CSC3150 Project 3 Report

Tong Zhen 120090694

1. Program Environment

Linux kernel version

```
[120090694@node21 HM3]$ uname -r
3.10.0-862.el7.x86_64
```

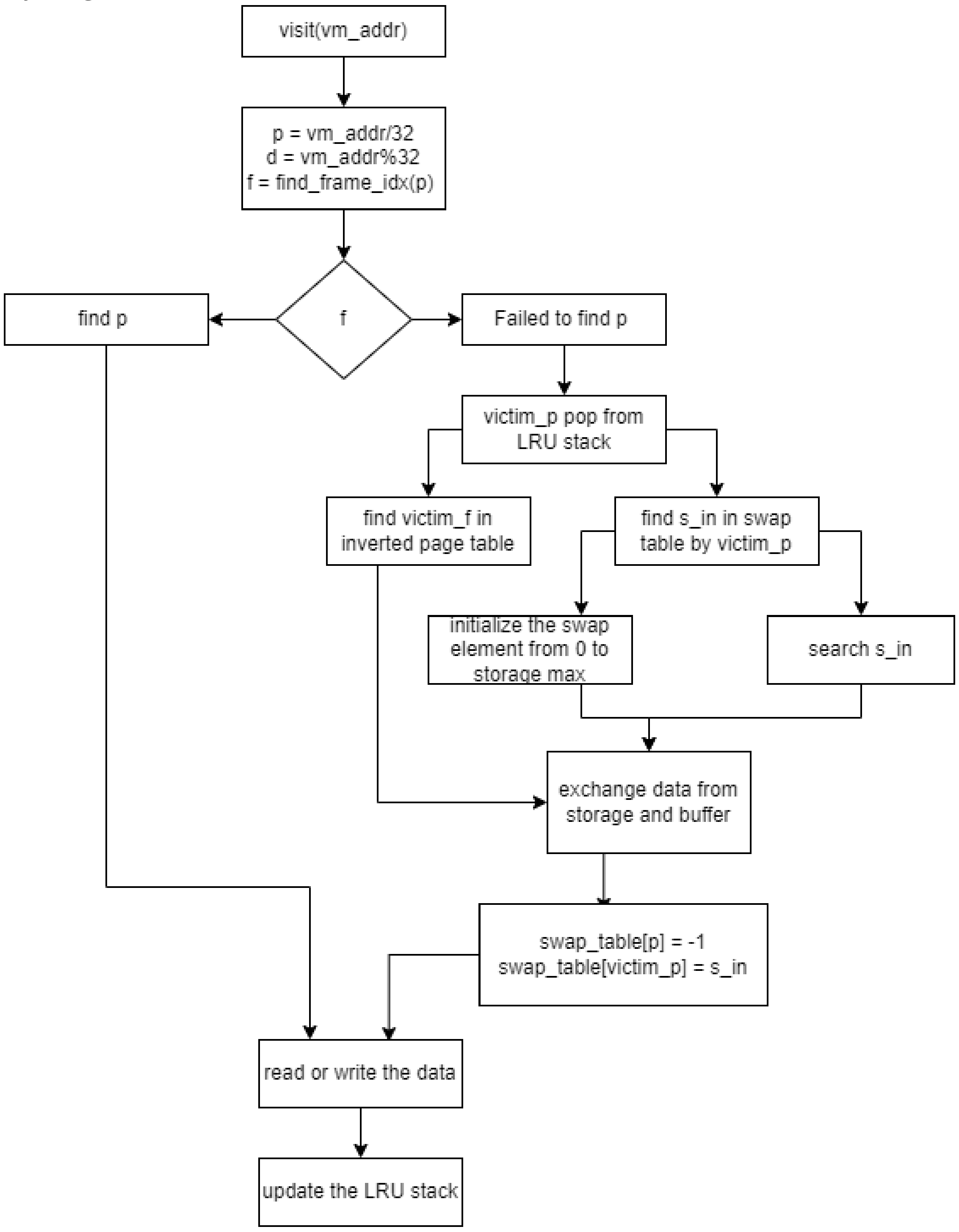
cuda version

```
[120090694@node21 HM3]$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Jun__8_16:49:14_PDT_2022
Cuda compilation tools, release 11.7, V11.7.99
Build cuda_11.7.r11.7/compiler.31442593_0
```

2. Program Design

Operation System Concept	**Size & Type Description**
User Logical Memory(160KB)	`STORAGE_SIZE + PHYSICAL_MEM_SIZE` (int[163,840])
Inverted Page Table(16KB)	`vm->invert_page_table` (int [2048])
Main Memory / Physical Memory (32KB)	`vm->buffer` (int [32768])
Disk / Secondary Memory (128KB)	`vm->storage` (int [131072])
LRU stack	`vm->LRU_bottom` to `vm->LRU_top`(1024 node)
Swap table(5KB)	`vm->swap_table`(int[5120])

