Operating Systems Coursework 1 - Advanced Task

Starvation and inflexibility are the main disadvantages to Multilevel Queue scheduling. As the MQ scheduler executes in order of highest priority to lowest, there is often starvation of lower priority processes. Though they are low priority, starvation should still very much be avoided in an effective scheduler in order to ensure a functional, fair system that does not impede performance. Starvation also leads to a longer wait time for processes to execute, increasing the average wait time of the scheduler.

As processes are added to the runqueue they are assigned to a queue based on priority. From here, the system is static and inflexible; processes do not move from their respective queues. This allows a low scheduling overhead, but comes with the trade-off of inflexibility. The proposed solution to both starvation and inflexibility is the introduction of movement through queues. This movement increases scheduling overhead, but greatly improves flexibility, which in turn aids starvation. This is done through the implementation of Multilevel Feedback Queue Scheduling.

Three queues are implemented, where queues 1 and 2 both use Round-Robin scheduling (with quantums of 8 and 16 time units respectively), and queue 3 uses first come first served (FCFS) scheduling. Entities are first assigned to q1. Entities in the highest queue are executed first; queue 1 executes first, followed by queue 2 when queue 1 is empty, and finally queue 3 when the other two queues are empty. Now, the flexibility comes from entity movement throughout these queues. When an entity in queue 1 executes for a quantum of 8 time units, if not fully executed, the entity will be moved to queue 2. Similarly if an entity executes for longer than queue 2's quantum of 16 time units, it will be moved to queue 3. This movement through queues prevents entities from hogging the processor, and reduces the bias toward high priority processes. In MLFQ scheduling, all processes are treated equally, rather than beginning with high priority processes.

This has been implemented in infOS in the file "sched_adv.cpp", and can be executed with: ./build-and-run.sh sched.algorithm=adv

Sources

Article title: Multilevel Queue Scheduling Algorithm | Studytonight
Website title: Studytonight.com
URL: https://www.studytonight.com/operating-system/multilevel-queue-scheduling

Article title: Multilevel Feedback Queue Scheduling Algorithm | Studytonight
Website title: Studytonight.com
URL: https://www.studytonight.com/operating-system/multilevel-feedback-queue-scheduling