電腦網路

LAB:LAB3

報告

班級:資工 2A

學號:01057020

姓名:林佑丞

1.

code:

Server 端:

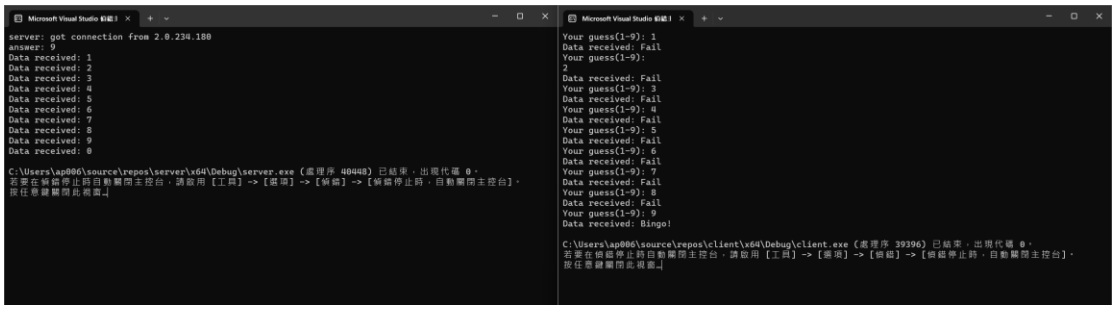
```
#include <winsock2.h>
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>
#include <cstdlib>
#include <ctime>
#include <thread>
#include <vector>
#include <string>
using namespace std;

int x;

const wchar_t* GetWC(const char* c) { //將字串從const char* 轉為 const wchar_t*
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}

int main() {
    WSAData wsaData;

    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sListen;
    SOCKET sConnect;

    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sListen = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    //AF_INET:IPv4    AF_INET6:IPv6
    //SOCK_STREAM:TCP  SOCK_DGRAM:UDP
    //IPPROTO_TCP:TCP  IPIROTO_UDP:UDP
```

```
    bind(sListen, (SOCKADDR*)&addr, sizeof(SOCKADDR)); //server端綁定位置
    listen(sListen, 20);

    srand(time(NULL));

    sockaddr_in clientAddr; // client 端位址資訊
    int clientAddrLen = sizeof(clientAddr);
    sConnect = accept(sListen, (SOCKADDR*)&clientAddr, &clientAddrLen);
    if (sConnect != INVALID_SOCKET)
    {
        // 有 client 端成功連線過來
        int x = rand() % 9 + 1;
        char clientIP[20];
        inet_ntop(AF_INET, (void*)&clientAddr, clientIP, 20);
        printf("server: got connection from %s\n", clientIP);
        printf("answer: %d\n", x);
        while (true) {
            char recvbuf[2];
            recv(sConnect, recvbuf, sizeof(recvbuf), 0);
            int guess = atoi(recvbuf);
            printf("Data received: %d\n", guess);
            if (guess == x) {
                const char* s = "Bingo!";
                send(sConnect, s, (int)strlen(s), 0);
            }
            else if (guess == 0)
                break;
            else {
                const char* s = "Fail";
                send(sConnect, s, (int)strlen(s), 0);
            }
            memset(recvbuf, 0, sizeof(recvbuf));
        }

    }

    closesocket(sConnect);
    closesocket(sListen);

    WSACleanup();

    return 0;
}
```

Client 端:

```c
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>

const wchar_t* GetWC(const char* c) {
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}

int main() {
    WSAData wsaData;

    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sock;

    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

    connect(sock, (SOCKADDR*)&addr, sizeof(SOCKADDR));

    char szBuffer[MAXBYTE] = { 0 };

    int guess;
    int iResult;
    while (true) {
        printf("Your guess(1-9): ");
        scanf_s("%d", &guess);

        char sendbuf[2];
        sprintf_s(sendbuf, sizeof(sendbuf), "%d", guess);

        iResult = send(sock, sendbuf, (int)strlen(sendbuf), 0);
        if (iResult == SOCKET_ERROR) {
            printf("send failed with error: %d\n", WSAGetLastError());
            closesocket(sock);
            WSACleanup();
        }
        else if (iResult > 0) {
            recv(sock, szBuffer, MAXBYTE, NULL);
            printf("Data received: %s\n", szBuffer);
            if (strcmp(szBuffer, "Bingo!") == 0)
                break;
        }
        else if (iResult == 0) {
            printf("Connection closed\n");
            break;
        }
        else {
            printf("recv failed with error: %d\n", WSAGetLastError());
            closesocket(sock);
            WSACleanup();
            return 1;
        }
    }

    closesocket(sock);

    WSACleanup();

    return 0;
}
```

