108062224\_report

**Server:**

宣告message，專門記錄client寫下的message，把message[0]設成’\0’以便辨認message內是否有內容

int main(int argc, char\* argv[]) {

    SOCKET serverSocket, clientSocket; // create a socket

    struct sockaddr\_in serverAddress, clientAddress; // sockaddr\_in：IP4 格式使用 ,  sockaddr\_in6：IP6 格式使用 , sockaddr：通用格式

    int clientAddressLen;

    int bytesRead;

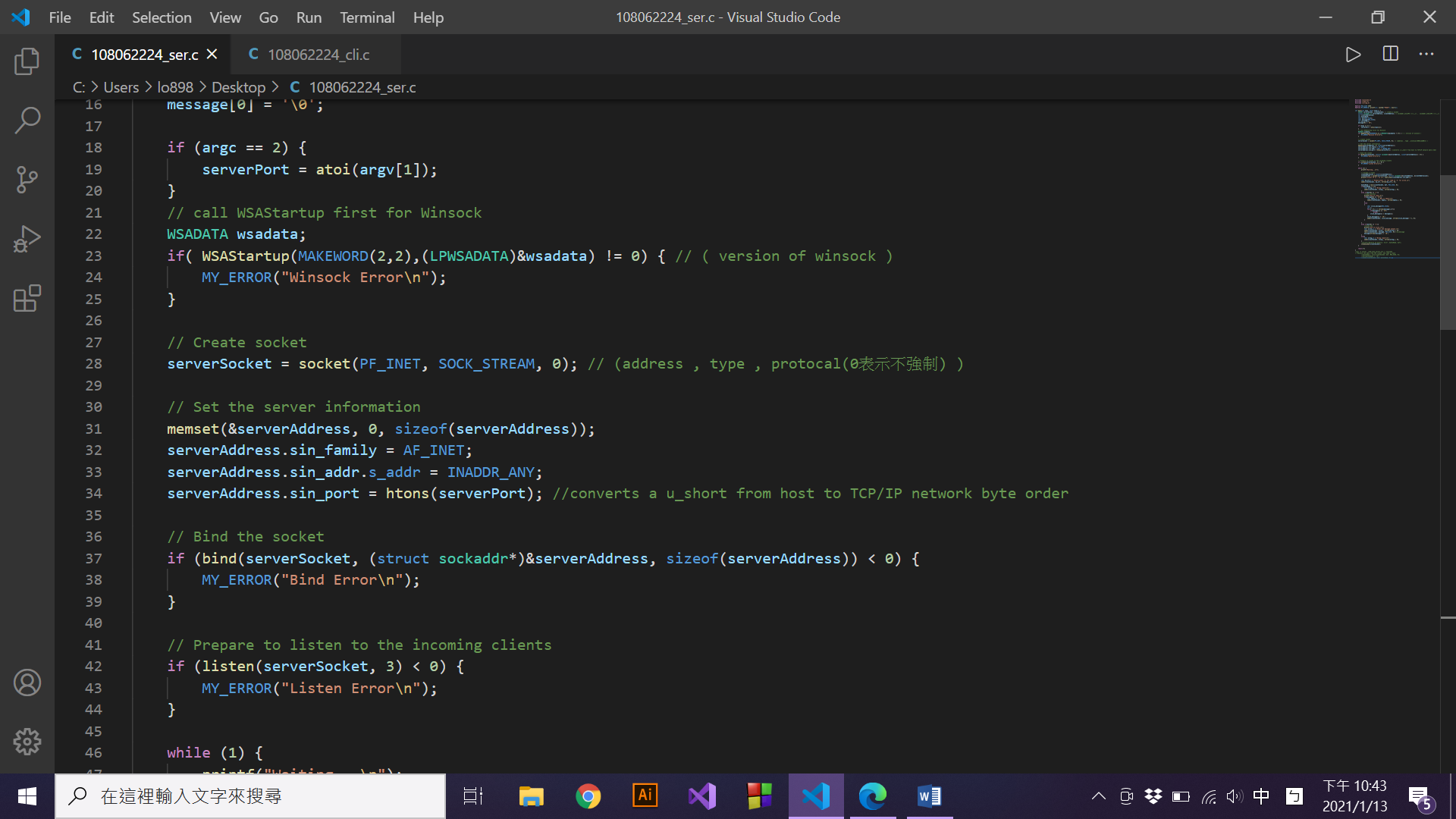
    char buf[MAX\_SIZE];

    char message[MAX\_SIZE];

    int serverPort;

    message[0] = '\0'; //can know whether is empty

一開始是sample寫好的create socket, set information, bind, listen等部分，



成功設定後進入while loop，這時候開始可以接受client connect，成功的話先讓server端印出client的ip

while (1) {

        printf("Waiting...\n");

