***NTCU OS HW3 report 2018***

Name: 袁鈺勛

Studient ID: 0510002

|  |  |
| --- | --- |
| Question | Answer |
| Q1.  Briefly describe about your data structure for recording process’ time or anything you need to record. | D:\課業\大三\上學期\作業系統概論\HW3\processStructure.jpg  每個process的data structure如上圖。processID用來記錄txt內的process順序。orderID是因為將process照arrivalTime排列後，順序可能跟原先不同，所以會在排完後重新給一個ID。arrivalTime和burstTime紀錄txt內相對應的資料。runTime用來記錄模擬時跑了多少次。waitingTime和turnaroundTime會在process跑完後記錄。 |
| Q2.  How to simulate process scheduling? | D:\課業\大三\上學期\作業系統概論\HW3\SJF.jpg  上圖是SJF的模擬過程，直到所有process跑完才會結束。在while中會判斷queue中是否有process，有的話要增加第一個的runTime和檢查是否已達到他要的burstTime，如果有達到burstTime就要記錄waitingTime和turnaroundTime。然後要加入已達arrivalTime的process進queue，並且判斷剛剛是否有process完成，有的話就sort全部的queue，沒有的話就只sort第一個以外的process，因為這表示第一個還在跑。  D:\課業\大三\上學期\作業系統概論\HW3\SRTF.jpg  上圖是SRTF的模擬過程，直到所有process跑完才會結束。在while中會判斷queue是否有process，有的話就要減少burstTime，並看burstTime是否已經只剩零，如果是就要記錄waitingTime和turnaroundTime。然後要將已達arrivalTime的process加入queue中並sort整個queue，因為要比較誰剩下的burstTime比較少。  D:\課業\大三\上學期\作業系統概論\HW3\RR_1.jpg  D:\課業\大三\上學期\作業系統概論\HW3\RR_2.jpg  上兩張圖是RR的模擬過程，直到所有process跑完才會結束。在while裡面會判斷queue中是否有process，有的話就要減少第一個的runTime並增加他已使用的time quantum，如果process的runTime剩零或time quantum已達規定值，就要中斷他，從queue中移除他並歸零time quantum，runTime歸零還要記錄waitingTime和turnaroundTime，如果是time quantum已達規定值還要記錄是哪一個被移除。接下來將以達arrivalTime的process加入queue中並且如果剛剛被移除的process還未做完，要再將他加入queue中。  D:\課業\大三\上學期\作業系統概論\HW3\MF.jpg  上圖是Multilevel Feedback的主要流程，直到所有process都跑完才會結束。會依照各level的優先順序檢查各queue是否有process要執行。在第二層的RR(lowQueue)中，會利用empty和removed確認SJFQueue被中斷的情況，如果是原本就空的或是在已完成一個process後才被中斷，那就要sort整個queue，如果不是這樣的情況，那就表示說是在process跑到一半的時候被中斷，那就只要sort第一個以外的process就好。  D:\課業\大三\上學期\作業系統概論\HW3\MF_highRR.jpg  上圖是MF的highRR的模擬流程，是high priority queue，直到highQueue沒有process才會結束。和上面的RR一樣會判斷是time quantum用完還是已達到burstTime，只是time quantum用完但未達到burstTime的process會被加到下一層的lowQueue中。  D:\課業\大三\上學期\作業系統概論\HW3\MF_lowRR.jpg  上圖是MF的lowRR模擬流程，是medium priority queue，直到lowQueue沒process才會結束。和上面的RR一樣會判斷是time quantum用完還是已達到burstTime，只是time quantum用完但未達burstTime的process是被加到下一層的SJFQueue，而且如果有達到arrivalTime的process，是會被加到上一層的highQueue並且這層的執行會被中斷。  D:\課業\大三\上學期\作業系統概論\HW3\MF_SJF.jpg  上圖是MF的SJF流程圖，直到SJFQueue沒有process才會結束。和上面的SJF一樣會將執行完的process從SJFQueue中移除，但當有達到arrivalTime的process出現，是被加入最上層的highQueue並且這層的執行會被中斷。其中又有加入一個oldNumber用來記錄上一次的未執行完process數量，會以他來判斷這層的執行在被中斷的時候是剛好有process執行完還是未跑完，以此來設定removed。 |
| Q3.  Some problems you meet and how to resolve. | 一開始以為txt給的順序都會照arrivalTime排列，所以是直接用processID來讀取我要的process，但後來知道順序可能會是亂的，所以加入了一個orderID用來給模擬的程序執行，執行完後再已processID排序成原本讀取到的順序再輸出至輸出檔。 |
| Q4.  What you learned from doing OS hw3 and something you want to discuss with TAs. | 在multilevel因為各個schedule會互相影響，所以常常要多考慮一些被中斷時要保留哪些狀態來判斷下一步應該要做什麼，還滿複雜的。 |