第三次作業

Page Replacement 方法

1. 開發環境

Window 10

使用visual Studio Code 環境

程式語言Python

1. 實作方法、資料結構及運作流程

資料結構 : 一個count用來計數，一個放Page Reference String

Page\_fault = 0 # 紀錄發生幾次Page fault

Page\_Replaces = 0 # 紀錄發生幾次Page Replaces

class info

|  |  |
| --- | --- |
| count | Page Reference String |

Queue

|  |  |  |  |
| --- | --- | --- | --- |
| info |  |  |  |

結構如圖

基本流程都一樣:

判斷有沒有發生page fault

判斷有沒有發生Page Replaces

* 1. FIFO

先判斷有沒有發生page fault，有就Page\_fault次數加一，再判斷queue是否已滿(Page Replaces)，沒有就將新的Page Reference String插入queue最前面；有就將Page\_Replaces次數加一， FIFO先進去先出去的判斷方式，找到該Page Reference String抽換掉，將新的新的Page Reference String插入queue最前面。

* 1. LRU

與FIFO大致相同，僅需修改兩部分。第一，沒有發生Page\_fault時，代表queue內已有該值，要將Page Reference String更新。第二，queue已滿(Page Replaces)，要使用LRU方法代換，此方法是找最不常被參考、修改的Page Reference String做抽換。

* 1. LFU + FIFO

(此方法會使用到count計數)

先判斷有沒有發生page fault，有就Page\_fault次數加一；若是沒有，就找到重複被參考的Page Reference String將count加一，再判斷queue是否已滿(Page Replaces)，沒有就將新的Page Reference String插入queue最前面；有就將Page\_Replaces次數加一，使用LFU + FIFO判斷方式，先找count最小的Page Reference String，如果一樣就找最先進queue的Page Reference String，找到該Page Reference String抽換掉，將新的新的Page Reference String插入queue最前面。

* 1. MFU + FIFO

大致與LFU + FIFO相同，僅需修改queue已滿(Page Replaces)，判斷條件使用MFU + FIFO，先找count最大的Page Reference String，如果一樣就找最先進queue的Page Reference String

* 1. LFU + LRU
  2. 大致與LFU + FIFO相同，僅需修改兩部分。第一，沒有發生Page\_fault時，代表queue內已有該值，要將Page Reference String更新，同時count次數加一。第二，queue已滿(Page Replaces)，判斷條件使用LFU + LRU，先找count最小的Page Reference String，如果一樣就找最不常被使用的Page Reference String。

1. 分析不同方法之間比較

(以下為實驗結果input2.txt)

* 1. Page Fault次數

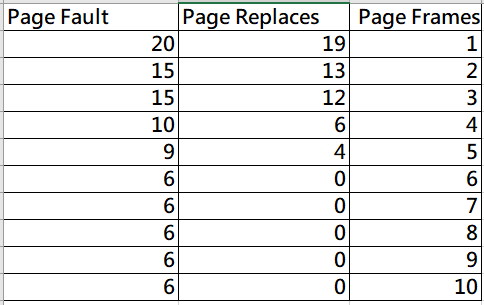
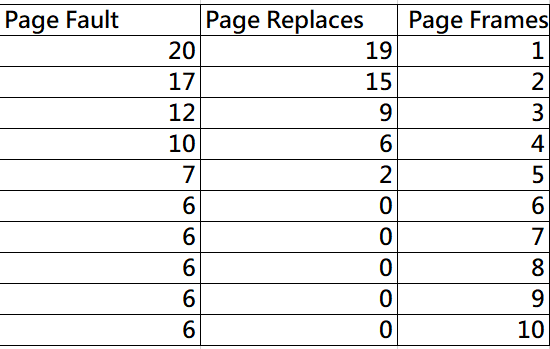
所有方法在超過Page Frame 6之後次數皆不在減少，維持在Page Fault次數6

Page Frame越大越不容易發生Page Fault，但是有一定上限

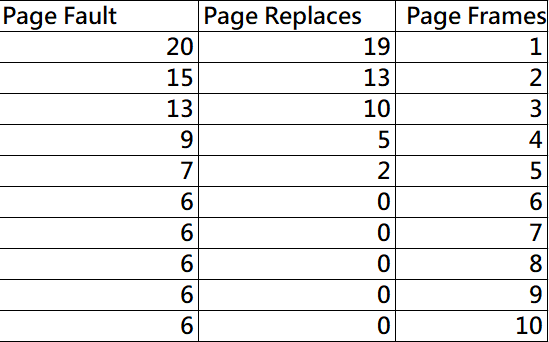
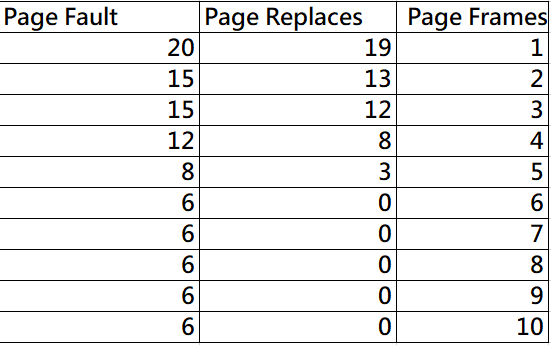
* 1. Page Replace 次數

