程式設計期末project-----socket programing

組員:資管一賴柏霖,林奇宥,徐紹軒,張凱庭

Q1:為何我們會想做這個:

Ans: 我們原本是想做校園地圖,可是後來要做的時候發現其他組已經有人寫過了,我們就決定換一個主題,後來我們就想作去年宿營casino的記分板,但是我們發現我們不會兩台電腦的傳輸,所以我們問小傑要怎麼連接兩台電腦,就走了上socket這條路.

Q2:我們在做的過程中遇到了甚麼困難:

ANS: 我們一開始就遇到了檔案無法執行的問題,因為網路上查到很多的SOCKET CODE編譯的環境都是在LINUX系統,所以我們找了很久才找到在WINDOW上編譯的CODE,而且DEV C++裡面的編譯環境也要多打一個參數才能編譯,後來我們建立了1台電腦對一台電腦的一對一連線,才想到我們應該要能多對一才對,於是我們又重新上網找資料,把原本的CODE刪掉,多用了THREAD來寫多對一的連線,至於怎麼做的,下面說.

SOCKET簡介:

從網路的角度來看，socket就是通訊連結的端點；從程式設計者的角度來看，socket提供了一個良好互動的介面，使程式設計者不用知道下層網路協定運作的細節便可以撰寫網路通訊程式。

例如我們現在寫得在windows上的socket事實作於Berkeley Software Distribution(BSD, release 4.3)中的UNIX sockets為基礎所發展出來的一套API,而我們這次是用他其中的TCP,但他也額外支援很多其他的格式,也提供電機碼的相容性, 因此在不同的作業系統上移植時並不需要做任何修改。

在TCP/或ip的架構下,socket可分為下面兩種傳輸方法

1. Datagram sockets:

資料在datagram sockets間是利用UDP封包傳送，因此接收端socket可能會收到次序錯誤的資料,且部分資料可能會遺失,所我們這組採用下面的方法.

1. Stream sockets:

資料在 stream socket間是利用 TCP的方式傳送, 因此接收端socket可以收到順序無誤、無重覆、正確的資料。此外TCP傳送時是採資料流的方式，因在傳送時會所有資料會視情況被分割在數個TCP封包中。

接下來是我們主程式架構的部分,上網查之後,發現所有網路應用程式皆可分為5個步驟:

1. 開啟一個socket
2. 為socket命名
3. 與另外一個socket結合
4. 在socket間傳送資料
5. 關閉socket

舉例來說

SERVER端的流程 Client 端的流程

I\_open() socket()

↓ ↓

↓ ↓

Bind() ↓

↓ ↓

↓ ↓

Listen() ↓

↓ ↓

↓ ↓

Accept()並block住等待clinet端的 連線請求 -----> connect()

連線請求 🡨連線建立 ↓

↓ ↓

↓ ↓

Read() < --------------- write()

↓ 資料傳送 ↓

↓ ↓

Write() -----------------------------> read()

↓ 資料傳送 ↓

↓ ↓

Close() close()

1. 我們開啟socket方法很簡單,只要呼叫socket()函式就好.socket是通訊網路的端點,就像介面卡,所以一個主機可以有很多socket.而用戶端跟伺服器端都要有一個socket才能存取資料.
2. 接下來要為socket命名,為socket命名需要設定三個變數

分別為(協定,連結埠號碼,位址),而用戶端就要用這些來與伺服器端連結,所以server端要先呼叫 bind()函式來設定好本身的連接埠號碼與ip位址.

1. 接下來要看如何將socket連結, 首先我們要先最少在client端跟 server端都開了一個socket,並且為server端的socket命名,接下來伺服器準備收封包,用戶端準備傳封包,這過程就叫結合,winsockapI提供了幾個函式,server端要開listen(),收到連線要求後,就會開一個新的socket,用accept()跟他連結,而用戶端則是呼叫connect()來連結.
2. 接下來我們在socket間傳送資料,如果兩個socket上,我們可以用recv() 跟 send() 來傳送資料,然而我們有查到如果沒有連接兩個socket的話,也可呼叫recvfrom()與sendto(),雖然這個我們不會用也不知道為甚麼QQ.
3. 接下來就是關閉 socket,其實這步很簡單,就是呼叫 closesocket(),關閉後可以將資源還給協定堆疊.

再來我們面臨的就是多人同時傳送資料給主機,因為我們現在的code都是執行緒一次往下讀,所以我們可以呼叫thread函式庫,來引用thread.每當呼叫一個thread,他就會新開一個背景執行緒,重頭往下跑,所以這樣就可以完成多人同時對一個server端傳送的工作.

後記: 我們原本只打算做賭場記分板的工作,不過後來因為還有時間,我們就又寫了blackjack這個遊戲,遊戲本身不算難,就是用class player這樣來分玩家,並把檔案存在server端,後來再把每個步驟按照順序丟在上面建立好的socket系統中,我們就寫好了多人連線的blackjack.

心得:

因為我們本來的東西規劃好的想說蠻簡單的,所以打算到後來在做,沒想到後來別組先做出來了,所以我們在最後一周才開始趕工,不過因為卡到很多期末考,尤其是微積分,讓我們寫起來備感艱辛,這裡要特別感謝其他三位組員大力的carry我,雖然我門是四個人一起研究socket,但是我到這周末才學會socket跟thread怎麼用 ,真的是感謝xd.