        // Accept a client

        clientAddressLen = sizeof(clientAddress);

        clientSocket = accept(serverSocket, (struct sockaddr\*)&clientAddress, &clientAddressLen);

        printf("Client IP is : %s \n", inet\_ntoa(clientAddress.sin\_addr));

接著要提示client輸入指令，把字串用send給client接收，client傳過來的指令以recv接收，

char my\_str[] = "Please press '1' for read or '2' for write.\n";

    send(clientSocket, my\_str, strlen(my\_str), 0);

    bytesRead = recv(clientSocket, buf, MAX\_SIZE, 0);

如果指令長度不是1，非法，把錯誤訊息傳給client

if(bytesRead != 1){

            char wrong\_[] = "Wrong input!\n";

            send(clientSocket, wrong\_, strlen(wrong\_), 0);

}

如果client想要read輸入1，server端印出client read的動作，此外除了傳existing message給client，若目前沒有的話也傳目前是空白的提示給client

else if(buf[0] == '1'){

            //read previous

            printf("Ask to read.\n");

            if(message[0] == '\0'){

                char empty\_[] = "Now is empty!\n";

                send(clientSocket, empty\_, strlen(empty\_), 0);

            }

            else

            {

                char slice\_message[MAX\_SIZE];

                int i;

                for(i = 0; i < strlen(message);i++){

                    if(message[i] == '\0')

                        break;

                    slice\_message[i] = message[i];

                }

                slice\_message[i] = '\0';

                send(clientSocket, slice\_message, strlen(slice\_message) + 1, 0);

            }

        }

如果client想要寫新的內容，一樣印出動作再server端，再把提示訊息傳給client，接收new message

else if(buf[0] == '2'){

            //write new

            printf("Ask to write.\n");

            char write\_[] = "Enter new message please.\n";

            send(clientSocket, write\_, strlen(write\_), 0);

            recv(clientSocket, message, MAX\_SIZE, 0);//nw\_message

            message[strlen(message)] = '\0';

        }

如果client指令長度為一卻不是1, 2，返回錯誤

else{

            char wrong\_[] = "Wrong input!\n";

            send(clientSocket, wrong\_, strlen(wrong\_), 0);

        }

最後關閉這次client socket重新進入while loop

closesocket(clientSocket);

**client:**

client端一樣是sample裡面的client socket基本設定後，進入while loop 嘗試connect，成功會先印出server的ip(這部分沒改動sample，所以沒有放截圖)

接收server傳過來的press 1 or 2的提示訊息，放到buf並在client端印出

recv(serverSocket,buf, MAX\_SIZE, 0);//"press 1 or 2"

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

            if(buf[i] == '\n')

                break;

            printf("%c",buf[i]);

        }

        printf("\n");

client輸入的指令scanf到cmd再傳給server

//client enter '1' or '2'

        char cmd[MAX\_SIZE];

        scanf("%s", cmd);

        send(serverSocket, cmd, strlen(cmd), 0);

之後根據client是否還有訊息要傳送分兩種情況，第一種是client想寫新訊息cmd 2傳送給server後，server將會提示輸入，client接收後印出該提示，然後輸入新訊息內容，scanf完send給server，完成提示成功傳送

if( cmd[0] == '2' && strlen(cmd) == 1){

            recv(serverSocket,buf, MAX\_SIZE, 0);//"write"

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

                if(buf[i] == '\n')

                    break;

                printf("%c",buf[i]);

            }

            printf("\n");

            char nw\_messsage[MAX\_SIZE];

            scanf("%s",nw\_messsage);

            nw\_messsage[strlen(nw\_messsage)] = '\0';

            send(serverSocket, nw\_messsage, strlen(nw\_messsage)+1, 0);

            printf("New message is sent!\n");

        }

第二種情況是server做一次回應即可，像是cmd屬於wrong input和想要 read，client recv後印出

else{

            recv(serverSocket,buf, MAX\_SIZE, 0);//"wrong input","empty" or existing message

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

                if(buf[i] == '\0' || buf[i] == '\n')//end message,input or end empty\_str

                    break;

                printf("%c",buf[i]);

