RTOS Overview

- Real Time Operating Systems
  - A Powerful Partitioning Tool
  - Manages Resources
  - Manages Conflicts
  - Aids in partitioning problems

Resources

- Tasking and Scheduling - Actions
- ISR Support - Hardware Interface
- Semaphores - Communication
- Mailboxes - Communication
- Queues - Communication
Actions and RTOS

Tasks, Threads and Processes

- Task = Process - Independently executing entity that may control resources. May contain multiple threads
- Thread - A path through the code - A task may have multiple threads iff each thread has own data area

Task Swapping

- Co-operative - The task gives up control
- Time-Slice - Context switcher shares time
- Preemptive - Context switcher shares time, budgeting both on fairness and dynamic task priorities
Real-Time Concepts

- Doing Hard Time - Tasks have to be performed correctly and on time
- Doing Soft Time - Tasks have to be done as fast as possible, but not on a specific time
- Most systems are mixes

Foreground/Background

- Foreground - ISR/Asynchronous events
  - (Also called Interrupt Level)
- Background - “infinite loop” that calls modules to perform desired functions
  - (Also called Task Level)

Foreground/Background Systems

Code Execution
Types of Communication

• Interprocess Communications

• OS Communications

Interprocess Communications

• Resource Allocation
  – Test-And-Set
    • First function to resource takes

Interprocess Communications

• Semaphores
  – Control Access to shared Resource
  – Signal the occurrence of an event
  – Allow two tasks to synchronize
Semaphores

• "Give me the key. If someone else is using it, I am willing to wait!"
  – Binary Semaphores
    • 0 or 1
  – Counting Semaphores
    • 0 - size of counter
  – OS must keep track of tasks waiting for availability

Semaphores

• Operations
  – INITIALIZE - CREATE
  – WAIT - PEND
  – SIGNAL - POST

Semaphores

• Task desiring resource does a WAIT (PEND)
  – Time outs are generally available
  – If Semaphore is greater than zero, task continues, semaphore decremented
• Task releasing semaphore does a SIGNAL (POST)
Semaphores

• When Semaphore is available, is given to the highest priority task in wait queue or FIFO (First In First Out)

Synchronizing Tasks using Semaphores

ISR → POST PEND TASK

TASK has done a PEND (Waits), ISR POSTS (Signals) and TASK Continues

Tasks Synchronizing Activities

POST PEND TASK

PEND POST TASK
Issues

- Everyone have a UCOSII book?

- Lab Door Codes ????

Homework Assignment

Today In History!

1959 Luna 2 launched by USSR; 1st spacecraft to impact on the Moon

Homework
- Read Chapter 2 in UCOSII Book
- Work hard on the Lab!