ECEE 5623, Real-Time Systems:

Exercise #6 – Real-Time Software Systems

Final Extended Lab (Individual or Group Work)

DUE: As Indicated on D2L by midnight and Syllabus

Please thoroughly read Chapters 12, 13 & 14 in RTECS with Linux and RTOS

Please see example code provided - Linux, FreeRTOS, VxWorks, Zephyr

This lab is written to be completed with embedded Linux running on the DE1-SoC, Jetson or Beagle board(s), but you are welcome to complete it using FreeRTOS, VxWorks or Zephyr as an option. Note that you will either have to adapt the example code or use equivalent examples from the links above for RT-Clock, use of tasks in place of pthreads, and the appropriate board to boot FreeRTOS, VxWorks, or Zephyr as described on the course main web page.

This group exercise (teams of 2, 3 or 5) asks you to explore real-time software system analysis and design concepts by applying RM theory we have learned as well as software engineering skills you have learned as a computer and/or software engineering student. The goal is for your team to design an aspect of a real-time system that includes 2+ real-time services and requires RM analysis along with priority-preemptive (POSIX SCHED_FIFO) or RTOS priority scheduling to meet required periodic deadlines.

<u>Exercise #6 Requirements and High Level Design</u>:

1) [10 points] Provide all major functional capability requirements for your real-time software system [not just for the proof-of-concept aspect to be demonstrated and analyzed, but the whole design concept envisioned]. You should have at least 5 major requirements or more.

2) [10 points] Complete a high-level real-time software system functional description and proposal along with a single page block diagram showing major elements (hardware and software) in your system (example). You must include a single page block diagram, but also back this up with more details in corresponding CFD/DFD, state-machine, ERD and flow-chart diagrams as you see fit (2 more minimum). Please provide this in your report for this assignment.

3) [10 points] Provide all major real-time service requirements with a description of each S_i including C_i, T_i, D_i with a description of how the request frequency and deadline was determined for each as well as how C_i was determined, estimated or measured as WCET for the service.

4) [10 points] For your 2+ services S_i with requirements C_i, T_i, D_i provide Cheddar Simulation and Worst-Case analysis, but double check this with your own Scheduling Point and Completion Test analysis as well as hand-drawn timing (if feasible – i.e. if LCM is not ridiculously long). Provide an estimate of the safety margin your system should have compared to what it must have for feasibility and margin of error [variations seen in C_i to account for expected C_i and WCET].

5) [10 pts] Analyze your system in terms of how well it meets real-time requirements as outlined and used in Lab Exercise #5, found here. Present the results along with a description of how predictable the responses are relative to the request times as well as how constant the request frequency is for your system design.

6) [50 pts] Write a report that includes [16 total pages including cover page, but not appendices or code]. The report should be composed of results from your work in Lab Exercise #5 and work above, but documented in complete form as follows:

[1 page] **Cover Page (list all group members clearly)**

[1 paragraph] **Introduction**

[1 page] **Functional (capability) Requirements**

[1 page] **Real-Time Requirements**

[N pages] **Functional Design Overview and Diagrams**

[N pages] **Real-Time Analysis and Design with Timing Diagrams, both measured and expected based upon theory**

[N pages] **Proof-of-Concept with Example Output and Tests Completed** - **(Must contain final time-lapse MPEGs and verification RAW frames with time-stamps and uname stamp)**

[1 paragraph] **Conclusion**

[1 page] **Formal References (and Attributions to Anyone who helped not on the team)**

[N pages] **Appendices with results, code and supporting material** (to stay in page bounds)

Grading Rubric

[15 points] Functional (capability) requirements for system design:

[2 points] R#1 ____

[2 points] R#2 ____

[2 points] R#3 ____

[2 points] R#4 ____

[2 points] R#5 to N____

[5 pts] Completeness_______________________________________

[15 points] High level system and software design:

[5 points] Block diagram ____

[5 points] Detail diagram #1____

[5 points] Detail diagram #2____

[15 points] Real-Time Services and Requirements:

[5 points] Services and descriptions ___________________________________

[5 points] C_i and WCET specification__________________________________

[5 points] T_i and D_i specification______________________________________

[15 points] Real-Time Services Feasibility, Safety and Margin:

[5 points] Cheddar Worst-Case and Simulation ___________________________

[5 points] Scheduling Point / Completion tests____________________________

[5 points] Safety margin analysis_______________________________________

[15 points] Proof-of-concept code and time-stamp tracing:

[5 points] Complete well-written code ________________________________

[5 points] Time-stamp tracing_______________________________________

[5 points] Test cases shown____

[25 points] Report required elements:

[5 pts] Function requirements documentation______________________________

[5 pts] Real-time requirements documentation______________________________

[5 pts] Design documentation____

[5 pts] RT analysis documentation____

[5 pts] Overall report completeness_______________________________________