1. Environment

   OS: windows 10

   VS version: VS 2017

   CUDA version: 9.2

   GPU: GTX 1050 Ti

   Notice: due to the computer is in Chinese system, the outputs in terminal might contains come Chinese, so the corresponding translation is attached at some outputs screenshots.

2. Execution steps of the program
   a) Use VS 2017 to open the project
   b) Use ctrl+f7 to compile the .cu files in the folder
   c) Press ctrl+f5 to run the program
   d) Wait for the result (notice that the program last for 6 min in my own computer)

3. Design of the program
   a) Inverted page table structure

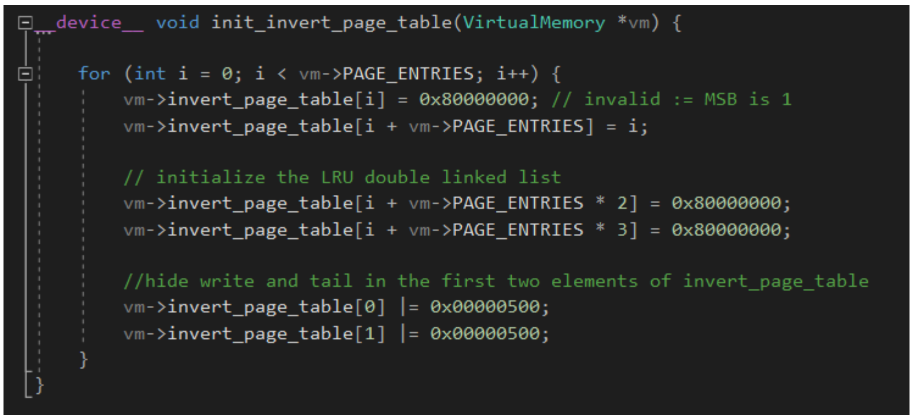
      Since the memory size given is limited in this project, the program decided to implement the LRU list using double linked list.

      i. For each element in the page table, they are seen as a node of double linked list, and their previous and next node information is store in the same row of that element in the table, that is, for the ith element, its previous and next nodes' information in "vm->invert_page_table[i + vm->PAGE_ENTRIES * 2]" and "vm->invert_page_table[i + vm->PAGE_ENTRIES * 3]" respectively.

      ii. Also, to avoid using extra memory, the program stores the head and tail information of the list in the first two element of the table, since the size of the inverted page table is 1024, which means they will only occupy the lower bits of the elements and will not affect the valid-invalid bit.

      iii. For valid-invalid bit, 0 in the MSB indicate the bit is valid, and 1 means invalid. For LRU node, 0x80000000 in the previous or next nodes' information means the LRU node is not inserted to the LRU list. For head and tail hiding in the first two elements in the list, 0x500 in head or tail means the list is empty.

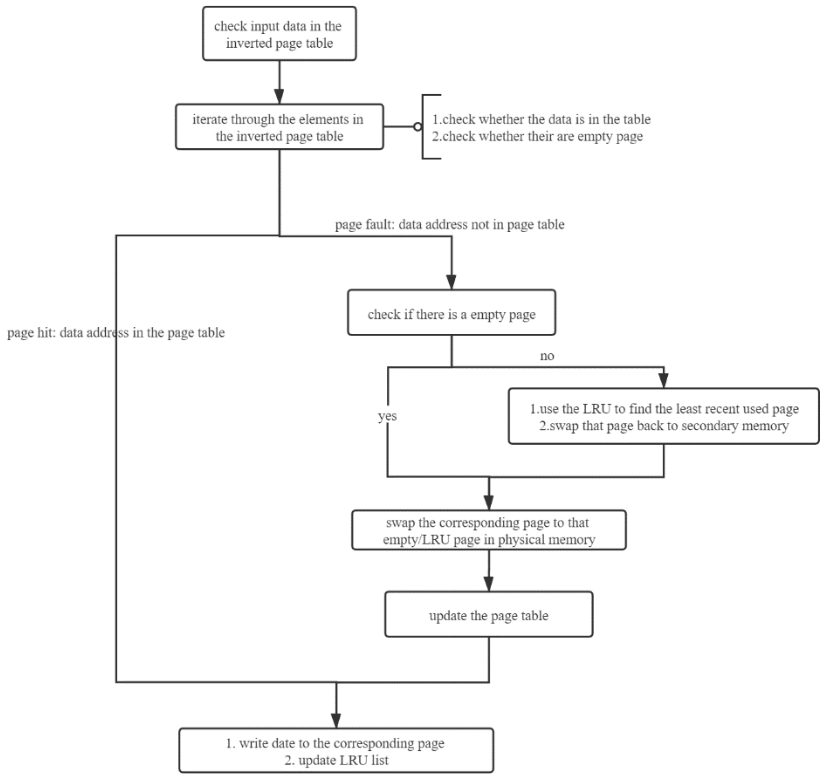
iv. the codes of initialization of inverted page table:



```
__device__ void init_invert_page_table(VirtualMemory *vm) {

    for (int i = 0; i < vm->PAGE_ENTRIES; i++) {
        vm->invert_page_table[i] = 0x80000000; // invalid := MSB is 1
        vm->invert_page_table[i + vm->PAGE_ENTRIES] = i;

        // initialize the LRU double linked list
        vm->invert_page_table[i + vm->PAGE_ENTRIES * 2] = 0x80000000;
        vm->invert_page_table[i + vm->PAGE_ENTRIES * 3] = 0x80000000;

        //hide write and tail in the first two elements of invert_page_table
        vm->invert_page_table[0] |= 0x00000500;
        vm->invert_page_table[1] |= 0x00000500;
    }
}
```

b) vm_write(VirtualMemory *vm, u32 addr, uchar value)

i. flow chart of the function:


check input data in the inverted page table
iterate through the elements in the inverted page table
1.check whether the data is in the table
2.check whether their are empty page
page fault: data address not in page table
page hit: data address in the page table
check if there is a empty page
no
yes
1.use the LRU to find the least recent used page
2.swap that page back to secondary memory
swap the corresponding page to that empty/LRU page in physical memory
update the page table
1. write date to the corresponding page
2. update LRU list

ii. basic logic of the function

1) The function will first go through the invert page table to check whether the data exist in the table and also check whether there exists empty page at the same time.

   a) Page hit: If the corresponding page is found, then the iteration will terminate and write data to that address directly.

   b) Page fault: in this case, the iteration will go through all the elements to find empty page.

      i. If there is an empty page: the program would directly swap the corresponding page in the secondary storage to the frame in physical memory.

