
uC-OS-II operating System for Applications in Space Robotics

Steven Erdmanczyk



Characteristics of Space System

Real time

Minimal task coupling

Low requirement for synchronization

Non-interface display

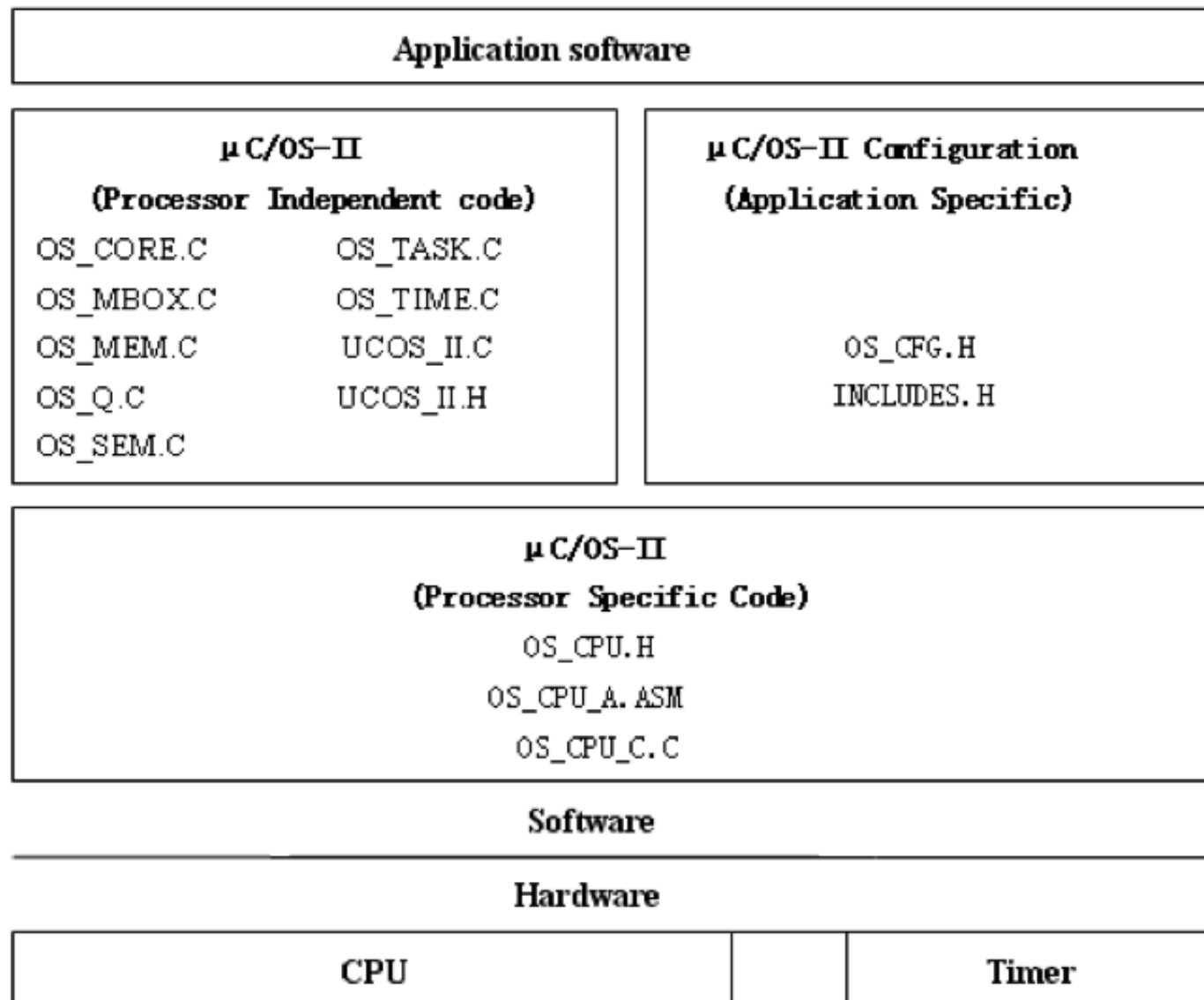


Current Systems

- PSOSystem+
- RTLinux
- QNX
- Disadvantages
 - Large
 - Task switching too slow



uC-OS-II



uC-OS-II Features

- Manages up to 64 tasks
 - 8 reserved by OS
 - 56 for user
- 64 Priority levels
- Scalable
 - Add or remove system features to minimize size
- Services
 - Mailboxes
 - Queues
 - Semaphores
 - Fixed size memory partitions
- Provides API for inter-task communication
- Uses bit tables for faster task scheduling



uC-OS-II Task Management

- Creating a Task
 - OSTaskCreate()
 - *task
 - task name
 - *pdata
 - Argument passed to task
 - *ptos
 - Points to top of stack
 - Prio
 - Desired priority

uC-OS-II Task Stacks

- Stack Space can be allocated either
 - statically
 - Dynamically
 - Malloc()

uC-OS-II ISRs

1. Should save all CPU data onto stack
2. Call OSIntEnter or OSIntNesting++
3. Clear interrupting device
4. Re-enable interrupts
5. Execute ISR code
6. Call OSIntExit()
7. Restore CPU registers
8. Return from Interrupt

uC-OS-II Memory Management

- uC-OS provides an alternative to malloc() and free()
- From fixed size memory blocks from a partition made of a contiguous memory area
 - All blocks same size
- To enable memory management
 - Set configuration constants in OS_CFG.H



uC-OS-II Memory Management

- Memory Control Block Structure
 - *OSMemAddr
 - *OSMemFreeList
 - OSMemBlkSize
 - OSMemNBlks
 - OSMemNFree



uC-OS-II Memory Management

- Creating a partition
- OSMemCreate()
 - Beginning address of partition
 - Number of blocks to allocate
 - Size in bytes of block
 - Pointer to a variable that contains an error code
- Returns a pointer to the memory block



Task Switching

