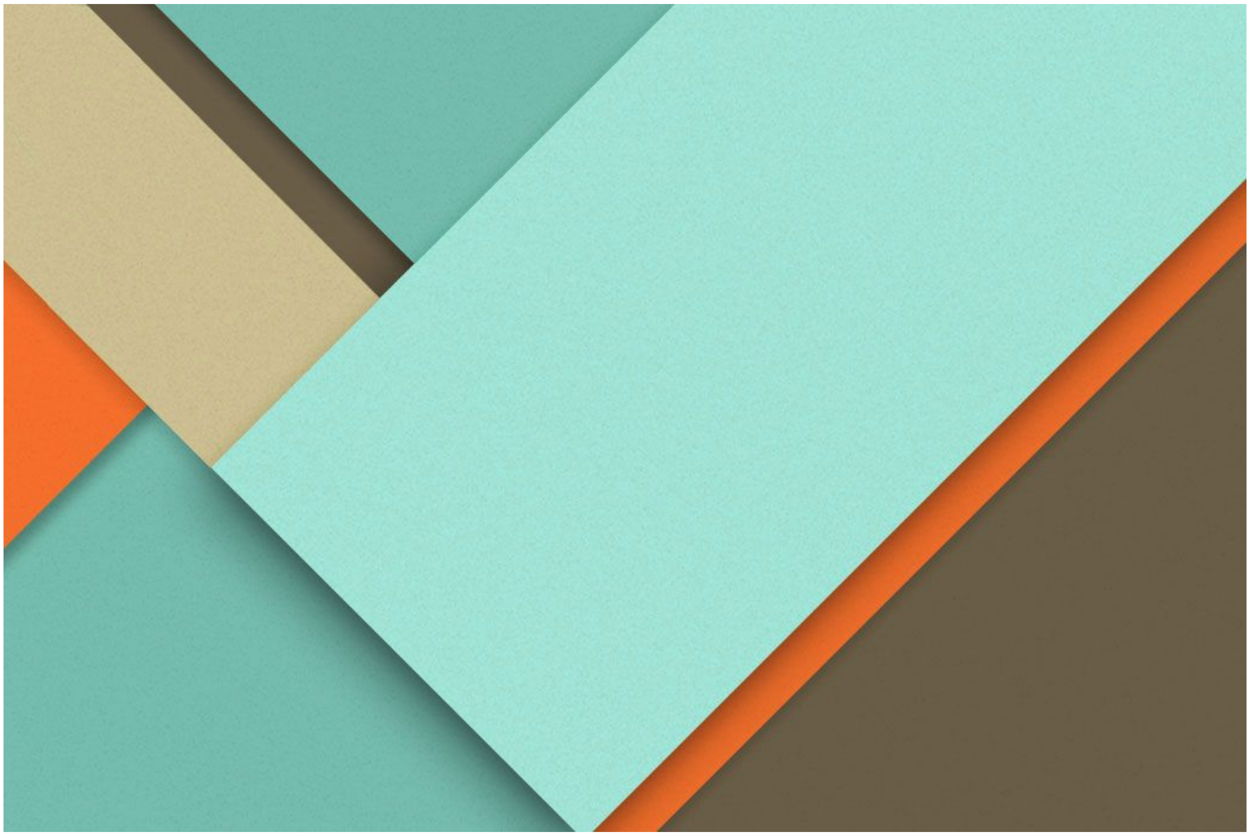


OS Assignment 1 Report

Rate Monotonic Scheduling and Earliest Deadline First Scheduling Simulations

SRIHITH PARUCHURU
ES17BTECH11016

Challenges Faced

- Representing process information i.e coming up with a data model

  Solution : Clearly there are types of process information , general and cycle specific . The period of a process is cycle invariant , so it comes under general attribute while system_joining_time is specific to the cycle number of a process.

- Updating deadline_miss_count , waiting_time for a given process

  Solution : Using a hashmap that maps process id to process info , will enable us to edit the corresponding process info in logarithmic time.

- Creating a data structure for ready queue , that enables process objects to be inserted and deleted . The data structure should also iterable so as to find out the least priority

  Solution : map STL in c++

  It supports insertion,deletion and lookups in O(log n) time

  It is iterable , so for getting max priority process we just have to iterate throughout the map

- Moving across time in discrete intervals

  Solution : All the required time intervals will be multiples of a process's period . Build a map that links time t with the list of processes that shall be joining the system at time t. Initialise the remaining_time of each of these processes with their respective processing_times .

Final Program Design

Both rm and edf programs involve the same design except for the priority functions.

Process data was stored in two types of structures .

- *struct process_info* : storing the general properties of a process like period,number_of_repetititons etc
- *struct process* : cycle specific properties of a process like system_joining_time,remaining_time were stored in another

Three hashmaps are maintained for quick lookups

- *all_process_hashmap* : maps pid(process id) to process_info . Initialised in constructor.

- *ready_hashmap* : maps pid to process . The ready hashmap essentially represents all the processes in a ready queue . The map data structure was used because it enables quick lookups (log n) and quick deletions.
- *time_process_hashmap* : maps time to a list of processes . This map is initialised in the constructor itself directly from the input. This map enables the simulation to make discrete time hops as compared to the unoptimised unit hop alternative.