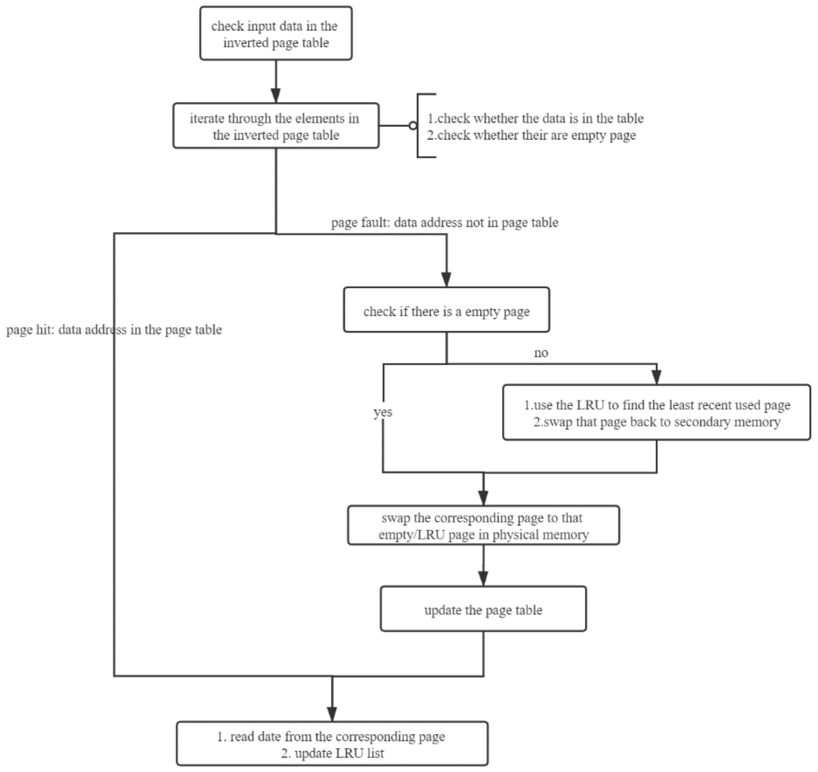
      ii. Otherwise: it will first use LRU to find the recently least used page and swap that page back to secondary memory. Notice that, to save time, the program directly use that LRU least used page to store the new page, which will directly skip the setting valid bit to invalid and iterating through the table to find an new empty page. After that, the program would swap the corresponding frame in the physical memory.

   Then the function can directly write the data into the new page. And update the address of that page in the page table.

2) After writing back the data, the function would set the LRU node of the corresponding element to the front of the LRU list. And also update the head and tail information in the page table.

c) vm_read(`VirtualMemory *vm, u32 addr`)

   i. flow chart of the function


check input data in the inverted page table
iterate through the elements in the inverted page table
1.check whether the data is in the table
2.check whether their are empty page
page fault: data address not in page table
page hit: data address in the page table
check if there is a empty page
no
yes
1.use the LRU to find the least recent used page
2.swap that page back to secondary memory
swap the corresponding page to that empty/LRU page in physical memory
update the page table
1. read date from the corresponding page
2. update LRU list

ii. basic logic of the function

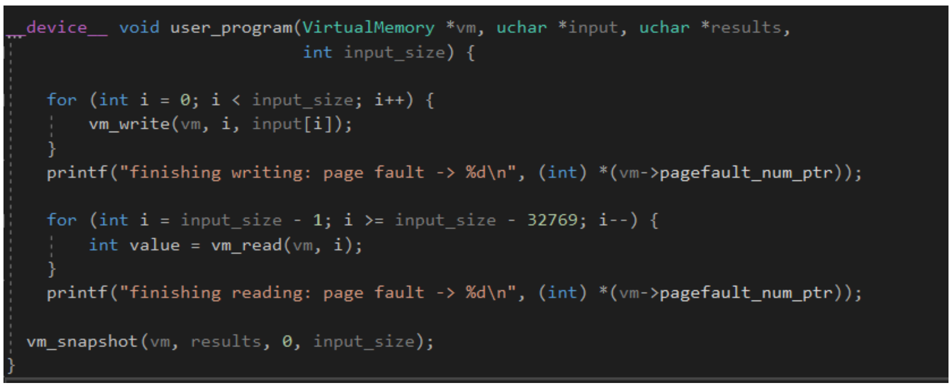
The logic is quite similar to the vm_write( ) function, except that the function is reading data instead of writing data.

d) vm_snapshot(`VirtualMemory *vm, uchar *results, int offset,int input_size`)

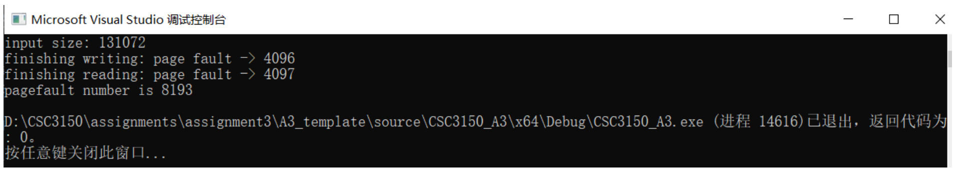
This function will use vm_read( ) to transfer the data of the address from `offset` to `input_size` to the `results` buffer.

4. Page fault of the program:

1) We can modify the user_program to show the page fault of each stages clearly. We can print the page fault of different stages of the function:


```
__device__ void user_program(VirtualMemory *vm, uchar *input, uchar *results,
                             int input_size) {

    for (int i = 0; i < input_size; i++) {
        vm_write(vm, i, input[i]);
    }
    printf("finishing writing: page fault -> %d\n", (int) *(vm->pagefault_num_ptr));

    for (int i = input_size - 1; i >= input_size - 32769; i--) {
        int value = vm_read(vm, i);
    }
    printf("finishing reading: page fault -> %d\n", (int) *(vm->pagefault_num_ptr));

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

2) outputs:



```
input size: 131072
finishing writing: page fault -> 4096
finishing reading: page fault -> 4097
pagefault number is 8193