            }

            printf("\n");

        }

最後一樣斷掉連接，重新進入while loop

closesocket(serverSocket);

**遇到的問題:**

主要有兩個，一個是在vscode上用terminal跑要輸入什麼，gcc -o 檔名後要再加上-lwsock32，compile完後執行exe的部分server 加port, client則是ip(自己本機測試用127.0.0.1)跟port(跟server同一個port)才會成功。這部分嘗試很久也詢問很多人，最後拼拼湊湊出可以執行的指令。

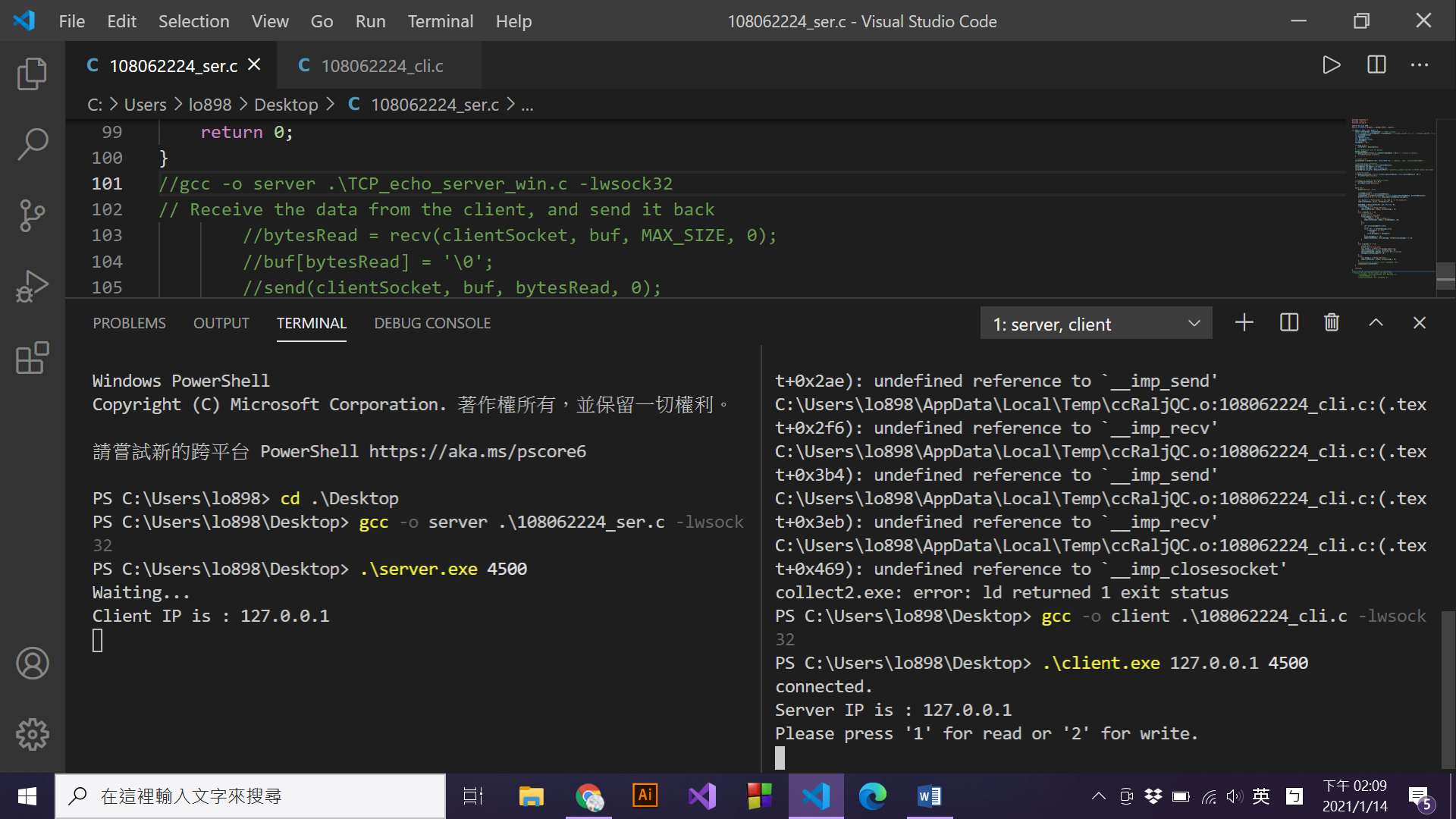
第二個困難點是我要的訊息常常會遮蔽失敗，例如server傳”Now is empty”的時候，client會印出” Now is empty’1’ for read ‘2’ for write”，要求read existing message如”test”時印出”testse press’1’ for read ‘2’ for write”這樣的情況。

