

## 一、開發環境

Dev C++

## 二、實作方法與流程

**FCFS:** 預先將所有的 process 依照抵達時間小到大排好序(若抵達時間相同則按照 ID 小到大排序)，再一個一個依照順序將 process 完整執行至結束

**RR:** 預先將所有的 process 依照抵達時間小到大排好序，然後取第一個 process 塞入 queue 中，每次做 RR 則取 queue 中的第一個 process 做排程，若此 process 還沒做完則放到 queue 的最後面。而每做完一個 process 則要去檢查再剛剛做完 process 的處理時間(time slice)裡面有沒有新到達的 process，並同時標記此新 process 進入 queue 中(若是剛剛好在處理完 process 之後抵達的 new process，要讓它先進入 queue 中)，再一個一個依照 queue 中的順序輪替直到結束。

**SRTF:** 預先將所有的 process 依照抵達時間小到大排好序，並找出在已抵達的 process 中的最小剩餘執行時間的 process 即可。先從排好序的 process 中找出最小剩餘執行時間的 process，做完之後要隨時檢查是否有新進的 process 且找出最小剩餘執行時間的 process，這樣繼續做到所有 process 都做完。

**PPRR:** 預先將所有的 process 依照 priority 小到大排好序，並將已抵達的 process 塞到待處理 process 的陣列中，同時在陣列中按照 priority 小到大排好順序，每次執行只做待處理 process 的陣列中的第一個 process，並同時記錄此 process 為前一個執行的 process，並塞到他該輪的位置(若是陣列中有相同 priority 的其他 process，則放在同 priority process 的最後一位；若在陣列中沒有相同 priority 的 process，則找到陣列中比目前 priority 大的位置的前一位放)，並繼續做下一個 process。而在執行下一個 process 時檢查前一個做完的 process 是否為同 priority process(需要做 Round Robin)，若是相同，則將上一個 process 跟目前 process 的位置對調並執行；若否，則繼續執行目前的 process。每做完一次都會檢查有沒有新進來的 process，並放進待處理 process 的陣列中(放在該放的位置，判斷方式如上)。

**HRRN:** 預先將所有的 process 依照抵達時間小到大排好序，並找出在已抵達的 process 中的最大 HRRN ratio 的 process 即可。先從排好序的 process 中找出最大 HRRN ratio 的 process，做完之後要隨時檢查是否有新進的 process 且找出最大 HRRN ratio 的 process，這樣繼續做到所有 process 都做完。

最後在每個排程每做完一個 process 的時候計算其 turnaround 與 waiting 時間。

## 三、不同排程法的比較

```

Continue(1) Finish(0) --> 1
Input filename : input1
===== FCFS ===== average waiting --> 14.3333
=====                average turnaround --> 18.2
===== RR ===== average waiting --> 18.4
=====                average turnaround --> 22.2667
===== SRTF ===== average waiting --> 8.06667
=====                average turnaround --> 11.9333
===== PPRR ===== average waiting --> 14.6667
=====                average turnaround --> 18.5333
===== HRRN ===== average waiting --> 11.6
=====                average turnaround --> 15.4667
Time --> 0.008s
Continue(1) Finish(0) --> 1
Input filename : input2
===== FCFS ===== average waiting --> 8.4
=====                average turnaround --> 13.2
===== RR ===== average waiting --> 6.4
=====                average turnaround --> 11.2
===== SRTF ===== average waiting --> 3
=====                average turnaround --> 7.8
===== PPRR ===== average waiting --> 9.4
=====                average turnaround --> 14.2
===== HRRN ===== average waiting --> 8.2
=====                average turnaround --> 13
Time --> 0.008s
Continue(1) Finish(0) --> 1
Input filename : input3
===== FCFS ===== average waiting --> 6.66667
=====                average turnaround --> 24.1667
===== RR ===== average waiting --> 11.6667
=====                average turnaround --> 29.1667
===== SRTF ===== average waiting --> 6.66667
=====                average turnaround --> 24.1667
===== PPRR ===== average waiting --> 12.5
=====                average turnaround --> 30
===== HRRN ===== average waiting --> 6.66667
=====                average turnaround --> 24.1667

```

|                | FCFS    | RR      | SRTF    | PPRR    | HRRN    |
|----------------|---------|---------|---------|---------|---------|
| Waiting (1)    | 14.3333 | 18.4    | 8.06667 | 14.667  | 11.6    |
| Turnaround (1) | 18.2    | 22.2667 | 11.9333 | 18.5333 | 15.4667 |
| Waiting (2)    | 8.4     | 6.4     | 3       | 9.4     | 8.2     |
| Turnaround (2) | 13.2    | 11.2    | 7.8     | 14.2    | 13      |
| Waiting (3)    | 6.66667 | 11.6667 | 6.6667  | 12.5    | 6.6667  |
| Turnaround (2) | 24.1667 | 29.1667 | 24.1667 | 30      | 24.1667 |

(紅色：最長時間，藍色：最短時間)

**RR 與 PPRR** 所需要的等待時間比其他三種排程法的等待時間較久，因為此兩種方法是將相同元素(同 arrival time 或同 priority)的 process 輪替直到做完，當過程中有遇到新的、可以奪取目前輪替的 process 後則進行奪取，所以會導致早到的 process 被拖延、每個 process 的 turnaround 時間被拉長。

**SRTF** 是以等待時間做評估的最佳演算法，因為是選取目前已經抵達的 process 所剩餘的最小時間優先做，所以平均等待時間跟平均 turnaround 時間都會是最短的。

**FCFS** 的平均等待時間與平均 turnaround 都較長，因為當有 process 的 CPU burst 較長且較早抵達的時候，會拖延到之後抵達的 process，所以之後所有的 process 的等待時間都會被延長，等待時間被延長 turnaround 時間也會被延長。

而 **HRRN** 是採取變動優先等級的方式，滾動調整優先等級，所以不會讓 CPU burst 長的 process 永遠拿不到執行權，但是因為是 non preemptive，所以會萬一有早抵達但 CPU burst 較長的 process 出現，會使得所有 process 平均等待時間與平均 turnaround 時間會較長。

#### 四、結果與討論

**SRTF** 的結果為最符合理想 CPU 排程的狀態，平均等待時間與平均 turnaround 時間都會是最短的，但是此排程會造成 starvation，執行時間較久的 process 有可能會永遠拿不到執行權。而 **RR** 所需要的等待時間雖然比其他四種排程法的等待時間較久，但是不會造成 starvation，因為此排程會輪流執行已經抵達的 process，所以不會造成 process 無限延宕的情形。**HRRN 與 FCFS** 也不會造成 starvation，因為 HRRN 會滾動式調整優先等級，response ratio 高的優先處理，而 FCFS 會依抵達時間依序執行。**PPRR** 有機會會造成 starvation，萬一之後有晚抵達但 priority 較小的 process，則會不斷插隊執行，所以可能會造成早抵達但 priority 大的 process 無限延宕的情形。