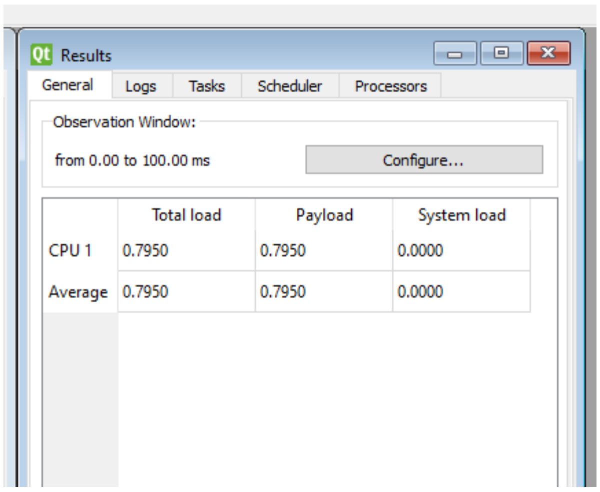
ASSIGNMENT 4

SIMULATION PART

T1(3, 0.5), T2(4, 1.5, 3), T3(7, 1.0, 5) A sporadic job arrives at t=50 having the execution time of 10 and a relative deadline of 30.
Compare EDF with RM

EDF:

- What is the minimum/maximum/average response time of all tasks?



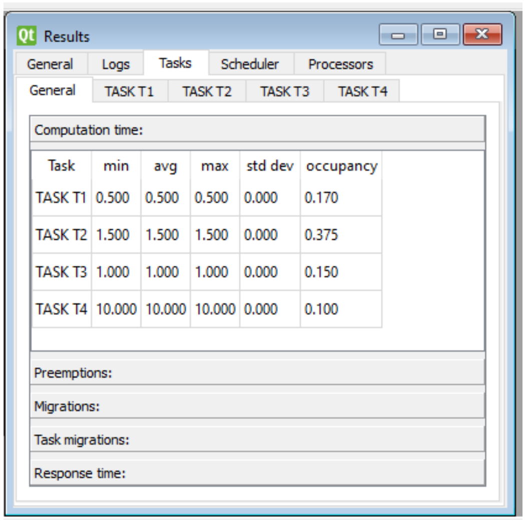
Results

General | Logs | Tasks | Scheduler | Processors

Observation Window:

from 0.00 to 100.00 ms — Configure...

	Total load	Payload	System load
CPU 1	0.7950	0.7950	0.0000
Average	0.7950	0.7950	0.0000



Results

General | Logs | Tasks | Scheduler | Processors

General | TASK T1 | TASK T2 | TASK T3 | TASK T4

Computation time:

Task	min	avg	max	std dev	occupancy
TASK T1	0.500	0.500	0.500	0.000	0.170
TASK T2	1.500	1.500	1.500	0.000	0.375
TASK T3	1.000	1.000	1.000	0.000	0.150
TASK T4	10.000	10.000	10.000	0.000	0.100

Preemptions:

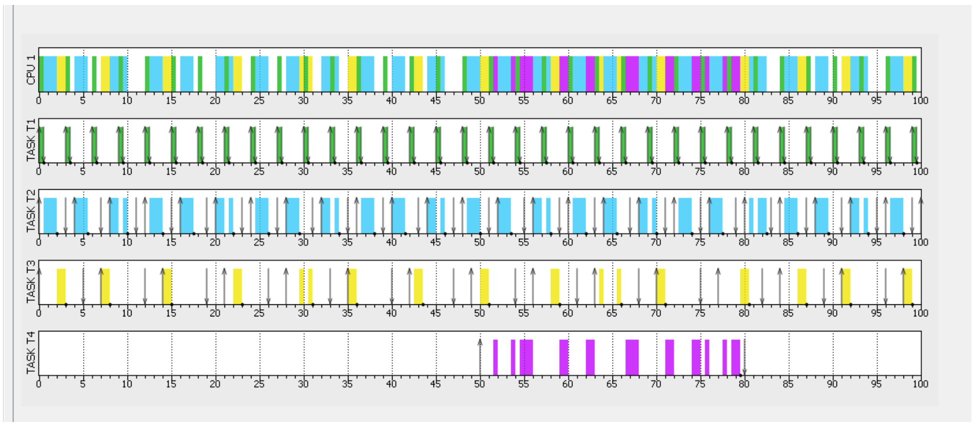
Migrations:

