**Embedded System Software Design Project 1**

M10907324吳俊逸

⚫ Part 1 [Global Scheduling. 10%]

▪ Describe how to implement Global scheduling by using pthread. 5%





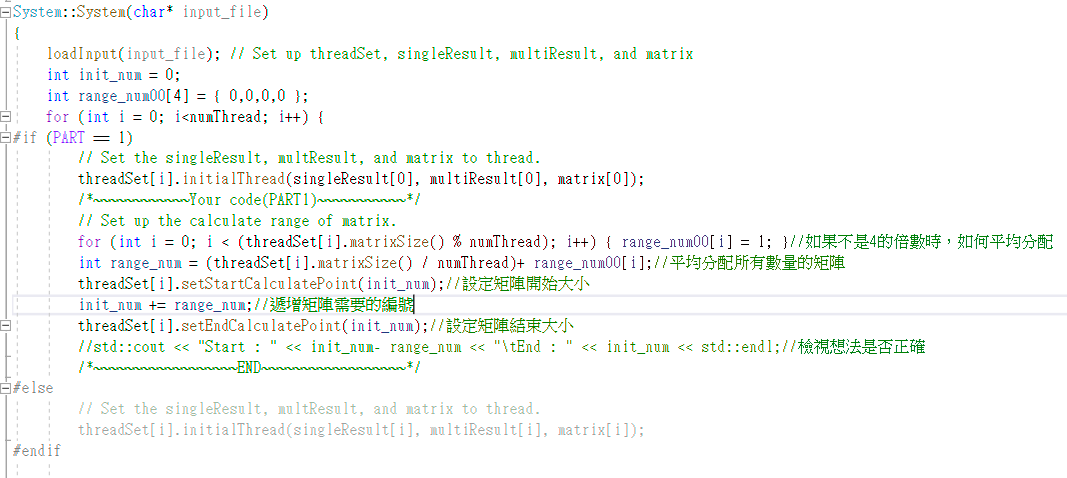
▪ Describe how to observe task migration. 5%





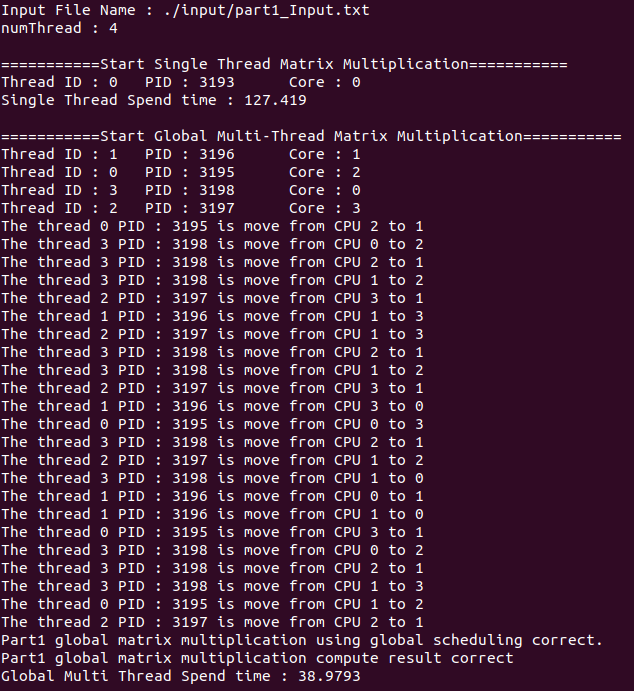
[Partition Scheduling. 5%]

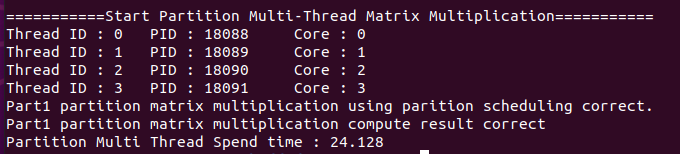
▪ Describe how to implement partition scheduling by using pthread.



