

REPORT - GRADUATE PROJECT

Advanced Real Time Systems - ECE 5550G

Implementation of EDF and Its Comparison Against RM and DM

Rate Monotonic Scheduling Algorithm

Section 1 - A Brief Description

- It is a priority-based preemptive scheduling algorithm that assigns fixed priorities to each task based on their periods. The tasks with shorter periods receive a higher priority.
- We assume that the tasks are periodic and that their execution times are deterministic and known beforehand, and the context switching overhead is negligible compared to the execution time of the tasks.
- The task with the highest priority i.e. shortest period is always executed first, this ensures that all the deadlines are met. If a task misses its deadline, it is dropped and rescheduled to the next period of the task.

Section 2 - Code Implementation

By implementing the following edits to the scheduler.cpp, we can get the RM algorithm to work :

- **prvFindEmptyElementIndexTCB** : This function iterates over xTCBArray, and checks which TCB block is not in use and returns the index of that TCB block.
- **prvDeleteTCBFromArray** : This function deletes the TCB block at the index specified by the user by setting the xInUse parameter to pdFALSE.
- **prvCreateAllTasks** : This function creates a task and adds it to the list of tasks that are ready to run for each TCB block that is in use in the xTCBArray.
- **vSchedulerPeriodicTaskCreate** : This function adds the task's properties specified by the user to an empty TCB block in the xTCBArray.
- **vSchedulerPeriodicTaskDelete** : This function is used for deleting a task from the FreeRTOS kernel as well as its data in the xTCBArray.
- **prvPeriodicTaskRecreate** : This function recreates the task that has been deleted using the data from the task's corresponding TCB block in the xTCBArray.
- **prvPeriodicTaskCode** : This is a wrapper function that wraps each task created by the user. This function is called by FreeRTOS when a task is in the ready state and has highest priority within the list of ready tasks. This function in turn calls the corresponding task that needs to be run and delays until the release time after executing the task.
- **prvSetFixedPriorities** : This function assigns priorities to the tasks according to

the RM Algorithm. We keep iterating through the xTCBArray until all tasks have been assigned priorities; We find the task with the shortest period and assign the highest priority and so on and so forth.

- **prvSchedulerCheckTimingError** : This function checks for 2 types of timing errors: Deadline Miss and Maximum Execution Time.
 - For a Deadline miss, If a task has executed once and hasn't finished running, **prvCheckDeadline** function is called to check for a deadline miss. If a deadline miss has occurred, the task is deleted and recreated with new timing values.
 - For a Maximum Execution Exceeded, If a task has exceeded its WCET, the task is suspended. For a task that has already been suspended, check if the next release time of the task has arrived and resume the task. (**prvExecTimeExceedHook** function checks if any task has exceeded its execution time and sets the required flags).
- **prvSchedulerFunction** : This function is used to check for timing errors that might have occurred for any of the tasks. We iterate through the xTCBArray and for every task that is in use, we pass it to the **prvSchedulerCheckTimingError** to check for timing errors.
- **prvDeadlineMissedHook** : This function is called if a task has had a deadline miss. The task is deleted from the FreeRTOS kernel and recreated using **prvPeriodicTaskRecreate** function.
- **prvCheckDeadline** : This function checks if a given task has exceeded its absolute deadline and calls **prvDeadlineMissedHook** function.

Deadline Monotonic Scheduling Algorithm

Section 1 - A Brief Description

- It is similar to the Rate Monotonic (RM) Scheduling Algorithm. But, instead of prioritizing tasks based on their periods, the Deadline Monotonic (DM) Scheduling Algorithm prioritizes tasks based on their relative deadlines. The tasks with shorter relative deadlines will have a higher priority.
- The algorithm works by continuously monitoring the state of the system and selecting the highest priority task that is ready to execute. If a higher-priority task becomes ready while a lower-priority task is executing, the lower-priority task is preempted, and the higher-priority task is scheduled to run.

Section 2 - Code Implementation

By implementing the following edits to the scheduler.cpp, we can get the DM algorithm to work. Most of the code implementation is similar to the RM code Implementation except for the following :

- **prvSetFixedPriorities** : This function assigns priorities to the tasks according to the DM Algorithm. We keep iterating through the xTCBArray until all tasks have been assigned priorities; We find the task with the earliest deadline and assign the highest priority and so on and so forth.

