CSCE 410
MP6

Kranthi Kumar Katam
225009204

Files changed:
Kernel.C
Scheduler.H
Scheduler.C
Thread.H
Thread.C
simple_timer.C
Blocking_disk.C
Blocking_disk.H

In this MP, i have to implement a blocking disk drive on top of simple disk to make the disk without busy waiting.
I took the Scheduler from previous MP as it is working fine and added that Scheduler to this MP and changed the Kernel code accordingly to implement Scheduler.
The Scheduler implementation is RoundRobin which will raise the interrupt handler for every 50sec.

There are two ways to implement Blocking disk device:

1. Write code for Read and Write functions in Blocking disk. A new wait until ready is implemented in Blocking disk which will mask parent simple disk's wait until ready. When there is read or write operation then we have to check for is CPU ready for writing the operation on disk if not add the thread to ready queue implemented in Scheduler and yield the CPU for next threads.
2. Adding the current thread to ready queue in the scheduler which will give away the CPU for other threads until ready.

I have implemented the second step. Adding a new wait_until_ready in Blocking disk, this is important because whenever there is new simple disk operation is called then the wait_until_ready in Blocking disk will be used instead of parent Simple DIsk wait_until_ready. Already Is_ready is implemented in Simple disk so we can use that directly.

In this implementation there is no need of Read and Write in the Blocking disk class because we are only adding the Current thread to ready queue and yielding the CPU whenever the CPU is not ready for this current thread.