PINWEN MU

CS526 Data Structures and Algorithms

Term Project

Discussion:

The data structure I used for "D" is a sorted priority queue. “D” is designed to store process data read from the input file, and sort them by their arrival time. This sorted priority queue incorporates a Comparator interface to organize each process according to its arrival time. The "min" method returns the process with the smallest arrival time, and it will be removed by the "removeMin" method when its arrival time is less than or equal to the current time.

If processes have equal priority, it might be more efficient to execute the one with an earlier arrival time, rather than choosing arbitrarily. On a higher level, I would add another comparator parameter to the HeapAdaptablePriorityQueue. This second comparator would be used when the first one returns equal values, setting priority as the first comparator and arrival time as the second.

To enhance my project's readability and reusability, I think I can write a class named "process." This class could store the data for each process, including the id, priority, duration, arrival time, waiting time, and the remaining time after execution.