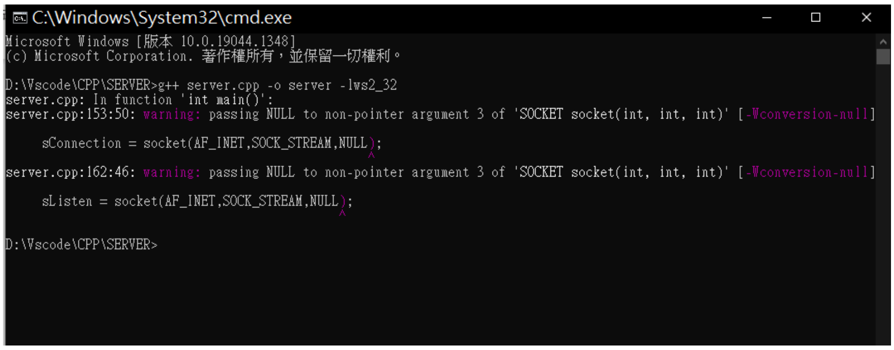
計算機網路作業報告

B0829039 王語堂

2021 年 12 月 20 日

操作方式:

由於本程式使用的是 winsock2.h，因此需在 windows 系統上做測試，首先須用 terminal cd 到放置 server.cpp 的資料夾，並以 g++ server.cpp -o server -lws2_32 對 server.cpp 進行編譯，並以./server 或.\server 執行，之後回答是否要設定 port number，如要設定的話請輸入 Y 後再輸入 port number，否的話則輸入 N，則 port number 將維持預設的 80。接著便會看到 waiting for connect 之字樣，此時便可開啟瀏覽器並在網址列輸入 127.0.0.1:(你所設定的 port name，若無設定可不用加上)，此時會看到 404 not found 之字樣，並且狀態碼也為 404。

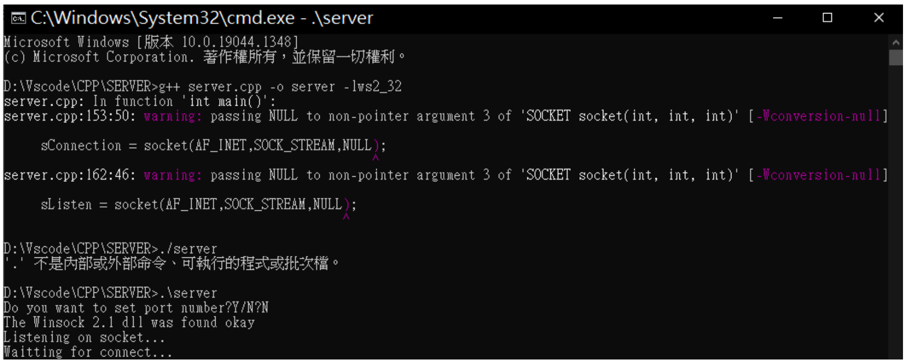


C:\Windows\System32\cmd.exe

```
Microsoft Windows [版本 10.0.19044.1348]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

D:\Vscode\CPP\SERVER>g++ server.cpp -o server -lws2_32
server.cpp: In function 'int main()':
server.cpp:153:50: warning: passing NULL to non-pointer argument 3 of 'SOCKET socket(int, int, int)' [-Wconversion-null]
     sConnection = socket(AF_INET,SOCK_STREAM,NULL);
                                                  ^
server.cpp:162:46: warning: passing NULL to non-pointer argument 3 of 'SOCKET socket(int, int, int)' [-Wconversion-null]
     sListen = socket(AF_INET,SOCK_STREAM,NULL);
                                              ^

D:\Vscode\CPP\SERVER>
```