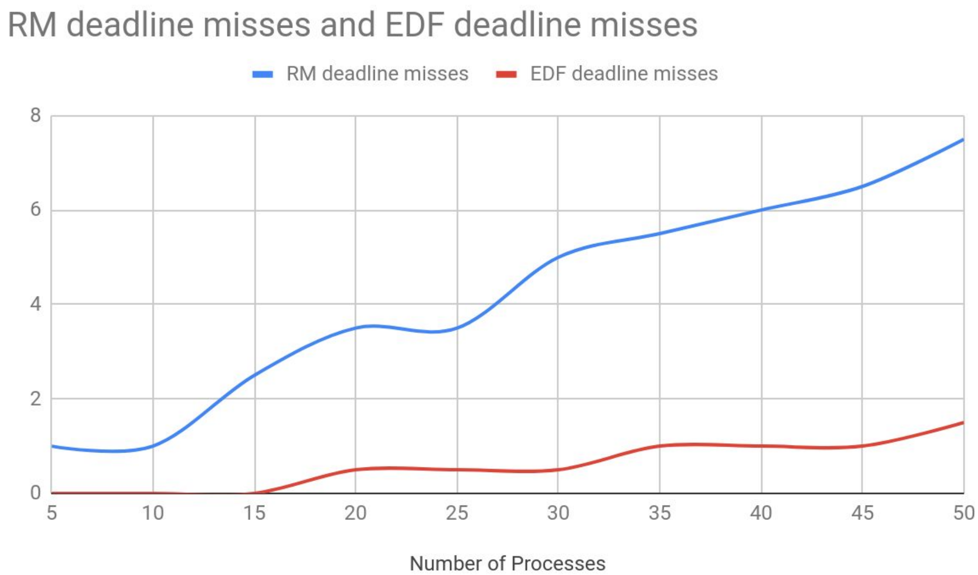
Simulation Algorithm

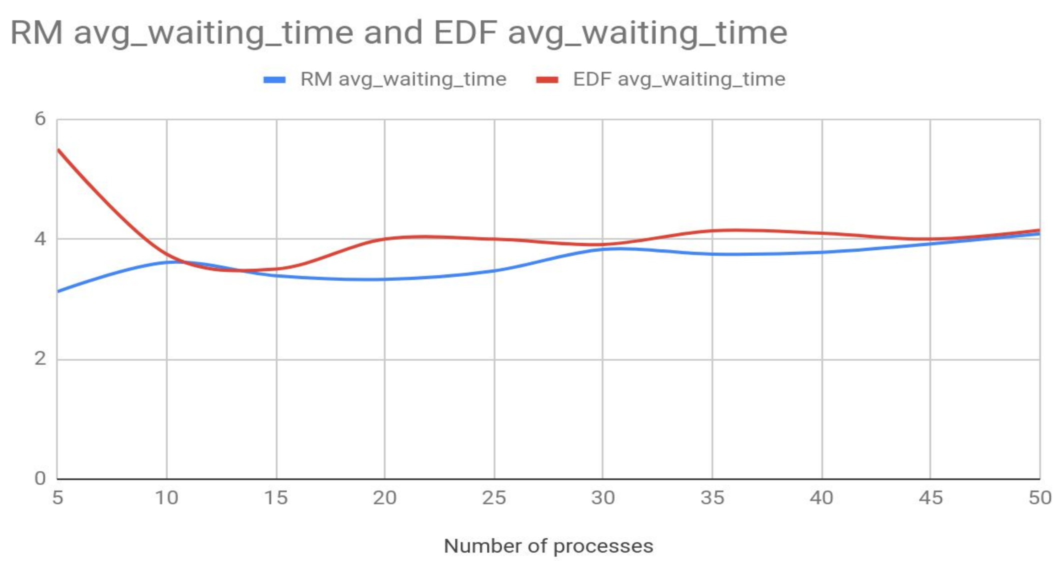
1. Initialise remaining_time of all processes to their corresponding processing_times. Build time_process_map and compute time_max from input and add an extra dummy process cycle for each process to ensure deadline checks for the last cycle. Initialise current_time to zero.
2. Insert all process at current_time into ready_hashmap . Before insertion , check if there is already the same process in ready_hashmap in which case the existing process had missed a deadline.
3. After the check, insert the entering process into the hashmap
4. If current_time equals time_max , Stop simulation
5. Based on the scheduling , fetch the process inside ready_hashmap with the highest priority
    a. In case of rm : The process with the least period has the highest priority
    b. In case of edf : The process with the least deadline has the highest priority. Deadline of the process is computed by multiplying the process period with it's iteration count.

6. If a process is already running , preempt it and set running_process to the above fetched process
7. Fetch the next_time value from time_process_map and keep running highest priority values till from the ready set till next_time occurs or ready_hashmap becomes empty
8. Set time to next_time and go to 2 .

RM vs EDF

Test data : Two cases were considered , one with CPU utilisation below 50% and another with 75% . The results of both these tests were averaged out to get the below plots.





Observations :

- The deadline missed for both scheduling algorithms are non decreasing
- RM deadline misses are greater than EDF deadline misses for any number of processes
- The rate of increase of deadline misses with respect to number of processes i.e the slope is higher for RM when compared to EDF
- The average waiting time seems to be more for EDF when compared to RM maybe because EDF has a higher chance of preemption than RM (because RM has static priorities , chance of preemption decreases).