The program used cuda memory storage to simulate the concepts above
My design flow chart is as follows:


visit(vm_addr)
p = vm_addr/32
d = vm_addr%32
f = find_frame_idx(p)
f
find p
Failed to find p
victim_p pop from LRU stack
find victim_f in inverted page table
find s_in in swap table by victim_p
initialize the swap element from 0 to storage max
search s_in
exchange data from storage and buffer
swap_table[p] = -1
swap_table[victim_p] = s_in
read or write the data
update the LRU stack

1. First, the `vm_read()` and `vm_read()` both visit a virtual address, and the page should be found. The logical page number is the vm_addr divided by page size 32. The offset is the vm_addr mod the page size 32.
2. Second, find the physical page number by the logical page number p in inverted page table
   a. If we find the physical page number successfully in the inverted page tabel, which means the data we want to visit is in the physical memory (buffer), go to step 4

   b. If we didn't find the physical page in the inverted page table, which means the data we want to visit is in the storage, do swap

3. Swap: We need to get the vitim frame in the logical memory(buffer), say `vicitim_p` that is popped from the LRU stack. Search the physical memory, say `vicitim_f` in the inverted page table. Search the storage location in the swap table, say `s_in`. If the storage location we get is -1, which means it un-used, we assign it to a number less than storage maximum to activate. Remove the buffer data to the storage and remove the storage data to the buffer. Assign the victim item in swap table to -1 to inactivate, and assign the visit page item to `s_in` in swap table
4. Do write new data over the old data or read out the data in the target position, then update the LRU stack

3. Function Information

`init_invert_page_table`
The function uses 1024 integer to storage the page number cooresponding to the physical frame number. Initialize them by -1
`init_swap_table`
The function uses 1024+2096 integer to storage the storage location information for the virtual page, Initialize them by -1
`showLRU`
This is a function used to type the LRU stack information, used for debug.
`update_LRU`
When a new item comes in a LRU stack, it will find the previous position of the item and remove it to the top of the LRU stack
When a victim is needed because of swapping, pop the bottom of the LRU stack
`get_frameIdx`
This function is used to find the physical frame page in the inverted page table and return the corresponding index.
`vm_read`
This function will read a data by the virtual address given. The operation is described above.
`vm_write`
Same as vm_read, do the visit operation and write in the data.
`vm_snapshot`
This function is used to read all the data in the physical memory and the storage, output it as a array. The offset argument is used to do the circular move in the output array.

4. Run

use make file

```
# build
[120090694@node21 HM3]$ make
```

use nvcc directly build

```
# buid
[120090694@node21 HM3]$ nvcc --relocatable-device-code=true main.cu
user_program.cu virtual_memory.cu -o test
```

run

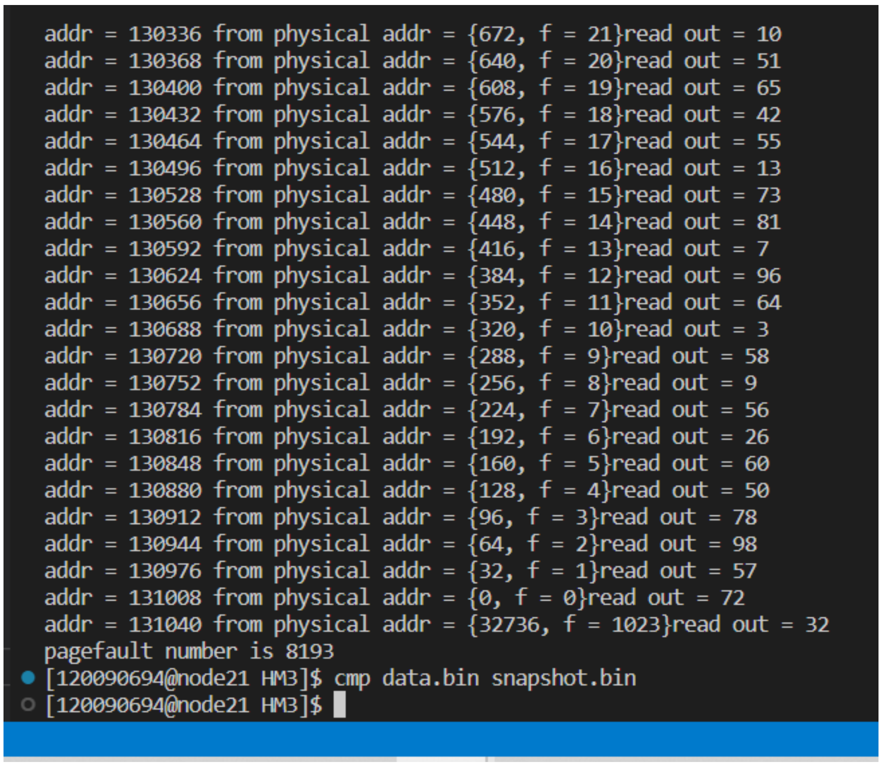
```
# run
/[120090694@node21 HM3]$ ./test
```