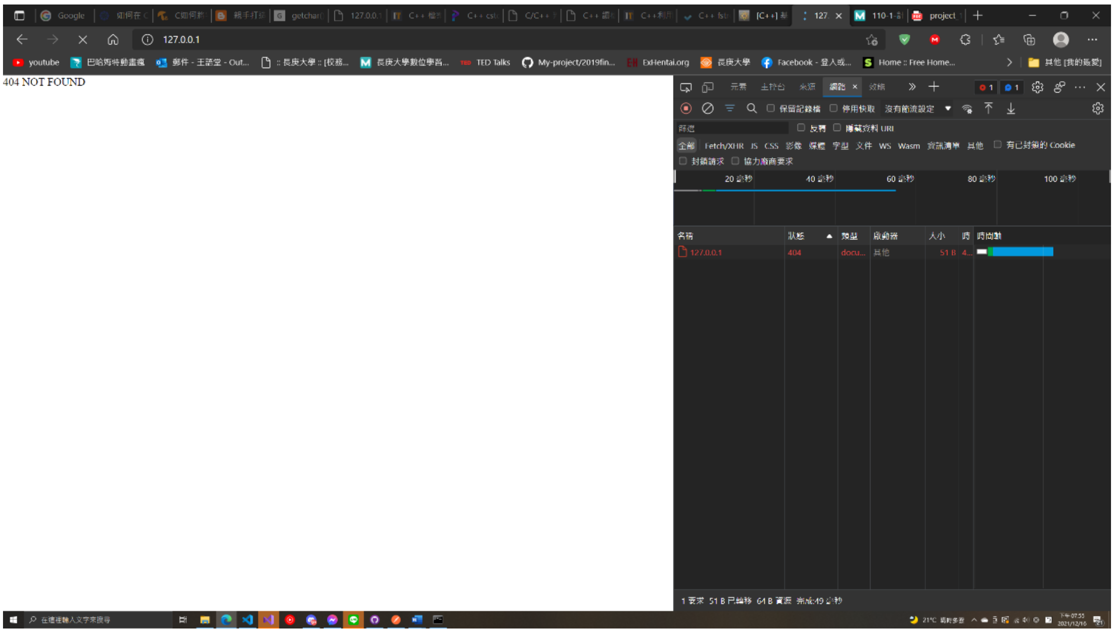
C:\Windows\System32\cmd.exe - .\server

```
Microsoft Windows [版本 10.0.19044.1348]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

D:\Vscode\CPP\SERVER>g++ server.cpp -o server -lws2_32
server.cpp: In function 'int main()':
server.cpp:153:50: warning: passing NULL to non-pointer argument 3 of 'SOCKET socket(int, int, int)' [-Wconversion-null]
     sConnection = socket(AF_INET,SOCK_STREAM,NULL);
                                                  ^
server.cpp:162:46: warning: passing NULL to non-pointer argument 3 of 'SOCKET socket(int, int, int)' [-Wconversion-null]
     sListen = socket(AF_INET,SOCK_STREAM,NULL);
                                              ^

D:\Vscode\CPP\SERVER>./server
'.' 不是內部或外部命令、可執行的程式或批次檔。

D:\Vscode\CPP\SERVER>.\server
Do you want to set port number?Y/N?N
The Winsock 2.1 dll was found okay
Listening on socket...
Waitting for connect...
```

