Senior Project Proposal: Implementing Real-Time Scheduling in Theseus

Jacob Earle
New Haven, CT

Abstract

Theseus is an experimental operating system written entirely in Rust and meant to push the boundaries of traditional operating systems design by utilizing features of the Rust language to offload tasks often handled by the operating system onto the compiler. Although originally built for x86-based computers, recent efforts have been made to port Theseus to highly embedded systems such as ARM Cortex-M microcontrollers. As a part of these ongoing efforts, I would like to focus my senior project for the 2021 fall semester on implementing real-time scheduling for the ARM port of Theseus. [**?**]

Operating Systems, Scheduling

Introduction

In the study of operating systems, real-time scheduling is the field concerned with scheduling algorithms for tasks that have a deadline before which the task must be completed in order for the order of scheduled tasks to be considered ”correct.” Within the field of real-time scheduling, there are two main subcategories. The first of these subcategories is referred to as ”hard” real-time scheduling, in which all tasks must be completed before their deadlines with no tolerance for a late task. If any of the tasks cannot be scheduled to finish before its associated deadline, then the ordering of the scheduled tasks is considered to have failed. The second subcategory is referred to as ”soft” real-time scheduling, in which an ordering of tasks can result in some tasks completing after their deadlines. However, each task has a certain penalty associated with lateness, and as long as the total penalty across all tasks in the ordering is below a certain threshold, the ordering is considered to be correct. However, if the total penalty is above the threshold, then the ordering is considered to have failed. [**?**]

Whereas soft real-time scheduling algorithms often require complex statistical analyses to validate, many of the most common hard real-time algorithms are fairly simple to analyze and implement. Thus, this project will be more concerned with hard real-time scheduling, as a thorough exploration and implementation of soft real-time scheduling is outside the scope of a single semester-long undergraduate project.

I believe that this project is worthwhile to pursue because real-time scheduling algorithms allow for greater reliability for time-critical applications and provide a method for theoretically verifying that a set of tasks can be scheduled to execute successfully in a timely manner. Especially in embedded systems, where computing resources are meager but the cost of failure can be catastrophic, a real-time scheduler can enable developers to create efficient embedded applications with confidence.

Project Overview

I will be undertaking this project to fulfill my senior requirement according to the guidelines set out in the syllabus for CPSC 490. My advisor for this project shall be Professor Lin Zhong, and my research will be conducted as part of the Yale Efficient Computing Lab. Since I am currently purusing a joint major in Mathematics and Computer Science, I will have to make a presentation on the mathematical aspects of my project to members of the Yale mathematics faculty. The rest of this paper will focus on outlining the deliverables I hope to provide by the end of the project, as well as a list of actionable milestones that will keep me on track to creating the deliverables by the end of the semester.

Deliverables

There are two main deliverables associated with this project:

1. **Software:** A contribution to the open-source Theseus repository including a real-time scheduler for the ARM port of the operating system, as well as a system for developers of applications on top of Theseus to define periodic tasks with specified priorities and deadlines. This scheduler should be fairly robust in that it can handle tasks with specified periods, as well as aperiodic and sporadic tasks.

2. **Presentation:** As explained earlier, I am pursuing a joint major in Mathematics and Computer science, so I am required to present an approximately 15 minute presentation on the mathematical aspects of my project. In this case, my presentation will pertain particularly to explaining the proofs and analysis of the algorithms my code relies on.

Milestones

In order to stay on track to complete this senior project in a satisfactory manner, I have organized my work on the project into several weekly milestones. As the project is due at the end of reading period, which is on December 15th, 2021, there are 12 weeks to complete my senior project. The milestones are as follows:

1. **Week 1:** Read and understand the most important works on real-time scheduling, such as those of Liu and Layland [**?**], to decide upon an algorithm to implement.

2. **Week 2:** Research the implementation of periodic tasks in other embedded real-time operating systems, such as FreeRTOS.

3. **Weeks 3 - 4:** Implement a simpler real-time scheduling algorithm such as rate monotonic scheduling (RMS) that can handle simple cases with strictly periodic tasks.

4. **Weeks 5 - 6:** Test and debug implementation of simple scheduling algorithm. Devise system to let application developers specify custom periods for tasks.

5. **Weeks 7 - 10:** Implement more robust algorithm that can account for periodic and aperiodic tasks at the same time.

6. **Weeks 11 - 12:** Finalize code with proper documentation and bug fixes. Create presentation for math faculty.

However, it is important to note that as my understanding of real-time scheduling algorithms grows, these milestones may be revised to reflect the priorities of the project more accurately.

Takeaways

Through working on this senior project, I hope to not only gain knowledge on the design and analysis of real-time scheduling algorithms, but also to become more confident in working on a large-scale project. Through contributing to an open-source project, I hope my code will help future users of Theseus to develop reliable and efficient solutions to problems in the embedded domain.

Acknowledgments

This project builds on the prior work of Kevin Boos, Zhiyao Ma, Namitha Liyanage, Ramla Ijaz, and all the past contributors to Theseus.