Assignment 4

Coursera:"Development of real time systems" by EIT

Simulation assignment:

Task set 1:

For the first set task we have:

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

  The average and max response time are the same for task set 1, as can be aprecciated in *fig.1*

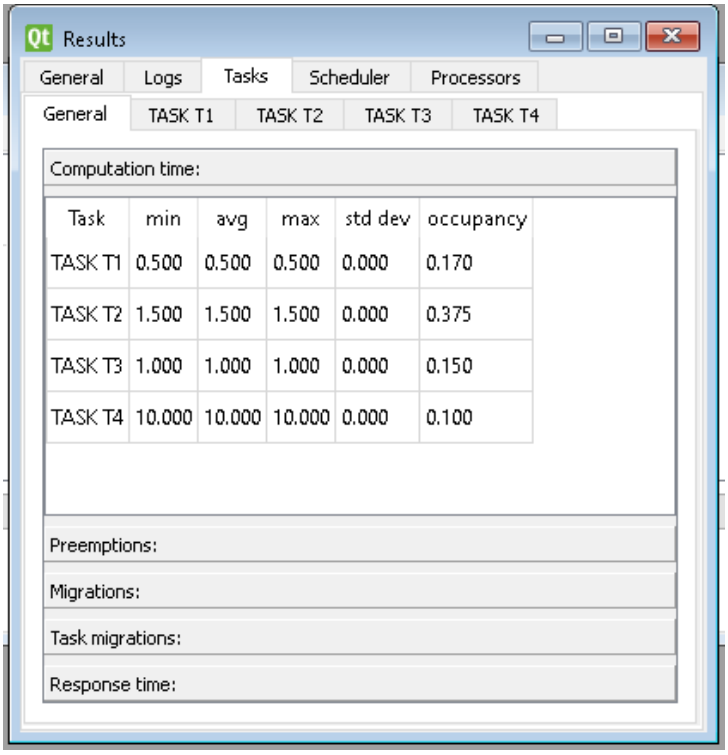

Fig 1: min/avg/max response time for EDF scheduler

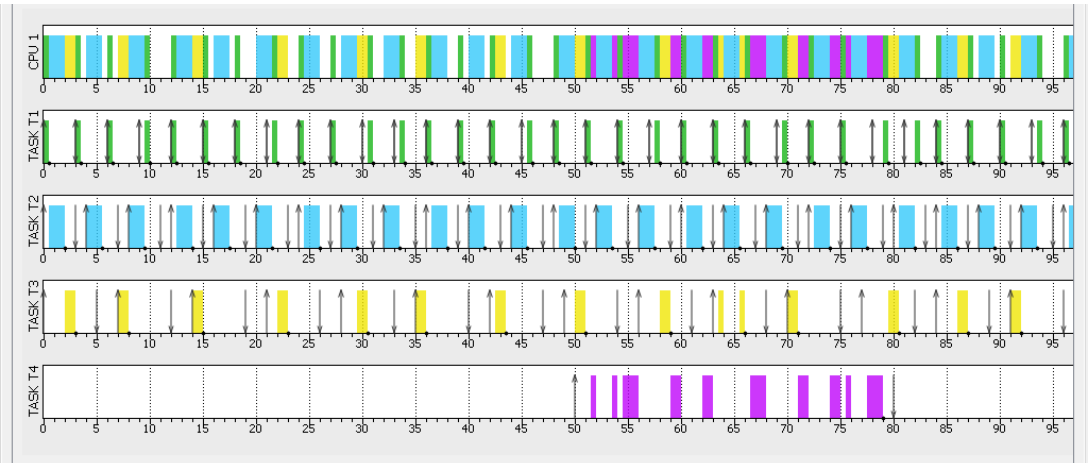

Fig 2: Time scheduling for set task 1

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

  None of the tasks misses a deadline. As you can see in *fig.2*

- ***Is the sporadic job meeting its deadline?***

  Yes, its execution finished at 79ms while its deadline was 80ms. As you can see in fig.3:

