1. 開發環境

- 作業系統：Windows 10

- 使用軟體：Visual Studio Code

- 使用語言：Python 3.10.1 64-bit

2. 流程與說明

2.1 流程

- 使用者輸入欲載入之檔名（檔案需放在input資料夾中）

- 方法 1：將資料進行FCFS CPU Scheduling

- 方法 2：將資料進行RR CPU Scheduling

- 方法 3：將資料進行SRTF CPU Scheduling

- 方法 4：將資料進行PPRR CPU Scheduling

- 方法 5：將資料進行HRRN CPU Scheduling

- 方法 6：將資料進行方法1-5的所有 CPU Scheduling

- 將結果的waiting time、turnaround time與各排程甘特圖寫入新的檔案

2.1 排程演算法說明

先到先服務（First Come First Served, FCFS）

Arrival time越早的Process越優先取得CPU控制權。

知更鳥式循環（Round Robin, RR）

作業系統會規定CPU Time Slice（Time Quantum），Process享有固定的時間取得CPU控制權，如未能在時間內完成則會被迫放棄CPU，回到Ready的狀態。

最短剩餘時間優先（Shortest Remaining Time First, SRTF）

Process剩餘的工作時間越小，越優先取得CPU控制權。

優先等級知更鳥循環（Priority Round Robin, PPRR）

由優先等級（Priority）較高的Process優先取得CPU控制權，如優先等級一樣則進行知更鳥循環，直到工作結束或更高等級的Process出現。

最高反應時間比率優先（Highest Response Ratio Next, HRRN）

計算Process的反應時間比率（Response Ratio），由最高的優先取得CPU控制權。

3. 使用的資料結構

- list

4. 完成的功能

- 全數完成

5. 不同排程法的比較

平均等待時間 Avg Waiting Time

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input1 | Input2 | Input3 |
| FCFS | 14.34 | 8.4 | 6.67 |
| RR | 18.4 | 6.4 | 11.67 |
| SRTF | 8.07 | 3 | 6.67 |
| PPRR | 14.67 | 9.4 | 12.5 |
| HRRN | 11.6 | 8.2 | 6.67 |

由上表可見無論測資為何，SRTF的平均等待時間和平均工作往返時間皆小於等於其他的排程演算法，但仔細看一下輸出結果後可以發現，在測資一裡，SRTF擁有將近單個Process最長的等待時間（ID=10, waiting time=49），而其他的等待時間很明顯小於該數字一大截，這正是突顯了SRTF演算法優先處理最短工作時間的特性。

當今天出現CPU Burst較長的Process時，可能SRTF就不太適合用來處理，即使他有著最短平均等待時間的稱號（沒有搭配其他機制容易會Starvation）。

相反地，如果今天有較長CPU Burst的Process要處理，FCFS可能因為花太久時間處理，造成後面較短的作業無法被處理，而出現護衛現象（Convoy Effect）。

而作為這兩者極端情況間，HRRN則扮演了一個很權衡的角色，除了有Response Ratio的機制可以避免Starvation的狀況之外，也不會總是讓較長CPU Burst的Process堵住整個CPU的占用，導致後面較短CPU Burst的Process塞住。

PPRR與RR則是差在是否有優先順序，兩者都是有Time Slice的機制，確保大家共享CPU的固定時間片段，可以說是最公平的演算法。不過要注意的地方則是PPRR可能也會有Starvation以及Content Switch造成的延遲，如果時間片段過短，容易因過於頻繁切換導致整體速度變慢，而設的太長又可能與FCFS太過相近。

平均工作往返時間 Avg Turnaround Time

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input1 | Input2 | Input3 |
| FCFS | 18.2 | 13.2 | 24.17 |
| RR | 22.27 | 11.2 | 29.17 |
| SRTF | 11.93 | 7.8 | 24.17 |
| PPRR | 18.53 | 14.2 | 30 |
| HRRN | 15.47 | 13 | 24.17 |

至於平均工作往返時間基本上就是等待時間再加上工作時間，可以看到SRTF在測資一的單個Process最長往返時間來到了驚人的57（ID=10, turnaround time=57），然而這邊有個有趣的現象就是在PPRR中也有個原本為最長等待時間的Process（ID=7, waiting time=55），但在Turnaround Time時卻輸給了SRTF演算法中的Process成為第二長（SRTF: ID=10, turnaround time=57 PPRR: ID=7, turnaround time=56），這也正說明在PPRR排程演算法中Priority的重要性！

6. 結果與討論

總觀所述，可以發現各排程演算法都有其特色在，依據不同情況而有不同優劣；這讓我聯想到之前讀計算機網路時，也常會出現根據不同情況而有不同對應機制的通訊協定等等，其中可能還會遇到許多像排程法中Starvation的致命危機；然而因應這些不同的情況而下去不斷改良演算法可以說是人類在資訊領域中的一大使命，因為永遠都沒有最好的演算法，只有最適合現代情況的最佳演算法！