# Project 2 Write Up

Group 22

October 27, 2014

---

## Our Solution

For this project, we started by grepping through the Linux Kernel v3.0.4 for the noop implementation that was mentioned in the assignment. We knew it was in the ../block folder and found the implementation to be in the noop-iosched.c file. Most of the functions in this C file could be kept as they were. The only changes we really made were to the data structure of a SSTF and the dispatch. Noop operates by taking the next request from a queue and then dispatching it, regardless of position. SSTF, or Shortest Seek Time First, behaves different since it selects an I/O request that is closer to its current position. As a result, we need to add two new variables to the SSTF data structure so that the function has positions to compare the values with.

Next, we needed to heavily modify the dispatch function so that it works as a SSTF scheduler. Since a SSTF I/O scheduler processes I/O requests according to relative position of the request to the disk head there are a few more things we needed to account for. While a I/O request could the shortest distance from the disk head, however, if the request location is behind the disk head at the time it could be the slowest seek time since the disk is moving in a constant direction. Our algorithm takes advantage of an elevator that essentially traverses the I/O queue picking up new requests on its way. Once the elevator reaches the end it will flip direction and do the same but in reverse. After we initialize our elevator, we have to make sure that the queue is not empty so that we may have values to compare. The current position is then checked against the head and the direction that it is going. If it is going forward and the request is ahead of the list head, then only those values will be compared until the smallest absolute distance between the two are found. If you are going backwards, then you are only considering and comparing against the values behind the head. After the smallest absolute values is found, that I/O process is dispatched. Afterwards, the head is changed either to the front or bottom of the sector that it made the dispatch on for future dispatches.

## Work Log

| Date | Author | Commit | Summary |
|---|---|---|---|
| Tue Oct 14 15:00 | Sam Quinn | 9402a8ddf13ad69b80eb9bf42c294822aff87b2f | Added the Linux folder for homework #2. |
| Fri Oct 17 13:39 | Bob | 8daeb619e3ebf201b079b3643e83c70c119f8de0 | Coppied the Noop I/) scheduler for a template in creating the new SSTF scheduler. |
| Wed Oct 22 18:42 | Bob | 6fbc561c60d26ea444b063b912ac01068f5fca44 | Implemented sstf_dispatch |
| Sat Oct 25 15:22 | Bob | 3c4249ea31657ed2d5eadf9fa9e8ac52923a38d7 | Fixed case logic in SSTF_dispatch |
| Sat Oct 25 15:55 | Bob | b57104fba0b40f41f79c19d08310b543f8b7f6b9 | Added a hw2 Writeup doc. |
| Sat Oct 25 21:36 | Bob | c918023c4d2f92f087e1cd721bde987b6291fe5a | Added Sam's project 2 writeup file. |
| Sun Oct 26 21:57 | Bob | 4737476b348f7fb7acc3b47daa5328d41d393117 | Update sstf-iosched.c fully working. |
| Mon Oct 27 02:38 | lawrencechau | 0fad02a3acef842d3d4b7ec6d4b1ebca2781cf29 | Update group_22_writeup.tex, Wrote the group writeup |
| Mon Oct 27 11:53 | Bob | 50ea3799a520d50c9ee0f32ff3bf0f7b3b56a180 | Added the patch file with our SSTF I/O Scheduler. |
| Mon Oct 27 11:55 | Bob | e06f0b76c25ebca39a0051d0132ba7c8ad74bf54 | Added the Linux Kernel source as a git ignored file. |
| Mon Oct 27 11:56 | Bob | 43e34595c135bd9258a2b2ceaab1e0a069a41c8b | Added untracked files from previous concurrency problem. |