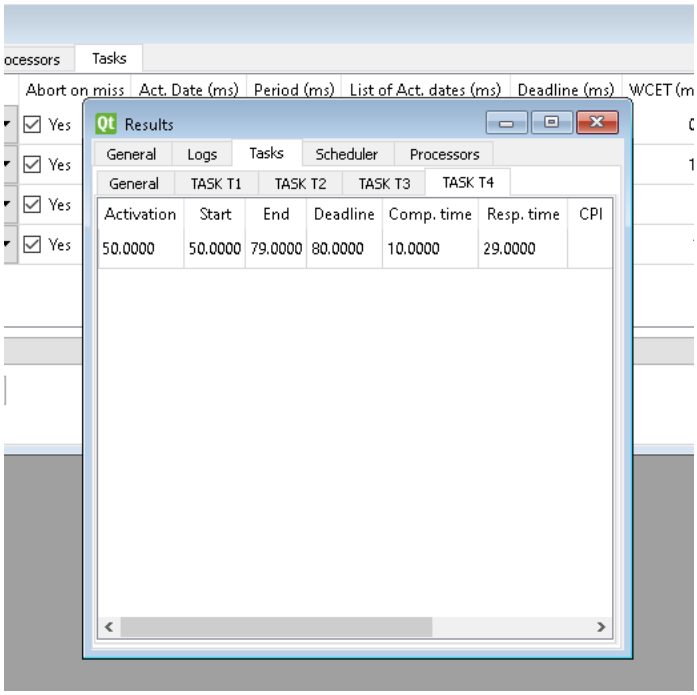

Fig 3: Results for the sporadic job

- **What is the response time for the sporadic job?**

  As you can see in fig.3, its response time is 29ms.

Task set 2:

For the second set task we have:

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

  The average and max response time are the same for task set 1, as can be aprecciated in fig.4

