**電腦網路實驗實驗報告 < Socket Programming >**

**姓名: 翁佳煌　　　學號: 409430030**

# 實驗名稱

網路程式設計，寫一個會標記「已讀」訊息的echo server，並寫一支client程式跟echo server對話。

# 實驗目的

這次實驗將讓我們理解通訊協議、網路設備、網路架構中的client/server模型以及 TCP socket programming的基本概念，讓我們對網路程式設計有整體的認識，並通過 getaddrinfo 和 TCP client/server 的示範演示來演示如何實現這些技術。通過本實驗的學習，將學會如何建立基於 TCP 協議的客戶端/伺服器應用程式，並掌握如何使用網路編程的基本工具和API。

# 實驗設備

Linux作業系統之電腦。

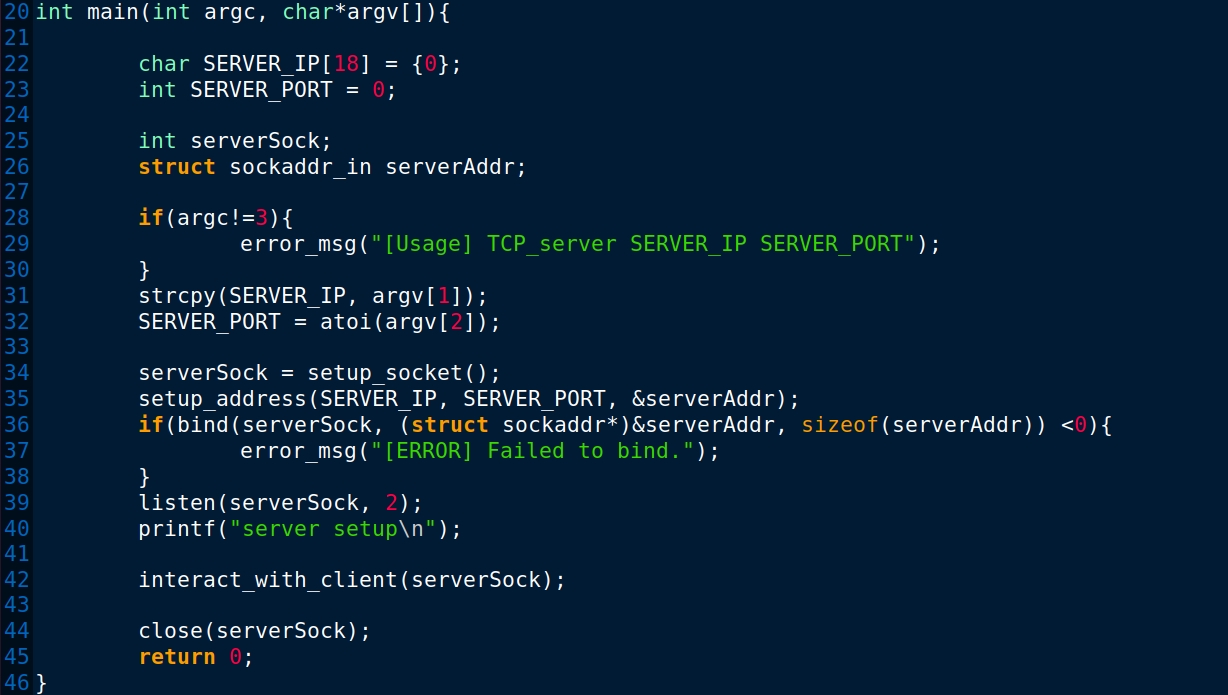
# 實驗步驟

首先看到**server端**，它會持續等待客戶端的連線，並接收、處理客戶端的訊息並回覆已讀給client，以下是詳細code介紹:



1~10行: 引用需要用到的標頭檔，除了c語言常見的標頭以外，還包含sys/socket.h、netinet/in.h、arpa/inet.h、unistd.h等等。

