

CS305 作業系統概論 Prog. #3 Process Coordination

2019.05.01

一、作業目的

在本作業中，利用pthread的方式來練習process coordination的觀念。

二、作業內容

【忙碌的TY教授】TY教授要在海豚書院開設新課，對海豚書院的學生而言，這可是能夠親自向TY教授請益的千載難逢機會。可是對於大批蜂擁而至的學生，TY教授實在不勝負荷，因為平日要討論的研究工作已經相當繁重，可是對於這些可愛的同學，TY教授也實在不忍拒絕。因此TY教授特別每星期找出一天來讓50位同學問問題。為了讓學生能夠盡快的進入情況，TY教授會先請學生找海豚書院中學識淵博的洋助教將問題預先討論一下，準備好問題，然後再與TY教授進一步討論。

因此在TY教授的討論室外，有5張椅子讓要詢問問題的同學先坐著等候。同時為了有良好秩序，不能有站著等椅子的同學。如果討論室裡沒有同學與洋助教討論，坐在椅子上的同學就依照FCFS的順序，進入討論室與洋助教討論10~30 milliseconds，然後再與TY教授討論50~100 milliseconds完成這次的請益（放心，不會有starvation）。如果都沒有同學來，TY教授就會與洋助教來一場LOL。如果討論室外面的5張椅子都坐滿了同學，此時新到的同學，就需要先離開30~50 milliseconds去找海豚書院中年輕有為的興助教去看一集快轉的「與餓距離」專輯，然後再回來看有沒有空位。如果還沒有空位，就需要再離開30~50 milliseconds去找興助教去看另一集快轉的「與餓距離」專輯。在與洋助教討論的同學如果討論完畢，卻發現TY教授仍然在與另一位同學討論，他就需要先坐回等候的椅子（仍然是FCFS），以便空出位子給下一同學來問洋助教。但如果此時剛好有新來的同學，新來的同學不能搶先排在這位等候同學之前。如果所有等候的6位同學都問完洋助教的問題，TY教授仍然沒空，此時最先來的同學就坐回與洋助教討論的位子一直等候TY教授，不再離開，但此時洋助教就可以休息。所有曾經和助教討論過的同學，如果再次回到與助教討論的位子，就都不需要離開位子，一直等候TY教授。

請設計一個程式模擬這個TY教授討論室裡的運作。模擬時，要輸出每個人狀態改變的時間，例如洋助教是什麼時候休息、打LOL或討論，TY教授是什麼時候休息、打LOL或討論，學生何時到達與離開等資訊，以及各種時間的長度，例如討論的時間長度、打LOL的時間長度、或是學生中途離開去看專輯的時間長度等資訊。時間的計算，是從0ms開始計算起，TY教授與洋助教一開始的時候都在休息。並假設同學到達的間隔時間是5~10 milliseconds。

因此你的程式可能會輸出

```
0 ms -- Prof. TY: rest
0 ms -- TA Y: rest
6 ms -- Student 01: enter
6 ms -- TA Y: discuss with Student 01 30 ms
11 ms -- Student 02: enter & wait
16 ms -- Student 03: enter & wait
...
36 ms -- Prof. TY: discuss with Student 01 100 ms
...
136 ms -- Student 01: leave
```

一、作業要點

1. 請注意，本作業使用的程式語言是C/C++，測試平台的作業系統：Ubuntu 18.10 64-bit。使用的編譯程式為gcc/g++ 編譯器：8.2。其他平台或程式語言不在本次作業考慮範圍之內。如在測試平台上無法編譯與執行，都不予給分。
2. **注意！**在程式中，每個時間長度，一定要以亂數函式rand(0)設定亂數序列，用rand()來取得亂數，亂數的取法如下，例如如果是取10~20的亂數，用(rand()%11)+10取得。

