

姓名：鄭琇櫻

系級：資工 111

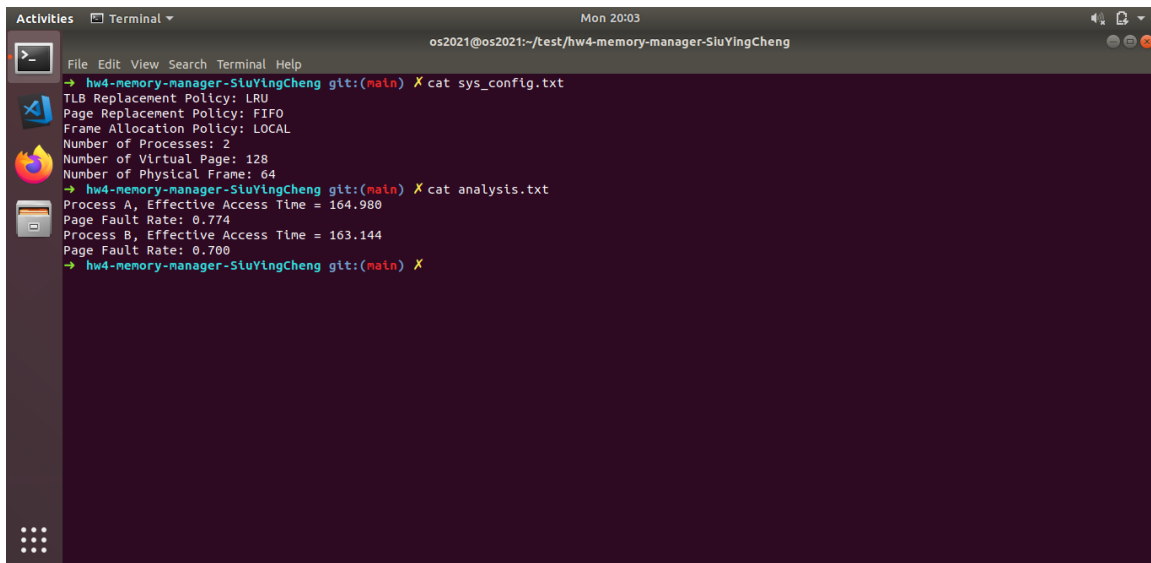
學號：I84077010

HW4：Algorithm 分析文件

1. 執行結果

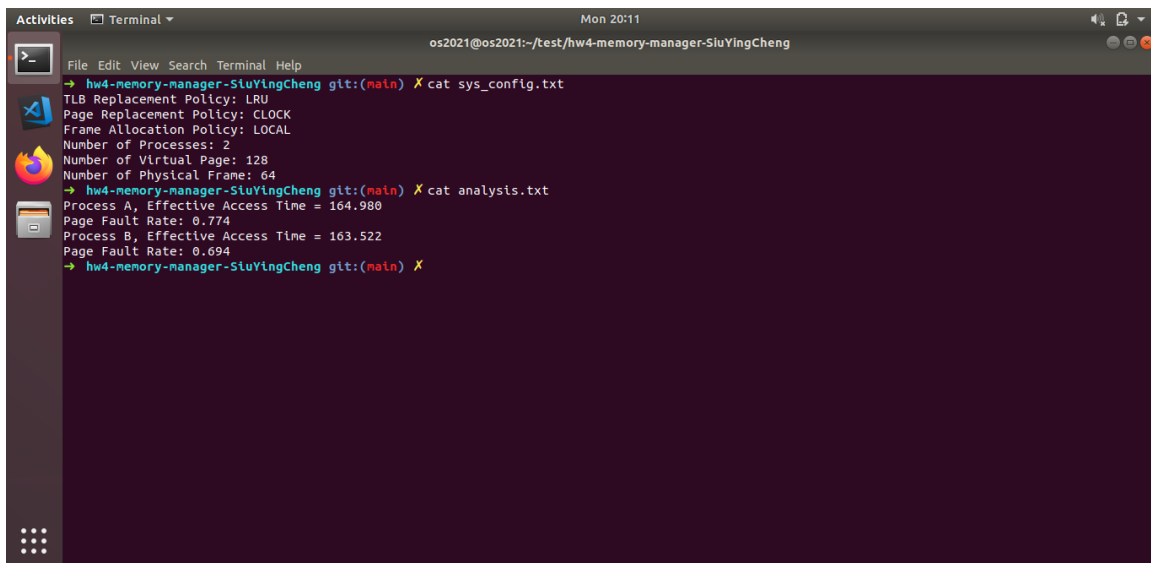
1) TLB 為 LRU algorithm

- FIFO LOCAL



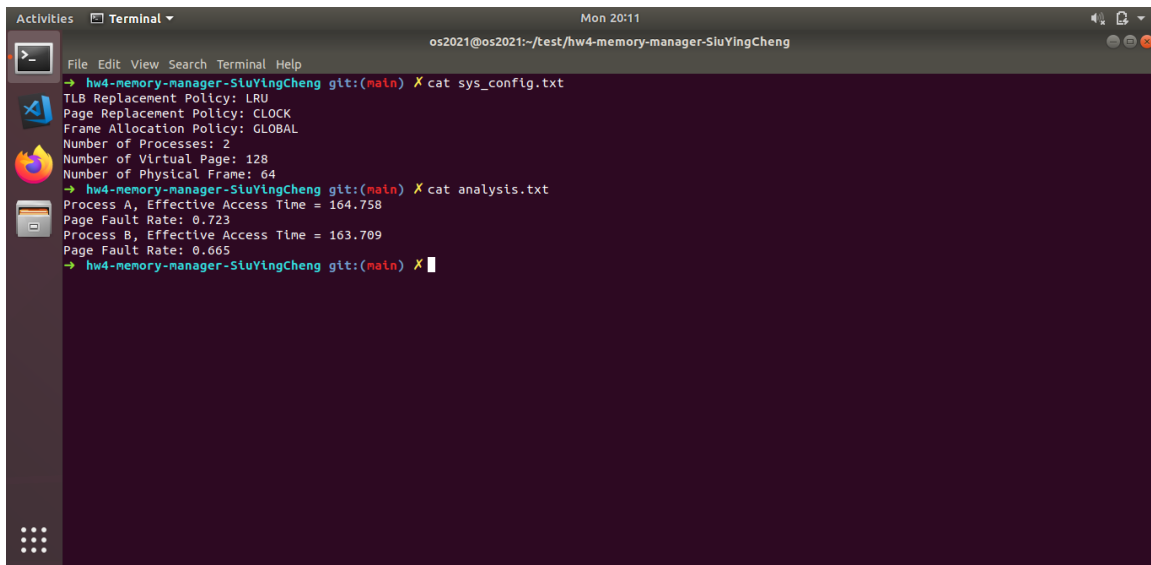
```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: LRU
Page Replacement Policy: FIFO
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.980
Page Fault Rate: 0.774
Process B, Effective Access Time = 163.144
Page Fault Rate: 0.700
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

- CLOCK LOCAL



```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: LRU
Page Replacement Policy: CLOCK
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.980
Page Fault Rate: 0.774
Process B, Effective Access Time = 163.522
Page Fault Rate: 0.694
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

