

Operating systems 2

Assignment 1

Report

The program consists of two, ready queue that contains the process ready to be executed and critical queue that contains the events that are supposed to arrive in the future.

These events consists of time of arrival of events and process that is going to come along with the event. The idea is to have events that can tell us when the next period is going to arrive.

These queues are sorted vectors that contains structs not objects. Struct is used rather than class because it is easier to implement and this program does not require classes specifically. So, structs are as good as classes for now.

At a simpler level, the program runs a for loop that iterates the current time. For each tm it is checked if current process is null or not, if yes then it is checked if it's time for a new event from critical queue. If found then a new process is created using critical queue's first event and this process is made the current process.

Then the current process is run until it completes but at each time while current process is running, we also check if some new process is arriving from the critical queue. If it is then, the new process preempts the current process if it's of high priority. Else it is put into ready queue. It is also checked if new process arriving is already in ready queue or not, if it is then its deadline miss and hence the same process in ready queue is terminated.

This whole procedure is iterated until both the ready queue and critical queue is empty. Then the program is terminated after printing the required stats.

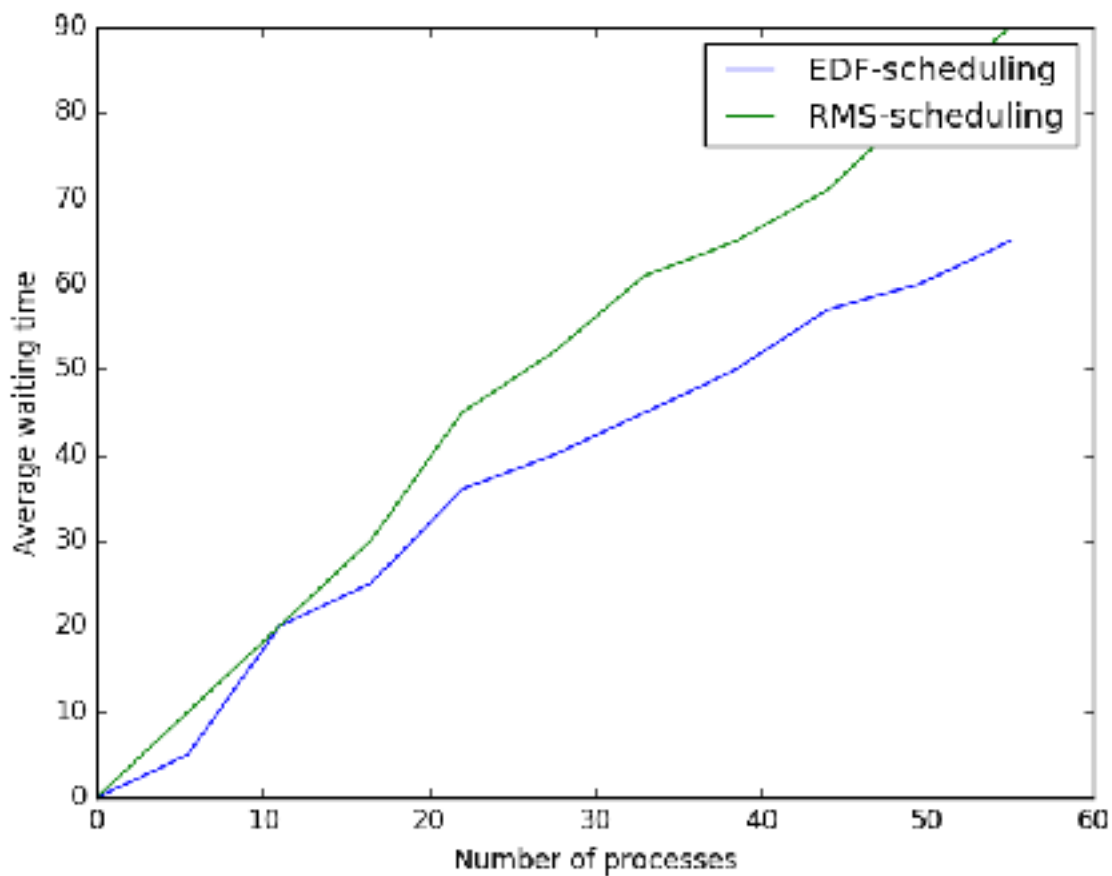
The difference between rms and edf simulations is just of the use of priority.

