OPERATING SYSTEMS-II (CS3523)

REPORT

1) **Design of Rate Montonic Scheduling Algorithm/Earliest Deadline First Scheduling Algorithm**: In the program two classes have been used. First class is used for storing the data about the processes. Second class is used for simulating the scheduling algorithm and creating output files.

For scheduling a priority queue(ready_queue) has been used. Priority is decided based on period of the process. Priority is inversely proportional to period.

While taking the input, maximum time required for complete execution of all process is calculated. It is calculated as max_time=max(period*number_of_processes) or max(k*p).

Events considered are when a process:

a) starts its execution.
b) finishes its execution.
c) is preempted by another process.
d) resumes its execution.

The algorithm is simulated by running a loop from 0 to max_time. At each time unit a process is checked whether a process should enter the ready queue, be terminated for missed deadline, be eliminated from ready queue, has completed execution, be preempted by another process. Appropriate loops and conditions have been placed for these purposes.

For calculating the total number of missed processes a variable missed is initialised to zero. At each time all the process that misses the deadline are eliminated from the queue and missed is incremented by one.

Total number of processes is calculated by summing all k's of processes(ksum). Total processes that completed execution=ksum-missed.

Average waiting time of algorithm=(sum of waiting of all processes)/ksum
Average turnaround time of algorithm=(sum of turnaround of all processes)/ksum

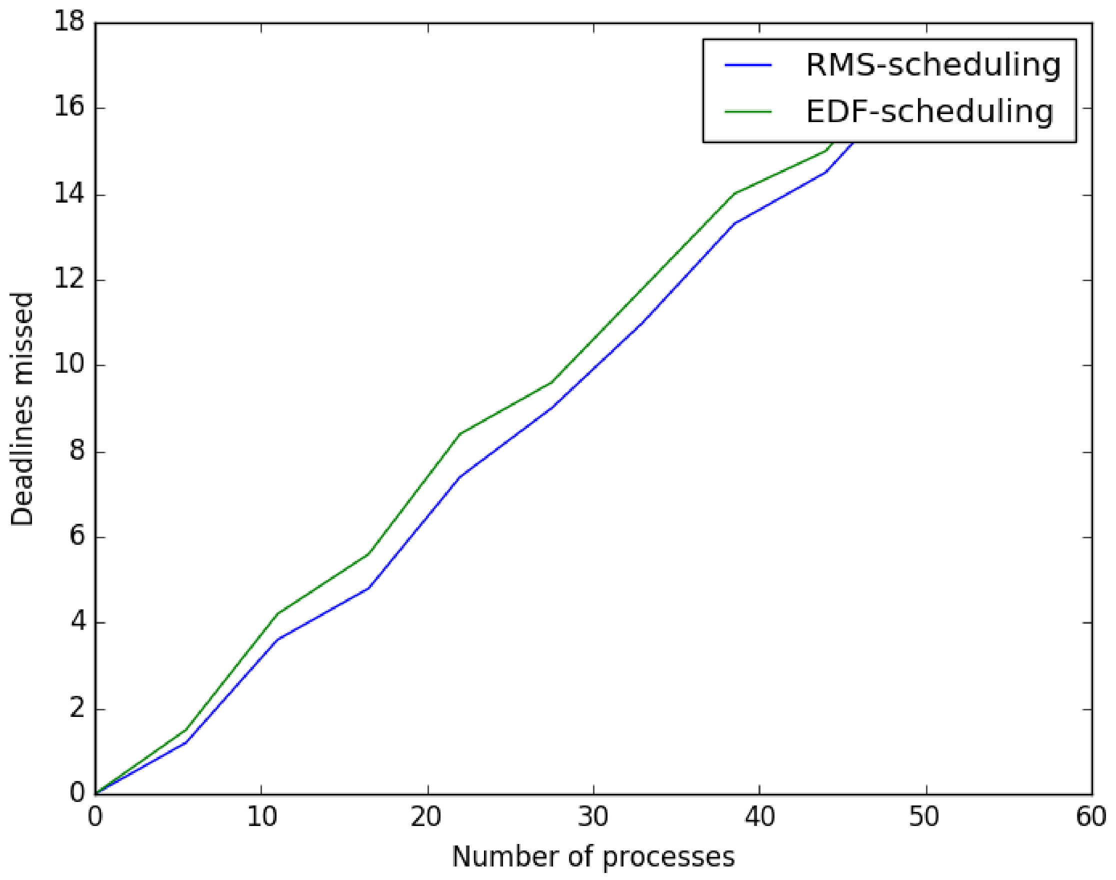
Calculation of turnaround time: When a process enters the ready_queue a start variable for that process is initialised. When a process leaves ready_queue turnaround time is calculated as (current_time-start).

