微算機系統實習

實驗六

日期:2021.06.14

* 組別與組員名單:

第<16>組

資工二 108590451陸詠涵

資工二 108830035陳佳均

* 實驗步驟截圖與說明:

(1)<說明>

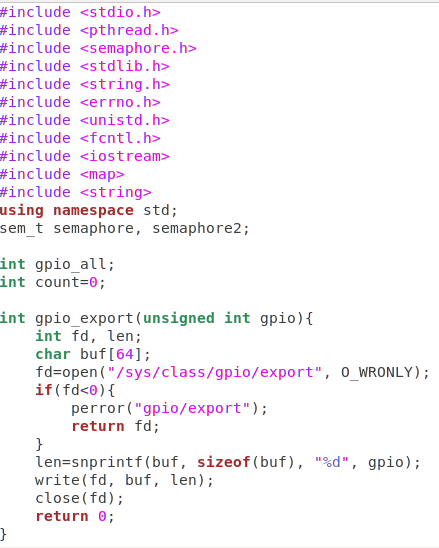
實驗目的: 學習MUTEX 及 SEMAPHORE 兩種多執行緒的觀念，然後實際撰寫程式與tx2連接 實作LED燈閃爍的功能

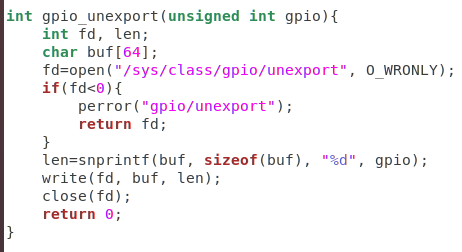
項目一:實作SEMAPHORE

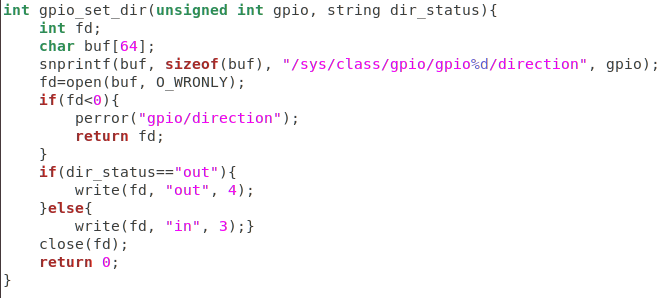
指令:

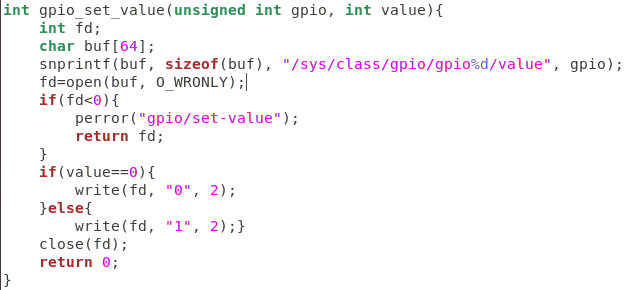
1. gedit lab6\_sema\_v03.cpp
2. g++ lab6\_sema\_v03.cpp -lpthread -o lab6\_sema\_v03
3. sudo ./ lab6\_sema\_v03 0001 4

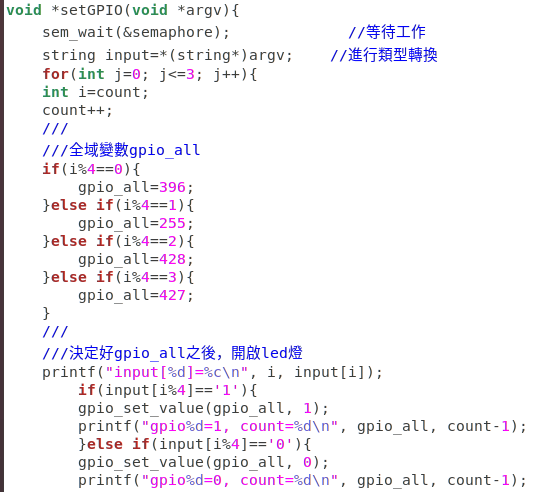
(picture 1) SEMEPHORE的程式lab6\_sema\_v03.cpp