Task migrations:

Response time:

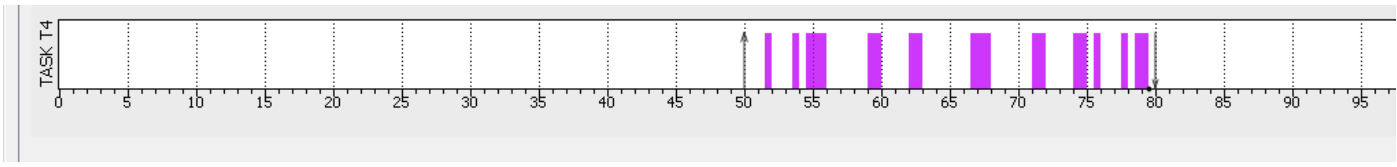
- Is any task missing the deadline? Which task? Where?

no

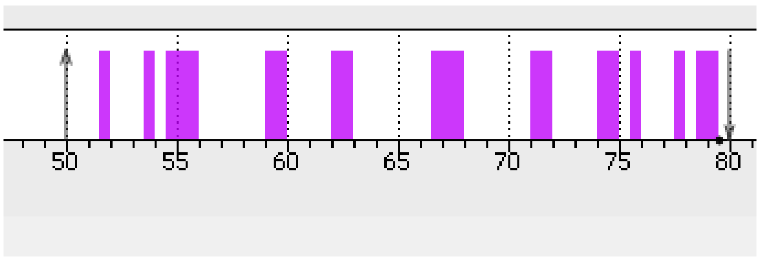


- Is the sporadic job meeting its deadline?

yes



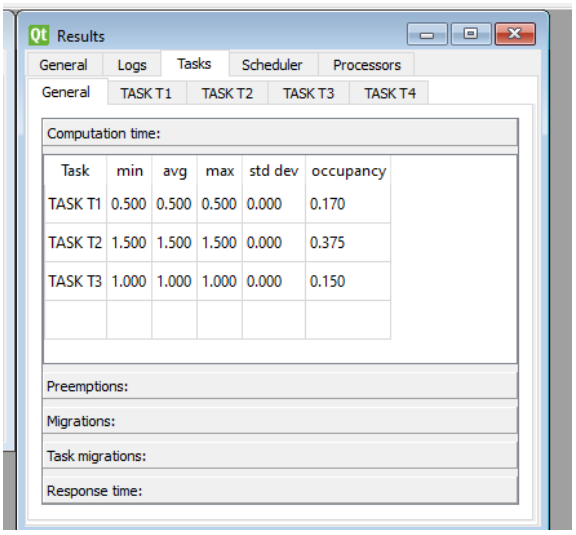
- What is the response time for the sporadic job?



79-51=28 msec

RM

- What is the minimum/maximum/average response time of all tasks?



Results

General | Logs | Tasks | Scheduler | Processors

General | TASK T1 | TASK T2 | TASK T3 | TASK T4

Computation time:

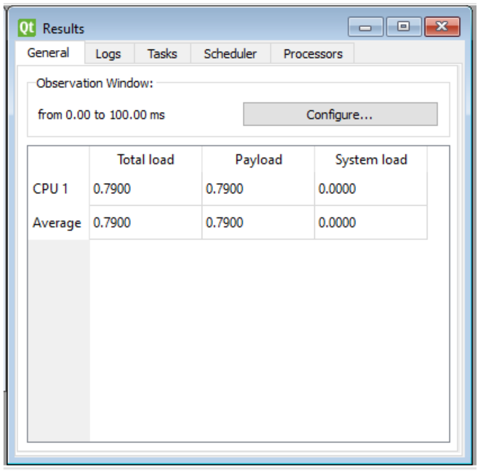
Task	min	avg	max	std dev	occupancy
TASK T1	0.500	0.500	0.500	0.000	0.170
TASK T2	1.500	1.500	1.500	0.000	0.375
TASK T3	1.000	1.000	1.000	0.000	0.150