執行結果:



```
server: got connection from 2.0.234.180
answer: 9
Data received: 1
Data received: 2
Data received: 3
Data received: 4
Data received: 5
Data received: 6
Data received: 7
Data received: 8
Data received: 9
Data received: 0

C:\Users\ap006\source\repos\server\x64\Debug\server.exe (處理序 40448) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...
```

```
Your guess(1-9): 1
Data received: Fail
Your guess(1-9):
2
Data received: Fail
Your guess(1-9): 3
Data received: Fail
Your guess(1-9): 4
Data received: Fail
Your guess(1-9): 5
Data received: Fail
Your guess(1-9): 6
Data received: Fail
Your guess(1-9): 7
Data received: Fail
Your guess(1-9): 8
Data received: Fail
Your guess(1-9): 9
Data received: Bingo!

C:\Users\ap006\source\repos\client\x64\Debug\client.exe (處理序 39396) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...
```

2.

code:

Server 端

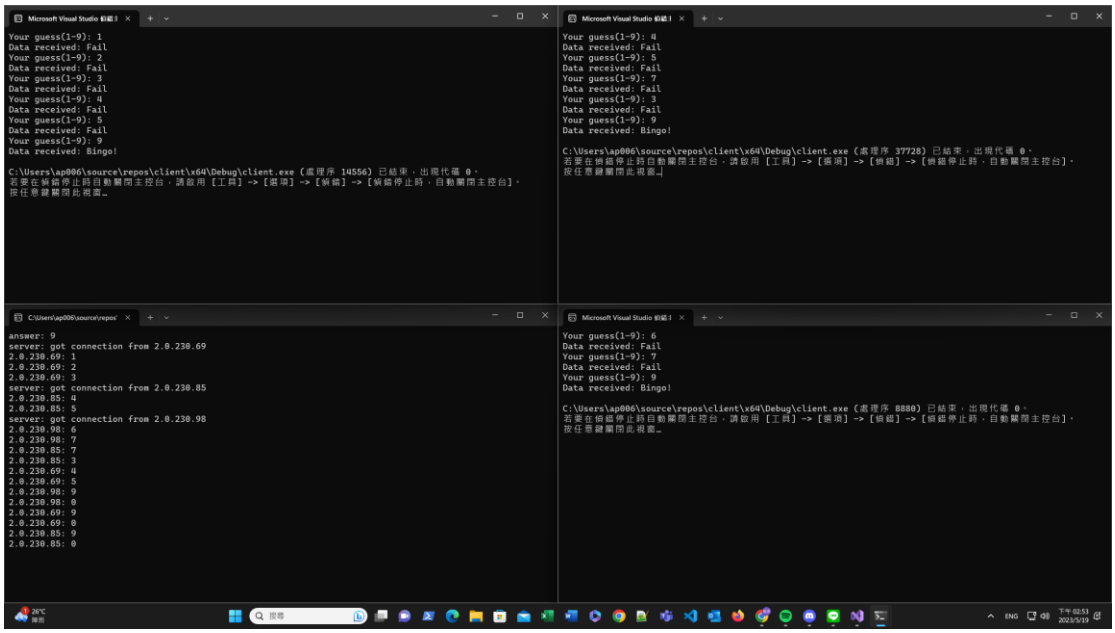
```
#include <winsock2.h>
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>
#include <cstdlib>
#include <ctime>
#include <thread>
#include <vector>
#include <string>
using namespace std;

int x;

const wchar_t* GetWC(const char* c) { //將字串從const char* 轉為 const wchar_t*
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}

void check(SOCKET Connect, string ip) {
    while (true) {
        char recvbuf[2];
        recv(Connect, recvbuf, sizeof(recvbuf), 0);
        int guess = atoi(recvbuf);
        printf("%s: %d\n", ip.c_str(), guess);
        if (guess == x) {
            const char* s = "Bingo!";
            send(Connect, s, (int)strlen(s), 0);
        }
        else if (guess == 0)
            break;
        else {
            const char* s = "Fail";
            send(Connect, s, (int)strlen(s), 0);
        }
        memset(recvbuf, 0, sizeof(recvbuf));
    }
}
//*
```

```
int main() {
    WSAData wsaData;
    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sListen;
    SOCKET sConnect;
    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sListen = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    //AF_INET:IPv4    AF_INET6:IPv6
    //SOCK_STREAM:TCP  SOCK_DGRAM:UDP
    //IPPROTO_TCP:TCP  IPIROTO_UDP:UDP
    bind(sListen, (SOCKADDR*)&addr, sizeof(SOCKADDR)); //server端綁定位置
    listen(sListen, 20);

    srand(time(NULL));

    vector<thread> clients;
    vector<string> ip;
    x = rand() % 9 + 1;
    printf("answer: %d\n", x);

    while (true) {
        sockaddr_in clientAddr; // client 端位址資訊
        int clientAddrLen = sizeof(clientAddr);
        sConnect = accept(sListen, (SOCKADDR*)&clientAddr, &clientAddrLen);

        if (sConnect != INVALID_SOCKET)
        {
            // 有 client 端成功連線過來
            string clientIP;
            inet_ntop(AF_INET, (void*)&clientAddr, const_cast<char *>(clientIP.c_str()), 20);
            printf("server: got connection from %s\n", clientIP.c_str());
            ip.push_back(clientIP);

            thread newConnect(check, sConnect, ip[ip.size() - 1]);
            newConnect.detach();

        }
    }

    closesocket(sConnect);
    closesocket(sListen);

    WSACleanup();

    return 0;
}//*/
```