[Result. 10%]

▪ Show the scheduling states of tasks. (You have to show the screenshot result of using the input part1\_Input.txt)

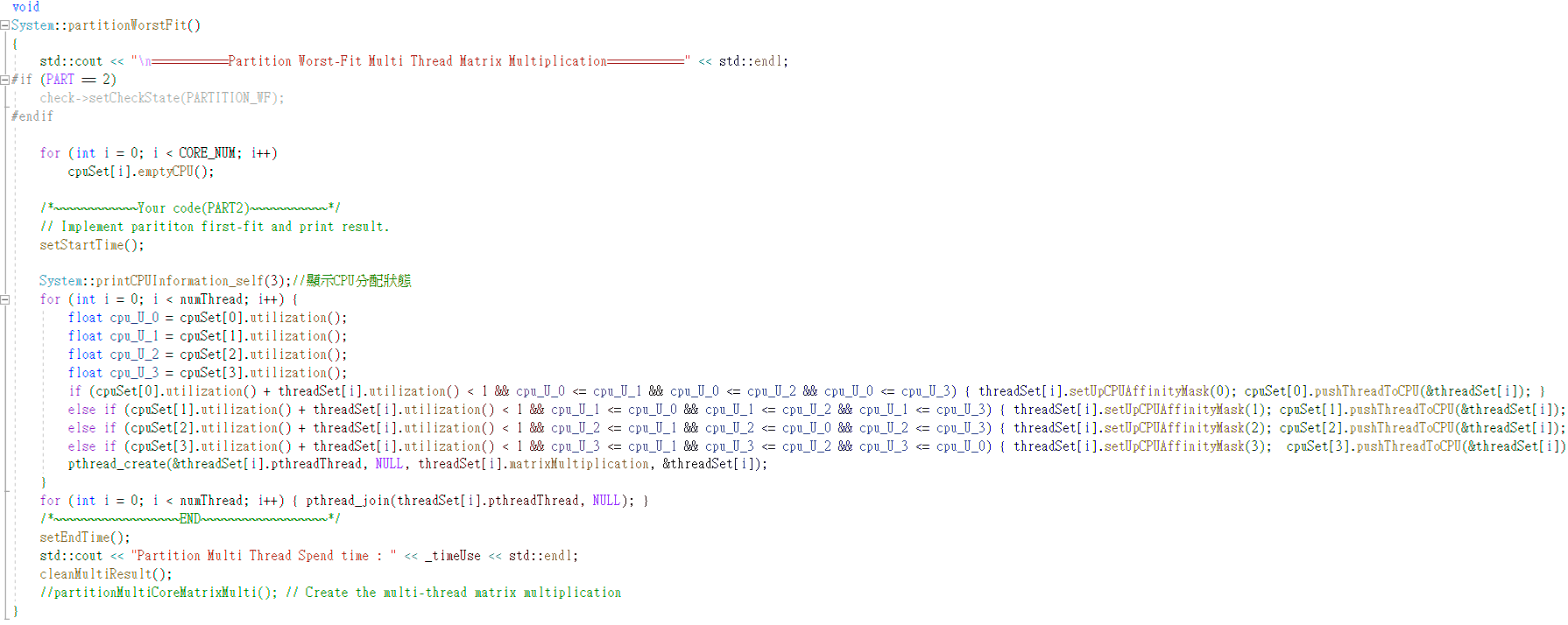




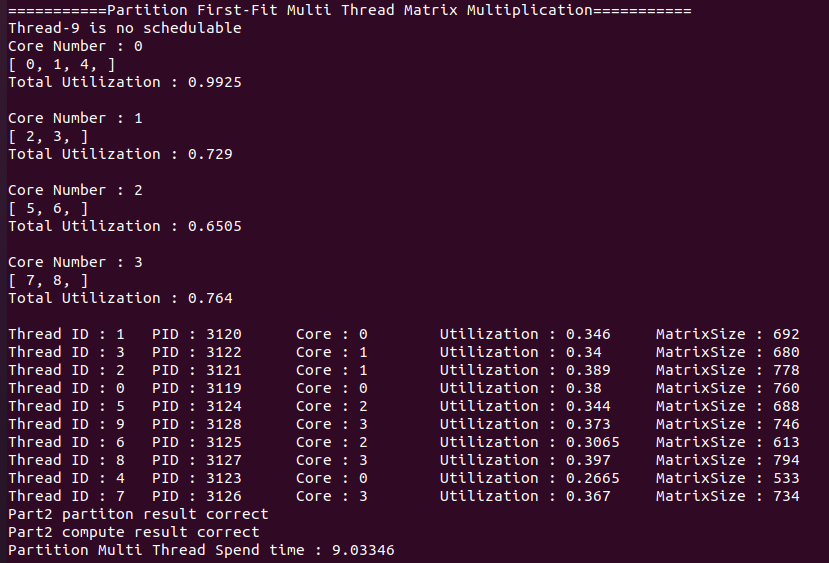
⚫ Part 2 [Partition method Implementation. 10%]

