OPERATING SYSTEMS 2

INSTRUCTOR: DR. SATHYA PERI
Hitesh Sehrawat - ma17btech11004@iith.ac.in

Introduction

Discrete Event Simulation(DES) to simulate the schedule dispatcher. The scheduler does real-time scheduling with two different algorithms i.e. Rate Monotonic (RM) and Earliest Deadline First (EDF) .

Theory

RMS : Rate monotonic scheduling algorithm schedules periodic tasks using a static priority policy with preemption. If a lower-priority process is running and a higher priority process becomes available to run, it will preempt the lower-priority process. The priority is inversely proportional to their periods. The rationale rms scheduling algorithms is that to assign a higher priority tasks that require the CPU more often.

EDF : Earliest deadline first scheduling assigns priority dynamically according to deadline. The earlier the deadline, the higher the priority; the later the deadline , the lower the priority. Under EDF policy, when a process becomes runnable, it must its deadline requirement to the system.

Implementation

In the program the code takes input parameters from the file “input.txt” and the parameters are process ID , period , CPU burst time , deadline , number of repetitions for each process. We then make a ready queue which contains all the processes which are waiting for their turn for the CPU.

Also parameters such as queue_status (returns whether process is in ready queue or not) , deadline of process, time left for finishing are defined.

Estimation of Parameters

1. Deadline Miss : A process misses a deadline if the deadline expires before completion of execution of the process.

2. Waiting Time : The time the process spends in ready queue not executing on CPU.

3. Turnaround Time : The total time elapsed between completion of a process and its arrival.

Working of programs

We define priority of process in RM as one with least period whereas in EDF it is defined as the one with least deadline.

I have declared a variable of type Process "current" . The "current" has a attribute .on_cpu which contains whether any process is running on CPU at that time.

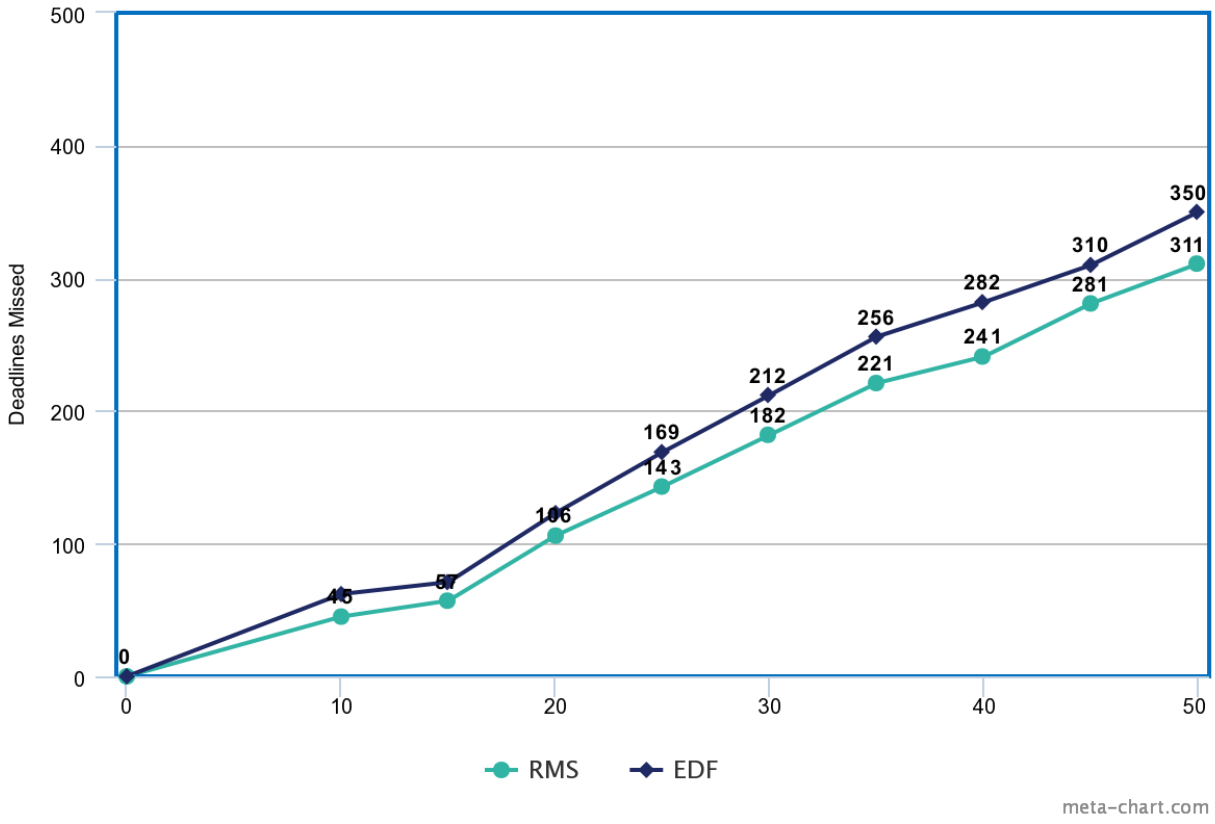
In DES, at each second we check the arrival of processes. At any given time t , a process enters the ready queue if t%(P[i].priority) = 0 , where P[i] is i th process. If such a process is already in queue , then it misses the deadline and we kill the process and increase it's deadline by its period. If "current" ==P[i] then we kill that process too and hence no process will be on CPU at that time. Now in such a situation if the ready queue is not empty we pick the element with highest priority from the queue and put it on CPU. Else , if current.on_cpu is false and ready queue is empty then CPU will be idle until another process comes into queue at its periodic interval.We compare the priority of "current" with that of the highest prior element present in the queue , if it is not empty then preempt the running process if it is priority is more. Now we decrement the time_left of the "current" process at each iteration until it is greater than zero. If it is zero , then we mark that process as completed and increase the deadline of

