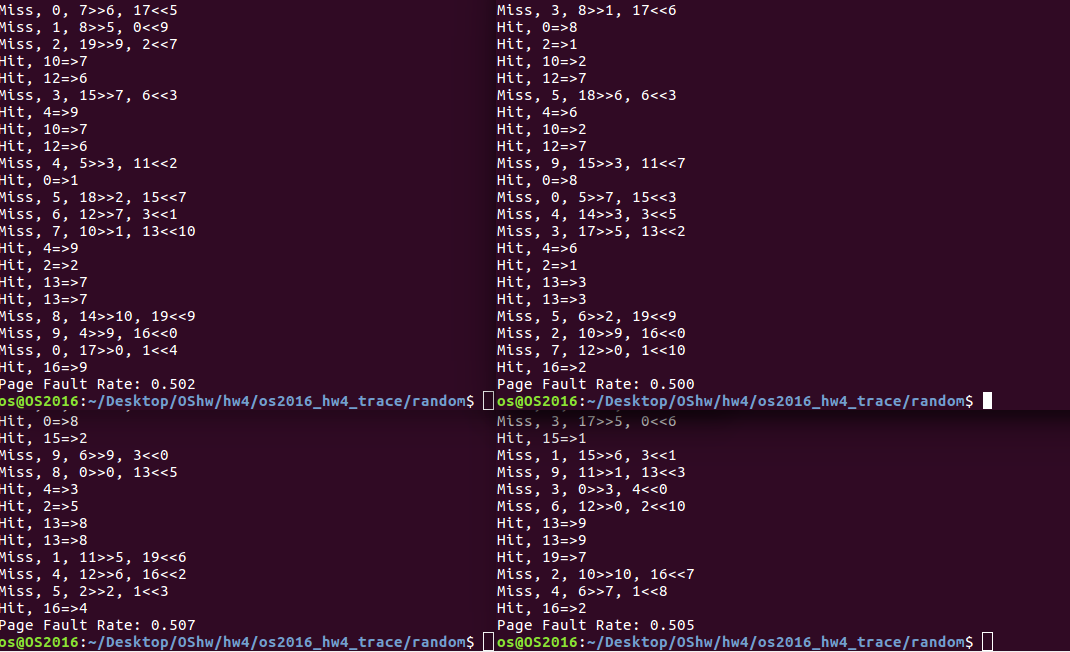
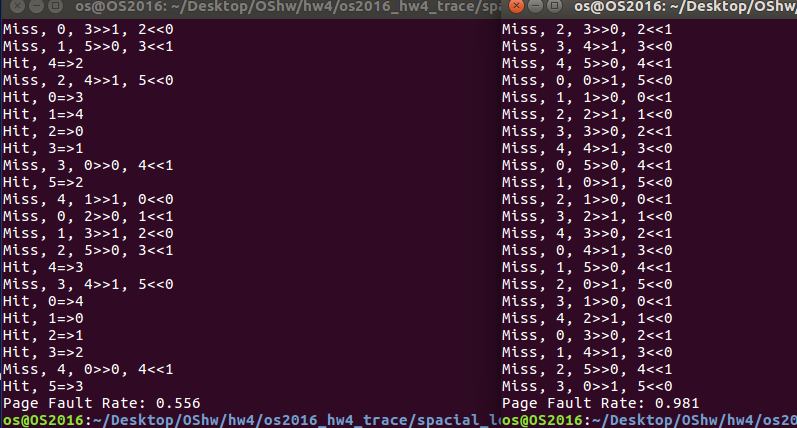
下圖(一)為random資料夾中的5000筆data的測資，fifo(左上)、lru(右上)、2次random(下)，測資為random產生，在分布上很平均，若相同的frames被reference的間隔時間較短，lru的替換策略上較佔優勢，反之，fifo較佔優勢，因為fifo有可能剛好把剛reference過的page-out，而下次又很久才會reference到，把空間留給了其他frames，在random的測資中，兩種情況應幾乎是平均發生，因此fifo與lru在這邊看不太出區別。而random的replacement中，有50%的機率會hit(因為有1/2的frames在virtual page中)，因此random replacement的模擬也都差不多是0.5的page fault rate。



