In this project, Dylan and I worked to implement four thread schedulers for a uni-processor environment: First Come First Served (FCFS) - no preemption, Shortest Remaining Time First (SRTF) - with preemption, Priority Based Scheduling (PBS) - with preemption, and Multi Level Feedback Queue (MLFQ) - with preemption and ageing. The FCFS scheduling scheme is non-preemption based and schedules the threads based on who arrives first at the scheduler. The SRTF scheduling scheme is preemption based and schedules the threads based on who has the least 'remainingTime' value. When two or more threads have the same 'remainingTime' value, then the thread with least 'currentTime' is chosen for scheduling. The PBS scheduling scheme is preemption based and schedules the threads based on who has the highest priority. If two or more threads have the same priority value, then the thread with least 'currentTime' is chosen for scheduling. The MLFQ scheduling scheme is preemption based with multi-level feedback queues. The READY queue is a multi-level queue with 5 levels. The time quantum for the 5 levels (starting from the highest level) is 5, 10, 15, 20, and 25 time units respectively. Threads initially join the READY queue at the highest level. When a thread completes its time quantum on a particular level, it is moved to the next lower level. The scheduler selects threads from a lower level queue only when there are no threads in any of the higher levels. A thread entering the READY queue at the highest level will preempt threads at the lower levels at the next integral time unit. Within a level, we use FCFS to select the threads. We used round-robin at the last level.

Contributions

Dylan Nguyen: FCFS, MLFQ, synchronizing global time

Jacob Klapper: PBS, SRTF, queue inserts