In513 FINAL ASSIGNMENT: Real-Time Operating System Design and Implementation

M. Jasdeep SINGH

Abstract

This project focuses on the design and implementation of a Real-Time Operating System (RTOS) with the integration of various periodic and one aperiodic task. The tasks include printing status messages, temperature conversion, large integer multiplication, binary search, and an optional task simulating an aperiodic execution using the wait() function. This report will provide insights into the design decisions, the implementation details, the trouble meet along the way and an interpretation of the results.

Introduction

Real-Time Operating Systems play a crucial role in applications where tasks must meet specific timing constraints. With this project we aim to design and implement a Real-Time Operating System (RTOS) using FreeRTOS. The RTOS handles various tasks, each serving a specific purpose that we will describe a bit further.

Task Descriptions

- Task 1 - Print Status Messages (Task_1)

Periodically prints the message "working" every 200ms.

- Task 2 - Temperature Conversion (Task_2)

Converts a fixed Fahrenheit temperature value to degrees Celsius and prints the result every 300ms.

- Task 3 - Large Integer Multiplication (Task_3)

Multiplies two large integers and prints the result every 400ms.

- Task 4 - Binary Search (Task_4)

Conducts a binary search on a fixed list of 50 elements (target: 37) every 400ms and prints the result of the search.

- Task 5 - Aperiodic Task Simulation (Task_5)

Generates a random number between 1 and 10 and test If the number is 7 to simulate a sort of aperiodic task. Then it simulate a duration of the task using usleep(100000) (100ms) and a wait().

Implementation details and thinking process.

The tasks are created using FreeRTOS functions. For this I had to spend a considerable time to understand a maximum of things about how the FreeRTOS way of working, using retro-engineering based on the provided example.

Conclusion

This project demonstrates the successful design and implementation of a Real-Time Operating System using FreeRTOS. The system meets its real-time requirements, and the use of FreeRTOS facilitates efficient task management.

I wasn’t comfortable with the FreeRTOS environment, and I did my best to understand and use the system, however I’m perfectly aware that my project is far from perfect. It has been a real challenge to make sense and links between the knowledge about RTOS systems and the applications on FreeRTOS.