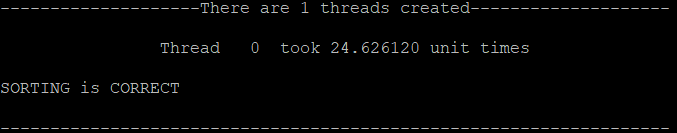
**B0521229 林威廷**

**平行程式設計-Training6**

**Results of Implementation**

**(Compare to MPI)**

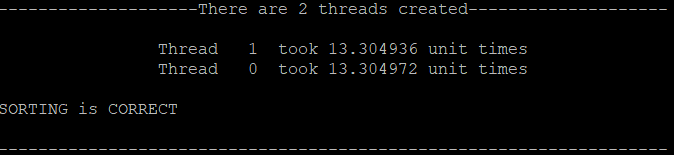
1 Thread (Base)



OMP Time:24.6261 (Base)

MPI Time:26.2480 (Base)

2 Threads:



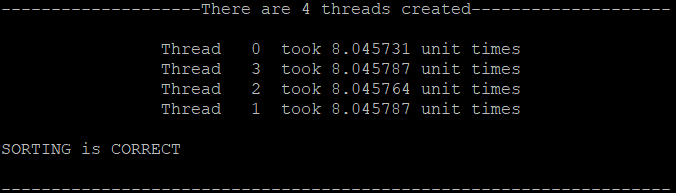
OMP Time:13.3049

Speedup factor(OMP):1.85

MPI Time:13.8216

Speedup factor(MPI):1.89

4 Threads:



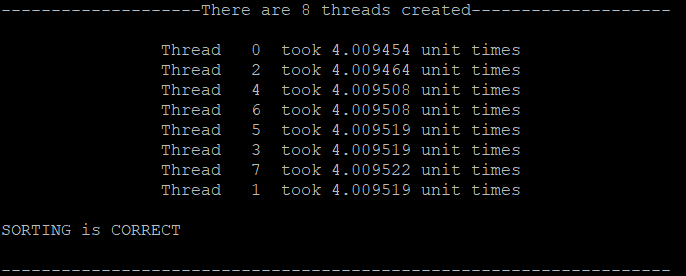
OMP Time:8.0457

Speedup factor(OMP): 3.06

MPI Time:7.155

Speedup factor(MPI):3.67

8 Threads:



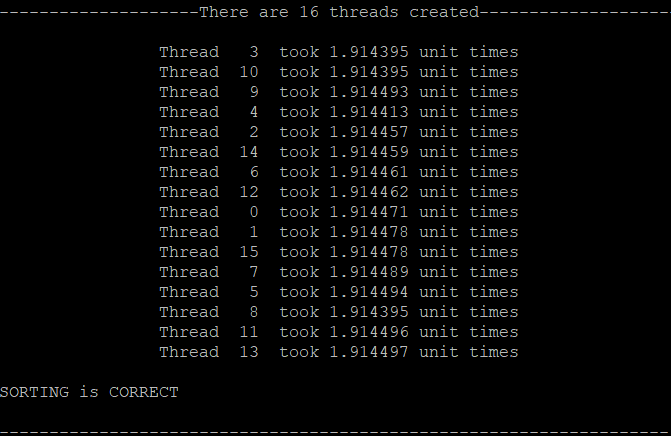
OMP Time:4.0095

Speedup factor(OMP):6.14

MPI Time:4.273

Speedup factor(MPI):6.14

16 Threads:



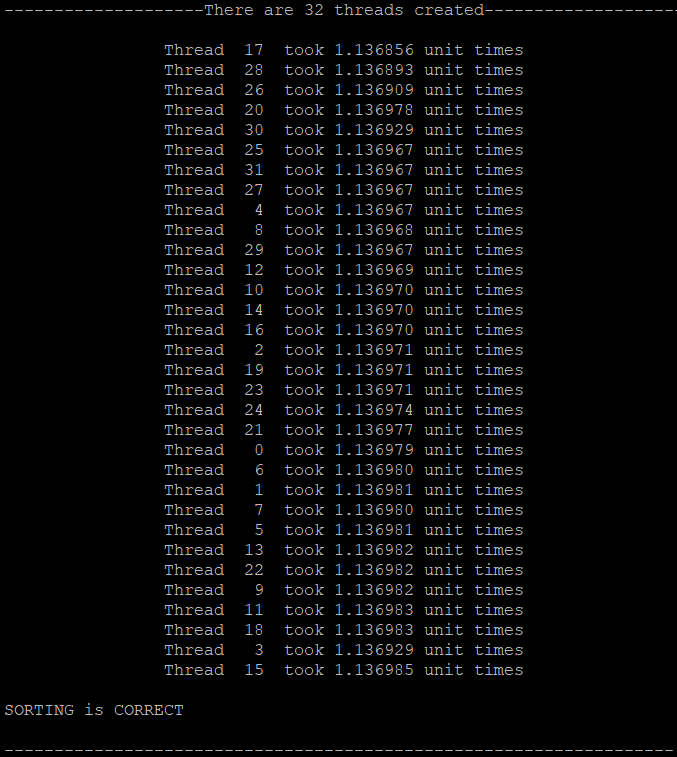
OMP Time:1.9144

Speedup factor(OMP):12.86

MPI Time:3.554

Speedup factor(MPI):7.38

32 Threads:



OMP Time:1.1369

Speedup factor(OMP):21.66

MPI Time:3.121

Speedup factor(MPI):8.41

**心得(與MPI比較)**

# 本次作業的實作，與Training3 的MPI比較，都是使用第二種平行版的bucket sort，結果很明顯指出,在processors(threads)數量少的情況下，OMP、MPI差距甚小，甚至有可能MPI加速效果比較佳，當processors(threads) 數量達到8的時候兩者加速幾乎相同，但在繼續增加processors(threads)後，MPI這種message-passing方法反而加速效果受限了，而OpenMP共享記憶體方法還是穩定的加速。