CANOVAS-ANDRIEU Benjamin

DIDTSCH Maël

Ln422 : Report for the Design of an RTOS

Image from*: Gl-inet.com*

Jasdeep SINGH – April 2024

Introduction

The goal in this project is to design a basic RTOS to grasp the fundamentals of real-time systems and task scheduling. We're collaborating on creating four periodic tasks that will handle various basic operations such as displaying a message, converting temperature, multiplying large integers, and executing a binary search, each with predetermined Worst-Case Execution Times (WCET) and periods. We're thoroughly analyzing the execution times of these tasks and ensuring their schedulability. Next, we plan to schedule them using Fixed Priority Scheduling in FreeRTOS.

Method

The implementation consists of the following steps:

1. **Task Definition**: Four periodic tasks were defined:
    - **Task 1**: Prints "Working" every second.
    - **Task 2**: Converts a fixed temperature from Fahrenheit to Celsius every two seconds.
    - **Task 3**: Multiplies two large integers and prints the result every three seconds.
    - **Task 4**: Performs a binary search in a fixed list of 50 elements every four seconds.
2. **WCET and Period Calculation**: The WCET and periods were determined based on the estimated execution times and required frequencies:
    - **Task 1**: $WCET = 10\,ms, Period = 1000\,ms$
    - **Task 2**: $WCET = 10\,ms, Period = 2000\,ms$
    - **Task 3**: $WCET = 50\,ms, Period = 3000\,ms$
    - **Task 4**: $WCET = 20\,ms, Period = 4000\,ms$
3. **Schedulability Analysis**: The CPU utilization was calculated to ensure that the tasks are schedulable:

$$U = 101000 + 102000 + 503000 + 204000 = 0.0367U$$
$$= 100010 + 200010 + 300050 + 400020 = 0.0367$$

Since $U = 0.0367$ is less than 1, the tasks are schedulable.

4. **Implementation in FreeRTOS**: The tasks were implemented and scheduled using FreeRTOS. The **main.c** and **FreeRTOSConfig.h** files were configured accordingly.

Results/Conclusion

After finishing the implementation part and creating the tasks in the codes, we were not able to run the project. We tried running main or just some part of the code, but nothing would work even with Ubuntu. So, we tried to explain the method and doing a little analyse even if it didn't work. Thus, this project should have demonstrated the fundamental principles of RTOS and task scheduling using FreeRTOS.