**Analysis:**

Below are the average turnaround times for the schedulers

|  |  |  |
| --- | --- | --- |
|  | RR | MLFQ |
| Input1 | 102 | 100 |
| Input2 | 172 | 167 |
| Input3 | 222 | 195 |
| Input4 | 290 | 267 |
| Input5 | 137 | 130 |

**Graph showing the average turnaround time for RR and MLFQ schedulers :**

**Conclusion:**

The fundamental problem MLFQ tries to address is two-fold. First, it would like to optimize turnaround time, which may be obtained by running shorter jobs first; unfortunately, the OS doesn’t generally know how long a job will run for, exactly the knowledge that algorithms like SJF (or STCF) require. Second, MLFQ would like to make a system feel responsive to interactive users (i.e., users sitting and staring at the screen, waiting for a process to finish), and thus minimize response time; Round Robin algorithm would reduce response time but are terrible for turnaround time. The above graph shows that the average turnaround time is always better in MLFQ than in RR.

**Citations:** <http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched-mlfq.pdf>

<http://stackoverflow.com/questions/17474058/understanding-multilevel-feedback-queue-scheduling>

<https://www.youtube.com/watch?v=nr28T7dp-50>