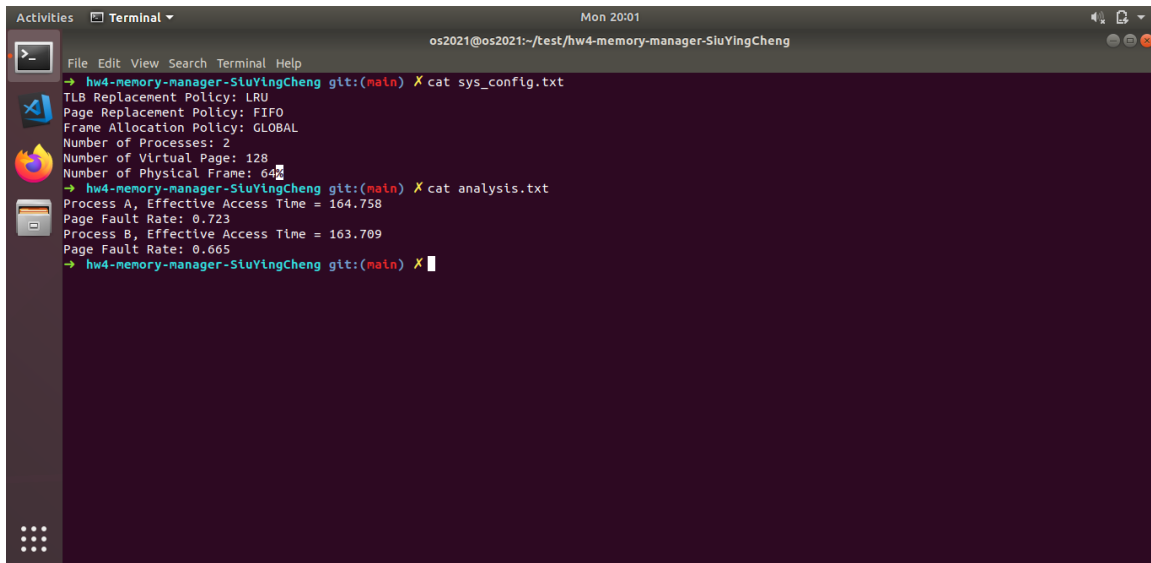
- CLOCK GLOBAL



A terminal window titled "os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng" showing the output of a program. The program displays configuration and performance metrics for the CLOCK GLOBAL policy. The configuration includes LRU for TLB replacement, CLOCK for page replacement, and GLOBAL for frame allocation, with 2 processes, 128 virtual pages, and 64 physical frames. Performance metrics show Process A with an effective access time of 164.758 and a page fault rate of 0.723, and Process B with an effective access time of 163.709 and a page fault rate of 0.665.

```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: LRU
Page Replacement Policy: CLOCK
Frame Allocation Policy: GLOBAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.758
Page Fault Rate: 0.723
Process B, Effective Access Time = 163.709
Page Fault Rate: 0.665
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

- FIFO GLOBAL

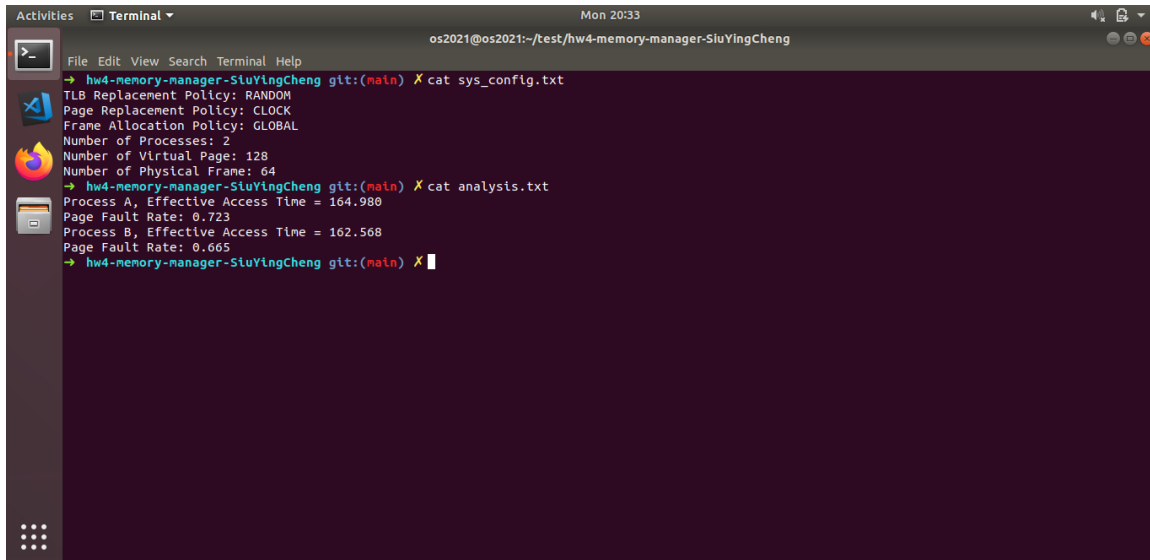


A terminal window titled "os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng" showing the output of a program. The program displays configuration and performance metrics for the FIFO GLOBAL policy. The configuration includes LRU for TLB replacement, FIFO for page replacement, and GLOBAL for frame allocation, with 2 processes, 128 virtual pages, and 64 physical frames. Performance metrics show Process A with an effective access time of 164.758 and a page fault rate of 0.723, and Process B with an effective access time of 163.709 and a page fault rate of 0.665.

```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: LRU
Page Replacement Policy: FIFO
Frame Allocation Policy: GLOBAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.758
Page Fault Rate: 0.723
Process B, Effective Access Time = 163.709
Page Fault Rate: 0.665
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

