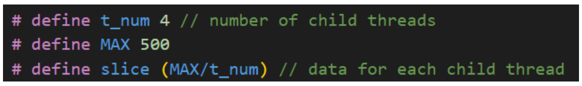
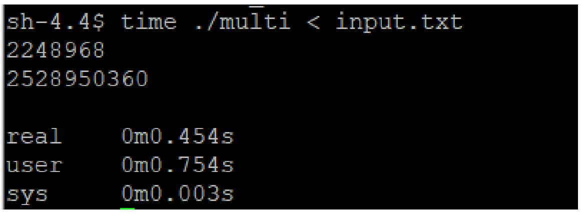
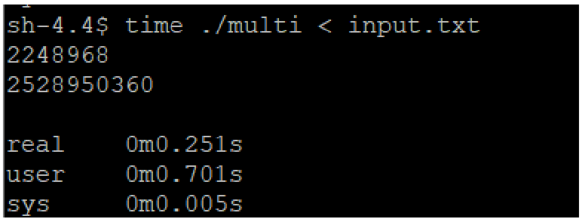
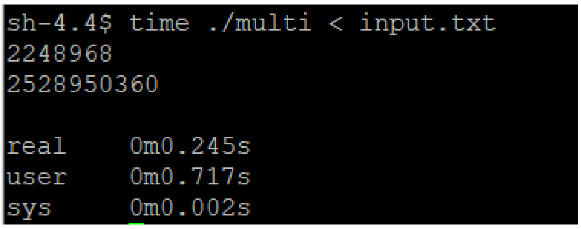
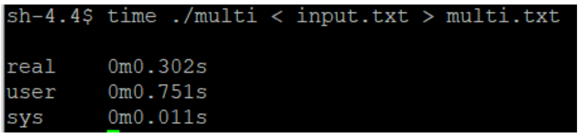
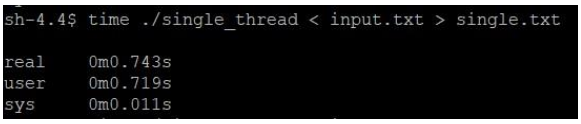
# 2022 NYCU OS HW2 report

109550134 梁詠晴

| Question | Answer |
|---|---|
| Q1. (5pts)<br>Briefly describe your design for the add, multiple function of matrix, the thread management.<br>Also, describe the number of threads in the Multi-thread program. | <br>`# define t_num 4 // number of child threads`<br>`# define MAX 500`<br>`# define slice (MAX/t_num) // data for each child thread`<br>child thread 均分須處理的資料，也就是各自負責計算出 matC, mat 中 (MAX/thread 數量)個 row 的答案。<br>替 child thread 編號，並在使用 pthread_create 時將編號作為參數傳入 child thread 中，使用編號依序分配該 child thread 負責的 row。若 MAX 無法被 thread 數整除，則最後一個 child thread 會負責剩餘的 row。 |
| Q2. (15pts)<br>Try at least 3 kinds of number of threads, and compare the difference in time.(Take screenshots of the time of each case)<br>Also, explain the results. | 分別測試 2, 4, 6 個 thread 的時間差異:<br>Thread*2:<br><br>`sh-4.4$ time ./multi < input.txt`<br>`2248968`<br>`2528950360`<br>`real    0m0.454s`<br>`user    0m0.754s`<br>`sys     0m0.003s`<br>Thread*4:<br><br>`sh-4.4$ time ./multi < input.txt`<br>`2248968`<br>`2528950360`<br>`real    0m0.251s`<br>`user    0m0.701s`<br>`sys     0m0.005s`<br>Thread*6:<br><br>`sh-4.4$ time ./multi < input.txt`<br>`2248968`<br>`2528950360`<br>`real    0m0.245s`<br>`user    0m0.717s`<br>`sys     0m0.002s`<br>所需時間: 2 > 4 >= 6，使用越多 thread 時，每個 thread 需要處理的資料量少，且可平行處理，因此所需時間越短。然而在多次測試中，使用 6 個 thread 和 4 個 thread 的結果並無顯著差異，故推測 |

|  |  |
|---|---|
|  | 可能可能已使用最大硬體資源，已達到 multi-thread 能帶來的最大加速。 |
| Q3. (10pts)<br>Show the best speedup between multi-thread and single-thread. (Take screenshots of the time of single-thread and multi-thread)<br>Also, explain why multi-thread is faster. | multi-thread:<br><br>sh-4.4$ time ./multi < input.txt > multi.txt<br><br>real    0m0.302s<br>user    0m0.751s<br>sys     0m0.011s<br>single-thread:<br><br>sh-4.4$ time ./single_thread < input.txt > single.txt<br><br>real    0m0.743s<br>user    0m0.719s<br>sys     0m0.011s<br>使用 single-thread 時，資料全部由同一個 thread 處理; 使用 multi-thread 時，資料可分給多個 thread 平行處理，其中每個 thread 僅被分配到部分資料，故可較快完成計算。 |