OS-2 Programming Assignment-3 Report

Input:

This programs takes the input as “inp-params.txt” in which
The first line has the number of process
Then the next lines with each line having process id , processing time , period , the no of times to be executed.

Code Design:

1) Created a struct pcb for storing all the information about each process. It has pid , pt(processing time) , pp(period) , k(no of times to be executed) , the next available time, remaing time , status whether it is can or it is not available at this time, a vector waiting which stores waiting time when it runs every time.

2) we define a pointer of type pcb which is dynamically allocated in main, this is used for storing information about all the processes.

3) Here we define a function called highest priority which takes array as input and gives the index of process with highest priority among the process which are available to run.
The only difference between RMS and EDF code is that
In RMS code , while finding the index we consider the process with low period as high priority process.
In EDF code , while finding the imdex we consider the process with low deadline as high priority process.

4)we aslo define a function called low_avail_time which takes array as input and gives the index of process with low available time.

5) In main function , all the inputs are taken and stored and remaining values are also initialised. Now all the procesess status is true. And 2 new file pointers are created for getting log and stats of algorithms.we also define 2 variables for calculating the no of processes completed and no of process missed deadline.

6) here the cputime is 0 , and highest priority index is found.and now the for loop starts with cputime as parameter.

7)In that first it checks whether any function is available for execution or not using the available time variable.If no process is available i.e all the available time of the process is INT_MAX, then its prints all processes completed.

8) Now , we have to schedule the processwhich has to be executed.we will check if available time is less than cputime we make status of those process true.

9)Now if cpu index is less than zero we have 2 cases. The cpu can idle or a new process is about to run.if none of the status is true, then its a case of CPU sleep.then we will skip to the time which is calculated using low_available time function and update cpuindex.or if some processes have status true then its a case of new process.

10) else if cpu index is equal to highest priority index it will continue else the present process is preempted and cpu index is updated to highest priority index.now we have found which process has to run.

11) Now,If the process deadline is more than cpu time then remaining time for the process is lowered and updated else it means that it have missed the deadline.It is terminated , no of processes that missed deadline count is increased by 1 and the no of time it should be executed is reduced by 1.Now if remaining no of time it should run is non-zero then the new available time , new deadline , new remaining time is updated and status is updated to false, cpu_indexis made -1.if the remaining no of time is zero then process is completely terminated now the deadline , available time is made INT_MAX , cpu index is made -1.

12)Now, the remaining time is zero , now no of times to be run is lowered by 1 i.e, the current process is completed one iteration , no of procceses completed count is increases by 1 and Now if remaining no of time it should run is non-zero then the new available time , new deadline , new remaining time is updated and status is updated to false, cpu_indexis made -1.if the remaining no of time is zero then process is completely terminated now the deadline , available time is made INT_MAX , cpu index is made -1.

13) This type of loops continues and the cpu time is updated.

14) In the loop when the procceses misses the deadline and also when the process has completed on iteration the waiting time is updated.

15)Using a for loop and also itearting over the vector waiting time the average time for each process is calculated. And also the overall waiting time for all processes is calculated and we have to free the dynamically allocated pointers.

Output:

It will generate 4 files (each algorithm 2 files). The txt files log.txt and stats.txt.

Graphs:

