

Design Document | Operating Systems | MP6

In this machine problem we added a simple device driver to support the same blocking read and write operations as mentioned in the simple implementations, but here we do away with the `busy_wait`.

I have added the device driver code and also completion **Options 1 2 3 & 4**.

For this machine problem, I have touched the following files:

1. Makefile
2. Kernel.c
3. Blocking_disk.H
4. Blocking_disk.C
5. Mirroring_disk.H
6. Mirroring_disk.C
7. Scheduler.H
8. Scheduler.C

Here are the design details:

1. For implementing blocking disk, we inherit the class from the simple disk class already present in the folder, this file supports the reads and write without the busy waiting. I have implemented all the functions mentioned in the design and a few more. Detailed comments are added in the code.
2. For Option 1: we implement the `mirroring_disk.H` and `mirroring_disk.C` files, In this file the data is written both to the master and the dependent disks, based on the availability. Detailed implementation comments are added to the code.
3. For Option 2: here I have implemented interrupts and this interrupt is invoked once the disk is ready and data can be read/written from/to the disk. Here, we first register the interrupt handler and use the ready queue(push the thread id, if not ready into the queue, and yield CPU).
4. For Option 3, we have used mutex(mutual exclusion) locks. We have made the functions atomic by masking the interrupts and implementing `test_and_set` functions to protect the disk when the read/write API is called by user.
5. For Option 4: we implemented the design in above files, and added `test_and_set` and mutex lock for achieving thread safety.
6. We have added the macro `"ENABLE_BLOCKING_DISK"` for running the basic code, please comment on this macro to check option 1. To check options 3-4 please uncomment `"ENABLE_THREAD_SYNC"` macro if uncommented.

Here's the output snapshot:

```
csce410@csce410-VirtualBox: ~/Downloads/MP6Sources34/MP6_Sources
FUN 3 IN BURST[6]
FUN 3: TICK [0]
FUN 3: TICK [1]
FUN 3: TICK [2]
FUN 3: TICK [3]
FUN 3: TICK [4]
FUN 3: TICK [5]
FUN 3: TICK [6]
FUN 3: TICK [7]
FUN 3: TICK [8]
FUN 3: TICK [9]
FUN 4 IN BURST[6]
FUN 4: TICK [0]
FUN 4: TICK [1]
FUN 4: TICK [2]
FUN 4: TICK [3]
FUN 4: TICK [4]
FUN 4: TICK [5]
FUN 4: TICK [6]
FUN 4: TICK [7]
FUN 4: TICK [8]
=====
Bochs is exiting with the following message:
[XGUI ] POWER button turned off.
=====
```