Operating Systems–II: CS3523

Report On Programming Assignment-3

Implementing Rate-Monotonic Scheduling & Earliest Deadline First Scheduling through Discrete Event Simulation

CS20BTECH11018

Design:

Rate Monotonic Scheduling:

small data structures:

Here we created three kinds of small data structures.

- proc_vars – contains variables which describes the process regardless of which instant it is like period,processing_time,pid,avg_waiting_time,no of repetations.
- processing_token_t – contains variables which are related to the instance of the process like arrival time,deadline,runned time etc,.. and proc_vars pointer.
- event_token_t – its attributes describe about the event like it’s type,time when it arrives, and proc_vars pointer of the process it effects.

Compound data structures:

We also use big data structures like priority queues for queuing events according to arrival time,priority and event type for good structured order of occurrence of events in discrete event simulation,imitating ready queues which order processes according to it’s priority.Maps for storing states of processes etc,..

state variables:

Initially running_proc’s process points towards null all the other state variables like burst start time,estimated termination time,current event,simulation time are set to 0.

Event functions:

In DES,all arrival events invokes arrival event function which decides whether to run the process in the event or put it in the ready queue according to its priority with some complications.all deadline events invokes deadline event function which just changes states and helps in calculating stats related to simulation.

Simulation mechanisms:

Here whether the process is terminated successfully or not is decided by the while loop which runs when the estimated termination time or deadline of the running process is less than the present time.It will either terminate the process or discard it according to the relation between est. termination time and deadline(which is greater).It also contains a mechanism to choose the next process which is valid(whose deadline is greater than time of termination or discarding).

Event triggers:

All the necessary stuff to start a new process,to preempt the running process with a new process and to terminate the running process are done by functions related to them which are like triggers in DES ensuring the consistency of the system

Earliest Deadline First Scheduling:

Design of Earliest deadline first scheduling is similar to RMS.The only difference is we use deadlines to order events,processes in queues rather than priorities and other miscellaneous variables.

Small data structures:

Here everything are same except that proc_vars structures don’t contain priority attribute as there is no use of it.

Also processing_token_t contains extra attribute deadline for fast access because deadline is really important in EDF scheduling.

Also events are of only one type “arrival”.We decided to do this because their isn’t any important function that we miss if we don’t have deadline events.We can replace the functionality provided by deadline events by some modifications DFS loop and termination functions and decrease total no. of events by half.

Compound data structures:

Here also we arrange events in a priority queue according to ascending order of its arrival time but we don’t define how to arrange when arrival times of processes are same.

Here ready queues are arranged according to it’s deadline where nearer the deadline the closer the process to running in the ready queue.

State variables:

There are no changes in how state variables of DES are described.

Event Functions:

Here definition of every function is same except that we don't define deadline event function as there is no need for it(no deadline event is defined here).

Simulation mechanisms:

All simulation mechanisms are same.Except that we need to coverup for the lack of deadline events which helps in terminating the process.Especially terminating of processes at the end.

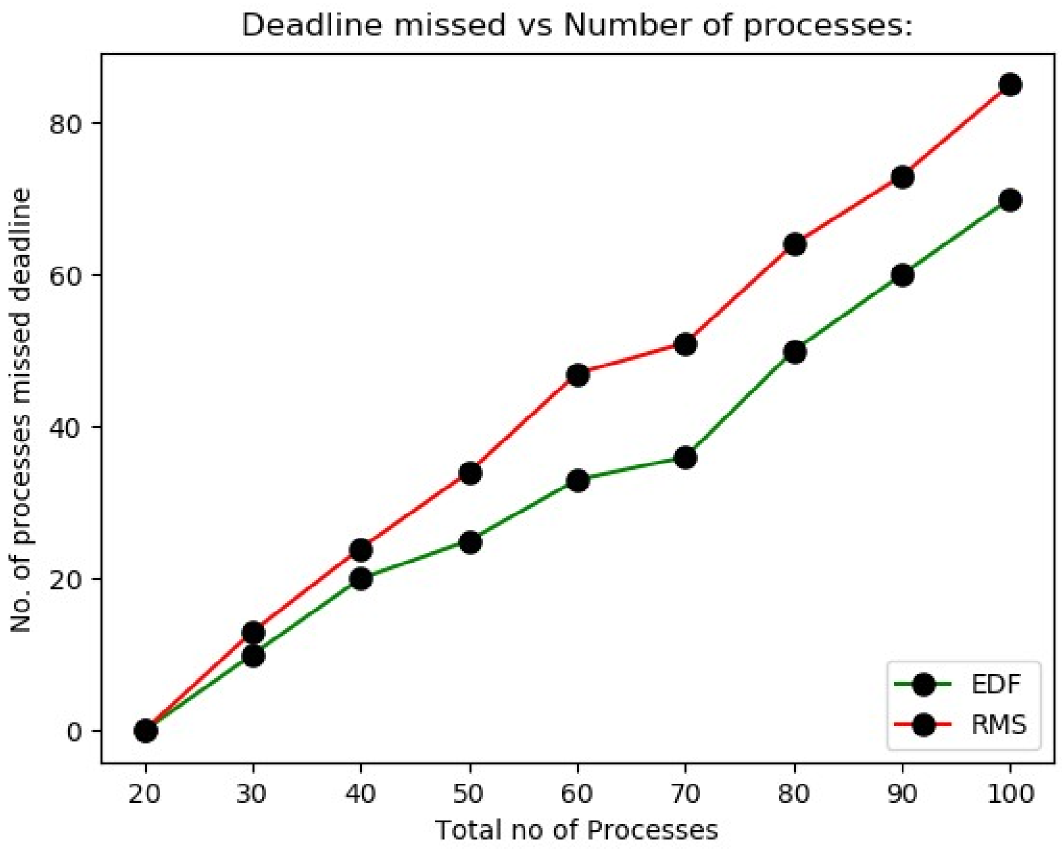
Event triggers:

Here everything are same except that we change the conditions of preemption (deadlines replacing priorities).

In this way,we tried to design RMS and EDF scheduling discrete event simulations accurately according to their scheduling algorithms.

Graphs:

Deadline missed vs Number of processes:



Input data:

Here input data is created by generating random integers which are constraint bound.As no. Of processes increase we just appended a new distinct process with ten repetations.

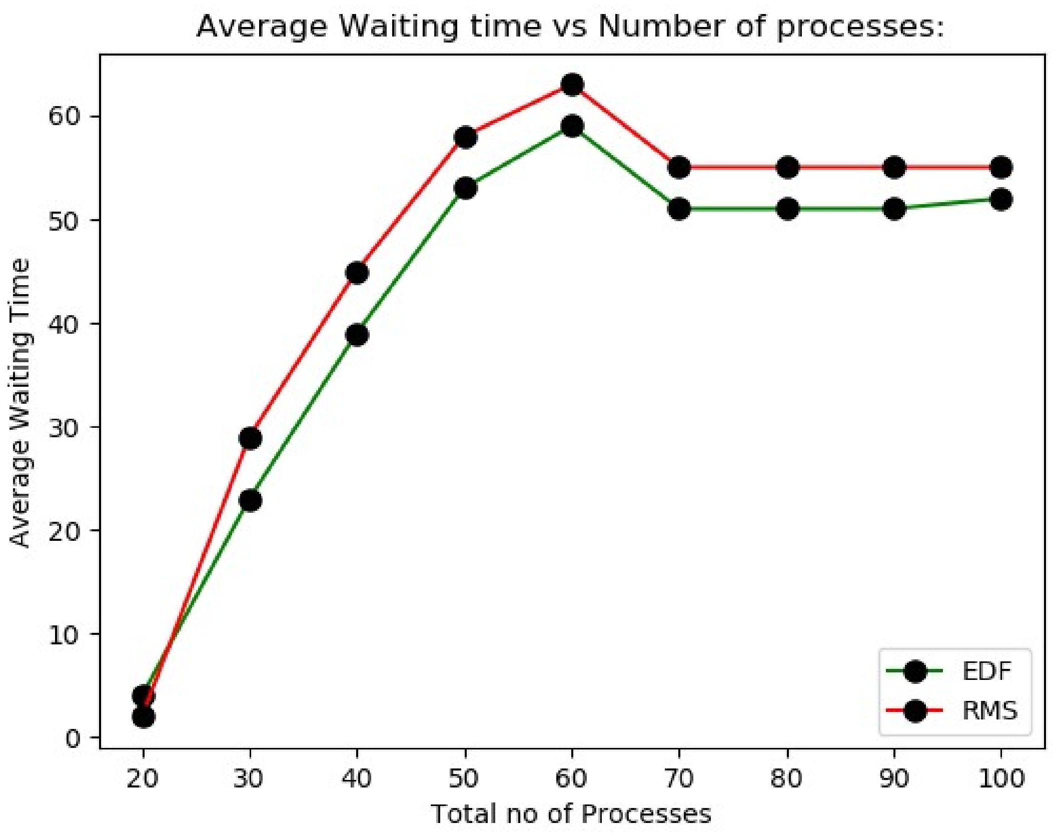
Analysis:

- No. of processes missing deadline in rms in higher than edf may be because in the input data we gave the relation between process time and period is suitable for edf than rms.
- As processes increase no. Of processes missing deadline increased rapidly.It depends on how the new process effects the old processes cycles.

Average waiting time vs Number of processes:

Input data:

Here input data is created by generating random integers which are constraint



bound.As no. Of processes increase we just appended a new distinct process with ten repetations.

Analysis:

- Average waiting times of rms and edf algorithms are close but distribution of waiting times among processes is more even in EDF algorithm than in RMS algorithm
- Average waiting times depend on the processes entered the system as we can observe from the graph average waiting time decreased when it goes 60 to 70.