# CS 746 : Linux Kernel Programming

## Course Project Proposal

# “IO scheduler”


**Submitted by**

**Aurobindo Mondal  (143050082)**

**Under the guidance of**

**Prof. Purushottam Kulkarni**




**Department of Computer Science and Engineering**
**IIT Bombay**


**April 29March 20, 2016**

## Objective :

Design and implementation of a new block IO scheduler. I have implemented a Multilevel Priority Queue scheduler.

## Approach :

Linux 4.3.3 has 3 block IO schedulers - noop, deadline and CFQ.

Noop is a basic IO scheduler which is simply FIFO with minor merging. This is the simplest IO scheduler in Linux Kernel.

The deadline IO scheduler uses separate queues for reads and writes and adds a deadline to every requests to prevent starvation. Reads and writes have different preferences.

The CFQ(Completely Fair Scheduling) is an IO scheduler having queues per process and allocates time slices to each queue. The time slice and the number of requests allowed to depends on the IO priority of the process.

I have modified the noop scheduler and tried to make a Multilevel Priority Queue Scheduler. The basic building block of the code remains the same as the noop scheduler. Only some of the functions that deals in adding and dispatching requests needs to be modified according to our design.

## Design:

The basic idea of the Multilevel Priority Scheduler is to prioritize the IO requests based on the processes. The basic idea behind the thought is to identify processes which does more IO and put them in high priority queue.

The IO scheduler consists of 2 main parts :

- ➢ The Multilevel Priority Queues

➢ The Dispatcher

In my model, I have implemented 3 Queues (the 1st queue having the highest priority). Queue 1 and 2 are reserved for processes (which can be given as input) and Queue 3 acts as a default queue for all.

Whenever a process in priority 1 queue does an IO request, the next dispatch event would be from the priority 1 queue.

## Implementation Details:

An High level algorithm for the IO scheduler can be explained as :

1. INIT_FUNCTION :
    a. Created a kobject for sysfs entries. These sysfs entries are used by the users to communicate with the kernel about making the decision of which process should own the priority 1 and 2 queues.
    b. Register the mpq IO scheduler in the Kernel.
2. The structure *struct elevator_type* contains all the function pointers that needs to be overridden. The new functions are defined in the module. The functions that are overridden are :
    a. elevator_merge_req_fn
    *b. elevator_dispatch_fn*
    *c. elevator_add_req_fn*
    d. elevator_former_req_fn
    e. elevator_latter_req_fn
    f. elevator_init_fn
    g. Elevator_exit_fn
3. A structure *struct mpq_data* is made which contains an array of queues of size 3. These are the queues which stores the IO requests based on the processes.
4. INIT_FN:
    a. Initialise the structure that contains the queues.
    b. Initialise all the queues using *INIT_LIST_HEAD().*
5. ADD_REQUEST :
    a. Get *task_struct* of the current process. Check its PID and match to that of the *sysfs* entries to decide which queue to put it in.

b. Add the requests to the end of the particular queue that it belong to.

6. DISPATCH_REQUEST:
    a. Check the 1st Queue :
        i. If it is not empty, dispatch request from this queue.
    b. Else goto the next queue and so on.
    c. Obviously, if the first 2 queues are empty, the IO scheduler behaves like a single queue FIFO scheduler.

## Experiment :

I have written a c programs that reads infinitely from the disk.
I used the flag *O_DIRECT* in the open system call to open the file in order to ensure that the reading the file is always done from the disk bypassing the cache.

The steps involved in the experiment is listed below :

- ➢ Insert the module into kernel using the following command :

  $ make

  $ sudo insmod multiQ-iosched.ko
- ➢ After executing the read program, I add the PID of the read program to the 1st queue using the command :

  $ echo *'PID'* > /sys/kernel/multiQ/q1
- ➢ I change the default scheduler to Multilevel priority queue using the following command :

  $ echo mpq > /sys/block/sda/queue/scheduler
- ➢ Previously the IO scheduler was serving request from queue 3. As the new process ID is put in the sysfs entry, the IO scheduler starts reading from Queue1, because Queue1 has more priority than Q3.
- ➢ I identified 4 cases :
    - ○ Q1 & Q2 has processes
    - ○ Only Q1 has processes
    - ○ Only Q2 has processes
    - ○ Both Q1 & Q2 are empty

- ➢ In all these 4 cases, the module prints a log of which request is served from which queue. We can see the results to analyse whether the Multilevel Priority queue scheduler work.
- ➢ To remove the module, we first change the IO scheduler back to cfq, deadline or noop using

  $ echo *'sched_policy'* > /sys/block/sda/queue/scheduler

  Then, we can *rmmod* to remove the module.