Calculation of waiting time: A variable waiting for the process is initialised to 0.When a process has been preempted by some other process or enters the ready queue while another process is executing a temp variable for the process is iniatilised equal to current time. When a process resumes execution or starts execution waiting time is calculated as waiting=waiting+current_time-temp.

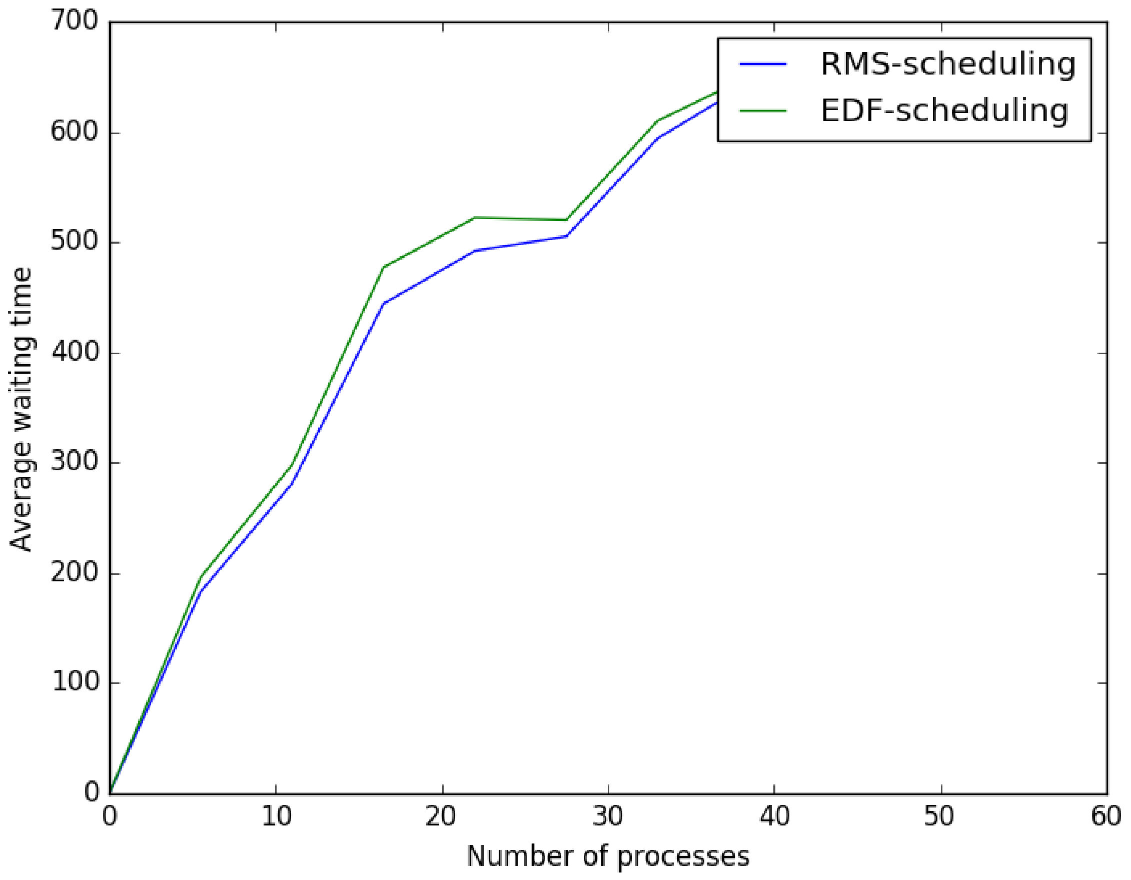
Complications arosing during the course f programming: Various complications arose during the programming phases. For finding the processes that missed the deadline many problems occured.

GRAPHS:

1: deadlines vs #processes graphs



It is clear from the graphs ,that the performance of RMS is turning out to be slightly better than the EDF. It may depend on input.



It is clear from the graphs that for smaller values of no of processes(<10) ,the average waiting time for RMS and EDF is the same. But for the higher values(>10),there is a deviation (EDF > RMS). However the result might vary depending on the input.