

Group 14 Writeup

Our plan to implement the this algorithm is to sort upon adding an element. So that the queue is always ordered from shortest distance to greatest distance. We are beginning with the NOOP scheduler as a base. Outside of adding a couple of helper functions we also modified some pre-existing functions. In add we sorted based upon sector relative to head position. In the struct data we added a field to track the head position. After we dispatch each time we also sort the queue.

As it currently functions it is the dumbest version of sstf. It really just tries to find the request closest to the head. However, we haven't been in any documented way been tasked with doing anything more complex. So right now our scheduler will function like something somewhat worse than NOOP applied to sequential-access hardware. In fact this scheduler could quite feasibly starve requests. Although, even more "intelligent" versions of this algorithm would suffer this flaw. Hopefully by the time we submit we will have corrected our stutter issue. According to Love, NOOP does merging though I can't see where this occurs at all. As of now we do not have any intention of implementing a merge function.

References

<http://code.google.com/p/cs411g16/source/browse/branches/project4/block/sstf-iosched.c?r=100>
<http://searchcode.com/codesearch/view/24999107>
https://github.com/ryleyherrington/linux_kernel_411/blob/master/sstf-io/sstf-iosched.c