

# CS305 作業系統概論 Prog. #1 Proc. Generation & Communication

2017.03.20

## 一、作業目的

熟悉如何利用fork()系統呼叫產生新的process，以及process彼此之間如何使用POSIX shared memory互相傳遞資料。

## 二、作業內容

【基大師的排序分析】風之塔學院的基大師教學多年，經驗豐富。尤其在排序方法上，更有獨到教學之處。然而他有鑑於課程中若只有講授虛擬碼，往往仍有隔靴搔癢之感。因此他特別要找風之塔學院的勇達人帶著他的高徒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。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fork產生三個子行程，分別測試 bubble sort, merge sort 以及quick sort。在fork之前，parent process 還不會讀入資料檔，在fork 之後，parent process才讀入資料檔，然後透過 POSIX shared memory 傳給三個children processes 來執行排序。這三個children processes會紀錄排序所花費的時間，結束後，將時間結果傳回給 main process，最後由 main process 按照時間結果，由小至大印出這三個children processes的成績。如果時間相同，則按照傳回的順序印出。

在程式執行中，有下面幾項要求：

1. 父行程產生child process 的時候，要印出每個子行程的pid，以及自己的 pid。
2. 父行程讀入檔案時，也要表明自己的動作。
3. 父行程透過POSIX shared memory與子行程溝通資料時，所有的行程（父行程與子行程）都要用同一塊POSIX shared memory。
4. 在行程溝通的過程中，所有溝通的動作也都需要輸出，印出該行程的pid及動作。
5. 子行程傳回時間時，要印出來它的執行時間，以 millisecond 為單位。
6. 子行程傳回時間時後，最後在結束前，要輸出排序結果到文字檔案中。（例如：bubblesort.txt, mergesort.txt等等）
7. 父行程在印出結果時，也要印出自己的pid。

以下是個範例。假設在程式執行中，若父行程 pid為 2370，子行程 pid分別為2380, 2390, 2400，程式進行的輸出如下

```
> prog1 data
[2370 Parent]: fork 2380 for bubblesort
[2370 Parent]: fork 2390 for mergesort
[2370 Parent]: fork 2400 for quicksort
[2370 Parent]: read data
[2370 Parent]: write data to shared memory
[2390 Child]: mergesort begins
[2400 Child]: quicksort begins
...
[2380 Child]: bubblesort ends
[2380 Child]: bubblesort 20000 ms
[2380 Child]: bubblesort results written to file bubblesote.txt
[2370 Parent]: 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，本作業一定要用的機制為fork()與POSIX shared memory。任何沒有使用這些機制來完成的程式，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. 父行程能用fork()至少順利產生一個子行程。本項滿分20分。
    - ii. 父行程能用 POSIX shared memory來傳遞資料給子行程。本項滿分10分。
    - iii. 子行程成功以平行方式計算三個排序演算法。每個排序功能滿分10分，本項滿分30分。。如以子行程以循序方式計算三個排序演算法，本項最多給12分。
    - iv. 子行程能用同一塊 POSIX shared memory來傳遞結果給父行程。本項滿分20分。
    - v. 父行程可以正確從命令列讀入檔名，並讀取資料。本項滿分10分。
    - vi. 父行程可以正確使用getpid()，wait()函式，而不會產生zombie。本項滿分10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. 父行程產生第4個子行程來計算進階版mergesort (advmerge)。此子行程會再產生2個它的子行程分別處理前後5000個資料，然後再彙整起來作最後的merge sort。當2個它的子行程排序完畢時，要分別輸出它們的 pid以及個別的排序結果。所有的資料傳遞使用另一個不同的shared memory進行。本項滿分20分。
    - ii. 父行程產生第5個子行程來計算進階版quicksort (advquick)。此子行程會再產生2個它的子行程分別處理第一次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個它的子行程排序完畢時，要分別輸出它們的 pid以及個別的排序結果。所有的資料傳遞也使用另一個與前面都不同的shared memory進行。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 2017.04.13 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3.17與3.18。

2. 參考課本programming problem 3.14, 3.15。

3. 參考上課講解之範例程式。