compare

```
[120090694@node21 HM3_bonus]$ cmp data.bin snapshot.bin
```

5. Output information and explanation

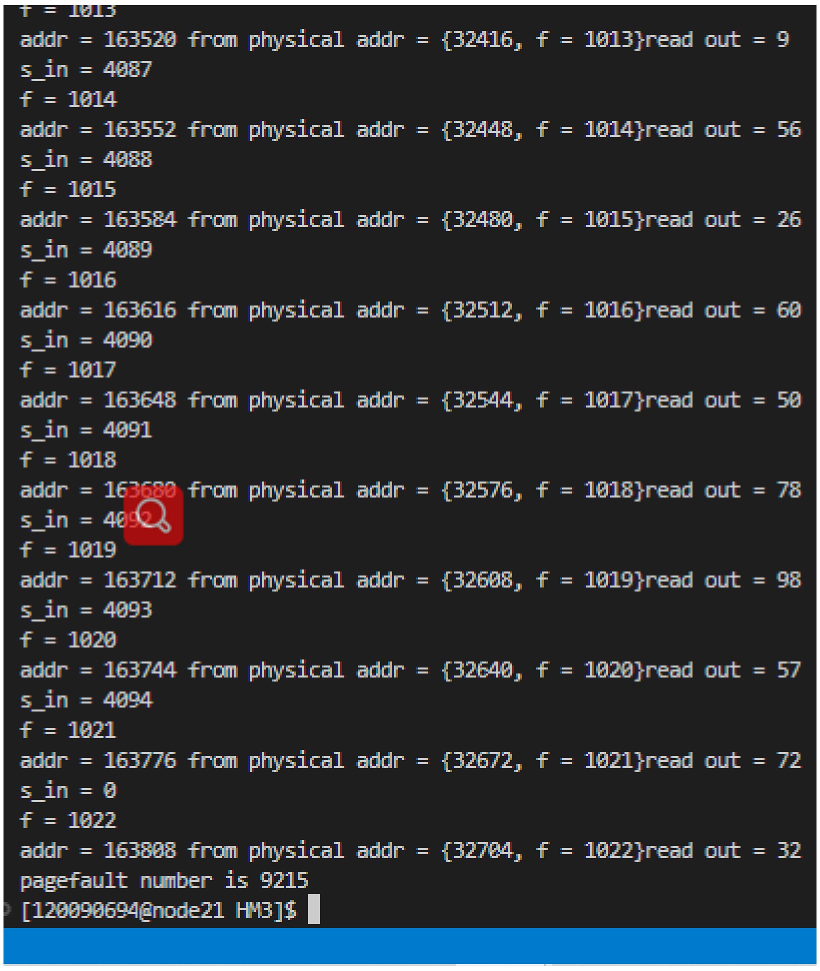
Task 1



```
addr = 130336 from physical addr = {672, f = 21}read out = 10
addr = 130368 from physical addr = {640, f = 20}read out = 51
addr = 130400 from physical addr = {608, f = 19}read out = 65
addr = 130432 from physical addr = {576, f = 18}read out = 42
addr = 130464 from physical addr = {544, f = 17}read out = 55
addr = 130496 from physical addr = {512, f = 16}read out = 13
addr = 130528 from physical addr = {480, f = 15}read out = 73
addr = 130560 from physical addr = {448, f = 14}read out = 81
addr = 130592 from physical addr = {416, f = 13}read out = 7
addr = 130624 from physical addr = {384, f = 12}read out = 96
addr = 130656 from physical addr = {352, f = 11}read out = 64
addr = 130688 from physical addr = {320, f = 10}read out = 3
addr = 130720 from physical addr = {288, f = 9}read out = 58
addr = 130752 from physical addr = {256, f = 8}read out = 9
addr = 130784 from physical addr = {224, f = 7}read out = 56
addr = 130816 from physical addr = {192, f = 6}read out = 26
addr = 130848 from physical addr = {160, f = 5}read out = 60
addr = 130880 from physical addr = {128, f = 4}read out = 50
addr = 130912 from physical addr = {96, f = 3}read out = 78
addr = 130944 from physical addr = {64, f = 2}read out = 98
addr = 130976 from physical addr = {32, f = 1}read out = 57
addr = 131008 from physical addr = {0, f = 0}read out = 72
addr = 131040 from physical addr = {32736, f = 1023}read out = 32
pagefault number is 8193
[120090694@node21 HM3]$ cmp data.bin snapshot.bin
[120090694@node21 HM3]$
```

Specifically, the total pagefault number for vm_write() is 4096, and 1 for vm_read(), and 4096 for vm_snapshot().
Explanation: Page fault occurs when the targeted page is not found in the page table. When the logical address is transmitted to the function, the page number is obtained by certain calculation. The program will search the page number in the page table. As long as it does not find the page number, a page fault will occur.

