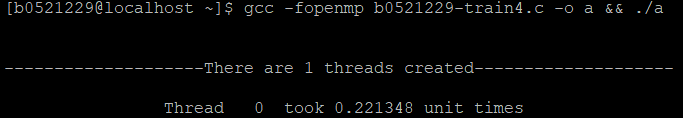
**B0521229 林威廷**

**平行程式設計-Training4**

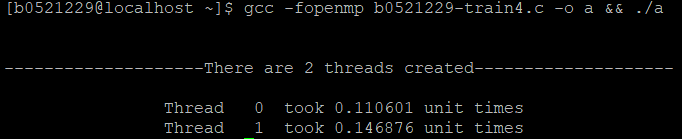
**Implementation of Results**

1 Thread (Base)



Time:0.221348 (Base)

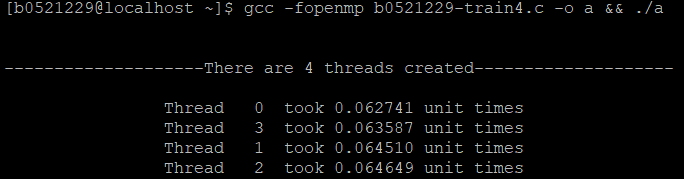
2 Threads:



Time:0.146876

Speedup factor=0.221348/0.146876=1.507

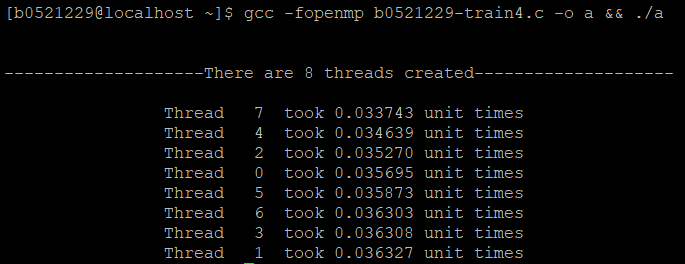
4 Threads:



Time:0.064649

Speedup factor=0.221348/0.064649=3.424

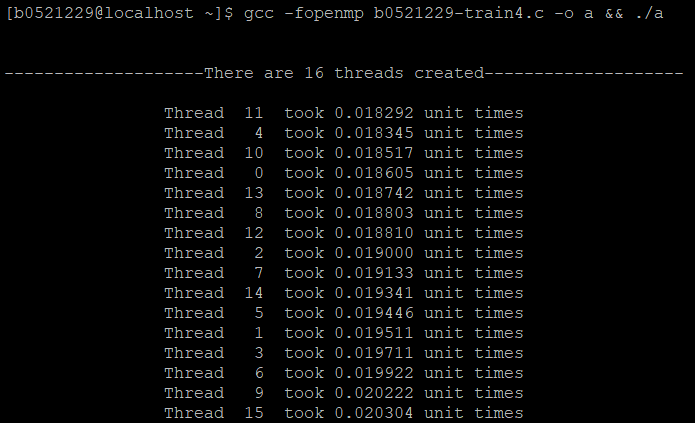
8 Threads:



Time:0.036327

Speedup factor=0.221348/0.036327=6.093

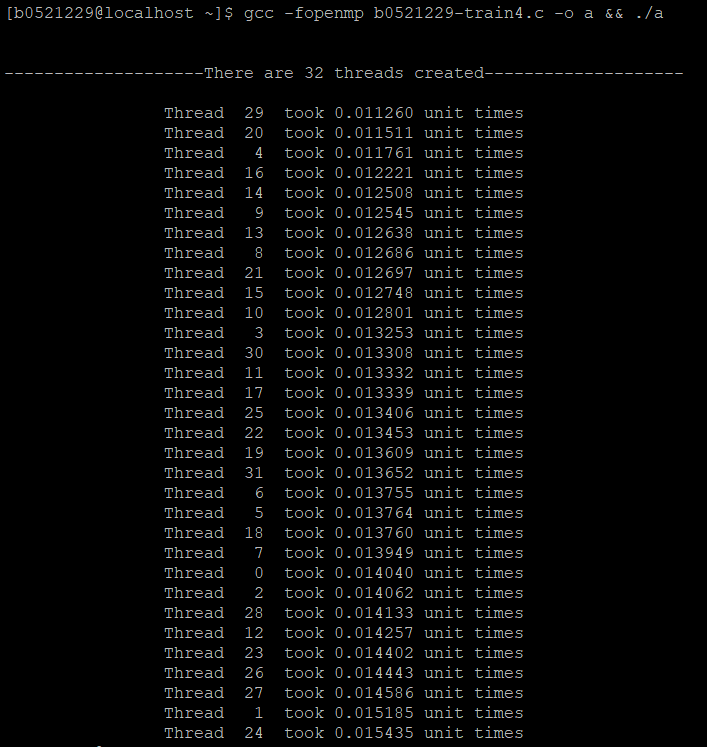
16 Threads:



Time:0.020304

Speedup factor=0.221348/0.020304=10.901

32 Threads:



Time:0.015435

Speedup factor=0.221348/0.015435=14.340

**心得**

本次作業的實作，為了便捷，如要改變需要的threads數量，僅需在程式碼中更改Function:omp\_set\_num\_threads(int threads) 裡面的參數即可，整體來說OpenMP這種寫法對共享記憶體進行操作容易有同步問題，必須加入一些操作來確保資料共享變數結果是正確的，像本次我就使用Date Scope Attribute Clauses裡REDUCTION的設定來滿足存取互斥，並最後加總；可惜的是這次是產生隨機數字，以至於最後面的結果不能得知是否加總正確，不過我有加入註解，用簡單的數字去測試，是沒有問題的。