武林秘笈-想成為武林大師ㄇ

感想

整份作業難度不高，但很複雜，因為需要考慮到很多東西的牽扯
我大部分是用struct搭配array和linked list實作
選擇array是需要random access需要，但也可以不用
然後有部分東西我用struct in struct，但是是因為找不到更好的辦法

介紹

程式剛開始的時候，TLB及Page Table都為空，且Physical Mem皆無使用
此時的Free-Frame List為所有Frame的集合，且Replacement List為空
所以接下來的前幾個輸入一定都是 TLB miss&Page fault 然後慢慢建立起TLB和Page Table

TLB

1. VPN: 即輸入Reference(A, 1)中的後值
2. PFN: 即對應的Physical Frame的位置
3. 當換process時，要全部重新設置

如果滿了就會用到下面的演算法：

- Random: 不解釋
- LRU: 實踐方法很多種，選個自己喜歡的

沒滿就塞空位

Page Table

1. PNF/DBI：當今天在physical mem中時，紀錄Frame的位置，當在Disk中時，紀錄Disk的位置
2. Reference: 配合Clock algo（即二次機會演算法）時使用，若沒用到，可以放空，建議跟FIFO一起寫
3. Present: 代表是否在Disk中，若是，則設為1，否則相反

Free-Frame List

基本上只有第一次輸入會用到，剛開始建好就好，因為之後就算有page fault釋出空間也會瞬間被拿走

Replacement List

要搭配Page Replacement policy，可用circular link list或single

- Global: 即所有的process擁有的frame皆可當作victim page，可用一條link list建立就好
- Local: 只有當前的process擁有的才能當作victim page，要想辦法建立很多條link list各自儲存page資訊

Page Replacement policy

當有victim page被洗入disk時，要注意新拿到的page也要放入replacement list中

- FIFO: 不解釋，反正如果是按照順序建立的，一直把第一個拔走就好
- Clock: 即二次機會演算法（https://ithelp.ithome.com.tw/articles/10208696 (https://ithelp.ithome.com.tw/articles/10208696)

Process

假設輸入為 Reference(A, x)

1. 當TLB Hit時，即尋找的資料在Physical mem中 =>將答案寫入txt
2. 當TLB Miss = >也就是當TLB中找不到對應的vpn，若：
    - Page Fault: 即不在page table中
        - 若有Free-Frame: 將Free Frame第一個Frame y拔除，Page Table的第vpn個的pfn改成y(填上page table的空缺)，將Frame x設置成Process A擁有
        - 沒有Free-Frame: 用(FIFO/Clock)選擇victim page(Ex: 假設這裡為Page 3即代表Page Table中vpn為3的地方)，將其對應的Frame洗入disk最低位置（要自己記，會用到），將page table的第 x 個的pfn改成page 3的pfn，設置reference bit為1及present bit為0
            - 洗進去的disk的page 3，其pfn要設成disk的位置，presen設為1
            - 將page 3對應的Frame設成Process A擁有
            - 修改replacement list
    - Page Hit: 就hit，不解釋

- 只要TLB Miss，不管Page有沒有hit，TLB都要重新填寫，若TLB還有位置，則寫入，若沒位置，用Random/LRU替換，然後要將Reference(A, x)再跑一次，此次必定TLB Hit

EMAT and Page fault ratio

PPT公式有給，基本上Page fault要小於0.5，因為TLB refill後也要計算（寄信問過助教ㄌ），所以1次Page fault必定有一次TLB Hit

Document

寫每一個演算法的優缺點