Task 2



```
f = 1013
addr = 163520 from physical addr = {32416, f = 1013}read out = 9
s_in = 4087
f = 1014
addr = 163552 from physical addr = {32448, f = 1014}read out = 56
s_in = 4088
f = 1015
addr = 163584 from physical addr = {32480, f = 1015}read out = 26
s_in = 4089
f = 1016
addr = 163616 from physical addr = {32512, f = 1016}read out = 60
s_in = 4090
f = 1017
addr = 163648 from physical addr = {32544, f = 1017}read out = 50
s_in = 4091
f = 1018
addr = 163680 from physical addr = {32576, f = 1018}read out = 78
s_in = 4092
f = 1019
addr = 163712 from physical addr = {32608, f = 1019}read out = 98
s_in = 4093
f = 1020
addr = 163744 from physical addr = {32640, f = 1020}read out = 57
s_in = 4094
f = 1021
addr = 163776 from physical addr = {32672, f = 1021}read out = 72
s_in = 0
f = 1022
addr = 163808 from physical addr = {32704, f = 1022}read out = 32
pagefault number is 9215
[120090694@node21 HM3]$
```

The total page fault for the first loop is 4096, and the secodn for loop is 1023, and the snapshot read take 4096 page fault.

6. Bonus

In bonus version 1,
 the general visit step is similar to the flow chart above. Here declear the difference.

1. First, in `main.cu` we need to launch 4 threads:

    ```
    mykernel<<<1, 4, INVERT_PAGE_TABLE_SIZE>>>(input_size);
    ```

2. Second, in the `user_program.cu` the task is done concurrently, which need to divide the give each thread its own logical address to visit.It is the same when we need to read all the data and do the snapshot

    ```
        for(int i = 0; i < input_size/4; i++){
            int addr = i + thread_id*(input_size/4);
    ```

3. Third, the `invert_page_table` adds 1024 integer for the `thread_id`. They are firstly initialized as -1.

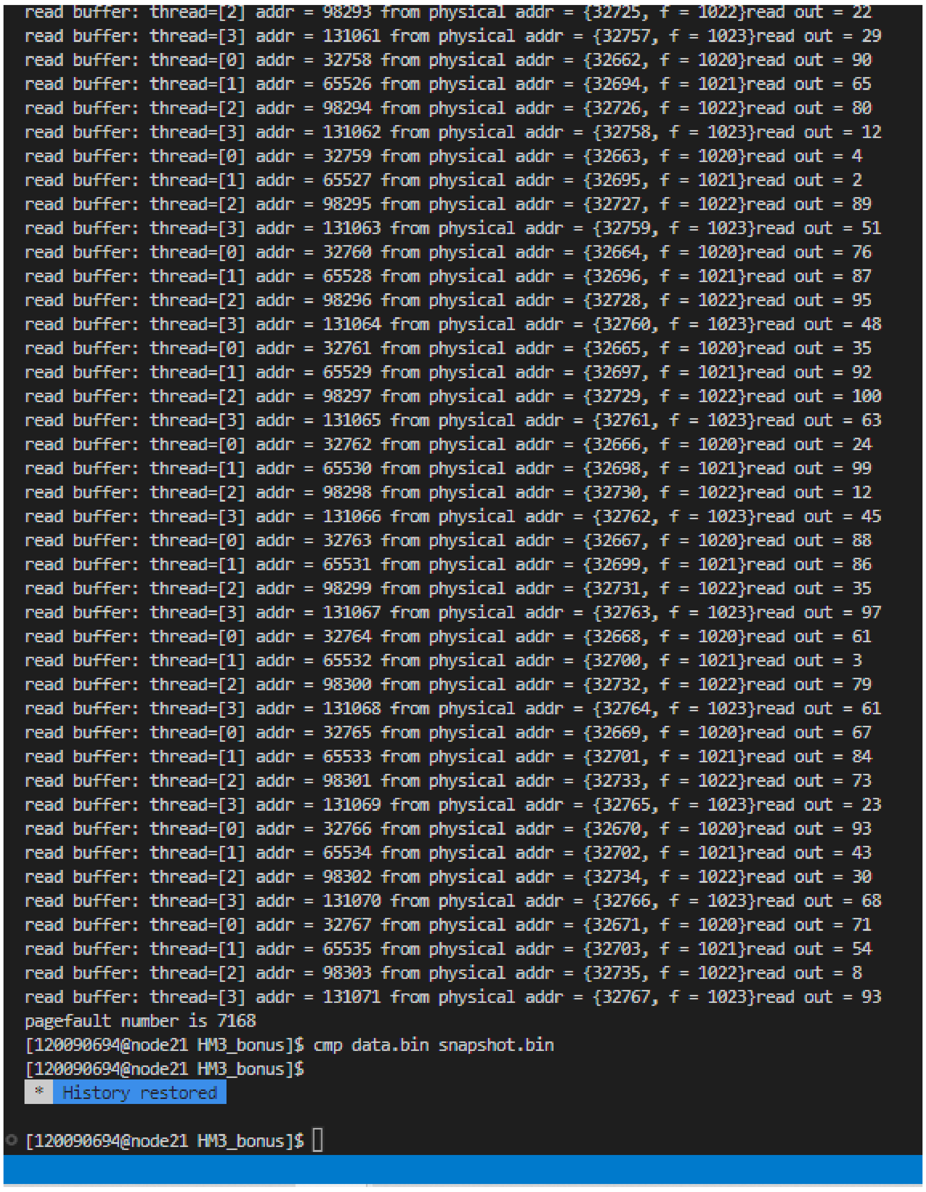
4. Fourth, when finding index of the frame of a given virtual page number in the inverted page table, we need to confirm the thread_id also.