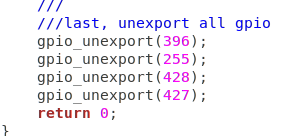
Semaphore觀念

1. 為同步物件，用於保持在0至指定最大值之間的一個計數值
2. 提供2種操作，signal()增加計數值和wait()減少計數值
3. 當執行緒執行wait()時，計數值

-1並進入至critical section；當thread執行完critical section後，接續執行signal()將計數值+1







項目二:實作MUTEX

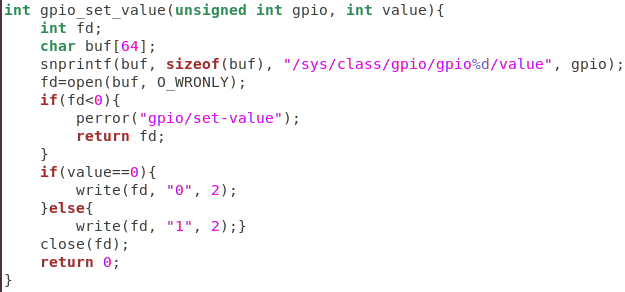
指令:

1. gedit lab6\_sema\_v02.cpp
2. g++ lab6\_ mutex\_v02.cpp -lpthread -o lab6\_ mutex\_v02
3. sudo ./ lab6\_mutex\_v02 0001 4

(picture 2) MUTEX的程式 lab6\_mutex\_v02.cpp









Mutex觀念

1. 為了保護資料完整性，防止兩條thread同時對同一個公共資源(ex:全域變數)進行讀寫的機制
2. 當一個thread對global variable進行r/w時，此時上鎖mutex， 其餘thread若要r/w the same global variable便會被阻塞，直到原thread完成r/w操作並將mutex解鎖後，其餘thread才可以store global variable







* execution results(項目一,項目二)影片中呈現

<https://youtu.be/kN41ksorS6w>

* 組員貢獻比例

陸詠涵: 50% 負責最後實驗數據驗證與連接虛擬機

陳佳均: 50% 負責程式碼撰寫

* 心得:

陸詠涵:

這次mutex和semaphore的實作，我是負責連接實作led燈的那一方。在一開始，我們因為指標的問題，所以程式碼一直沒有通過編譯，耗了一大功夫之後，程式碼編譯過了，但是led燈並沒有照著指令亮對燈。所以在semaphore的程式中，我的夥伴在程式中加入counter來實際看看執行到哪一個led燈了，後來慢慢試之後，終於成功亮燈了，但是，我們明明led燈有在最後set\_value = 0但是，第四個燈有時候無法在最後關起來。遠距所花的時間真的比當場做還多出太多太多了，真的很感謝我的夥伴，讓我們能順利完成這個實作。

陳佳均:

這次的實驗看似簡單，實則不然。並且深深地體會到了遠距的不便。先講實驗的部分。一開始就覺得只要在程式裡共用全域變數的地方使用mutex或等待訊號就行，結果一直遇到問題。譬如有訊號被跳過啊、有燈泡沒亮等等之類的問題。後來是把程式精簡到一個迴圈就一個訊號才把問題給解決。另外還有變數傳遞的問題，函數因為只有一個變數格，所以只能傳遞一個。尤其他變數的格式是之前沒接觸過的，費了好大的勁才終於可以傳遞變數。至於遠距的不便，因為我跟同伴相隔甚遠，又因為我是屬於從測試中找出問題的類型，所以我們在這次的實驗不斷地互傳訊息。我改一個地方，就要傳給她測試，她測完又要截圖給我結果。就這樣一來一回，令我不禁想唱"時間都去哪了"。實在是辛苦我的同伴了~