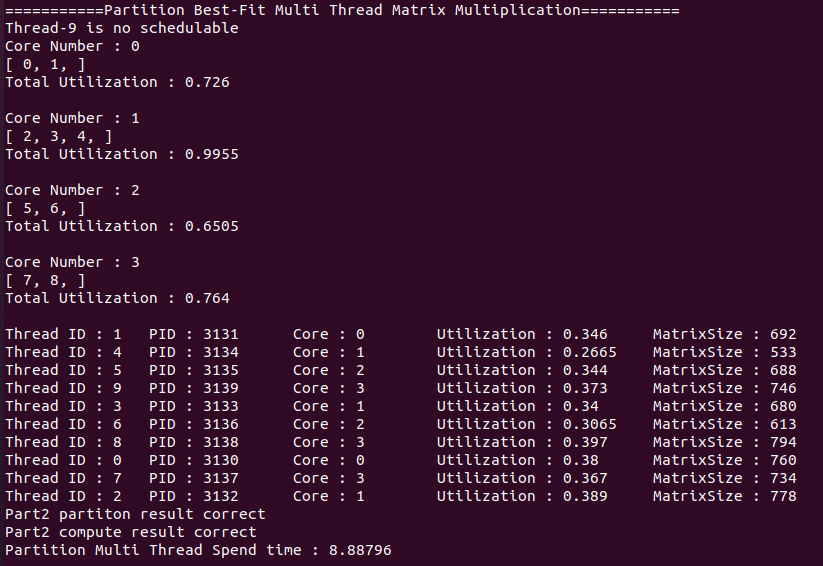
▪ Describe how to implement the three different partition methods (First-Fit, Best-Fit, Worst-Fit) in partition scheduling. [Result. 30%]

