作業系統

## 作業(二) - Sockets, Multithread練習

D00426511

萬栩峰

程式:winsocket\_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);

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];

char recvbuf[200],name[200],nametmp[200];

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

bool index = true;

int read\_size;

ZeroMemory(recvbuf, 200);

//傳送訊息給 client 端

//strcpy\_s(sendbuf, "Hello client!! please input your name: ");

char sendbuf[200];

ZeroMemory(sendbuf, 200);

strcat\_s(sendbuf, "Hello client!! ");

ZeroMemory(name, 200);

recv(connections[sockIndex], name, sizeof(name), 0);

//printf("your user name is %s", name);

strcat\_s(name, " : ");

strcpy\_s(nametmp , name);

send(connections[sockIndex], sendbuf, (int)strlen(sendbuf), 0);

do {

ZeroMemory(name, 200);

ZeroMemory(recvbuf, 200);

recv(connections[sockIndex], recvbuf, sizeof(recvbuf), 0);

strcpy\_s(name,nametmp);

strcat\_s(name, recvbuf);

if (strcmp(recvbuf, "exit") == 0) { //判斷是否為結束

send(connections[sockIndex], "exit", (int)strlen("exit"), 0);

connections[sockIndex] = 0; //回收socket

index = false;

}

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

if (i == sockIndex) continue;

send(connections[i], name, (int)strlen(name), 0);

//send(connections[i], recvbuf, (int)strlen(recvbuf), 0);

}

} while (index);

//接收 client 端的訊息,

//recv(input\_socket, sendbuf, sizeof(sendbuf), 0);

return 0;

}

程式:\_winsocket\_client

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

//

#include "stdafx.h"

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

#include <WinSock2.h>

#include <WS2tcpip.h>

#include <stdio.h>

#include <string.h>

#include <windows.h>

SOCKET sConnect;

DWORD WINAPI InputThread(LPVOID);

void main()

{

char confirm;

char recvMsg[200];

char name[200];

char sendbuf[200];

bool index = true;

char server\_reply[200];

DWORD ThreadID;

HANDLE ThreadHandle;

int r;

//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

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("connect to server?[Y] or [N] --> ");

scanf\_s("%c", &confirm,1);

if (confirm == 'N')

{

exit(1);

}

else {

if (confirm == 'Y')

{

printf("please input your name: ");

scanf\_s("%s",&name,sizeof(name));

//printf("name%s", name);

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

send(sConnect, name, (int)strlen(name), 0);

ZeroMemory(recvMsg, 200);

r=recv(sConnect, recvMsg, sizeof(recvMsg), 0);

printf("\n%s", recvMsg);

//send(sConnect, name, (int)strlen(name), 0);

// memset(&sendbuf, 0, sizeof(sendbuf));

// memset(&recvMsg, 0, sizeof(recvMsg));

ThreadHandle = CreateThread(NULL, 0, InputThread, 0, 0, &ThreadID);

//接收 server 端的訊息

do

{

ZeroMemory(recvMsg, 200);

recv(sConnect, recvMsg, sizeof(recvMsg), 0);

if (strcmp(recvMsg, "exit") == 0) {

break;

}

printf("%s\n", recvMsg);

// memset(&sendbuf, 0, sizeof(sendbuf));

// memset(&recvMsg, 0, sizeof(recvMsg));

} while (index);

}

}

closesocket(sConnect);

WSACleanup();

printf("bye bye...\n");

system("pause");

}

DWORD WINAPI InputThread(LPVOID) { //接收 server 端的訊息 D0409929

char sendMsg[200];

printf("( Enter \"exit\" to close )\n");

while (true) {

ZeroMemory(sendMsg, 200);

printf("\nenter Msg: \n");

gets\_s(sendMsg);

send(sConnect, sendMsg, (int)strlen(sendMsg), 0);

if (strcmp(sendMsg, "exit") == 0) {

break;

