

作業系統程式作業 3

編譯方式：“g++ -o TYSIM5 1061506_03.cpp -lpthread”

程式執行：“./TYSIM5 (TA 數量:1-2) (教授能力啟用: 0 或 1)”

Code 由作業 2 修改而成，使用到了 pthread 的 mutex 來保持各個 thread 之間的同步。

Line 26~57: struct my_sem_t

運用三個 mutex 組合而成的 semaphore，有了這個在某種程度上提供了不少的方便性。

Line 61~71: struct student_data_t 和 pt_data_t

用來保存傳給各個 thread 資料的 struct。

Line 85~93: function simespec operator-

方便計算時間用的輔助 function。

Line 95~97: function rnd

生成給定區間內亂數的輔助 function。

Line 101~125: struct initializer

用來進行初始化的 struct，在建構式裡面初始化了所有的 mutex 還有一些資料，與此同時解構式會幫忙 destroy 使用到的 mutex 以免自己忘記加。

Line 127~164: function main

初始化資料、建立 thread、join thread。

Line 172~177: function clock_now

計算從起始時間開始後已經過了多少毫秒的輔助 function。

Line 179~245: function student_behavior

定義 student thread 該作甚麼的 function。基本上就是照著規則做事。

Line 247~256: function discuss_with_student

讓 TA 或是教授和同學討論的輔助 function

Line 258~320: function Prof_behavior 和 TA_behavior

非常相似的兩個 function，分別定義了教授和助教的行為。雖然架構差不多但是差在一些小地方的行為不同所以拆成兩個 function 寫。曾經想整合起來卻因為沒時間而作罷。

基本上寫這份作業到後來就會聯想到網路通訊，會有這種感覺的產生是因為教授的雙核心能力，TA 和教授就是 server 在 listen 他們的 port 等待事件發生，而

學生則是客戶端。因此他們每個 **thread** 都持有的 **semaphore** 取名叫 **ack** 也是源自於 **TCP** 中的那個 **ack** 雖然後來發現好像不太一樣就是了。