OS Project2 書面報告

資訊三甲 11027164 趙怡儒

一、 開發環境 ：Visual Studio Code 使用 Python

二、 實作方法和流程 ：

每個 methon 都有三個 list，pageframe 用來記錄要輸出 Page Frames，Ex. {“7”,”07”,”107”,...}。pagefault 用來記錄要輸出 Page Fault，ex.{“F”,”F”,””,”F”...}，其中空字元代表那一個時間未發生 pagefault。pagelist 則用來記錄每個在 pageframe 裡面的 page 的資訊，其中存在此 list 中的 page 是一個 class，有 id、time、counter 三種資訊。將 input 檔案切好放到 methon、frame_size、page_string 中開始執行指定的方法

1. FIFO : FIFO 跟名字一樣，先到的先出去，所以在要置換的時候就把待在 frame 最久的那一個 page 移除，也就是 page.time 最小的那個，把最小的那個設定為 target，從 pagelist 跟 pageframe 中移除，此方法還用不到 counter，所以紀錄時都直接先用 0 存，
2. LRU:與 FIFO 大致上相同，不同的地方是，FIFO 設定的 page.time 是這個 page 第一次進到 frame 中的時間，再次被 reference 的話並不會更新，LRU 是只要此 page 又被 reference 到它的時間就需要更新，換句話說，就是換掉最久沒有被 reference 的 page。此方法也還用不到 counter，所以記錄時也先用 0 紀錄。
3. LFU+FIFO: 與前兩個方法不一樣的地方是，每當 page 被 reference 的時候，就會將其 counter 值+1,如果 page 被置換出 pageframe 後，直接從 pagelist 中移除它，也就是下一次再進來 pageframe 的話 counter 就從 1 開始。會被當成犧牲者移出的就是 counter 最小者，若 counter 有不止一個最小，則用 FIFO 的方法看誰的待在 frame 的時間最久，設為 target 被從 pagelist 移除它。
4. MFU+FIFO:與 LFU+FIFO 需要參照的變數一樣，只是方法不同，會被當成犧牲者移出的是 counter 最大者，若 counter 有不止一個最大，則用 FIFO 的方法看誰的待在 frame 的時間最久，設為 target 被從 pagelist 移除它。
5. LFU+LRU:與 LFU+FIFO 大致上相同，但是在 counter 不只一個最小的時候，會依照 LRU 的規則，是換掉最久沒有被 reference 的 page。
6. 方法中共同用到的 function 或 class

(1) Page

用來記錄每個在 pageframe 中的 page 資料有 id、time 和 counter

```
class Page:
    def __init__(self, id, time, counter):
        self.id = id
        self.time = time
        self.counter = counter
```

(2) checkin

用來確定目前被 reference 的 page 是否已經在 pageframe 中

```
def checkin(item,list):
    return item in list
```

(3) padefaultlength

用來數 pagefault 這個 list 中”F”出現的次數作為 Page Fault 發生的次數

```
def pagefaultlength(pagefault):
    count = 0
    for i in pagefault:
        if i == "F":
            count += 1
    return count
```