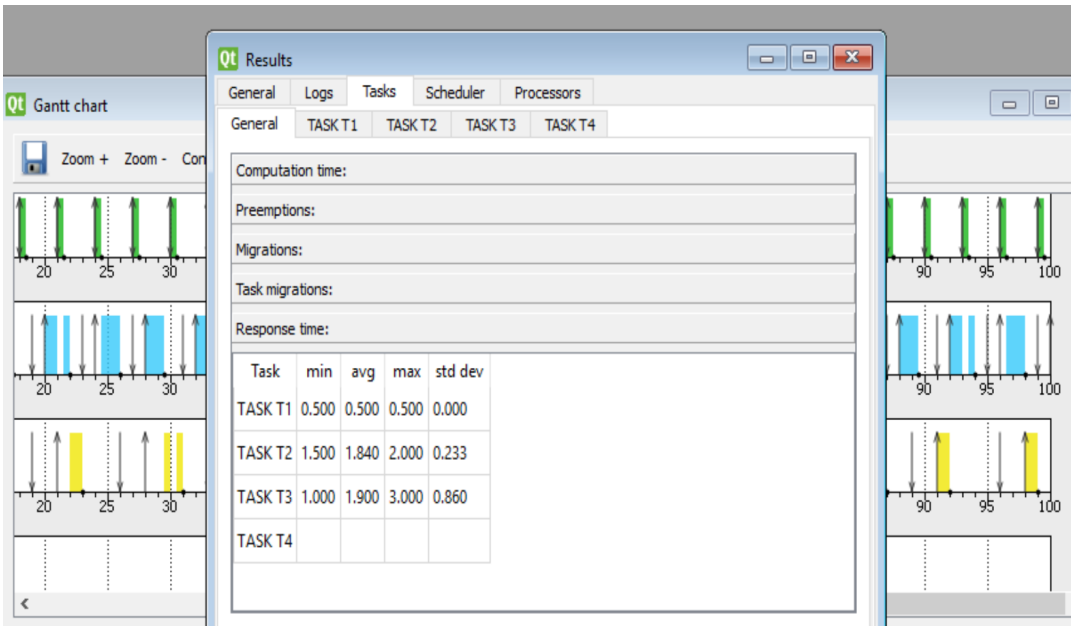

Fig 4: min/avg/max response time for EDF scheduler

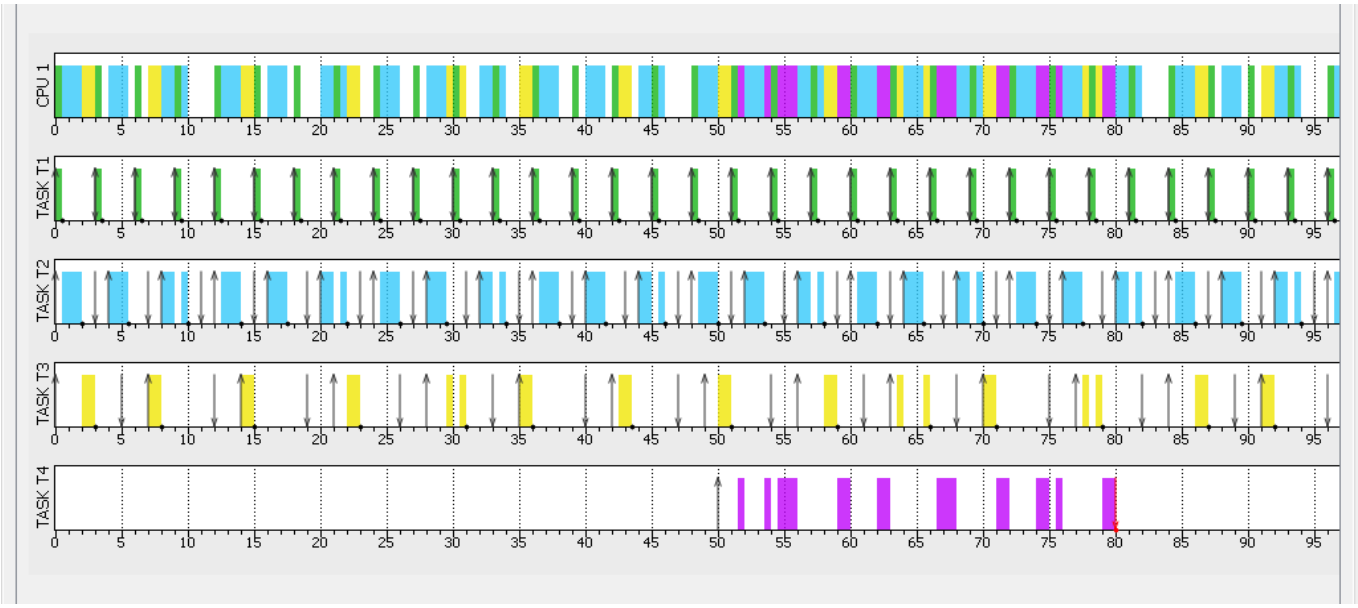

Fig 5: Time scheduling for EDF scheduler

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

  The sporadic job finishes at the exact time of the deadline, so if we are conservative we can say that its misses the deadline. As you can see in fig.5 & fig.6.

- ***Is the sporadic job meeting its deadline?***

  As i previously said, if *we are conservative we can say that its misses the deadline.* As you can see in *fig.5 & fig.6:*