5. Fifth, the `vm->storage_cnt` (a int) will add 1, whenever it meets the unused swap table elements before it meets the storage maximum. However, each thread will add 1 indepentantly, which the actual `vm->storage_cnt` is uncertain. Therefore, the `vm->storage_cnt` is divided to 4 parts for the the threads to add without mistake.

Run
The same way as the normal task, **CHANGE THE TEST IN USER_PROGRAM.CU**

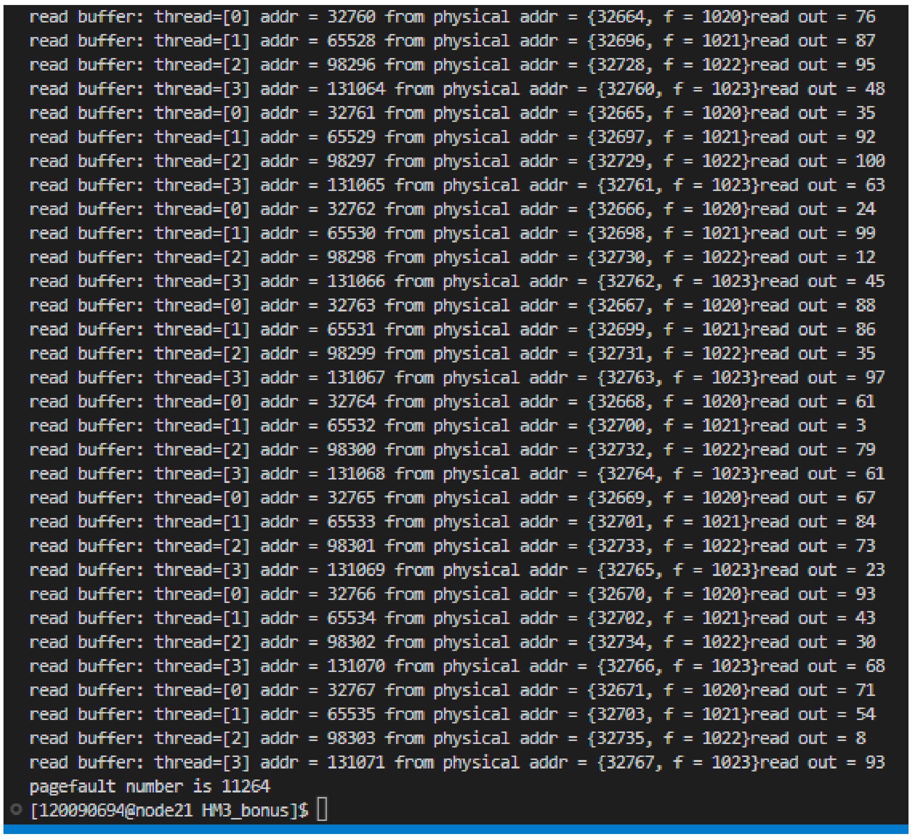
Output
Run the write all data, read all data, and do snapshot

```
    for(int i = 0; i < input_size/4; i++){

        int addr = i + thread_id*(input_size/4);
        // printf("thread_id = %d addr = %d\n",thread_id, addr);
        if(thread_id == 0)
            vm_write(vm, addr, input[addr], thread_id);
        __syncthreads();
        if(thread_id == 1)
            vm_write(vm, addr, input[addr], thread_id);
        __syncthreads();
        if(thread_id == 2)
            vm_write(vm, addr, input[addr], thread_id);
        __syncthreads();
        if(thread_id == 3)
            vm_write(vm, addr, input[addr], thread_id);
        __syncthreads();
    }

    for(int i = 0; i < input_size/4 ; i ++){

        int addr = i + thread_id*(input_size/4);
        if(thread_id == 0)
            vm_read(vm, addr, thread_id);
        __syncthreads();
        if(thread_id == 1)
            vm_read(vm, addr, thread_id);
        __syncthreads();
        if(thread_id == 2)
            vm_read(vm, addr, thread_id);
        __syncthreads();
        if(thread_id == 3)
            vm_read(vm, addr, thread_id);
        __syncthreads();
    }

    vm_snapshot(vm, results, 0, input_size, thread_id);
```