因原本只是讓client 接收訊息到buf內直接printf(“%s”,buf)， 所以印出整條內容，我修改的方法是在每次新到buf內容的後面都加入’\0’(或者是’\n’)，然後相應的send內長度是原本strlen()+1，加上’\0’，印出的時候就可以印到必要訊息就好。

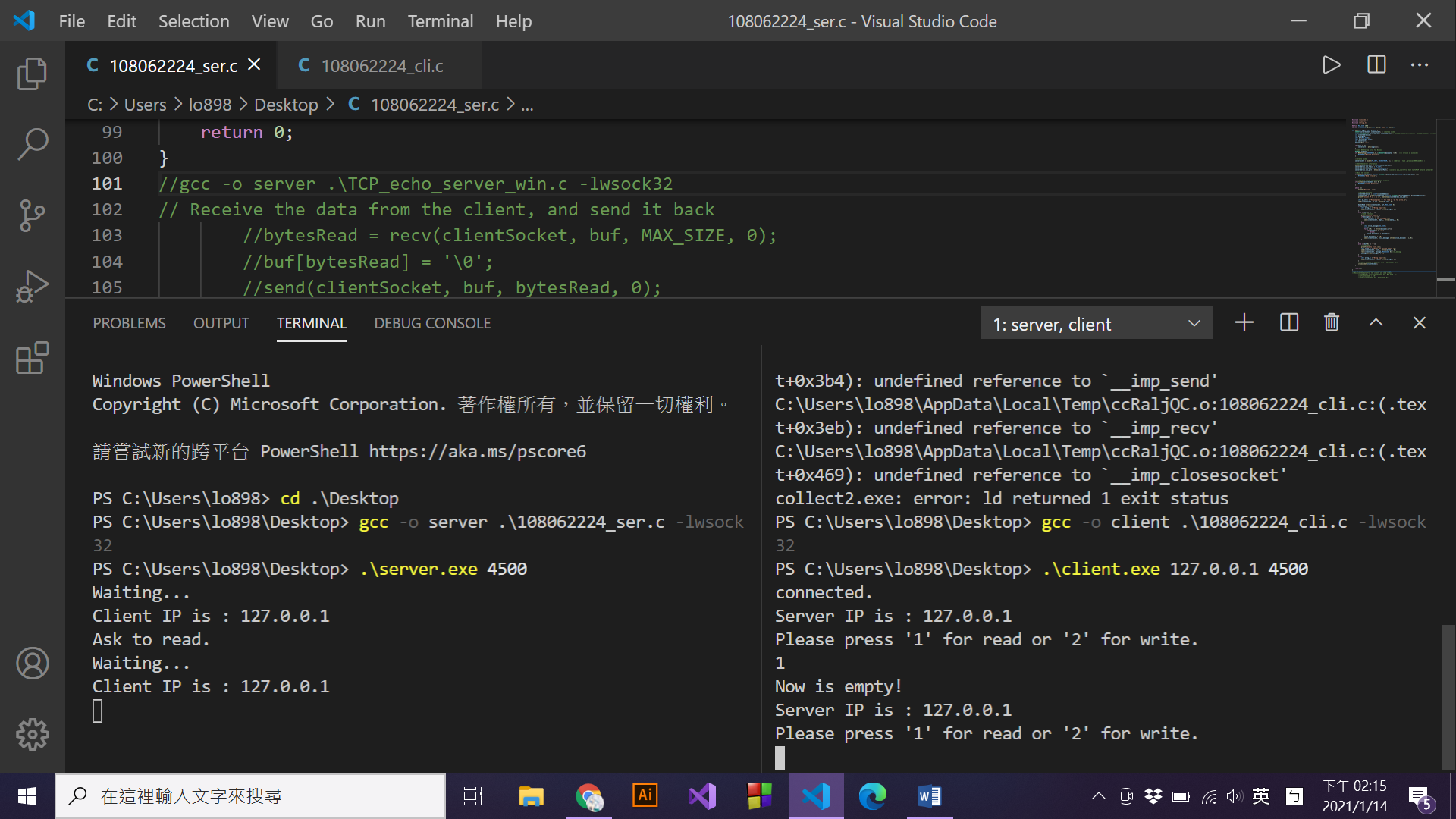