Earliest Deadline First Scheduling Algorithm

Section 1 - A Brief Description

- Earliest Deadline First (EDF) is a scheduling algorithm used in real-time operating systems to prioritize tasks based on their deadlines. The basic idea of EDF is to schedule the task with the earliest deadline first, which ensures that the task with the closest deadline gets executed first.
- The algorithm works by maintaining a priority queue of tasks, ordered by their deadlines. When a new task arrives, it is added to the queue in its appropriate position, based on its deadline. The scheduler then selects the task with the earliest deadline from the head of the queue and executes it. If two tasks have the same deadline, the one with the higher priority is executed first.
- As tasks are executed, the deadline of each remaining task in the queue is updated, and the queue is re-sorted based on the new deadlines. This ensures that the scheduler always selects the task with the earliest deadline for execution.

One advantage of EDF is that it guarantees that all tasks meet their deadlines as long as the system is schedulable.

Section 2 - Code Implementation

By implementing the following edits to the scheduler.cpp, we can get the EDF algorithm to work. Most of the code implementation is similar to the RM code Implementation except :

- **prvSetPriorites** : This function assigns dynamic priorities to the tasks that are stored in the TCB List which is sorted based on the absolute deadline of the task.
- **prvUpdateTaskList** : This function adds tasks to the TCB List in a sorted fashion based on the absolute deadline of the task. It also stores the tasks that have overrun their deadline in an overflow TCB list.

Results

The following example sets will be used to compare and contrast the performance of EDF against that of DM and RM Algorithms.

Example Task Set 1

Table 1: Task set #1

Task	C	D	T
τ_1	100	400	400
τ_2	200	700	800
τ_3	150	1000	1000
τ_4	300	5000	5000

Example Task Set 2

Table 2: Task set #2

Task	C	D	T
τ_1	100	400	400
τ_2	150	200	500
τ_3	200	700	800
τ_4	150	1000	1000

- Running Task Set 1 Using EDF Algorithm

```

TASK: T1
TASK: T4
STAT: T1, ST:0000, ET:0008, RT:08, DT: 0024
STAT: T4, ST:0001, ET:0018, RT:17, DT: 0309
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
STAT: T2, ST:0019, ET:0032, RT:13, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0032, ET:0039, RT:07, DT: 0063
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T2
STAT: T2, ST:0052, ET:0061, RT:09, DT: 0090
TASK: T3
STAT: T3, ST:0063, ET:0069, RT:06, DT: 0126
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
TASK: T1
STAT: T1, ST:0096, ET:0100, RT:04, DT: 0120
TASK: T2
STAT: T2, ST:0100, ET:0109, RT:09, DT: 0138
TASK: T1
STAT: T1, ST:0120, ET:0124, RT:04, DT: 0144
TASK: T3
STAT: T3, ST:0126, ET:0132, RT:06, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T2
STAT: T2, ST:0148, ET:0157, RT:09, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T3
TASK: T1
STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T2
STAT: T2, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T3, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T1
STAT: T1, ST:0216, ET:0220, RT:04, DT: 0240
TASK: T1
STAT: T1, ST:0240, ET:0244, RT:04, DT: 0264
TASK: T2
STAT: T2, ST:0244, ET:0253, RT:09, DT: 0282
TASK: T3
STAT: T3, ST:0253, ET:0260, RT:07, DT: 0315
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T2
STAT: T2, ST:0292, ET:0301, RT:09, DT: 0330