12~18行: 宣告函式，分別為error\_msg、setup\_socket、setup\_address、interact\_with\_client。

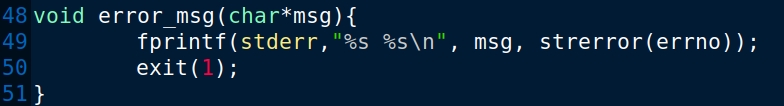


接下來看到main函式。

22~26行:定義了一些變數，包括了 SERVER\_IP、SERVER\_PORT、serverSock 和 serverAddr。

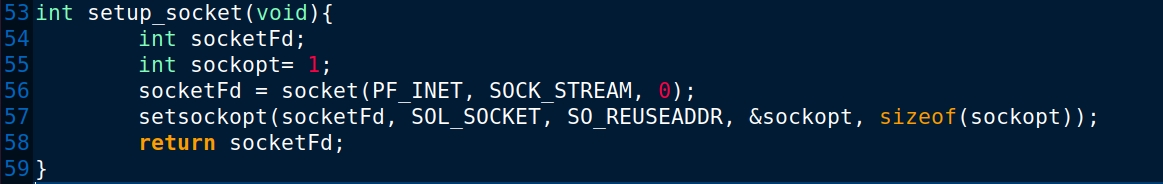
28~32行:用來判斷命令行參數是否正確，如果命令行參數不是3個，則調用 error\_msg 函數輸出錯誤信息並退出程序。否則，將 SERVER\_IP 和 SERVER\_PORT 的值賦值為命令行參數中指定的值。

34~46行:呼叫setup\_socket函數來建立一個socket，並且將描述符存放到serverSock中。呼叫setup\_address函數來設置serverAddr結構體的IP地址和端口號。使用bind函數將socket描述符和IP地址與端口號綁定起來，如果失敗則執行error\_msg函數。接著使用listen函數開始監聽伺服器端的socket描述符，最多可以接受2個客戶端連線請求。再來呼叫interact\_with\_client函數來開始與客戶端進行互動，最後的close(serverSock)的功能為關閉伺服器端的socket。



48~51行:

出現錯誤時輸出錯誤訊息到標準錯誤輸出(stderr)，然後直接結束程式執行(exit(1))。



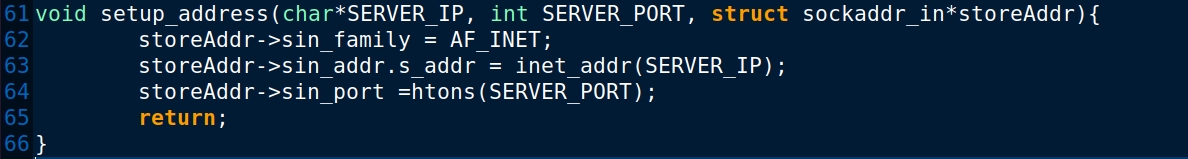
再來看到setup\_socket函數。

53~59行:

函數內部先宣告了一個 socketFd 變數，用來存放 socket 的檔案描述符。接著，宣告一個 sockopt 變數，並將它初始化為 1。

接著，使用 socket()函數建立一個 TCP socket，並將它的檔案描述符存放在 socketFd 變數中。socket 函數的第一個參數指定 socket 的 domain，這裡使用了 PF\_INET，表示使用 IPv4 協議；第二個參數指定 socket 的類型，這裡使用了 SOCK\_STREAM，表示使用 TCP 協議；第三個參數指定協議，這裡使用了 0，表示使用預設協議。

接著，使用 setsockopt()函數設定 socket 的屬性。setsockopt 函數的第一個參數指定要設定的 socket；第二個參數指定要設定的屬性，這裡使用 SOL\_SOCKET，表示設定的是 socket 級別的屬性；第三個參數設定了 SO\_REUSEADDR ，使得當 socket 關閉時，端口可以立刻被重用。第四個參數是sockopt的位址；第五個參數指定 sockopt 的大小。



