

OS HW3 report

Name: 龔祐萱

Student ID: 0713216

Question	Answer
<p>Q1.</p> <p>Briefly describe about your data structure for recording process' time or anything you need to record.</p>	<p>1. 紀錄時間</p> <p>使用一個 integer 變數去記錄目前的時間為何，並且每經過一次 while loop 就加 1。</p> <p>2. SRTF 紀錄 process</p> <p>先宣告一個 struct，並存取 process 的 arrival time, burst time 等相關資訊，再宣告 struct process array 儲存 input 的資訊。</p> <p>3. RR, Multilevel Feedback Queue 紀錄 process 使用順序</p> <p>使用 queue 去紀錄 process 的執行順序，如果 process 沒有在 time quantum 內執行完，會重新把 process push 進 queue 中；如果 process 在 time quantum 內執行完，就會把 process pop 出 queue 中。</p>
<p>Q2.</p> <p>How to simulate process scheduling?</p>	<p>1. SRTF</p> <p>SRFT 是要選擇 remaining time 最短的 process 作為下一個執行的 process，因此每次迴圈都會宣告一個 min 值，並將大設為極大值，用於比較所有 process 目前的 remaining time，並使用變數 index 紀錄 process 的編號。如果多個 process 都有相同的 remaining time，則選取 arrival time 最早的 process 作為下一個執行的 process。</p>

2. RR

RR 主要是使用 queue 去記錄下一個執行的 process 是哪個。在進入迴圈之前會先看是否有 process 在第 0 秒時進入，如果有就把它 push 進 queue 中。進入迴圈之後，每次迴圈都會檢查下一秒是否會有 process 進入，如果有就把它 push 進 queue 中。

接著就會從 queue 中選取目前在 queue 中最前端的 process 作為下一個執行的 process，因此會把它從 queue 中 pop 出來。之後就會執行減少 burst remaining time 以及其他 process 增加 waiting time 的程序，同時檢查該 process 是否會在 time quantum 內完成執行，如果完成執行就會計算它的 turnaround time 以及計算整個系統的 total waiting time 和 total turnaround time。如果無法在 time quantum 內完成執行，就會把 process push 到 queue 中的最後面。

3. Multilevel Feedback Queue

Multilevel feedback queue 主要是使用兩個 queue 來實作，而這兩個 queue 會分別使用不同的 function 來實作不同的 scheduling 機制。

RR 的實作方式大致於上述 hw3_2 相同，只是當 process 無法在 time quantum 內完成執行時，會把 process push 到 FCFS 的 queue 中，等到 RR queue 空了之後才會執行 FCFS queue 中的 process。

FCFS 的實作方式就比較複雜，會需要考慮到兩種情況來決定下一個執行的 process。其一為系統因為 RR queue 空了所以到 FCFS queue 選擇下一個執行的 process，這個時候將 queue 中最前面的 process pop 出來作為下一個要執行 process。另一個情況為連續多個迴圈都是執行 FCFS，因此會使用一個變數去記

	<p>錄 process 的 index，使得多個迴圈都是執行同一個 process，而不是從 queue 中 pop process。</p> <p>如果 process 執行完成後就會處理 turnaround 以及 total waiting time 和 total turnaround time 等計算以及相關變數的設定。</p>
<p>Q3. Some problems you meet and how to resolve.</p>	<p>1. 處理 RR 遇到第 0 秒或前幾秒沒有 process 進入系統的情況以及給予同時間進入 ready queue 的 process，進入系統的 process 會有較高的優先權。</p> <p>因為考慮可能會有第 0 秒沒有任何 process 進入系統的情況，因此在 while loop 前會先確認是否有 process 在第 0 秒進入系統，如果有就會將變數 first_value 更改為 1；如果沒有，變數會維持為 0。</p> <p>進入 while loop 後會檢查下一秒是否會有 process 進入，如果有就會將 process push 到 queue 中，並更改 next_value 為 1。</p> <p>透過每次 while loop 開始時檢查下一秒是否會有 process 進入系統，來讓進入系統的 process 會比因 time quantum 時間到還未執行完成的 process 有更高的優先權。</p> <p>在 while loop 後端使用 if 來判斷 queue 中是否有 process，如果有，可以去執行後續的程序；如果沒有，就只會增加 timer，不會執行其他的程序，以此法來處理 RR 遇到第 0 秒或前幾秒沒有 process 進入系統的情況。</p> <p>2.處理 multilevel feedback queue 中 FCFS 選擇 process 的方式。</p> <p>FCFS 的會出現的情況可以分為以下三種。</p>

	<p>第一種為沒有要切到 RR 執行，把目前的 process 執行到完成，在第一個 while loop 會先從 FCFS queue 中 pop 出一個 process 執行，接著使用變數 fcfs_last_index 去紀錄目前正在執行的 process 為哪個，在接下來的 while loop 中都會把 fcfs_last_index 指派到 index，使得同一個 process 可以經過多個 while loop 把 process 執行完。</p> <p>第二種為沒有要切到 RR 執行，但目前的 process 執行完畢，因此更改變數 finish 為 1 讓程式知道目前為 FCFS 的情況下 process 執行完畢，而下一個迴圈會需要從 FCFS queue 中 pop 出一個 process 執行。</p> <p>第三種為原本為 FCFS 執行，切到 RR 執行之後有回到 FCFS 執行。如果 process 還沒有執行完，程式會先把 process push 回 queue 中，等到回到 FCFS 時就會再從 queue 中 pop 出一個 process 執行。</p>
<p>Q4. What you learned from doing OS hw3 and something you want to discuss with TAs.</p>	<p>透過這次的作業實作讓我更加了解不同種 scheduling 的概念以及其運作方法，同時也發現實作程式需要考慮很多種問題以及可能發生的情況，來讓程式更完善，使得執行時更加流暢，減少錯誤率。</p>