```

```
TASK: T4
TASK: T1
STAT: T1, ST:0313, ET:0321, RT:08, DT: 0336
STAT: T4, ST:0309, ET:0327, RT:18, DT: 0618
TASK: T3
STAT: T3, ST:0327, ET:0334, RT:07, DT: 0378
TASK: T1
STAT: T1, ST:0336, ET:0340, RT:04, DT: 0360
TASK: T2
STAT: T2, ST:0340, ET:0349, RT:09, DT: 0378
TASK: T1
STAT: T1, ST:0360, ET:0364, RT:04, DT: 0384
TASK: T3
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T2
STAT: T2, ST:0388, ET:0397, RT:09, DT: 0426
STAT: T3, ST:0378, ET:0398, RT:20, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
TASK: T1
STAT: T1, ST:0432, ET:0436, RT:04, DT: 0456
TASK: T2
STAT: T2, ST:0436, ET:0445, RT:09, DT: 0474
TASK: T3
STAT: T3, ST:0445, ET:0452, RT:07, DT: 0504
TASK: T1
STAT: T1, ST:0456, ET:0460, RT:04, DT: 0480
TASK: T1
STAT: T1, ST:0480, ET:0484, RT:04, DT: 0504
TASK: T2
STAT: T2, ST:0484, ET:0493, RT:09, DT: 0522
TASK: T1
STAT: T1, ST:0504, ET:0508, RT:04, DT: 0528
TASK: T3
STAT: T3, ST:0508, ET:0515, RT:07, DT: 0567
TASK: T1
STAT: T1, ST:0528, ET:0532, RT:04, DT: 0552
TASK: T2
STAT: T2, ST:0532, ET:0541, RT:09, DT: 0570
TASK: T1
STAT: T1, ST:0552, ET:0556, RT:04, DT: 0576
TASK: T3
STAT: T3, ST:0567, ET:0573, RT:06, DT: 0630
TASK: T1
STAT: T1, ST:0576, ET:0580, RT:04, DT: 0600
TASK: T2
STAT: T2, ST:0580, ET:0589, RT:09, DT: 0618
TASK: T1
STAT: T1, ST:0600, ET:0604, RT:04, DT: 0624
---- Closed the serial port COM3 ----
```

- Running Task Set 1 Using DM Algorithm

```

----- Opened the serial port COM3 -----
----- Program Started -----
FUNC: vSchedulerInit
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T1
Phase Tick : 0
Max. Execution Tick : 6
Rel. Deadline Tick : 24
Period Tick : 24
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T2
Phase Tick : 0
Max. Execution Tick : 12
Rel. Deadline Tick : 42
Period Tick : 48
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T3
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 63
Period Tick : 63
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T4
Phase Tick : 0
Max. Execution Tick : 18
Rel. Deadline Tick : 309
Period Tick : 309
Priority : 0
-----
FUNC: vSchedulerStart
FUNC: prvSetFixedPriorities
----Using DM Scheduling Algorithm---
Task : T1, Priority : 5, Tick : 24
Task : T2, Priority : 4, Tick : 42
Task : T3, Priority : 3, Tick : 63
Task : T4, Priority : 2, Tick : 309
-----
FUNC: prvCreateSchedulerTask
---- Scheduler Details ----
Period Tick : 3
Priority : 6
Overhead : 0
-----
FUNC: prvCreateAllTasks
FUNC: prvPeriodicTaskCode -> TASK: T1, INIT RUN
TASK: T1
STAT: T1, ST:0000, ET:0004, RT:04, DT: 0024
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
STAT: T2, ST:0004, ET:0013, RT:09, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0013, ET:0020, RT:07, DT: 0063
FUNC: prvPeriodicTaskCode -> TASK: T4, INIT RUN
TASK: T4
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
STAT: T4, ST:0020, ET:0039, RT:19, DT: 0309
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T2
STAT: T2, ST:0052, ET:0061, RT:09, DT: 0090
TASK: T3
STAT: T3, ST:0063, ET:0069, RT:06, DT: 0126
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
TASK: T1
STAT: T1, ST:0096, ET:0100, RT:04, DT: 0120
TASK: T2
STAT: T2, ST:0100, ET:0109, RT:09, DT: 0138
TASK: T1
STAT: T1, ST:0120, ET:0124, RT:04, DT: 0144
TASK: T3
STAT: T3, ST:0126, ET:0132, RT:06, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T2
STAT: T2, ST:0148, ET:0157, RT:09, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T3
TASK: T1
STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T2
STAT: T2, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T3, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T1
STAT: T1, ST:0216, ET:0220, RT:04, DT: 0240
TASK: T1
STAT: T1, ST:0240, ET:0244, RT:04, DT: 0264
TASK: T2
STAT: T2, ST:0244, ET:0253, RT:09, DT: 0282