Client 端:

```c
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>

const wchar_t* GetWC(const char* c) {
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}

int main() {
    WSAData wsaData;

    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sock;

    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

    connect(sock, (SOCKADDR*)&addr, sizeof(SOCKADDR));

    char szBuffer[MAXBYTE] = { 0 };

    int guess;
    int iResult;
    while (true) {
        printf("Your guess(1-9): ");
        scanf_s("%d", &guess);

        char sendbuf[2];
        sprintf_s(sendbuf, sizeof(sendbuf), "%d", guess);

        iResult = send(sock, sendbuf, (int)strlen(sendbuf), 0);
        if (iResult == SOCKET_ERROR) {
            printf("send failed with error: %d\n", WSAGetLastError());
            closesocket(sock);
            WSACleanup();
        }
        else if (iResult > 0) {
            recv(sock, szBuffer, MAXBYTE, NULL);
            printf("Data received: %s\n", szBuffer);
            if (strcmp(szBuffer, "Bingo!") == 0)
                break;
        }
        else if (iResult == 0) {
            printf("Connection closed\n");
            break;
        }
        else {
            printf("recv failed with error: %d\n", WSAGetLastError());
            closesocket(sock);
            WSACleanup();
            return 1;
        }
    }

    closesocket(sock);

    WSACleanup();

    return 0;
}
```

執行結果:


Microsoft Visual Studio 偵錯主
Your guess(1-9): 1
Data received: Fail
Your guess(1-9): 2
Data received: Fail
Your guess(1-9): 3
Data received: Fail
Your guess(1-9): 4
Data received: Fail
Your guess(1-9): 5
Data received: Fail
Your guess(1-9): 9
Data received: Bingo!
C:\Users\ap006\source\repos\client\x64\Debug\client.exe (處理序 14556) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...
Microsoft Visual Studio 偵錯主
Your guess(1-9): 4
Data received: Fail
Your guess(1-9): 5
Data received: Fail
Your guess(1-9): 7
Data received: Fail
Your guess(1-9): 3
Data received: Fail
Your guess(1-9): 9
Data received: Bingo!
C:\Users\ap006\source\repos\client\x64\Debug\client.exe (處理序 37728) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...
C:\Users\ap006\source\repos
answer: 9
server: got connection from 2.0.230.69
2.0.230.69: 1
2.0.230.69: 2
2.0.230.69: 3
server: got connection from 2.0.230.85
2.0.230.85: 4
2.0.230.85: 5
server: got connection from 2.0.230.98
2.0.230.98: 6
2.0.230.98: 7
2.0.230.85: 7
2.0.230.85: 3
2.0.230.69: 4
2.0.230.69: 5
2.0.230.98: 9
2.0.230.98: 0
2.0.230.69: 9
2.0.230.69: 0
2.0.230.85: 9
2.0.230.85: 0
Microsoft Visual Studio 偵錯主
Your guess(1-9): 6
Data received: Fail
Your guess(1-9): 7
Data received: Fail
Your guess(1-9): 9
Data received: Bingo!
C:\Users\ap006\source\repos\client\x64\Debug\client.exe (處理序 8880) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...