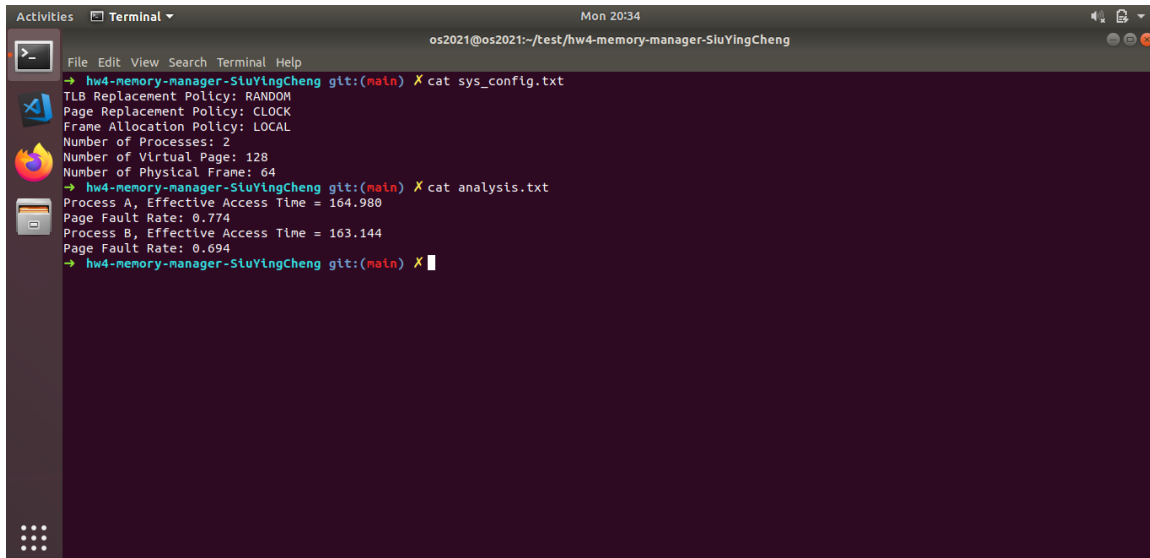
2) TLB 為 RANDOM algorithm

- CLOCK GLOBAL

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng, Mon 20:33). The terminal output shows the execution of a program named "hw4-memory-manager-SiuYingCheng". The program displays configuration parameters: TLB Replacement Policy: RANDOM, Page Replacement Policy: CLOCK, and Frame Allocation Policy: GLOBAL. It also shows system parameters: Number of Processes: 2, Number of Virtual Page: 128, and Number of Physical Frame: 64. Two test cases are shown: Process A with Effective Access Time = 164.980 and Page Fault Rate: 0.723, and Process B with Effective Access Time = 162.568 and Page Fault Rate: 0.665. The prompt "hw4-memory-manager-SiuYingCheng git:(main) X" is visible at the end of the output.

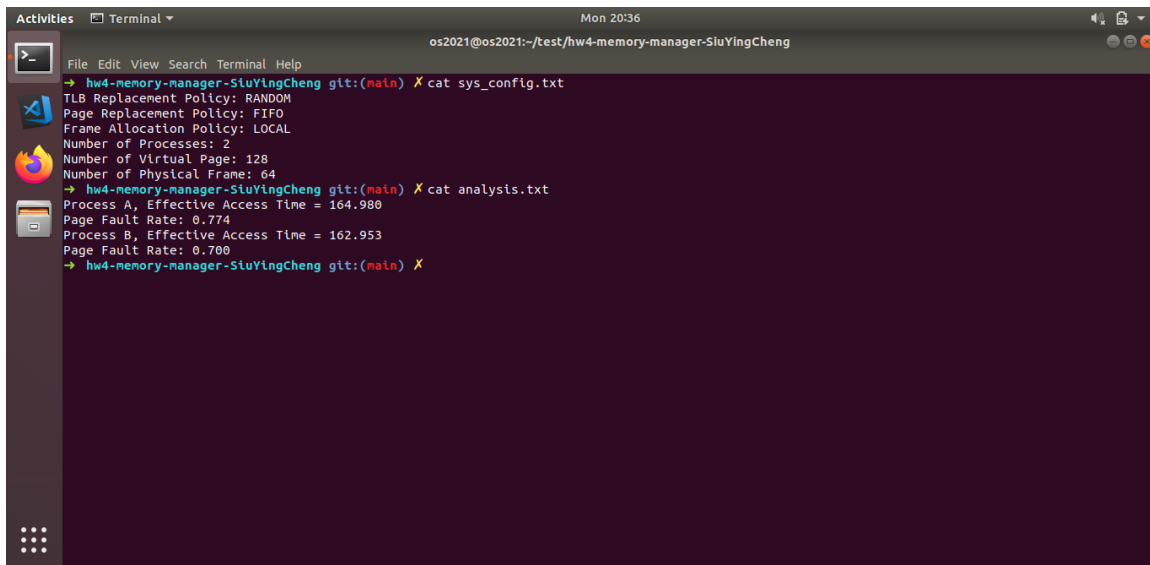
```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: RANDOM
Page Replacement Policy: CLOCK
Frame Allocation Policy: GLOBAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.980
Page Fault Rate: 0.723
Process B, Effective Access Time = 162.568
Page Fault Rate: 0.665
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

- CLOCK LOCAL

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng, Mon 20:34). The terminal output shows the execution of a program named "hw4-memory-manager-SiuYingCheng". The program displays configuration parameters: TLB Replacement Policy: RANDOM, Page Replacement Policy: CLOCK, and Frame Allocation Policy: LOCAL. It also shows system parameters: Number of Processes: 2, Number of Virtual Page: 128, and Number of Physical Frame: 64. Two test cases are shown: Process A with Effective Access Time = 164.980 and Page Fault Rate: 0.774, and Process B with Effective Access Time = 163.144 and Page Fault Rate: 0.694. The prompt "hw4-memory-manager-SiuYingCheng git:(main) X" is visible at the end of the output.

