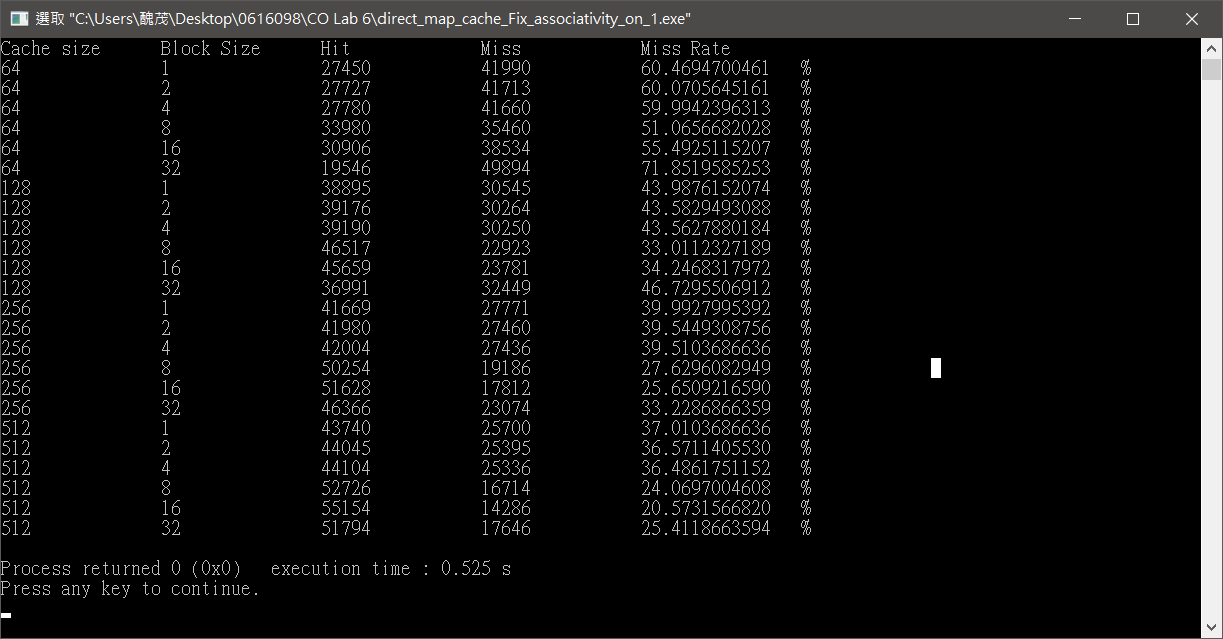
**Computer Organization Lab 6**

0616098 黃秉茂

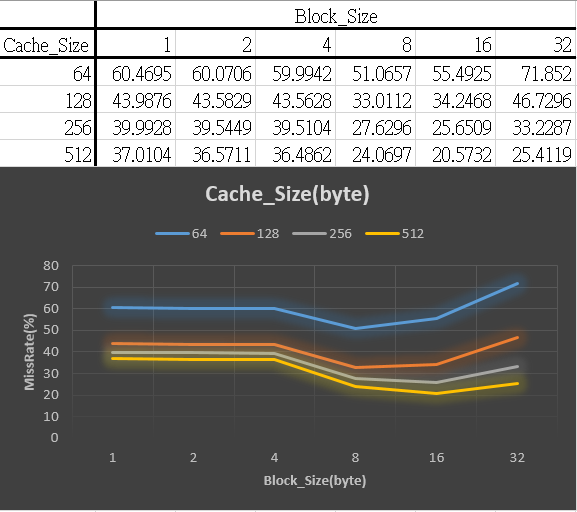
**Result:**

1. direct\_map\_cache\_Fix\_associativity\_on\_1.cpp (使用的資料為Trace.txt)

更改direct\_map\_cache的路徑，多include其他函式庫，跑個雙層for-loop和處理輸出部分就完成了。



以不同的 Cache Size 和 Block Size 觀察 Miss Rate 的變化，結果如下圖：



從結果中可以發現到，在相同 Block Size 下， Cache Size 從 64 Bytes 到 512 Bytes時，因為可用的 Lines 增加，所以不容易發生重複，因此 Miss Rate 顯著下降。

而在相同 Cache Size 下，加大 Block Size 可以提升 spatial locality 效率，進而降低 Miss Rate。在Block Size由1到4時下降的極為緩慢，結果並不顯著；4變成8時下降的效果最為明顯，但是在16到32時出現不同結果，Miss Rate 反而大幅增加，推測是因為Block Size已經大幅超越 spatial locality的區域，反而會因為 block size 增加，line減少，導致 index 容易重複，產生 replace。