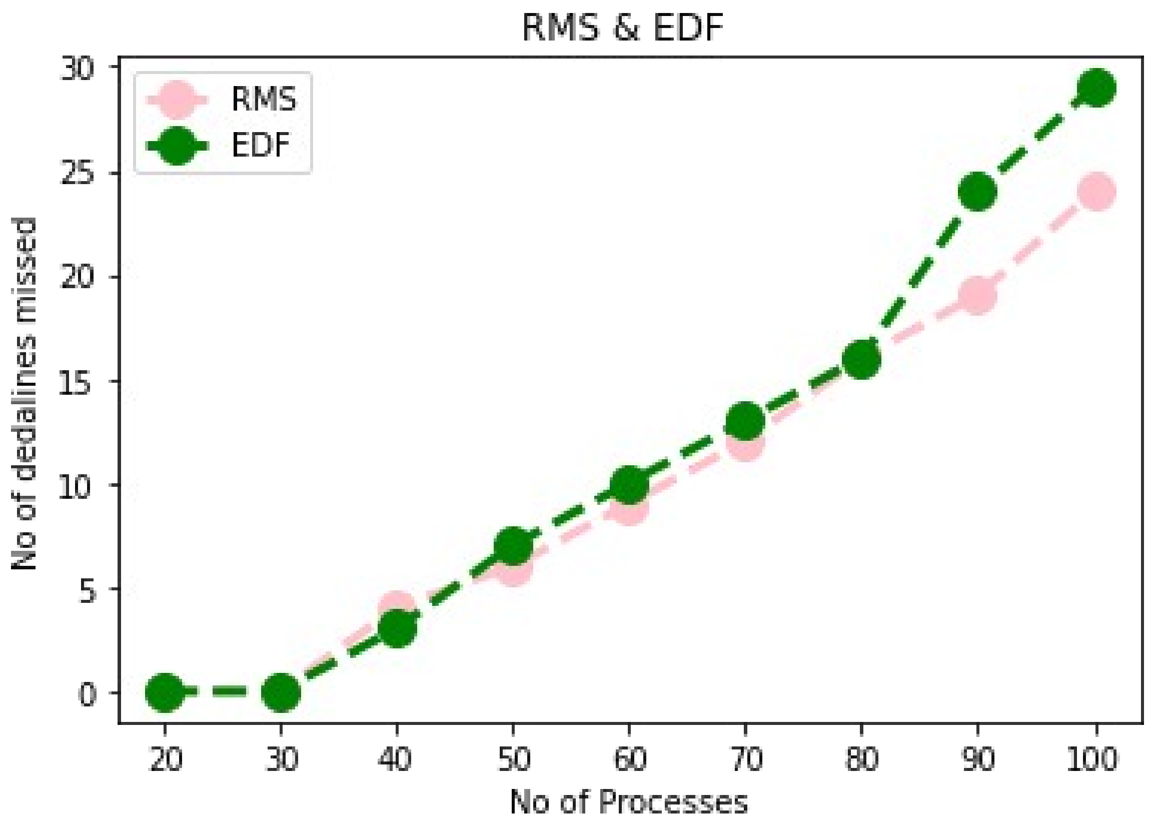

Fig1

Fig1 shows the plot of no of processes that missed deadline vs total no of processes for both RMS and EDF algorithm.

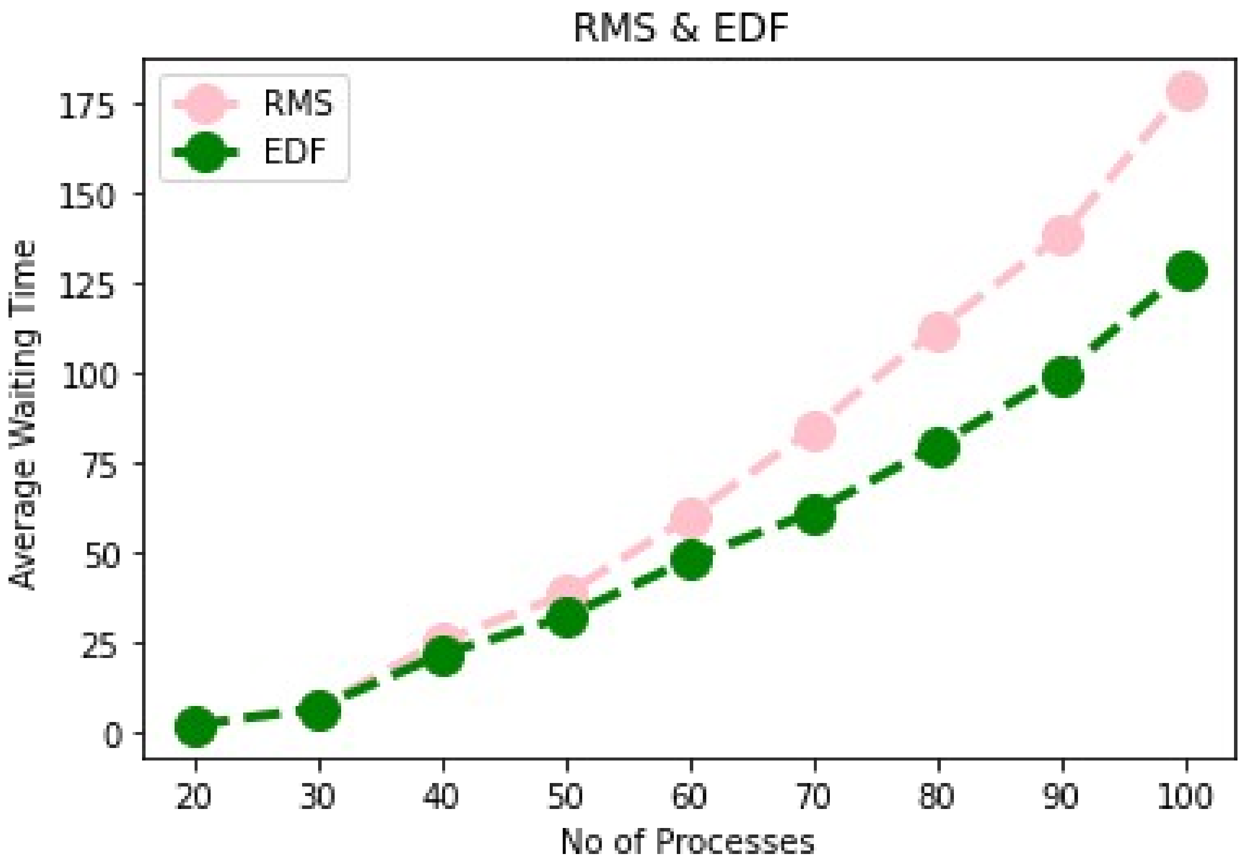

Fig2

Fig2 show the plot of Average waiting time vs total no of processes for both RMS and EDF Algorithm.