D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3\x64\Debug\CSC3150_A3.exe (进程 14616)已退出，返回代码为
: 0。
按任意键关闭此窗口. . .
```

(the last line of Chinese in the terminal means the programs execute successfully)

This result shows that the final page fault numbers can be decomposed into 4096+1+4096.

a. In the first write iteration, the program writes the data of address from 0 to 131071 consecutively. Every time when writing the page that is multiple of 32, (like 0, 32, …), the function would generate a page fault. After that, the next 31 data would not generate page fault, since they are in the same page of that data of address in the multiple of 32. Totally, there are 131072/32 = 4096 of that kind of data (whose address of is multiple of 32) , which will generate 4096 page faults.

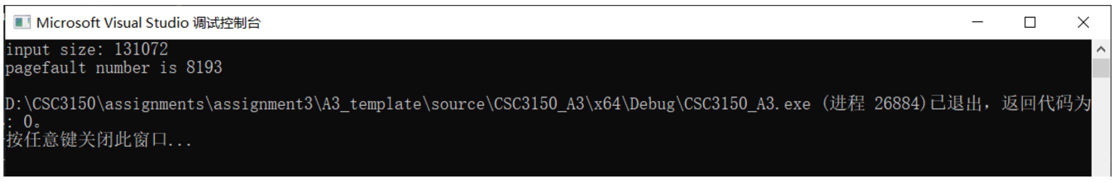
b. After writing, the program would read 32769 pieces of data from address 131071 consecutively in the decreasing order. Since the first 1024 pages of the last 32768 pieces of data which means they will not generate any page fault. However, for the 32769th data, its page does not in the table, which will generate 1 page fault to bring that page in physical memory.

c. The vm_snapshot function will transfer the date of all 131072 data from address 0 consecutively in the increasing order. Notice that since it is transfer data from address 0, the previous 1024 pages in the page table won’t get any page hit since the pages are set for last 1024 pages of data.

In this case, the situation is quite similar to the vm_write process. Then it will also generate 4096 page faults.

5. Outputs (with the original user_program( ))

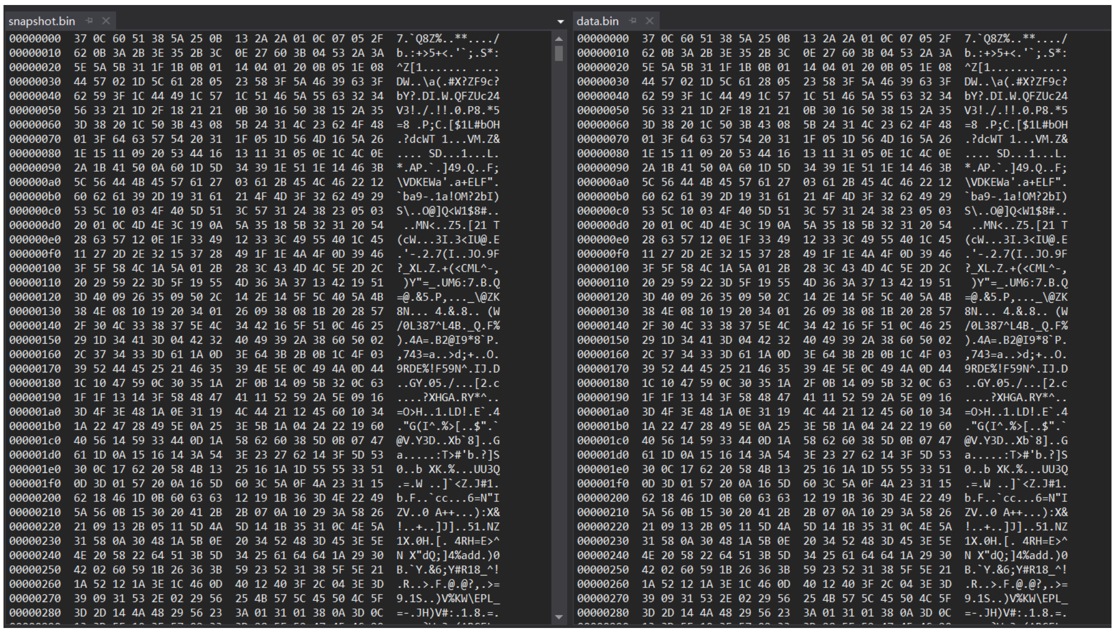
   a. terminal:



```
Microsoft Visual Studio 调试控制台
input size: 131072
pagefault number is 8193

D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3\x64\Debug\CSC3150_A3.exe (进程 26884)已退出，返回代码为
: 0。
按任意键关闭此窗口...
```

   (the last line of Chinese in the terminal means the programs execute successfully)

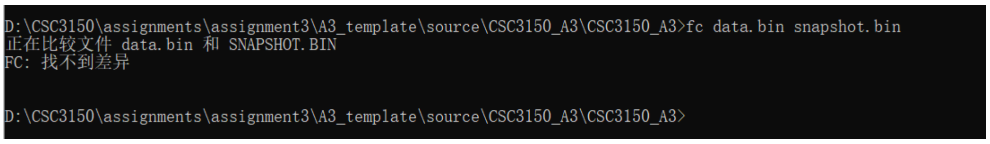
   b. snapshot.bin



