



EDF Scheduler Project

Team Members

Arafa Arafa

Anas Mahmoud

Bassel Yasser

Mahmoud Adel

Youssef Abbas

Table of Contents

<i>Project Description</i>	<i>2</i>
<i>Requirements</i>	<i>2</i>
<i>Tasks.....</i>	<i>3</i>
<i>Hyper period</i>	<i>3</i>
<i>CPU Load.....</i>	<i>4</i>
<i>URM Calculation</i>	<i>4</i>
<i>Time demand Calculation</i>	<i>5</i>
<i>Simulation on SimSo.....</i>	<i>8</i>

Project Description

The EDF Scheduler project aims to implement an Earliest Deadline First (EDF) scheduling algorithm using FreeRTOS. The EDF scheduler is a dynamic priority-based preemptive scheduling policy commonly used in real-time systems. This project involves implementing the necessary changes outlined in a provided thesis, creating tasks with specific criteria, verifying the system implementation, and optionally enhancing the functionality of the scheduler.

Requirements

The following requirements must be fulfilled to successfully complete the EDF scheduler implementation project:

- Read the provided thesis, specifically Chapter 2: "FreeRTOS Task Scheduling" and Chapter 3: "EDF Scheduler."
- Implement the changes mentioned in Chapter 3.2.2: "Implementation in FreeRTOS." The changes should be made in the tasks.c source file.
- Implement any missing changes not mentioned in the thesis:
- Modify the prvIdleTask function to ensure the idle task always has the farthest deadline.
- In the xTaskIncrementTick function, calculate the new task deadline and insert it in the correct position in the EDF ready list for every tick increment.
- Modify the preemption mechanism in the xTaskIncrementTick function, so that a task with a sooner deadline preempts a task with a larger deadline.
- Implement four tasks using the EDF scheduler. The tasks and their criteria are as follows:
 - Task 1: "Button_1_Monitor," Periodicity: 50ms, Deadline: 50ms.
 - Task 2: "Button_2_Monitor," Periodicity: 50ms, Deadline: 50ms.
 - Task 3: "Periodic_Transmitter," Periodicity: 100ms, Deadline: 100ms.
 - Task 4: "Uart_Receiver," Periodicity: 20ms, Deadline: 20ms.
 - Add two additional tasks to simulate a heavier load:
 - Task 5: "Load_1_Simulation," Periodicity: 10ms, Deadline: 10ms, Execution time: 5ms.
 - Task 6: "Load_2_Simulation," Periodicity: 100ms, Deadline: 100ms, Execution time: 12ms.
- Implement all the tasks mentioned above in the same main.c source file.

Tasks

Task id	Periodicity	Execution time (ms)
T1_Button_1_Monitor	50	0.0144
T2_Button_2_Monitor	50	0.0144
T3_Periodic_Transmitter	100	0.013
T4_Uart_Receiver	20	0.0246
T5_Load_1_Simulation	10	5
T6_Load_2_Simulation	100	12

The following table is shown the tasks with each (Periodicity, Deadline, Busy time)

Hyper period

Hyper period = LCM (tasks periodicity) = LCM (100,20,10,50) = 100

Task id	Periodicity	Deadline	Busy time ($E \cdot (P/H)$) (ms)
T1_Button_1_Monitor	50	50	0.0288
T2_Button_2_Monitor	50	50	0.0288
T3_Periodic_Transmitter	100	100	0.01308
T4_Uart_Receiver	20	20	0.0984
T5_Load_1_Simulation	10	10	50
T6_Load_2_Simulation	100	100	12
Total			62.169

CPU Load

CPU load = (Total busy time / Hyper period)

CPU load = 62.169 / 100

= 62.169%

URM Calculation

Task id	Periodicity	Execution time	U
T1_Button_1_Monitor	50	0.0144	0.0144/50
T2_Button_2_Monitor	50	0.0144	0.0144/50
T3_Periodic_Transmitter	100	0.013	0.013/100
T4_Uart_Receiver	20	0.0246	0.0246/20
T5_Load_1_Simulation	10	5	5/10
T6_Load_2_Simulation	100	12	12/100
Total			0.6219

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \leq n(2^{\frac{1}{n}} - 1)$$

$$U = 0.6219 \text{ (62.19\%)}$$

$$URM = n \left(2^{\frac{1}{n}} - 1 \right) = 6 \left(2^{\frac{1}{6}} - 1 \right) = 0.7347 \text{ (73.47\%)}$$

$$U < URM \rightarrow U < URM$$

Time demand Calculation

$$w(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil e_k \text{ for } 0 < t \leq p_i$$

Task1

$$w(20) = 0.0144 + \left\lceil \frac{20}{10} \right\rceil * 5 + \left\lceil \frac{20}{20} \right\rceil * 0.0137 = 10.0281$$

$$w(25) = 0.0144 + \left\lceil \frac{20}{10} \right\rceil * 5 + \left\lceil \frac{20}{20} \right\rceil * 0.0137 = 12.53$$

$$w(40) = 0.0144 + \left\lceil \frac{40}{10} \right\rceil * 5 + \left\lceil \frac{40}{20} \right\rceil * 0.0137 = 20.0418$$

$$w(50) = 0.0144 + \left\lceil \frac{50}{10} \right\rceil * 5 + \left\lceil \frac{50}{20} \right\rceil * 0.0137 = 25.04$$

