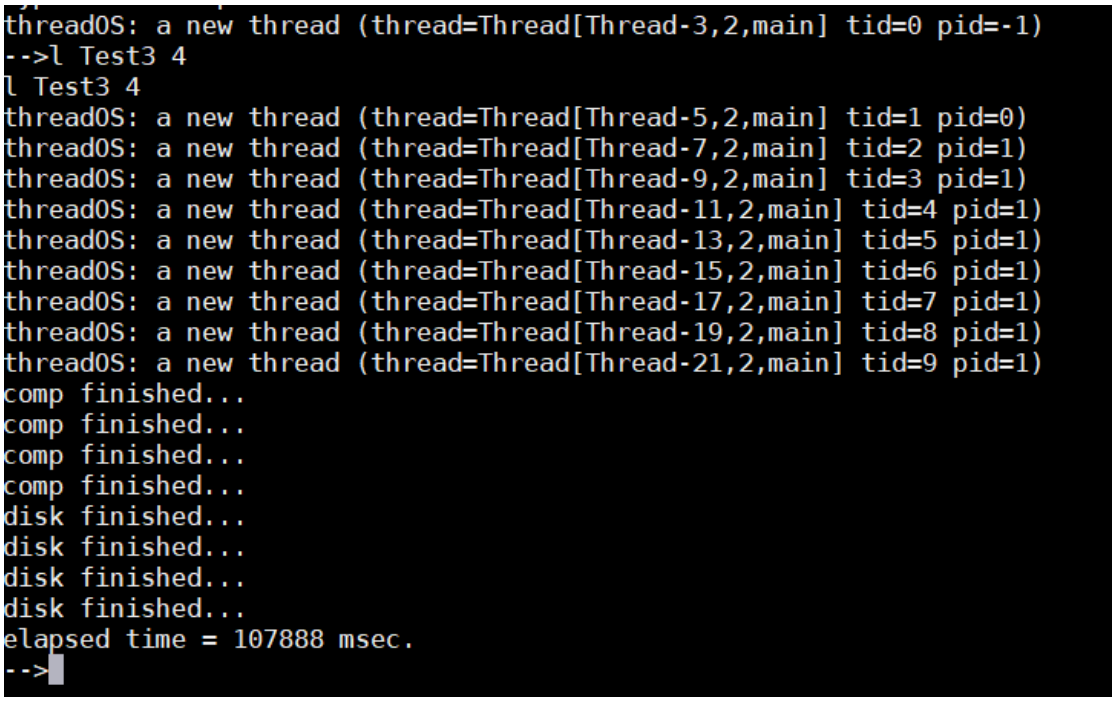
Part1:

SyncQueue works as a monitor for the SysLib.join() call. It will put current thread to sleep and keep track of thread. When SysLib.exit() called. Kernel will utilize the queue to wake up the thread waiting under the condition which is equal to current thread's parent ID. So that parent thread will be notified.

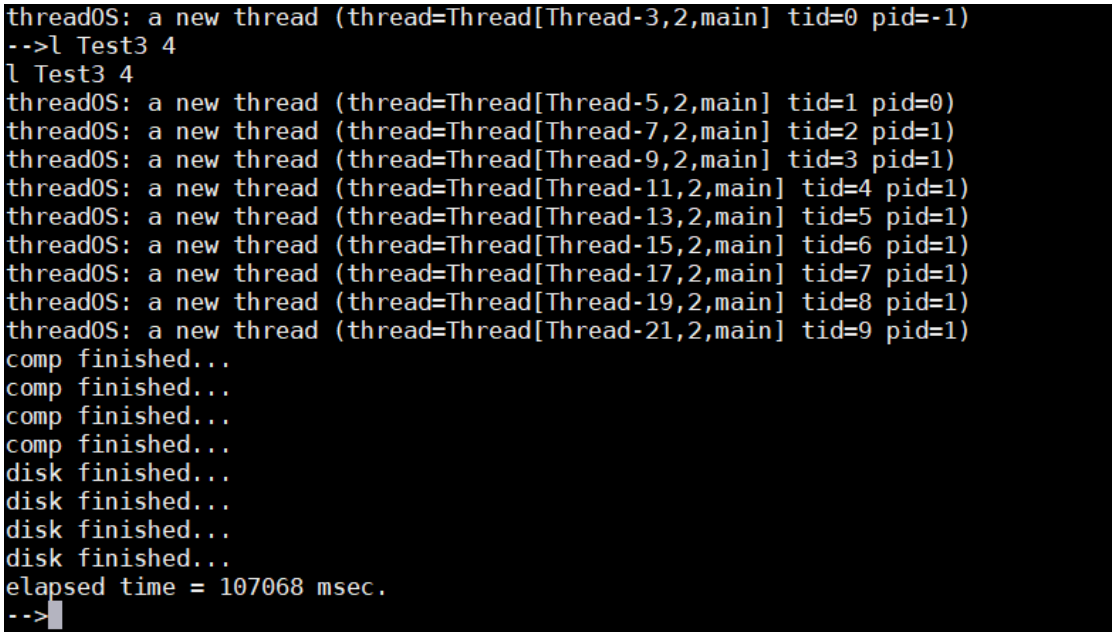
Part2:

When the disk busy waiting, user thread needs to keep checking the disk serve state in a spinning loop. So I use enqueueAndSleep to put the thread into wait, and let CPU to server other read thread. Test3.java will call Testthread3A (computation thread) and Testthread3B (disk thread) number of argument times and keep track the beginning and ending time for all the thread finished.

Result comparison:



(Spinning kernel)



(Non-spinning kernel)

According to the elapsed time results. The non-spinning kernel works better than the spinning one with 3 pairs of threads. And this advantage will be more distinctly when running large pairs of threads. This because when put the thread into waiting queue, it will wait until be waked up instead of keeping check the ready state. In this way, CPU resources will be relinquished and other ready thread can use these freed resources.