```

```
TASK: T3
STAT: T3, ST:0253, ET:0260, RT:07, DT: 0315
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T2
STAT: T2, ST:0292, ET:0301, RT:09, DT: 0330
TASK: T4
TASK: T1
STAT: T1, ST:0312, ET:0316, RT:04, DT: 0336
TASK: T3
STAT: T3, ST:0316, ET:0323, RT:07, DT: 0378
STAT: T4, ST:0309, ET:0334, RT:25, DT: 0618
TASK: T1
STAT: T1, ST:0336, ET:0340, RT:04, DT: 0360
TASK: T2
STAT: T2, ST:0340, ET:0349, RT:09, DT: 0378
TASK: T1
STAT: T1, ST:0360, ET:0364, RT:04, DT: 0384
TASK: T3
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T2
STAT: T2, ST:0388, ET:0397, RT:09, DT: 0426
STAT: T3, ST:0378, ET:0398, RT:20, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
TASK: T1
STAT: T1, ST:0432, ET:0436, RT:04, DT: 0456
TASK: T2
STAT: T2, ST:0436, ET:0445, RT:09, DT: 0474
---- Closed the serial port COM3 ----
```

- Running Task Set 1 Using RM Algorithm

```

---- Opened the serial port COM3 ----
---- Program Started ----
FUNC: vSchedulerInit
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name          : T1
Phase Tick    : 0
Max. Execution Tick : 6
Rel. Deadline Tick : 24
Period Tick   : 24
Priority      : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name          : T2
Phase Tick    : 0
Max. Execution Tick : 12
Rel. Deadline Tick : 42
Period Tick   : 48
Priority      : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name          : T3
Phase Tick    : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 63
Period Tick   : 63
Priority      : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name          : T4
Phase Tick    : 0
Max. Execution Tick : 18
Rel. Deadline Tick : 309
Period Tick   : 309
Priority      : 0
-----
FUNC: vSchedulerStart
FUNC: prvSetFixedPriorities
----Using RM Scheduling Algorithm---
Task : T1, Priority : 5, Tick : 24
Task : T2, Priority : 4, Tick : 48
Task : T3, Priority : 3, Tick : 63
Task : T4, Priority : 2, Tick : 309
-----
FUNC: prvCreateSchedulerTask
---- Scheduler Details ----
Period Tick  : 3
Priority     : 6
Overhead     : 0
-----
FUNC: prvCreateAllTasks
FUNC: prvPeriodicTaskCode -> TASK: T1, INIT RUN
TASK: T1
STAT: T1, ST:0000, ET:0004, RT:04, DT: 0024
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
STAT: T2, ST:0004, ET:0013, RT:09, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0013, ET:0020, RT:07, DT: 0063
FUNC: prvPeriodicTaskCode -> TASK: T4, INIT RUN
TASK: T4
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
STAT: T4, ST:0020, ET:0039, RT:19, DT: 0309
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T2
STAT: T2, ST:0052, ET:0061, RT:09, DT: 0090
TASK: T3
STAT: T3, ST:0063, ET:0069, RT:06, DT: 0126
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
TASK: T1
STAT: T1, ST:0096, ET:0100, RT:04, DT: 0120
TASK: T2
STAT: T2, ST:0100, ET:0109, RT:09, DT: 0138
TASK: T1
STAT: T1, ST:0120, ET:0124, RT:04, DT: 0144
TASK: T3
STAT: T3, ST:0126, ET:0132, RT:06, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T2
STAT: T2, ST:0148, ET:0157, RT:09, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T3
TASK: T1
STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T2
STAT: T2, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T3, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T1
STAT: T1, ST:0216, ET:0220, RT:04, DT: 0240
TASK: T1
STAT: T1, ST:0240, ET:0244, RT:04, DT: 0264
TASK: T2
STAT: T2, ST:0244, ET:0253, RT:09, DT: 0282

```

```
TASK: T3
STAT: T3, ST:0253, ET:0260, RT:07, DT: 0315
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T2
STAT: T2, ST:0292, ET:0301, RT:09, DT: 0330
TASK: T4
TASK: T1
STAT: T1, ST:0312, ET:0316, RT:04, DT: 0336
TASK: T3
STAT: T3, ST:0316, ET:0323, RT:07, DT: 0378
STAT: T4, ST:0309, ET:0334, RT:25, DT: 0618
TASK: T1
STAT: T1, ST:0336, ET:0340, RT:04, DT: 0360
TASK: T2
STAT: T2, ST:0340, ET:0349, RT:09, DT: 0378
TASK: T1
STAT: T1, ST:0360, ET:0364, RT:04, DT: 0384
TASK: T3
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T2
STAT: T2, ST:0388, ET:0397, RT:09, DT: 0426
STAT: T3, ST:0378, ET:0398, RT:20, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
TASK: T1
STAT: T1, ST:0432, ET:0436, RT:04, DT: 0456
TASK: T2
STAT: T2, ST:0436, ET:0445, RT:09, DT: 0474
TASK: T3
STAT: T3, ST:0445, ET:0452, RT:07, DT: 0504
TASK: T1
STAT: T1, ST:0456, ET:0460, RT:04, DT: 0480
---- Closed the serial port COM3 ----
```