```
snapshot.bin
00000000  37 0C 60 51 38 5A 25 0B  13 2A 2A 01 0C 07 05 2F  7.`Q8Z%..**..../
00000010  62 0B 3A 2B 3E 35 2B 3C  0E 27 60 3B 04 53 2A 3A  b.:+>5+<.'`;.S*:
00000020  5E 5A 5B 31 1F 1B 0B 01  14 04 01 20 0B 05 1E 08  ^Z[1....... ....
00000030  44 57 02 1D 5C 61 28 05  23 58 3F 5A 46 39 63 3F  DW..\a(.#X?ZF9c?
00000040  62 59 3F 1C 44 49 1C 57  1C 51 46 5A 55 63 32 34  bY?.DI.W.QFZUc24
00000050  56 33 21 1D 2F 18 21 21  0B 30 16 50 38 15 2A 35  V3!./.!!.0.P8.*5
00000060  3D 38 20 1C 50 3B 43 08  5B 24 31 4C 23 62 4F 48  =8 .P;C.[$1L#bOH
00000070  01 3F 64 63 57 54 20 31  1F 05 1D 56 4D 16 5A 26  .?dcWT 1...VM.Z&
00000080  1E 15 11 09 20 53 44 16  13 11 31 05 0E 1C 4C 0E  .... SD...1...L.
00000090  2A 1B 41 50 0A 60 1D 5D  34 39 1E 51 1E 14 46 3B  *.AP.`.]49.Q..F;
000000a0  5C 56 44 4B 45 57 61 27  03 61 2B 45 4C 46 22 12  \VDKEWa'.a+ELF".
000000b0  60 62 61 39 2D 19 31 61  21 4F 4D 3F 32 62 49 29  `ba9-.1a!OM?2bI)
000000c0  53 5C 10 03 4F 40 5D 51  3C 57 31 24 38 23 05 03  S\..O@]Q<W1$8#..
000000d0  20 01 0C 4D 4E 3C 19 0A  5A 35 18 5B 32 31 20 54   ..MN<..Z5.[21 T
000000e0  28 63 57 12 0E 1F 33 49  12 33 3C 49 55 40 1C 45  (cW...3I.3<IU@.E
000000f0  11 27 2D 2E 32 15 37 28  49 1F 1E 4A 4F 0D 39 46  .'-.2.7(I..JO.9F
00000100  3F 5F 58 4C 1A 5A 01 2B  28 3C 43 4D 4C 5E 2D 2C  ?_XL.Z.+(<CML^-,
00000110  20 29 59 22 3D 5F 19 55  4D 36 3A 37 13 42 19 51   )Y"=_.UM6:7.B.Q
00000120  3D 40 09 26 35 09 50 2C  14 2E 14 5F 5C 40 5A 4B  =@.&5.P,..._\@ZK
00000130  38 4E 08 10 19 20 34 01  26 09 38 08 1B 20 28 57  8N... 4.&.8.. (W
00000140  2F 30 4C 33 38 37 5E 4C  34 42 16 5F 51 0C 46 25  /0L387^L4B._Q.F%
00000150  29 1D 34 41 3D 04 42 32  40 49 39 2A 38 60 50 02  ).4A=.B2@I9*8`P.
00000160  2C 37 34 33 3D 61 1A 0D  3E 64 3B 2B 0B 1C 4F 03  ,743=a..>d;+..O.
00000170  39 52 44 45 25 21 46 35  39 4E 5E 0C 49 4A 0D 44  9RDE%!F59N^.IJ.D
00000180  1C 10 47 59 0C 30 35 1A  2F 0B 14 09 5B 32 0C 63  ..GY.05./...[2.c
00000190  1F 1F 13 14 3F 58 48 47  41 11 52 59 2A 5E 09 16  ....?XHGA.RY*^..
000001a0  3D 4F 3E 48 1A 0E 31 19  4C 44 21 12 45 60 10 34  =O>H..1.LD!.E`.4
000001b0  1A 22 47 28 49 5E 0A 25  3E 5B 1A 04 24 22 19 60  ."G(I^.%>[..$".`
000001c0  40 56 14 59 33 44 0D 1A  58 62 60 38 5D 0B 07 47  @V.Y3D..Xb`8]..G
000001d0  61 1D 0A 15 16 14 3A 54  3E 23 27 62 14 3F 5D 53  a.....:T>#'b.?]S
000001e0  30 0C 17 62 20 58 4B 13  25 16 1A 1D 55 55 33 51  0..b XK.%...UU3Q
000001f0  0D 3D 01 57 20 0A 16 5D  60 3C 5A 0F 4A 23 31 15  .=.W ..]`<Z.J#1.
00000200  62 18 46 1D 0B 60 63 63  12 19 1B 36 3D 4E 22 49  b.F..`cc...6=N"I
00000210  5A 56 0B 15 30 20 41 2B  2B 07 0A 10 29 3A 58 26  ZV..0 A++...):X&
00000220  21 09 13 2B 05 11 5D 4A  5D 14 1B 35 31 0C 4E 5A  !..+..]J]..51.NZ
00000230  31 58 0A 30 48 1A 5B 0E  20 34 52 48 3D 45 3E 5E  1X.0H.[. 4RH=E>^
00000240  4E 20 58 22 64 51 3B 5D  34 25 61 64 64 1A 29 30  N X"dQ;]4%add.)0
00000250  42 02 60 59 1B 26 36 3B  59 23 52 31 38 5F 5E 21  B.`Y.&6;Y#R18_^!
00000260  1A 52 12 1A 3E 1C 46 0D  40 12 40 3F 2C 04 3E 3D  .R..>.F.@.@?,.>=
00000270  39 09 31 53 2E 02 29 56  25 4B 57 5C 45 50 4C 5F  9.1S..)V%KW\EPL_
00000280  3D 2D 14 4A 48 29 56 23  3A 01 31 01 38 0A 3D 0C  =-.JH)V#:.1.8.=.
```

