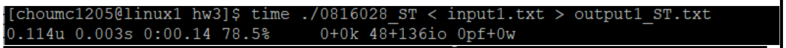
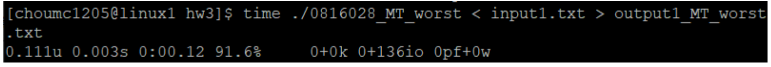
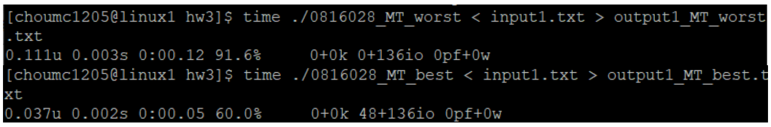
2021 NYCU OS HW3 report

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.	我的方法就像是在用 merge sort，但是多了一個數字 cut 來記說已經切了多少次，由於每次都是切成兩塊，因此最後會切出 2^cut 塊，在每塊做了 bubble sort 後，再兩塊兩塊 merge 直到剩下一塊。MT 的部份在切出 2^cut 塊後，會分配給每一塊一個 thread 做 bubble sort，再分配給每兩塊一個 thread 去做 merge 直到剩下一塊。

Q2.
Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)
	
	`[choumc1205@linux1 hw3]$ time ./0816028_ST < input1.txt > output1_ST.txt`
	`0.114u 0.003s 0:00.14 78.5%      0+0k 48+136io 0pf+0w`
	
	`[choumc1205@linux1 hw3]$ time ./0816028_MT_worst < input1.txt > output1_MT_worst`
	`.txt`
	`0.111u 0.003s 0:00.12 91.6%      0+0k 0+136io 0pf+0w`

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.
	Best multi-threads 是切成 8 塊而 worst multi-threads 是切成 2 塊
	
	`[choumc1205@linux1 hw3]$ time ./0816028_MT_worst < input1.txt > output1_MT_worst`
	`.txt`
	`0.111u 0.003s 0:00.12 91.6%      0+0k 0+136io 0pf+0w`
	`[choumc1205@linux1 hw3]$ time ./0816028_MT_best < input1.txt > output1_MT_best.t`
	`xt`
	`0.037u 0.002s 0:00.05 60.0%      0+0k 48+136io 0pf+0w`

Q4. What did you learn from doing hw3?	sub array 切得越多或是 array 本身不長，mt 的優勢反而顯現不出來，因為這時 sorting 時間本來就不長，而 mt 還要另外花時間開 thread，就會造成 st 比 mt 快的情況。