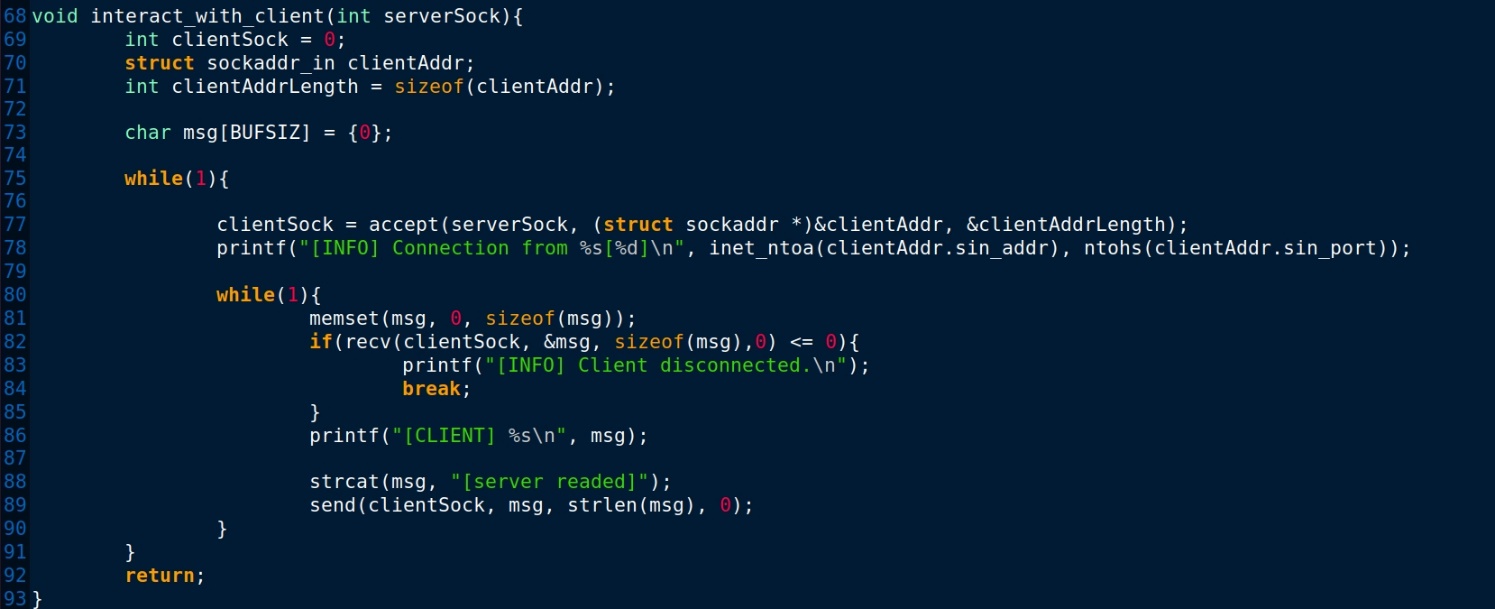
接下來看到setup\_address函數。

61~66行:

storeAddr->sin\_family = AF\_INET; 將伺服器地址的協定設為 IPv4。

storeAddr->sin\_addr.s\_addr = inet\_addr(SERVER\_IP); 用 inet\_addr() 函數將伺服器的 IP 位址轉換為網絡字節序（network byte order），然後將其儲存到伺服器地址的 sin\_addr 欄位中。

storeAddr->sin\_port =htons(SERVER\_PORT); 用 htons() 函數將伺服器的連接埠號碼轉換為網絡字節序，然後將其儲存到伺服器地址的 sin\_port 欄位中。



最後看到interact\_with\_client，負責接收client端的訊息以及回復。

68~93行:

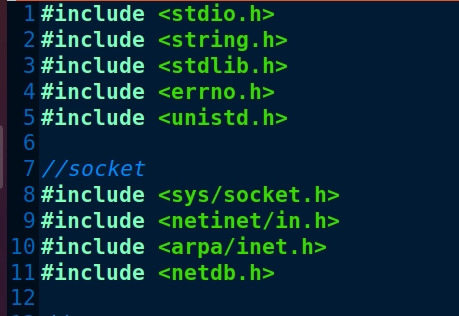
一開始宣告了一個變數 clientSock 並初始化為 0，用來接收與客戶端的連接。

宣告了一個變數 msg 來接收從客戶端傳來的訊息。在 while 迴圈中，使用 accept 函數來接受來自客戶端的連接，並將客戶端的 IP 位址和連接埠號印出。