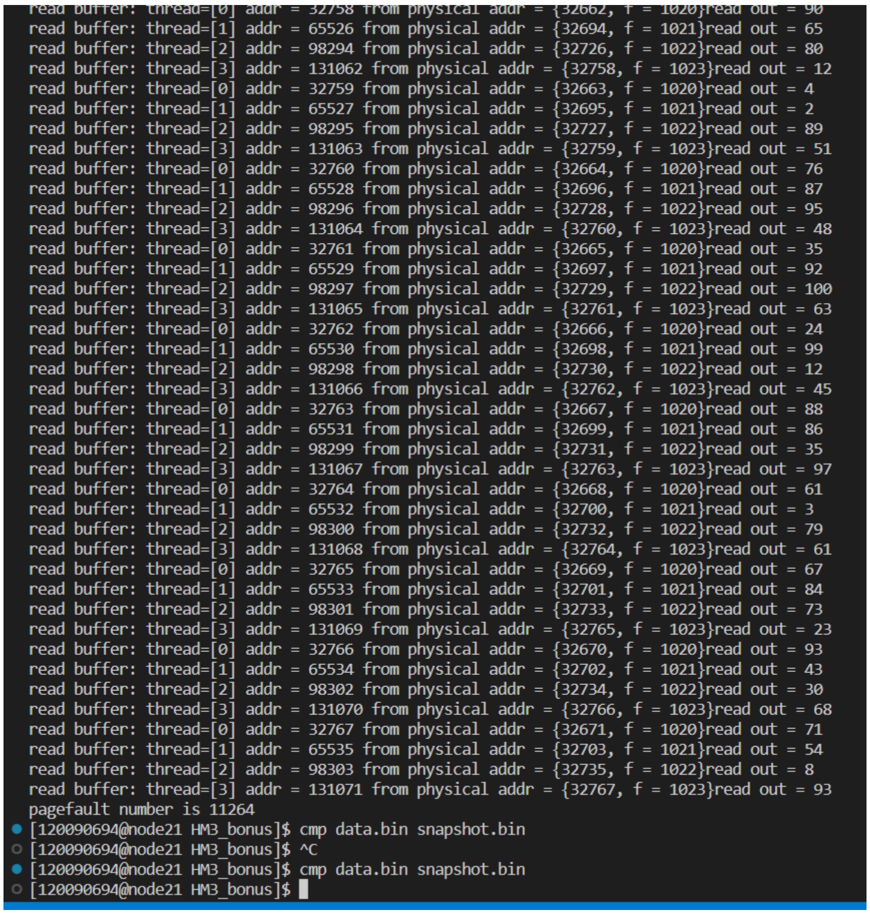
```
read buffer: thread=[2] addr = 98293 from physical addr = {32725, f = 1022}read out = 22
read buffer: thread=[3] addr = 131061 from physical addr = {32757, f = 1023}read out = 29
read buffer: thread=[0] addr = 32758 from physical addr = {32662, f = 1020}read out = 90
read buffer: thread=[1] addr = 65526 from physical addr = {32694, f = 1021}read out = 65
read buffer: thread=[2] addr = 98294 from physical addr = {32726, f = 1022}read out = 80
read buffer: thread=[3] addr = 131062 from physical addr = {32758, f = 1023}read out = 12
read buffer: thread=[0] addr = 32759 from physical addr = {32663, f = 1020}read out = 4
read buffer: thread=[1] addr = 65527 from physical addr = {32695, f = 1021}read out = 2
read buffer: thread=[2] addr = 98295 from physical addr = {32727, f = 1022}read out = 89
read buffer: thread=[3] addr = 131063 from physical addr = {32759, f = 1023}read out = 51
read buffer: thread=[0] addr = 32760 from physical addr = {32664, f = 1020}read out = 76
read buffer: thread=[1] addr = 65528 from physical addr = {32696, f = 1021}read out = 87
read buffer: thread=[2] addr = 98296 from physical addr = {32728, f = 1022}read out = 95
read buffer: thread=[3] addr = 131064 from physical addr = {32760, f = 1023}read out = 48
read buffer: thread=[0] addr = 32761 from physical addr = {32665, f = 1020}read out = 35
read buffer: thread=[1] addr = 65529 from physical addr = {32697, f = 1021}read out = 92
read buffer: thread=[2] addr = 98297 from physical addr = {32729, f = 1022}read out = 100
read buffer: thread=[3] addr = 131065 from physical addr = {32761, f = 1023}read out = 63
read buffer: thread=[0] addr = 32762 from physical addr = {32666, f = 1020}read out = 24
read buffer: thread=[1] addr = 65530 from physical addr = {32698, f = 1021}read out = 99
read buffer: thread=[2] addr = 98298 from physical addr = {32730, f = 1022}read out = 12
read buffer: thread=[3] addr = 131066 from physical addr = {32762, f = 1023}read out = 45
read buffer: thread=[0] addr = 32763 from physical addr = {32667, f = 1020}read out = 88
read buffer: thread=[1] addr = 65531 from physical addr = {32699, f = 1021}read out = 86
read buffer: thread=[2] addr = 98299 from physical addr = {32731, f = 1022}read out = 35
read buffer: thread=[3] addr = 131067 from physical addr = {32763, f = 1023}read out = 97
read buffer: thread=[0] addr = 32764 from physical addr = {32668, f = 1020}read out = 61
read buffer: thread=[1] addr = 65532 from physical addr = {32700, f = 1021}read out = 3
read buffer: thread=[2] addr = 98300 from physical addr = {32732, f = 1022}read out = 79
read buffer: thread=[3] addr = 131068 from physical addr = {32764, f = 1023}read out = 61
read buffer: thread=[0] addr = 32765 from physical addr = {32669, f = 1020}read out = 67
read buffer: thread=[1] addr = 65533 from physical addr = {32701, f = 1021}read out = 84
read buffer: thread=[2] addr = 98301 from physical addr = {32733, f = 1022}read out = 73
read buffer: thread=[3] addr = 131069 from physical addr = {32765, f = 1023}read out = 23
read buffer: thread=[0] addr = 32766 from physical addr = {32670, f = 1020}read out = 93
read buffer: thread=[1] addr = 65534 from physical addr = {32702, f = 1021}read out = 43
read buffer: thread=[2] addr = 98302 from physical addr = {32734, f = 1022}read out = 30
read buffer: thread=[3] addr = 131070 from physical addr = {32766, f = 1023}read out = 68
read buffer: thread=[0] addr = 32767 from physical addr = {32671, f = 1020}read out = 71
read buffer: thread=[1] addr = 65535 from physical addr = {32703, f = 1021}read out = 54
read buffer: thread=[2] addr = 98303 from physical addr = {32735, f = 1022}read out = 8
read buffer: thread=[3] addr = 131071 from physical addr = {32767, f = 1023}read out = 93
pagefault number is 7168
[120090694@node21 HW3_bonus]$ cmp data.bin snapshot.bin
[120090694@node21 HW3_bonus]$
 *  History restored

[120090694@node21 HW3_bonus]$
```