## Results :

Queue 1 and Queue 2 both has processes :

[610562.520231] io scheduler mpq registered
[611185.943276] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611185.951607] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611185.959948] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611185.968241] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611185.976669] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611185.984897] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611185.993257] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611186.001561] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611186.009904] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611186.018204] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611186.026648] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611186.034906] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611186.043207] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611186.051524] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611186.059967] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611186.068301] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611186.076573] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0

Only Queue 1 has processes :

[611362.097134] Dispatched from Queue 3 for general process: Time required 0
[611362.105518] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.113826] Dispatched from Queue 3 for general process: Time required 0
[611362.122090] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.130423] Dispatched from Queue 3 for general process: Time required 0
[611362.138802] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.147095] Dispatched from Queue 3 for general process: Time required 0
[611362.155407] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.163793] Dispatched from Queue 3 for general process: Time required 0
[611362.172110] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.180381] Dispatched from Queue 3 for general process: Time required 0
[611362.188759] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.197066] Dispatched from Queue 3 for general process: Time required 0
[611362.205410] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.213708] Dispatched from Queue 3 for general process: Time required 0
[611362.222148] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0

[611362.230360] Dispatched from Queue 3 for general process: Time required 0
[611362.238704] Dispatched from Queue 1 with ProcessID : 9474 : Time required 0
[611362.247005] Dispatched from Queue 3 for general process: Time required 0

## Only Queue 2 has processes :

[611473.412460] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.412494] Dispatched from Queue 3 for general process: Time required 0
[611473.412596] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.412637] Dispatched from Queue 3 for general process: Time required 0
[611473.412770] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.412815] Dispatched from Queue 3 for general process: Time required 0
[611473.412894] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.412932] Dispatched from Queue 3 for general process: Time required 0
[611473.413037] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413067] Dispatched from Queue 3 for general process: Time required 0
[611473.413145] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413181] Dispatched from Queue 3 for general process: Time required 0
[611473.413263] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413303] Dispatched from Queue 3 for general process: Time required 0
[611473.413385] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413419] Dispatched from Queue 3 for general process: Time required 0
[611473.413504] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413538] Dispatched from Queue 3 for general process: Time required 0
[611473.413622] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413658] Dispatched from Queue 3 for general process: Time required 0
[611473.413744] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413780] Dispatched from Queue 3 for general process: Time required 0
[611473.413862] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.413910] Dispatched from Queue 3 for general process: Time required 0
[611473.413988] Dispatched from Queue 2 with ProcessID : 9475 : Time required 0
[611473.414025] Dispatched from Queue 3 for general process: Time required 0

## Both Queue1 & Queue 2 are empty :

[611584.154897] Dispatched from Queue 3 for general process: Time required 0
[611584.163192] Dispatched from Queue 3 for general process: Time required 0
[611584.171487] Dispatched from Queue 3 for general process: Time required 0
[611584.179821] Dispatched from Queue 3 for general process: Time required 0
[611584.188175] Dispatched from Queue 3 for general process: Time required 0
[611584.196525] Dispatched from Queue 3 for general process: Time required 0
[611584.204810] Dispatched from Queue 3 for general process: Time required 0
[611584.213136] Dispatched from Queue 3 for general process: Time required 0
[611584.221515] Dispatched from Queue 3 for general process: Time required 0
[611584.229815] Dispatched from Queue 3 for general process: Time required 0
[611584.238222] Dispatched from Queue 3 for general process: Time required 0
[611584.246418] Dispatched from Queue 3 for general process: Time required 0
[611584.254810] Dispatched from Queue 3 for general process: Time required 0
[611584.263063] Dispatched from Queue 3 for general process: Time required 0
[611584.271442] Dispatched from Queue 3 for general process: Time required 0
[611584.278487] Dispatched from Queue 3 for general process: Time required 0
[611584.279742] Dispatched from Queue 3 for general process: Time required 0

[611584.280023] Dispatched from Queue 3 for general process: Time required 0
[611584.280140] Dispatched from Queue 3 for general process: Time required 0
[611584.280237] Dispatched from Queue 3 for general process: Time required 0
[611584.280335] Dispatched from Queue 3 for general process: Time required 0
[611584.288174] Dispatched from Queue 3 for general process: Time required 0
[611584.296391] Dispatched from Queue 3 for general process: Time required 0
[611584.304743] Dispatched from Queue 3 for general process: Time required 0
[611665.681762] GoodBye!!!!

## Analysis :

Based on the results, it is clear that the IO scheduler selects from the high priority queues if there are requests in them. When there are no request in high priority queues, it serves the low priority queues.
If the high priority queues are empty, it serves the 3rd queue which is a default for all processes.

## Drawbacks and Future Plans :

The IO scheduler suffers from starvation. Consider a scenario when the 1st queue is always full i.e. the highest priority process continuously does IO. In this case, the requests waiting in the lower priority queues can starve.

One solution to this problem could be addition of timer to the process as in deadline scheduler and as the request meets its deadline increment it by one queue.