**實際操作畫面:**

左邊是server右邊是client，右邊上半部出現的錯誤訊息是由於沒加-lwsock32

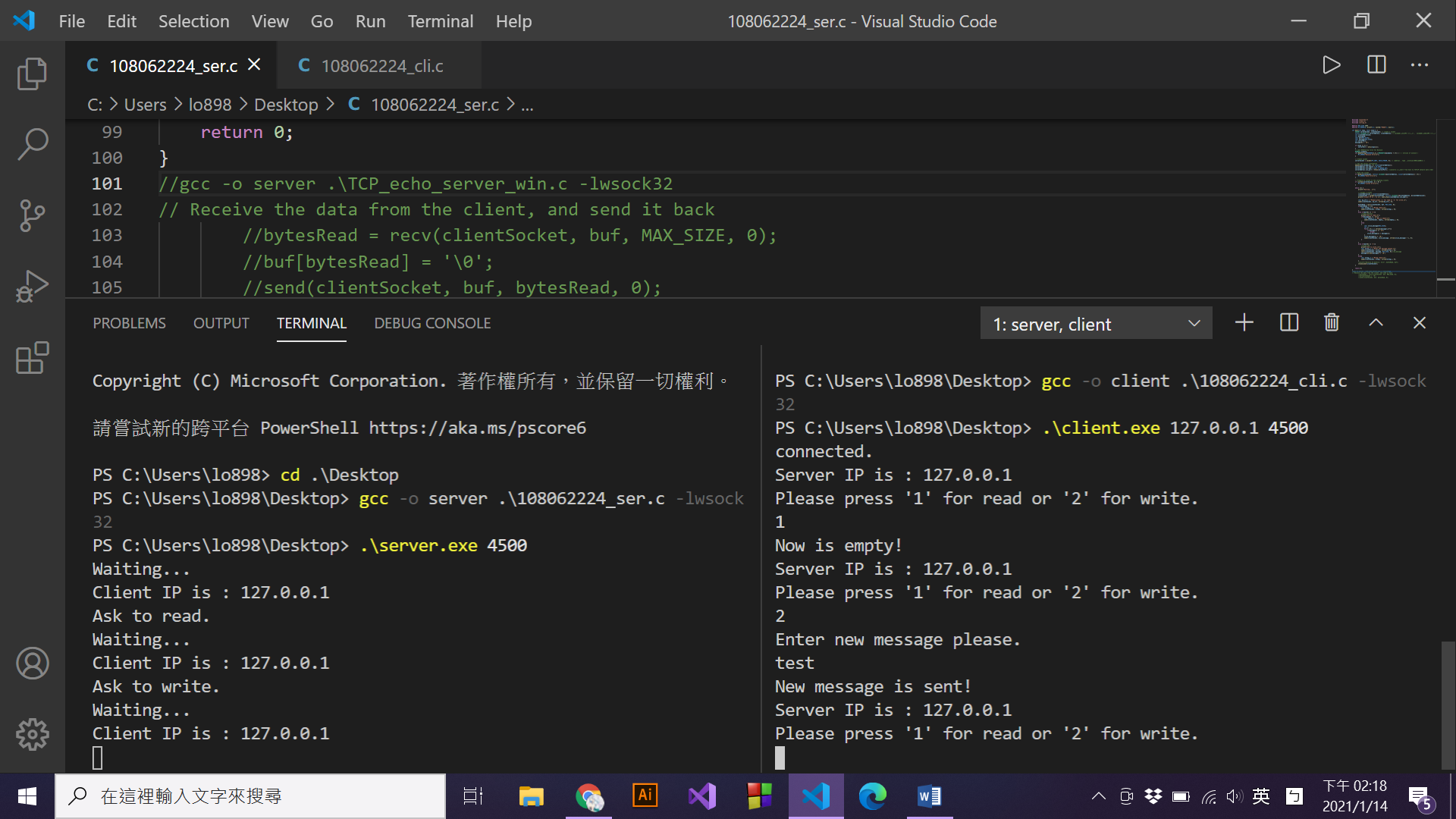
兩邊都compile完，輸入ip跟相同port(4500)後，server給”Please…”提示，client收到後印出



