

# Real-time Scheduling

After [isolating the scheduler](#), it is now straightforward to implement additional scheduling policies. Real-time periodic scheduling algorithms<sup>1</sup> are of special interest, since they can be easily implemented through a combination of the techniques explored in the previous exercises: scheduler, synchronizer and alarm. Remember that any scheduling policy is now implemented by keeping the scheduling queues ordered, so your new criteria must define [operator int\(\)](#).

## To do

Implement the following traditional periodic real-time scheduling algorithms: [Earliest Deadline First](#) and [Rate-monotonic](#).

You can use the [Cheddar](#) real-time scheduling tool to check your implementation. It can produce execution traces for a task set under both scheduling policies that should be closely matched by the output of the following test programs: [RM test \(traits\)](#) and [EDF test \(traits\)](#). For your convenience, Cheddar traces for a task set compatible with that of the test programs are available here: [RM Cheddar trace](#) and [RM Cheddar trace](#).

---

1. For a deeper insight on real-time scheduling, please refer to the following paper:

- **Giovani Gracioli, Antônio Augusto Fröhlich, Sebastian Fischmeister and Rodolfo Pellizzoni**, [Implementation and Evaluation of Global and Partitioned Scheduling in a Real-Time OS](#), In: Real-Time Systems, 49(6):669-714, 2013.