ECEN 5623, Real-Time Embedded Systems:

Exercise #2 – Service Scheduling Feasibility

DUE: As Indicated on Canvas.

Please thoroughly read Chapters 3, 4 & 5 in RTECS with Linux and RTOS
Please see example code provided - Linux, FreeRTOS, and VxWorks, and also on Canvas

Exercise #2 Requirements:

1) [5 points] make yourself an account on your Dev Kit.
   a) To do this, use the reset button if the system is locked, use your password to login, and then use "sudo adduser", enter a password, and enter user information as you see fit. Add your new user account as a "sudoer" using "visudo" right below root with the same privileges (if you need help with "vi", here's a quick reference or reference card– use arrows to position cursor, below root hit Esc, "i" for insert, type username and privileges as above, and when done, Esc, ":", "wq"). The old unix vi editor was one of the first full-screen visual editors – it still has the advantage of being found on virtually any Unix system in existence, but is otherwise cryptic – along with Emacs it is still widely used in IT, by developers and systems engineers, so it's good to know the basics. If you really don't like vi or Emacs, your next best bet is "nano" for Unix systems (you can normally do "sudo apt-get install nano" or any alternative editor you like best). You are welcome to use whatever editor works best for you in this class. Finally, you'll want an SSH tool (e.g. MobaXterm for Windows) and you'll want to enable SSH via the GUI or with headless methods. Do a quick "sudo whoami" to demonstrate success.
   b) Logout of Linux and test your login, then logout. Use Alt+Print-Screen to capture your desktop and save as proof you set up your account. Note that you can always get a terminal with Ctrl+Alt+t key combination. If you don't like the desktop, you can try "GNOME Flashback" and please play around with customizing your account as you wish. Show evidence that you have created a custom login with screenshots or photos from your phone.
   c) Make sure you can access graphical tools with MobaXterm or VNC and show that tools you may need in the future work, including a tool to display graphics in PPM or PGM format. Overall, make sure you are comfortable with development, debug, and compiler tools either native or cross-development and document and demonstrate that you know them.

2) [10 points] Read the paper "Architecture of the Space Shuttle Primary Avionics Software System" [also available on Canvas, "shuttle_paper.pdf"], by Gene Carlow.
   a) Provide an explanation and critique of the frequency executive architecture.

   b) What advantages and disadvantages does the frequency executive have compared to the real-time threading and tasking implementation methods for real-time software systems? Please be specific about the advantages and disadvantages and provide at least 3 advantages as well as 3 disadvantages.

3) [5 points] Read the paper "Building Safety-Critical Real-Time Systems with Reuseable Cyclic Executives", available from http://dx.doi.org/10.1016/S0967-0661(97)00088-9. In other embedded systems classes you built ISR (Interrupt Service Routine) processing software and polling/control loops to control for example stepper motors – describe the concept of the Cyclic Executive and how this compares to the Linux POSIX RT threading and RTOS approaches we have discussed.

4) [50 points] CUSTOM FEASIBLITY TEST CODE. Download Feasibility example code (or get it from Canvas) and build it on a Jetson, Raspberry Pi, DE1-SoC or TIVA or Virtual Box and execute the code.
   a) Compare the feasibility tests provided by the example code to analysis using Cheddar for the first 5 examples (0-4).
   b) Now, implement the remaining examples [5 more, 5-9] that we reviewed in class (found here, and on Canvas) by modifying the example code to include the other examples. Complete analysis for all three policies using Cheddar (RM, EDF, LLF) and by adding EDF and LLF feasibility tests to the code, except for example 6, which should use RM and DM. In cases where RM fails, but EDF or LLF succeeds, explain why. Cheddar uses both service simulations over the LCM of the periods as well as feasibility analysis based on the RM LUB and scheduling-point/completion-test algorithms, referred to as "Worst Case Analysis".
   c) Does your modified Feasibility code agree with Cheddar analysis in all 5 additional cases? Why or why not?

5) [30 points] Read Chapter 3 of the textbook.
   a) Briefly describe and provide 3 constraints that are made on the RM LUB derivation and 3 assumptions as documented in the Liu and Layland paper and in Chapter 3 of the text. Describe whether you think each is reasonable for actual practice or whether you think each is only applicable to and idealized model of practice.
   b) Finally, list 3 key derivation steps in the RM LUB derivation that you either do not understand or that you would consider "tricky" math. Attempt to describe the rationale for those steps as best you can do based upon reading in Chapter 3 of the text.

Overall, provide a well-documented professional report of your findings, output, and tests so that it is easy for a colleague (or instructor) to understand what you've done. Include any C/C++ source code you write (or modify) and Makefiles needed to build your code. I will look at your report first, so it must be well written and clearly address each problem providing clear and concise responses to receive credit.

Note: Linux manual pages can be found for all system calls (e.g. fork()) on the web at http://linux.die.net/man/ - e.g. http://linux.die.net/man/2/fork

In this class, you'll be expected to consult the Linux manual pages and to do some reading and research on your own, so practice this in each lab and try to answer as many of your own questions as possible, but do come to office hours and ask for help if you get stuck.

Upload all code and your report completed using MS Word or as a PDF to Canvas and include all source code (ideally example output should be integrated into the report directly, but if not, clearly label in the report and by filename if test and example output is not pasted directly into the report). ***Your code must include a Makefile so I can build your solution on an embedded Linux system (R-Pi or Jetson). Please zip or tar.gz your solution with your first and last name embedded in the directory name, and/or provide a GitHub public or private repository link. Note that I may ask you or SA graders may ask you to walk-through and explain your code. Any code that you present as your own that is "re-used" and not cited with the original source is plagiarism. So, be sure to cite code you did not author and be sure you can explain it in good detail if you do re-use, you must provide a proper citation and prove that you understand the code you are using.***

Grading Rubric

1) [5 points] Create account on your target board.

Section Evaluation	Points Possible	Score	Comments
Account Creation and Login	2		
Desktop or Remote Terminal Access	2		
Development Environment and Graphics Tools	1		
Total			

2) [10 points] Shuttle PASS paper review:

Section Evaluation	Points Possible	Score	Comments
Three advantages	3		
Three disadvantages	3		
Overall understanding of paper and key point articulation	4		
Total	10		

3) [5 points] Cyclic Executive paper review:

Section Evaluation	Points Possible	Score	Comments
Overall understanding of paper and key point articulation	3		
Comparison to Linux and RTOS approaches	2		
Total	5		

4) [50 points] Shared CPU system overload – for each provide custom feasibility code, screenshots of code output, including EDF and LLF test output, and Cheddar results screenshots:

Section Evaluation	Points Possible	Score	Comments
Examples 0-4 comparison	5		
EDF, LLF, and RM code	5		
Example #5 code and output for 3 policies	4		
Example #6 code and output for 2 policies (RM and DM)	4		
Example #7 code and output for 3 policies	4		
Example #8 code and output for 3 policies	4		
Example #9 code and output for 3 policies	4		
Example #5 Cheddar confirm	4		
Example #6 Cheddar confirm	4		
Example #7 Cheddar confirm	4		
Example #8 Cheddar confirm	4		
Example #9 Cheddar confirm	4		
Total	50		

Don’t forget to explain why in some cases RM fails, but EDF or LLF succeeds.

5) [30 points] Liu and Layland paper:

Section Evaluation	Points Possible	Score	Comments
Constraint #1, Assumption #1	5		
Constraint #2, Assumption #2	5		
Constraint #3, Assumption #3	5		
key step #1	5		
key step #2	5		
key step #3	5		
Total	30		