Coursera's
Development of Real-Time Systems
Peer-graded Assignment: Assignment 4

Requirement

1- Simulation assignment

Consider the tasks T1(3, 0.5), T2(4, 1.5, 3), T3(7, 1.0, 5) and the EDF scheduler.

A sporadic job arrives at t=50 having the execution time of 10 and a relative deadline of 30.

Create the sporadic task in SimSo by selecting: "generate task set" and then list of act.

Dates to the release time

-Use SimSo to schedule the task set and provide a report answering the following questions:

- What is the minimum/maximum/average response time of all tasks?
- Is any task missing the deadline? Which task? Where?
- Is the sporadic job meeting its deadline?
- What is the response time for the sporadic job?

Consider the tasks T1(3, 0.5), T2(4, 1.5, 3), T3(7, 1.0, 5) and the RM scheduler.

A sporadic job arrives at t=50 having the execution time of 10 and a relative deadline of 30.

Create the sporadic task in SimSo by selecting: "generate task set" and then list of act.

Dates to the release time

-Use SimSo to schedule the task set and provide a report answering the following questions:

- What is the minimum/maximum/average response time of all tasks?
- Is any task missing the deadline? Which task? Where?
- Is the sporadic job meeting its deadline?
- What is the response time for the sporadic job?
- Which scheduler is better is better in this example; EDF or RM?

Task1- EDF Scheduler

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)
1	TASK T1	Periodic	☑ Yes	0.0	3.0	-	3.0	0.5
2	TASK T2	Periodic	☑ Yes	0.0	4.0	-	3.0	1.5
3	TASK T3	Periodic	☑ Yes	0.0	7.0	-	5.0	1.0
4	TASK T4	Sporadic	☑ Yes	-	-	50.0	30	10

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

Response time:

Task	min	avg	max	std dev
TASK T1	0.500	0.676	1.500	0.294
TASK T2	1.500	1.700	2.000	0.245
TASK T3	1.000	1.967	3.500	0.921
TASK T4	29.000	29.000	29.000	0.000

- **Is any task missing the deadline? Which task? Where?**
    - No deadline is missed
- **Is the sporadic job meeting it's deadline?**
    - Yes
- **What is the response time for the sporadic job ?**
    - Min = 29 & avg =29 & max =29

Task2 – RM Scheduler

General | Scheduler | Processors | Tasks

Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
TASK T1	Periodic	☑ Yes	0	3	-	3	0.5
TASK T2	Periodic	☑ Yes	0	4	-	3	1.5
TASK T3	Periodic	☑ Yes	0	7	-	5	1
TASK T4	Sporadic	☑ Yes	-	-	50.0	30	10

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

Response time:

Task	min	avg	max	std dev
TASK T1	0.500	0.500	0.500	0.000
TASK T2	1.500	1.840	2.000	0.233
TASK T3	1.000	1.900	3.000	0.860
TASK T4				

- **Is any task missing the deadline? Which task? Where?**
  - Sporadic job miss deadline at t = 80
- **Is the sporadic job meeting its deadline?**
  - No it isn't
- **What is the response time for the sporadic job?**
  - T=30
- **Which scheduler is better is better in this example; EDF or RM?**
  - EDF as non of the task missed the deadline

1- Programming Assignment

In this programming assignment, you will handle aperiodic jobs.

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

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

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

```
/* 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:

