CSCE 611
Operating systems

MP 6
Primitive disk device driver

Student: K Ram Sankar

This machine program assignment aims to implement primitive disk device driver.

I have also implemented all the bonus tasks (1, 2, 3, 4)

Initital implementation:

I have used the scheduler file from MP5 for this assignment. The blocking disk is derived from the simple disk class. The wait_until_ready() function contains the major change where the CPU is yielded to the next thread while the i/o operation of the current thread is being executed. This is done by calling the scheduler to yield the current thread

Option 1

Mirror Disk Implementation

This is done by using the design principle of composition over inheritance. The Mirror disk contains 2 blocking disks namely master and dependent.
Read function():
Here the is_ready function of the both master and dependent is called and based on the return value, we read the content from the master or the dependent disk
Note the yielding of the CPU is delegated to function present in the

Write function():
Here we directly write to both the master and dependent disks by calling the blocking disk write function

Option 2

Enable Interrupts

Here we register interrupt handler for interrupt 14 which indicates the completion of the disk operation in kernel.c. we use a blocking queue to store the threads which need to execute the i/o operation. When the disk operation is completed, we deque from the blocking queue and add it to the ready queue again.

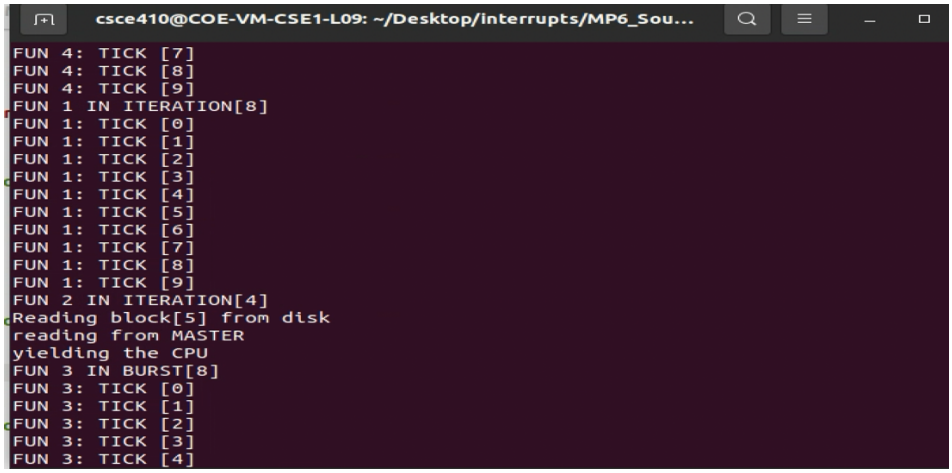
Option 3 and Option 4

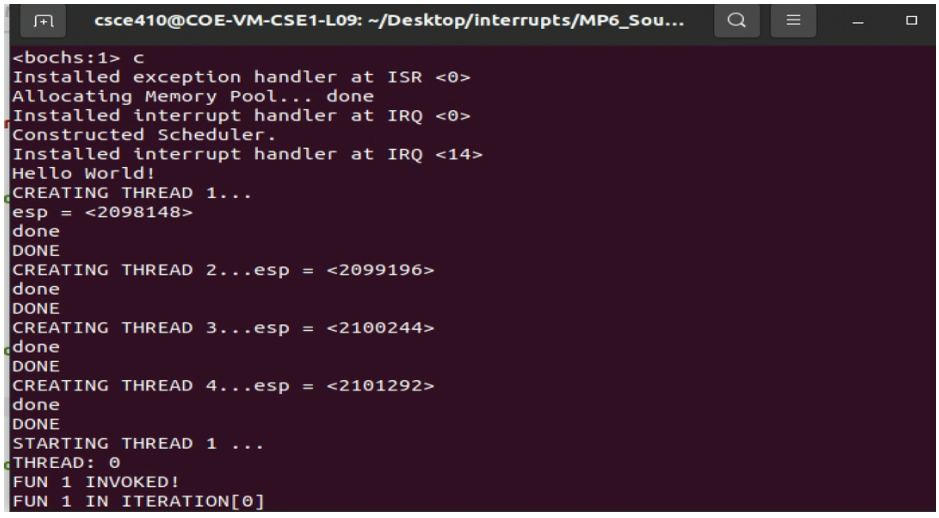
Design and Implement thread safe i/o operations

