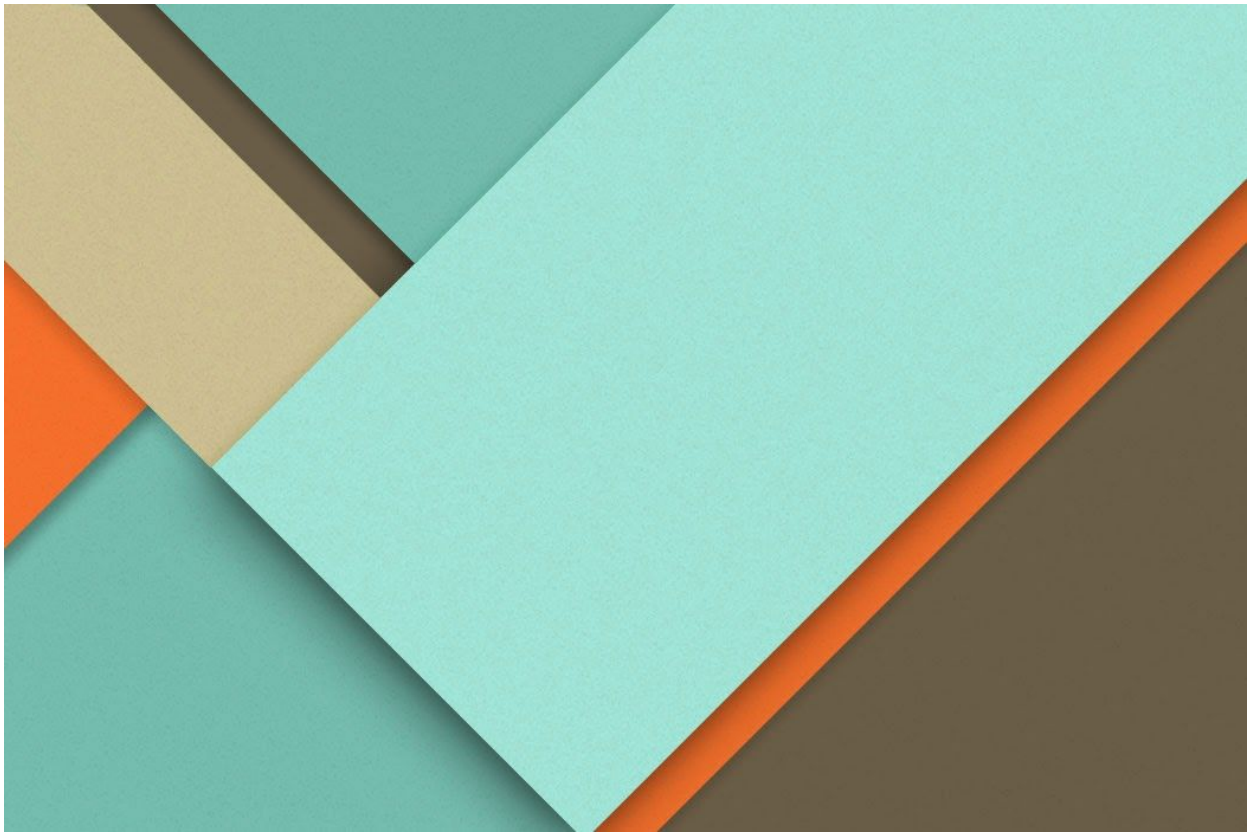


Rate Monotonic vs EDF - Judgement Day

GIORGIO C. BUTTAZZO, University of Pavia, Italy

Real-Time Systems, 29, 5–26, 2005 c 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands

Sridhar M
2018111021
Monsoon 2020
IIIT Hyderabad

Overview

The project is to understand and implement the selected research paper. In this case, I have taken the paper titled 'Rate monotonic vs EDF - Judgment Day'. This paper compares the two most popularly used priority-based scheduling algorithms EDF and RM. Since, many misconceptions still exist about the properties of these two scheduling methods, which usually tend to favor RM more than EDF. Typical misleading statements often-heard claim that RM is better than EDF, But the complete context is not provided.

