Operating Systems - 2
Programming Assignment - 3 - Implementing Rate-Monotonic Scheduling & Earliest Deadline First Scheduling through Discrete Event Simulation

Vanga Aravind Shounik - CS20BTECH11055

Question :

implement a program to simulate the Rate-Monotonic & Earliest Deadline First (EDF) scheduling algorithms. Then compare the average waiting time and deadlines missed of both algorithms. Implement both the algorithms in C++ using discrete event simulation. Develop a program to simulate both the algorithms. Your program will read the input from the file and generate the correct sequence of events (for each process execution), occurring in the system.

Input:

The input to the program will be a file, named inp-params.txt, consisting of the following parameters: n, the number of processes; followed by n lines consisting of Pi (process id), t (processing time), p (period) and k (the number of times each process repeats).

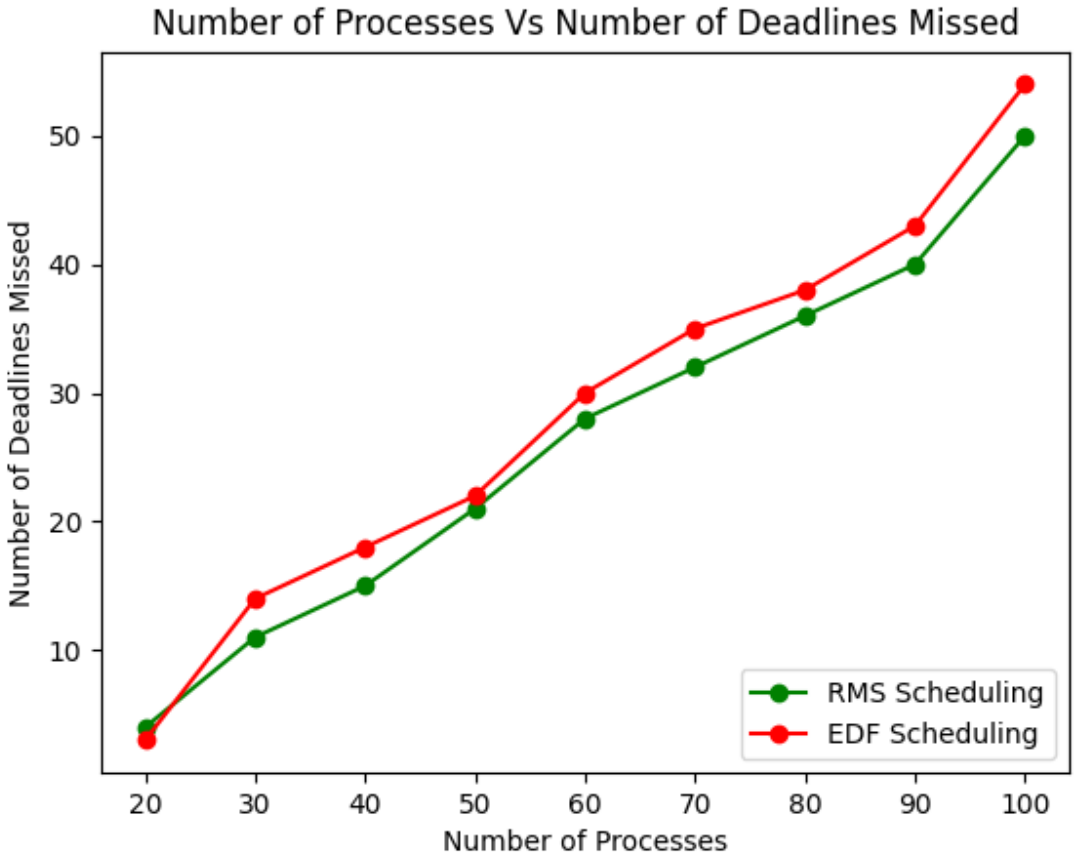
Implementation:

Here, initially, we take the input from a text file named inp-params.txt and then we store them in a struct Process which contains the information of the process like pid, t (time taken), k (number of times the process repeats), remaining(the remaining time for the process to stop executing), then a waitingqueue is created which is a priority queue which has all the processes and because it is a priority queue, it sets the processes in the order to be executed directly using a comparator which we define in the code which compares p's and puts the process in its position. And in the comparator of edf, it checks the deadline. Now a while loop runs until all the processes are executed.