$\therefore w(50) < 50$, *T1 is schedulable*

Task2

$$w(20) = 0.0144 + \left\lceil \frac{20}{10} \right\rceil * 5 + \left\lceil \frac{20}{20} \right\rceil * 0.0137 = 10.0281$$

$$w(25) = 0.0144 + \left\lceil \frac{20}{10} \right\rceil * 5 + \left\lceil \frac{20}{20} \right\rceil * 0.0137 = 12.53$$

$$w(40) = 0.0144 + \left\lceil \frac{40}{10} \right\rceil * 5 + \left\lceil \frac{40}{20} \right\rceil * 0.0137 = 20.0418$$

$$w(50) = 0.0144 + \left\lceil \frac{50}{10} \right\rceil * 5 + \left\lceil \frac{50}{20} \right\rceil * 0.0137 = 25.04$$

$\therefore w(50) < 50$, *T2 is schedulable*

Task3

$$w(50) = 0.013 + \left\lceil \frac{50}{10} \right\rceil * 5 + \left\lceil \frac{50}{20} \right\rceil * 0.0137 + \left\lceil \frac{50}{50} \right\rceil * 0.014 + \left\lceil \frac{50}{50} \right\rceil * 0.014 = 25.07$$

$$w(70) = 0.013 + \left\lceil \frac{70}{10} \right\rceil * 5 + \left\lceil \frac{70}{20} \right\rceil * 0.0137 + \left\lceil \frac{70}{50} \right\rceil * 0.014 + \left\lceil \frac{70}{50} \right\rceil * 0.014 = 35.1$$

$$w(90) = 0.013 + \left\lceil \frac{90}{10} \right\rceil * 5 + \left\lceil \frac{90}{20} \right\rceil * 0.0137 + \left\lceil \frac{90}{50} \right\rceil * 0.014 + \left\lceil \frac{90}{50} \right\rceil * 0.014 = 45.125$$

$$w(100) = 0.013 + \left\lceil \frac{100}{10} \right\rceil * 5 + \left\lceil \frac{100}{20} \right\rceil * 0.0137 + \left\lceil \frac{100}{50} \right\rceil * 0.014 + \left\lceil \frac{100}{50} \right\rceil * 0.014 = 50.1375$$

$\therefore - w(100) < 100$, *T3 is schedulable*

Task4

$$w(10) = 0.013 + \left\lceil \frac{10}{10} \right\rceil * 5 = 5.013$$

$$w(20) = 0.013 + \left\lceil \frac{20}{10} \right\rceil * 5 = 10.013$$

$\therefore - w(20) < 20$, *T4 is schedulable*

Task5

$$w(1) = 5 + 0 = 5$$

$$w(2) = 5 + 0 = 5$$

$$w(5) = 5 + 0 = 5$$

$\therefore - w(5) < 5$, *T4 is schedulable*

Task6

$$w(50) = 12 + \left\lceil \frac{50}{10} \right\rceil * 5 + \left\lceil \frac{50}{20} \right\rceil * 0.0137 + \left\lceil \frac{50}{50} \right\rceil * 0.014 + \left\lceil \frac{50}{50} \right\rceil * 0.014 = 25.07$$

$$w(70) = 12 + \left\lceil \frac{70}{10} \right\rceil * 5 + \left\lceil \frac{70}{20} \right\rceil * 0.0137 + \left\lceil \frac{70}{50} \right\rceil * 0.014 + \left\lceil \frac{70}{50} \right\rceil * 0.014 = 35.1$$

$$w(90) = 12 + \left\lceil \frac{90}{10} \right\rceil * 5 + \left\lceil \frac{90}{20} \right\rceil * 0.0137 + \left\lceil \frac{90}{50} \right\rceil * 0.014 + \left\lceil \frac{90}{50} \right\rceil * 0.014 = 5.112$$

$$w(100) = 12 + \left\lceil \frac{100}{10} \right\rceil * 5 + \left\lceil \frac{100}{20} \right\rceil * 0.0137 + \left\lceil \frac{100}{50} \right\rceil * 0.014 + \left\lceil \frac{100}{50} \right\rceil * 0.014 = 62.12$$

$\therefore - w(100) < 100$, *T6 is schedulable*

NB: After we made URM calculation and Time demand calculation on this task , we noticed that the two calculations leading to the same result that the system is schedulable

Simulation on SimSo

Qt Model data

General Scheduler Processors **Tasks**

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
2	TASK T2	Periodic	<input checked="" type="checkbox"/> Yes	0	50	-	50	0.0144	▼	1
3	TASK T3	Periodic	<input checked="" type="checkbox"/> Yes	0	100	-	100	0.01308	▼	1
4	TASK T4	Periodic	<input checked="" type="checkbox"/> Yes	0	20	-	20	0.0246	▼	
5	TASK T5	Periodic	<input checked="" type="checkbox"/> Yes	0	10	-	10	5	▼	
6	TASK T6	Sporadic	<input checked="" type="checkbox"/> Yes	0	100	-	100	12	▼	
1	TASK T1	Periodic	<input checked="" type="checkbox"/> Yes	0	50	-	50	0.0144	▼	

Edit data fields...

Remove selected task(s) Add task Generate Task Set



