作業系統　OS HW3　Page Replacement

I.開發環境

資訊三乙 10827234 彭桂綺

開發環境:Windows 11 (21H2)

開發工具:visual studio code [1.67.2(2022.05.17)]

使用語言:C/C++

II.實作方法及流程

首先我會選擇在 windows 以 c/c++ 實作此次作業是因為我最習慣的環境及語言就是如此,如果遇到 bug 也熟悉如何 debug。

總的來說,我將 5 個方法用 4 個 class 分別包裝,各自擁有自己的 pageFault, pageReplaces, counter 和 pageReplace 變數,方便呼叫執行。

- FIFO

  依照 reference string 的順序,先去 page frame 中找找看目前的 page 有沒有在裡面,有的話不用做事,如果沒有的話則發生 page fault,然後將最舊的 page 置換出 page frame ,page replace + 1,再將 目前 page 丟進 queue(page frame)中。

- LRU

  跟 FIFO 差不多的作法,差別在於當某個 page 已經在 page frame 中,且被參考到,就會將此 page 換到 queue(page frame)的最前面(最新的)位置。其餘步驟跟 FIFO 相同。

- LFU + LRU

  此方法每個 pgae 被參考時,counter+1,當找不到此 page,發生 page fault 時,會將 counter 值最小的置換出。當要置換時有多個 page 的 counter 相同,則依照 LRU 的 規則處理。

- MFU + FIFO

  此方法被參考時,counter+1,當找不到此 page,發生 page fault 時,會將 counter 值最大的置換出。當要置換時有多個 page 的 counter 相同,則依照 FIFO 的 規則處理。

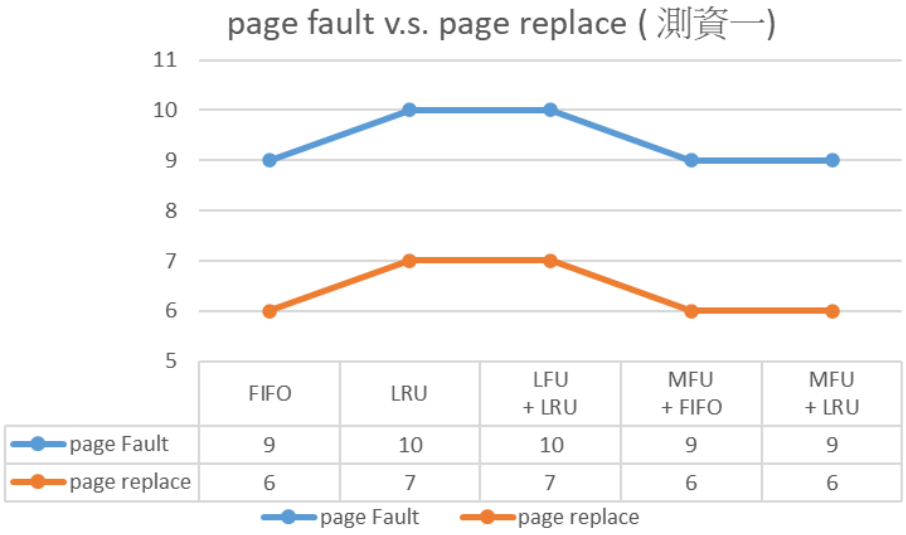
- MFU + LRU

  此方法跟上述相同,唯獨當要置換時有多個 page 的 counter 相同,則依照 LRU 的 規則處理。

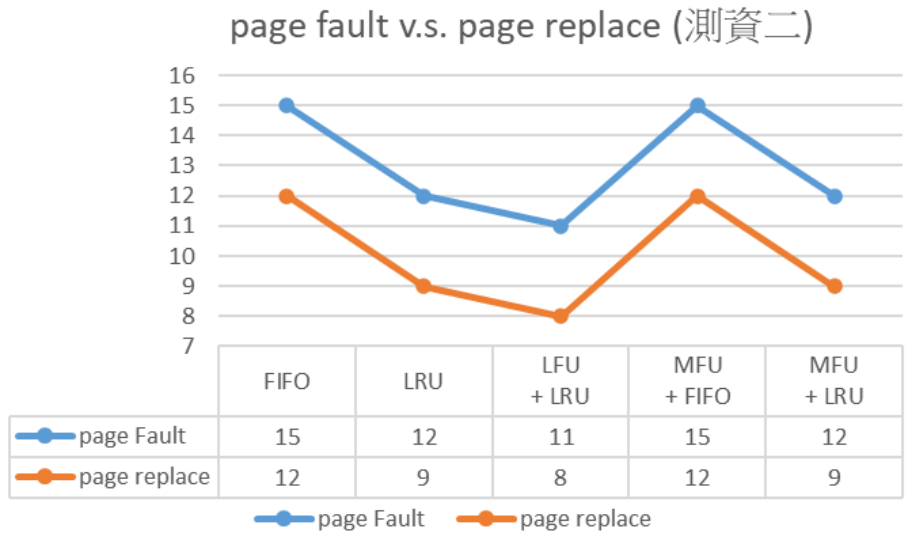
III. 不同排方法的比較

- Page Fault 次數 && Page Replace 次數

  這兩個項目還是放在一起比較會比較方便。原先以為不同的排程法,其中的 page Fault 及 page replace 次數的趨勢會差不多,但是實際上不盡然。我們直接看圖表。