Preemptions:

Migrations:

Task migrations:

Response time:



Results

General | Logs | Tasks | Scheduler | Processors

Observation Window:

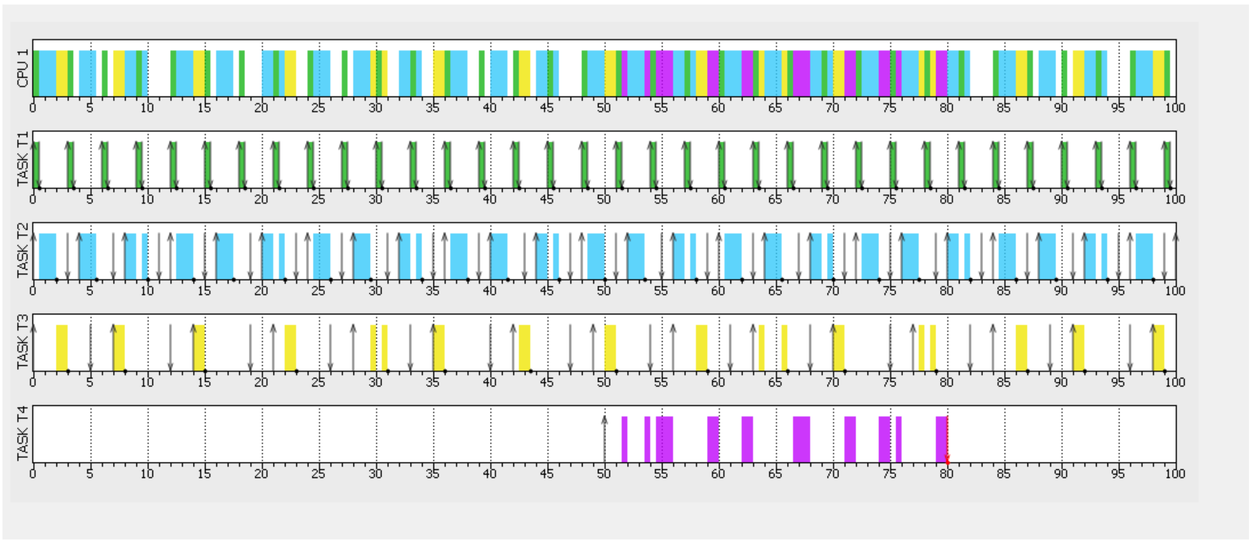
from 0.00 to 100.00 ms

Configure...

	Total load	Payload	System load
CPU 1	0.7900	0.7900	0.0000
Average	0.7900	0.7900	0.0000

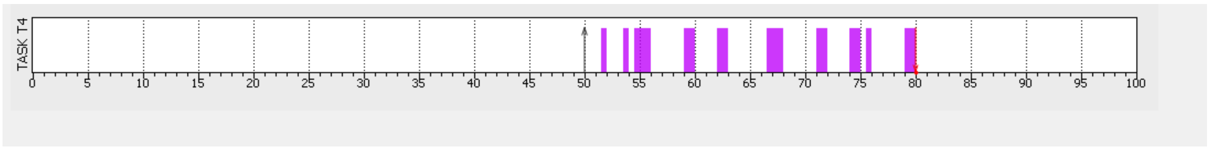
- Is any task missing the deadline? Which task? Where?