3.

code:

Server 端:

```
#include <winsock2.h>
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>
#include <cstdlib>
#include <thread>
#include <vector>
#include <string>
#include <iostream>
using namespace std;

vector<pair<string, SOCKET> > ip;

const wchar_t* GetWC(const char* c) {
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}

void check(SOCKET Connect, int idx) {
    while (true) {
        char recvbuf[20000] = { '\0' };
        int now = recv(Connect, recvbuf, sizeof(recvbuf), 0);
        if (now == 0 || now == -1) break;

        string mes = ip[idx].first + ": " + recvbuf;
        cout << mes << endl;

        for (int i = 0; i < ip.size(); i++)
            send(ip[i].second, mes.c_str(), (int)mes.size(), 0);
        memset(recvbuf, 0, sizeof(recvbuf));
    }
}
//*
```

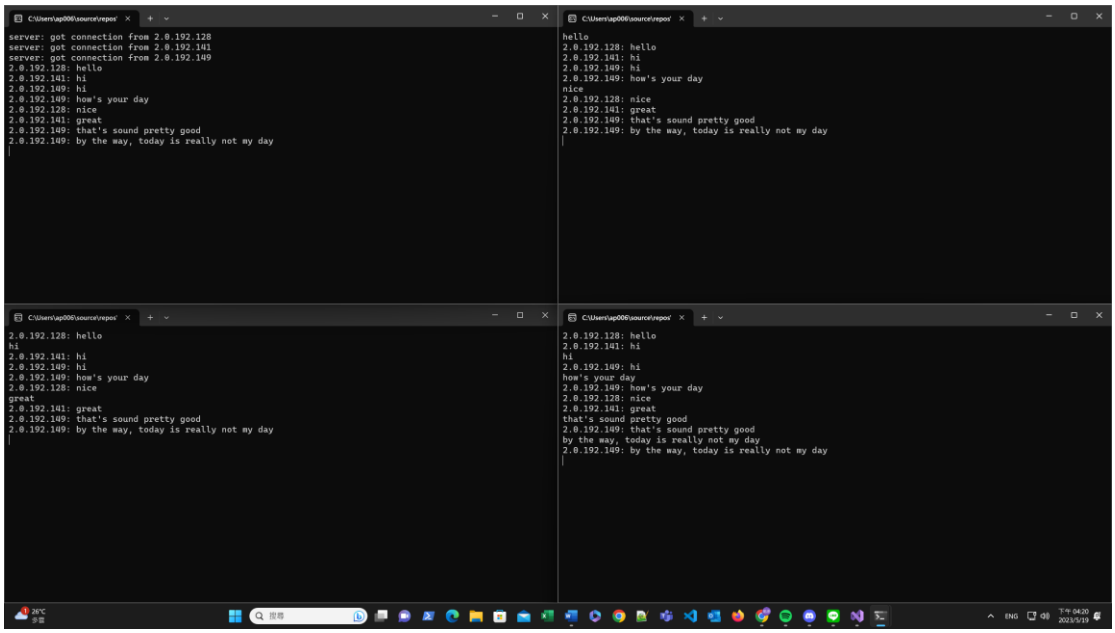
```
int main() {
    WSAData wsaData;
    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sListen;
    SOCKET sConnect;
    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sListen = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    //AF_INET:IPv4    AF_INET6:IPv6
    //SOCK_STREAM:TCP  SOCK_DGRAM:UDP
    //IPPROTO_TCP:TCP  IPIROTO_UDP:UDP
    bind(sListen, (SOCKADDR*)&addr, sizeof(SOCKADDR)); //server端綁定位置
    listen(sListen, 20);


    vector<thread> clients;

    while (true) {
        sockaddr_in clientAddr; // client 端位址資訊
        int clientAddrLen = sizeof(clientAddr);
        sConnect = accept(sListen, (SOCKADDR*)&clientAddr, &clientAddrLen);

        if (sConnect != INVALID_SOCKET)
        {
            // 有 client 端成功連線過來
            char clientIP[20];
            inet_ntop(AF_INET, (void*)&clientAddr, clientIP, 20);
            printf("server: got connection from %s\n", clientIP);
            ip.emplace_back(clientIP, sConnect);

            thread newConnect(check, sConnect, ip.size() - 1);
            newConnect.detach();
        }
    }

    closesocket(sConnect);
    closesocket(sListen);

    WSACleanup();

    return 0;
}//*/
```