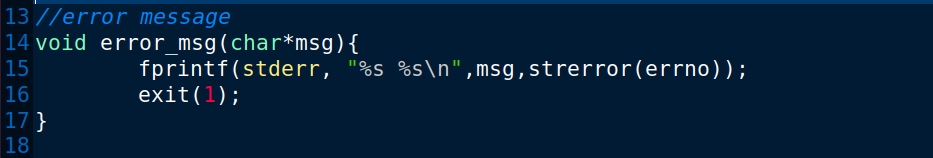
接著使用一個內層 while 迴圈來持續接收客戶端傳來的訊息。使用 memset 函數將 msg 陣列中的元素全部初始化為 0，以確保接收到的訊息不會錯誤。接著使用 recv 函數來接收來自客戶端的訊息，並判斷接收到的長度是否小於等於 0，如果是則認為客戶端已經斷線，印出相關訊息後跳出內層 while 迴圈。否則，印出客戶端傳來的訊息。將固定的回應訊息串接在 msg 陣列後面，使用 send 函數將回應訊息傳回給客戶端。在內層 while 迴圈結束後，外層 while 迴圈會繼續等待下一個客戶端的連接。

---------------------------------------------------------------------

接下來是介紹**client端**，它需要輸入訊息並傳送到server端後，要接收到server端送來的已讀訊息，以下是詳細code介紹:

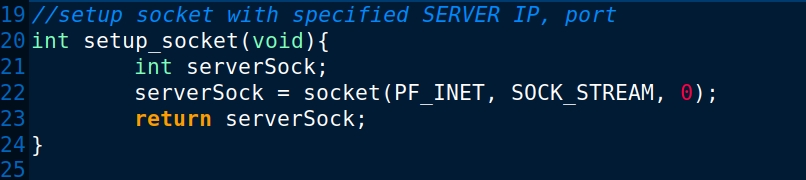


首先1~12行和上方server介紹的一樣include相關的標頭檔。



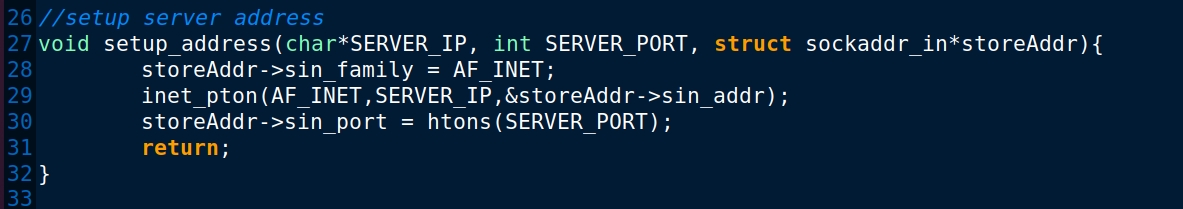
再來第13~17行:

出現錯誤時輸出錯誤訊息到標準錯誤輸出(stderr)，然後直接結束程式執行(exit(1))。



19~25行:

它使用了 socket() 函數來創建 serverSocket。socket() 函數的第一個參數是要使用的地址族（AF\_INET 表示 IPv4 地址族），第二個參數是要使用的 socket 類型（SOCK\_STREAM 表示 TCP 協議），第三個參數是要使用的協議（0 表示自動選擇）。



26~32行:

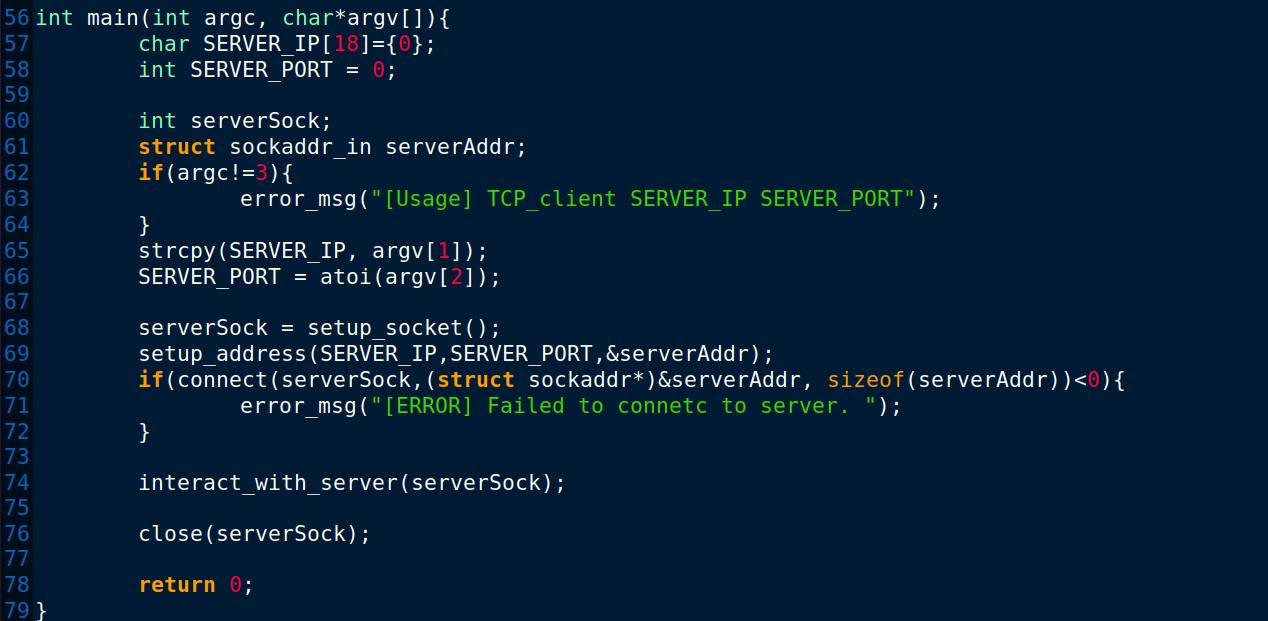
這個函數用來設置指定 SERVER\_IP 和 SERVER\_PORT 的伺服器地址。它會將伺服器地址存儲到傳遞的指針 storeAddr 所指向的 sockaddr\_in 結構中。

