REAL TIME OPERATING SYSTEM PROGRAMMING-I: μC/OS-II and VxWorks

Lesson-2:
μC/OS-II Features

1. Developer

Developed by Jean J. Labrosse

- Jean J. Labrosse designed it in 1992
- µC/OS-II name derives from Micro-Controller Operating System
- Also known as MUCOS, UCOS, ..

2. Characteristics

Characteristics

- Preemptive RTOS
- Multitasking
- Deterministic
- Portable as ROM image
- Scalable
- Different Platforms support

Characteristics

- Full source code availability

www.micrium.com

- Elegantly and very well documented in the book by its designer

Basic Feature

- Scalable OS – only needed OS functions become part of the application codes
- Configuration file includes the user definitions for needed functions
- Multitasking preemptive scheduler
- Elegant code
- Is said to offer best high quality/performance ratio

Basic Feature…

- It has a pre-certifiable software component for safety critical systems, including avionics system clockA DO-178B and EUROCAE ED-12B, medical FDA 510(k), and IEC 61058 standard for transportation and nuclear systems, respectively
- Source code has been certified by Department of Defense, USA for use in Avionics and in medical applications.

Applications

- Automotive,
- avionics,
- consumer electronics,
- medical devices,
- military,
- aerospace,
- networking, and
- systems-on-a-chip.

MUCOS real time kernel additional support

- µC/BuildingBlocks [an embedded system building blocks (software components) for hardware peripherals, for example clock (µC/Clk) and LCD (µC/LCD)]
- µC/FL (an embedded flesh memory loader)
- µC/FS (an embedded memory file system)
- µC/GUI (an embedded GUI platform),
- µC/Probe (a real time monitoring tool),

MUCOS real time kernel additional support …

- µC/TCP-IP (an embedded TCP/IP stack),
- µC/CAN (an embedded Controller Area Network bus)
- µC/MOD (an embedded Modbus) and
- µC/USB device and µC/USB host (an embedded USB devices framework).

Code Language

- ‘C’ and CPU Specific Codes in Assembly

3. Source Files

Source Files

- Master Header file, which has the '#include's and Processor dependent source files and ucos.ii.h and ucos.ii.c files
- os_cfg.h — for Kernel building configuration file and
- os_cpu.h —a header file for preprocessor definitions

Source Files...

- os_tick.c — timer related codes file
- os_cpu_c. c — preprocessor C Codes
- os_cpu_a.s12 — An example of assembly codes file for 68HC12
- os_mem.c for memory functions
- os_sem.c, os_mbox.c and os_q.c for semaphores, mailbox and queues

4. Naming Basics

MUCOS Naming Basics

- OS or OS_ prefix denotes that the function or variable is a MUCOS operating system function or variable
- For examples, OSTaskCreate ( ) — a MUCOS function that creates a task,

Macros

- OS_NO_ERR– a MUCOS macro that returns true in case no error is reported from a function
- OS_MAX_TASKS– user definable constant for specifying maximum number of tasks in user application

5. MUCOS Basic Functions

MUCOS Basic Functions

- System Level – OS initiate, start, system timer set, ISR enter and exit
- Task Service Functions – create, run, suspend, resume, ..
- Task delay
- Memory allocation, partitioning, …

MUCOS Basic Functions…

- IPCs – Semaphore, Queue and Mailbox
- Same Semaphore function usable as event flag, resource key and counting semaphore
- Mailbox one message pointer per mailbox
- Queue permit array of message-pointers

Summary

We learnt

- MUCOS characteristics, applications, basic features and functions

End of Lesson 2 of Chapter 9