1. direct\_mapped\_cache\_lru\_Fix\_block\_size\_on\_32B.cpp

(使用的資料為Trace.txt)

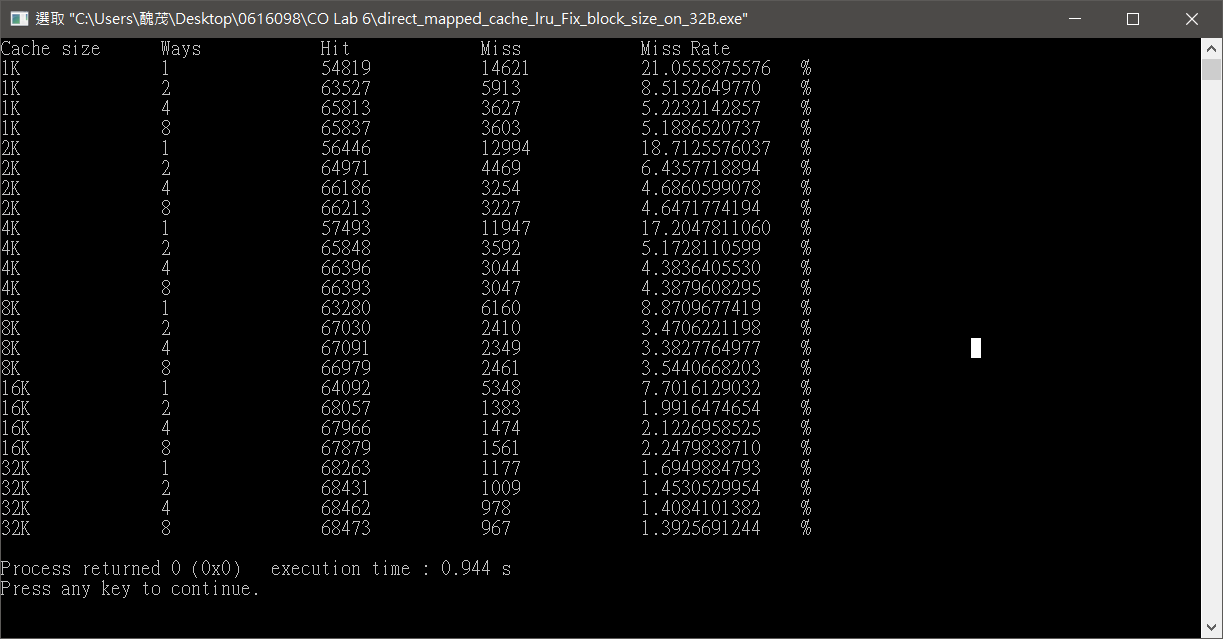
1. N-way set associative： 將原本的 Cache size 切分為 N 份，因此 set 總共有 Cache size/N 個，由 index bits 決定哪一個 set，並且每個 set 又各自給予 N 個 cache content，總共大小仍為 Cache size。

2. LRU (Least-Recent Used)： 在每個 Cache content 中儲存 count 值，共 log2(N)個 bits，用來記錄該content在同一set中的使用新舊順序，並按照以下方法進行：

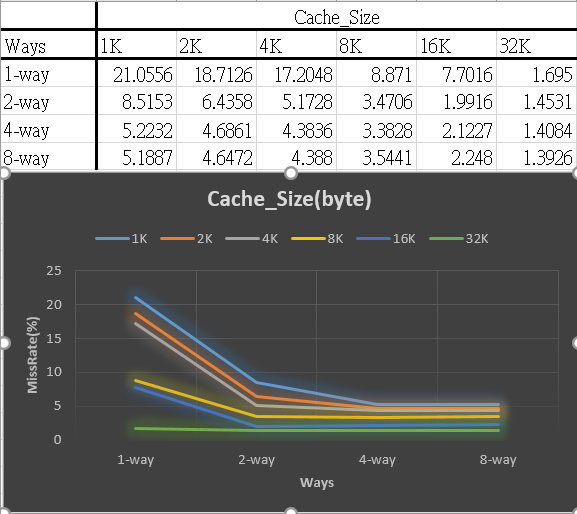
a. 一開始每個 set 中的 N 個 content 各自寫入 0, 1, 2, …, N-1 的值，0為最舊， N-1 為最新。

b. 若搜尋時沒有找到目標，則會將 count為0的 content 進行replace，同時 count 更新為 N-1，表示replace後的資料變成最新的，並將相同 set 其餘content 的 count 都減一。

c. 若搜尋時找到目標，則將該目標的 count 改為 N-1 表示其為最新，其餘 content 的 count 若大於目標原本 count，則 count 減一，否則 count 不變，如此一來目標順序被提升至最新，中間的content也會被往下推一個順位。



