

## **COMPE571 – Embedded Operating Systems – Fall 2025 – Programming Assignment 3**

**NOTE: You should form groups of 2 or 3 for this assignment.**

### **Assignment Description**

This programming assignment requires you to create a set of real-time task schedules along with DVFS settings corresponding to each task. This homework can be done on your PC using any programming language you choose to use. Your script or program will process an input text file that includes a list of tasks to be executed, their deadlines, and generates an output file listing a sequence of scheduling decisions based on four different scheduling policies: EDF, RM, Energy Efficient (EE) EDF and EE RM. Assume each task has a period that is equal to its deadline. For this programming assignment, we limit CPU to maximum four different frequencies/power settings. When CPU is not running a task, it consumes idle power at the lowest frequency.

The command line arguments of your program should be \$ your\_program <input\_file\_name> <EDF or RM> [EE]

In other words, the parameters are the file name, the scheduling strategy (either EDF or RM), and optionally EE (which stands for energy efficient). Two example calls could be:

```
$ ./my_solution input.txt EDF EE
```

```
$ ./my_solution input.txt RM
```

Below is an example of the input file (we use tab here for readability, but the actual file will be space-delimited). You may copy the following into a text file and use it for testing. Be sure to substitute the tabs with space characters. Note that all power values are in mW, and all execution times/deadlines are specified in seconds.

5	1000	625	447	307	212	84
w1	520	53	66	89	141	
w2	220	40	50	67	114	
w3	500	104	134	184	313	
w4	200	57	74	103	175	
w5	300	35	45	62	104	

The first row represents: <# of tasks> < the time the system will execute up to in seconds> <active CPU power @ 1188 Mhz> <active CPU power @ 918 Mhz> <active CPU power @ 648 Mhz> <active CPU power @ 384 Mhz> <idle CPU power @ lowest frequency>

All other rows represent: <name of task> <deadline/period> <WCET @ 1188 Mhz> <WCET @ 918 Mhz> <WCET @ 648 Mhz> <WCET @ 384 Mhz>. (WCET = worst-case execution time)

As the deliverable of this programming assignment, your program should parse such an input file, and generate scheduling sequences based on following four algorithms:

- EDF: Earliest deadline first when all tasks run at maximum CPU frequency
- RM: Rate-monotonic when all tasks run at maximum CPU frequency
- EE EDF/RM: Your scheduler needs to adjust the frequency of the CPU to be as low as possible (however, you still need to make sure that lowering the frequency for a task results in lower energy – remember that low power does not always mean low energy) while still meeting the deadlines and ensuring tasks are still scheduled in order as defined by either EDF or RM policy. An intuitive way to think about this is that you change each task's frequency to fill in ‘gaps’ in the schedule, while not violating the deadlines.

When a task is scheduled, it should output the following: <time started> <task name> <CPU frequency task runs at> <how long it ran for> <energy consumed in Joules>. If you are scheduling an idle period, use ‘IDLE’ as the task name and frequency value.

An example of an output might look like:

```

1      w1      1188  53      33.125J
54     IDLE   IDLE  17      1.428J
71     w3      648   20      6.140J

```

...etc...

At the end of the output schedule, you should also output the total energy consumption during the execution, percentage of time spent idle, and the total system execution time.

You will be required to turn in your script/source code into Canvas, the results of your real-time schedules (EDF,RM, EE EDF, EE RM) for the above sample input, along with discussion.

**25% Additional Credit:** If you find the optimal energy-efficient schedules, i.e., the schedules that lead to the least energy consumption and are still feasible, i.e., no deadline misses.

## Deliverables

Your assignment submission must include TWO files:

- A project report that describes how you designed your EDF, RM, EE EDF, and EE RM task schedulers (overview of the algorithms, how you determine how much to slow down jobs for EE, etc.). Show and discuss results for the input cases we provided (e.g., %idle time left in your energy efficient task scheduler, total energy cost). Please submit your report in DOCX or PDF (preferred) format.
- A zip file containing functional program(s)/script(s) (Note: you can use any programming language you want. Just make sure that your zip file includes a README file showing how to execute your program/script).

Your final score will be determined by the combination of your report and turning in a functional program on time. **Note that the report should be at most 4 pages long regardless of your extra credit attempt.**