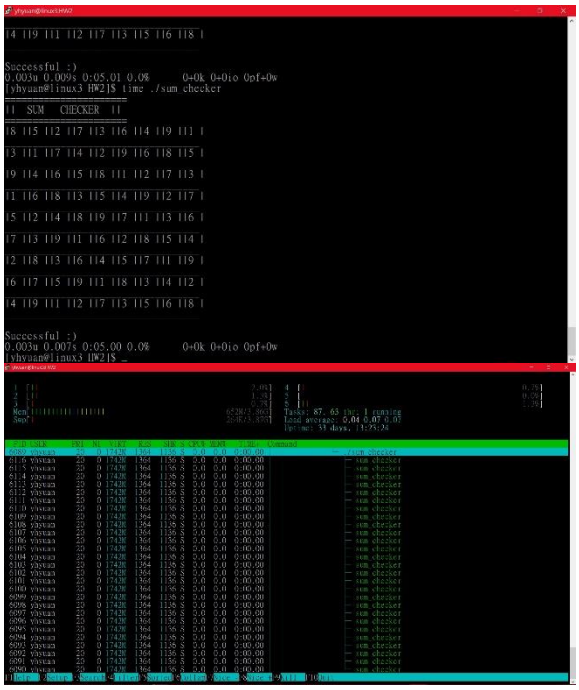
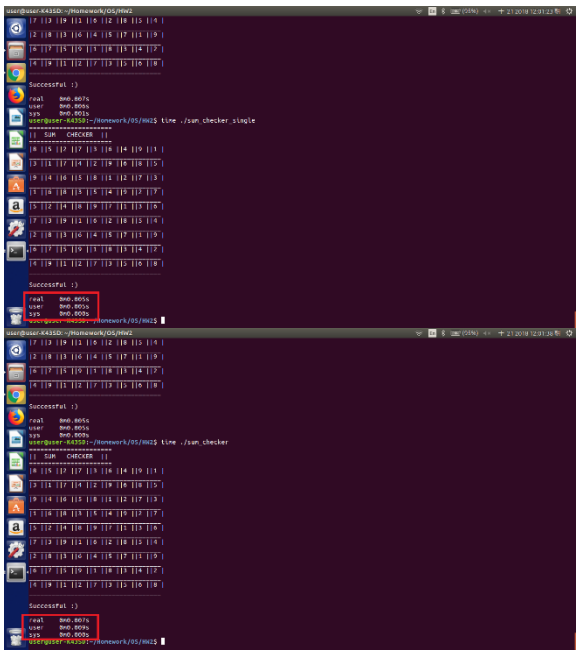
NCTU OS HW2 report 2018

Name: 袁鈺勛

Student ID: 0510002

Question	Answer
Q1. **Briefly describe about your design for the problem "Sum Checker" and total num of threads you used in your code.**	```14 #define NUMBER_OF_THREADS 27 /* Hint */ 15 #define PUZZLE_SIZE 9 16 17 18 19 int answers[NUMBER_OF_THREADS];``` 如上圖，我使用了 **27** 個 **thread**。 ```93 void* child(void* data) { 94 parameters* args = (parameters*) data; 95 96 int sum = 0; 97 if (args->thread_number < 9) { // Check row 98 for (int col = 1; col < 10; col++) 99 sum += puzzle[args->row][col]; 100 } else if (args->thread_number < 18) { // Check column 101 for (int row = 1; row < 10; row++) 102 sum += puzzle[row][args->col]; 103 } else { // Check subgrid 104 for (int i = args->row; i < args->row + 3; i++) 105 for (int j = args->col; j < args->col + 3; j++) 106 sum += puzzle[i][j]; 107 } 108 answers[args->thread_number] = sum; 109 sleep(5); 110 111 pthread_exit(NULL); 112 }``` 在上圖中，前 **9** 個 **thread** 檢查 **row**，中間 **9** 個 **thread** 檢查 **column**，後 **9** 個 **thread** 檢查 **sub-grid**，並且都將 **sum** 存到 **answers** 供 **main** 檢查。 ```123 for (int i = 0; i < 27; i++) { // Create threads 124 args[i].thread_number = i; 125 126 if (i < 9) { // Check row 127 args[i].row = i + 1; 128 } else if (i < 18) { // Check column 129 args[i].col = i - 8; 130 } else { // Check subgrid 131 switch (i) { 132 case 18: args[i].row = 1; args[i].col = 1; break; 133 case 19: args[i].row = 1; args[i].col = 4; break; 134 case 20: args[i].row = 1; args[i].col = 7; break; 135 case 21: args[i].row = 4; args[i].col = 1; break; 136 case 22: args[i].row = 4; args[i].col = 4; break; 137 case 23: args[i].row = 4; args[i].col = 7; break; 138 case 24: args[i].row = 7; args[i].col = 1; break; 139 case 25: args[i].row = 7; args[i].col = 4; break; 140 case 26: args[i].row = 7; args[i].col = 7; break; 141 } 142 } 143 144 pthread_create(&(t[i]), NULL, child, &(args[i])); 145 }``` 上圖中在 **main** 裡建立 **27** 個 **thread** 並給予每個 **thread** 檢查 **puzzle** 的起始位置。 ```147 // Wait for all child thread finished and check the answers 148 pthread_join(t[0], NULL); 149 int answer = answers[0]; 150 for (int i = 1; i < 27; i++) { 151 pthread_join(t[i], NULL); 152 if (answer != answers[i]) 153 rv = false; 154 }``` **Main** 會等待每個 **thread** 結束並檢查

	sum 是否相同。
Q2. **Show your thread info screenshots while "Sum Checker" code running.**	
Q3. **Compare the time between Single-thread and Multi-thread.**	 **Speed up = 71.4%**
Q4. **What you learned from doing OS hw2 or some improvements you want to say to TAs.**	這次作業學會了怎麼使用 **thread**，同時也發現 **thread** 將工作切得太細也不太好，因為 **main** 最後要確認的 **sum** 比只用 **9** 個的更多，反而會使整體表現更慢。

p.s You can reference to homework info pdf and show your answer as the format for Q2 and Q3.