```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: RANDOM
Page Replacement Policy: CLOCK
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.980
Page Fault Rate: 0.774
Process B, Effective Access Time = 163.144
Page Fault Rate: 0.694
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

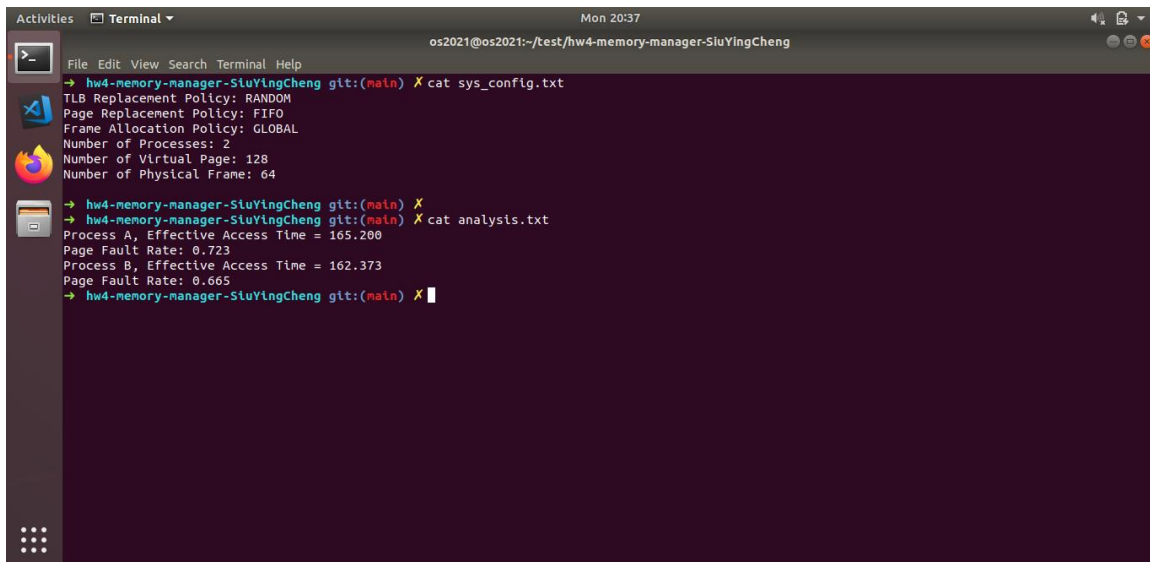
- FIFO LOCAL



A terminal window titled 'Terminal' showing the execution of a program named 'hw4-memory-manager-SiuYingCheng'. The program outputs configuration details for a FIFO LOCAL memory manager. The user runs 'cat sys_config.txt' and 'cat analysis.txt' to view the settings and performance metrics.

```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: RANDOM
Page Replacement Policy: FIFO
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 164.980
Page Fault Rate: 0.774
Process B, Effective Access Time = 162.953
Page Fault Rate: 0.700
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

- FIFO GLOBAL



A terminal window titled 'Terminal' showing the execution of the same program but with a GLOBAL frame allocation policy. The configuration and analysis outputs are displayed, showing a slight change in performance metrics compared to the LOCAL policy.

```
os2021@os2021:~/test/hw4-memory-manager-SiuYingCheng
→ hw4-memory-manager-SiuYingCheng git:(main) X cat sys_config.txt
TLB Replacement Policy: RANDOM
Page Replacement Policy: FIFO
Frame Allocation Policy: GLOBAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
→ hw4-memory-manager-SiuYingCheng git:(main) X
→ hw4-memory-manager-SiuYingCheng git:(main) X cat analysis.txt
Process A, Effective Access Time = 165.200
Page Fault Rate: 0.723
Process B, Effective Access Time = 162.373
Page Fault Rate: 0.665
→ hw4-memory-manager-SiuYingCheng git:(main) X
```

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。原因是因為在 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，原因是因為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。