T3 at 80 msec

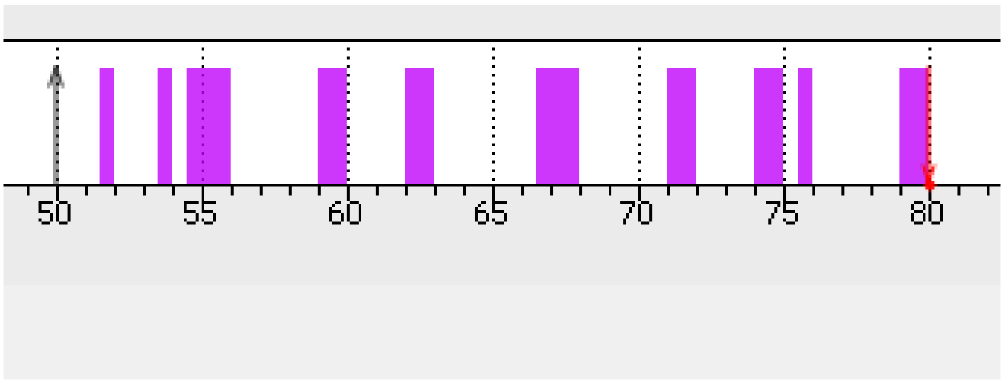


- Is the sporadic job meeting its deadline?

YES



- What is the response time for the sporadic job?



80-51=29 msec

- Which scheduler is better in this example; EDF or RM?

EDF

PROGRAMMING PART

Here create a task "matrixtask" containing the functionality given in Assignment 2.

(Copy the C-code from matrixtask in Assignment 2)

-Add a software timer in main() to trigger a software interrupt every 5 seconds.
(Documentation found Here.)

-Define a Timer callback function outside main() with the following

```
functionality:TimerHandle_t t__1;

    t__1=xTimerCreate("Timer1", pdMS_TO_TICKS(5000), pdTRUE, ( void * ) 0,
vTimerCallback );
    xTimerStart(t__1,0);
```

```
/* A variable to hold a count of the number of times the timer expires. */
long lExpireCounters = 0;
void vTimerCallback(TimerHandle_t pxTimer)
{
    printf("Timer callback!\n");
    xTaskCreate((pdTASK_CODE)aperiodic_task, (signed char *)"Aperiodic",
configMINIMAL_STACK_SIZE, NULL, 2, &aperiodic_handle);
    long lArrayIndex;
    const long xMaxExpiryCountBeforeStopping = 10;
    /* Optionally do something if the pxTimer parameter is NULL. */
    configASSERT(pxTimer);
    /* Increment the number of times that pxTimer has expired. */
    lExpireCounters += 1;
    /* If the timer has expired 10 times then stop it from running. */
    if (lExpireCounters == xMaxExpiryCountBeforeStopping) {
        /* Do not use a block time if calling a timer API function from a
        timer callback function, as doing so could cause a deadlock! */
        xTimerStop(pxTimer, 0);
    }}
```

-Create an aperiodic task using the following functionality:

```
sstatic int ap_period = 0;
/* A variable to hold a count of the number of times the timer expires. */
long lExpireCounters = 0;
xTaskHandle aperiodic_handle;
static void aperiodic_task() {
    ap_running = TRUE;
    printf("Aperiodic task started!\n");
    fflush(stdout);
    long i;
    for (i = 0; i < 1000000000; i++)
        ; //Dummy workload
    printf("Aperiodic task done ,%d!\n",ap_period*portTICK_PERIOD_MS);
    fflush(stdout);
    vTaskDelete(aperiodic_handle);
    ap_running = FALSE;
}
```

- Is the system fast enough to handle all aperiodic tasks? Why?

NO because both tasks take long time with same piorty

- If not, solve this problem without alter the functionality of any task

Raise aperiodic task to 3

```
xTaskCreate((pdTASK_CODE) aperiodic_task, (signed char*) "Aperiodic",
    configMINIMAL_STACK_SIZE, NULL, 3, &aperiodic_handle);
```

- What is the response time of the aperiodic task?

