

Individual Lesson Learned

Course: System-Oriented Programming

Spring 2025

Group : 404ErrorNotFound

Edison Kurteshi

Université de Fribourg

May 21, 2025

Introduction

I started this course with almost no background in computer science, and I quickly felt overwhelmed by the material. Diving into C programming, especially pointer arithmetic and manual memory management, was much harder than I expected. From day one, the lectures and exercises pushed me well beyond my comfort zone. Knowing how little I understood, I realized that the final project would be a real challenge given my limited experience with C. Nonetheless, I was determined to work through these difficulties and improve step by step.

Technical Challenges Encountered

One of the toughest, in my opinion, concepts was grasping how pointers actually refer to specific memory locations and making sure our circular buffer never lost or corrupted any messages. We had to carefully plan a fixed-size array with wrap-around indexing so it would always keep the last 16 entries without using dynamic memory. Testing edge cases—like pushing exactly 16 messages and then one more—helped us confirm that the buffer handled overflow correctly. This process taught us the importance of thinking through every index calculation before moving on.

Bringing together the Morse interface, the timestamp system and the circular buffer was quite challenging. I had to make sure each module exchanged data correctly and stayed synchronized, even when messages arrived in quick succession. With real-time requirements, there was no margin for delay, so every function call and data handoff needed precise timing. Through this process, I learned how important it is to design components that interact reliably under strict timing constraints.

Skills Acquired

Working on this project helped me develop several key skills that are directly useful in software development:

- Structuring code into well-organized static libraries with clear header files
- Translating binary data into text
- Implementing precise parameter calibration
- Utilizing GitLab workflows effectively for version control
- Working confidently with WSL as a development environment

The calibration routine—brilliantly implemented by Mathilde—provided a concrete demonstration of how precise parameter tuning and thorough validation form the foundation of system reliability. I also learned to use GitLab and WSL—tools that I think will be really useful in our field.

Conclusion

Even though the course material was challenging, working through each module step by step has noticeably improved my skills in low-level programming, memory management, and real-time system design compared to when I started. Tackling the exercises and debugging on my own taught me how to approach problems methodically and build practical solutions.

I'm looking forward to applying these skills in future projects and feel more confident taking on new challenges. With what I've learned, I know I can handle the next technical tasks with greater ease.