**OS HW3 report**

Name:紀竺均

Student ID:109550027

|  |  |
| --- | --- |
| Question | Answer |
| Q1.  Briefly describe about your data structure for recording process’ time or anything you need to record. | I use **vector** to store the arrival time and burst time of each process.  In hw3-1, I use a **reverse priority queue** to store the remaining time of each process so that I can pop out the process with the shortest time.  In hw3-2 and hw3-3, I use **queue** to store the process id so that I can pop out the process that join the queue first. |
| Q2.  How to simulate process scheduling? | I use a while loop and let time+=1 at the end of the loop to simulate the clock. Inside the loop, I decide which process to run and minus 1 of the chosen process’s remaining time, i.e. remainTime[process]--, to simulate that this process occupy cpu for 1 time quantum. |
| Q3.  Some problems you meet and how to resolve. | Initially, in hw3-3, if there’s a switch from FCFS queue to RR queue, my program won’t pop out the current process and push it back in FCFS queue.  To solve this, I add a variable “sw” to keep track of this scenario and do the action.  // In RR queue:    // In FCFS queue: |
| Q4.  What you learned from doing OS hw3 and something you want to discuss with TAs. | 雖然本來就熟悉這些演算法，但實作又是另一回事，舉例來說，第二題的round robin，原本要讓clock一次跳min(time quantum, remain time)的值，但發現這樣無法準確抓出新的process進來的順序，所以還是改成一次跳一個time quantum。  另外這次助教很佛心的只測已經處理過的側資(arrival time依時間順序排好且不會重疊)，感覺如果要將實際資料的演算法寫出來，需要思考的就更複雜了。 |