)[學期總成績]**均以**零分**計算。

作業上傳

1. 請壓縮成 **zip** 壓縮檔,並上傳至 **中山網路大學**,作業命名規則為
“**學號_SP_HW9**”。
Example: M013040001_SP_HW9.zip
2. 作業截止時間為 2014/12/30 (Tue.) 23:59,請在時間內上傳作業。