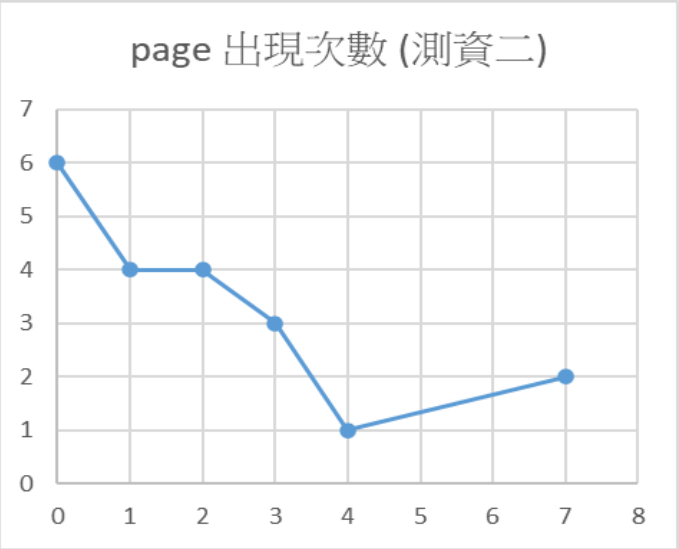
	FIFO	LRU	LFU + LRU	MFU + FIFO	MFU + LRU
page Fault	9	10	10	9	9
page replace	6	7	7	6	6



	FIFO	LRU	LFU + LRU	MFU + FIFO	MFU + LRU
page Fault	15	12	11	15	12
page replace	12	9	8	12	9

光看這兩個測資跑出來的圖表,完全看不出來那個頁置換法是最好的(optimal)。真的是百思不得其解,到底為甚麼會有這樣的差別,既然方法都一樣,page frame 數量也一樣,那問題就一定出在 page reference string 。

以這兩個測資來說,應該在於 LFU 及 MFU 之間的 差異,



page	次數
1	3
2	3
3	2
4	2
5	2
平均出現次數	2.4

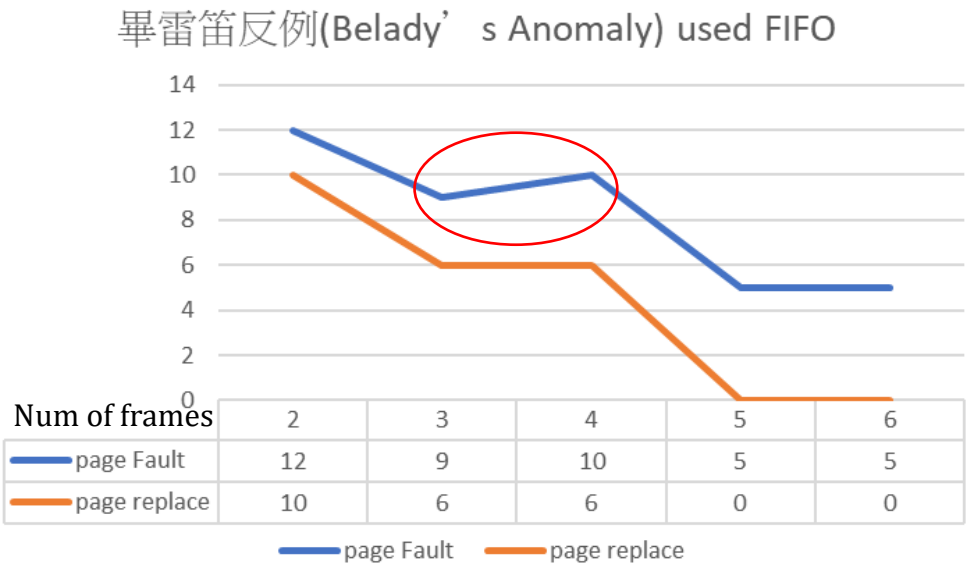
page	次數
0	6
1	4
2	4
3	3
4	1
7	2
平均出現次數	3.34

比較不嚴謹的觀察是,當 page 的平均出現次數高的時候, 使用 MFU 的方法就會造成更高的 Page Fault 次數 && Page Replace 次數;相反的, 當 page 的平均出現次數比較低,Page Fault 次數 && Page Replace 次數就會減少。

IV.結果與討論

- 畢雷笛反例(Belady’s Anomaly)

  理想情況下,當增加 Page frame,預想較少發生 page fault,但是有一個電腦科學家 Belady 發現了一串 reference string, 使得增加 page frame 反而造成更多的 page fault 及 page replacement, 這就是所謂的畢雷笛反例,而那串神奇的字串就是測資一的數據”123412512345“。



Num of frames	2	3	4	5	6
page Fault	12	9	10	5	5
page replace	10	6	6	0	0

上圖即可清楚看出 增加 page frame 反而造成更多的 page fault。

- 有趣的發現

  不知道是巧合還是其中有種規律,所有的方法中,page fault = Page Replace + page frame。我有試著放不同的 input 觀察結果,像是同組 reference string 給予不同的 page frame ,或是將前面幾個 page 重複數次(為了避免還未出現過的 page 在前期就 page fault) ……。但是到繳交作業為止都沒有想到怎麼證明這個現象。