Run the task 1(Change user_program.cu to support the multi-threads)



```
read buffer: thread=[0] addr = 32760 from physical addr = {32664, f = 1020}read out = 76
read buffer: thread=[1] addr = 65528 from physical addr = {32696, f = 1021}read out = 87
read buffer: thread=[2] addr = 98296 from physical addr = {32728, f = 1022}read out = 95
read buffer: thread=[3] addr = 131064 from physical addr = {32760, f = 1023}read out = 48
read buffer: thread=[0] addr = 32761 from physical addr = {32665, f = 1020}read out = 35
read buffer: thread=[1] addr = 65529 from physical addr = {32697, f = 1021}read out = 92
read buffer: thread=[2] addr = 98297 from physical addr = {32729, f = 1022}read out = 100
read buffer: thread=[3] addr = 131065 from physical addr = {32761, f = 1023}read out = 63
read buffer: thread=[0] addr = 32762 from physical addr = {32666, f = 1020}read out = 24
read buffer: thread=[1] addr = 65530 from physical addr = {32698, f = 1021}read out = 99
read buffer: thread=[2] addr = 98298 from physical addr = {32730, f = 1022}read out = 12
read buffer: thread=[3] addr = 131066 from physical addr = {32762, f = 1023}read out = 45
read buffer: thread=[0] addr = 32763 from physical addr = {32667, f = 1020}read out = 88
read buffer: thread=[1] addr = 65531 from physical addr = {32699, f = 1021}read out = 86
read buffer: thread=[2] addr = 98299 from physical addr = {32731, f = 1022}read out = 35
read buffer: thread=[3] addr = 131067 from physical addr = {32763, f = 1023}read out = 97
read buffer: thread=[0] addr = 32764 from physical addr = {32668, f = 1020}read out = 61
read buffer: thread=[1] addr = 65532 from physical addr = {32700, f = 1021}read out = 3
read buffer: thread=[2] addr = 98300 from physical addr = {32732, f = 1022}read out = 79
read buffer: thread=[3] addr = 131068 from physical addr = {32764, f = 1023}read out = 61
read buffer: thread=[0] addr = 32765 from physical addr = {32669, f = 1020}read out = 67
read buffer: thread=[1] addr = 65533 from physical addr = {32701, f = 1021}read out = 84
read buffer: thread=[2] addr = 98301 from physical addr = {32733, f = 1022}read out = 73
read buffer: thread=[3] addr = 131069 from physical addr = {32765, f = 1023}read out = 23
read buffer: thread=[0] addr = 32766 from physical addr = {32670, f = 1020}read out = 93
read buffer: thread=[1] addr = 65534 from physical addr = {32702, f = 1021}read out = 43
read buffer: thread=[2] addr = 98302 from physical addr = {32734, f = 1022}read out = 30
read buffer: thread=[3] addr = 131070 from physical addr = {32766, f = 1023}read out = 68
read buffer: thread=[0] addr = 32767 from physical addr = {32671, f = 1020}read out = 71
read buffer: thread=[1] addr = 65535 from physical addr = {32703, f = 1021}read out = 54
read buffer: thread=[2] addr = 98303 from physical addr = {32735, f = 1022}read out = 8
read buffer: thread=[3] addr = 131071 from physical addr = {32767, f = 1023}read out = 93
pagefault number is 11264
[120090694@node21 HW3_bonus]$
```

