CS 411 Operating Systems II Project 2

Implement the Shortest Seek Time First(SSTF) I/O scheduler

Li Li

1 What do you think the main point of this assignment is?

This assignment now start to talk about the linux kernel. Last time we learned about process scheduler, this time it is I/O scheduler. A serials of structs and concepts that implement in linux system should be made clear, these include:

- *concepts* This would be like sector, block ,request, SSFT and etc.
- *structs and functions* bio struct, request struct, dispatch function, elevator.
- *some reviews on queues, link list these kinds of stuff* It seems to be the core how the program works.

2 How did you personally approach the problem? Design decisions, algorithm, etc.

In the first place, I thought the scheduler would be just find the nearest request and dispatch it. I knew it would cause problem like a overload system keeping generate I/O request near the driver hand, that the farside request would never appear in the dispatch queue. But I thought it is meant to be and I even write some demo codes to generate these kinds of I/O requests to the system. Than if the system runs into it, I demoed the SSTF. Then, I was correctted by other group members. It should include the the elevator machanism.

Our implementation is based on noop-iosched.c, most of the functions are similar excluding sstf_dispatch() and sstf_add_request(). Basiclly, we start with an increace direction. And when a request is added, we add it into a double linked list. In the dispatch function, we first check if the list is empty, than return. If there is one have sector larger than the prev one, we dispatch it. We go over this procedure and hit he sentinel. After that, turn it aroud and do the similar stuff but to find the smaller sector and dispatch. In this method, it continues until the cpu halts.

3 How did you ensure your solution was correct? Testing details, for instance.

We test the I/O scheduler by add printk() to each step in the sstf_dispatch() and sstf_add_request() and set a [sstf] label to each message. Thus the kernel would have a logfile be written. We could use ”dmesg |grep [sstf]” to see the results.

4 What did you learn?

In the beginning, it is quit hard for me to find useful fuctions, structs in the linux. But now I can find them very quick by either go to https://www.kernel.org/doc/htmldocs/kernel-api/ ,use grep command in bash or C-i in emacs.