作業系統HW2

實作人 10827240 吳振瑋

1. 開發環境

OS:Windows11

撰寫平台:Visual Studio Code

使用語言:python

版本3.9.12

1. 實作方式及流程
   1. FIFO

使用lsit當作頁框，將新的頁放入list[0]的位置，如果頁匡滿了將list最後一個位置移除

* 1. LRU

使用lsit當作頁框，將新的頁放入list[0]的位置，當頁被重新使用時就將該頁更新到list[0]的位置，如果頁匡滿了將list最後一個位置移除

* 1. LFU+LRU

使用lsit當作頁框並用dict來儲存頁的使用次數，將新的頁放入list[0]的位置，當頁被重新使用時就將該頁更新到list[0]的位置並更新該頁的使用次數，如果頁匡滿了將list中使用次數最少最後面的移除

* 1. MFU+FIFO

使用lsit當作頁框並用dict來儲存頁的使用次數，將新的頁放入list[0]的位置，當頁被重新使用時就更新該頁的使用次數，如果頁匡滿了將list中使用次數最多最後面的移除

* 1. MFU+LRU

使用lsit當作頁框並用dict來儲存頁的使用次數，將新的頁放入list[0]的位置，當頁被重新使用時就將該頁更新到list[0]的位置並更新該頁的使用次數，如果頁匡滿了將list中使用次數最多最後面的移除

1. 不同排程法比較

Waiting time

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | FCFS | RR | SRTF | PPRR | HRRN |
| Input1.txt | 14.3 | 18.4 | 8 | 14.7 | 11.6 |
| Input2.txt | 8.8 | 6.4 | 3.0 | 9.4 | 8.2 |
| Input3.txt | 6.7 | 11.7 | 6.7 | 12.5 | 6.7 |
| Input4.txt | 14.3 | 14.9 | 8.01 | 14.4 | 11.6 |
| Input5.txt | 41.33 | 41.3 | 18.8 | 74.5 | 32.6 |

Turnaround time

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | FCFS | RR | SRTF | PPRR | HRRN |
| Input1.txt | 18.2 | 22.3 | 11.9 | 18.5 | 15.5 |
| Input2.txt | 13.2 | 11.2 | 7.8 | 14.2 | 13 |
| Input3.txt | 24.1 | 29.2 | 24.2 | 30 | 24.2 |
| Input4.txt | 18.7 | 18.7 | 11.9 | 18.3 | 15.5 |
| Input5.txt | 49 | 49 | 26.6 | 82.2 | 40.3 |

1. 結果討論

由上圖可觀察到SRTF的平均waiting time表現最好，觀察waiting time的時候會發現有很多process是沒有等待的，SRTC不斷在盡快清理queue所以不會有很多人在等，所以平均waiting time就較其他排程法低，但是當process的CPU Burst很高時process會進入很長的等待時間。

若是觀看分布狀態的話RR、SRTF、HRRN均不太會出現極端狀況，其中又以HRRN平均waiting time較低，我認為的原因是HRRN會根據process的CPU Burst來調整每個process 會經歷的waiting time，CPU Burst小waiting time會跟著小，CPU Burst很大時也不會出現無限等待。

PPRR則不列入waiting time討論，因為其waiting time 還要按照priority實作。