Class: Real-Time Embedded Systems
Professor: Prof. Steve Rizor
TAs: Mohit Chaudhari, Krishna Suhagiya, Samiksha Patil
Student: Steve Gillet
Date: January 20, 2025
Assignment: Homework 1

1. Check out or purchase a Raspberry Pi 4 kit, and install the latest Raspberry Pi 64-bit OS on your Raspberry Pi 4 (there are plenty of guides online). Answer this question by pasting the result of the command "cat /proc/cpuinfo—grep Ser; uname -a"

I am using the Jetson Nano Developer Kit for now, though I do have a Raspberry Pi 4 for backup. The Jetson Nano does not have the serial number in the cpuinfo so I improvised the command a little bit.

steve@steve-desktop: $ cat /proc/device-tree/serial-number; echo ; uname -a;
1425222005022
Linux steve-desktop 4.9.253-tegra #1 SMP PREEMPT Sat Feb 19 08:59:22 PST 2022 aarch64 aarch64 aarch64 GNU/Linux

2. Create a github account if you don't already have one. Make a private repository for the course. If you haven't used git before, check out some guides. Install git and gh on your rpi (hint: sudo apt install git gh), generate a personal access token on github, authenticate your rpi, and clone your repo. This will probably require some internet searching to figure out, and you can also ask questions on slack or use an LLM to help you figure out the steps. Answer this question by pasting the result of the command "git remote -v" from inside your local clone of your repository.

steve@steve-desktop: /Desktop/realTimeEmbedded$ git remote -v
origin git@github.com:steveGillet/realTimeEmbedded.git (fetch)
origin git@github.com:steveGillet/realTimeEmbedded.git (push)

3. Choose a partner for your exercises and post your team on Slack. Answer this question by giving your team members as well.

TODO

4. Provide examples of real-time embedded systems you are familiar with and describe how these systems meet the common definition of real-time and embedded.

I'd say the one I am most familiar with is my truck that uses a computer to control several subsystems. The real-time elements that come to mind are the ABS brake system and the traction control system since they are using functions that have to perform in real time with lives/property at stake. There is some computer that is deciding how/when to move the wheels/brakes and it has to be done as the braking or slipping is happening. I would say that these meet the deinition of embedded because they aren't actively controlled or modified by the user but operate based on input from some sensor in the brake pedal or axle.

5. Find the Liu and Layland paper and read through Section 3. Why do they make the assumption that all requests for services are periodic? Why might this be a problem with a real application?

They make the assumption that all requests are periodic so that the tasks can be expressed formulaically and characterized by request period and run-time. This could be a problem because nothing is truly periodic in a real application and the assumption may lead to some miscalculations.

6. Define hard and soft real-time services and describe why and how they are different.

The definition seems to be pretty fuzzy and depends on the importance of the deadlines being met. A hard real-time system costs lives or property if the deadlines aren't met and perhaps will be designed so that some safe thing will always be returned before the deadline no matter what. Soft real-time systems seem to more cost annoyance, if the deadline is missed then some packet doesn't get sent or performance is suboptimal. In the Liu paper it says that service must be guaranteed by the deadline in hard-real-time systems whereas a statistical distribution is acceptable in soft.