OS 第四次作業

一、分工

鄧智宇：撰寫報告、程式邏輯、相關資料查找

黃信維：撰寫程式、程式邏輯、相關資料查找

二、作業步驟以及遇到問題

1. 在「replace_policy.c」中進行 paging 需調度系統資源，但引入<linux/xx>系列標頭時 make 會出錯。詢問助教後，知道這次作業不需要引入其他的標頭檔。
2. 如何比較輸入的 id 和已存在 page 裡面的 id 是否一致？需檢查是否與舊資料的 request id 相符, 就要另外再檢查原始的 request 到底給的是多少, 所以也需要在 ioctl 一開始就要去紀錄原始 request 與轉換後的 request 對應。
3. 在 1/8 後複製修改助教提供的提示後，更改如下(包含加分題)：

```
    switch(cmd_choice){
        case CMD_WRITE_REQUEST:

            //紀錄新近來 request 的 metadata
            //紀錄 virtual addr, physical addr, 是否已經在 disk內, 是否屬於 update
            //Modify here
            reqMeta[currReqNo].virtual_addr = virtual_addr;
            reqMeta[currReqNo].physical_addr = virtual_addr%256;
//          reqMeta[currReqNo].update = 0;
//|         reqMeta[currReqNo].alreadyInDisk = 0;

            paging_find_free_page(reqMeta[currReqNo]);
            break;

        case CMD_PRINT_INFO:
            print_information();
            break;
    }
```

```
    //步驟三之二：
    //當前沒有可用空間, 所以要先找 victim frame 並執行置換
    else{
        //找 victim frame 之 free list node 的位置
        //並找出 victim frame 代表的 virtual address
        victim_physical = paging_victim_selection();
        int i=0;
        list_for_each_entry(tmp_node, &free_list_header->next, next){
            //Modify here
            if(tmp_node->virtual_addr%256 == victim_physical){
                break;
            }
        }
        victim_virtual = tmp_node->virtual_addr;

        //倒回去檢查 victim 在 reqMeta 的紀錄 (因為越後面越新的狀態), 找出 victim 最新的資訊
        for(i=currReqNo; i>=0; i--){
            //若 history 找到最新 victim frame 所存的 victim_virtual 資訊
            if(reqMeta[i].virtual_addr == victim_virtual){
                //更新 victim 在 free_list 的紀錄, 把 victim frame 內容覆蓋成 user_req
                //Modify here
                tmp_node->virtual_address = reqMeta[i].virtual_address;
```

```c
//回傳 victim physical frame
static int paging_victim_selection(void)
{
    int victim_no;
    if(replace_policy == FIFO){
        //Modify here
        victim_no = reqMeta[currentVictimNo].physical_addr;
        if(currentVictimNo == 255){
            currentVictimNo = 0;
        }
        else{
            currentVictimNo += 1;
        }
    }
    else if(replace_policy == LRU){
        //Modify here
        struct free_list *tmp_node;
        int ref_count_min = 9999;
        victim_no = 0;

        list_for_each_entry(tmp_node, &free_list_header->next, next){
            if(tmp_node->ref_count < ref_count_min){
                ref_count_min = tmp_node->ref_count;
                victim_no = tmp_node->virtual_addr%256;
            }
        }
    }
    return victim_no;
}
```

4. 不過 makefile 卻無法成功，錯誤訊息如圖

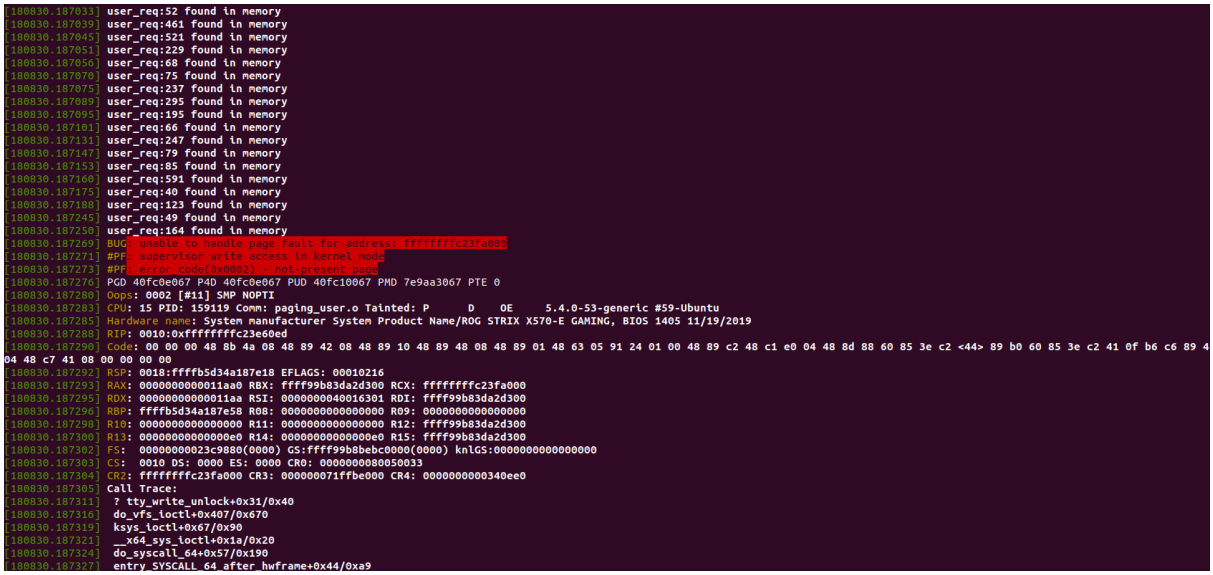
```
veridas@veridas-VirtualBox:~/linux-5.7.9/MyChrDevice$ sudo make
[sudo] password for veridas:
\make -C /lib/modules/5.7.9406315017/build M=/home/veridas/linux-5.7.9/MyChrDevi
ce modules
make[1]: Entering directory '/home/veridas/linux-5.7.9'
scripts/Makefile.build:44: /home/veridas/linux-5.7.9/MyChrDevice/Makefile: No su
ch file or directory
make[2]: *** No rule to make target '/home/veridas/linux-5.7.9/MyChrDevice/Makef
ile'.  Stop.
Makefile:1732: recipe for target '/home/veridas/linux-5.7.9/MyChrDevice' failed
make[1]: *** [/home/veridas/linux-5.7.9/MyChrDevice] Error 2
make[1]: Leaving directory '/home/veridas/linux-5.7.9'
makefile:12: recipe for target 'modules' failed
make: *** [modules] Error 2
```

在助教的建議下，刪去 EXTRA_FLAG，並重新編譯 kernel，但問題變成無法讀取檔案：

```
veridas@veridas-VirtualBox:~/Desktop/oshw_2020/hw4/hw$ ./paging_user ~/home/veridas/Desktop/oshw_2020/hw4/hw/skew_dataset
Open dataset failed.
: No such file or directory
```

後來自己探索後，發現更改檔名後就可以執行。不過仍會出現以下錯誤：

```
$ sudo ./paging_user.o skew_dataset.txt
Now send request:2610
[1]    174937 killed     sudo ./paging_user.o skew_dataset.txt
```



目前猜測是記憶體操作錯誤，問題應是出在 vicim_select。

三、引用資料

list_for_each_entry

https://www.itread01.com/p/158186.html

ioctl

https://blog.xuite.net/yang44/dd/49077684