

Rate Monotonic Scheduling :

The design includes a class which would contain all the information regarding a process. Two vectors of this class is made to store all the processes, which are read from the input file, of which one is to store it permanently which the second one is used as a modifiable version in the later part of the code.

After getting the inputs we run a function named scheduler() which in essence runs from starting time 0 to the maximum time(which would be maximum of the product $k * \text{period}$), and saves the process id for each unit time into an array.

In the function scheduler() we call another function calculate_priority(), which returns the highest priority process id. Which in this case is calculated with the one having least period and is still in the waiting queue.

In this scheduler() function we also calculate the total waiting time by all the processes and also the number of times each process missed their respective deadline.

Lastly we print all the related information into the log file and the stat file.

Earliest Deadline First Scheduling :

The design of this is essentially the same as that of the RMS but the difference lies in the function which calculate the priority : calculate_priority().

In this function instead of returning those process ids with the shortest period it returns the process id which has the earliest deadline. This is achieved by introducing a new variable deadline in the class which is modified whenever the period of the process is completed.

Comparison between RMS and EDFS :

