網路程式設計 HW-非攔阻模式作業

一、 作業說明

此次作業採用 TCP 協定非攔阻模式來進行傳輸,可避免無法接收到資料時,程式會被攔住的情況。

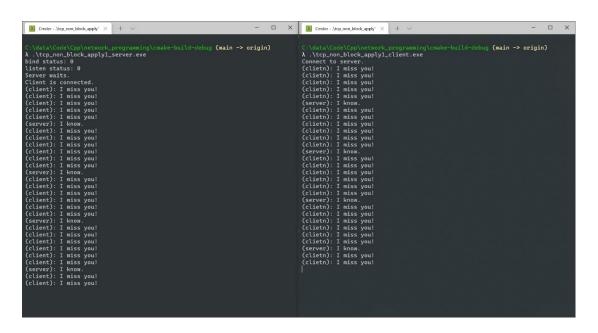
二、 執行畫面

應用1:聊天室第1版解除一人一句限制

左方視窗為 server 端的畫面,右方視窗為 client 端的畫面。

server 端接收 client 端傳來的訊息,並每5秒回傳一次 I know.。

client 端每 1 秒傳送 1 次 I miss you!到 server, 並接收 server 回傳的訊息。



應用 2: 聊天室第 2 版解除一人一句限制

左方視窗為 server 端的畫面,中間視窗為第1個 client 端的畫面, 右方視窗為第2個 client 端的畫面。

第 1 個 client 端名稱為 111,每 1 秒傳送 1 次 111 到 server, 並接收 server 回傳的訊息。

第 2 個 client 端名稱為 222, 每 2 秒傳送 1 次 222 到 server, 並接 收 server 回傳的訊息。

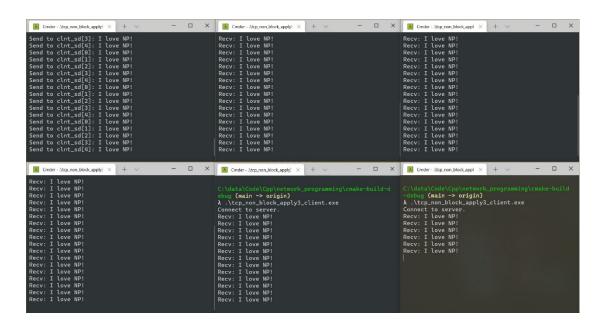


應用 3: server 可服務多個 client, 且 client 可陸續連入

左上方視窗為 server 端的畫面。 中間上方視窗為第 1 個 client 端的畫面。 右上方視窗為第 2 個 client 端的畫面。 左下方視窗為第 3 個 client 端的畫面。 中間下方視窗為第 4 個 client 端的畫面。 右下方視窗為第 5 個 client 端的畫面。

server 端隨時接收所有 client 端所傳送的訊息,並且隨時接受 client 端得連入。

所有的 client 端每 1 秒傳送 1 次 I love NP!到 server 端。



三、 心得

這次作業相較前一次來說,整體的量較為適宜,可以順利在課堂中完成,不過在製作的過程中發現 accept 若不加上 while 迴圈,將其重複執行,程式有機率無法正常執行,老師的範例程式沒有加入,但是我實際在撰寫時沒有加入會一直在等待 accept 的情況,在跟老師討論後,理論上 accept 做一次即可,我猜測有可能是被系統其他服務攔截住,導致無法正常執行,但多嘗試幾次卻可以正常執行。