Run task 2, the offset write process need to do the index division for the 4 thread.



```
read buffer: thread=[0] addr = 32758 from physical addr = {32662, f = 1020}read out = 90
read buffer: thread=[1] addr = 65526 from physical addr = {32694, f = 1021}read out = 65
read buffer: thread=[2] addr = 98294 from physical addr = {32726, f = 1022}read out = 80
read buffer: thread=[3] addr = 131062 from physical addr = {32758, f = 1023}read out = 12
read buffer: thread=[0] addr = 32759 from physical addr = {32663, f = 1020}read out = 4
read buffer: thread=[1] addr = 65527 from physical addr = {32695, f = 1021}read out = 2
read buffer: thread=[2] addr = 98295 from physical addr = {32727, f = 1022}read out = 89
read buffer: thread=[3] addr = 131063 from physical addr = {32759, f = 1023}read out = 51
read buffer: thread=[0] addr = 32760 from physical addr = {32664, f = 1020}read out = 76
read buffer: thread=[1] addr = 65528 from physical addr = {32696, f = 1021}read out = 87
read buffer: thread=[2] addr = 98296 from physical addr = {32728, f = 1022}read out = 95
read buffer: thread=[3] addr = 131064 from physical addr = {32760, f = 1023}read out = 48
read buffer: thread=[0] addr = 32761 from physical addr = {32665, f = 1020}read out = 35
read buffer: thread=[1] addr = 65529 from physical addr = {32697, f = 1021}read out = 92
read buffer: thread=[2] addr = 98297 from physical addr = {32729, f = 1022}read out = 100
read buffer: thread=[3] addr = 131065 from physical addr = {32761, f = 1023}read out = 63
read buffer: thread=[0] addr = 32762 from physical addr = {32666, f = 1020}read out = 24
read buffer: thread=[1] addr = 65530 from physical addr = {32698, f = 1021}read out = 99
read buffer: thread=[2] addr = 98298 from physical addr = {32730, f = 1022}read out = 12
read buffer: thread=[3] addr = 131066 from physical addr = {32762, f = 1023}read out = 45
read buffer: thread=[0] addr = 32763 from physical addr = {32667, f = 1020}read out = 88
read buffer: thread=[1] addr = 65531 from physical addr = {32699, f = 1021}read out = 86
read buffer: thread=[2] addr = 98299 from physical addr = {32731, f = 1022}read out = 35
read buffer: thread=[3] addr = 131067 from physical addr = {32763, f = 1023}read out = 97
read buffer: thread=[0] addr = 32764 from physical addr = {32668, f = 1020}read out = 61
read buffer: thread=[1] addr = 65532 from physical addr = {32700, f = 1021}read out = 3
read buffer: thread=[2] addr = 98300 from physical addr = {32732, f = 1022}read out = 79
read buffer: thread=[3] addr = 131068 from physical addr = {32764, f = 1023}read out = 61
read buffer: thread=[0] addr = 32765 from physical addr = {32669, f = 1020}read out = 67
read buffer: thread=[1] addr = 65533 from physical addr = {32701, f = 1021}read out = 84
read buffer: thread=[2] addr = 98301 from physical addr = {32733, f = 1022}read out = 73
read buffer: thread=[3] addr = 131069 from physical addr = {32765, f = 1023}read out = 23
read buffer: thread=[0] addr = 32766 from physical addr = {32670, f = 1020}read out = 93
read buffer: thread=[1] addr = 65534 from physical addr = {32702, f = 1021}read out = 43
read buffer: thread=[2] addr = 98302 from physical addr = {32734, f = 1022}read out = 30
read buffer: thread=[3] addr = 131070 from physical addr = {32766, f = 1023}read out = 68
read buffer: thread=[0] addr = 32767 from physical addr = {32671, f = 1020}read out = 71
read buffer: thread=[1] addr = 65535 from physical addr = {32703, f = 1021}read out = 54
read buffer: thread=[2] addr = 98303 from physical addr = {32735, f = 1022}read out = 8
read buffer: thread=[3] addr = 131071 from physical addr = {32767, f = 1023}read out = 93
pagefault number is 11264
[120090694@node21 HM3_bonus]$ cmp data.bin snapshot.bin
[120090694@node21 HM3_bonus]$ ^C
[120090694@node21 HM3_bonus]$ cmp data.bin snapshot.bin
[120090694@node21 HM3_bonus]$ 
```