姓名：鄭琇櫻

系級：資工 111

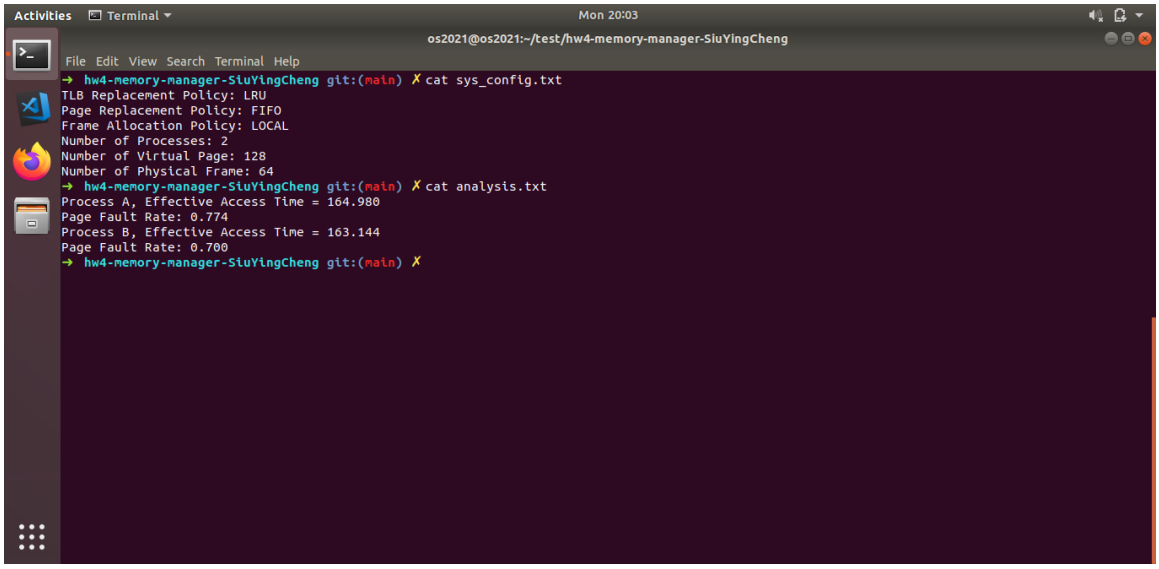
學號：I84077010

HW4：Algorithm 分析文件

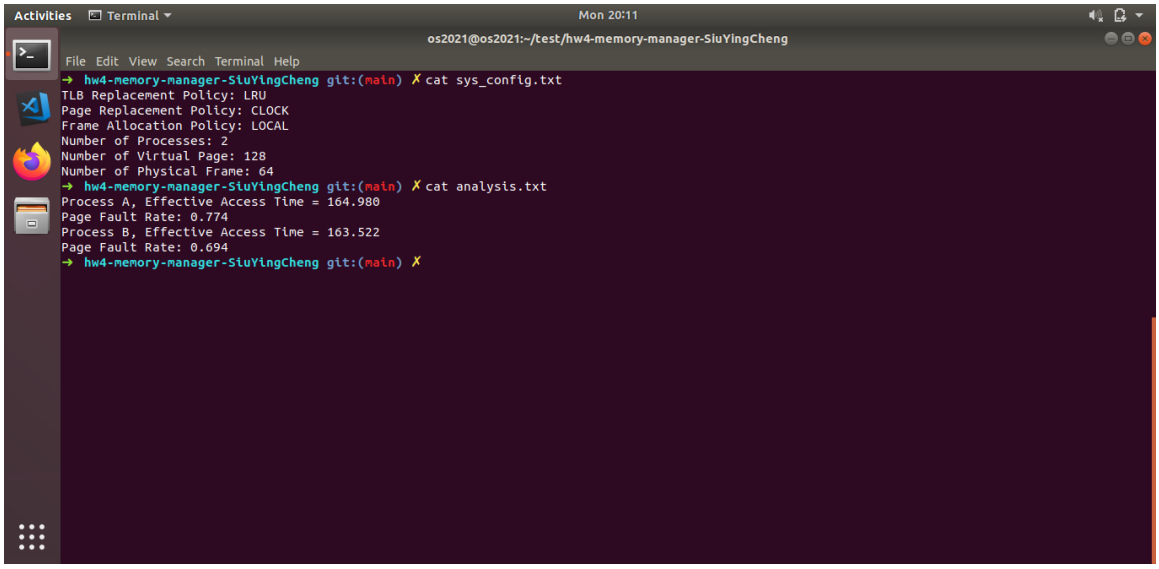
1. 執行結果

1) TLB 為 LRU algorithm

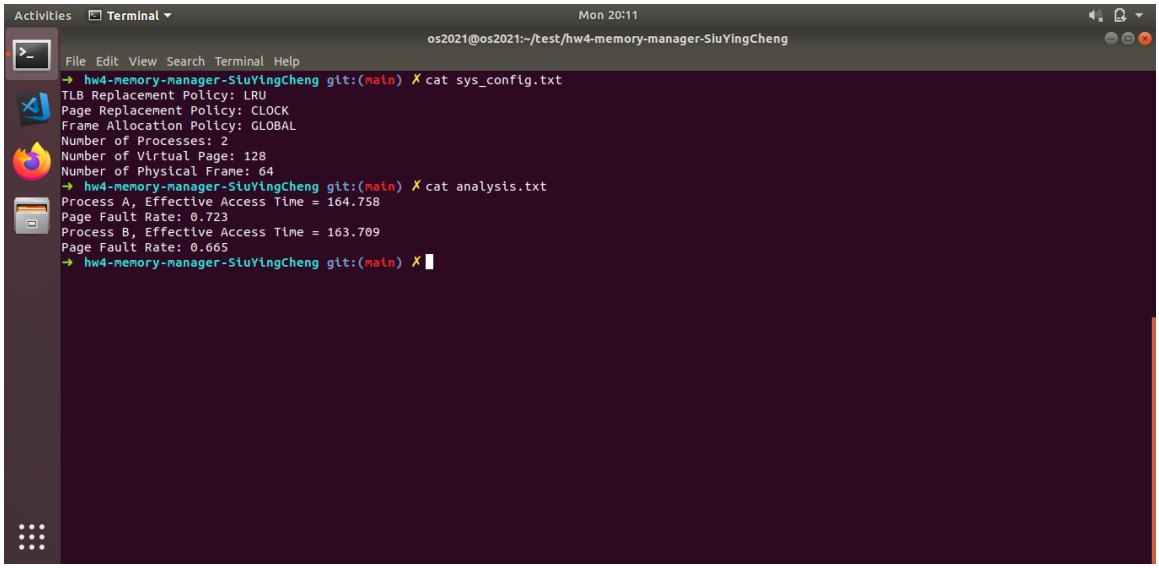
- FIFO LOCAL



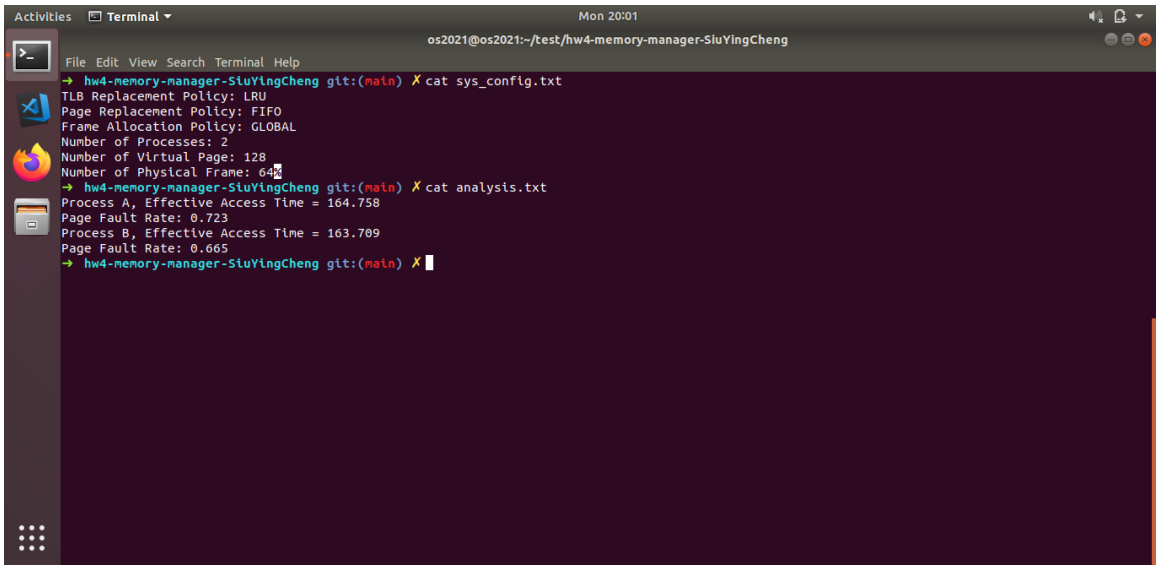
- CLOCK LOCAL



- CLOCK GLOBAL

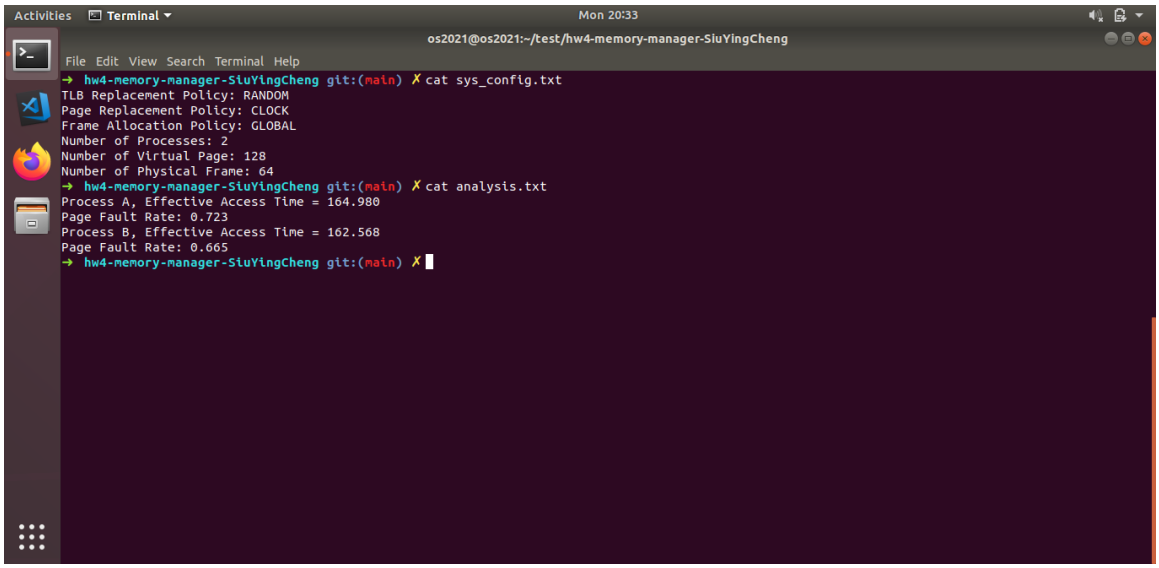


- FIFO GLOBAL

