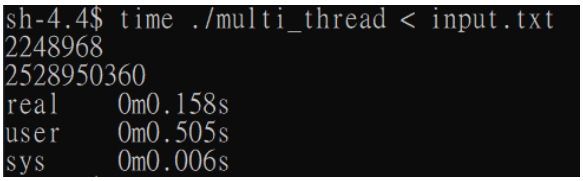
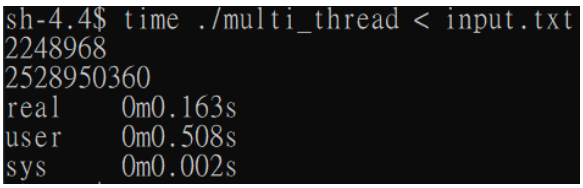
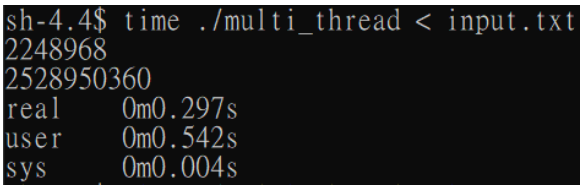
2022 NYCU OS HW2 report

Question	Answer
Q1. (5pts) Briefly describe your design for the add, multiple function of matrix, the thread management. Also, describe the number of threads in the Multi-thread program.	I use a for loop to create threads, and distribute the rows to each thread. In the thread function, I use two for loops for addition and three for loops for multiplication, like single_thread.cpp does. It's also worth mentioning that I directly sum up matC and matD inside the thread function to make the calculation faster. Then, I use pthread_join to wait for all the threads to complete its calculation. Finally, I sum up all the results of each thread and get the final answer.

Q2. (15pts)
Try at least 3 kinds of number of threads, and compare the difference in time.(Take screenshots of the time of each case)
Also, explain the results.
	#threads = 20
	 
	`sh-4.4$ time ./multi_thread < input.txt`
	`2248968`
	`2528950360`
	`real 0m0.158s`
	`user 0m0.505s`
	`sys 0m0.006s`
	
	#threads = 15
	 
	`sh-4.4$ time ./multi_thread < input.txt`
	`2248968`
	`2528950360`
	`real 0m0.163s`
	`user 0m0.508s`
	`sys 0m0.002s`
	
	#threads = 2
	 
	`sh-4.4$ time ./multi_thread < input.txt`
	`2248968`
	`2528950360`
	`real 0m0.297s`
	`user 0m0.542s`
	`sys 0m0.004s`
	
	When number of threads = 2, each thread has to do 250 rows calculation, which is much larger than doing 25 rows calculation when number of threads = 20.
	Besides, the time of #threads=15 ~ 20 is roughly the same (in the 0.15~0.17 interval). I guess it's because the total calculation isn't that large, though

	#threads=20 saves some time in parallel calculation but thread creation also need some cpu time.

Q3. (10pts)
Show the best speedup between multi-thread and single-thread. (Take screenshots of the time of single-thread and multi-thread)
Also, explain why multi-thread is faster.
	Multi_thread:
	`sh-4.4$ time ./multi_thread < input.txt`
	`2248968`
	`2528950360`
	`real 0m0.158s`
	`user 0m0.505s`
	`sys 0m0.006s`
	
	Single_thread:
	`sh-4.4$ time ./single_thread < input.txt`
	`2248968`
	`2528950360`
	`real 0m0.708s`
	`user 0m0.690s`
	`sys 0m0.011s`
	
	Speedup = 4.481
	Multi-thread is faster because parallel calculation is faster than compute it all along. Also, threads shared the same global variable so we can save some time transferring the data.