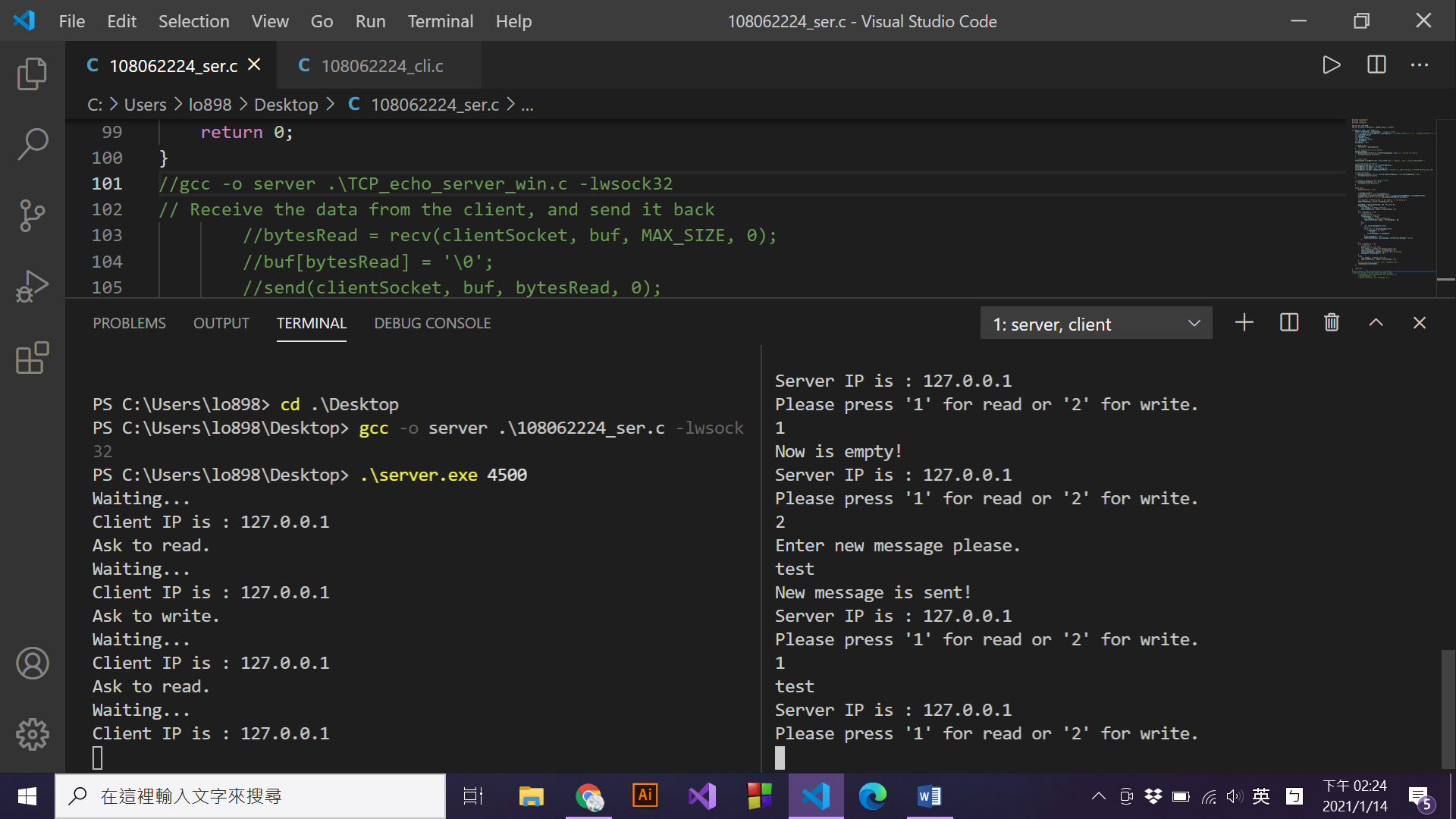
此時client若按1，server端印出request，因為目前尚無任何message所以對於request，server回應是”Now…”提示，client收到後印出，結束此次連結



新一次連接，Client按2，server端印出 quest並傳送提示訊息”Enter…”讓client印出，client端輸入的訊息傳給server紀錄，同時顯示傳送完畢。



之後要讀取之前訊息:按1之後，server把先前紀錄傳給client印出



若是輸入非法指令，server傳送錯誤訊息給client印出