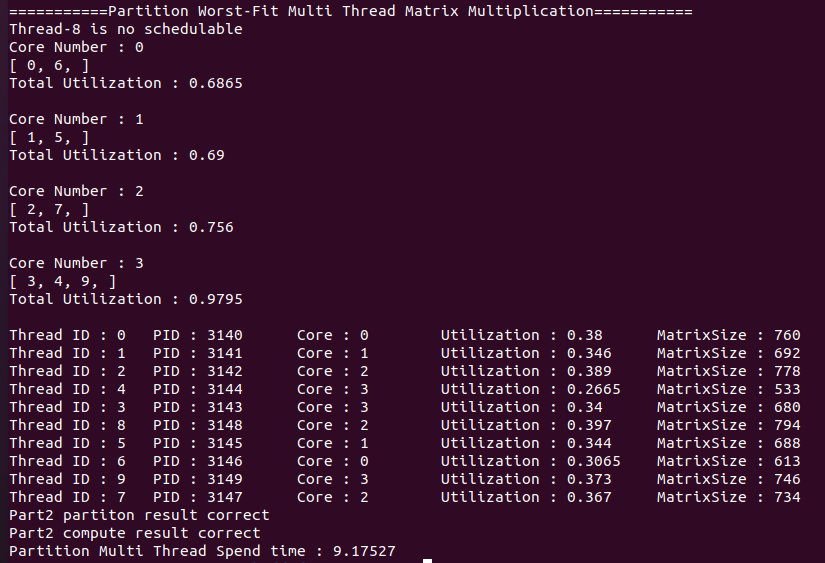


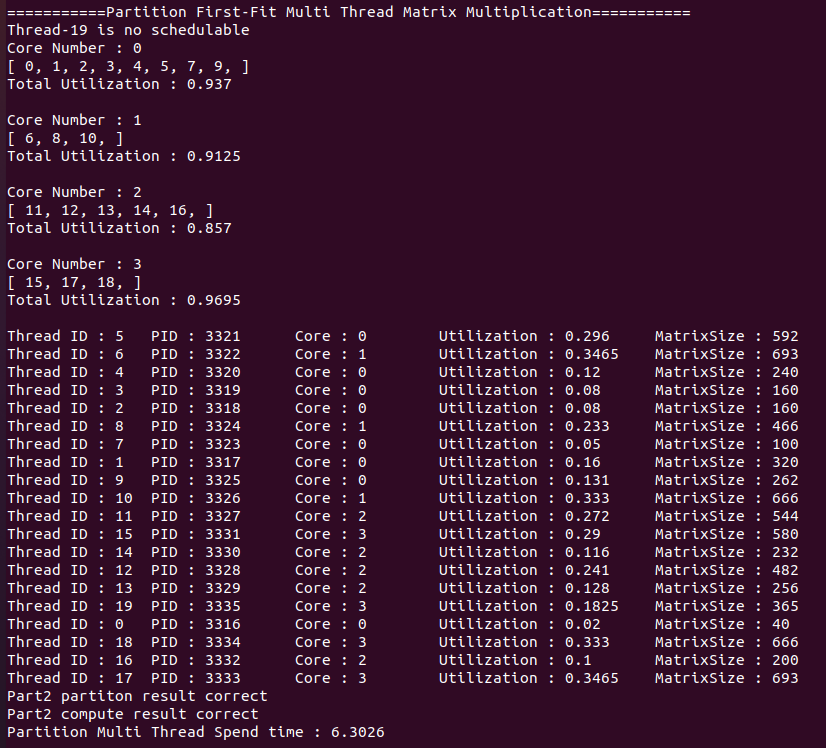


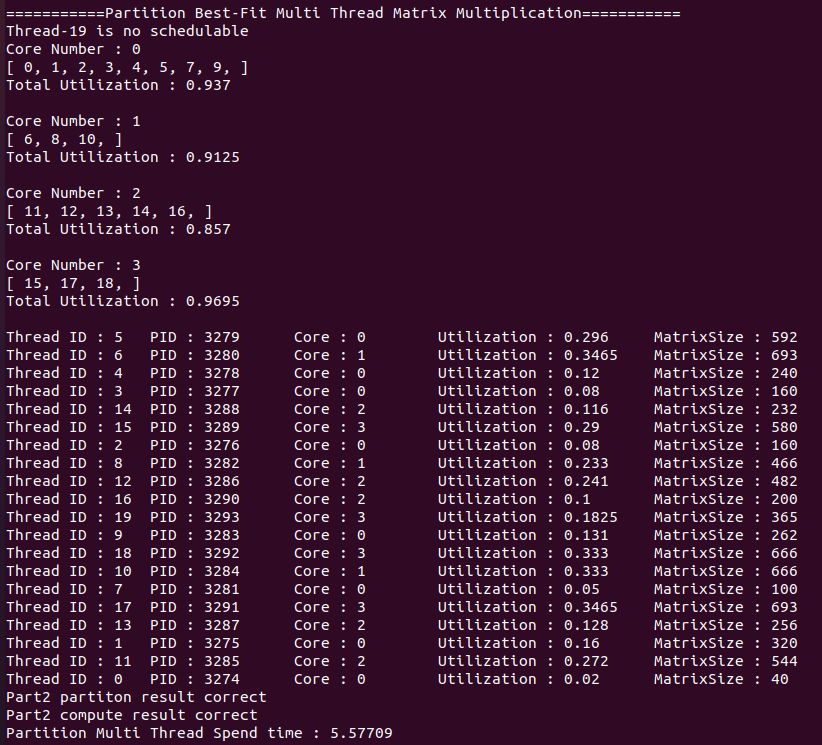
▪ Show the scheduling states of tasks. (You have to show the screenshot result of using input part2\_Input\_10.txt and part2\_Input\_20.txt)

