ECE 3574: Applied Software Design:

Introduction to FreeRTOS

Today we are going to look at FreeRTOS, a popular open-source minimal real-time OS layer for embedded devices.

- Introduction to FreeRTOS
- FreeRTOS Tasks
- FreeRTOS Memory Management
- FreeRTOS hardware independent demo

Other RTOS's:

- QNX (proprietary)
- VxWorks (proprietary)
- Windows Embedded Compact (proprietary, limited architecture support)
- Zephyr (open-source, limited architecture support)
- many others . . .

FreeRTOS is a popular RTOS for embedded systems.

- small, fits into 6-12k of ROM
- preemptive or cooperative scheduling
- provides mutexes and semaphores
- provides a message passing implementation
- can uses tasks or co-routines

The core implementation is just three source files. The API is C.

FreeRTOS Tasks

In FreeRTOS you implement *tasks*, functions that never return

```
void vATaskFunction( void *pvParameters )
{
    for( ;; )
    {
     // Task application code here.
     // typically:
     //  - set a software timer
     //  - suspend, call vTaskSuspend
    }
}
```

- In main you call `xTaskCreate` for the tasks with a priority parameter and then start the scheduler.
- tasks can communicate using message queues

FreeRTOS Memory Management

FreeRTOS abstracts memory management through functions that you can use to do your own memory management

- `pvPortMalloc()`
- `pvPortFree()`

and provides several possible heap implementations, e.g.

- `heap_1` - never free
- `heap_2` - "best fit" memory pool, no block consolidation
- `heap_3` - uses your compiler's malloc/free implemenation (wrapper)
- `heap_4` - "best fit" memory pool, block consolidation for large objects

FreeRTOS comes with many examples across several architectures

- Each architecture has it's own settings in `FreeRTOSConfig.h`
- There are also toolchain configuration settings in `portmacro.h`
- Usually copy the closest existing demo and modify it.

We will look at the Posix Simulator Examples.

Next Actions and Reminders

- The last class we will review the course, answer any questions, and discuss the format of the final
- Be sure to tag and push Milestone 4 before midnight on Wednesday!

Please, be sure to fill out the SPOT survey. If you have already, Thank You!