this process by its period and mark the .on_cpu member of “current” as false. We use a vector to implement a priority queue and store the addresses of processes in the queue such that change in the Process object would reflect in the queue. Finally we find the average waiting time by incrementing it for a process at each step if it is in queue and not on CPU. Average Turnaround time would be simply calculated as subtracting the time between end of a process , either successful or killed.

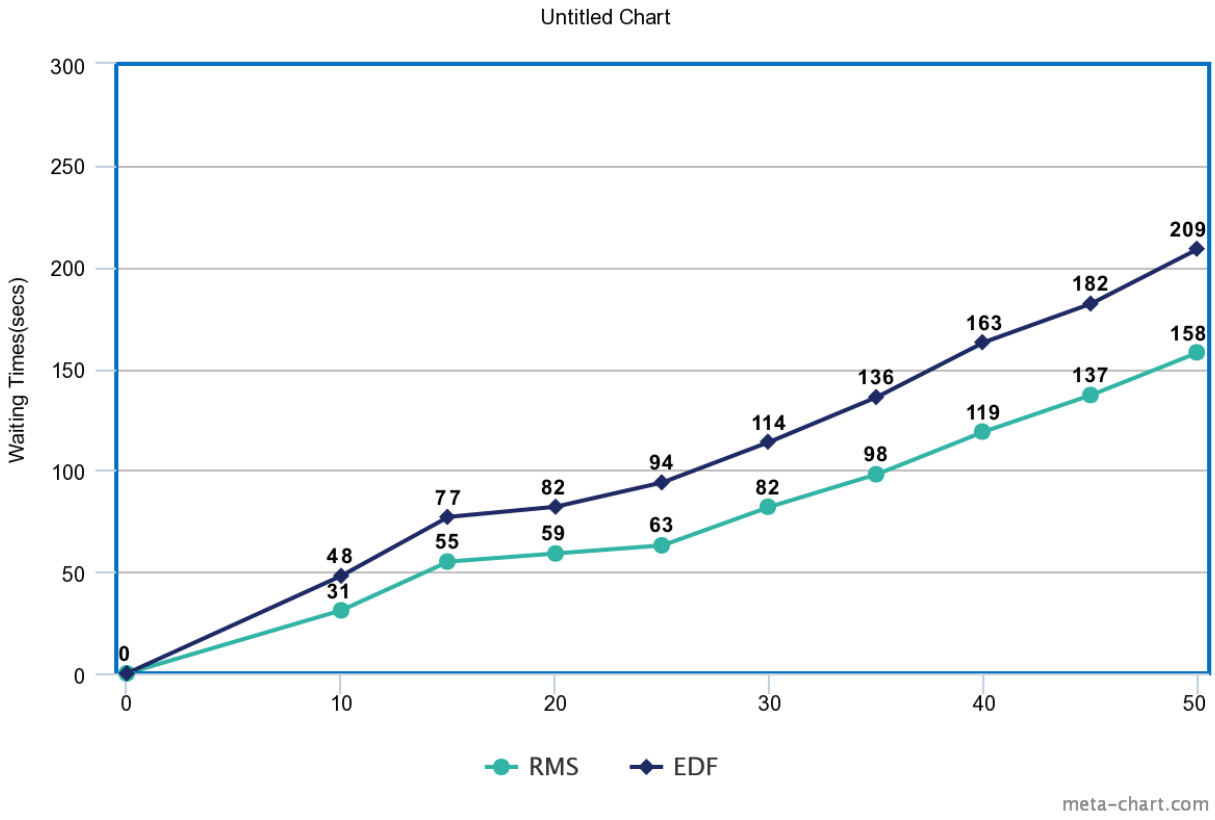
Analysis of both algorithms

Graphs

Graph 1 : No. of Deadlines Missed Vs No. of Process



Graph 2 : Average Waiting Time vs Number of Process



Conclusion

From the graph that as N approach very large value , waiting time rise a little with N . And RMS shows better performance than EDF , both in terms of waiting time , as well as fewer processes with exceeded deadline.