ASU_CSE345_FreeRTOS **Project 2** (as an alternative assessment)

The goal of the class project is to give you more experience in Real-Time Embedded System Design. The design experience will utilize one of the prominent processor technologies, namely, the application specific instruction set processors, or microcontrollers. The ~~TivaC board will be the hardware platform for the project implementation and the~~ FreeRTOS will be used to manage the concurrent tasks.

This course project2 will focus on the software implementation of the FreeRTOS. You will examine them closely and modify/create a specific feature. Then you will provide detailed documentation on how to use this new feature. A brief research about the theoretical part of the output should be included. Further details below.

Desired new features:

Real Time Task admittance and scheduling via an updated rate monotonic task scheduling, with adjustable priorities.

Let N be the maximum number of tasks to arrive at a Real-time system.

1) It is desired to design a task admittance function, with up to n periodic and concurrent tasks. Each task will have three parameters: Ta, Tp, Tc for arrival, period and computation times. In the task admittance functions, the number of tasks, and each of the task parameters will be set randomly.

   n: current number of tasks between 2 to N

   tst: Processor slice time, which will be the unit of measurements for the following parameters

   Ta(i): Arrival time of task i; integer random from 0 to latest_arrival_time

   Tc(i): Computation time of task I, in multiples of tst. Tc(i) should be integer random number from 1 to maximum_computation_time

   Tp(i): Period of task i, in multiples of tst,

      Two options:

Safe mode: random, to ensure schedulablity, set to from 3xTc(i) to maximum_period_multipler x Tc(i)

No Guarantee mode: random, from 3xTc(i) to 10xTc(i)

Example data set

ID)	A	P	C	CPU%	rate=1/P	--->priority
T1)	3	11	2	2/11=0.181	1/11	5
T2)	9	20	1	1/20=0.05	1/20	3
T3)	12	32	5	5/32=0.156	1/32	1
T4)	6	27	4	4/27=0.148	1/27	2
T5)	13	13	1	1/13=0.077	1/13	4

2) Calculate total CPU utilization, and perform a Schedulablity test.
For the example above, Ucpu=1/13 +4/27+5/32+1/20+2/11 < 0.7
So, schedulablity check is ok

3) If admitted tasks are schedulable, set their priorities based on their rates. Sort tasks based on their periods, then give the longest period priority 1, and increase priorities for each next higher level by 1. Thus, the highest priority should not be bigger than n in any case. If created tasks have the same period, then assign them the same priority. Use static data structures to store task parameters, and perform the sorting algorithm of your choice to order task priorities. Call the FreeRTOS scheduler after creating the tasks with their computed priorities, (use xTaskCreate() and adjusted periods (use exact Task Delay function call in the task functions)

4) In the latter part of this project, our real-time system might allow more dynamics. So, you will need to use dynamic data structures instead of static. You will need to assume that tasks can arrive after time=0, and not ignore their random arrival time. Finally, you will also need to consider that tasks can be deleted (randomly) after scheduler has been started. So, a scheduler suspension and restart will be needed during the recalculation of task priorities in such cases.

Demos: Make text output on the debug terminal clear enough to demonstrate the proper working of your scheduled tasks (show CPU slice allocation to which task ID, and textual progress of time line). Record a video demonstrating the working of your project and submit online over the LMS at

least two hours before the posted deadline. ~~send to the TA's email (mostafa.a.arafa91@asu.edu.eg).~~

Apply best design models to represent your design. Make sure the code fits the limitations of the IDE used ~~to program the TivaC board~~. Use µVision IDE, CodeComposer or any other IDE to test/debug the code ~~into the TivaC board~~.

Test parameters (use configuration compile time parameter)):

N=5 (n is from 2 to N)

tst=1

latest_arrival_time=15

maximum_computation_time=8

maximum_period_multipler= 17

5) Research the theorical upper bound of a single CPU utilization when scheduling multiple concurrent tasks via the Rate Monotonic Scheduling Algorithm. Can a single CPU be fully utilized to schedule multiple tasks via this algorithm?

Size and formats: this section should be within the range of 1500 to 2000 words (A4 size, font size 12, 1.5 line spacing).

Notes: You should include an introduction, body, summary, then references in this brief research section. Please make sure to provide proper references of all cited research. Must have at least 3 references and, at least, one of them must be a recent one (after 2010). Check the grammar and the spelling before submission. Report might undergo similarity checks for allocating plagiarism cases.

Schedule:

04/8/2020: Project Introduction, and team formation. Team leader: send the TA an email including names of your team members.

05/31/2020: Full team submission, as detailed below.

05/31/2020: Individual report submission, as detailed below.

General Requirements and Constraints

- Work in groups of **three to five** students each.
- You select your group members. Each team should elect a team leader. Notify the TA via email with team members and team leader, no later than a week after project Introduction.
- Each member must design part of the project and must write his/her **OWN part of** the report. Each member must **clearly** show his/her individual contribution to the project.
- Essential to help each other in the same team, but not across teams!
- All diagrams must be done on the computer; Use Program State Machine Model to layout your system software components and their dependencies.

The Full team submission report should be divided to two main parts:

System Design Part

The system design part should have the following contents in order:

- Title Page with project title, ASU class number and title, date of submission, team name, and members.
- Project Specifications and Description, expanded and clarified from these given specs.
- Design Choices, including any choices you made that were not given in the specification or any functionality you will expand.
- Detailed block diagram (graphic) and diagram descriptions.
- Team member responsibilities for the project.
- Plan of timeline: what is expected to get done, and at when. Make a proposed timeline.

Technical Details Part

This part includes following contents in order:

- Design details, operation manual. What should we do to test your prototype? This can be documenting what you present in your demonstration video.

- Code listings. What functions you used, and what modifications you made. What new functions you designed.
- **Lessons learned (important)**
- Problems faced (if any)

Each team will email to the TA a **zipped file** which contains all electronic documents related to this project (compressed software project files, final report document) plus a video URL of your demonstration. The video should have audio narration (explaining all test cases of your system).

In addition, each **team member** will submit a separate short report to the TA's email which includes following contents in order:

- Title Page (team name, your name, class number and name, submission date)
- Your **individual contribution**, Be specific.
- **Evaluate your team members' work.**
- Provide any suggestions or comments on the project. There is always something that can be done better.

Grade = system design part (20%) + technical details part (40%) + report organization, and clarity (10%) + research section (20%) + teamwork (10%).

If some team members are left not performing, this will usually affect everyone else's performance. So, try to resolve such problems yourselves, early on, and don't ignore them. But if needed, you will have to report it to TAs and they will report to the course professor.

Deadline will be enforced by the LMS. You will not be able to submit past deadline. Note the time zone difference. Please submit at least two hours before the deadline.

With my best wishes,

Prof. Dr. Omer Alkelany