Example Task Set 2

- Running Task Set 2 Using EDF Algorithm

```

FUNC: vSchedulerInit
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T1
Phase Tick : 0
Max. Execution Tick : 6
Rel. Deadline Tick : 24
Period Tick : 24
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T2
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 12
Period Tick : 30
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T3
Phase Tick : 0
Max. Execution Tick : 12
Rel. Deadline Tick : 42
Period Tick : 48
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T4
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 63
Period Tick : 63
Priority : 0
-----
FUNC: vSchedulerStart
FUNC: prvInit
FUNC: prvCreateSchedulerTask
---- Scheduler Details ----
Period Tick : 3
Priority : 6
-----
FUNC: prvCreateAllTasks
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
STAT: T2, ST:0000, ET:0006, RT:06, DT: 0012
FUNC: prvPeriodicTaskCode -> TASK: T1, INIT RUN
TASK: T1
STAT: T1, ST:0007, ET:0011, RT:04, DT: 0024
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0011, ET:0020, RT:09, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T4, INIT RUN
TASK: T4
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
TASK: T2
STAT: T2, ST:0030, ET:0036, RT:06, DT: 0042
STAT: T4, ST:0020, ET:0038, RT:18, DT: 0063
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T3
TASK: T2
STAT: T2, ST:0060, ET:0066, RT:06, DT: 0072
STAT: T3, ST:0052, ET:0068, RT:16, DT: 0090
TASK: T4
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
STAT: T4, ST:0068, ET:0079, RT:11, DT: 0126
TASK: T2
STAT: T2, ST:0090, ET:0096, RT:06, DT: 0102
TASK: T1
STAT: T1, ST:0096, ET:0101, RT:05, DT: 0120
TASK: T3
STAT: T3, ST:0101, ET:0110, RT:09, DT: 0138
TASK: T2
STAT: T2, ST:0120, ET:0126, RT:06, DT: 0132
TASK: T1
STAT: T1, ST:0126, ET:0131, RT:05, DT: 0144
TASK: T4
STAT: T4, ST:0131, ET:0138, RT:07, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T3
TASK: T2
STAT: T2, ST:0150, ET:0156, RT:06, DT: 0162
STAT: T3, ST:0148, ET:0164, RT:16, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T2
STAT: T2, ST:0180, ET:0186, RT:06, DT: 0192
TASK: T4
TASK: T1
STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T3
STAT: T3, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T4, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T2
STAT: T2, ST:0210, ET:0216, RT:06, DT: 0222
TASK: T1
STAT: T1, ST:0216, ET:0221, RT:05, DT: 0240

```

```
TASK: T2
STAT: T2, ST:0240, ET:0246, RT:06, DT: 0252
TASK: T1
STAT: T1, ST:0246, ET:0251, RT:05, DT: 0264
TASK: T3
STAT: T3, ST:0251, ET:0260, RT:09, DT: 0282
TASK: T4
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T2
STAT: T2, ST:0270, ET:0276, RT:06, DT: 0282
STAT: T4, ST:0260, ET:0278, RT:18, DT: 0315
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T3
TASK: T2
STAT: T2, ST:0300, ET:0306, RT:06, DT: 0312
STAT: T3, ST:0292, ET:0308, RT:16, DT: 0330
TASK: T1
STAT: T1, ST:0312, ET:0316, RT:04, DT: 0336
TASK: T4
STAT: T4, ST:0316, ET:0323, RT:07, DT: 0378
TASK: T2
STAT: T2, ST:0330, ET:0336, RT:06, DT: 0342
TASK: T1
STAT: T1, ST:0336, ET:0341, RT:05, DT: 0360
TASK: T3
STAT: T3, ST:0341, ET:0350, RT:09, DT: 0378
TASK: T2
STAT: T2, ST:0360, ET:0366, RT:06, DT: 0372
TASK: T1
STAT: T1, ST:0366, ET:0371, RT:05, DT: 0384
TASK: T4
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T3
TASK: T2
STAT: T2, ST:0390, ET:0396, RT:06, DT: 0402
STAT: T3, ST:0388, ET:0404, RT:16, DT: 0426
STAT: T4, ST:0378, ET:0405, RT:27, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
---- Closed the serial port COM3 ----
```