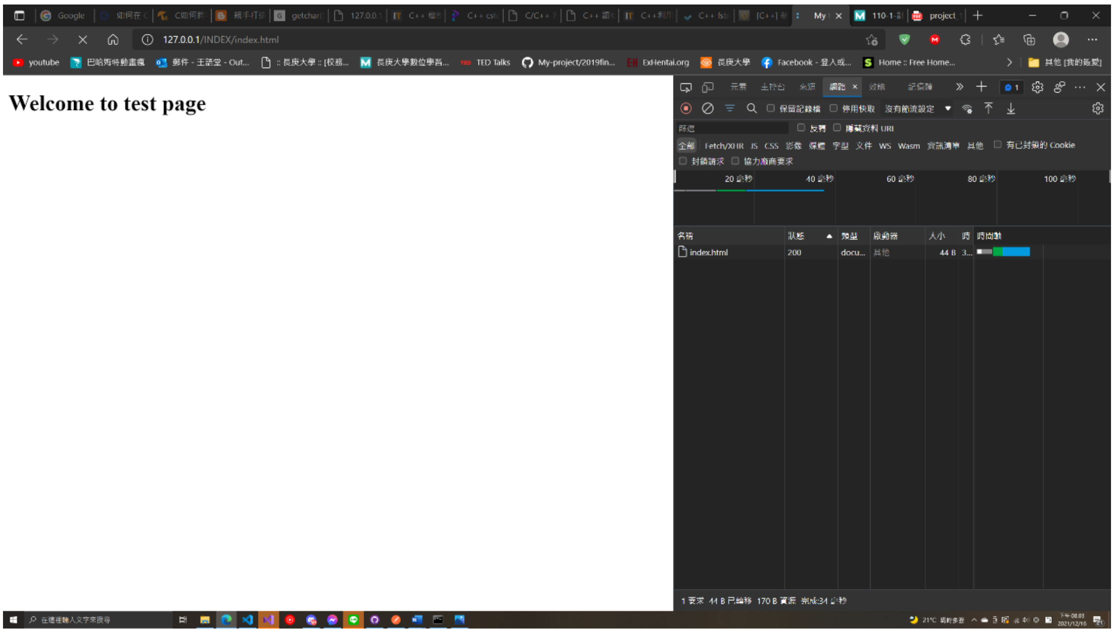


127.0.0.1

404 NOT FOUND

名稱	狀態	類型	啟動器	大小	時	時間軸
127.0.0.1	404	docu...	其他	51 B	4...	

而在這之後我們可以在網址後加上所要 get 的檔案路徑，要注意本程式只能 get html 或 text 檔，如 127.0.0.1/INDEX/index.html 或者 127.0.0.1/test.txt，我們便能於瀏覽器上看到檔案之內容，且狀態碼為 200。

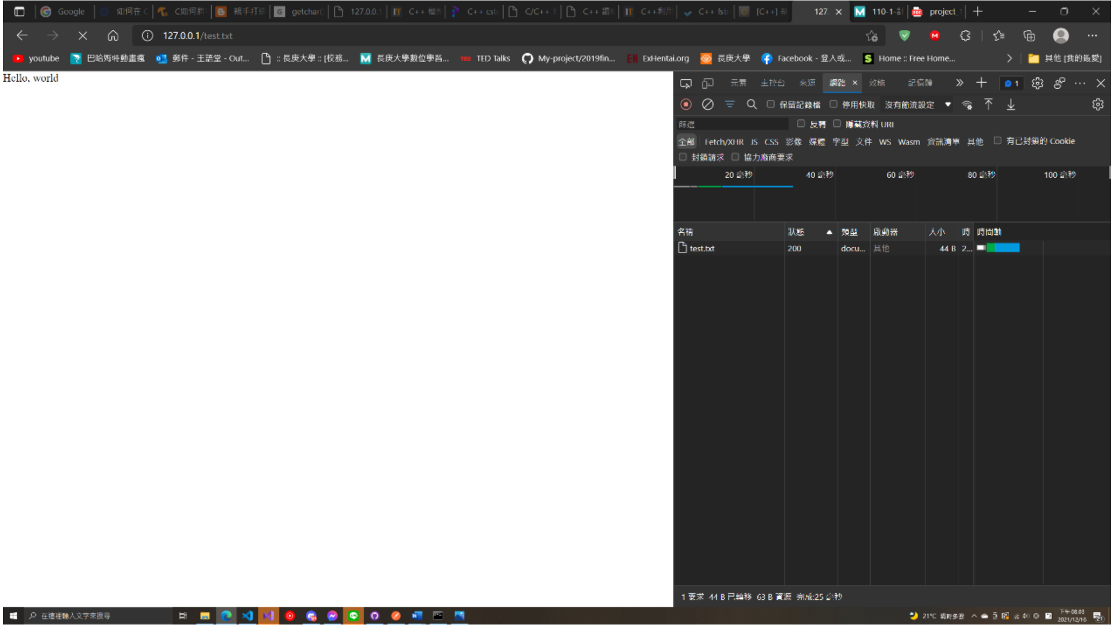


127.0.0.1/INDEX/index.html

Welcome to test page

名稱	狀態	類型	啟動器	大小	時
index.html	200	docu...	其他	44 B	3...

