

# Take-Home Final Exam

ECE 455 S24

## DESCRIPTION

---

Implement a simulator for the Rate Monotonic single-core scheduling algorithm and report information about the results.

## WHAT TO DO

---

Your objective is to implement a simulator for the Rate Monotonic scheduling algorithm on a single-core CPU. Your solution should be a **Python3** program. Your program will read a set of periodic tasks from a file and report if the task set is schedulable under Rate Monotonic, as well as the number of times each task is preempted per hyperperiod. Note that the scheduling bound is a *not* a necessary condition for schedulability, you have to actually run the simulation to know.

Before you begin: set up a new Git repository on your computer. This repository will be handed in on Learn. You should commit your progress to this repository on a regular basis.

More formally, your Python3 program should have a main file called `ece_455_final.py` that takes a single command-line parameter: the name of the input file containing the tasks to be run. For example, if the workload you are simulating was defined in the file `workload1.txt` then your program should be run as:

```
python3 ece_455_final.py workload1.txt
```

The workloads you are to simulate will be given in text files with one task per line. So task  $T_0$  is defined on the first line of the file, then task  $T_1$  on the next line, task  $T_2$  on the following line, and so on. Each line contains information about that task in three positive numbers (not necessarily integers - precision up to 0.001) separated by commas. The first number will be the execution time of the task, the second number will be the period of the task, and the third number will be the relative deadline of the task. For example, an input file with  $T_0(e_0 = 1, P_0 = 3, D_0 = 3)$  and  $T_1(e_1 = 2, P_1 = 4, D_1 = 5)$  would be represented as a file containing only exactly the following two lines:

```
1,3,3
2,4,5
```

The output of your program should be printed to standard output. The output of your program should be structured as follows. On the first line, output a 1 if the scheduling algorithm produces a feasible schedule for this workload, otherwise output a 0. If you output a zero on the first line, the second line should be a blank line. If you output a one on the first line, the second line should be the number of times each task is preempted per hyperperiod. These preemption counts should be in-order of task and separated by a comma. For example, for the above input, task  $T_0$  is preempted zero times per hyperperiod and  $T_1$  is preempted once per hyperperiod. This should result exactly and only the following two lines as output:

```
1
0,1
```

The program will be run from the directory containing the `ece_455_final.py` file. You may include additional python files along with this file if you wish. Do not use Python features newer than version 3.10. Do not import any libraries that are not included in a default installation of Python 3.10 (except that you may optionally import numpy). Your program will be given a maximum of 20 seconds of execution time for each task set we test it on. Your program should not produce any other output beyond the two lines specified above.

There are additional test cases on Learn alongside this description file, please use them to **test your program before submitting**. This is not an exhaustive set of test cases, we will test your program on further test cases beyond those that are posted on Learn.

## DELIVERABLES

---

- A **zip file containing your Git repository** which contains your `ece_455_final.py` file along with any other files necessary to run your program (including the `.git` directory). The `ece_455_final.py` file should be at the top level of the repository (and the top level of the zip file), not in a nested directory. Submit this zip file to the dropbox on Learn.

## NOTICE

---

This is a final examination. It must be done individually. Collaboration is not permitted. Your work must be entirely your own.