在函數中，sin\_family 屬性被設置為 AF\_INET，表示使用 IPv4 網絡協議。sin\_addr 屬性使用 inet\_pton() 函數將 SERVER\_IP 轉換成網絡字節順序，並將其存儲到結構體中。sin\_port 屬性使用 htons() 函數將 SERVER\_PORT 轉換成網絡字節順序，並將其存儲到結構體中。



34~54行:

這個函式是用來與伺服器進行互動的。它使用了一個無限迴圈，不斷讀取使用者從終端輸入的訊息，然後將該訊息傳送到伺服器。接著它使用 recv 函式來等待從伺服器回傳的訊息，一旦收到，就會在終端輸出 "[SERVER] " 和接收到的訊息。如果 recv 函式返回的是小於或等於零的值，就代表與伺服器的連接已經斷開了，此時該函式會跳出無限迴圈並結束。



最後是主函式:

使用者必須提供要連線的伺服器的 IP 位址和埠號。接下來，主程式會使用 setup\_socket() 函式建立一個 TCP socket，並使用 setup\_address() 函式設定要連線的伺服器的位址。最後，主程式使用 connect() 函式建立與伺服器的 TCP 連線，並進入 interact\_with\_server() 函式與伺服器進行資料交換。

在主程式中，如果命令列參數的數量不是 3 個，則會顯示一個使用說明並結束程式。如果連線失敗，主程式會顯示錯誤訊息並結束程式。如果正常建立連線，主程式會進入資料交換的迴圈。

# 問題與討論

1. 有許多socket的函數仍然需要再更進一步理解，包括其用途以及需要的參數。
2. 如果改使用ipv6有許多地方需要更動，例如將 PF\_INET 改成 PF\_INET6，以支援 IPv6等等。這問題在下禮拜的實驗課將會實作。
3. Socket Programming 在 Linux 和 Windows 上的差異，包括Socket 創建方式不同、IP 位址與埠號表示方式不同、Socket 的清除方式不同等等。

# 心得與感想

這次的實驗讓我更深入地了解了Socket Programming的概念，也學習到如何使用TCP協定建立client-server連線。在這個實驗中，我學會了如何在C語言中使用Socket API，包括建立Socket、綁定Socket到特定地址和埠口、聆聽連線請求、接受連線請求、建立連線、傳送和接收資料等。另外，我也學會了如何使用簡單的命令行參數來讓使用者指定Server的IP地址和埠口，以及如何在程式中動態分配和釋放資源。這些都是 socket programming 中非常重要的技能。

總而言之，這次的實驗讓我對Socket的概念以及使用方法有了更深入的了解，我相信這些知識對我未來的程式開發之路會非常有幫助。

# 參考文獻

<http://zake7749.github.io/2015/03/17/SocketProgramming/>

<https://www.kshuang.xyz/doku.php/programming:c:socket>

<https://github.com/davidleitw/socket>