

# CS305 作業系統概論 Prog. #2 Multithreading

2017.04.24

## 一、作業目的

熟悉如何使用Pthreads的API，撰寫multithreaded program。

## 二、作業內容

【基大師的排序分析】風之塔學院的勇達人覺得高徒的能力應該不僅於此，因此他決定用多執行緒的方式來開發一個可以平行執行各個排序方法程式，來比較各個排序方法的效能。

這個程式要能從命令列讀入一個資料輸入檔名以及一個work thread 檢查啟動變數的頻率  $n$  ( $1 \leq n \leq 5$ ) 秒。該檔案中有10000個1到10000的不同整數隨機放置，每個整數之間用一個空格隔開。程式執行時，就會先產生三個work thread，分別測試 bubble sort, merge sort 以及quick sort。在產生work thread之前，main thread 還不會讀入資料檔，在產生work thread 之後，main thread才讀入資料檔，然後透過共用變數傳給三個work thread 來執行排序。Main thread 與 work thread 之間的同步，用另一個共用的啟動變數來達成。如果該啟動變數為0，表示main thread 還沒有準備好資料。如果該啟動變數為1，表示排序資料已經放置完畢，work thread 可以開始執行。work thread 啟動以後，每 $n$ 秒去檢查這個啟動變數。如果檢查的結果是0，就先沉睡 $n$ 秒（使用sleep()）然後再檢查。

這三個work thread會紀錄排序所花費的時間，結束後，將(1) 回傳時間以及(2) 排序時間傳回給main thread，然後將排序結果輸出到檔案中，作為檢查。main thread取得所有work thread的排序時間結果後，由 main thread 按照時間結果，由小至大印出這三個work thread的成績。如果時間相同，則按照回傳時間的順序印出。如果回傳時間相同，以bubble sort, merge sort 以及quick sort的順序印出。

在程式執行中，有下面幾項要求：

1. main thread產生work thread 的時候，要印出每個work thread的tid。
2. main thread讀入檔案時，也要表明自己的動作。
3. 在執行緒溝通的過程中，所有溝通的動作也都需要輸出，印出該執行緒的tid及動作。
4. 在work thread 運作時，狀態改變的時候，要印出改變的時間，改變時間以年/月/日/時/分/秒來表示。
5. work thread傳回時間時，要印出來它的(1) 回傳時間及(2) 排序時間。回傳時間以年/月/日/時/分/秒來表示，排序時間以 millisecond 為單位。
6. work thread傳回時間時後，要輸出排序結果到各自的文字檔案中。（例如：bubblesort.txt, mergesort.txt等等）
7. main thread在印出結果時，也要印出 “[Main thread]”。

以下的範例中，假設在程式執行時，work thread 的 tid分別為100, 101, 102，程式輸出如下

> prog2 data 5

[Main thread]: work thread 100 for bubblesort

[Main thread]: work thread 101 for mergesort

[Main thread]: work thread 102 for quicksort

[TID=100]: [2017.05.01 17:50:00] sleep 5 seconds

[TID=101]: [2017.05.01 17:50:00] sleep 5 seconds

[TID=102]: [2017.05.01 17:50:00] sleep 5 seconds

[Main thread]: read data

[Main thread]: write data to the shared area

[TID=101]: [2017.05.01 17:50:05] mergesort begins

[TID=102]: [2017.05.01 17:50:05] quicksort begins

...

[TID=100]: [2017.05.01 17:50:30] bubblesort ends

[TID=100]: [2017.05.01 17:50:30] bubblesort 20000 ms  
[TID=100]: [2017.05.01 17:50:31] bubblesort results written to file bubblesote.txt  
[Main thread]: the execution time rank  
mergesort: 15000 ms  
quicksort: 16000 ms  
bubblesort: 20000 ms  
>

### 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 16.10 64-bit。使用的編譯程式為g++ 編譯器：5.4。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，本作業一定要用Pthread API來進行。任何不用Pthread API的程式，都不予給分。
3. 注意：印出訊息的重點是要能讓助教清楚判斷程式執行過程，如助教無法清楚了解執行過程，只能酌量給分。
4. 注意：所有計算方式，不可有cheating行為。如果發現，本作業0分。
5. 本作業的評分方式如下：
  - a. 基本功能：依照下面項目的完成程度來給分，如果只能完成部份，將部份給分。
    - i. main thread 可以正確從命令列讀入檔名，並讀取資料。本項滿分10分。
    - ii. main thread 能正確傳遞檢查啟動變數的  $n$  給work thread。本項滿分10分。
    - iii. main thread 能用pthread\_create() 順利產生work thread。本項滿分20分。
    - iv. main thread 能正確啟動 work thread。本項滿分10分。
    - v. work thread 成功以平行方式計算三個排序演算法得到正確結果。每個排序功能滿分10分，本項滿分30分。。如以work thread以循序方式計算三個排序演算法，本項最多給12分。
    - vi. work thread 能正確傳遞結果給main thread，main thread可以正確印出結果。本項滿分20分。
  - b. 進階功能：完成以上基本功能者，才可實作以下項目來得分。
    - i. main thread 產生第4個work thread來計算進階版mergesort (advmerge)。此work thread會再產生2個它的work thread分別處理前後5000個資料，然後再彙整起來作最後的merge sort。當2個它的子work thread排序完畢時，要分別輸出它們的 tid及狀態。本項滿分20分。
    - ii. main thread 產生第5個work thread來計算進階版quicksort (advquick)。此work thread會再產生2個它的子work thread分別處理第一次 pivot element 決定位置後的左右兩邊數列，然後再彙整起來作最後的結果。當2個它的work thread排序完畢時，要分別輸出它們的 tid及狀態。本項滿分20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。不可含執行檔。助教會重新編譯你們的程式。
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，被判定抄襲者，一律0分。

#### 四、繳交方式：

##### 1. 最終繳交時間：

a. 電子檔在 **2017.05.12** 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. 參考課本圖 4.9。

2. PThread: <https://computing.llnl.gov/tutorials/pthreads/>

3. POSIX 線程 (pthread) 入門文章分享：<http://dragonspring.pixnet.net/blog/post/32963482-posix%E7%B7%9A%E7%A8%8B%28pthread%29%E5%85%A5%E9%96%80%E6%96%87%E7%AB%A0%E5%88%86%E4%BA%AB>