17219 ms = 17.219 sec roughly at worst case and most cases between 2 seconds and 7 seconds

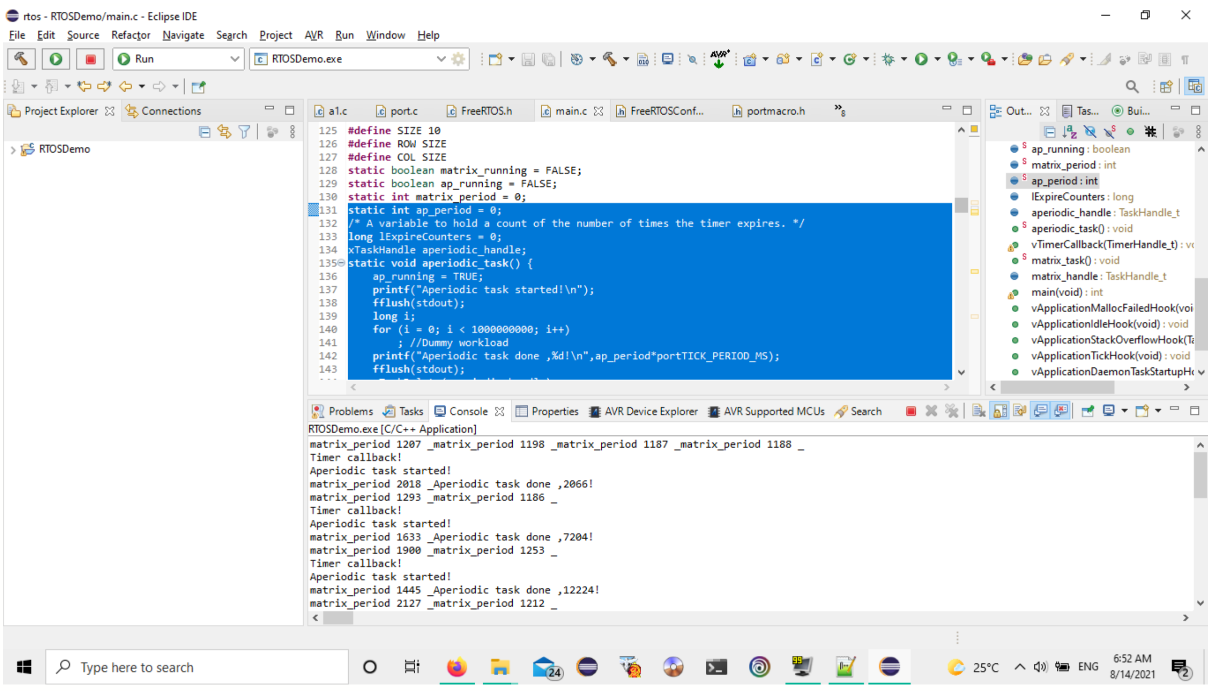
	response
1	2066
2	7204
3	2333
4	1224
5	1220
6	1229
	44268
MEAN	MEDIAN
8506.285714	2066

```
void vApplicationTickHook(void) {

    if (matrix_running == TRUE) {
       matrix_period++;
    }

    if (ap_running == TRUE) {
       ap_period++;     }}
```

- Provide a screenshot of the running system



```
#define SIZE 10
#define ROW SIZE
#define COL SIZE
static boolean matrix_running = FALSE;
static boolean ap_running = FALSE;
static int matrix_period = 0;
static int ap_period = 0;
/* A variable to hold a count of the number of times the timer expires. */
long lExpireCounters = 0;
xTaskHandle aperiodic_handle;
static void aperiodic_task() {
    ap_running = TRUE;
    printf("Aperiodic task started!\n");
    fflush(stdout);
    long i;
    for (i = 0; i < 1000000000; i++)
        ; //Dummy workload
    printf("Aperiodic task done ,%d!\n",ap_period*portTICK_PERIOD_MS);
    fflush(stdout);
```

RTOSDemo.exe [C/C++ Application]

```
matrix_period 1207 _matrix_period 1198 _matrix_period 1187 _matrix_period 1188 _
Timer callback!
Aperiodic task started!
matrix_period 2018 _Aperiodic task done ,2066!
matrix_period 1293 _matrix_period 1186 _
Timer callback!
Aperiodic task started!
matrix_period 1633 _Aperiodic task done ,7204!
matrix_period 1900 _matrix_period 1253 _
Timer callback!
Aperiodic task started!
matrix_period 1445 _Aperiodic task done ,12224!
matrix_period 2127 _matrix_period 1212 _
```