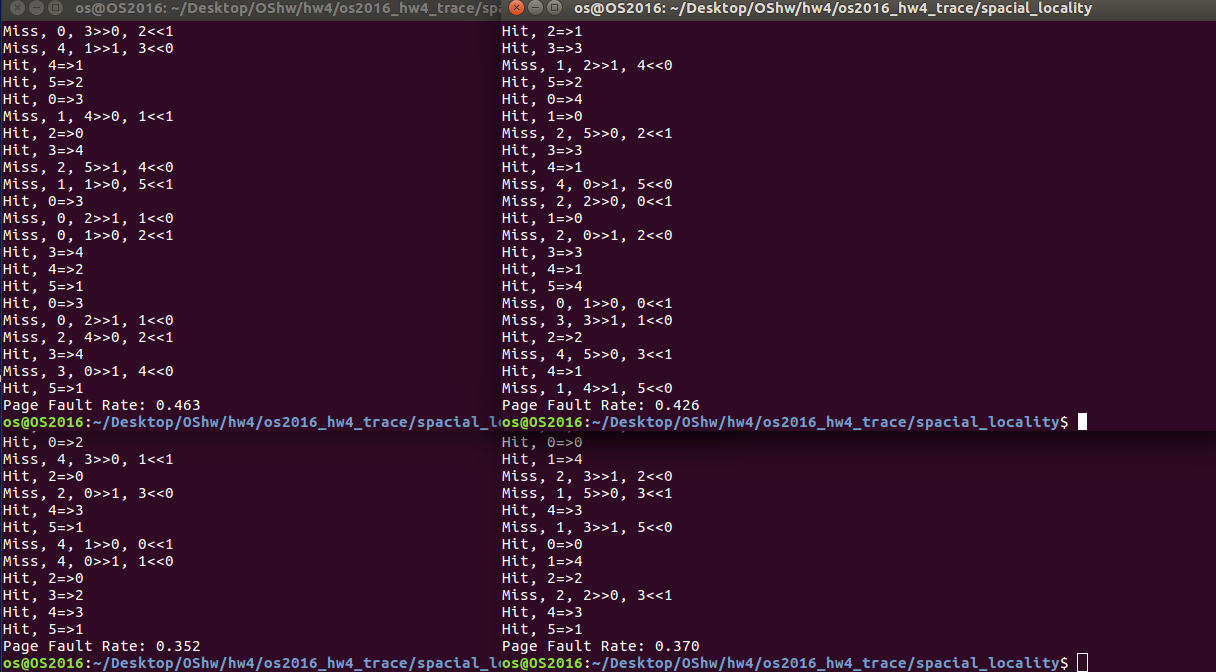
圖(一)：random測資，fifo(左上)、lru(右上)、2次random(下)

下圖(二)為spacial\_locality資料夾中的fifo\_worse(左)與lru\_worse(右)的比較，檢查input之後發現測資是完全一樣的，試著自己trace了一下，幾乎是6個frames依序輪流被reference到，而只有5個virtual pages，如果使用lru的方式，經歷過5次page replacement之後才又reference到自己，那自己已經被page-out了，因此幾乎都是Miss，而在fifo的情況下，即使自己才剛剛被reference到，但替換的位置又輪替到這邊後，還是得把自己page-out，不過因為此筆測資中，自己被reference後要過很久才會再reference自己一次，因此用fifo的替換方式，把自己page-out之後反而留下了待會會被reference的frames，因此提高了很多的hit次數。



圖(二)：special\_locality/fifo\_worse(左) vs lru\_worse(右)

下圖(三)為spacial\_locality資料夾中的random\_worse測試4次，因virtual page numbers/最常使用的frames大於1/2，因此hit的機率應大於1/2，導致page fault rate幾乎都是小於1/2的，在特殊的例子下，random的replacement反而得到最好的結果。



圖(三) special\_locality/random\_worse

**優缺點比較**

FIFO：若各frames reference的間隔時間較長，用fifo比lru更有機會把未來最慢被reference到的page-out，騰出空間。但若reference間隔時間較短，fifo則更可能把未來即將reference到的page-out，但大部分情況下是較平均的。

LRU：與fifo相反，若各frames reference的間隔時間較長，則LRU就會把未來即將被reference到的page-out，反之，LRU會一直保留剛使用過的，因reference間隔較短，最常使用的frames幾乎一直會被保留在virtual pages中，不會被page-out，可使reference他們時hit率提高

RANDOM：不論測資如何，hit的機率只依據frames和pages的數量，不受特殊測資的影響，在其他replacement都難以處理的情況下，random replacement是一個可以考慮的選擇。