Since these statements are either wrong or not precise, the authors of this paper have tried to clarify these issues in a systematic fashion, because the use of EDF allows better exploitation of the available resources and significantly improves the system's performance. Following are the criteria in which these algorithms were compared

- Implementation Complexity
- Runtime Overhead
- Schedulability analysis
- Robustness
- Jitter and Latency
- Resource Sharing
- Aperiodic task handling

Deliverables

1. C++ implementation of simulation of EDF and RM schedulers
2. Project report
3. Project presentation
4. Graph plots(ipynb file)

Experiments

1. **Preemptions introduced by RM and EDF as a function of the number of tasks**
   For each point in the graph, the average was computed over 200 independent simulations, each running for 1000 units of time. In each simulation, periods were generated as random variables, whereas execution times were computed to create a total processor utilization U = 0.5
2. **Preemptions introduced by RM and EDF as a function of the load**
   The behavior of RM and EDF as a function of the processor load, for a fixed number of tasks. The

figure shows the average number of preemptions as a function of the load for a set of **10 periodic tasks**. Periods and computation times were generated with the same criterion used in the previous experiment, but to create an average load ranging from 0.5 to 0.95

3. **Average normalized jitter for (a)U = 0.7, (b)U = 0.8, and (c)U = 0.9**
   The experiment has been performed to verify the jitter behavior under the two scheduling algorithms. Task sets of 10 periodic tasks were randomly generated and fixed total utilization U. The results refer to the Absolute Response-time Jitter (ARJ), which has been normalized with respect to task periods. Hence a value of 1 on task τ i corresponds to a jitter equal to its period T i . The results reported in the figures show the jitter introduced by the algorithms for each individual task (tasks are ordered by increasing periods).

Note

Conditions are needed to be updated in code as per requirements. Comments are made on the entire code base as well as I could. Helper/print code is to be uncommented if required. To get plots, update the output value in the .ipynb file.

Code execution

- To execute use the g++ compiler first open the terminal using ctrl+alt+t then type the below commands.
- First, unzip by using the "unzip *filename.zip*" command
- Then change the directory using cd in the terminal
- Then type "g++ *filename.cpp*" to compile the program
- To execute type "./a.out"

Understanding i/o

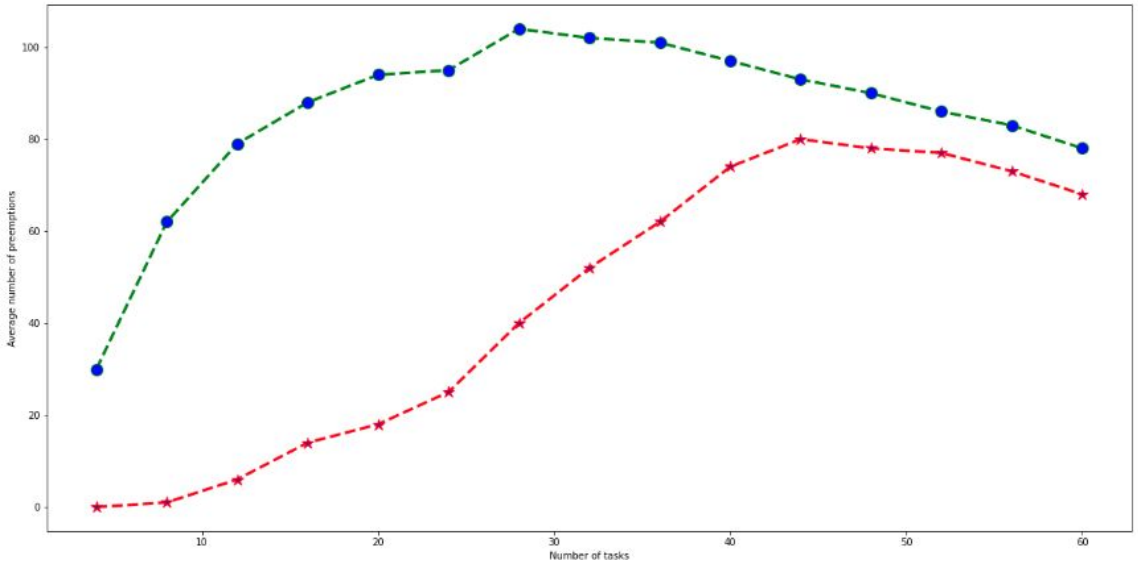
Input Taskset is generated inside the code only. It is randomly generated. Later it can be modified to take input from the input file (code portion for this is commented). Format for taskset is {n- number of tasks, n*{pid- giving id to the task; should be a natural number, p- period, t- execution time, k-number of times task will be running}}

The output shows plot points for all three experiments. **Log file and Stats file is also generated**.

For common input to schedulers, taskset has to be taken from a common input file.

Plots

I. Preemptions introduced by RM and EDF as a function of the number of tasks



II. Preemptions introduced by RM and EDF as a function of the load