In rms- priority is kept constant (inverse of period).

In edf- priority can change (inverse of next critical queue event time corresponding to the process)

The complications faced while implementing the algorithm were :-

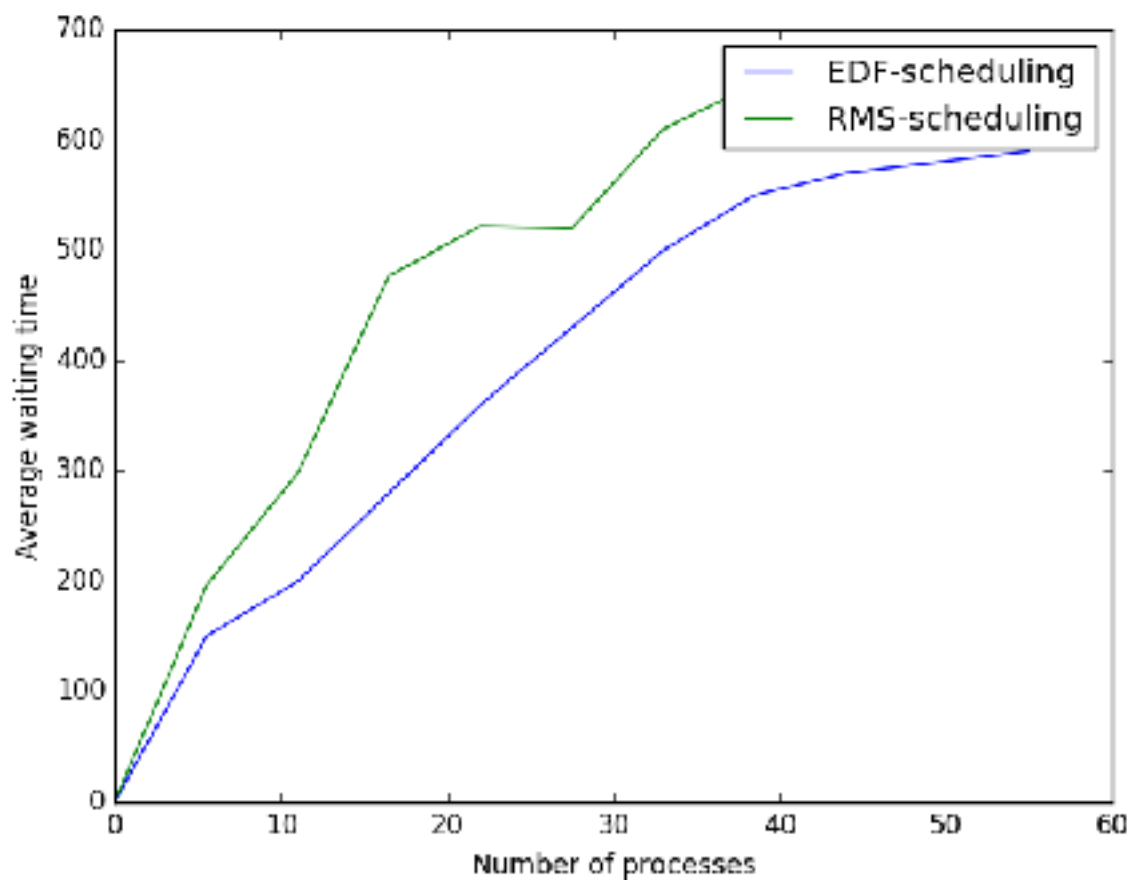
- If more than one process arrive at the same time from the critical queue then we have to check if each row process is missing ready queue's deadline.
- Calculating the waiting time for each process was hard because we have to keep track of each process events including entry to ready queue, preemption, start, stop, deadline miss and exists.



Analysing the graph:

Earliest deadline first should generally work better than rate monotonic scheduling because EDF assigns priority to the process dynamically depending on which process most needs it but in rms the priority remains unchanged throughout the process.

EDF prevents starvation while in rms, process with worst priority is kept from running and thus can lead To starvation.



Therefore the deadline misses in EDF should be less than rms. Hence, the graph of EDF is below rms.

Similarly, process will wait lesser for execution in EDF as compared to rms, as starvation is minimum in EDF.

One point to be noted is that at lesser number of processes the waiting time and misses are almost the same.

*But the above mentioned analysis is not always not true. Because the output also depend on the process.