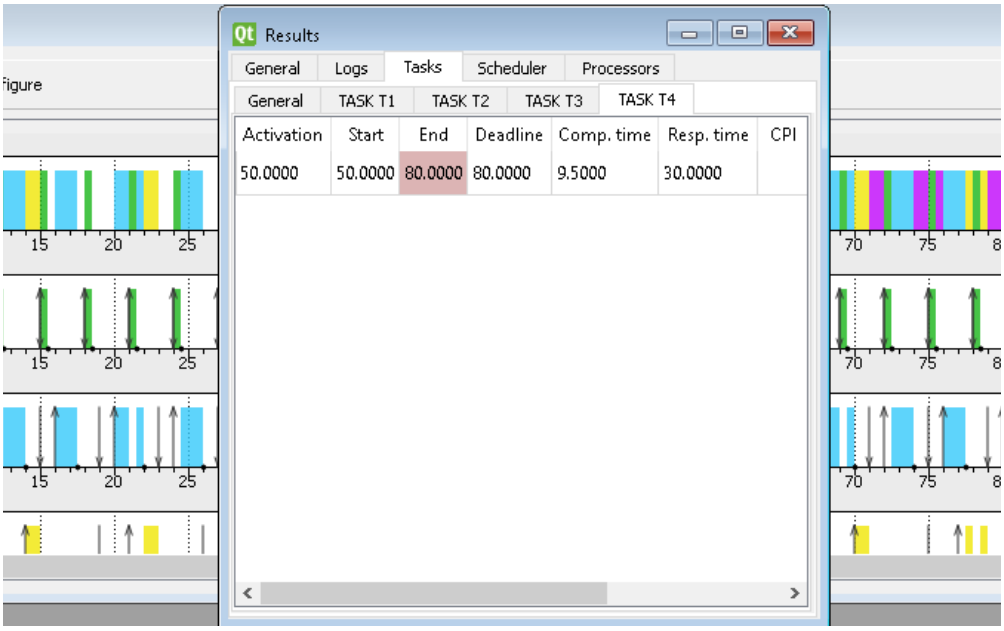

Fig 6: Results for the sporadic job

- **What is the response time for the sporadic job?**

  As you can see in fig.6, its response time is 30ms.

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

  In this example, the EDF scheduler is better, as can meet the deadline for the sporadic job without compromising the other jobs.

Programming assignment:

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

  No, because the matrix_task monopolizes the CPU resources and the aperiodic task doesn't meet its deadline every time. As you can appreciate in fig.7:

