

# Operating Systems Course Assignment 3

Year: 2016/2017

## 1 Linux I/O Scheduling

Now that you have seen how process scheduling algorithms affect your performance in assignment 2, let us now look at a different part of the kernel that can significantly affect your performance, the I/O Scheduler!

Simply sending out requests to the block devices in the order that the kernel issues them, as soon as it issues them, results in poor performance. One of the slowest operations in a modern computer is disk seeks. Each seek—positioning the hard disk’s head at the location of a specific block—takes many milliseconds. Minimizing seeks is absolutely crucial to the system’s performance.

The I/O scheduler divides the resource of disk I/O among the pending block I/O requests in the system. It does this through the merging and sorting of pending requests in the request queue.

An I/O scheduler works by managing a block device’s request queue. It decides the order of requests in the queue and at what time each request is dispatched to the block device. It manages the request queue with the goal of reducing seeks, which results in greater global throughput. The modifier “global” here is important. An I/O scheduler, very openly, is unfair to some requests at the expense of improving the overall performance of the system. I/O schedulers perform two primary actions to minimize seeks: merging and sorting. Merging is the coalescing of two or more requests into one.

## 2 Your task

Write a program to test and compare three of the different Linux I/O schedulers. Your program should run in user space. You should include performance metrics that make sense for your choice of the benchmarked schedulers, and you do not need to confine yourself to the ones you used previously.

For each one of the three policies, describe how you think we can improve the policy and optimize the performance even more.

Your program should be adequate to what you are trying to test and should be capable of running for time sufficient to draw conclusions.

Deliver a well written report, and some beautiful code :). Use the department’s gitlab, your online github repo, or some other code sharing facility to share code with both teachers.

Three different policies is the absolute minimum. Anything more will be a bonus. Considering how the same policy’s configuration affects the scheduling is a bonus too.

### **3 Deadline**

13 January, 2017.