

Socket programming 的實作與實驗設計

專題編號：103-CSIE-S004

執行期限：102 年第 1 學期至 103 年第 1 學期

指導教授：吳和庭

專題參與人員：100590328 魏齊呈

100590329 顏澤安

一、摘要

本專題是利用實作 Socket 程式，藉此了解伺服端 Socket 和客戶端 Socket 的建立到關閉流程，再透過使用微軟提供的 WinSock API 搭配 C++ 程式語言設計出實驗。

我們總共會實作三項實驗，第一項實驗為猜數字的互動小遊戲，第二項實驗是透過 Server 執行的兩人文字聊天室程式，最後一項實驗是可用來連結 Gmail、Hotmail、Yahoo SMTP Server 寄發電子郵件的客戶端程式，藉由以上三項實驗希望能讓大家擁有對網路通訊擁有基礎的概念。

關鍵詞：Socket Programming、WinSock、網路通訊。

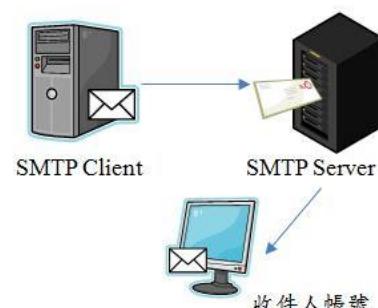
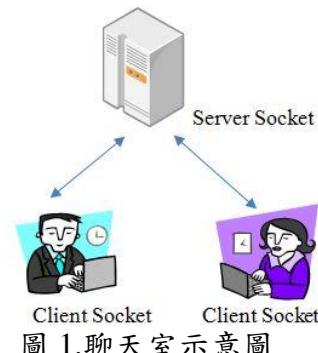
二、緣由與目的

在接觸計算機網路這門課後，我們都知道網路通訊最主要分為伺服端跟用戶端這兩端可以互相收送資料，但是我們卻不知道在收送資料之前必須先經過哪些步驟才能正確地將資料送給另一方，而 socket 就在這過程中扮演重要的角色。

Socket 讓我們不用熟悉下層網路協定運作的細節就能完成網路通訊程式，而 socket 本身如何運作也是我們很想知道的，所以我們就試著先去完成簡單的網路程式並且確實了解通訊過程中的資料傳遞完整流程，最終的目的就是要讓

學習計算機網路的人不是只有在紙本上理解而已，透過完成實驗並撰寫程式的過程中，我們能深入了解資料傳輸的過程之中運用了哪些協定、使用了哪些連接埠等重要的資訊，並且認識 Winsock 的相關函式，這也是為什麼要從實驗學習的最主要原因。

三、實驗示意圖



四、系統運作流程

在 Socket 程式運作時主要是採用主從式架構(Client-server model)來執行的，而伺服端與伺服端 Socket 要連線前必須先經過多項步驟才能讓雙方互相傳送訊息、資料。

伺服端 Socket 在通訊前是最先被開啟並接著設定之後資料傳輸所用的協定、連結埠號，在這步驟被稱為 Bind，執行完 bind 步驟後伺服端便會開始執行下一個步驟 Listen 來等待用戶端的 connet 指令，一旦雙方的傳輸協定、連結埠號、IP 位置互相符合後便可以進行資料的傳輸，資料傳輸完畢後便 close socket 完成這次的通訊。

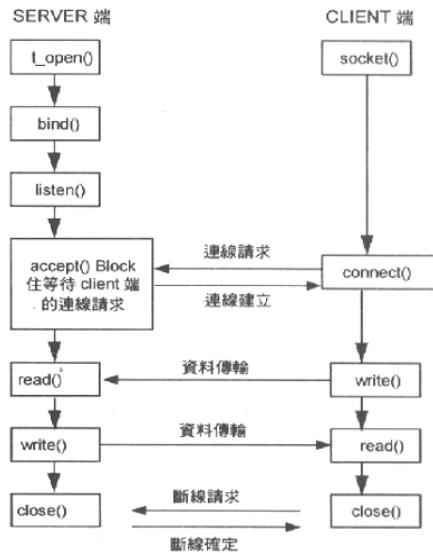


圖 3. Socket 資料傳輸流程圖

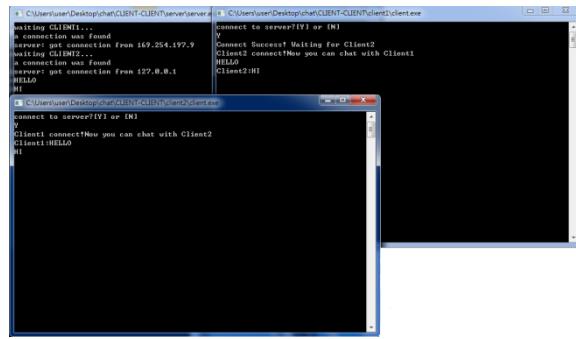
五、實驗功能說明

(一). 猜數字遊戲小實驗

這實驗主要的目的是透過猜數字這個基本的功能讓第一次學習 socket programming 的同學能學習 socket 跟 socket 之間的互動，並了解基礎的 Winsock API 函式功能，在之後的實驗中才能更快速地完成比較複雜的 socket programming 實驗。

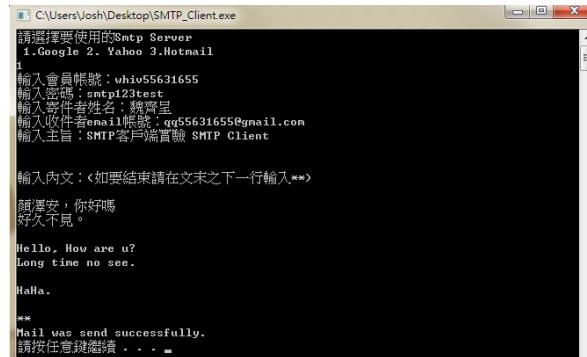
(二). 簡易文字聊天室

這實驗主要的目的是可以讓大家了解主從式架構的運作模式與 socket 設定。一開始透過伺服端 socket 程式的建立，指定 port number、IP address 後，讓客戶端 socket 程式輸入資訊後便可以開始文字聊天的功能，最多可以擁有兩位用戶進行互動聊天跟一個固定的伺服端進行傳送資料跟建立連線的功能。



(三). SMTP Client

這實驗的目的首先是要讓大家了解電子郵件傳輸協議(SMTP)並認識 Server 溝通的基本指令，如：HELO、RCPT TO。在進行完上述的基礎練習後，最後的目標是要完成輸入自己的電子郵件會員帳號密碼傳送信件，不過 Gmail 這些常見的會員信箱提供的 SMTP server 都是需要經過 TLS/SSL 認證才能寄信，所以我們用到了”OpenSSL”這套開放的函式庫來進行會員驗證的功能，驗證完成就能正常地使用寄信的功能。在程式中可以自由輸入主旨、內文、寄件者姓名來完成整封電子郵件。下圖為程式執行過程



SMTP客戶端實驗 SMTP Client

魏齊呈
寄給我
魏齊呈，你好嗎
好久不見。

Hello, How are u?
Long time no see.

HaHa.

參考文獻

[1] Winsock Reference

[http://msdn.microsoft.com/en-us/library/windows/desktop/ms741416\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741416(v=vs.85).aspx)

[2] OpenSSL

<https://www.openssl.org/>