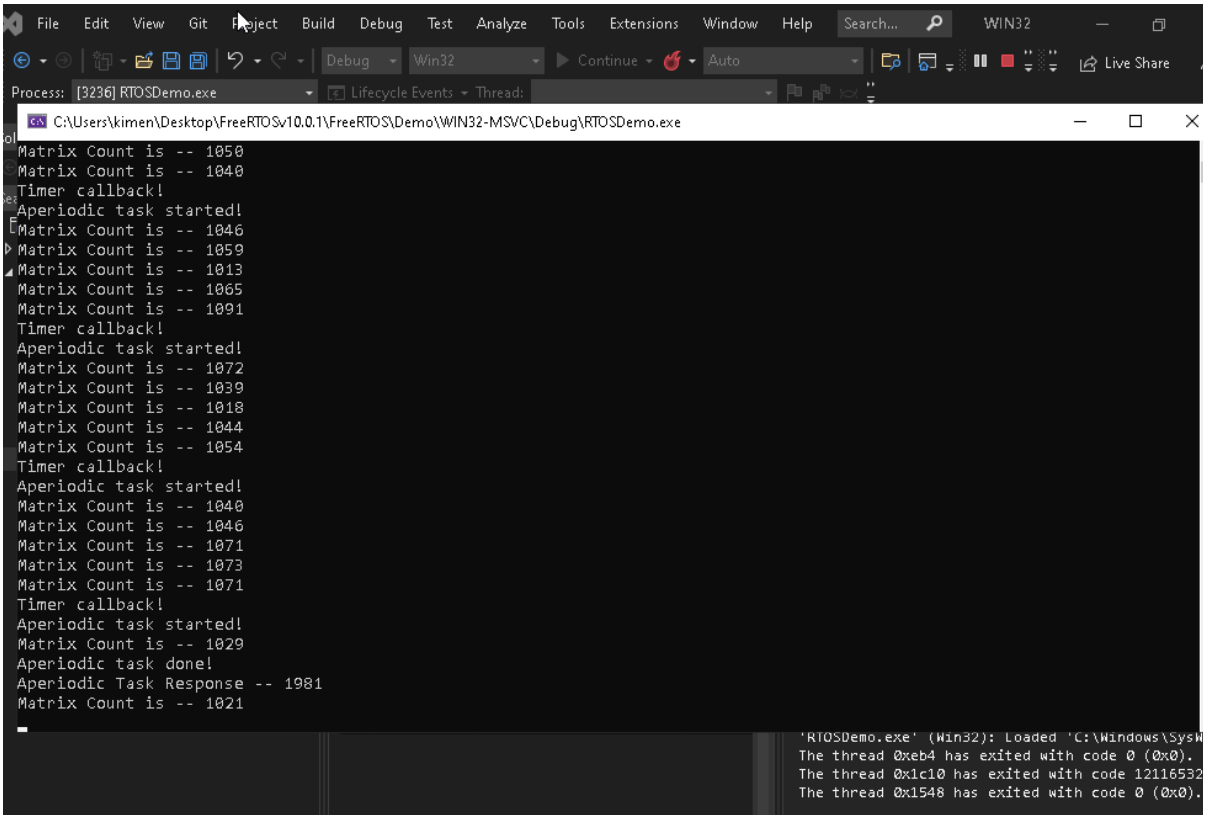

Fig 7: Screenshot of the running application

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

  We can change the priority of the aperiodic task, that would be the equivalent in the deferrable server to augmenting it's resources. The results are shown in fig.8

- **What is the response time of the aperiodic task?**

  It's ~1800 as you can see in fig.8

- **Provide a screenshot of the running system**

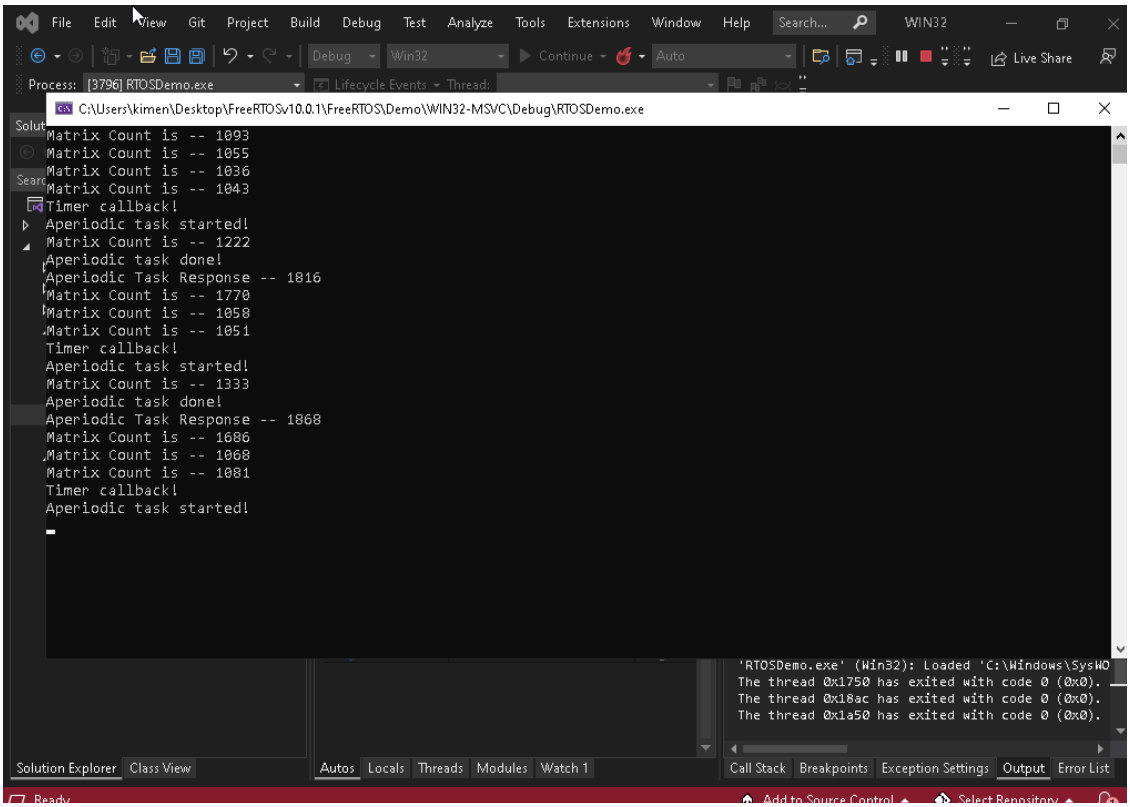

Fig 8: Screenshot of the running application with modified priority