CSE4354 Real-time Operating Systems
CSE5354 Real-time Operating Systems
CSE6351 Advanced Topics in Computer Engineering
Fall 2020 Mini Project 2 (Memory Protection Unit)

1 Overview

The goal of this assignment is to introduce split provileged/unprivileged operation and provide memory protection that will be used in the RTOS project.

2 Hardware Description

Microcontroller:
An ARM M4F core (TM4C123GH6PMI microcontroller) is required.

Serial interface:
If using the EK-TM4C123GXL evaluation board, then the UART0 tx/rx pair is routed to the ICDI that provides a virtual COM port through a USB endpoint.

3.3V supply:
The circuit is powered completely from the 3.3V regulator output on the evaluation board.

3 Software Description

A virtual COM power using a 115200 baud, 8N1 protocol with no hardware handshaking.

For this design, you must set the ASP bit in the CONTROL register so that the thread code uses the PSP (the handler mode will always use the MSP).

The software must provide the following functions. As in the shell interface, you cannot call C library functions.

If a bus fault ISR occurs, display “Bus fault in process N”, where N will be a variable provided by the OS. Or now, just use a variable named pid.

If a usage fault ISR occurs, display “Usage fault in process N”, where N will be a variable provided by the OS. Or now, just use a variable named pid.

If a hard fault ISR occurs, display “Hard fault in process N”, where N will be a variable provided by the OS. Or now, just use a variable named pid. Also, provide the value of the PSP, MSP, and hard fault flags (in hex).

If an MPU fault ISR occurs, display “MPU fault in process N”, where N will be a variable provided by the OS. Or now, just use a variable named pid. Also, provide the value of the PSP, MSP, and mfault flags (in hex). Also, print the offending instruction and data addresses. Display the process stack dump (xPSR, PC, LT, R0-3, R12. Clear the MPU fault pending bit and trigger a pendsv ISR call.

If a pendsv ISR occurs, display “Pendsv in process N”. If the MPU DERR or IERR bits are set, clear them and display the message “called from MPU”.

[Optionally, add support for systick ISR and the SVCall ISR with similar messages – saves time on main project later – no extra credit]

Understand how all of these ISRs are called by code.

To configure the MPU, following these steps:

Create a full-access MPU aperature for flash with RWX access for privileged and unprivileged access.

While in privileged mode, verify the program still runs as expected when turning on MPU with flash access control.

Temporarily switch to unprivileged mode and verify that you can access flash and run code.

Revert to privileged mode before continuing.

Create a full-access MPU aperature for RAM, peripherals, and bitbanded addresses with RW access for privileged and unprivileged access.

Verify that the program still runs as expected when turning on MPU with flash, RAM and peripheral access control.

Temporarily switch to unprivileged mode and verify that you can access flash (run code) and r/w memory.

Revert to privileged mode before continuing.

Use multiple MPUs regions to cover the 32kiB SRAM (each MPU region covers 8kiB, with 8 subregions of 1kiB each. With RW access for privileged mode and no access for unprivileged mode. Disable all sub-regions.

Verify that the program still runs as expected when turning on MPU with flash, RAM and peripheral access control.

Temporarily switch to unprivileged mode and verify that you can access flash (run code) and r/w memory.

Revert to privileged mode before continuing.

Enable subregions for 31 of the blocks (only one 1kiB block of SRAM is accessible in unprivileged mode).

Verify that in privileged mode, you can still access all SRAM.

Switch to unprivileged mode and verify that you can access flash (run code) and r/w memory from the peripherals.

Verify that you can access the 1kiB subregion and are blocked from other subregions (you will see an MPU ISR with information about the errant read/write from memory).

4 Testing

Your code will be tested in the ERB 124-127 labs by the grader.

5 Deadlines

The assignment is due on the date and at the time indicated in the syllabus. This is an individual project, so do not exchange or share code with others.

6 Safety Issues

When utilizing the lab, please observe all safety rules as stated in the syllabus.

Have fun!