以不同的 Cache Size 和 Associativity 觀察 Miss Rate 的變化，結果如下圖：



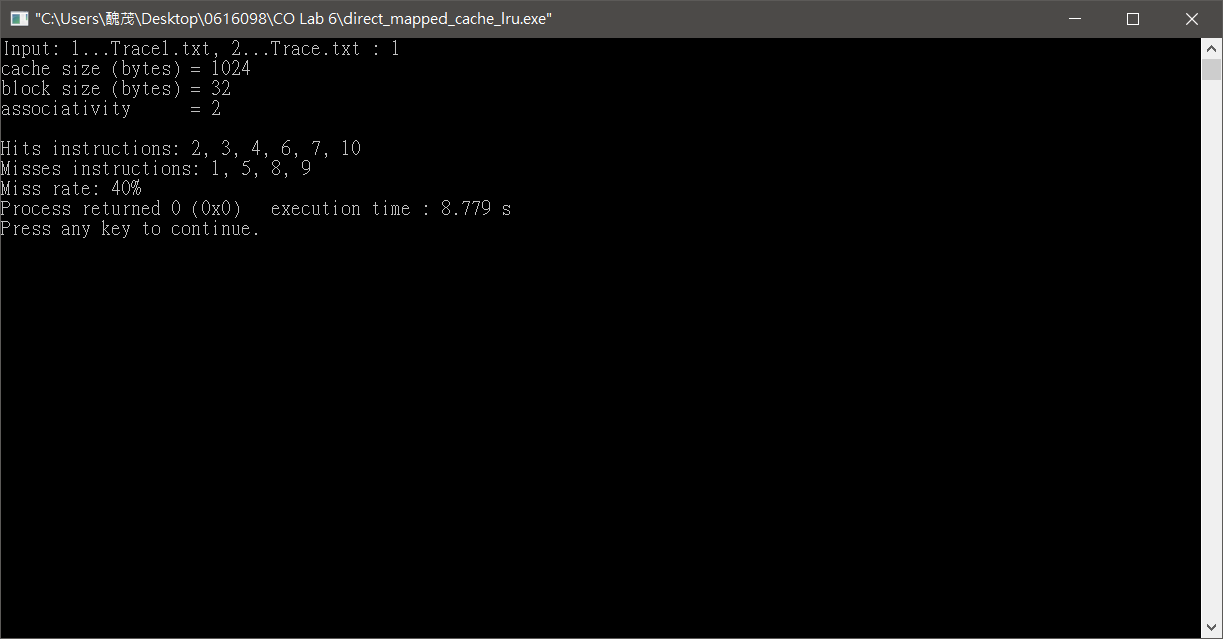
從結果可以看到，Ways和Cache size增加都會顯著降低 Miss Rate，Cache size的影響算是很明顯，尤其是Associativity很低時，因為可用的 Lines 增加，所以不容易發生重複，因此 Miss Rate 顯著下降。

而Associativity造成的改變則是在1-way變成2-way時最為顯著，推測原因為其spatial locality特徵明顯，容易重複 index，因此使用更多的ways可以避免頻繁 replace 導致的 miss，並且利用LRU安排replace次序，進一步降低 miss rate。

1. direct\_mapped\_cache\_lru.cpp

改變direct\_mapped\_cache\_lru\_Fix\_block\_size\_on\_32B.cpp，讓block\_size

成為變數，並用vector紀錄instruction就結束了。



**Problems you met and solutions:**

最困難的在於LRU要如呵紀律使用頻率，後來特別在cache content多了cnt來記錄。而因為set內能存指一筆資料所以cache要用二維的方式儲存，後來用double pointer跟 2D-array的方式完成。比較難的大概就是處理這些問題和理解cache的實際操作。

**Summary:**

我更了解了cache的操作，也了解到當有spatial locality的特性時，cache是多麼的重要，也知道Associativity能夠造成的影響，這個lab讓我更理解cache的細節和實作，很多地方尤其是LRU更需要一些巧思。