

# 2021 NYCU OS HW3 report

學號：0717001

系級：生物科技學系大四

姓名：李柏毅

[illegible]

<p>Q2. Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)</p>	<p>According to the screenshot image below:</p> <p>Input data : input1.txt Single thread (split into 8 part) : 0:00.53 Multi thread (split into 8 part) : 0:00.39 Multi thread (split into 16 part) : 0:00.26</p> <pre>[c0717001@linux1 hw_3]\$ g++ 0717001_ST.cpp -o ST [c0717001@linux1 hw_3]\$ time ./ST &lt; input1.txt &gt; output1.txt 0.150u 0.001s 0:00.53 28.3%    0+0k 104+136io 1pf+0w [c0717001@linux1 hw_3]\$ g++ 0717001_MT_worst.cpp -o MT_worst -lpthread [c0717001@linux1 hw_3]\$ time ./MT_worst &lt; input1.txt &gt; output1.txt 0.149u 0.003s 0:00.39 35.8%    0+0k 104+136io 1pf+0w [c0717001@linux1 hw_3]\$ g++ 0717001_MT_best.cpp -o MT_best -lpthread [c0717001@linux1 hw_3]\$ time ./MT_best &lt; input1.txt &gt; output1.txt 0.077u 0.002s 0:00.26 26.9%    0+0k 104+136io 1pf+0w</pre> <p>但是因為 input1.txt 比較小所以有可能會出現 multi thread worst case 跑的跟 best case 差不多快的狀況。因此我也跑了 input2.txt，但是我在跑 single thread 的時候居然忘記下 time……，只知道大概跑了半小時左右。依然還是可以跟其他兩個做出很明顯的比較啦。如下圖</p> <p>According to the screenshot image below:</p> <p>Input data : input2.txt Single thread (split into 8 part) : 約 30 mins Multi thread (split into 8 part) : 11:36.82 Multi thread (split into 16 part) : 4:45.95</p> <pre>[c0717001@linux1 hw_3]\$ time ./MT_worst &lt; input2.txt &gt; output2.txt 1463.819u 0.344s 11:36.82 210.1%    0+0k 0+13456io 0pf+0w [c0717001@linux1 hw_3]\$ diff output2.txt answer2.txt [c0717001@linux1 hw_3]\$ time ./MT_best &lt; input2.txt &gt; output2.txt 727.474u 0.239s 4:45.95 254.4%    0+0k 0+13456io 0pf+0w [c0717001@linux1 hw_3]\$ diff output2.txt answer2.txt</pre>
<p>Q3. You need a brief description of the best multi-threads and worst multi-threads methods. The content includes the number of threads used and the way of partitioning, comparing the difference in time, and taking the screenshot</p>	<p>Number of threads used: MT_best : 31(16 for bubble sort + 15 for merge process) MT_worst : 15(8 for bubble sort + 7 for merge process)</p> <p>The way of partitioning: 都是用一個大的 array 包著數個 vectors。 MT_best : split input data into 16 parts. MT_worst : split input data into 8 parts.</p> <p>The way of comparing: Bubble sort 用以 compare and sort，Merge 部分就是分別兩個要合併的 vector 要吃進來的數字誰大誰就放在比較後面。</p> <p>The difference in time: MT_best : 4:45.95 MT_worst : 11:36.82</p>

<p>between two multi-thread results.</p>	<pre>[c0717001@linux1 hw_3]\$ time ./MT_worst &lt; input2.txt &gt; output2.txt 1463.819u 0.344s 11:36.82 210.1%      0+0k 0+13456io 0pf+0w [c0717001@linux1 hw_3]\$ diff output2.txt answer2.txt [c0717001@linux1 hw_3]\$ time ./MT_best &lt; input2.txt &gt; output2.txt 727.474u 0.239s 4:45.95 254.4%    0+0k 0+13456io 0pf+0w [c0717001@linux1 hw_3]\$ diff output2.txt answer2.txt</pre>
<p>Q4. What did you learn from doing hw3?</p>	<p>這次作業真的學到很多！尤其是 c++語法的各種複習，因為 pthread 吃的 input 必須要是 void 特別的形式，所以就需要用到 struct 去封包全部需要 input 的變數等等，另外就是如果把 vector 設定在全域變數也會更加方便整特資料的拿取跟存取，其實整體來說不會算太難，但是我花了好一段時間才真正搞懂 pthread 韓式到底的底式怎沒運做的，了解完成以後基本上就是超級，！個人覺得是滿好玩的一次作業！</p>