

Assignment 2 - Process Scheduling

Axel Steingrimsson - axels16@ru.is

First Come First Served (FCFS)

I created one controller function for each scheduling algorithm which takes the parameter "adding" to determine whether the process is being added or finishing. The algorithm for FCFS was very simple. If the processes queue is empty the process being added is switched to and added to the queue, otherwise it's just added to the queue. If the current process is finishing it is removed from the queue and the front of the queue is switched to.

Round Robin (RR)

Similar implementation to FCFS except a time slice is created whenever a process is switched to. This is done by creating a new thread to wait for the quantum amount of time and then if it is not interrupted before then it switches to the next process in the queue. If a time slice is created but one already exists, it interrupts the exists slice.

Shortest Process Next (SPN)

Simple implementation, whenever a process is added it is inserted into the queue so that it is always sorted from shortest to longest, not including the current process because it's not preemptive.

Shortest Remaining Time (SRT)

Similar implementation to SPN, except the insertion is determined by shortest remaining time. Because SRT is preemptive I create a new thread the exists while the queue is not empty, which I call monitor, that constantly checks if the current process has a shorter remaining time than the second element in the queue, if it isn't then the current process is removed and reinserted into the queue at the appropriate place. That way the queue is always ordered by shortest remaining time, and the only work is having the monitor check that the current is the shortest.

Highest Response Ratio Next (HRRN)

Similar to SPN and SRT, when a process is added it's inserted based on HRRN. Then after a process finishes I sort using a custom comparator that reorders them based on HRRN. That way it's always sorted when a new process is chosen the current is removed, and then process with the next highest response ratio is then the first element in the queue.

Feedback (FB)

I initialize a map with priority levels as the keys, and a list of processID's as the value. Then I add processes and create the time slice similar to Round Robin, always starting with the highest priority, but when a process is switched I move it to one lower, but never lower than 7. That way when I'm looking for the next process to switch to I just take the first element from the first priority level that has elements. I was facing issues with the time slice changing the current process level at the same time as the main thread so I used a lock so that only one thread could be switching a process at any given time.