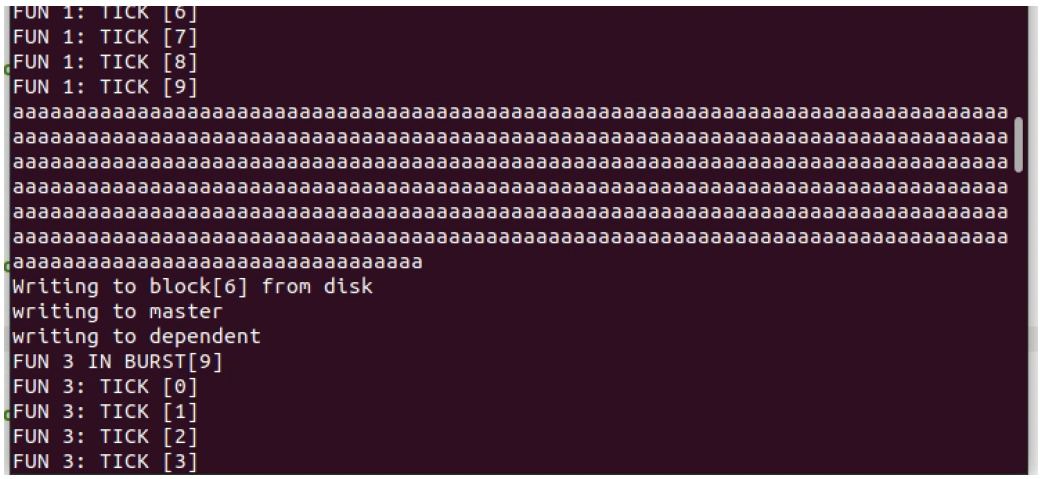
We use a mutex lock to prevent threads from issuing read and write operations at the same time. Acquire and release locks before and after the i/o operation. I have used test and set algorithm to implement the mutex lock

```
int mutex;
int TAS(int *mutex)
{
    int tmp = *mutex;
    *mutex = 1;
    return tmp;
}

void init(int *mutex)
{
   *mutex = 0;
}

void acquire()
{
   while(TAS(&mutex));
}

void release()
{
   mutex = 0;
}
```

Results


csce410@COE-VM-CSE1-L09: ~/Desktop/interrupts/MP6_Sou...
FUN 4: TICK [7]
FUN 4: TICK [8]
FUN 4: TICK [9]
FUN 1 IN ITERATION[8]
FUN 1: TICK [0]
FUN 1: TICK [1]
FUN 1: TICK [2]
FUN 1: TICK [3]
FUN 1: TICK [4]
FUN 1: TICK [5]
FUN 1: TICK [6]
FUN 1: TICK [7]
FUN 1: TICK [8]
FUN 1: TICK [9]
FUN 2 IN ITERATION[4]
Reading block[5] from disk
reading from MASTER
yielding the CPU
FUN 3 IN BURST[8]
FUN 3: TICK [0]
FUN 3: TICK [1]
FUN 3: TICK [2]
FUN 3: TICK [3]
FUN 3: TICK [4]


csce410@COE-VM-CSE1-L09: ~/Desktop/interrupts/MP6_Sou...
<bochs:1> c
Installed exception handler at ISR <0>
Allocating Memory Pool... done
Installed interrupt handler at IRQ <0>
Constructed Scheduler.
Installed interrupt handler at IRQ <14>
Hello World!
CREATING THREAD 1...
esp = <2098148>
done
DONE
CREATING THREAD 2...esp = <2099196>
done
DONE
CREATING THREAD 3...esp = <2100244>
done
DONE
CREATING THREAD 4...esp = <2101292>
done
DONE
STARTING THREAD 1 ...
THREAD: 0
FUN 1 INVOKED!
FUN 1 IN ITERATION[0]


FUN 1: TICK [6]
FUN 1: TICK [7]
FUN 1: TICK [8]
FUN 1: TICK [9]
Writing to block[6] from disk
writing to master
writing to dependent
FUN 3 IN BURST[9]
FUN 3: TICK [0]
FUN 3: TICK [1]
FUN 3: TICK [2]
FUN 3: TICK [3]