**Socket Programming作業**

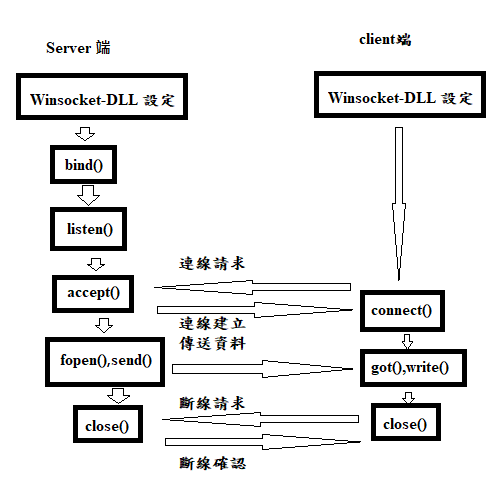
**406262319 資工三乙 黃育晧**

**環境**

使用windows在本機進行架構，server與client都是在本機上去連線與傳資料。

在windows上使用socket的話在編譯時要掛參數 -lws2\_32，令其連結到winsocket的lib去，因此我有寫一個編譯的腳本去進行編譯

**架構**



Server端

由於在windows環境下進行需設置Winsocker-DLL的配置，之後再將本機IP綁定到socket上，開始監聽連線請求，在接受請求後隨即傳送檔案改client端

