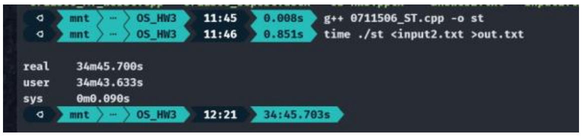
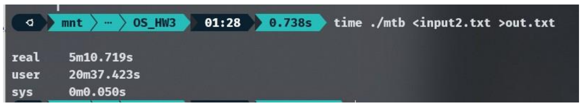
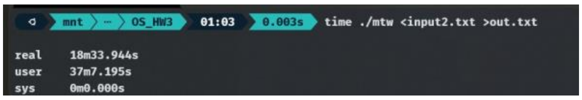
2021 NYCU OS HW3 report

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.	Sort 演算法我是採用最簡單的 bubble sort。merge 則是與 merge sort 的 merge 部分類似，不斷比較兩個 sorted 部分的數字大小，再去更改 vector 中的數字。Thread 部分使用了 7 個(best)與 3 個(worst)，分別是 best: 4(sort)+3(merge)與 worst: 2(sort)+1(merge)。

Q2.
Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)
	ST:
	
	`mnt … OS_HW3 11:45 0.008s g++ 0711506_ST.cpp -o st`
	`mnt … OS_HW3 11:46 0.851s time ./st <input2.txt >out.txt`
	`real    34m45.700s`
	`user    34m43.633s`
	`sys     0m0.090s`
	`mnt … OS_HW3 12:21 34:45.703s`
	MT-best:
	
	`mnt … OS_HW3 01:28 0.738s time ./mtb <input2.txt >out.txt`
	`real    5m10.719s`
	`user    20m37.423s`
	`sys     0m0.050s`
	MT-worst:
	
	`mnt … OS_HW3 01:03 0.003s time ./mtw <input2.txt >out.txt`
	`real    18m33.944s`
	`user    37m7.195s`
	`sys     0m0.000s`
	2085.7/310.719=6.712
	MT-best 比 ST 快約 6.712 倍。

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 部分我是將其分為四等分，分給 4 個 thread 做 sort，最後由 3 個 thread 去將其兩兩 merge，最後得出答案。 Worst 部分則是分為兩等分，利用 2 個 thread 去做 sort，再用 1 個 thread 將兩個部分 merge，最後得出答案。 1113.9/310.719=3.585 截圖在 Q2 部分，可以看見 best 比 worst 快了約 3.6 倍。

Q4. What did you learn from doing hw3?	我學到如何運用上課學到的 pthread api，並且實際體驗了使用 multi-thread 的好處，還有運用 big-O 估計程式執行時間與加速倍率。