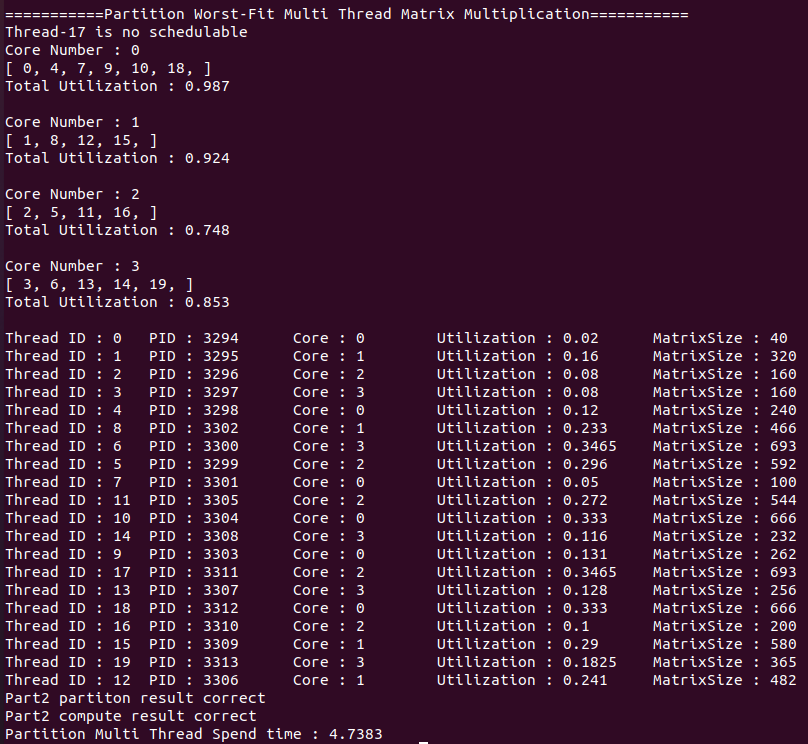












⚫ Part 3 [Scheduler Implementation. 10%]

▪ Describe how to implement the scheduler setting in partition scheduling. (FIFO with FF, RR with FF) [Result. 10%]

▪ Show the process execution states of tasks. (You have to show the screenshot result of using input part3\_Input.txt)

⚫ Discussion

▪ Analyze and compare the response time of the program, with single thread and multi-thread using in part1 and part2. (Including Single, Global, FirstFit, Best-Fit, Worst-Fit) 10%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Single | Global | FirstFit | Best-Fit | Worst-Fit |
| responsetime | 127.419 | 38.9793 | 9.03346 | 8.88796 | 9.17527 |

我們可以從實驗數據得知，單執行序一定會比多執行序還要花時間。本次的例子所使用的核芯數量最大為4核，故多執行序所花費的時間為單執行序的四分之一。再來討論後續的FirstFit, Best-Fit, Worst-Fit這三種，這三種多執行序皆是先將Thread分配好再執行，從而得知這三種多執行序會比一般的多執行序(Glabal)還要快完成。我們還可以利用上表所示的數據，間接的證明的這三種多執行序的英文名稱由來。

▪ Analyze and compare the response time of the program, with two different schedulers. (FIFO with FF, RR with FF) 5%

|  |  |  |
| --- | --- | --- |
|  | FIFO with FF | RR with FF |
| response time | 12.8467 | 12.5775 |

FIFO ( First Input First Output)簡單說就是指先進先出；RR(Round-robin) 通常指將多個某物輪流用於某事。在用於此系統中RR是將thread分成相同大小，這樣有利於傳輸至BUS的等候時間，因為在FIFO(沒有切割)的時候，需要先等候該thread執行完之後才會接上下一個thread；然而在RR中我們可以大幅減少在兩個(或多個)不同core中的thread傳輸至BUS的時間。從實驗結果也可以證明將thread切割之後會比沒有切割還要來的快，但是由於本次實驗中的thread的Utilization較小，如果有較大Utilization的thread放入此系統，則會發現明顯的差異。