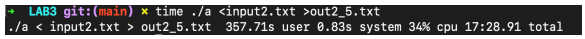
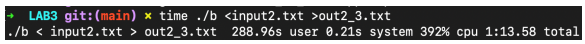
2021 NYCU OS HW3 report

Question	Answer
Q1. Briefly describe your design for the sorting algorithm, merge function, the thread management. Also, describing the number of sort threads and number of merge threads in the Multi-thread program.	我設一個變數紀錄thread的數目（之後比較好調參），然後寫一個for loop不斷create_thread直到所要求得數目，其中要將輸入測資平均分到每個thread，多餘的擺到最後一個thread。 Create 完thread 後，再寫一個for loop去join所有的thread。 接下來進入merge function。merge 內部做merge thread 的管理，也是要記得其中兩倆thread 做完排序後要先join，再繼續大範圍 merge thread。 然後，我thread 的傳參都是以 一個struct 的方式傳進去。 Sort Thread 的數目根據我的調動，我最好的結果sort Thread 為 4 個。 Merge Thread 數目，根據sort Thread 數目而有所改變，兩個小sort Thread 組一個 merge Thread 依此類推。 而我single-thread 也是拆四份個字做 bobble sort 最後再慢慢merge 再一起。 Partition 跟 multi-thread無異。

Q2. Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)	此為serial 總共花17分秒  此為multi thread best solution總共花1分13秒  此為multi thread worst solution總共花1分16秒 這些數值是我的電腦跑的結果 實際在工作站跑的結果可能會有些微落差
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 between two multi-thread results.	其實我覺得這次thread的實作效能主要還是跟thread 數目有關，怎麼編排怎麼merge 都只是其次，因為merge 的動作是seriel 影響不大。 我最好的實作thread數目用4個，而比較差的實作thread 數目也是4個。 Best multi-thread : 我採用前兩個thread, 後兩個thread先組合，最後再全部合併再一起即完成。 Worst multi-thread : 我用遞回的方式實作，然後直到thraed個數為一時，進行兩兩合併，記得要加thread_join否則結果會亂掉。
Q4. What did you learn from doing hw3?	Thread 的實作？ Thread 其實我過去曾經碰過所以不算難。