1 要求 44 B 已轉移 170 B 資源 完成:34 毫秒



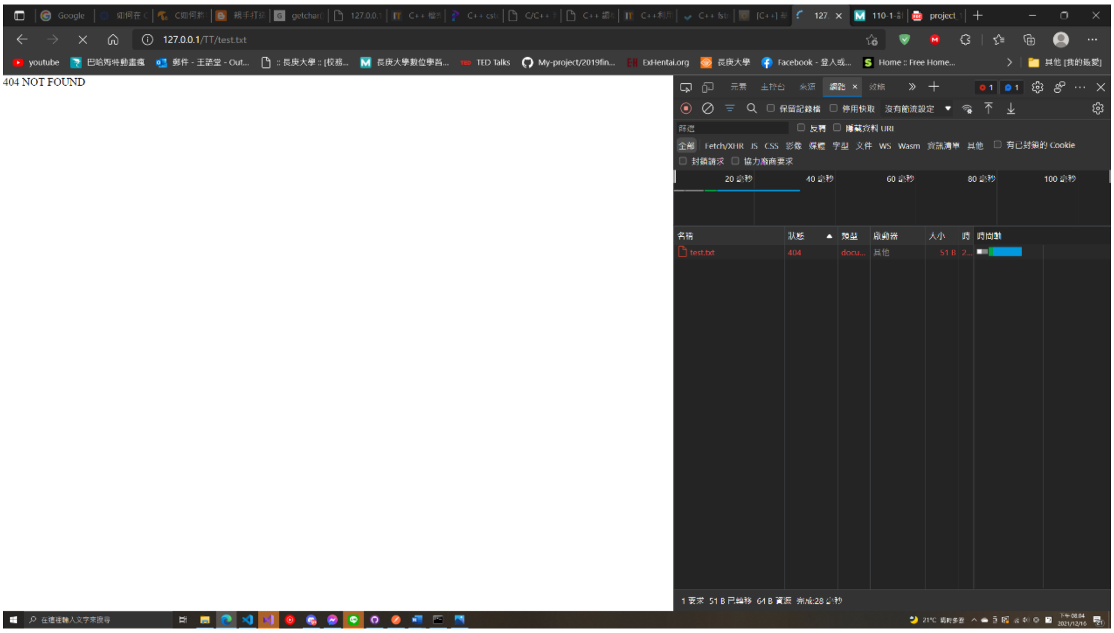
127.0.0.1/test.txt

Hello, world

名稱	狀態	類型	啟動器	大小	時
test.txt	200	docu...	其他	44 B	2...

1 要求 44 B 已轉移 63 B 資源 完成:25 毫秒

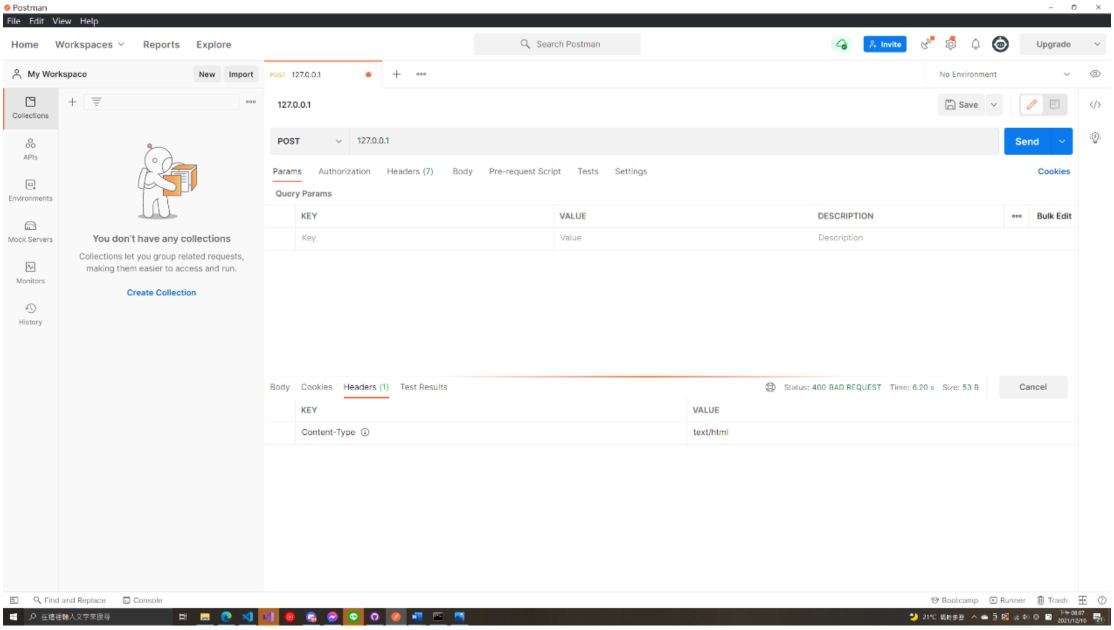
若要求的檔案路徑不存在則會回傳 404



```
404 NOT FOUND
```

