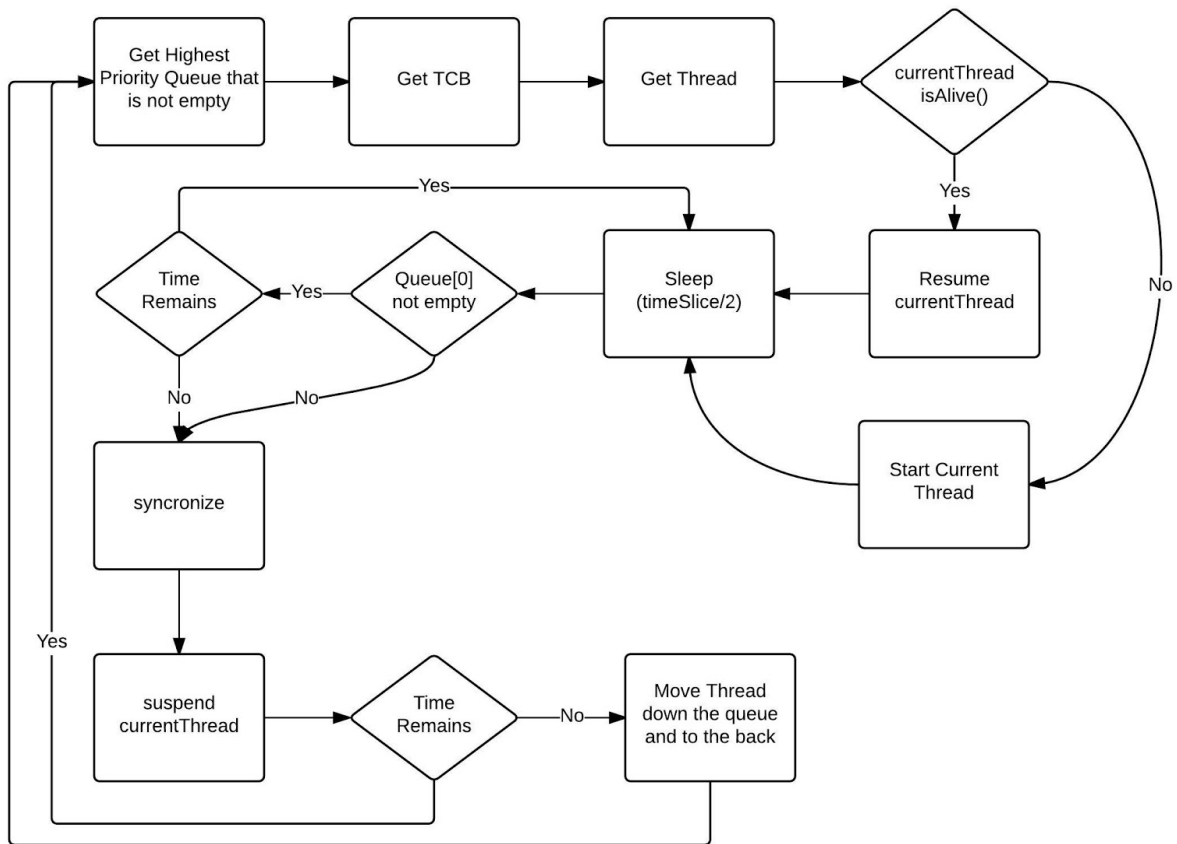


Algorithm



Compare RR and Multi-queue

Part 1

threadOS: a new thread (thread=Thread[Thread-5,5,main] tid=1 pid=0)
threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[e]: response time = 5999 turnaround time = 6500 execution time = 501
Thread[b]: response time = 2997 turnaround time = 9999 execution time = 7002
Thread[c]: response time = 3998 turnaround time = 21005 execution time = 17007
Thread[a]: response time = 1998 turnaround time = 29007 execution time = 27009
Thread[d]: response time = 4999 turnaround time = 33007 execution time = 28008

Part 2

```
threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[b]: response time = 991 turnaround time = 5493 execution time = 4502
Thread[e]: response time = 2491 turnaround time = 7992 execution time = 5501
Exception in thread "Thread-0" java.lang.IllegalThreadStateException
    at java.lang.Thread.start(Thread.java:705)
    at Scheduler.run(Scheduler.java:211)
```

Whilst Part2 didn't run as I expected it to, I can see how it is running. I foresee the Part2 taking longer to run the the RR because the programs very quickly get pushed down to the bottom queue and are only ran if the top queue does not get any new jobs. This would be a problem if many jobs are set to the scheduler, basically the bottom jobs have the possibility to never run if the scheduler continuously gets new jobs. because the RR guarantees the Threads to get a fair equal share of the processor, the RR should run faster.

RR Queue2

I assume we are talking about making the queue2 a RR and not checking queue0 until the queue2 is Empty. again I see the tread loop getting stuck in queue2 if you get a large program, slowing down the higher jobs. I think what would be better is if the queue2 is ran by a different scheduler and it ran a RR while the multi queue works on queue0 and queue1 to kick out all the small programs and sifting all the larger jobs to the RR queue2.