- Running Task Set 2 Using DM Algorithm

```

FUNC: vSchedulerInit
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T1
Phase Tick : 0
Max. Execution Tick : 6
Rel. Deadline Tick : 24
Period Tick : 24
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T2
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 12
Period Tick : 30
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T3
Phase Tick : 0
Max. Execution Tick : 12
Rel. Deadline Tick : 42
Period Tick : 48
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T4
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 63
Period Tick : 63
Priority : 0
-----
FUNC: vSchedulerStart
FUNC: prvSetFixedPriorities
----Using DM Scheduling Algorithm----
Task : T2, Priority : 5, Tick : 12
Task : T1, Priority : 4, Tick : 24
Task : T3, Priority : 3, Tick : 42
Task : T4, Priority : 2, Tick : 63
-----
FUNC: prvCreateSchedulerTask
---- Scheduler Details ----
Period Tick : 3
Priority : 6
Overhead : 0
-----
FUNC: prvCreateAllTasks
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
STAT: T2, ST:0000, ET:0006, RT:06, DT: 0012
FUNC: prvPeriodicTaskCode -> TASK: T1, INIT RUN
TASK: T1
STAT: T1, ST:0007, ET:0011, RT:04, DT: 0024
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0011, ET:0020, RT:09, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T4, INIT RUN
TASK: T4
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
TASK: T2
STAT: T2, ST:0030, ET:0036, RT:06, DT: 0042
STAT: T4, ST:0020, ET:0038, RT:18, DT: 0063
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T3
TASK: T2
STAT: T2, ST:0060, ET:0066, RT:06, DT: 0072
STAT: T3, ST:0052, ET:0068, RT:16, DT: 0090
TASK: T4
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
STAT: T4, ST:0068, ET:0079, RT:11, DT: 0126
TASK: T2
STAT: T2, ST:0090, ET:0096, RT:06, DT: 0102
TASK: T1
STAT: T1, ST:0096, ET:0101, RT:05, DT: 0120
TASK: T3
STAT: T3, ST:0101, ET:0110, RT:09, DT: 0138
TASK: T2
STAT: T2, ST:0120, ET:0126, RT:06, DT: 0132
TASK: T1
STAT: T1, ST:0126, ET:0131, RT:05, DT: 0144
TASK: T4
STAT: T4, ST:0131, ET:0138, RT:07, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T3
TASK: T2
STAT: T2, ST:0150, ET:0156, RT:06, DT: 0162
STAT: T3, ST:0148, ET:0164, RT:16, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T2
STAT: T2, ST:0180, ET:0186, RT:06, DT: 0192
TASK: T4
TASK: T1

```

```
STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T3
STAT: T3, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T4, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T2
STAT: T2, ST:0210, ET:0216, RT:06, DT: 0222
TASK: T1
STAT: T1, ST:0216, ET:0221, RT:05, DT: 0240
TASK: T2
STAT: T2, ST:0240, ET:0246, RT:06, DT: 0252
TASK: T1
STAT: T1, ST:0246, ET:0251, RT:05, DT: 0264
TASK: T3
STAT: T3, ST:0251, ET:0260, RT:09, DT: 0282
TASK: T4
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T2
STAT: T2, ST:0270, ET:0276, RT:06, DT: 0282
STAT: T4, ST:0260, ET:0278, RT:18, DT: 0315
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T3
TASK: T2
STAT: T2, ST:0300, ET:0306, RT:06, DT: 0312
STAT: T3, ST:0292, ET:0308, RT:16, DT: 0330
TASK: T1
STAT: T1, ST:0312, ET:0316, RT:04, DT: 0336
TASK: T4
STAT: T4, ST:0316, ET:0323, RT:07, DT: 0378
TASK: T2
STAT: T2, ST:0330, ET:0336, RT:06, DT: 0342
TASK: T1
STAT: T1, ST:0336, ET:0341, RT:05, DT: 0360
TASK: T3
STAT: T3, ST:0341, ET:0350, RT:09, DT: 0378
TASK: T2
STAT: T2, ST:0360, ET:0366, RT:06, DT: 0372
TASK: T1
STAT: T1, ST:0366, ET:0371, RT:05, DT: 0384
TASK: T4
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T3
TASK: T2
STAT: T2, ST:0390, ET:0396, RT:06, DT: 0402
STAT: T3, ST:0388, ET:0404, RT:16, DT: 0426
STAT: T4, ST:0378, ET:0405, RT:27, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
TASK: T2
STAT: T2, ST:0420, ET:0426, RT:06, DT: 0432
---- Closed the serial port COM3 ----
```