名稱	狀態	類型	啟動器	大小
test.txt	404	docu...	其他	51 B

若要測試 BAD REQUEST 的話可使用 POSTMAN 等軟體。由於本伺服器只接受 GET request，因此這之外的 request 如 POST，或是錯誤格式的 request 都會收到 HTTP/1.0 400 BAD REQUEST。



POST 127.0.0.1

Status: 400 BAD REQUEST Time: 6.20 s Size: 53 B

KEY	VALUE
Content-Type	text/html

程式碼及說明:

功能分為幾個函式，分別為 GET(SOCKET sConnection,char *filename)，BADREQUEST(SOCKET sConnection)，NOTFOND(SOCKET sConnection)及 EXIST(SOCKET sConnection,char *filename)，GET 用在 request 的物件存在時對檔案進行讀取並 send 至 client，NOTFOUND 則是在 request 要求的物件不存在時進行回覆，若 request 的語法有誤，BADREQUEST 便會回覆符合格式的錯誤訊息(400)，而 EXIST 則是藉由 ifstream 來判斷檔案存在與否，好決定要使用 GET 還是 NOTFOUND。程式內的 SOCKET 連線皆藉由 Ws2_32.lib 來達成。

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include<WinSock2.h>
#include <fstream>
#pragma comment(lib, "Ws2_32.lib")
using namespace std;

int GET(SOCKET sConnection,char *filename)
{
    ifstream reader;
    char file[2048] = {0}; //用來放檔案內容
    char sendbuf[4096] = {0};
    reader.open(filename);
    reader.read(file, sizeof(file));
    reader.close();
    snprintf(sendbuf, sizeof(sendbuf), "HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n%s", file);
    //printf("Send buf to client (0x%x) \n", &sendbuf);
    int iResult;
    //----------------------
    // Send an initial buffer
    iResult = send(sConnection,sendbuf,(int)strlen(sendbuf),0);
    if (iResult == SOCKET_ERROR)
    {
        //terminate the program when send fail with error
        printf("send have failed with error :%d \n", WSAGetLastError());
        closesocket(sConnection);
        WSACleanup();
        return 1;
    }
    else
    {
        return 0;
    }
}

int BADREQUEST(SOCKET sConnection)
{
    const char *sendbuf = "HTTP/1.0 400 BAD REQUEST\r\nContent-Type: text/html\r\n\r\n<style>body{background: #ffffff;margin: 0;}</style>400 BAD REQUEST";
    //printf("Send buf to client (0x%x) \n", &sendbuf);
    int iResult;
    //----------------------
    // Send an initial buffer
    iResult = send(sConnection,sendbuf,(int)strlen(sendbuf),0);
    if (iResult == SOCKET_ERROR)
    {
        //terminate the program when send fail with error
        printf("send have failed with error :%d \n", WSAGetLastError());
        closesocket(sConnection);
        WSACleanup();
        return 1;
    }
    else
    {
        return 0;
    }
}
int NOTFOND(SOCKET sConnection)
{
    const char *sendbuf = "HTTP/1.0 404 NOT FOUND\r\nContent-Type: text/html\r\n\r\n<style>body{background: #ffffff;margin: 0;}</style>404 NOT FOUND";
    //printf("Send buf to client (0x%x) \n", &sendbuf);
    int iResult;
    //----------------------
    // Send an initial buffer
    iResult = send(sConnection,sendbuf,(int)strlen(sendbuf),0);
    if (iResult == SOCKET_ERROR)
    {
        //terminate the program when send fail with error
        printf("send have failed with error :%d \n", WSAGetLastError());
        closesocket(sConnection);
        WSACleanup();
        return 1;
    }
    else
    {
        return 0;
    }
}
int EXIST(SOCKET sConnection,char *filename)
{

