**作業系統**

**Project3**

Paging Manegement

資工三甲

彭義翔

10527142

**開發平台**

Python 3.6.3 |Anaconda, Inc.| (default, Oct 15 2017, 03:27:45) [MSC v.1900 64 bit (AMD64)] on win32

**使用開發環境**

Window10 1803 64-bit

**使用的程式語言**

python

**說明你的程式設計**

1. **功能**

**輸入兩行數字一行是要的頁框大小，第二行是頁進來的順序，輸出依照頁進來的順序每一步都印出來，並且如果有BIT的判斷會一起印出來，還有如果發生page false會印”F”，最後跑完的時候會印出來發生Page Fault, Page Replaces, Page Frames 的次數，總共有六組分別對照FIFO.LRU.Additional reference bit.Second change .Least/Most Frequently use**

1. **流程**

**每一步最完都會印出來**

**最後再把PAGE全部資訊那行印一遍**

**FIFO直到要進來的頁都處裡完 如果PAGE是空的直接印第一個人出來Fault+1 否則找看看有沒有已經近來過 有近來過不管他 沒進來過的話Fault+1再看LIST有沒有滿 有滿就丟掉最後一個人Replaces+1加新的進來**

**LRU直到要進來的頁都處裡完 如果PAGE是空的直接印第一個人出來Fault+1 否則找看看有沒有已經近來過 有近來過要把她TIME更新最新(我的作法是放到最前面) 沒進來過的話Fault+1再看LIST有沒有滿 有滿就丟掉最後一個人Replaces+1加新的進來**

**ARB這裡多一個BITS的字串00000000共有和頁數相同多組 分別對應各頁 每次都要跟著頁跑。直到要進來的頁都處裡完 如果PAGE是空的直接印第一個人出來Fault+1 BIT放1到最前面 否則找看看有沒有已經近來過 有近來過那個人的BIT放1到最前 其他人都是0 沒進來過的話Fault+1再看LIST有沒有滿 有滿就找最小的BIT丟掉Replaces+1 加新的進來(那個人的BIT放1到最前 其他人都是0) 沒滿也要SHIFT**

**SC這裡多一個BITS的位元1/0共有和頁數相同多組 分別對應各頁 每次都要跟著頁跑。直到要進來的頁都處裡完 如果PAGE是空的直接印第一個人出來Fault+1 BIT1放到最前面 否則找看看有沒有已經近來過 有近來過要把她TIME更新最新(那個人的BIT1放到最前) 沒進來過的話Fault+1再看LIST有沒有滿 有滿就找最後的BIT = 0丟掉(如果是1的話改成0放到最前面繼續找)如果每個人都1 就會丟最後一個BIT改成第一人1 其他0 Replaces+1 加新的進來(那個人的BIT1放到最前) 沒滿直接放**

**LF/MF 大同小異 唯一差別是找頁的時候要找換的次數最大或最小 其餘條件都一樣 這裡多一個BITS的計數器 共有和頁數相同多組 分別對應各頁 每次都要跟著頁跑。直到要進來的頁都處裡完 如果PAGE是空的直接印第一個人出來Fault+1 BIT0 否則找看看有沒有已經近來過 有近來那個人的BIT+1 沒進來過的話Fault+1再看LIST有沒有滿 有滿就找最後的小/大頻率丟掉 如果每個人都一樣 就會丟最後一個Replaces+1 加新的進來(那個人的BIT=0) 沒滿直接放**

1. **data structure**

**頁的list**

**每個任務分別要自己用的計算東西(BIT的list)**

**其餘皆為區域變數**

**未完成的功能**

基本上作業要求的都完整的處理好也表達出來