```
data.bin
00000000  37 0C 60 51 38 5A 25 0B  13 2A 2A 01 0C 07 05 2F  7.`Q8Z%..**..../
00000010  62 0B 3A 2B 3E 35 2B 3C  0E 27 60 3B 04 53 2A 3A  b.:+>5+<.'`;.S*:
00000020  5E 5A 5B 31 1F 1B 0B 01  14 04 01 20 0B 05 1E 08  ^Z[1....... ....
00000030  44 57 02 1D 5C 61 28 05  23 58 3F 5A 46 39 63 3F  DW..\a(.#X?ZF9c?
00000040  62 59 3F 1C 44 49 1C 57  1C 51 46 5A 55 63 32 34  bY?.DI.W.QFZUc24
00000050  56 33 21 1D 2F 18 21 21  0B 30 16 50 38 15 2A 35  V3!./.!!.0.P8.*5
00000060  3D 38 20 1C 50 3B 43 08  5B 24 31 4C 23 62 4F 48  =8 .P;C.[$1L#bOH
00000070  01 3F 64 63 57 54 20 31  1F 05 1D 56 4D 16 5A 26  .?dcWT 1...VM.Z&
00000080  1E 15 11 09 20 53 44 16  13 11 31 05 0E 1C 4C 0E  .... SD...1...L.
00000090  2A 1B 41 50 0A 60 1D 5D  34 39 1E 51 1E 14 46 3B  *.AP.`.]49.Q..F;
000000a0  5C 56 44 4B 45 57 61 27  03 61 2B 45 4C 46 22 12  \VDKEWa'.a+ELF".
000000b0  60 62 61 39 2D 19 31 61  21 4F 4D 3F 32 62 49 29  `ba9-.1a!OM?2bI)
000000c0  53 5C 10 03 4F 40 5D 51  3C 57 31 24 38 23 05 03  S\..O@]Q<W1$8#..
000000d0  20 01 0C 4D 4E 3C 19 0A  5A 35 18 5B 32 31 20 54   ..MN<..Z5.[21 T
000000e0  28 63 57 12 0E 1F 33 49  12 33 3C 49 55 40 1C 45  (cW...3I.3<IU@.E
000000f0  11 27 2D 2E 32 15 37 28  49 1F 1E 4A 4F 0D 39 46  .'-.2.7(I..JO.9F
00000100  3F 5F 58 4C 1A 5A 01 2B  28 3C 43 4D 4C 5E 2D 2C  ?_XL.Z.+(<CML^-,
00000110  20 29 59 22 3D 5F 19 55  4D 36 3A 37 13 42 19 51   )Y"=_.UM6:7.B.Q
00000120  3D 40 09 26 35 09 50 2C  14 2E 14 5F 5C 40 5A 4B  =@.&5.P,..._\@ZK
00000130  38 4E 08 10 19 20 34 01  26 09 38 08 1B 20 28 57  8N... 4.&.8.. (W
00000140  2F 30 4C 33 38 37 5E 4C  34 42 16 5F 51 0C 46 25  /0L387^L4B._Q.F%
00000150  29 1D 34 41 3D 04 42 32  40 49 39 2A 38 60 50 02  ).4A=.B2@I9*8`P.
00000160  2C 37 34 33 3D 61 1A 0D  3E 64 3B 2B 0B 1C 4F 03  ,743=a..>d;+..O.
00000170  39 52 44 45 25 21 46 35  39 4E 5E 0C 49 4A 0D 44  9RDE%!F59N^.IJ.D
00000180  1C 10 47 59 0C 30 35 1A  2F 0B 14 09 5B 32 0C 63  ..GY.05./...[2.c
00000190  1F 1F 13 14 3F 58 48 47  41 11 52 59 2A 5E 09 16  ....?XHGA.RY*^..
000001a0  3D 4F 3E 48 1A 0E 31 19  4C 44 21 12 45 60 10 34  =O>H..1.LD!.E`.4
000001b0  1A 22 47 28 49 5E 0A 25  3E 5B 1A 04 24 22 19 60  ."G(I^.%>[..$".`
000001c0  40 56 14 59 33 44 0D 1A  58 62 60 38 5D 0B 07 47  @V.Y3D..Xb`8]..G
000001d0  61 1D 0A 15 16 14 3A 54  3E 23 27 62 14 3F 5D 53  a.....:T>#'b.?]S
000001e0  30 0C 17 62 20 58 4B 13  25 16 1A 1D 55 55 33 51  0..b XK.%...UU3Q
000001f0  0D 3D 01 57 20 0A 16 5D  60 3C 5A 0F 4A 23 31 15  .=.W ..]`<Z.J#1.
00000200  62 18 46 1D 0B 60 63 63  12 19 1B 36 3D 4E 22 49  b.F..`cc...6=N"I
00000210  5A 56 0B 15 30 20 41 2B  2B 07 0A 10 29 3A 58 26  ZV..0 A++...):X&
00000220  21 09 13 2B 05 11 5D 4A  5D 14 1B 35 31 0C 4E 5A  !..+..]J]..51.NZ
00000230  31 58 0A 30 48 1A 5B 0E  20 34 52 48 3D 45 3E 5E  1X.0H.[. 4RH=E>^
00000240  4E 20 58 22 64 51 3B 5D  34 25 61 64 64 1A 29 30  N X"dQ;]4%add.)0
00000250  42 02 60 59 1B 26 36 3B  59 23 52 31 38 5F 5E 21  B.`Y.&6;Y#R18_^!
00000260  1A 52 12 1A 3E 1C 46 0D  40 12 40 3F 2C 04 3E 3D  .R..>.F.@.@?,.>=
00000270  39 09 31 53 2E 02 29 56  25 4B 57 5C 45 50 4C 5F  9.1S..)V%KW\EPL_
00000280  3D 2D 14 4A 48 29 56 23  3A 01 31 01 38 0A 3D 0C  =-.JH)V#:.1.8.=.
```

   file compare result:



```
D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3\CSC3150_A3>fc data.bin snapshot.bin
正在比较文件 data.bin 和 SNAPSHOT.BIN
FC: 找不到差异

D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3\CSC3150_A3>
```

(the result means there are no difference between two files)

6. Problem met in the project

   When I implemented the first version of the project, it last about 15 min to output the final result. And I found that the main reason for that is I need to go through all the elements in page tables at every function call to search for the head and tail of the LRU linked list. Then I decided to hide them in the lower bit in the first two elements of the page table. In this way, it save a lot of time to search for the head and tail of the LRU list and reduce the running time a lot.

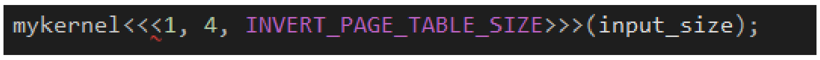
7. What I learn from the project

   Firstly, I learnt some basic knowledge about cuda programming like declaring variables and functions in different positions using _device_ and _host_ . Also, I learnt about the basic structure and mechanism of the virtual memory, and also the detail implementation of simple invert page table. I also learnt about the LRU mechanism and know how to implement it using double linked list.

8. Bonus

   The main structure of virtual memory of the bonus part is same as the project, so we can do some modifications to finish the bonus part.

   a. First, we need to launch 4 cuda threads to run the program, which means that in the main.cu function, the kernel part is needed to be changed like this:

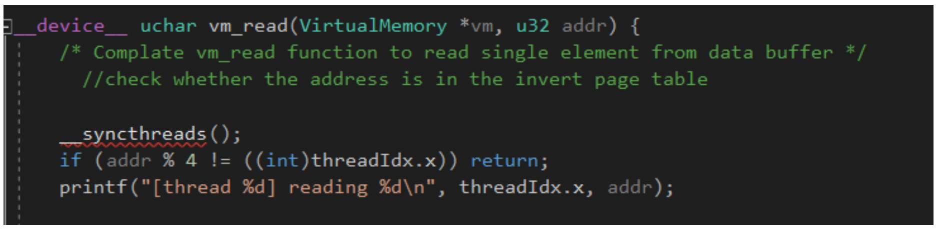
      

      ```
      mykernel<<<1, 4, INVERT_PAGE_TABLE_SIZE>>>(input_size);
      ```

   b. After testing, I found that for any function call for vm_read, vm_write and vm_snapshot, the program would use 4 threads to perform the exact same function call. Due to this feature, we can use their thread ID to control some threads when their execution. Since the user program always read or write date in consecutive order, in this case, to maintain the non-preemptive order, we can just require each threads to perform the read or write data of corresponding address, that is, we always allow thread 0, 1, 2, 3 to perform the read/write function for data of address that satisfy addr%4 = 0, 1, 2 and 3 respectively. Also, since for each call of a function, only one thread will do the corresponding operations, which means there will be no threads racing for the same resource of memory.

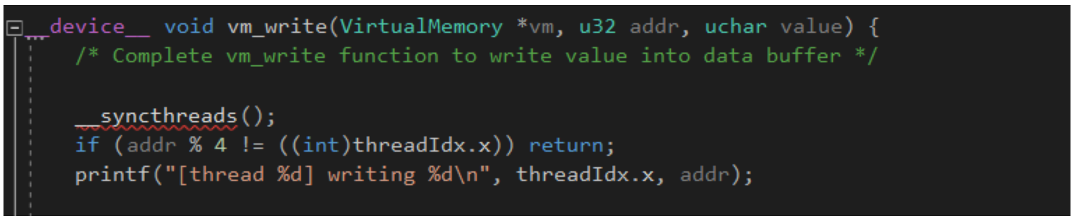
c. For vm_read and vm_write function, we can just add the following codes at the beginning to control the threads:

a) vm_read



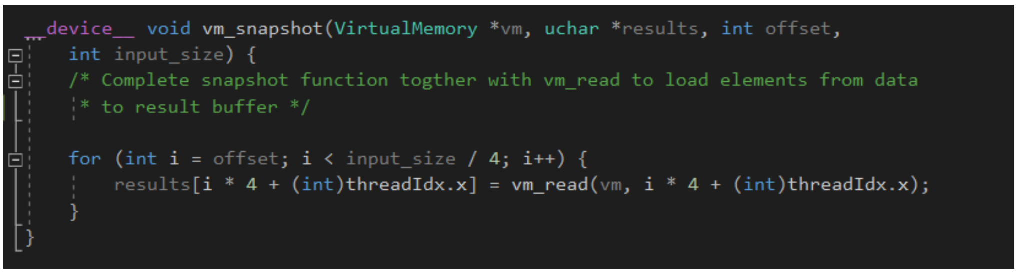
```
__device__ uchar vm_read(VirtualMemory *vm, u32 addr) {
    /* Complate vm_read function to read single element from data buffer */
      //check whether the address is in the invert page table

