Matthew Coppolo

November 24, 2019

COP 4710

Lab 4

Analysis of Functions:

static int clook\_dispatch(struct request\_queue \*q, int force)

and

static void clook\_add\_request(struct request\_queue \*q, struct request \*rq)

These two functions in clook-iosched.c were modified so that the noop-iosched isn’t what is being used. The first function clook\_dispatch is taking data and dispatching them to the queue. Every time data is read or written to, that memory address is printed out to see.

The second function clook\_add\_request is organizing the data in terms of physical address. Once they are ordered they are printed out for the user to see.

Analysis of Elevator:

The C-Look I/O scheduler uses the FCFS disk scheduling algorithms. CLOOK is similar the CSCAN I/O scheduler but has some differences. One difference is that the CLOOK algorithm goes to the last request to be serviced in front of the head and then goes to the other ends last request. This prevents the extra delay from the traversal of the two ends of the disk.

In short the C-Look algorithm makes ‘jumps’ from requests instead of traversing the entire disk which in turn shortens delays. C-Look isn’t necessarily the fastest I/O scheduler. What is the fastest depends on the state of memory at the time.

Process:

The test.sh displays what the IO Scheduler is doing to the user.

Example:

[CLOOK] add W 15432

[CLOOK] add W 15542

[CLOOK] dsp R 17544

[CLOOK] dsp W 17558

[CLOOK] add W 528542

[CLOOK] dsp W 538542

[CLOOK] add R 528584

The numbers at the end represent the physical address of the data being written or read form. At first the addresses will appear to be unorganized, however once C-Look gets the data the physical addresses are organized by FCFS.

[CLOOK] add W 472384

[CLOOK] add W 483547

[CLOOK] dsp W 472384

[CLOOK] dsp W 483547

Here you can see that the order is kept from when they are added to when they are dispatched.