Design of MP6

**Basic Assignment**

In this MP, I only implemented the basic required disk function. What I submitted in the basic assignment folder include 8 files. In order to realize FIFO, I added *scheduler.H, scheduler.C*, and modified *makefile*. And I also modified *simple\_disk.C* and *simple\_disk.H* in order to derive blocking\_disk in a more convenient way. Also, in *kernel.C*, I enabled *\_USES\_SCHEDULER\_* and changed *SYSTEM\_DISK* to the pointer to the BlockingDisk object instead of the pointer to the SimpleDisk Object. So directly link these 8 files with other files provided on ecampus, we can realize the basic assignment of MP6. Now let’s begin.

In my submission, I set a blocked queue in the class of BlockingDisk. And I used the FIFO scheduler from the previous MP5, instead of taking time on implementing new scheduler with the interrupts currency disk system. However, in yield function, I made a little modification as is shown in Figure 1. Every time we yield the CPU, we check the status of registers of our device and decide whether resume the thread from the blocked queue in BlockingdDisk to the ready queue in the scheduler.

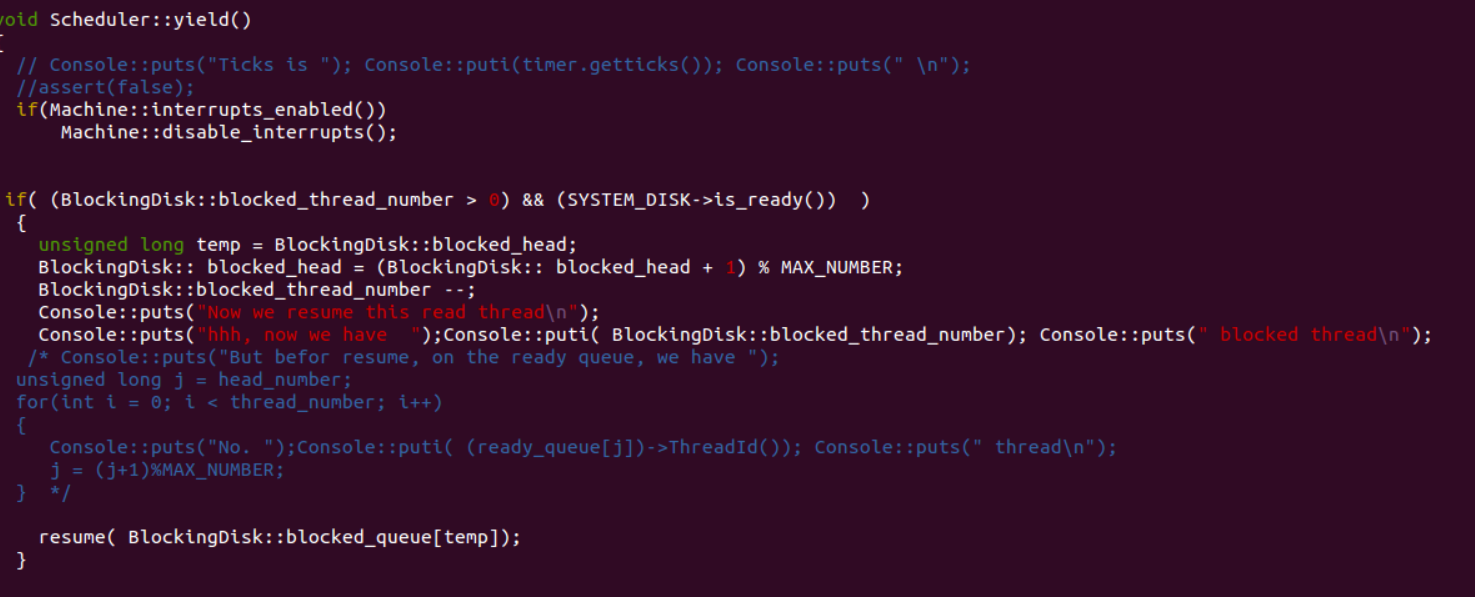


Figure 1

And the design of my BlockingDisk is quite simple. Instead of using busy waiting, we yield the CPU and dispatch to the next thread. The only tricky thing needed to present here is that Bochs really has a bad simulation of the write operation. In this condition, issue\_operation (write, \_block\_no) always return very fast. So in order to simulate the real condition, I force it to yield the CPU, as is shown in Figure 2. And our final result is shown in Figure 3. And I changed the output of the other three thread functions to highlight the result of thread 2, which does not have any negative influence on our project.



Figure 2

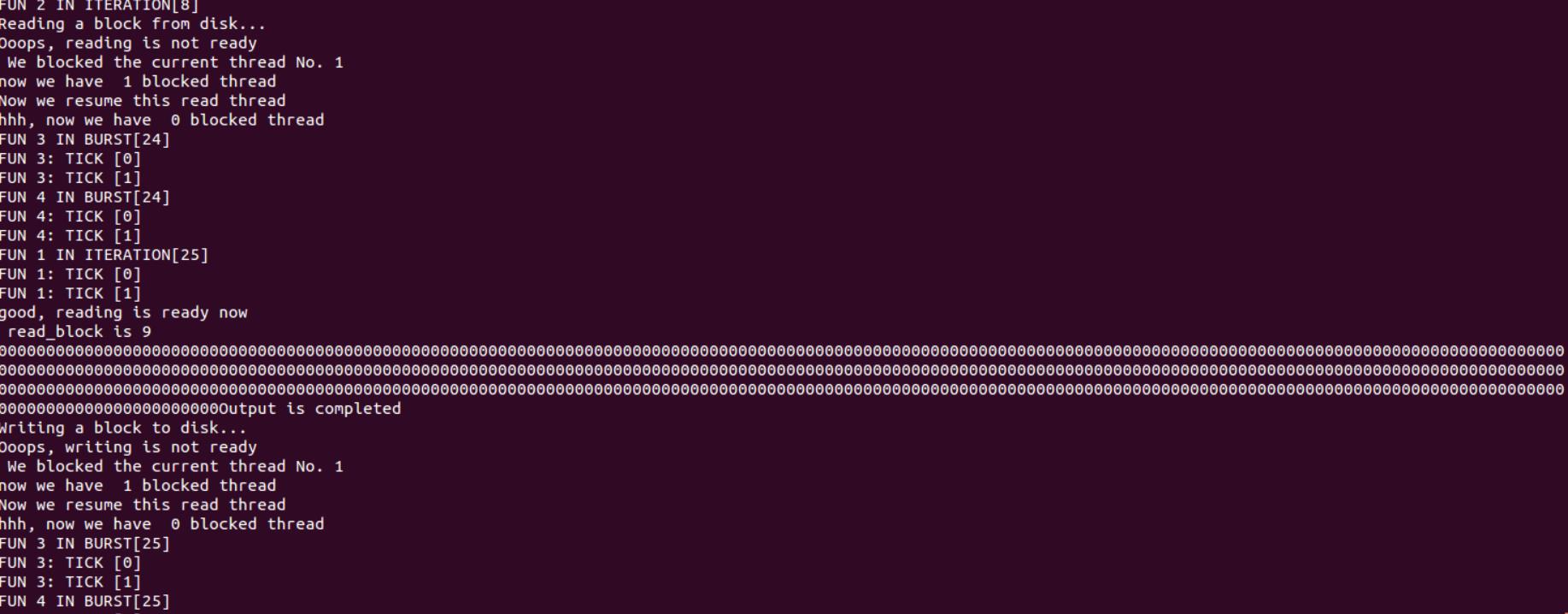


Figure 3