    __syncthreads();
    if (addr % 4 != ((int)threadIdx.x)) return;
    printf("[thread %d] reading %d\n", threadIdx.x, addr);
```

b) vm_write



```
__device__ void vm_write(VirtualMemory *vm, u32 addr, uchar value) {
    /* Complete vm_write function to write value into data buffer */

    __syncthreads();
    if (addr % 4 != ((int)threadIdx.x)) return;
    printf("[thread %d] writing %d\n", threadIdx.x, addr);
```

d. For vm_snapshot, since the program would use 4 threads to execute it, which means we can just reduce the iteration in it to 1/input_size and assign different threads with different task.

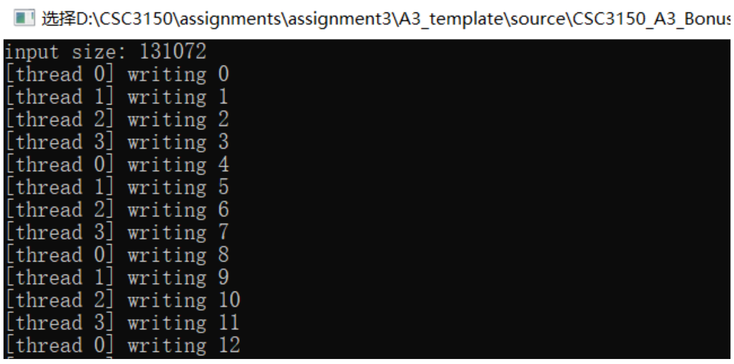


```
__device__ void vm_snapshot(VirtualMemory *vm, uchar *results, int offset,
    int input_size) {
    /* Complete snapshot function togther with vm_read to load elements from data
     * to result buffer */

    for (int i = offset; i < input_size / 4; i++) {
        results[i * 4 + (int)threadIdx.x] = vm_read(vm, i * 4 + (int)threadIdx.x);
    }
}
```

e. Outputs:

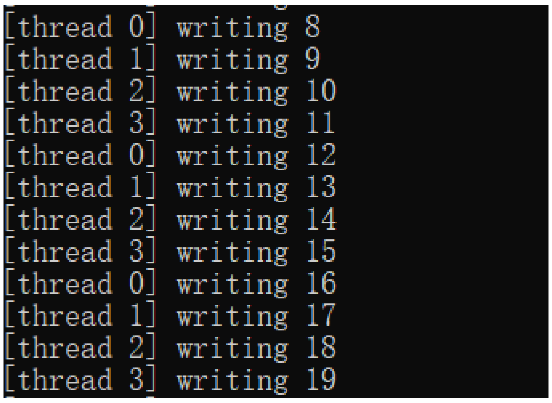
a) Terminal:

i. Start execution


选择D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3_Bonus

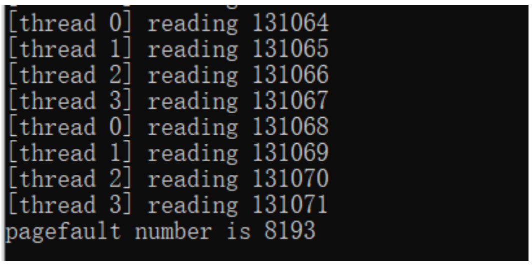
```
input size: 131072
[thread 0] writing 0
[thread 1] writing 1
[thread 2] writing 2
[thread 3] writing 3
[thread 0] writing 4
[thread 1] writing 5
[thread 2] writing 6
[thread 3] writing 7
[thread 0] writing 8
[thread 1] writing 9
[thread 2] writing 10
[thread 3] writing 11
[thread 0] writing 12
```

ii. During execution



```
[thread 0] writing 8
[thread 1] writing 9
[thread 2] writing 10
[thread 3] writing 11
[thread 0] writing 12
[thread 1] writing 13
[thread 2] writing 14
[thread 3] writing 15
[thread 0] writing 16
[thread 1] writing 17
[thread 2] writing 18
[thread 3] writing 19
```

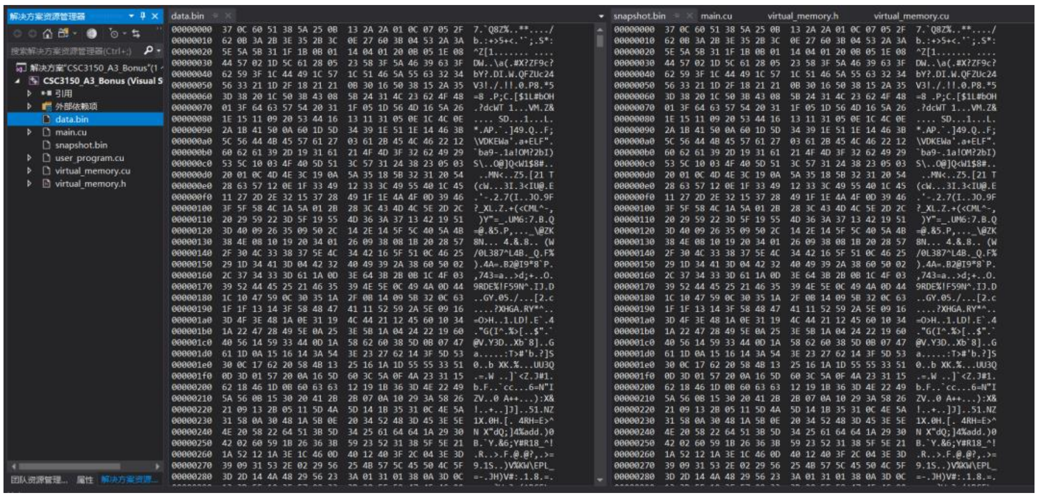
iii. Finishing execution