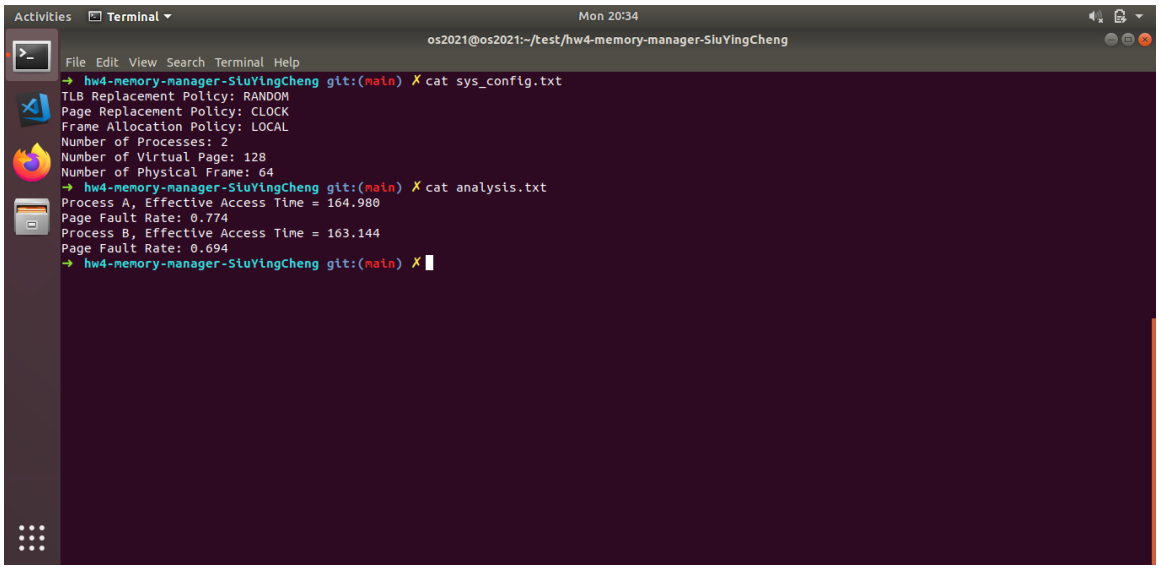


2) TLB 為 RANDOM algorithm

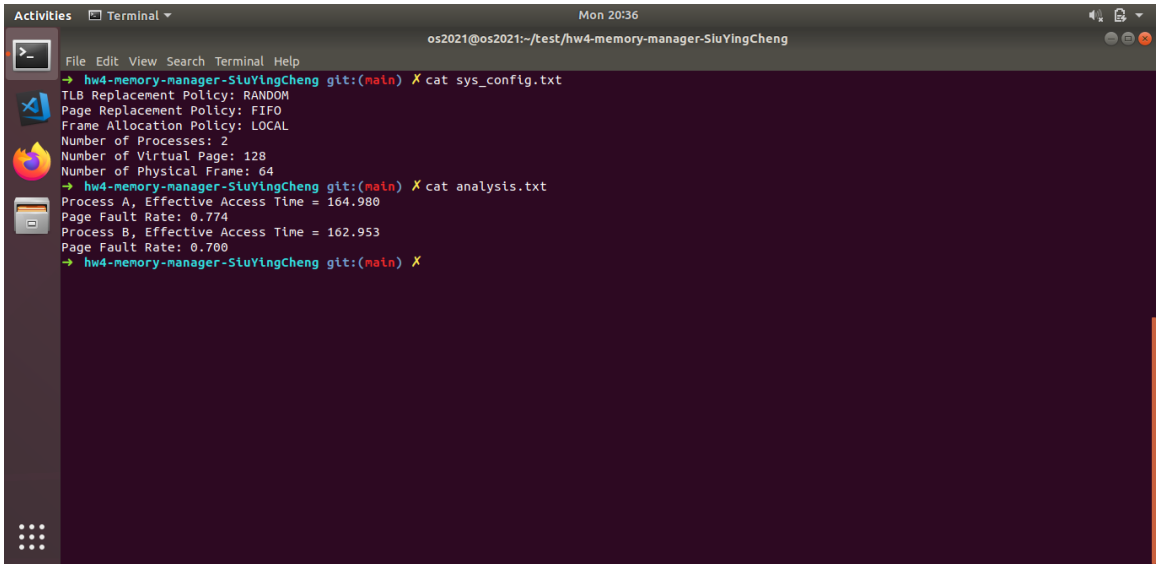
- CLOCK GLOBAL



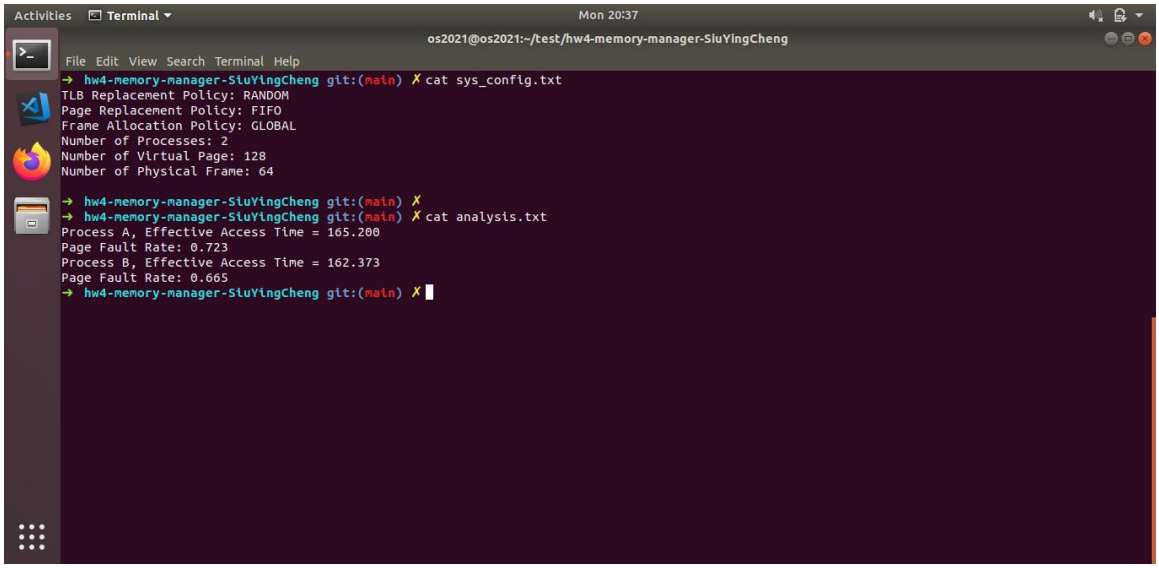
- CLOCK LOCAL



- FIFO LOCAL



- FIFO GLOBAL



2. 技術架構概述

- TLB Replacement Policy:
  - Random: 在 TLB 中任選一值為被取代的 page number，並將新的 page 和其 physical frame number 放到該位置。