所有方法在超過Page Frame 6之後次數皆不在減少，維持在Page Replace次數0

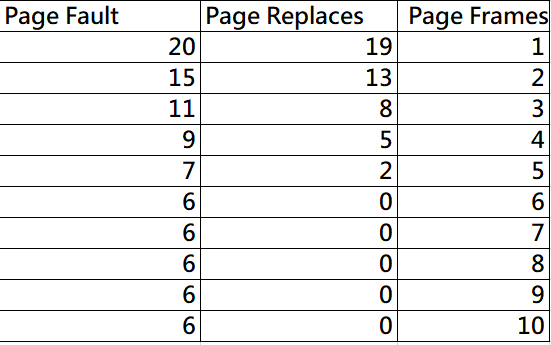
Page Frame越大越不容易發生Page Replace

FIFO實驗結果 LRU實驗結果

LFU+FIFO實驗結果 MFU+FIFO實驗結果



LFU+LRU實驗結果

1. 結果與討論

(以下為實驗結果input1\_method6.txt)

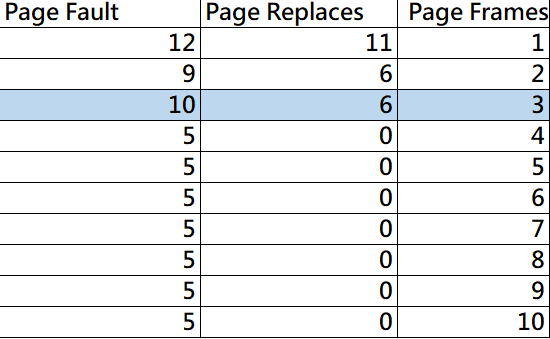
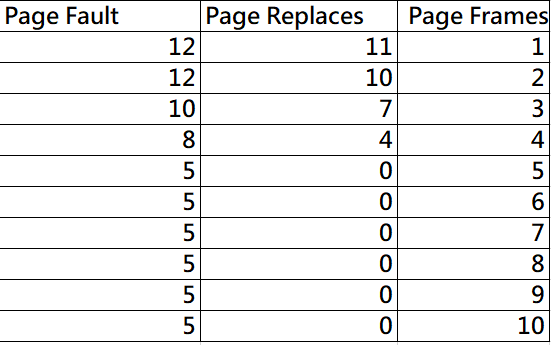
* 1. 實驗數據與發現[畢雷笛反例]
  2. FIFO與MFU+FIFO會發生畢雷笛反例

(對照下圖)

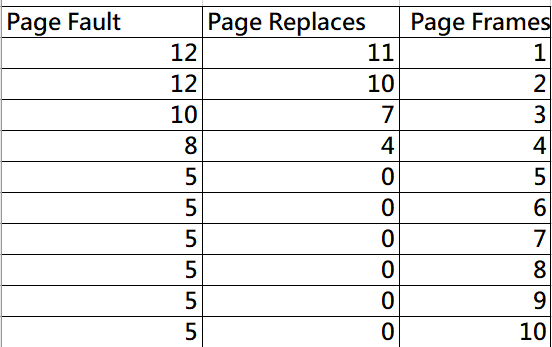
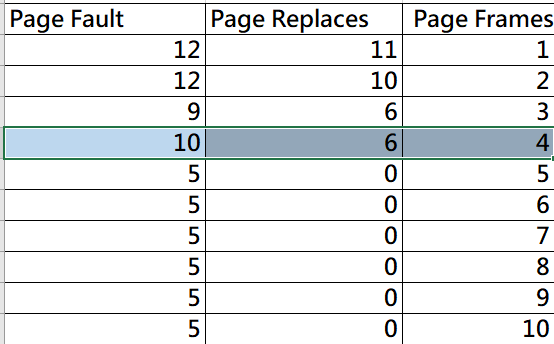
當Page Frame = 4 時， page fault = 9

當Page Frame = 4 時， page fault = 10 [Page Fault不減反增]

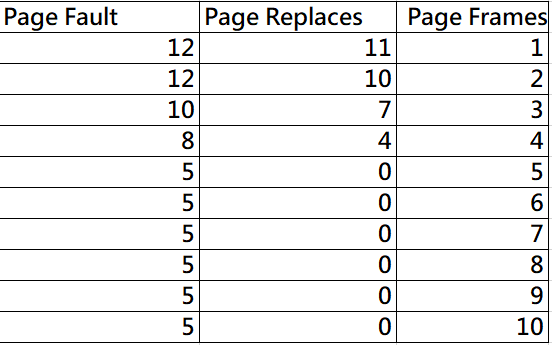
畢雷笛反例:所以指的是增加Page Frame反而造成更多的page fault和page Replace

FIFO實驗結果 LRU實驗結果

LFU+FIFO實驗結果 MFU+FIFO實驗結果



LFU+LRU實驗結果