```
[thread 0] reading 131064
[thread 1] reading 131065
[thread 2] reading 131066
[thread 3] reading 131067
[thread 0] reading 131068
[thread 1] reading 131069
[thread 2] reading 131070
[thread 3] reading 131071
pagefault number is 8193
```

b) Snapshot.bin:

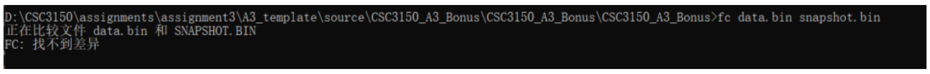
i. Output file:



```
00000000  37 0C 60 51 38 5A 25 0B  13 2A 2A 01 0C 07 05 2F  7.`Q8Z%..**..../
00000010  62 0B 3A 2B 3E 35 2B 3C  0E 27 60 3B 04 53 2A 3A  b.:+>5+<.'`;.S*:
00000020  5E 5A 5B 31 1F 1B 0B 01  14 04 01 20 0B 05 1E 08  ^Z[1....... ....
00000030  44 57 02 1D 5C 61 28 05  23 58 3F 5A 46 39 63 3F  DW..\a(.#X?ZF9c?
00000040  62 59 3F 1C 44 49 1C 57  1C 51 46 5A 55 63 32 34  bY?.DI.W.QFZUc24
00000050  56 33 21 1D 2F 18 21 21  0B 30 16 50 38 15 2A 35  V3!./.!!.0.P8.*5
00000060  3D 38 20 1C 50 3B 43 08  5B 24 31 4C 23 62 4F 48  =8 .P;C.[$1L#bOH
00000070  01 3F 64 63 57 54 20 31  1F 05 1D 56 4D 16 5A 26  .?dcWT 1...VM.Z&
00000080  1E 15 11 09 20 53 44 16  13 11 31 05 0E 1C 4C 0E  .... SD...1...L.
00000090  2A 1B 41 50 0A 60 1D 5D  34 39 1E 51 1E 14 46 3B  *.AP.`.]49.Q..F;
000000a0  5C 56 44 4B 45 57 61 27  03 61 2B 45 4C 46 22 12  \VDKEWa'.a+ELF".
000000b0  60 62 61 39 2D 19 31 61  21 4F 4D 3F 32 62 49 29  `ba9-.1a!OM?2bI)
000000c0  53 5C 10 03 4F 40 5D 51  3C 57 31 24 38 23 05 03  S\..O@]Q<W1$8#..
000000d0  20 01 0C 4D 4E 3C 19 0A  5A 35 18 5B 32 31 20 54   ..MN<..Z5.[21 T
000000e0  28 63 57 12 0E 1F 33 49  12 33 3C 49 55 40 1C 45  (cW...3I.3<IU@.E
000000f0  11 27 2D 2E 32 15 37 28  49 1F 1E 4A 4F 0D 39 46  .'-.2.7(I..JO.9F
00000100  3F 5F 58 4C 1A 5A 01 2B  28 3C 43 4D 4C 5E 2D 2C  ?_XL.Z.+(<CML^-,
00000110  20 29 59 22 3D 5F 19 55  4D 36 3A 37 13 42 19 51   )Y"=_.UM6:7.B.Q
00000120  3D 40 09 26 35 09 50 2C  14 2E 14 5F 5C 40 5A 4B  =@.&5.P,..._\@ZK
00000130  38 4E 08 10 19 20 34 01  26 09 38 08 1B 20 28 57  8N... 4.&.8.. (W
00000140  2F 30 4C 33 38 37 5E 4C  34 42 16 5F 51 0C 46 25  /0L387^L4B._Q.F%
00000150  29 1D 34 41 3D 04 42 32  40 49 39 2A 38 60 50 02  ).4A=.B2@I9*8`P.
00000160  2C 37 34 33 3D 61 1A 0D  3E 64 3B 2B 0B 1C 4F 03  ,743=a..>d;+..O.
00000170  39 52 44 45 25 21 46 35  39 4E 5E 0C 49 4A 0D 44  9RDE%!F59N^.IJ.D
00000180  1C 10 47 59 0C 30 35 1A  2F 0B 14 09 5B 32 0C 63  ..GY.05./...[2.c
00000190  1F 1F 13 14 3F 58 48 47  41 11 52 59 2A 5E 09 16  ....?XHGA.RY*^..
000001a0  3D 4F 3E 48 1A 0E 31 19  4C 44 21 12 45 60 10 34  =O>H..1.LD!.E`.4
000001b0  1A 22 47 28 49 5E 0A 25  3E 5B 1A 04 24 22 19 60  ."G(I^.%>[..$".`
000001c0  40 56 14 59 33 44 0D 1A  58 62 60 38 5D 0B 07 47  @V.Y3D..Xb`8]..G
000001d0  61 1D 0A 15 16 14 3A 54  3E 23 27 62 14 3F 5D 53  a.....:T>#'b.?]S
000001e0  30 0C 17 62 20 58 4B 13  25 16 1A 1D 55 55 33 51  0..b XK.%...UU3Q
000001f0  0D 3D 01 57 20 0A 16 5D  60 3C 5A 0F 4A 23 31 15  .=.W ..]`<Z.J#1.
00000200  62 18 46 1D 0B 60 63 63  12 19 1B 36 3D 4E 22 49  b.F..`cc...6=N"I
00000210  5A 56 0B 15 30 20 41 2B  2B 07 0A 10 29 3A 58 26  ZV..0 A++...):X&
00000220  21 09 13 2B 05 11 5D 4A  5D 14 1B 35 31 0C 4E 5A  !..+..]J]..51.NZ
00000230  31 58 0A 30 48 1A 5B 0E  20 34 52 48 3D 45 3E 5E  1X.0H.[. 4RH=E>^
00000240  4E 20 58 22 64 51 3B 5D  34 25 61 64 64 1A 29 30  N X"dQ;]4%add.)0
00000250  42 02 60 59 1B 26 36 3B  59 23 52 31 38 5F 5E 21  B.`Y.&6;Y#R18_^!
00000260  1A 52 12 1A 3E 1C 46 0D  40 12 40 3F 2C 04 3E 3D  .R..>.F.@.@?,.>=
00000270  39 09 31 53 2E 02 29 56  25 4B 57 5C 45 50 4C 5F  9.1S..)V%KW\EPL_
00000280  3D 2D 14 4A 48 29 56 23  3A 01 31 01 38 0A 3D 0C  =-.JH)V#:.1.8.=.
```