  - LRU: 選取 TLB 中最久未被使用的值當作被取代的 page number，並將新的 page 和其 physical frame number 放到該位置。
- Page Replacement Policy:
  - FIFO：先進入 memory 的 Page 會先被替換。

    - Clock： 為每個 page 附上一個 reference bit，根據 reference bit 的狀態來替代 page。若 reference bit 為 0 的話才替換，若是 1 的話就換回 0 來達到 LRU 的效果。
- Page Allocation Policy:
    - Global: 被 page out 的 page 可在 system 中所有 process 的 page 做選取，缺點是自己 process 不能掌控自己的 page fault rate.
    - Local: 被 page out 的 page 只能從自己的 process 中的 page 做選取，未能剔出其他 process 比較少用的 page.

3. 實驗結果討論

- EAT 與 page fault rate 有著密切的聯動關係，因此在 process A 和 process B 執行過程中，Page Allocation Algorithm 為 GLOBAL 的 EAT 和 page fault rate 最少。因此在 EAT 和 page fault rate 的分析中，我們應該選用 Global allocation 來降低原來的 page fault 以及 process 的 effective access time。原因是因爲在 GLOBAL allocation 的情況下，所需要的 victim page searching 時間較少，且其他 process 有機會替換許久未被使用的 page，較不會去替換近期剛進 memory 的 page。
- 結論
    - 透過觀察得出，無論在 EAT 或 Page Fault Rate 的分析上，Global Allocation 的 Algorithm 都會取得較佳的結果。此點也在現實中的系統得到驗證，許多現實中的系統都使用 Global Allocation 的設計。
    - 另外，若 TLB 的 Replacement Policy 為 Random 時，page fault rate 和 EAT 會在每一次程式執行時稍有波動，原因是因爲我們無法掌握被替換的 TLB index，有時候被替換的是已經許久未被使用的 index，有時候也可能是剛被 assign 的 index。