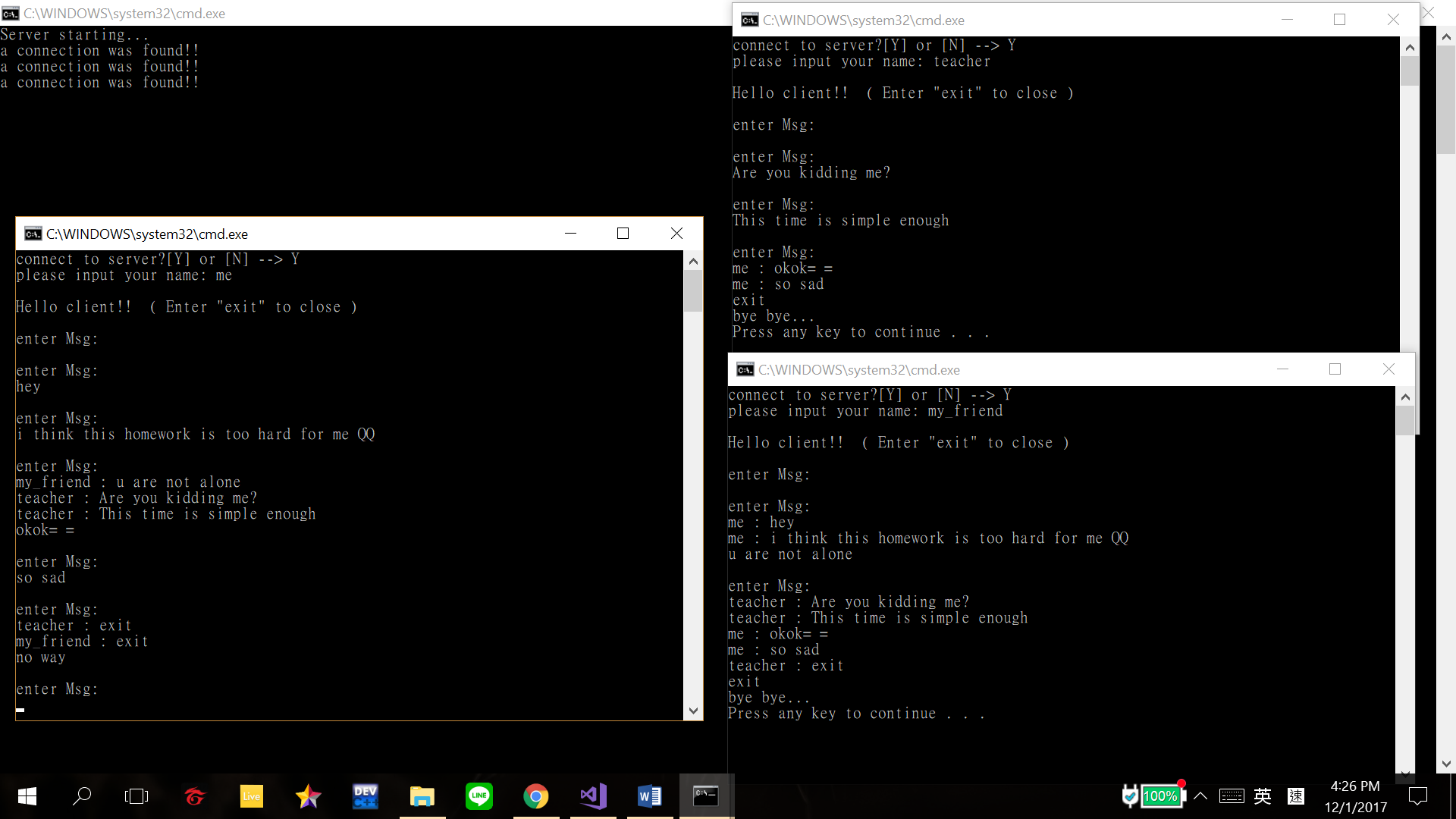
}

}

return 0;

}

運算畫面:



遇到的困難:

1. 很難。
2. Sockets好復雜。
3. 一開始沒想到用Multithread去寫client的，就有delay的問題。後來準備要交屍體時才看到題目說要用thread，後來改一下就好了。