**作業系統–作業報告**

# 班級 : 資訊三丙

**座號 :**

**姓名 : 吳柏亞**

**目錄**

1. 題目定義---------------------p.3
2. 程式碼-----------------------p.4
3. 執行結果---------------------p.10
4. 討論-------------------------p.11
5. 心得-------------------------p.12

**題目定義**



**程式碼(server)**

// winsocket\_server.cpp: 定義主控台應用程式的進入點。

//

#include "stdafx.h"

#include <WinSock2.h>

#include <WS2tcpip.h>

#include <windows.h>

#include <stdio.h>

#include <string.h>

#define MAX\_CONNECTIONS 10

#pragma comment(lib, "Ws2\_32.lib")

DWORD WINAPI RunForClientThread(LPVOID);

//Global Varible

SOCKET connections[MAX\_CONNECTIONS];

int connection\_count = 0;

int main()

{

    DWORD ThreadID;

    HANDLE ThreadHandle;

    //用WSAStartup開始Winsock-DLL

    WSADATA wsaData;

    WSAStartup(MAKEWORD(2, 1), &wsaData); //MAKEWORD(2, 1)為Winsocket-DLL版本

    //宣告socket位址資訊

    struct sockaddr\_in addr;

    int addrLen = sizeof(addr);

    //建立socket

    SOCKET sConnect;

    sConnect = socket(AF\_INET, SOCK\_STREAM, NULL);

    //設定位址資訊的資料

    inet\_pton(AF\_INET, "127.0.0.1", &addr.sin\_addr);

    addr.sin\_family = AF\_INET;

    addr.sin\_port = htons(1234);

    //設定監聽Listen Socket

    SOCKET sListen;

    sListen = socket(AF\_INET, SOCK\_STREAM, NULL);

    bind(sListen, (sockaddr\*)&addr, addrLen);

    listen(sListen, SOMAXCONN);

    //宣告clientAddr儲存client的位址資訊

    SOCKADDR\_IN clientAddr;

    printf("Server starting...\n");

    while (true)

    {

        //等待client連線

        if (sConnect = accept(sListen, (SOCKADDR\*)&clientAddr, &addrLen))

        {

            printf("a connection was found!!\n");

            //檢查是否有未建立連線的Socket可用

            int sokcet\_index = -1;

            for (int i=0; i<MAX\_CONNECTIONS; i++)

            {

                if (connections[i] == 0)

                {

                    sokcet\_index = i;

                    break;

                }

            }

            if (sokcet\_index == -1)

            {

                printf("Connection full... \n");

                return 1;

            }

            connections[sokcet\_index] = sConnect;

            connection\_count++;

            ThreadHandle = CreateThread(NULL, 0, RunForClientThread, &sokcet\_index, 0, &ThreadID);

        }

    }

    for(int i=0; i<MAX\_CONNECTIONS;i++)

        closesocket(connections[i]);

    WSACleanup();

return 0;

}

DWORD WINAPI RunForClientThread(LPVOID input\_sIndex) {

    char sendbuf[200];

    int sockIndex = \*(int\*)input\_sIndex;

    while (1) {

        //接收 client 端的訊息

        //ZeroMemory(sendbuf, sizeof(sendbuf));

        if (recv(connections[sockIndex], sendbuf, sizeof(sendbuf), 0) <= 0) {

            closesocket(connections[sockIndex]);

            ZeroMemory(&connections[sockIndex], sizeof(SOCKET));

            printf("Client %d is logout.\n", sockIndex);

            break;

        }

        if (!strlen(sendbuf)) continue;

        //傳送訊息給其他 client 端

        int i = 0;

        printf("%s\n", sendbuf);

        for (i = 0; i < MAX\_CONNECTIONS; i++) {

            if (connections[i] == 0 || i == sockIndex) continue;

            send(connections[i], sendbuf, (int)strlen(sendbuf) + 1, 0);

        }

    }

    return 0;

}

**程式碼(client)**

// winsocket\_client.cpp: 定義主控台應用程式的進入點。

//

#include "stdafx.h"