- Running Task Set 2 Using RM Algorithm

```

FUNC: vSchedulerInit
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T1
Phase Tick : 0
Max. Execution Tick : 6
Rel. Deadline Tick : 24
Period Tick : 24
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T2
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 12
Period Tick : 30
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T3
Phase Tick : 0
Max. Execution Tick : 12
Rel. Deadline Tick : 42
Period Tick : 48
Priority : 0
-----
FUNC: vSchedulerPeriodicTaskCreate
---- Task Details ----
Name : T4
Phase Tick : 0
Max. Execution Tick : 9
Rel. Deadline Tick : 63
Period Tick : 63
Priority : 0
-----
FUNC: vSchedulerStart
FUNC: prvSetFixedPriorities
----Using RM Scheduling Algorithm---
Task : T1, Priority : 5, Tick : 24
Task : T2, Priority : 4, Tick : 30
Task : T3, Priority : 3, Tick : 48
Task : T4, Priority : 2, Tick : 63
-----
FUNC: prvCreateSchedulerTask
---- Scheduler Details ----
Period Tick : 3
Priority : 6
----- Overhead : 0 -----
FUNC: prvCreateAllTasks
FUNC: prvPeriodicTaskCode -> TASK: T1, INIT RUN
TASK: T1
STAT: T1, ST:0000, ET:0004, RT:04, DT: 0024
FUNC: prvPeriodicTaskCode -> TASK: T2, INIT RUN
TASK: T2
STAT: T2, ST:0004, ET:0011, RT:07, DT: 0012
FUNC: prvPeriodicTaskCode -> TASK: T3, INIT RUN
TASK: T3
STAT: T3, ST:0011, ET:0020, RT:09, DT: 0042
FUNC: prvPeriodicTaskCode -> TASK: T4, INIT RUN
TASK: T4
TASK: T1
STAT: T1, ST:0024, ET:0028, RT:04, DT: 0048
TASK: T2
STAT: T2, ST:0030, ET:0036, RT:06, DT: 0042
STAT: T4, ST:0020, ET:0038, RT:18, DT: 0063
TASK: T1
STAT: T1, ST:0048, ET:0052, RT:04, DT: 0072
TASK: T3
TASK: T2
STAT: T2, ST:0060, ET:0066, RT:06, DT: 0072
STAT: T3, ST:0052, ET:0068, RT:16, DT: 0090
TASK: T4
TASK: T1
STAT: T1, ST:0072, ET:0076, RT:04, DT: 0096
STAT: T4, ST:0068, ET:0079, RT:11, DT: 0126
TASK: T2
TASK: T1
STAT: T1, ST:0096, ET:0100, RT:04, DT: 0120
STAT: T2, ST:0090, ET:0101, RT:11, DT: 0102
TASK: T3
STAT: T3, ST:0101, ET:0110, RT:09, DT: 0138
TASK: T1
STAT: T1, ST:0120, ET:0124, RT:04, DT: 0144
TASK: T2
STAT: T2, ST:0124, ET:0131, RT:07, DT: 0132
TASK: T4
STAT: T4, ST:0131, ET:0138, RT:07, DT: 0189
TASK: T1
STAT: T1, ST:0144, ET:0148, RT:04, DT: 0168
TASK: T3
TASK: T2
STAT: T2, ST:0150, ET:0156, RT:06, DT: 0162
STAT: T3, ST:0148, ET:0164, RT:16, DT: 0186
TASK: T1
STAT: T1, ST:0168, ET:0172, RT:04, DT: 0192
TASK: T2
STAT: T2, ST:0180, ET:0186, RT:06, DT: 0192
TASK: T4
TASK: T1

