各位同學好

今天要出第二次作業

我們要利用作業1的TCP socket連線來實作一個群組聊天窗

這是outline

前面3個是上次作業有說過的socket講解，可以回去自己看內容都一樣。

我們跳到第4個Thread的部分。

我們先用這張圖稍微帶一下這次作業大概要做什麼

有很多的client跟一個server

其中一個client發送訊息，server接收後轉送廣播給其他的client

讓其他client可以收到A發出的訊息

也就是說server要可以同時和不同client溝通

我們介紹2種可以讓server跟不同client溝通的方法

第一個是使用thread

同一個process的每個thread會共用memory空間

也就是thread可以存取全域變數

每個thread可以執行不同的函式

每個thread有自己的stack可以存區域變數

cpu會輪流執行每個thread

每個thread可以執行不同的函式

也就是說我們可以在主程式做server-client連線，由thread去幫我們針對各個client做接收訊息的動作，因為cpu會由各個thread輪流執行，所以可以接收到各個client的訊息。

接下來介紹thread會用的API

thread的function會以pthread底線開頭，要使用這些function必須include pthread.h這個標頭檔。在編譯的時候後面要加上負pthread以連接到pthread library。

第一個介紹的是pthread create這個function

它的功用就是建立一個thread，我們直接看到他帶的參數，

第一個是指向這個新thread的id的指標

第二個是thread的屬性，通常設NULL。

第三個是放這個thread要執行的function，這個利字是他要去執行PrintHello這個function。

第四個是這個PrintHello function需要帶入的參數

接下來介紹pthread\_join這個function

它的功用是等待某個特定的thread結束

那第一個參數帶的就是這個thread的id

第二個參數是用來存前面這個thread結束後的回傳值，如果沒有設回傳值的話這裡就設NULL。

接下來是pthread\_exit這個function

用來結束thread

帶的參數就是結束的回傳值

如果沒有要設回傳值的話這裡就設NULL。

接下來看到這是一個thread的範例，

先看到main的地方，一開始宣告一個thread id的變數，接下來看到第24行，我們create一個thread叫他去做threadFunc的事情並傳送一個參數過去

再來看到第6行threadFunc接收參數，執行func裡的事情

接著看到第17行，呼叫thread\_exit並回傳了一個剛剛輸入的參數。

再來我們回到main function看到地29行

這裡main呼叫了一個pthread\_join來等待thread完成並接收回傳值。

再接下去執行。輸出Return value

這個就是剛剛那個程式的執行結果，可以看到Return value在threadFunc的輸出之後，也就是利用thread\_join主程式會等thread執行完之後再繼續執行。

那注意在編譯的時候要加負pthread。

接下來介紹第二個讓server和多個client連線的方法，select。

先介紹一下Blocking IO跟Multiplexing IO

Blocking IO是一般的system call會使用的方法。

先看到左上角，當process 呼叫recv這個system call之後，會進到kernel，kernel現在如果沒有看到任何datagram ready，他就會繼續等，一直等到kernel收到datagram之後才會做copy的動作把datagram回傳給user。在這中間process是被block住的，沒有辦法做其他的事。

那Multiplexing IO是有使用select這個system call的方法，可以讓單一一個process多工化。

先看到左上的部分，一開始由process呼叫select這個system call，給select一段限制時間，由select去檢查目前有沒有datagram ready，那select的多工的部分在於它可以一次監控多個socket，只要其中有人datagram ready的話，select就會回傳readable。這時process再呼叫recv的system call，去針對ready的socket做接收的動作。

select允許我們在所有POSIX兼容平台下做I / O Multiplexing。

select 會讓process指定kernal去等待某些事情的發生，在下面得兩種情況下會wake up process

1. 當一個或多個指定事件發生時
2. time out

我們使用select去告訴kernal

第一個，那些fd以及那些情況是我們有興趣的，想要去監控他的。

第二個，我們可以等多久時間