    ifstream fin(filename);
    if(fin.fail ())
    {
        cout << "404 NOT FOUND" << endl;
        return NOTFOND(sConnection);
    }
    else
    {
        cout << "FILE EXIST" << endl;
        return GET(sConnection, filename);
    }
}
int main()
{
    char YN;
    int PORT_NUM = 80;
    //set default port number=80
    while(1)
    {
        cout << "Do you want to set port number?Y/N?";
        cin >> YN;
        if(YN=='Y')
        {
            cout << "Please input port number:";
            cin >> PORT_NUM;
            break;
        }
        else if(YN=='N')
        {
            break;
        }
        else
        {
            continue;
        }
    }
    WSADATA wsaData;
    WORD    DLLVersion;
    DLLVersion = MAKEWORD(2,1);//winsocket-dll version
    // 用 WSAStartup 開始 Winsocket-DLL
    int err = WSAStartup(DLLVersion,&wsaData);
    if (err!=0)
    {
        // Tell the user that we could not find a usable Winsock DLL.
        printf("WSAStartup failed with error: %d\n",err);
        return 1;
    }
    if (LOBYTE(wsaData.wVersion)!=2 || HIBYTE(wsaData.wVersion)!=1)
    {
        /* Tell the user that we could not find a usable */
        /* WinSock DLL.                                 */
        printf("Could not find a usable version of Winsock.dll\n");
        WSACleanup();
        return 1;
    }
    else
    {
        printf("The Winsock 2.1 dll was found okay\n");
    }
    SOCKADDR_IN    addr;
    int addrlen = sizeof(addr);

    // Create socket
    SOCKET sListen        ;        //listening for an incoming connection
    SOCKET sConnection    ;        //oerating if a connection was found
    SOCKET sRecv;//deal with recv

    // AF_INET        :  表示建立的 Socket 屬於 internet family
    // SOCK_STREAM    :  表示建立的 socket 是 connection-oriented socket
    sConnection = socket(AF_INET,SOCK_STREAM,NULL);

    // 設定位址資訊的資料
    addr.sin_addr.s_addr    = inet_addr("127.0.0.1");
    addr.sin_family            = AF_INET;
    addr.sin_port            = htons(PORT_NUM);


    // 設定 Listen
    sListen = socket(AF_INET,SOCK_STREAM,NULL);
    if (sListen == INVALID_SOCKET)
    {
        printf("socket function failed with error : %u \n",WSAGetLastError());
        WSACleanup();
        return 1;
    }
    // Bind the socket
    int iResult = bind (sListen,(SOCKADDR*)&addr,sizeof(addr));
    if (iResult == SOCKET_ERROR)
    {
        printf("Bind failed with error : %u \n", WSAGetLastError());
        closesocket(sListen);
        WSACleanup();
        return 1;
    }
    //SOMAXCONN: listening without any limit
    if(listen(sListen, SOMAXCONN) == SOCKET_ERROR)
    {
        printf("listen function failed with error: %d \n", WSAGetLastError());
        closesocket(sListen);
        WSACleanup();
        return 1;

    }
    else
        printf("Listening on socket...\n");


    // 等待連線
    SOCKADDR_IN clientAddr;
    while (1)
    {
        int result;
        char buffer[4096]={0};
        char filename[50] = {0};
        cout << "Waitting for connect... "<<endl;
        if(sConnection = accept(sListen,(SOCKADDR*)&clientAddr,&addrlen))
        {
            cout << "a connection was found."<<endl;
            sRecv=recv(sConnection, buffer, sizeof(buffer), 0);
            cout << buffer << endl;
            for(int i=5;i<4096;i++) {
                if(buffer[i] == ' ') {
                    break;
                }
                filename[i - 5] = buffer[i];
            }//分離出 filename
            cout << filename << endl;
            printf("Server : got a connection from : %s\n",inet_ntoa(addr.sin_addr));
            if (strncmp(buffer,"GET ",4)&&strncmp(buffer,"get ",4))//若request 不是格式正確的 get 則回傳 bad request
            {
                result = BADREQUEST(sConnection);
            }
            else
            {
                result = EXIST(sConnection, filename);
            }
            if(result==1)
            {
                return 1;
            }
        }
    }

}
```

參考資料:

本次使用的 SOCKET 參考自:

https://dangerlover9403.pixnet.net/blog/post/212391408-%5B%E6%95%99%E5%AD%B8%5Dc++-socket%E8%B3%87%E6%96%99%E6%95%B4%E7%90%86