OS Project2 書面報告

資訊三甲 11027164 趙怡儒

1. 開發環境 : Visual Studio Code 使用 Python
2. 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
7. Page

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

一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

1. checkin

用來確定目前被reference的page是否已經在pageframe中

一張含有 文字, 字型, 螢幕擷取畫面, 圖形 的圖片

自動產生的描述

1. padefaultlength

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

一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

1. 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ˋ 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。