In the while loop, if the waitingqueue is empty, then we directly check if any processes are added to the queue at that instant, if any processes are added, then we print a log (CPU is idle till this time) and then we start the next process and we print the log (Process started executing). If no process is added, we don't have to add anything to the log because nothing changes. If the top process of the waiting queue or the executing process right now has remaining time as 1, then the process should end it's execution in this second and print the log file (the process has completed its execution) and then we should check if any new processes are added to the waitingqueue and add

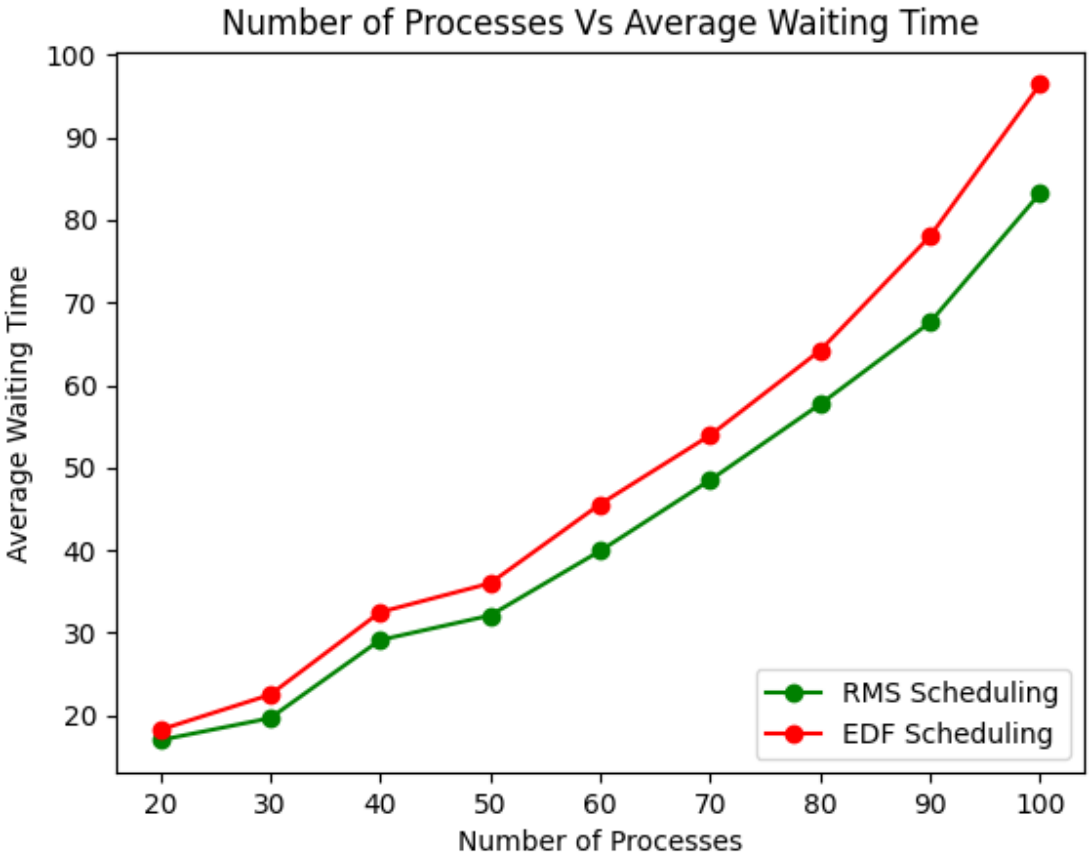
them and then we have to start the topmost process which is the process with the highest priority. If the waiting queue has a process which is not exited at this point of time, then, we have to check if the process is terminated at this point or not, if it is terminated at this point, then we have to print the log that it is terminated and then start the next process, if it is not terminated, then we have to add all the new processes which add to the waitingqueue at this time and then if any process with higher priority comes, then the running process is preempted by the other process which has higher priority and then that process starts running and then we have to write the log to the file.

Graph 1:



Here, this is the graph of Number of total processes vs number of deadlines missed graph. Here, we can see that many processes miss the deadline in EDF rather than the number of processes which missed the deadline in RMS which is less.

Graph 2:



Here, we can see the graph of Number of Processes Vs Average Waiting Time graph. Here, we can see that the average waiting time of EDF processes is higher than the average waiting time of RM scheduling.