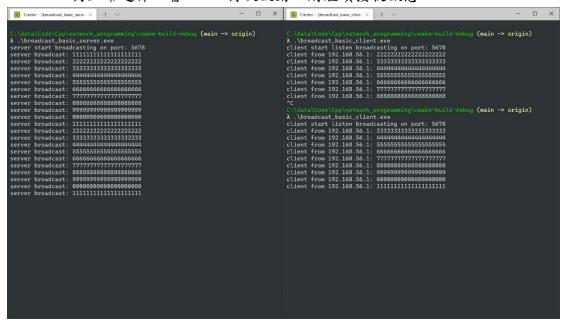
# 網路程式設計 HW-廣播網路程式作業

# 一、 執行畫面

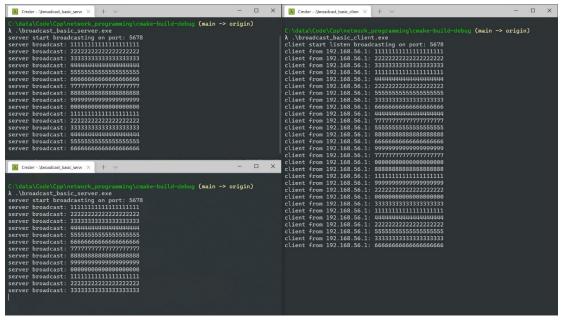
#### 練習1:基本廣播示範(client, server)

此程式 server 每秒傳送一則訊息給 client, client 隨時接收 server 傳送的訊息。

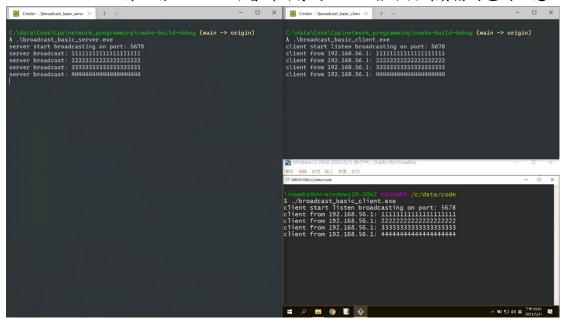
[一對一] 左方為 server 端的執行畫面,右方為 client 端的執行畫面,啟動左方的 server 後,再啟動右方的 client,當 client 關閉後, server 仍正常運作,當 client 再次啟動,則繼續接收訊息。



[多對一] 左上方為第 1 個 server 端的執行畫面,左下方為第 2 個 server 端的執行畫面,右方為 client 端的執行畫面,先啟動 client,再啟動左上方的 server,3 秒後後再啟動左下方的 server,可看到 client 可收到所有 server 的廣播封包。



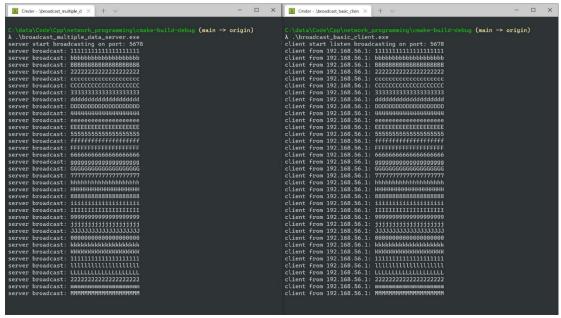
[一對多] 左方一樣為 server 端的執行畫面,右上方為本機 client 端的執行畫面,右下方為其他電腦 client 端的執行畫面,先啟動所有的 client,再啟動 server,可看到所有的 client 皆可收到廣播封包的訊息。



### 練習2:Server 同時傳送多種資料

此程式 server 每秒傳送三則訊息給 client, client 隨時接收 server 傳送的訊息。

左方為 server 端的執行畫面,右方為 client 端的執行畫面,啟動右方的 client 後,再啟動左方的 server,可看到 client 端可接收到訊息。

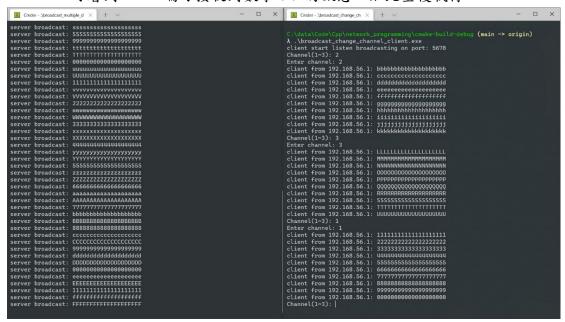


#### 練習3:Client 過濾接收的資料(模仿選台)

此程式 server 每秒傳送三則訊息給 client, client 可選擇要進入的頻道,並隨時接收 server 傳送的訊息,根據頻道編號顯示出對應的訊息。

左方為 server 端的執行畫面,右方為 client 端的執行畫面,啟動右方的 client 後,再啟動左方的 server。

首先,先在 client 端選擇第 2 個頻道,可看到 client 端可接收到小寫字母 a-z 的訊息。接下來,在 client 端選擇第 3 個頻道,可看到 client 端可接收到大寫字母 A-Z 的訊息。最後,在 client 端選擇第 1 個頻道,可看到 client 端可接收到數字 0-9 的訊息,如此重複執行。

