

## 作業系統程式作業 2

編譯方式：“g++ -o prog2\_1506 1061506\_02.cpp -std=c++17 -lpthread”

程式執行：“./prog2\_1506 (filename)”

```
s1061506@s1061506-VirtualBox:~/桌面$ ./prog2_1506 data
[Main thread]: create TID:140554289137408, DocID:0001
[Main thread]: create TID:140554280744704, DocID:0002
[Main thread]: create TID:140554272352000, DocID:0003
[Main thread]: create TID:140554263959296, DocID:0004
[TID=140554263959296] DocID:0004 [0,1,1,1,0,1,0]
[TID=140554280744704] DocID:0002 [1,0,0,1,1,0,1]
[TID=140554289137408] DocID:0001 [1,1,0,1,0,0,1]
[TID=140554272352000] DocID:0003 [2,2,1,1,0,0,0]
[TID=140554272352000] cosine(0003,0001)=0.790569
[TID=140554272352000] cosine(0003,0002)=0.474342
[TID=140554272352000] cosine(0003,0004)=0.632456
[TID=140554272352000] Avg_cosine: 0.632456
[TID=140554272352000] CPU time: 0ms
[TID=140554263959296] cosine(0004,0001)=0.500000
[TID=140554263959296] cosine(0004,0002)=0.250000
[TID=140554263959296] cosine(0004,0003)=0.632456
[TID=140554263959296] Avg_cosine: 0.460819
[TID=140554263959296] CPU time: 0ms
[TID=140554280744704] cosine(0002,0001)=0.750000
[TID=140554280744704] cosine(0002,0003)=0.474342
[TID=140554289137408] cosine(0001,0002)=0.750000
[TID=140554289137408] cosine(0001,0003)=0.790569
[TID=140554289137408] cosine(0001,0004)=0.500000
[TID=140554289137408] Avg_cosine: 0.680190
[TID=140554289137408] CPU time: 0ms
[TID=140554280744704] cosine(0002,0004)=0.250000
[TID=140554280744704] Avg_cosine: 0.491447
[TID=140554280744704] CPU time: 0ms
[Main thread] KeyDocID:0001 Highest Average Cosine: 0.680190
[Main thread] CPU time: 1ms
```

Code 由老師提供的 thread-ex1.cpp 修改而成，使用到了 pthread 的 mutex 來保持各個 thread 之間的同步。

Line 29~39: struct thdata

用來保存傳給各個 thread 資料的 struct，main function 後來判斷哪份文件是關鍵文件也是靠它還有它的比較運算子來判斷。

Line 49~67: function init

用來進行初始化的 function，在裡面初始化了所有的 mutex，讀入了所有文件的資料並同時計算有幾份文件要處理。

Line 69~77: function create\_threads

幫助創建剛好數量 thread 的 function。

Line 79~101: main function

- line 80~83: 檢查輸入的參數是否正確。
- line 89~90: 等待所有的 thread 計算完自己的平均餘弦相似係數。
- line 93~94: 尋找關鍵文件並印出。
- line 96~98: 印出 Main thread 使用的 cpu time。

line 103~163: function article\_analyzer

用來讓每一條 thread 執行的任務。

- line 109~114: 分析文件並找出自己的詞頻(不完整)。
- line 116~122: 每個 thread 將自己出現過的字統一更新到 Basis 裡面。
- line 126~127: 從 Basis 中尋找，將沒出現在自己文章內的字加入詞頻向量並將出現次數設為 0。
- line 128~142: 輸出自己處理的文件以及詞頻向量，並同時計算有多少 thread 完成了計算以及自身的向量長度。只有當所有 thread 都完成計算後，才去計算兩兩的相似度。
- line 146~155: 計算對其他文件的餘弦相似係數並輸出，同時也在計算對所有其他文件的平均餘弦相似係數。
- line 156~159: 輸出平均餘弦相似係數以及花費的 cpu time。