Client 端:

```cpp
#include <stdio.h>
#include <winsock2.h>
#include <Ws2tcpip.h>
#include <string>
#include <iostream>
#include <thread>
using namespace std;

const wchar_t* GetWC(const char* c) {
    const size_t cSize = strlen(c) + 1;
    wchar_t* wc = new wchar_t[cSize];
    size_t convertedChar = 0;
    mbstowcs_s(&convertedChar, wc, cSize, c, _TRUNCATE);
    return wc;
}
void getMes(SOCKET sock) {
    while (true) {
        char szBuffer[20000] = { 0 };
        recv(sock, szBuffer, 20000, NULL);
        cout << szBuffer << endl;
    }
}

int main() {
    WSAData wsaData;

    if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0) {
        fprintf(stderr, "WSAStartup failed.\n");
        exit(1);
    }

    SOCKET sock;

    sockaddr_in addr;
    InetPton(AF_INET, GetWC("127.0.0.1"), &addr.sin_addr.s_addr); // 設定 IP
    addr.sin_family = AF_INET;
    addr.sin_port = htons(1234); // 設定 port,htons()跟網路位元組順序有關

    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

    connect(sock, (SOCKADDR*)&addr, sizeof(SOCKADDR));
```

```cpp
    char szBuffer[20000] = { 0 };
    thread get_mes(getMes, sock);
    get_mes.detach();

    while (true) {

        string mes;
        getline(cin, mes);
        char* sendbuf = const_cast<char*>(mes.c_str());
        send(sock, sendbuf, (int)strlen(sendbuf), 0);

    }

    closesocket(sock);

    WSACleanup();

    return 0;
}
```

執行結果:


C:\Users\ap006\source\repos\
server: got connection from 2.0.192.128
server: got connection from 2.0.192.141
server: got connection from 2.0.192.149
2.0.192.128: hello
2.0.192.141: hi
2.0.192.149: hi
2.0.192.149: how's your day
2.0.192.128: nice
2.0.192.141: great
2.0.192.149: that's sound pretty good
2.0.192.149: by the way, today is really not my day
C:\Users\ap006\source\repos\
hello
2.0.192.128: hello
2.0.192.141: hi
2.0.192.149: hi
2.0.192.149: how's your day
nice
2.0.192.128: nice
2.0.192.141: great
2.0.192.149: that's sound pretty good
2.0.192.149: by the way, today is really not my day
C:\Users\ap006\source\repos\
2.0.192.128: hello
hi
2.0.192.141: hi
2.0.192.149: hi
2.0.192.149: how's your day
2.0.192.128: nice
great
2.0.192.141: great
2.0.192.149: that's sound pretty good
2.0.192.149: by the way, today is really not my day
C:\Users\ap006\source\repos\
2.0.192.128: hello
2.0.192.141: hi
hi
2.0.192.149: hi
how's your day
2.0.192.149: how's your day
2.0.192.128: nice
2.0.192.141: great
that's sound pretty good
2.0.192.149: that's sound pretty good
by the way, today is really not my day
2.0.192.149: by the way, today is really not my day