3. 在程式中，milliseconds的時間控制，可用usleep()函式來完成。（unistd.h中的 int usleep(useconds_t usec);）
4. 請注意，本作業一定要用pthread API來進行。任何不用pthread API的程式，都不予給分。
5. 本作業的評分方式如下：
 - a. 同步機制只能用 pthread mutex 機制（以“pthread_”開頭的 mutex函式）來完成 critical section 的方式，例如pthread_mutex_init()。本次作業不可使用POSIX Semaphore機制，例如sem_init()。Thread之間的訊息傳送，也不可經由檔案的方式或其他shared memory 等IPC 機制傳遞，一定要用 thread 之間共用的資料變數。不符規定者，本作業不予給分。
 - b. 本作業必須以multithreaded的方式來完成，每個人物分別用一個working thread（也就是Child thread）來模擬（包含TY教授、洋助教、及50位同學共52個threads。與助教可以不考慮。）。能夠create出這些thread，最多可得到基本30分。
 - c. 完成上述基本要求，且正確產生結果者，視正確執行的程度，最多可得50分。
 - d. 完成以上基本功能者，才可按照以下項目，多得其他的分數。但請注意，**如何demo出你程式中的這些進階功能，必須在你的說明文件檔案中詳細說明，如果助教看不懂，可能反而會扣分。**
 - i. 為了加速助教速度，再找成助教來幫忙洋助教。討論的時間也是10~30 milliseconds。助教的數目必須是命令列的參數。程式必須在一開始的時候就需輸入現在是一位或兩位助教。本項目最多可得到基本30分。
 - ii. 為了加快TY教授與學生討論的速度，TY教授利用「雙重核心」的能力，達成TY教授好像在同時回答2位同學問題的情況。每位同學都討論50~100 milliseconds。「雙重核心」的能力也必須是命令列的參數。程式必須在一開始的時候就需輸入現在TY教授是否發揮「雙重核心」功能做到multitasking 2位同學的特異效果。本項目最多可得到基本30分。
6. 本作業需繳交檔案：
 - a. 說明報告：檔案為docx或pdf格式。
 - i. 報告中必須說明程式的設計理念、程式如何編譯，以及**如何操作**。
 - ii. 報告中同時必須詳細說明你完成哪些部份。如有用到特殊程式庫，請務必說明。
 - iii. 請務必讓助教明白如何編譯及測試你的程式。助教如果無法編譯或測試，會寄信（**最多兩次**）通知你來說明，但每說明一次，**助教會少給你10分**。
 - b. 完整原始程式碼檔案（.c 或 .cpp）。**不可含執行檔。助教會重新編譯你們的程式。請注意：**也不可用.txt檔或是.docx檔等非正常方式繳交程式碼，如有類似情形，**助教會扣10分**。
 - c. **不可以含有病毒，如果含有病毒等惡意程式，本作業0分。**
7. 所有相關檔案，例如報告檔、程式檔、參考資料等，請壓縮成一個壓縮檔（不可超過2MB）後上傳至portal。**請注意，不可抄襲。助教不會區分何者為原始版本，被判定抄襲或雷同者，一律0分。**

二、繳交方式：

1. 最終繳交時間：
 - a. 程式作業檔在 2019.05.31 以前，上傳至個人portal。如有多個檔案，必須將所有檔案壓縮成一個zip（rar 亦可）檔案，然後上傳。
 - b. 上傳檔名格式：「學號_作業號碼.rar」或「學號_作業號碼.docx」。例如：912233_01.rar 或 912233_01.docx。
2. **如有違規事項者，依照課程規定處理。**
3. 如需請假，請上portal請假，並持相關證明文件，在請假結束後的第一次上課時完成請假手續，並在一週內完成補交。補交作業將以8折計算。
4. 老師不接受「門縫」方式繳交，助教也不接受任何作業。

三、如有未盡事宜，將在個人portal板面公告通知。

四、 If you need **any assistance in English**, please contact Prof. Yang.

五、 參考資料

1. 課本第6章
2. “How to Use C Mutex Lock Examples for Linux Thread Synchronization”,
<http://www.thegeekstuff.com/2012/05/c-mutex-examples/?refcom>
3. “pthread : mutex”, <http://angelonotes.blogspot.tw/2012/02/pthread-mutex.html>