Client端

一樣進行Winsocker-DLL的配置後對該IP進行連線，接受到資料後將資料寫檔並傳送斷線請求。

**過程**

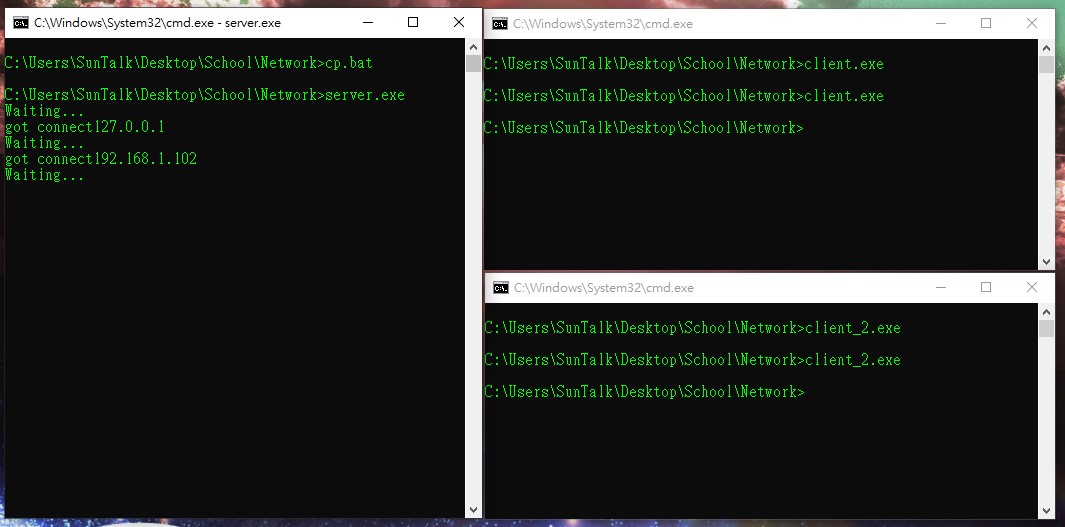
Client端對Server進行連線

Server端接受連線後開啟send.txt檔案並將其傳送給Client端

Client端接收到資料後將資料寫入got.txt檔中

Client端關閉socket斷掉連線

**結果圖**



右上為client.exe檔當作第一個client，會將得到的資料存在got.txt中

右下為client\_2.exe檔當作第二個client，會將得到的資料存在got\_2.txt中

由於都是在本機測試，將IP一個用127.0.0.1表示，另一個用192.168.1.102表示，都是使用內網進行傳輸作業

**Code**

Server.cpp

1. #pragma comment(lib, "Ws2\_32.lib")
3. #include <bits/stdc++.h>
4. #include <WinSock2.h>
6. #define MAXN 1000
7. **using** **namespace** std;
9. **int** main(**int** argc, **char**\* argv[]){
11. **char** message[MAXN];
12. //Winsocket-DLL 設定
13. WSAData wsaData;
14. **WORD** DLLVSERION;
15. DLLVSERION = MAKEWORD(2,1);
16. WSAStartup(DLLVSERION, &wsaData);
18. SOCKADDR\_IN addr;
19. **int** addrlen = **sizeof**(addr);
20. addr.sin\_addr.s\_addr = INADDR\_ANY;
21. addr.sin\_family = AF\_INET;
22. addr.sin\_port = htons(8080);
24. //建立 socket
25. SOCKET sListen;
26. SOCKET sConnect;
27. // windows的socket中 AF\_INET 跟 PF\_INET 是一樣的可以混用
28. sConnect = socket(AF\_INET, SOCK\_STREAM, 0);
29. sListen = socket(AF\_INET, SOCK\_STREAM, 0);
30. bind(sListen, (SOCKADDR\*)&addr, **sizeof**(addr));//將addr的資訊綁到socket上
31. listen(sListen, 3);
33. SOCKADDR\_IN clinetAddr;
34. **while**(**true**)
35. {
36. cout << "Waiting..." << endl;
37. **if**(sConnect = accept(sListen, (SOCKADDR\*)&clinetAddr, &addrlen))
38. {
39. cout<< "got connect" << inet\_ntoa(clinetAddr.sin\_addr) << endl;
41. //傳送訊息給 client 端
42. **FILE** \*fp = fopen("send.txt","r");
43. **char** sendbuf[MAXN],str;
44. **int** len = 0;
45. **while**( ~(str = fgetc(fp)) )
46. sendbuf[len++] = str;
47. sendbuf[len++] = '\0';
48. send(sConnect, sendbuf, (**int**)strlen(sendbuf), 0);
50. }
52. }
54. **return** 0;
56. }

Client.cpp

1. #pragma comment(lib, "Ws2\_32.lib")
3. #include <bits/stdc++.h>
4. #include <WinSock2.h>
6. #define MAXN 1000
7. #define IP "127.0.0.1"
8. **using** **namespace** std;
10. **int** main(**int** argc, **char**\* argv[]){
12. freopen("got.txt","w",stdout);
14. string confirm;
15. **char** message[MAXN];
17. //Winsocket-DLL 設定
18. WSAData wsaData;
19. **WORD** DLLVSERION;
20. DLLVSERION = MAKEWORD(2,1);
21. WSAStartup(DLLVSERION, &wsaData);
23. SOCKADDR\_IN addr;
24. **int** addlen = **sizeof**(addr);
25. addr.sin\_addr.s\_addr = inet\_addr(IP);
26. addr.sin\_family = AF\_INET;
27. addr.sin\_port = htons(8080);
29. //設定 socket
30. SOCKET sConnect;
31. sConnect = socket(AF\_INET, SOCK\_STREAM, 0);
32. connect(sConnect, (SOCKADDR\*)&addr, **sizeof**(addr));
34. //接收 server 端的訊息
35. ZeroMemory(message, MAXN);
36. recv(sConnect, message, **sizeof**(message), 0);
37. cout << message << endl;
39. //關閉socket
40. **BOOL** bDontLinger = FALSE;
41. setsockopt(sConnect,SOL\_SOCKET,SO\_DONTLINGER,(**const** **char**\*)&bDontLinger,**sizeof**(**BOOL**));
42. closesocket(sConnect);
44. **return** 0;
46. }

Client\_2.cpp

1. #pragma comment(lib, "Ws2\_32.lib")
3. #include <bits/stdc++.h>
4. #include <WinSock2.h>
6. #define MAXN 1000
7. #define IP "192.168.1.102"
8. **using** **namespace** std;
10. **int** main(**int** argc, **char**\* argv[]){
12. freopen("got\_2.txt","w",stdout);
14. string confirm;
15. **char** message[MAXN];
17. //Winsocket-DLL 設定
18. WSAData wsaData;
19. **WORD** DLLVSERION;
20. DLLVSERION = MAKEWORD(2,1);
21. WSAStartup(DLLVSERION, &wsaData);
23. SOCKADDR\_IN addr;
24. **int** addlen = **sizeof**(addr);
25. addr.sin\_addr.s\_addr = inet\_addr(IP);
26. addr.sin\_family = AF\_INET;
27. addr.sin\_port = htons(8080);
29. //設定 socket
30. SOCKET sConnect;
31. sConnect = socket(AF\_INET, SOCK\_STREAM, 0);
32. connect(sConnect, (SOCKADDR\*)&addr, **sizeof**(addr));
34. //接收 server 端的訊息
35. ZeroMemory(message, MAXN);
36. recv(sConnect, message, **sizeof**(message), 0);
37. cout << message << endl;
39. //關閉socket
40. **BOOL** bDontLinger = FALSE;
41. setsockopt(sConnect,SOL\_SOCKET,SO\_DONTLINGER,(**const** **char**\*)&bDontLinger,**sizeof**(**BOOL**));
42. closesocket(sConnect);
44. **return** 0;
46. }