```

STAT: T1, ST:0192, ET:0196, RT:04, DT: 0216
TASK: T3
STAT: T3, ST:0196, ET:0205, RT:09, DT: 0234
STAT: T4, ST:0189, ET:0209, RT:20, DT: 0252
TASK: T2
TASK: T1
STAT: T1, ST:0216, ET:0220, RT:04, DT: 0240
STAT: T2, ST:0210, ET:0221, RT:11, DT: 0222
TASK: T1
STAT: T1, ST:0240, ET:0244, RT:04, DT: 0264
TASK: T2
STAT: T2, ST:0244, ET:0251, RT:07, DT: 0252
TASK: T3
STAT: T3, ST:0251, ET:0260, RT:09, DT: 0282
TASK: T4
TASK: T1
STAT: T1, ST:0264, ET:0268, RT:04, DT: 0288
TASK: T2
STAT: T2, ST:0270, ET:0276, RT:06, DT: 0282
STAT: T4, ST:0260, ET:0278, RT:18, DT: 0315
TASK: T1
STAT: T1, ST:0288, ET:0292, RT:04, DT: 0312
TASK: T3
TASK: T2
STAT: T2, ST:0300, ET:0306, RT:06, DT: 0312
STAT: T3, ST:0292, ET:0308, RT:16, DT: 0330
TASK: T1
STAT: T1, ST:0312, ET:0316, RT:04, DT: 0336
TASK: T4
STAT: T4, ST:0316, ET:0323, RT:07, DT: 0378
TASK: T2
TASK: T1
STAT: T1, ST:0336, ET:0340, RT:04, DT: 0360
STAT: T2, ST:0330, ET:0341, RT:11, DT: 0342
TASK: T3
STAT: T3, ST:0341, ET:0350, RT:09, DT: 0378
TASK: T1
STAT: T1, ST:0360, ET:0364, RT:04, DT: 0384
TASK: T2
STAT: T2, ST:0364, ET:0371, RT:07, DT: 0372
TASK: T4
TASK: T1
STAT: T1, ST:0384, ET:0388, RT:04, DT: 0408
TASK: T3
TASK: T2
STAT: T2, ST:0390, ET:0396, RT:06, DT: 0402
STAT: T3, ST:0388, ET:0404, RT:16, DT: 0426
STAT: T4, ST:0378, ET:0405, RT:27, DT: 0441
TASK: T1
STAT: T1, ST:0408, ET:0412, RT:04, DT: 0432
TASK: T2
STAT: T2, ST:0420, ET:0426, RT:06, DT: 0432
TASK: T1
STAT: T1, ST:0432, ET:0436, RT:04, DT: 0456

Inference

Looking at the output of EDF for example set 1 and 2, we see that priorities are assigned dynamically to the tasks based on their absolute deadlines and also we can notice there are no deadline misses when we use this algorithm.

Looking at the output of DM for example set 1 and 2, we can see that the tasks are assigned static priorities based on the Deadline Monotonic Algorithm.

Looking at the output of RM for example set 1 and 2, we can see that the tasks are assigned static priorities based on the Rate Monotonic Algorithm.

RM and DM provide a more accurate response time analysis than EDF. This is because their priority assignment is based on the task's period and deadline respectively, which makes the computation of the worst-case response time simpler.

EDF requires more processing overhead than RM and DM because it has to check the deadlines of all pending tasks to determine the next task to execute. RM and DM require less processing overhead because the priority of each task is determined statically.

REFERENCES

1. Please read the README.md before getting started with the code
2. Help on Reading the Output —
 - a. FUNC : <Function Name> refers to the function called
 - b. TASK : <Task Name> refers to the task that started executing.
 - c. STAT : <Task Name>, ST : <num>, ET : <num>, RT : <num>, DT : <num> shows the stats of the task that finished executing:-
 - i. ST is the tick at which the task starts executing.
 - ii. ET is the tick at which the task finished executing.
 - iii. RT is the number of ticks that task took to complete (including how long the task was preempted, if it was preempted)
 - iv. DT is the absolute deadline of the task in ticks.