三、 不同方法之間的比較

Page Fault 的次數比較

	FIFO	LRU	LFU+FIFO	MFU+FIFO	LFU+LRU
Input1	9	10	10	9	10
Input2	15	12	13	15	11

Page Replace 的次數比較

	FIFO	LRU	LFU+FIFO	MFU+FIFO	LFU+LRU
Input1	6	7	7	6	7
Input2	12	9	10	12	8

四、 結果與討論

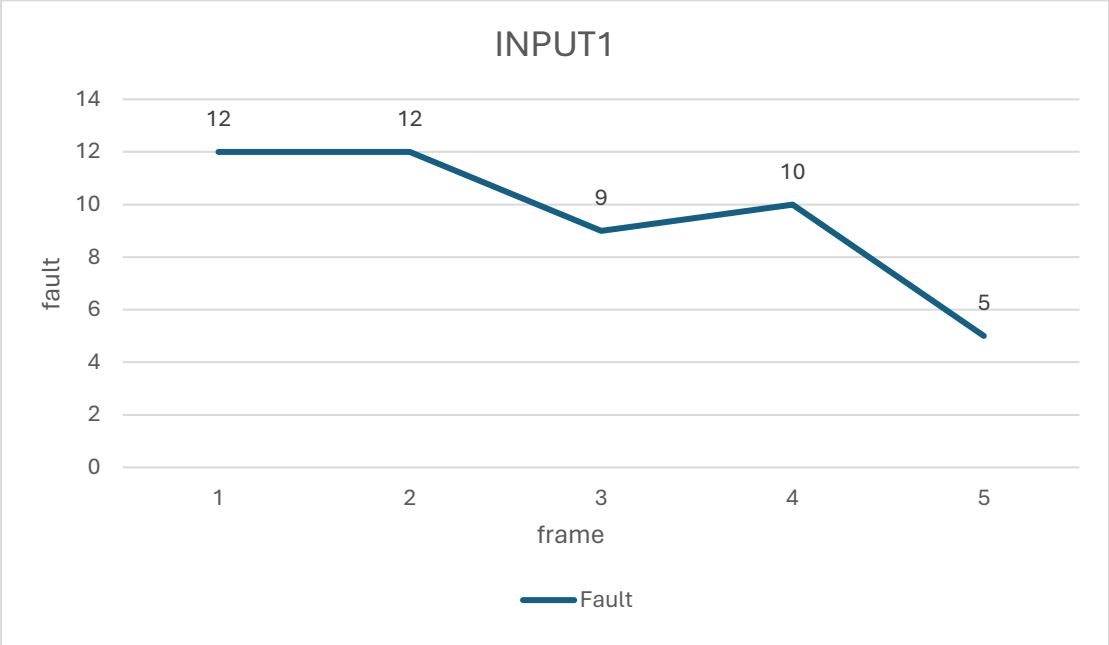
兩個表格可以看到 page fault 的次數一定大於 page replacement，原因是發生 page fault 不一定是發生 page replacement 的時候，因為一開始，page frame 還沒滿的情況底下，被 reference 的 page 不在 frame 裡面也不用置換，所以會有 fault 但不會有 replacement，又因為這種情況只會在 frame 沒滿的情況下發生，所以 fault 跟 replacement 的差值會剛剛好是 frame_size，後續每次 fault 都會伴隨著 replacement。

再來是五種方法的比較，input1 五種方法都差距只有 1，我想是因為 frame_size 有三個，且 page 只有五個，然後 page_string 又比 input2 短，還沒有大差距就結束了。Input2 的差距明顯 FIFO 跟 MFU+FIFO 比起其他人都多了一點，我在想是不是因為一直重複被 reference 的 page 也一直被換出去的，所以 fault 跟 replacement 都變多。

畢雷笛反例：

用 input1 實測老師上課提到的畢雷笛反例

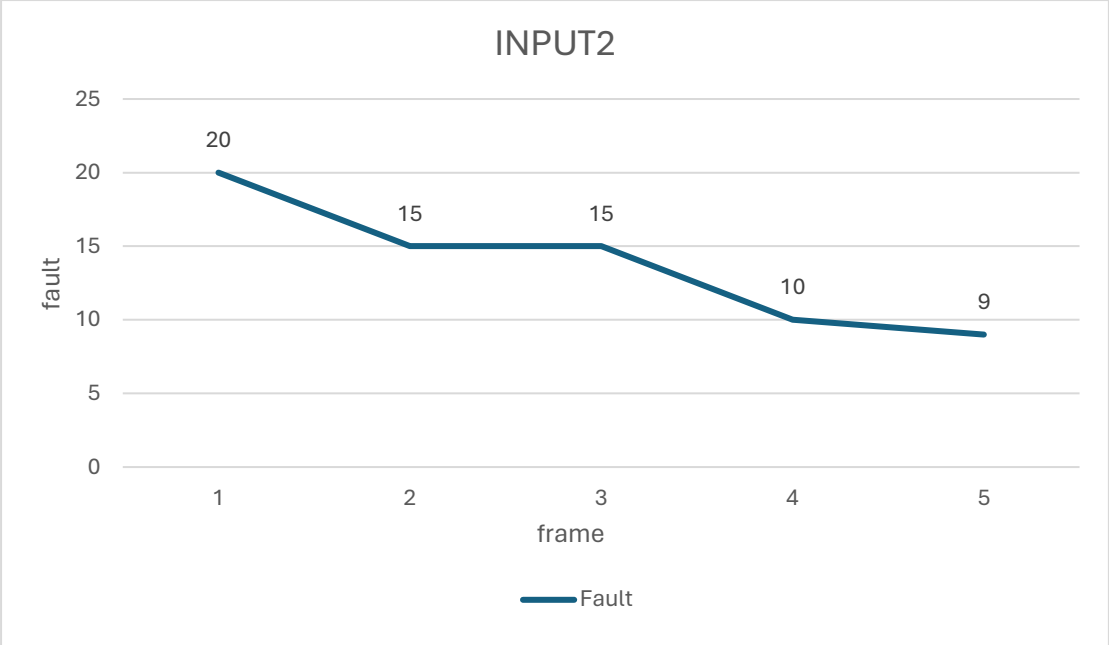
	Frame = 1	Frame = 2	Frame = 3	Frame = 4	Frame = 5
Fault	12	12	9	10	5
Replace	11	10	6	6	0



在 Frame = 4 的時候出現了畢雷笛反例。

用 input2 實測老師上課提到的畢雷笛反例

	Frame = 1	Frame = 2	Frame = 3	Frame = 4	Frame = 5
Fault	20	15	15	10	9
Replace	19	13	12	6	4



沒有出現了畢雷笛反例。

結論：不一定會出現