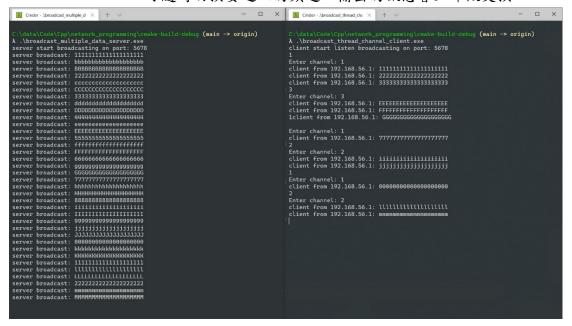


#### 練習 4: Client 使用 thread 作出選台效果

此程式 server 每秒傳送三則訊息給 client, client 可選擇要進入的頻道,並隨時接收 server 傳送的訊息。

左方為 server 端的執行畫面,右方為 client 端的執行畫面,啟動右方的 client 後,再啟動左方的 server。

client 可隨時切換要進入的頻道,輸出的訊息會立即做更換。

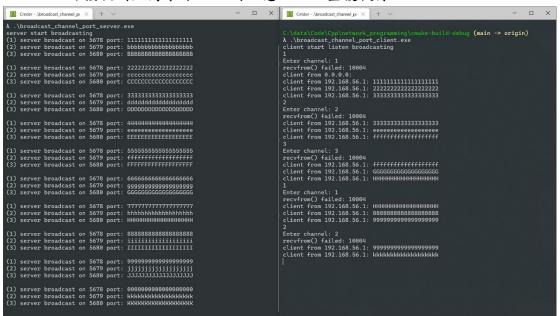


#### 練習 5: 用不同 port 實現頻道與選台(client, server)

程式 server 每秒從每個 port 傳送一則訊息給 client, client 可選擇要進入的頻道,並隨時接收 server 傳送的訊息。

左方為 server 端的執行畫面,右方為 client 端的執行畫面,啟動右方的 client 後,再啟動左方的 server。

client 可隨時切換要進入的頻道,輸出的訊息會立即做更換。首先, 先在 client 端選擇第 1 個頻道,可看到 client 端可接收到數字 0-9 的訊 息。接下來,在 client 端選擇第 2 個頻道,可看到 client 端可接收到小 寫字母 a-z 的訊息。最後,在 client 端選擇第 3 個頻道,可看到 client 端 可接收到大寫字母 A-Z 的訊息,如此重複執行。



## 二、 心得

在實作這次作業的過程中,我遇到了以下幾個問題。

第一個是我發現老師範例程式碼有用到 memset 函式的地方可能會有些問題,因為程式最上方 define 的傳送最大值 MAXLINE 定義為 20,可是memset 函式帶入要修改的長度為 MAXLINE,也就是 20,這樣子一來會導致一個問題,就是這個 char array 沒有地方存放結尾字元\0,所以會導致輸出時的句子出現了一些奇怪的情況,像是多輸出了方塊□的豆腐字元,或是o 之類的字元等,只要將 sizeof(str)改為 sizeof(str) – 1 少輸出一個字元即可解決此問題。

可參考 stack overflow 的這篇討論串:

https://stackoverflow.com/questions/62359687/does-memset-change-the-array-length-when-another-array-is-memset

另外一個是第五題當 client 端在切換頻道時,關閉 socket 再連線 socket 的期間,需要花一些時間,會導致暫時性的發生 recvfrom 函式無法正常運作的情況,如上方第五題右邊的截圖,我想到的解決方式是可以設定一個 flag 來判斷 socket 是否連接,在 socket 要關閉前,先將 flag 設為 false,等待連線成功,再將 flag 設為 true,若要執行 recvfrom 函式之前,需先判斷 flag 是否為 true 的狀態。

最後另外一個小建議是第五題老師 client 端的程式碼利用 switch case 判斷條件時,僅須把 port 的設定放在裡面即可,下面的 bind 可放在 switch case 之後,而在 switch case 中加入 default 的條件,當輸入要切換的頻道不符合規範時,可以就不改動 port 的設定,或是直接結束程式,這樣子可以讓程式碼更簡潔。

備註: 以上所有的程式碼皆放置在我的 GitHub repo, 詳見以下連結網址 https://github.com/linwebs/network programming#week15