```
static void aperiodic_task()
{
  printf("Aperiodic task started!\n");
  fflush(stdout);
  long i;
  for (i = 0; i<1000000000; i++); //Dummy workload
  printf("Aperiodic task done!\n");
  fflush(stdout);
  vTaskDelete(aperiodic_handle);
}

```

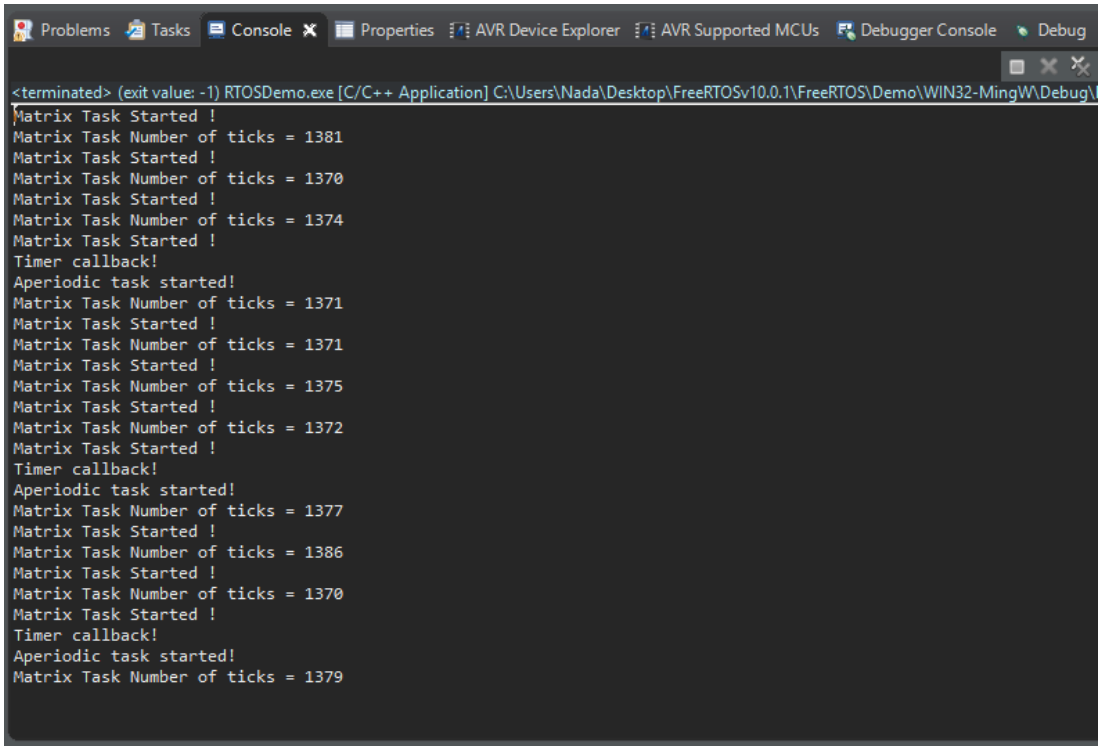
The following questions should be solved with programming and the questions should be answered in a report:

- Is the system fast enough to handle all aperiodic tasks? Why?
- If not, solve this problem without alter the functionality of any task
- What is the response time of the aperiodic task?
- Provide a screenshot of the running system

Task Solution :-

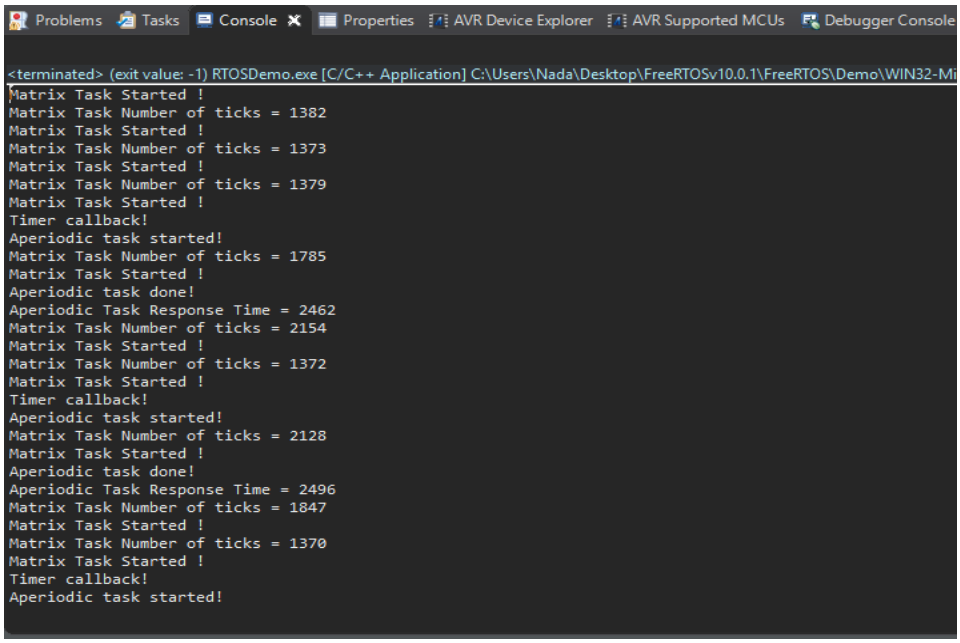
- **Is the system fast enough to handle all aperiodic tasks? Why?**

    - No , as Matrix Task consume most of cpu time as it has the highest priority
    -



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

    - By increasing of aperiodic Task priority to be equal to matrix task priority
    -



- **What is the response time of the aperiodic task?** 2.5 Sec
- **Provide a screenshot of the running system**