

CS305 作業系統概論 Prog. #3 Synchronization

2021.04.30

一、作業目的

在本作業中，利用pthread的方式來熟悉 synchronization 的觀念。

二、作業內容

【疫苗啊疫苗！】風之塔學院資工系透過了大數據的計算，研發出非常有效的疫苗，並且有自動注射機器人來進行注射。但是這個疫苗需要注射三次，才能發揮最好的效果。

為了同學的健康，即日起，每天開始可以為同學施打疫苗。但因為疫苗產量有限，每天最多只能有20位同學進行疫苗施打。在注射機器人的辦公室裡，會有一張椅子讓同學坐著施打疫苗。辦公室外面的走廊上有三把椅子，可以讓同學在那裡坐著等待。如果注射機器人目前正在為一名學生施打疫苗，其他到達的學生就坐在辦公室外面的椅子等待。如果在辦公時間內都沒有學生需要施打疫苗，注射機器人就暫時休眠節省電力。當學生在辦公時間內到達並發現注射機器人正在睡覺，就喚醒注射機器人以施打疫苗。如果某個學生到達時，發現走廊上的椅子也都坐滿，就先離開一下，稍後再回來排隊。

現在有10~20位同學，要來施打疫苗。因此為了要事先了解施打疫苗的情況，你要設計一個程式，利用 pthreads（不含main thread）來模擬這個注射機器人以及這 n 位學生互動的情況。注射機器人以及 n 位學生都個別用單獨的執行緒來運作，每個執行緒都只能代表一個人。以下是模擬的條件：

1. n 是從命令列讀入一個10~20的整數。並且讀入一個 0~100的整數亂數種子。
2. 每個同學第一次施打疫苗之前要等候一段0-10秒的時間才去排隊。這段時間是由亂數決定。注射完疫苗後，下一次會有10-30秒的時間間隔再去排隊施打疫苗，這個時間間隔也是亂數決定。
3. 機器人每次為學生施打疫苗的時間是 2 秒。
4. 如果沒有位子可以坐著等候，學生離開5-10秒鐘再回來重新排隊。由亂數決定。
5. 在走廊排隊等候的，採用 FCFS 的規則，不可以插隊。
6. 每位學生都施打完三次疫苗後以後，就結束當天的疫苗施打，程式結束。
7. 機器人與每個學生都必須各別用一個thread來模擬。學生的編號從1開始。

在程式進行中，請將機器人與學生的狀態都要詳細輸出，以便檢查。同時要印出時間(Wall-clock time)，至少是以秒為單位。例如假設程式開始執行模擬的時間是09:10:00，有20位同學，下面是可能的輸出：

```
> prog3 20 0
09:10:00 Robot: Sleep
09:10:06 Student01: Entering to get a shot
09:10:06 Student03: Sitting #1
09:10:06 Student04: Sitting #2
09:10:08 Student01: Leaving to write YZU CS305 programs
09:10:08 Student03: Entering to get a shot
...
09:30:06 Robot: All 20 students receive vaccines.
```

三、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 20.04 64-bit。使用的編譯程式為gcc/g++ 編譯器： 9.3。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. 請注意，本作業必須要用pthread API中的 mutex機制來進行，例如pthread_mutex_init()。不能使用 semaphore機制，例如sem_init()。**任何不用pthread API mutex機制的作業，本作業只能最多得到60分。**

3. 本作業的評分方式如下：

- a. 從命令列讀入所有整數並能夠處理命令列輸入的各種錯誤。本項滿分10分。
- b. 亂數的產生必須設定亂數種子。可以正確使用亂數相關函式，並且亂數種子只在main thread中設定一次。本項滿分10分。
- c. 正確產生模擬每個人物的 $n+1$ 個 thread。本項滿分20分。
- d. 正確使用pthread API 中的 mutex 機制形成critical section來處理進入機器人辦公室的情況。本項滿分20分。如果不使用此機制者，此部分得 0分。
- e. 正確使用pthread API 中的 mutex 機制形成critical section來處理辦公室走廊上等待的情況。本項滿分20分。如果不使用此機制者，此部分得 0分。
- f. 正確執行 FCFS 的規則。本項滿分10分。
- g. 正確由main thread 執行 join 讓程式結束。本項滿分10分。

4. 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：**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。**
5. 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，被判定抄襲者，一律0分。

四、 繳交方式：

1. 最終繳交時間：

- a. 電子檔在 2021.05.31以前，上傳至個人portal。如有多個檔案，將所有檔案壓縮成zip（rar 亦可）格式，然後上傳。
- b. 上傳檔名格式：「學號_作業號碼.doc」或「學號_作業號碼.rar」。例如：912233_01.doc 或 912233_01.rar。

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。

五、 如有未盡事宜，將在個人portal板面公告通知。

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

七、 參考資料

1. 課本第6-7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>