MP6: Primitive Disk Device Driver

By: Mehul Varma

Part 1 Introduction

In this project, we developed kernel-level device drivers that performed read and write operations without busy waiting and yielding to the CPU once it issues the operation.

Part 2 Architectural and component-level design

Blocking Disk

In this assignment, we only wrote code for the blocking disk where we derived code from the simple disk where we would just yield before we check if the operation is ready or not. This was done in the *wait_until_ready()*.

Linked List

The other main files that were added were the *linkedlist.C/H* to implement a linked list functionality. We add to the linkedlist before we yield to remember that we can come back to the thread to actually complete the operation.

The following are the functions for the linked list:

- Push : this function pushes to the end of the linked list
- Pop : this function gets the first element and removes it from the list

Scheduler

Some changes were made to the scheduler where the blocked threads were yielded to first the blocked thread and then to the ready threads if there were no blocked threads.

Part 4 Testing

The only testing that was done was running the kernel.C file which did read and write operations using the second thread.