

MLFQ

So, my MLFQ has three major components (Classes). First, the scheduler knows the level of the queues and which one has the highest priority. Second, the scheduler also needs to know what time it is right now, and it also needs to determine the next job by applying the 5 rules. Third, the scheduler must also reduce the priority based on the job's allotment. Finally, it also needs to track the next boost time. It also has the method to track how many jobs are left in all the queues. Notice the scheduler can hold as many queues as it needs. We will use the array to store the queues; the array's index means the queue's level.

Next, we have a queue which is a linked list. Each job as a node will store in the queue. The reason to use the linked list is when we do the round-robin operations makes it more accessible. And we enqueue as many jobs as we need without worrying about the size of the before head. When doing boost operations, the only thing to do is to change the head pointer to the highest queue, which is much more efficient.

Our next component is the base component job. The job component stores the process id, number of allotments left in the current level, how many time slices, the run time, and I/O time.

That said, we can use the above design to measure the response time by whether we use a boost or not. And we also measure the turnaround by tackling the time slice and making the round robin never happen or happen frequently. We can also adjust the I/O time the measure the turnaround time. Please check the following page UML Diagram for detail.