#include <WinSock2.h>

#include <WS2tcpip.h>

#include <stdio.h>

#include <string.h>

#define MSGBUF\_SIZE 1024

#define ID\_SIZE 100

#pragma comment(lib, "Ws2\_32.lib")

DWORD WINAPI ListenServerMsg(LPVOID);

int flushFile(FILE\*);

static char ID[ID\_SIZE];

void main()

{

    char sendMsg[MSGBUF\_SIZE];

    char sendToServer[ID\_SIZE + MSGBUF\_SIZE];

    //Server監聽Thread的變數

    DWORD ThreadID;

    HANDLE ListenServer;

    //開始 Winsock-DLL

    WSAData wsaData;

    WSAStartup(MAKEWORD(2, 1), &wsaData);

    //宣告給 socket 使用的 sockadder\_in 結構

    SOCKADDR\_IN addr;

    int addlen = sizeof(addr);

    //設定 socket

    SOCKET sConnect;

    sConnect = socket(AF\_INET, SOCK\_STREAM, NULL);

    //設定欲連線的Server的位址資訊

    inet\_pton(AF\_INET, "127.0.0.1", &addr.sin\_addr);

    addr.sin\_family = AF\_INET;

    addr.sin\_port = htons(1234);

    printf("Enter your chat ID.(enter \"quit\" to give up login) -->");

    scanf\_s("%s", &ID, (unsigned int)sizeof(ID));

    if (strcmp(ID, "quit") != 0) {

        connect(sConnect, (SOCKADDR\*)&addr, sizeof(addr));

        //建立執行緒監聽來自Server的他人訊息

        ListenServer = CreateThread(NULL, 0, ListenServerMsg, &sConnect, 0, &ThreadID);

        //工具函式，清除輸入緩衝區

        flushFile(stdin);

        while (1) {

            ZeroMemory(sendMsg, sizeof(sendMsg));

            printf("%s: ", ID);

            //取得一個非空的行(獲得一個非零長度的輸入字串)

            while (!strlen(sendMsg)) gets\_s(sendMsg, sizeof(sendMsg));

            //設計的登出指令(:q)

            if (sendMsg[0] == ':') {

                if (strlen(sendMsg) == 2 && sendMsg[1] == 'q') {

                    break;

                }

            }

            //送出包含ID的訊息

            sprintf\_s(sendToServer, sizeof(sendToServer), "%s: %s", ID, sendMsg);

            send(sConnect, sendToServer, (int)strlen(sendToServer) + 1, 0);

        }

    }

    closesocket(sConnect);

    WSACleanup();

    printf("\nBye Bye...\n\n");

    system("pause");

}

DWORD WINAPI ListenServerMsg(LPVOID listenSOCKET) {

    //char EnterBuf[MSGBUF\_SIZE];

    char recvMsg[ID\_SIZE + MSGBUF\_SIZE];

    SOCKET ServSOCKET = \*(SOCKET\*)listenSOCKET;

    while (1) {

        //接收 server 端的訊息

        ZeroMemory(recvMsg, sizeof(recvMsg));

        if (recv(ServSOCKET, recvMsg, sizeof(recvMsg), 0) <= 0) {

            printf("\nLost Connect to Server.\n");

            break;

        }

        printf("\r%s\n", recvMsg);

        printf("%s: ", ID);

    }

    return 0;