```
00000000  37 0C 60 51 38 5A 25 0B  13 2A 2A 01 0C 07 05 2F  7.`Q8Z%..**..../
00000010  62 0B 3A 2B 3E 35 2B 3C  0E 27 60 3B 04 53 2A 3A  b.:+>5+<.'`;.S*:
00000020  5E 5A 5B 31 1F 1B 0B 01  14 04 01 20 0B 05 1E 08  ^Z[1....... ....
00000030  44 57 02 1D 5C 61 28 05  23 58 3F 5A 46 39 63 3F  DW..\a(.#X?ZF9c?
00000040  62 59 3F 1C 44 49 1C 57  1C 51 46 5A 55 63 32 34  bY?.DI.W.QFZUc24
00000050  56 33 21 1D 2F 18 21 21  0B 30 16 50 38 15 2A 35  V3!./.!!.0.P8.*5
00000060  3D 38 20 1C 50 3B 43 08  5B 24 31 4C 23 62 4F 48  =8 .P;C.[$1L#bOH
00000070  01 3F 64 63 57 54 20 31  1F 05 1D 56 4D 16 5A 26  .?dcWT 1...VM.Z&
00000080  1E 15 11 09 20 53 44 16  13 11 31 05 0E 1C 4C 0E  .... SD...1...L.
00000090  2A 1B 41 50 0A 60 1D 5D  34 39 1E 51 1E 14 46 3B  *.AP.`.]49.Q..F;
000000a0  5C 56 44 4B 45 57 61 27  03 61 2B 45 4C 46 22 12  \VDKEWa'.a+ELF".
000000b0  60 62 61 39 2D 19 31 61  21 4F 4D 3F 32 62 49 29  `ba9-.1a!OM?2bI)
000000c0  53 5C 10 03 4F 40 5D 51  3C 57 31 24 38 23 05 03  S\..O@]Q<W1$8#..
000000d0  20 01 0C 4D 4E 3C 19 0A  5A 35 18 5B 32 31 20 54   ..MN<..Z5.[21 T
000000e0  28 63 57 12 0E 1F 33 49  12 33 3C 49 55 40 1C 45  (cW...3I.3<IU@.E
000000f0  11 27 2D 2E 32 15 37 28  49 1F 1E 4A 4F 0D 39 46  .'-.2.7(I..JO.9F
00000100  3F 5F 58 4C 1A 5A 01 2B  28 3C 43 4D 4C 5E 2D 2C  ?_XL.Z.+(<CML^-,
00000110  20 29 59 22 3D 5F 19 55  4D 36 3A 37 13 42 19 51   )Y"=_.UM6:7.B.Q
00000120  3D 40 09 26 35 09 50 2C  14 2E 14 5F 5C 40 5A 4B  =@.&5.P,..._\@ZK
00000130  38 4E 08 10 19 20 34 01  26 09 38 08 1B 20 28 57  8N... 4.&.8.. (W
00000140  2F 30 4C 33 38 37 5E 4C  34 42 16 5F 51 0C 46 25  /0L387^L4B._Q.F%
00000150  29 1D 34 41 3D 04 42 32  40 49 39 2A 38 60 50 02  ).4A=.B2@I9*8`P.
00000160  2C 37 34 33 3D 61 1A 0D  3E 64 3B 2B 0B 1C 4F 03  ,743=a..>d;+..O.
00000170  39 52 44 45 25 21 46 35  39 4E 5E 0C 49 4A 0D 44  9RDE%!F59N^.IJ.D
00000180  1C 10 47 59 0C 30 35 1A  2F 0B 14 09 5B 32 0C 63  ..GY.05./...[2.c
00000190  1F 1F 13 14 3F 58 48 47  41 11 52 59 2A 5E 09 16  ....?XHGA.RY*^..
000001a0  3D 4F 3E 48 1A 0E 31 19  4C 44 21 12 45 60 10 34  =O>H..1.LD!.E`.4
000001b0  1A 22 47 28 49 5E 0A 25  3E 5B 1A 04 24 22 19 60  ."G(I^.%>[..$".`
000001c0  40 56 14 59 33 44 0D 1A  58 62 60 38 5D 0B 07 47  @V.Y3D..Xb`8]..G
000001d0  61 1D 0A 15 16 14 3A 54  3E 23 27 62 14 3F 5D 53  a.....:T>#'b.?]S
000001e0  30 0C 17 62 20 58 4B 13  25 16 1A 1D 55 55 33 51  0..b XK.%...UU3Q
000001f0  0D 3D 01 57 20 0A 16 5D  60 3C 5A 0F 4A 23 31 15  .=.W ..]`<Z.J#1.
00000200  62 18 46 1D 0B 60 63 63  12 19 1B 36 3D 4E 22 49  b.F..`cc...6=N"I
00000210  5A 56 0B 15 30 20 41 2B  2B 07 0A 10 29 3A 58 26  ZV..0 A++...):X&
00000220  21 09 13 2B 05 11 5D 4A  5D 14 1B 35 31 0C 4E 5A  !..+..]J]..51.NZ
00000230  31 58 0A 30 48 1A 5B 0E  20 34 52 48 3D 45 3E 5E  1X.0H.[. 4RH=E>^
00000240  4E 20 58 22 64 51 3B 5D  34 25 61 64 64 1A 29 30  N X"dQ;]4%add.)0
00000250  42 02 60 59 1B 26 36 3B  59 23 52 31 38 5F 5E 21  B.`Y.&6;Y#R18_^!
00000260  1A 52 12 1A 3E 1C 46 0D  40 12 40 3F 2C 04 3E 3D  .R..>.F.@.@?,.>=
00000270  39 09 31 53 2E 02 29 56  25 4B 57 5C 45 50 4C 5F  9.1S..)V%KW\EPL_
00000280  3D 2D 14 4A 48 29 56 23  3A 01 31 01 38 0A 3D 0C  =-.JH)V#:.1.8.=.
```

ii. File compare:



```
D:\CSC3150\assignments\assignment3\A3_template\source\CSC3150_A3_Bonus\CSC3150_A3_Bonus\CSC3150_A3_Bonus>fc data.bin snapshot.bin
正在比较文件 data.bin 和 SNAPSHOT.BIN
FC: 找不到差异
```

(the result means there are no difference between two files)