}

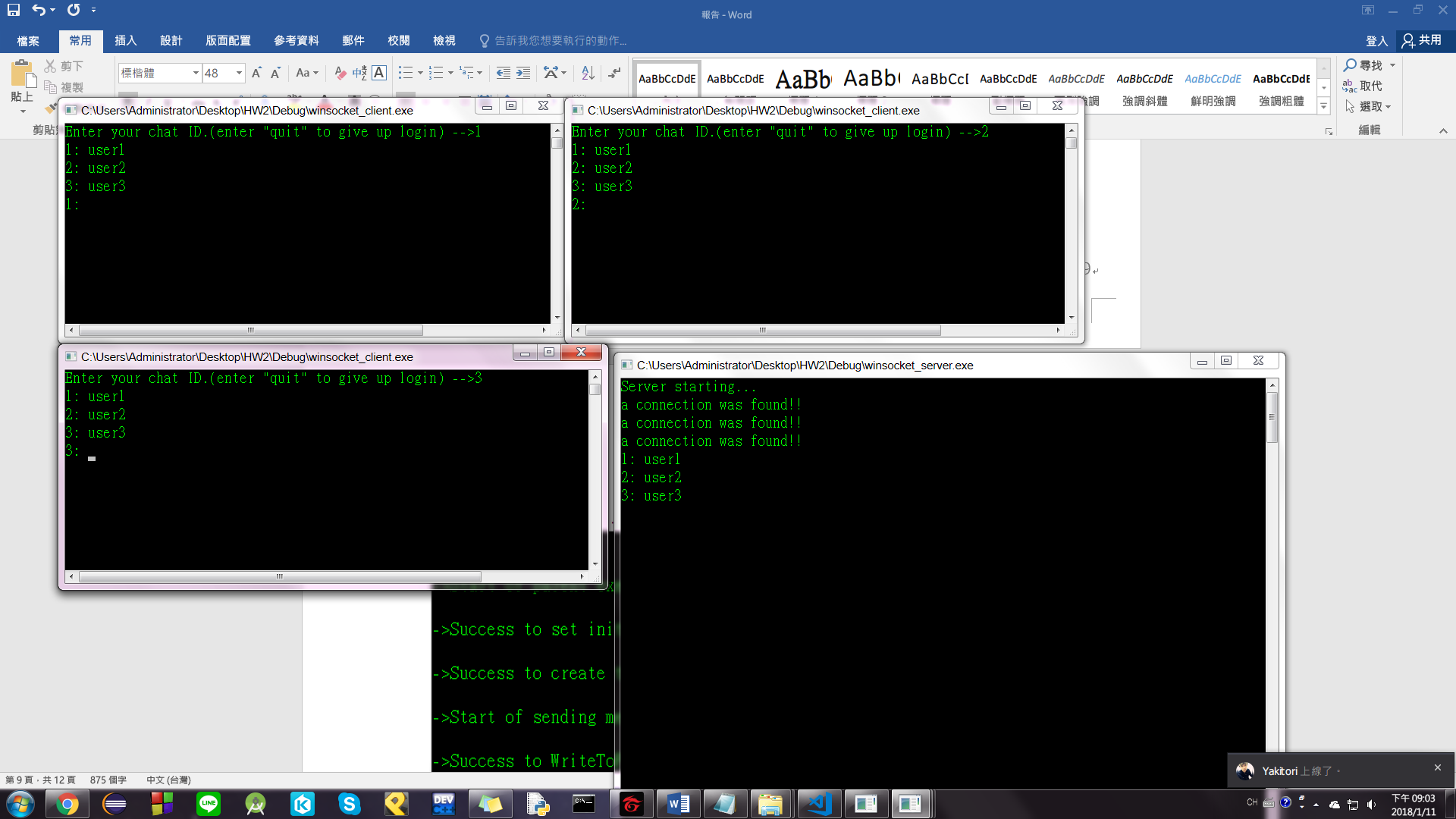
int flushFile(FILE\* target) {

    char discard;

    while ((discard = fgetc(stdin)) != '\n' && discard != EOF) {}

    return 0;

}

**執行結果**

**討論**

**學長提供的程式碼在client方面一登入馬上變結束程式，所以要使用迴圈來不斷得讓使用者可以輸入，問題在於要不斷的檢測input(使用者輸入)還是output(server訊息)，這兩個唯一必須不斷執行接受的就是顯示server的output，所以必須在成功建立連線的時候執行一個執行緒去跑接受訊息的工作。**

**心得**

**這次的作業需要再建立一個執行緒去接收訊息，雖然照著程式本來的執行緒複製一個，但弄了許多還是搞不出個頭緒，最後借同學的來參考，這就是差距阿!!!**